Unit 6

13 Memory Systems

13.1 : Characteristics of Memory Systems

Q.1 State the key characteristics of memory systems.

☞ [SPPU : Dec.-15, Marks 2]

Ans. : Table Q.1.1 lists the key characteristics of memory systems.

Location	:	CPU Internal (main) External (Secondary)
Capacity	:	Word size
Unit of transfer	:	Number of words Word Block
Access method	:	Sequential access Direct access Random access Associative access
Performance	:	Access time, Cycle time, Transfer rate
Physical type	:	Semiconductor Magnetic surface
Physical characteristics	:	Volatile / non-volatile
Organization	:	erasable / non-erasable

Table Q.1.1

13.2 : Memory Hierarchy

Q.2 Explain memory hierarchy.

Ans. :

- Ideally, computer memory should be fast, large and inexpensive. Unfortunately, it is impossible to meet all the three of these requirements using one type of memory.
- Increased speed and size are achieved at increased cost.
- In make efficient computer system it is not possible to rely on a single memory component, but to employ a **memory hierarchy**. Using memory hierarchy all of different types of memory units are employed to give efficient computer system. A typical memory hierarchy is illustrated in Fig. Q.2.1.
- In summary, we can say that a huge amount of cost-effective storage can be provided by magnetic disks. A large, yet affordable, main memory can be built with DRAM technology along with the cache memory to achieve better speed performance.

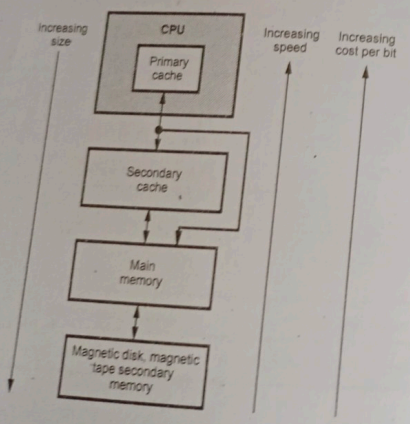


Fig. Q.2.1

13.3 : Signals to Connect Memory to Processor

Q.3 Draw the connection between memory and processor and explain how data transfer takes place between them.

Ans. : • The maximum size of the memory that can be used in any computer is determined from the number of address lines provided by the processor used in the computer. For example if processor has 20 address lines, it is capable of addressing upto 2^{20} = 1 M (Mega) memory locations. Similarly, processors whose instructions generate 32-bit address can access a memory that contains up to 2^{32} = 4 G (Giga) memory locations. The number of locations represents the size of the address space of the computer.

- The maximum number of bits that can be transferred from memory or to the memory depend on the data lines supported by the processor. The number of data lines supported by processor gives the word length of the processor and hence the computer.
- The control lines from the processor decides the memory operation. In case of read operation $\overline{RD}$ signal is activated. It is used to enable the active low output enable signal of the memory. In case of write operation $\overline{WR}$ signal is activated to indicate the write operation. The data transfer between the memory and processor takes place through the use of two processor registers, usually called **MAR** (Memory Address Register) or simply AR (Address Register) and **MDR** (Memory Data Register) or simply DR (Data Register). This is

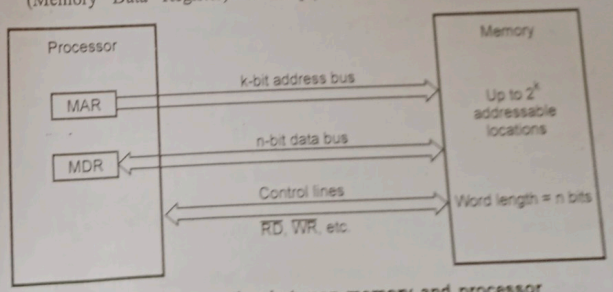


Fig. Q.3.1 Connection between memory and processor

illustrated in Fig. Q.3.1. If MAR is k-bit long and MDR is n bit long, it is possible to across upto 2^k memory locations, and during one memory cycle it is possible to transfer n-bit data.

13.4 : Memory Read and Write Cycle

Q.4 Draw and explain the memory read cycle.

Ans. : • Fig. Q.4.1 shows the timing diagram for memory read cycle.

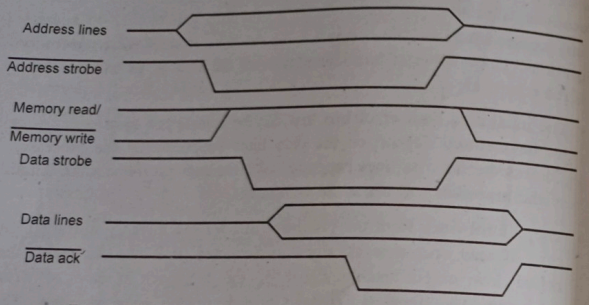


Fig. Q.4.1 Timing diagram for memory read cycle

- Processor initiates a memory read cycle by floating the address of the memory location on the address lines.
- Once the address lines are stable, the processor asserts the address strobe signal on the bus. The address strobe signals the validity of the address lines.
- Processor then sets the memory Read/$\overline{\text{Write}}$ signal to high, i.e. read cycle.
- Now the processor asserts the data strobe signal. This signals to the memory that the processor is ready to read data.
- The memory subsystem decodes the address and places the data on the data lines.
- The memory system then asserts the data acknowledge signal. This signals to the processor that valid data is available on the data bus..

- Processor latches in the data and negates the data strobe. This signals to the memory that the data has been latched by the processor.
- Processor negates the address strobe signal.
- Memory system then negates the data acknowledgement signal. This signals the end of the memory read cycle.

Q.5 Draw and explain the memory write cycle.

Ans. : Fig. Q.5.1 shows the timing diagram for memory write cycle.

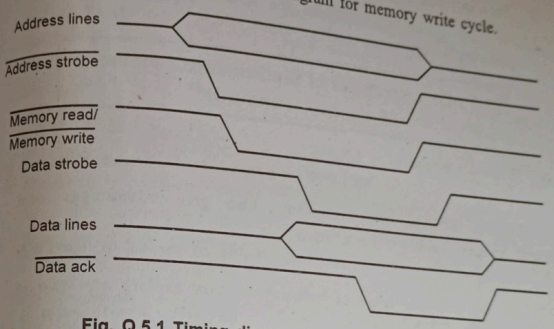


Fig. Q.5.1 Timing diagram for memory write cycle

1. Processor initiates a memory write cycle by floating the address of the memory location on the address lines.
2. Once the address lines are stable, the processor asserts the address strobe signal on the bus. The address strobe signals the validity of the address lines.
3. Processor then sets the memory Read/$\overline{\text{Write}}$ signal to low, i.e. write cycle.
4. The processor then places the data on the data lines.
5. Now the processor asserts the data strobe signal. This signals to the memory that the processor has valid data for the memory write operation.

6. The memory subsystem decodes the address and writes the data into the addressed memory location.
7. The memory system then asserts the data acknowledge signal. This signals to the processor that data has been written to the memory.
8. Then the processor negates the data strobe, signaling that the data is no longer valid.
9. Processor also negates the address strobe signal.
10. Memory system then negates the data acknowledgement signal, signaling an end to the memory write cycle.

13.5 : Characteristics of Semiconductor Memory : SRAM, DRAM and ROM

Q.6 Write a short note on SRAM.

☞[SPPU : May-06, 09, 11, 13, Dec.-06, Marks 6]

OR Explain DRAM with diagram. Also give advantages and disadvantages. Read write operation

☞[SPPU : May-06, 09, Dec.-06, 07, Marks 6]

OR List the different types of internal memory explain any two in brief.

☞[SPPU : June-15, Marks 6]

OR Draw DRAM cell and explain its read and write operation in detail.

☞[SPPU : Dec.-15, Marks 7]

Ans. : • The internal memory is a semiconductor random access memory.

- The Fig. Q.6.1 shows the classification of semiconductor memory.
- The two basic forms of semiconductor read/write memories are :
  - Static RAM (SRAM)
  - Dynamic RAM (DRAM)

Static RAM : • Memories that consists of circuits capable of retaining their state as long as power is applied are known as **static memories**.

- These are Random Access Memory (RAM) and hence combinely called **static RAM memories**.

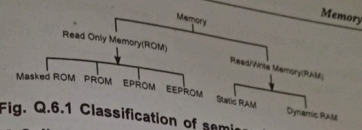


Fig. Q.6.1 Classification of semiconductor memory

Static RAM Cell :

- The Fig. Q.6.2 shows the implementation of static RAM cell. It consists of two cross-coupled inverters as a latch and two transistors T_1 and T_2 which act as a switches.

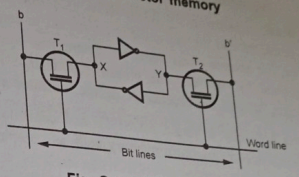


Fig. Q.6.2 Static RAM cell

- The latch is connected to two bit lines by transistors T_1 and T_2. The word line controls the opening and closing of transistors T_1 and T_2. When word line is at logic 0 level (Ground level), the transistors are off and the latch retains its state.

Read operation : • For read operation, word line is made logic 1 (high) so that both transistors are ON. Now if the cell is in state 1, the signal on bit line b is high and the signal on bit line b′ is low. The opposite is true if the cell is in state 0. The b and b′ are complements of each other. The sense/write circuits connected to the bit lines monitor the states of b and b′ and set the output accordingly.

Write operation : • For write operation, the state to be set is placed on the line b and its complement is placed on line b′ and then the word line is activated. This action forces the cell into the corresponding state and write operation is completed.

Dynamic RAMs : • Dynamic RAM stores the data as a charge on the capacitor. Fig. Q.6.3 shows the dynamic RAM cell.

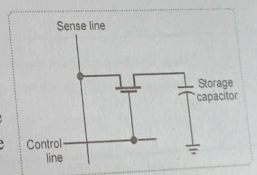


Fig. Q.6.3 Dynamic RAM

- A dynamic RAM contains thousands of such memory cells.

Write operation : • When COLUMN (Sense) and ROW (Control) lines go high, the MOSFET conducts and charges the capacitor.

- When the COLUMN and ROW lines go low, the MOSFET opens and the capacitor retains its charge. In this way, it stores 1 bit.

Read operation : • When ROW (Control) line goes high, the MOSFET conducts and the capacitor is connected to sence line. The level of charge on the memory cell capacitor determines whether that particular bit is a logical "1" or "0". The presence of charge in the capacitor indicates a logic "1" and the absence of charge indicates a logic "0".

- Since only a single MOSFET and capacitor are needed, the dynamic RAM contains more memory cells as compared to static RAM per unit area.
- The **disadvantage** of dynamic RAM is that it needs refreshing of charge on the capacitor after every few milliseconds. This complicates the system design, since it requires the extra hardware to control refreshing of dynamic RAMs.

Q.7 Compare SRAM Vs DRAM. ☞[SPPU : Dec.-08, 12, Marks 6]

Ans. :

Sr. No.	Static RAM	Dynamic RAM
1.	Static RAM contains less memory cells per unit area.	Dynamic RAM contains more memory cells as compared to static RAM per unit area.
2.	It has less access time hence faster memories.	Its access time is greater than static RAMs.
3.	Static RAM consists of number of flip-flops. Each flip-flop stores one bit.	Dynamic RAM stores the data as a charge on the capacitor. It consists of MOSFET and the capacitor for each cell.
4.	Refreshing circuitry is not required.	Refreshing circuitry is required to maintain the charge on the capacitors after every few milliseconds. Extra hardware is required to control refreshing. This makes system design complicated.
5.	Cost is more.	Cost is less.

Q.8 Draw and explain the working of ROM cell.

Ans. : • We can't write data in Read Only Memories (ROM). It is non-volatile memory i.e. it can hold data even if power is turned off.

- Generally, ROM is used to store the binary codes for the sequence of instructions and data such as look up tables. This is because this type of information does not change.
- ROMs are also accessed randomly with unique addresses.
- The Fig. Q.8.1 shows the typical configuration of a ROM cell. It consists of a transistor T and switch P.
- The transistor T is driven by the word line.
- The contents of cell can be read from the cell when word line is logic 1.

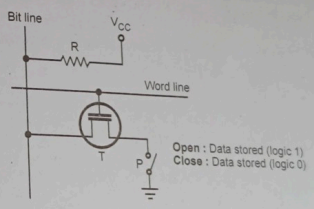


Fig. Q.8.1 ROM cell

- A logic value 0 is read if the transistor is connected to ground through switch P. If switch P is open, a logic value 1 is read.
- The bit line is connected through a resistor to the power supply.
- A sense circuit at the end of the bit line generates the proper output value.
- Data is stored into a ROM when it is manufactured.

- There are four types of ROM : Masked ROM, PROM, EPROM and EEPROM or E^2PROM.

Q.9 Write short note on : EPROM

☞[SPPU : May-06, 13, Dec.-08, 12, Marks 4]

Ans. : EPROM (Erasable Programmable Read Only Memory)

- Erasable programmable ROMs use MOS circuitry. They store 1's and 0's as a packet of charge in a buried layer of the IC chip.
- EPROMs can be programmed by the user with a special EPROM programmer.
- The important point is that we can erase the stored data in the EPROMs by exposing the chip to ultraviolet light through its quartz window for 15 to 20 minutes, as shown in the Fig. Q.9.1.

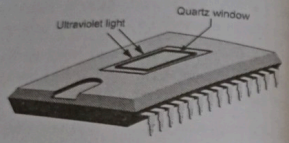


Fig. Q.9.1 EPROM

- It is not possible to erase selective information, when erased the entire information is lost.
- The chip can be reprogrammed.
- This memory is ideally suitable for product development, experimental projects and college laboratories, since this chip can be reused many times.

EPROM Programming : • When erased each cell in the EPROM contains 1. Data is introduced by selectively programming 0's into the desired bit locations. Although only 0's will be programmed, both 1's and 0's can be presented in the data.

- During programming address and data are applied to address and data pins of the EPROM. When the address and data are stable, program pulse is applied to the program input of the EPROM. The program pulse duration is around 50 ms and its amplitude depends on EPROM IC. It is typically 5.5 V to 25 V.

- In EPROM, it is possible to program any location at any time - either individually, sequentially or at random.

Q.10 Explain EEPROM. ☞[SPPU : May-06, 10, 11, Dec.-09, 10, Marks 4]

Ans. : • Electrically erasable programmable ROMs also use MOS circuitry very similar to that of EPROM.

- Data is stored as charge or no charge on an insulated layer or an insulated floating gate in the device.
- The insulating layer is made very thin (< 200 Å). Therefore, a voltage as low as 20 to 25 V can be used to move charges across the thin barrier in either direction for programming or erasing.
- EEPROM allows selective erasing at the register level rather than erasing all the information since the information can be changed by using electrical signals.
- The EEPROM memory also has a special chip erase mode by which entire chip can be erased in 10 ms. This time is quite small as compared to time required to erase EPROM. It can be erased and reprogrammed with device right in the circuit.

Disadvantage : • EEPROMs are most expensive and the least dense ROMs.

13.6 : Cache Memory

Q.11 Explain the role of cache in memory organisation.

☞[SPPU : Dec.-15, Marks 3]

OR Explain need of cache memory and direct mapping cache organization technique. ☞[SPPU : Dec.-16, Marks 6]

Ans. : • In a computer system the program which is to be executed is loaded in the main memory (DRAM). Processor then fetches the code and data from the main memory to execute the program. The DRAMs which form the main memory are slower devices. So it is necessary to insert wait states in memory read/write cycles. This reduces the speed of execution.

- To speed up the process, high speed memories such as SRAMs must be used. But considering the cost and space required for SRAMs, it is not desirable to use SRAMs to form the main memory. The solution for this problem is come out with the fact that most of the microcomputer programs work with only small sections of code and data at a particular time.
- **Definition** : The part of program (code) and data that work at a particular time is usually accessed from the SRAM memory. This is accomplished by loading the active part of code and data from main memory to SRAM memory. This small section of SRAM memory added between processor and main memory to speed up execution process is known as **cache memory.**

Q.12 Explain the cache memory system.

OR Define cache miss, hit rate, cache slot and cache line.

Ans. : • Fig. Q.12.1 shows a simplest form of cache memory system.

- A cache memory system includes a small amount of fast memory (SRAM) and a large amount of slow memory (DRAM). This system is configured to simulate a large amount of fast memory.

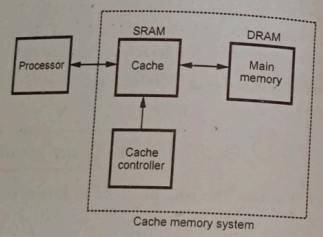


Fig. Q.12.1 Cache memory system

- Cache controller implements the cache logic. If processor finds that the addressed code or data is not available in cache - the condition referred to as **cache miss**, the desired memory block is copied from main memory to cache using cache controller. The cache controller decides which memory block should be moved in or out of the cache and in or out of main memory, based on the requirements. (The cache block is also known as **cache slot or cache line.**)

- The percentage of accesses where the processor finds the code or data word it needs in the cache memory is called the **hit rate/hit ratio**. The hit rate is normally greater than 90 percent.

$$\text{Hit rate} = \frac{\text{Number of hits}}{\text{Total number of bus cycles}} \times 100 \ \%$$

Q.13 The application program in a computer system with cache uses 1400 instruction acquisition bus cycle from cache memory and 100 from main memory. What is the hit rate ? If the cache memory operates with zero wait state and the main memory bus cycles use three wait states, what is the average number of wait states experienced during the program execution ?

Ans. : Hit rate $= \frac{1400}{1400 + 100} \times 100 = 93.3333 \ \%$

$$\text{Total wait states} = 1400 \times 0 + 100 \times 3 = 300$$

$$\text{Average wait states} = \frac{\text{Total wait states}}{\text{Number of memory bus cycles}} = \frac{300}{1500} = 0.2$$

Q.14 What is principle of locality or locality of reference ?

Ans. : • We know that program may contain a simple loop, nested loops or a few procedures that repeatedly call each other. The point is that many instructions in localized area of the program are executed repeatedly during some time period and the remainder of the program is accessed relatively infrequently. This is referred to as **locality of reference.**

- It manifests itself in two ways : **temporal** and **spatial.**
- The temporal means that a recently executed instruction is likely to be executed again very soon.
- The spatial means that instructions stored near by to the recently executed instruction are also likely to be executed soon.
- The temporal aspect of the locality of reference suggests that whenever an instruction or data is first needed, it should be brought into the cache and it should remain there until it is needed again.
- The spatial aspect suggests that instead of bringing just one instruction or data from the main memory to the cache, it is wise to bring several instructions and data items that reside at adjacent address as well. We

use the term block to refer to a set of contiguous addresses of some size.

Q.15 Explain commonly used cache organizations.

Ans. : Two most commonly used system organizations for cache memory are :

- Look-aside and
- Look-through

Look-aside system organization

- The Fig. Q.15.1 shows look-aside system of cache organization. Here, the cache and the main memory are directly connected to the system bus.

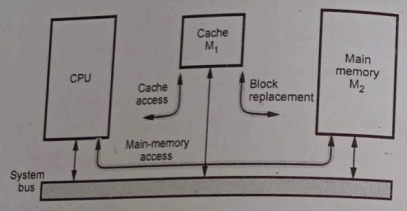


Fig. Q.15.1 Look-aside cache system organization

- In this system, the CPU initiates a memory access by placing a physical address on the memory address bus at the start of read or write cycle.
- The cache memory M_1 immediately compares physical address to the tag addresses currently residing in its tag memory. If a match is found, i.e., in case of **cache hit**, the access is completed by a read or write operation executed in the cache. The main memory is not involved in the process of read or write.
- If match is not found, i.e., in case of **cache miss**, the desired access is completed by a read or write operation directed to M_2. In response to cache miss, a block of data that includes the target address is transferred from M_2 to M_1. The system bus is used for this transfer and hence it is unavailable for other uses like I/O operations.

Look-through system organization

- Fig. Q.15.2 shows look-through system of cache organization. Here, the CPU communicates with cache via a separate (local) bus which is isolated from the main system bus. Thus during cache accesses, the system bus is available for use by other units, such as I/O controllers, to communicate with main memory.

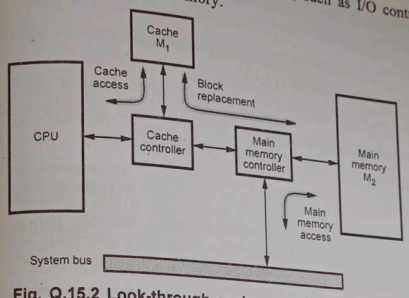


Fig. Q.15.2 Look-through cache system organization

- Unlike the look-aside system, look-through cache system does not automatically send all memory requests to main memory; it does so only after a cache miss.
- A look-through cache systems use wider local bus to link M_1 and M_2, thus speeding up cache-main-memory transfers (block transfers).
- Look-through cache system is faster.

Q.16 Explain the elements of cache design.

Ans. : • The cache design elements include cache size, mapping function, replacement algorithm write policy, block size and number of caches.

- **Cache size :** The size of the cache should be small enough so that the overall average cost per bit is close to that of main memory alone and large enough so that the overall average access time is close to that of the cache alone.
- **Mapping function :** The cache memory can store a reasonable number of blocks at any given time, but this number is small compared to the

total number of blocks in the main memory. Thus we have to use mapping functions to relate the main memory blocks and cache blocks. There are two mapping functions commonly used : direct mapping and associative mapping.

- **Replacement algorithm :** When a new block is brought into the cache, one of the existing blocks must be replaced, by a new block.
- There are four most common replacement algorithms :
  - Least-Recently-Used (LRU)
  - First-In-First-Out (FIFO)
  - Least-Frequently-Used (LFU)
  - Random
- Cache design change according to the choice of replacement algorithm.
- **Write policy :** It is also known as cache updating policy. In cache system, two copies of the same data can exist at a time, one in cache and one in main memory. If one copy is altered and other is not, two different sets of data become associated with the same address. To prevent this, the cache system has updating systems such as : write through system, buffered write through system and write-back system. The choice of cache write policy also change the design of cache.
- **Block size :** It should be optimum for cache memory system.
- **Number of caches :** When on-chip cache is insufficient, the secondary cache is used. The cache design changes as number of caches used in the system changes.

Q.17 What are the different cache mapping techniques ? Explain any one with neat diagram. ☞[SPPU : Dec.-08, Marks 8]

OR Explain direct cache mapping techniques along with its merits and demerits. ☞ [SPPU : Dec.-06, May-10, Marks 6]

OR Explain set associative cache mapping techniques along with its merits and demerits. ☞[SPPU : Dec.-06, 10, Marks 6]

OR Draw and explain :

i) Direct cache mapping ☞[SPPU : Dec.-16]

ii) Associative cache mapping ☞[SPPU : June-16]

iii) Set associative cache mapping techniques along with its merits and demerits. ☞[SPPU : Dec.-07, 09, May-11, 12, Marks 14]

OR Explain 2-way set associative cache organization. ☞[SPPU : May-14, Marks 8]

OR Explain direct mapping technique used in cache memory. ☞[SPPU : June-15, Marks 6]

OR Explain any one type of cache mapping technique with diagram. ☞[SPPU : June-17, Marks 6]

Ans. : • The mapping techniques are classified as :

1. Direct-mapping technique
2. Associative-mapping technique

▪ **Fully-associative** ▪ **Set-associative techniques.**

Direct-Mapping : • It is the simplest mapping technique.

- In this technique, each block from the main memory has only one possible location in the cache organization.
- This means that to determine whether requested word is in the cache, only tag field is necessary to be compared. This needs only one comparison.
- The main drawback of direct mapped cache is that if processor needs to access same memory locations from two different pages of the main memory frequently, the controller has to access main memory frequently. Since only one of these locations can be in the cache at a time. For example, if processor want to access memory location 100 H from page 0 and then from page 2, the cache controller has to access page 2 of the main memory. Therefore, we can say that direct-mapped cache is easy to implement, however, it is not very flexible.

Associative-Mapping (Fully-Associative Mapping) :

- Fig. Q.17.1 shows the associative-mapping technique. In this technique, a main memory block can be placed into any cache block position. As there is no fix block, the memory address has only two fields : **word** and **tag**.

- This technique gives complete freedom in choosing the cache location in which to place the memory block. Thus, the memory space in the cache can be used more efficiently.

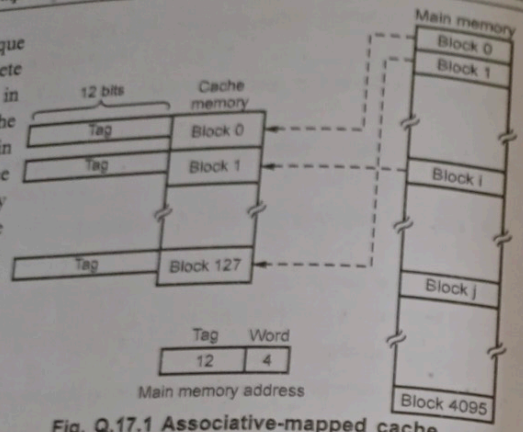


Fig. Q.17.1 Associative-mapped cache

- A new block that has to be loaded into the cache has to replace (remove) an existing block only if the cache is full.
- In such situations, it is necessary to use one of the possible replacement algorithm to select the block to be replaced.
- **Disadvantage :** In associative-mapped cache, it is necessary to compare the higher-order bits of address of the main memory with all tags corresponding to each block to determine whether a given block is in the cache. This is the main disadvantage of associative-mapped cache.

Set-Associative Mapping

- The set-associative mapping is a combination of both direct and associative mapping.
- It contains several groups of direct-mapped blocks that operate as several direct-mapped caches in parallel.
- A block of data from any page in the main memory can go into a particular block location of any direct-mapped cache. Hence the contention problem of the direct-mapped technique is eased by having a few choices for block placement.

- The required address comparisons depend on the number of direct-mapped caches in the cache system. These comparisons are always less than the comparisons required in the fully-associative mapping.
- In two-way set-associative cache, each block from main memory has two choices for block placement.
- As there are two choices, it is necessary to compare address of memory with the tag bits of corresponding two block locations of particular set. Thus for two-way set-associative cache, we require two comparisons to determine whether a given block is in the cache.
- Since there are two direct-mapped caches, any two bytes having same offset from different pages can be in the cache at a time. This **improves the hit rate** of the cache system.
- To implement set-associative cache system, the address is divided into three fields, Tag, set and word.

Q.18 Consider a cache consisting of 256 blocks of 16 words each, for a total of 4096 (4 K) words and assume that the main memory is addressable by a 16-bit address and it consists of 4 K blocks. How many bits are there in each of the TAG, BLOCK/SET and word fields for different mapping techniques ?

☞ [SPPU : May-07, Dec.-12, Marks 12]

Ans. : We know that memory address is divided into three fields. We will now find the exact bits required for each field in different mapping techniques.

a) Direct-mapping : Word bits : We know that each block consists of 16 words. Therefore, to identify each word we must have (2^4 = 16) four bit reserved for it.

Block bits : The cache memory consists of 256 blocks and using direct-mapped technique, block k of the main memory maps onto block k modulo 256 of the cache. It has one to one correspondence and requires unique address for each block. To address 128 block we require (2^8 =256) eight bits.

Tag bits : The remaining 4 (16 – 4 – 8) address bits are tag bits which stores the higher address of the main memory.

The main memory address for direct-mapping technique is divided as shown below :

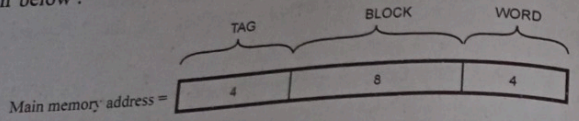


Fig. Q.18.1 (a)

b) Associative-mapping : Word bits : The word length will remain same i.e. 4 bits.

- In the associative-mapping technique, each block in the main memory is identified by the tag bits and an address received from the CPU is compared with the tag bits of each block of the cache to see if the desired block is present. Therefore, this type of technique does not have block bits, but all remaining bits (except word bits) are reserved as tag bits.

Block bits : 0 : Tag bits : To address each block in the main memory (2^{12}= 4096) 12 bits are required and therefore, there are 12 tag bits.

The main memory address for direct mapping technique is divided as shown below :

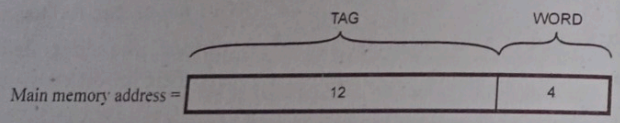


Fig. Q.18.1 (b)

c) Set-associative mapping : Let us assume that there is a 2-way set-associative mapping. Here, cache memory is mapped with two blocks per set. The set field of the address determines which set of the cache might contain the desired block.

Word bits : The word length will remain same i.e. 4 bits

Set bits : There are 128 sets (256/2). To identify each set ($2^7 = 128$) seven bits are required.

Tag bits : The remaining 5 (16 – 4 – 7) address bits are the tag bits which stores higher address of the main memory.

The main memory address for 2-way set associative mapping technique is divided as shown below :

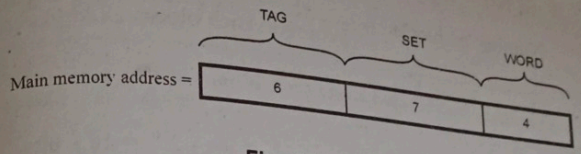


Fig. Q.18.1(c)

Q.19 A block set-associative cache consists of 64 blocks divided into 4 block sets. The main memory contains 4096 blocks, each consists of 128 words of 16 bits length :

i) How many bits are there in main memory ?

ii) How many bits are there in each of the TAG, SET and WORD fields ?

☞ [SPPU : May-05, Dec.-10, Marks 10]

Ans. : i) Number of bits in main memory :

= Number of blocks × Number of words per block × Number of bits per word

= 4096 × 128 × 16

= 8388608 bits

ii) **Number of bits in word field :** There are 128 words in each block. Therefore, to identify each word (2^7 = 128) 7 bits are required.

iii) **Number of T bits in set field :** There are 64 blocks and each set consists of 4 blocks.

Therefore, there are 16 (64/4) sets. To identify each set (2^4 = 16) four bits are required.

iv) Number of bits in tag field : The total words in the memory are :

$$4096 \times 128 = 524288.$$

To address these words we require (2^{19} = 524288) 19 address lines. Therefore, tag bits are eight (19 – 7 – 4).

Q.20 A direct mapped cache has the following parameters : cache size = 1 K words, Block size = 128 words and main memory size is 64 K words. Specify the number of bits in TAG, BLOCK and WORD in main memory address. ☞ [SPPU : May-10, Dec.-18, Marks 8]

Ans. : Word bits = $\log_2 128 = 7$-bits

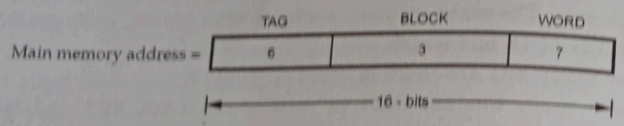
$$\text{Number of blocks} = \frac{\text{Cache size}}{\text{Words in each block}} = \frac{1K}{128} = 8$$

$\therefore$ Number of block bits = $\log_2 8 =$ **3-bits**

Number of address bits to address main memory

$= \log_2 64K = \log_2 2^{16} = 16$ - bits

$\therefore$ Tag bits = $16 - 3 - 7 =$ **6-bits**



Q.21 Give comparison between mapping techniques.

Ans. :

Sr. No.	Direct-mapping	Associative-mapping	Set-associative-mapping
1.	Each block from the main memory has only one possible location in the cache.	A block of data from main memory can be placed into any cache block position.	A block of data from main memory can go into a particular block location of any direct-mapped cache.
2.	Needs only one comparison.	Needs comparison with all tag bits.	Needs number of comparisons equal to number of blocks per set.

3.	Cache hit ratio decreases if processor needs to access same memory location from two different pages of the main memory frequently.	Cache hit ratio has no effect if processor needs to access same memory location from two different pages of the main memory frequently.	The effect of reduction in cache hit ratio in case of frequent access to the two different pages of the main memory is reduced.
4.	Main memory address is divided into three fields : TAG, BLOCK and WORD.	Main memory address is divided into two fields : TAG and WORD.	Main memory address is divided into three fields : TAG, SET and WORD.
5.	Searching time is less.	Searching time is more.	Searching time increases with number of blocks per set.

Table Q.21.1 Comparison between mapping techniques

Q.22 Write a note on cache coherency.

Ans. : • In a single CPU system, two copies of same data, one in cache memory and another in main memory may become different. This data inconsistency is called as **cache coherence problem**.

- Cache updating systems eliminates data inconsistency in the main memory caused by **cache write operations.**
- In multiprocessor systems, another bus master can take over control of the system bus. This bus master could write data into a main memory blocks which are already held in the cache of another processor. When this happens, the data in the cache no longer match those held in main memory creating inconsistency.
- There are four different approaches to prevent data inconsistency, that is to protect cache coherency :

1. Bus watching (snooping)
2. Hardware transparency
3. Non-cacheable memory
4. Cache flushing.

- **Bus watching :** In bus watching, cache controller invalidates the cache entry, if another master writes to a location in shared memory which also resides in the cache memory.
- **Hardware transparency :** In hardware transparency, accesses of all devices to the main memory are routed through the same cache or by copying all cache writes both to the main memory and to all other caches that share the same memory.
- **Non-cacheable memory :** The processor can partition its main memory into a cacheable and non-cacheable memory. By designing shared memory as non-cacheable memory cache coherency can be maintained, since shared memory is never copied into cache.
- **Cache flushing :** To avoid data inconsistency, a cache flush writes any altered data to the main memory and caches in the system are flushed before a device writes to shared memory.

Q.23 Write a detail note on MESI protocal.

Ans. : • The MESI protocol makes it possible to maintain the coherence in cached systems. It is based on the four states that a block in the cache memory can have. These four states are the abbreviations for MESI : **modified, exclusive, shared** and **invalid.** States are explained below.

- **Modified :** The line (block) in the cache, has been modified, i.e. it is different from main memory is modified and this line (block) is available only in this cache.
- **Exclusive :** The line in the cache is same as that in main memory and it is not present in any other cache.
- **Shared :** The line (block) in the cache is same as that in main memory and the same line may be present in one or more other caches.
- **Invalid :** The line (block) in the cache does not contain valid data.

- The cache coherence mechanism receives requests from both the processor and the bus, and responds to these based on the type of request such as read/write and hit/miss in the cache, and the state of the cache block specified in the request.
- The Fig. Q.23.1 (a) shows the state diagram for MESI protocol.

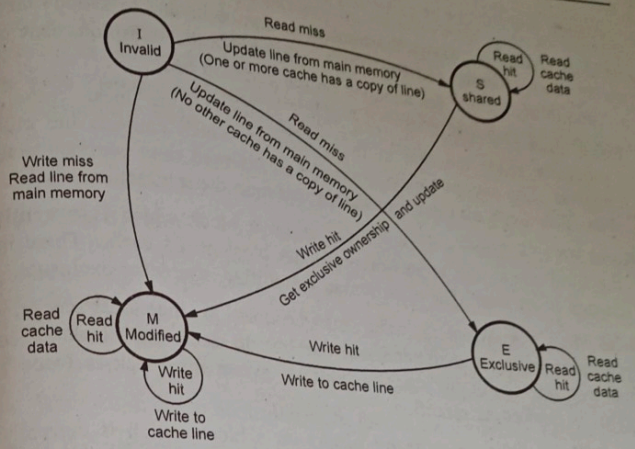


Fig. Q.23.1 (a) Transitions initiated by the processor

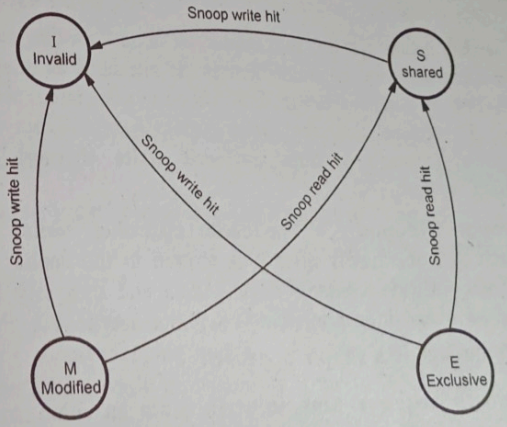


Fig. Q.23.1 (b) Transition initiated from common bus

- Each line (block) of cache has its own state bits.
- Fig. Q.23.1 (a) shows the transitions that occur due to actions initiated by the processor and Fig. Q.23.1 (b) shows the transitions that occur due to events that are snooped on the common bus.

Read Miss : • When a read miss occurs in the local cache, the processor reads the line of main memory which contains the missing address. The snoopy cache controller sends read miss command on the bus. This alerts all other processors to snoop the transaction.

Read Hit : • When a read hit occurs on a block which is currently in the local cache, the processor reads the block in its cache. There is no change in its state. The state remains modified, shared or exclusive.

Write Miss : • When a write miss occurs in the local cache, the processor gives a command on the bus to read the block of main memory containing the missing address. When the block is loaded, its status is marked as modified.

Write HIT : • When a write hit occurs on a block which is currently in the local cache, the action which is to be performed depends on the current state of that line in the local cache.

Q.24 List and explain write policies used with cache memory.

☞[SPPU : May-12, Marks 4]

Ans. : • In a cache system, two copies of same data can exist at a time, one in cache and one in main memory. If one copy is altered and other is not, two different sets of data become associated with the same address. To prevent this, the cache system has updating systems such as : **write through system, buffered write through system** and **write-back system.**

Write through Systems : • The cache controller copies data to the main memory immediately after it is written to the cache. Due to this, main memory always contains a valid data and thus any block in the cache can be overwritten immediately without data loss.

- The write through is a **simple approach**.
- This approach **requires time to write data** in main memory with increase in bus traffic.
- This in effect **reduces the system performance.**

Buffered Write through System : • In buffered write through system, the processor can start a new cycle before the write cycle to the main memory is completed. This means that the write accesses to the main memory are buffered.

- In such systems, read access which is a "cache hit" can be performed simultaneously when main memory is updated.
- However, two consecutive write operations to the main memory or read operation with cache "miss" require the processor to wait.

Write-Back System : • In a write-back system, the **alter** (update) bit in the tag field is used to keep information of the new data. If it is set, the controller copies the block to main memory before loading new data into the cache.

- Due to one time write operation, number of write cycles are reduced in write-back system. But this system has following disadvantages.
  - Write-back cache controller logic is more complex.
  - It is necessary that, all altered blocks must be written to the main memory before another device can access these blocks in main memory.
- In case of power failure, the data in the cache memory is lost, so there is no way to tell which locations of the main memory contain old data. Therefore, the main memory as well as cache must be considered volatile and provisions must be made to save the data in the cache.

Q.25 What are the different replacement algorithms ? Explain LRU algorithm in detail. ☞ [SPPU : Dec.-06, May-19, Marks 6]

OR Describe any cache replacement algorithm in short.

☞[SPPU : Dec.-15, Marks 4]

Ans. : • When a new block is brought into the cache, one of the existing blocks must be replaced, by a new block.

- In case of direct-mapping cache, we know that each block from the main memory has only one possible location in the cache, hence there is no choice. The previous data is replaced by the data from the same memory location from new page of the main memory.
- For associative and set-associative techniques, there is a choice of replacing existing block. The choice of replacement of the existing block should be such that the probability of accessing same block must

be very less. The replacement algorithms do the task of selecting the existing block which must be replaced.

- There are four most common replacement algorithms :
  - Least-Recently-Used (LRU)
  - First-In-First-Out (FIFO)
  - Least-Frequently-Used (LFU)
  - Random
- **Least-Recently-Used :** In this technique, the block in the set which has been in the cache longest with no reference to it, is selected for the replacement. Since we assume that more-recently used memory locations are more likely to be referenced again. This technique can be easily implemented in the two-way set-associative cache organization.
- **First-In-First-Out :** This technique uses same concept that stack implementation uses in the microprocessors. In this technique, the block which is first loaded in the cache amongst the present blocks in the cache is selected for the replacement.
- **Least-Frequently-Used :** In this technique, the block in the set which has the fewest references is selected for the replacement.
- **Random :** Here, there is no specific criteria for replacement of any block. The existing blocks are replaced randomly. Simulation studies have proved that random replacement algorithm provides only slightly inferior performance to algorithms just discussed.

Q.26 Calculate the average access time of memory for a computer with cache access time of 100 ns, a main memory access of 1000 ns and a hit ratio is 0.9.

Ans. : The average access time is given by,

$t_A = t_{A1} + (1 - h)\, t_{A2} = 100 \text{ ns} + (1 - 0.9) \times 1000 \times 10^{-9} = 200 \text{ ns}$

Q.27 Suppose a cache is 10 times faster than main memory and suppose that the cache can be used 90 % of the time. How much speed up do we gain by using the cache ?

Ans. : Given : Hit ratio = 90 % = 0.9.

END...

Unit 6

14 Input / Output Systems

14.1 : I/O Module

Q.1 Why does I/O devices can not be connected directly to the system bus ?

OR What is I/O module ?

Ans. : • I/O devices (peripherals) cannot be connected directly to the system bus. The reasons are discussed here.

1. A variety of peripherals with different methods of operation are available. So it would be impractical to incorporate the necessary logic within the CPU to control a range of devices.
2. The data transfer rate of peripherals is often much slower than that of the memory or CPU. So it is impractical to use the high speed system bus to communicate directly with the peripherals.
3. Generally, the peripherals used in a computer system have different data formats and word lengths than that of CPU used in it.

- So to overcome all these difficulties, it is necessary to use a module in between system bus and peripherals, called **I/O module** or **I/O system**, or **I/O interface**.

Q.2 State the functions performed by an I/O Interface.

Ans. : The functions performed by an I/O interface are :

1. Handle data transfer between much slower peripherals and CPU or memory.
2. Handle data transfer between CPU or memory and peripherals having different data formats and word lengths.
3. Match signal levels of different I/O protocols with computer signal levels.
4. Provides necessary driving capabilities - sinking and sourcing currents.

Q.3 State the difference between peripherals and CPU.

Ans. :

Sr. No.	Peripherals	CPU
1.	These are electro-mechanical and electromagnetic devices.	It is an electronic device.
2.	Data transfer rate is slower than that of the CPU.	Data transfer rate is faster than that of peripherals.
3.	Data is in form of codes.	Data is in word format.

Q.4 Draw and explain the block diagram of I/O module.

Ans. : Important blocks necessary in any I/O module are shown in Fig. Q.4.1.

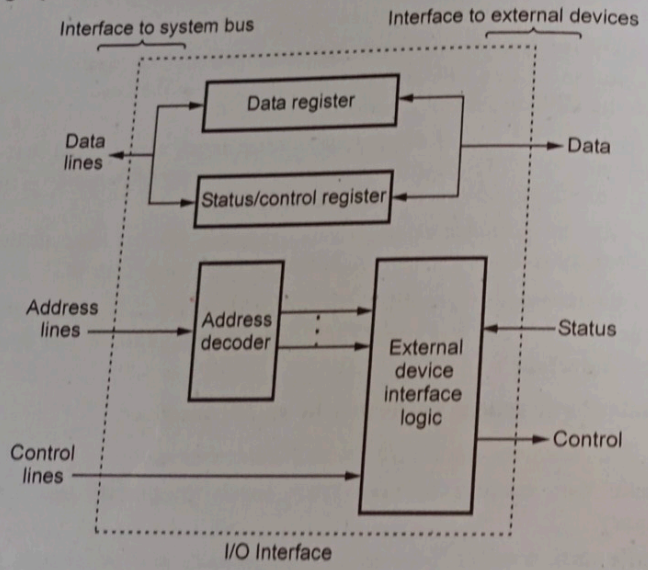


Fig. Q.4.1 Block diagram of I/O module

As shown in the Fig. Q.4.1, I/O module consists of data register, status/control register, address decoder and external device interface logic.

- The data register holds the data being transferred to or from the processor.
- The status/control register contains information relevant to the operation of the I/O device. Both data and status/control registers are connected to the data bus.
- Address lines drive the address decoder. The address decoder enables the device to recognize its address when address appears on the address lines.
- The external device interface logic accepts inputs from address decoder, processor control lines and status signal from the I/O device and generates control signals to control the direction and speed of data transfer between processor and I/O devices.

Q.5 Explain I/O interfacing techniques.

Ans. : • I/O devices can be interfaced to a computer system I/O in two ways, which are called interfacing techniques,

- Memory mapped I/O
- I/O mapped I/O

Memory mapped I/O :

- In this technique, the total memory address space is partitioned and part of this space is devoted to I/O addressing as shown in Fig. Q.5.1.
- When this technique is used, a memory reference instruction that causes data to be fetched from or stored at address specified, automatically becomes an I/O instruction if that address is made the address of an I/O port.

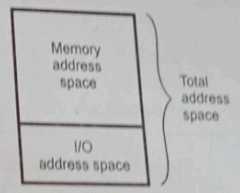


Fig. Q.5.1 Address space

Advantage : • The usual memory related instructions are used for I/O related operations. The special I/O instructions are not required.

Disadvantage : • The memory address space is reduced.

I/O mapped I/O : • If we do not want to reduce the memory address space, we allot a different I/O address space, apart from total memory space which is called I/O mapped I/O technique as shown in Fig. Q.5.2.

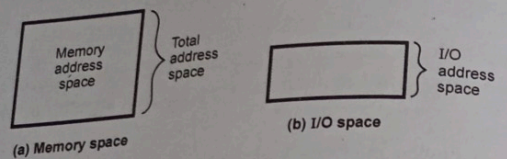


Fig. Q.5.2 Address space

Advantage : • The advantage is that the full memory address space is available.

Disadvantage : • The memory related instructions do not work. Therefore, processor can only use this mode if it has special instructions for I/O related operations such as I/O read, I/O write.

Q.6 Give comparison between memory mapped I/O and I/O mapped I/O. ☞ [SPPU : June-22, Marks 4]

Ans. :

Sr. No.	Memory mapped I/O	I/O mapped I/O
1.	Memory and I/O share the entire address range of processor.	Processor provides separate address range for memory and I/O devices.
2.	Usually, processor provides more address lines for accessing memory. Therefore more decoding is required control signals.	Usually, processor provides less address lines for accessing I/O. Therefore, less decoding is required.
3.	Memory control signals are used to control read and write I/O operations.	I/O control signals are used to control read and write I/O operations.

Q.7 Give comparison between memory and I/O bus.

Ans. :

Sr. No.	Memory bus	I/O bus
1.	Memory address bus shares entire address range.	I/O bus shares only I/O address range.
2.	Memory address bus width is greater than I/O address bus width	I/O address bus width is smaller than memory address bus width.
3.	Memory bus includes data bus, address bus and control signals to access memory.	I/O bus includes data bus, address bus and control signals to access I/O.

Q.8 State and explain different data transfer techniques. ☞ [SPPU : May-12, Marks 8]

Ans. : • In I/O data transfer, the system requires the transfer of data between external circuitry and the processor. Different ways of I/O data transfer are :

1. Program controlled I/O or polling control .
2. Interrupt program controlled I/O or interrupt driven I/O.
3. Hardware controlled I/O.
4. I/O controlled by handshake signals.

Program controlled I/O or polling control

- In program controlled I/O, the transfer of data is completely under the control of the processor program. This means that the data transfer takes place only when an I/O transfer instructions executed. In most of the cases it is necessary to check whether the device is ready for data transfer or not. To check this, processor polls the status bit associated with the I/O device.

Interrupt program controlled I/O or interrupt driven I/O

- In interrupt program controlled approach, when a peripheral is ready to transfer data, it sends an interrupt signal to the processor. This

indicates that the I/O data transfer is initiated by the external I/O device.

- When interrupted, the processor stops the execution of the program and transfers the program control to an interrupt service routine.
- This interrupt service routine performs the data transfer.
- After the data transfer, it returns control to the main program at the point it was interrupted.

Hardware controlled I/O

- To increase the speed of data transfer between processors memory and I/O, the hardware controlled I/O is used. It is commonly referred to as **Direct Memory Access (DMA).** The hardware which controls this data transfer is commonly known as **DMA controller.**
- The DMA controller sends a HOLD signal to the processor to initiate data transfer. In response to HOLD signal, processor releases its data, address and control buses to the DMA controller. Then the data transfer is controlled at high speed by the DMA controller without the intervention of the processor.
- After data transfer, DMA controller sends low on the HOLD pin, which gives the control of data, address, and control buses back to the processor.
- This type of data transfer is used for large data transfers.

I/O control by handshake signals

- The handshake signals are used to ensure the readiness of the I/O device and to synchronize the timing of the data transfer. In this data transfer, the status of handshaking signals are checked between the processor and an I/O device and when both are ready, the actual data is transferred.

14.2 : Programmed I/O and Interrupt Driven I/O

Q.9 What do you mean by modes of transfer ? Explain the following : i) Programmed I/O ii) Interrupt Initiated I/O.

☞ [SPPU : Dec.-05,09,12, May-08, June-16, Marks 7]

OR Write short notes on :
a) Programmed I/O b) Interrupt driven I/O.

Ans. : Programmed I/O

- I/O operations will mean a data transfer between an I/O device and memory or between an I/O device and the processor.
- If in any computer system I/O operations are completely controlled by the processor, then that system is said to be using '**programmed I/O**'.
- When such a technique is used, processor executes programs that initiate, direct and terminate the I/O operations, including sensing device status, sending a read or write command and transferring the data.
- It is the responsibility of the processor to periodically check the status of the I/O system until it finds that the opera tion is complete.

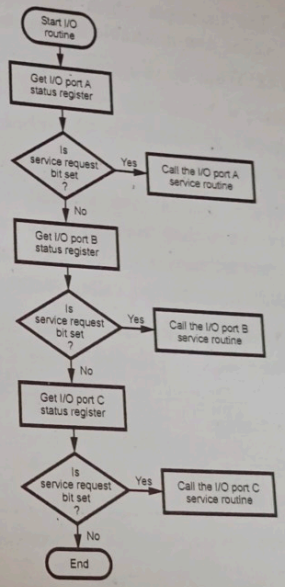


Fig. Q.9.1 Flowchart for I/O service routine

Interrupt-initiated I/O

- This method provides an external asynchronous input that would inform the processor that it should complete whatever instruction that is currently being executed and fetch a new routine (Interrupt Service Routine) that will service the requesting device.
- After this servicing is completed, the processor would resume exactly where it left off. The event that causes the interruption is called **interrupt** and the special routine executed to service the interrupt is called **Interrupt Service Routine (ISR).**

Q.10 What are hardware interrupt and software interrupts.

Ans. : • An interrupt caused by an external signal is referred as a **hardware interrupt**.

- Conditional interrupts or interrupts caused by special instructions are called **software interrupts**.

Q.11 What do you mean by maskable and non maskable interrupts ?

Ans. : • In the processor those interrupts which can be masked under software control are called **maskable interrupts**.

- The interrupts which cannot be masked under software control are called **non-maskable interrupts**.

Q.12 What do you mean by vector interrupt ? Explain.

Ans. : • When the external device interrupts the processor (interrupt request), processor has to execute interrupt service routine for servicing that interrupt.

- If the internal control circuit of the processor produces a CALL to a predetermined memory location which is the starting address of interrupt service routine, then that address is called **vector address** and such interrupts are called **vector interrupts**.
- For vector interrupts fastest and most flexible response is obtained since such an interrupt causes a direct hardware-implemented transition to the correct interrupt-handling program. This technique is called **vectoring**.
- When processor is interrupted, it reads the vector address and loads it into the PC.
- There are two ways to support vector interrupts :
  - Fixed vector address,
  - Programmable vector address.
- The processors which support fixed vector address approach have default vector address for each interrupt. The programmer cannot change this address.
- The processor which supports programmable vector address approach maintain the table in memory called the **interrupt vector table.**

- The Interrupt Vector Table (IVT) is an array of memory locations which holds the vector addresses of interrupts supported by the processor.
- When interrupt occurs the processor reads corresponding vector address given in the interrupt vector table and proceed for execution of interrupt service routine.
- The programmers are allowed to change the vector addresses stored in the interrupt vector table. Thus this approach is known as **programmable vector address**.

Q.13 What are nested interrupts.

Ans. : A system of interrupts that allows an interrupt service routine to be interrupted is known as **nested interrupts**.

Q.14 Compare programmed I/O and interrupt driven I/O.

Ans. :

Sr. No.	Programmed I/O	Interrupt driven I/O
1.	In programmed I/O, processor has to check each I/O device in sequence and in effect 'ask' each one if it needs communication with the processor. This checking is achieved by continuous polling cycle and hence processor cannot execute other instructions in sequence.	External asynchronous input is used to tell the processor that I/O device needs its service and hence processor does not have to check whether I/O device needs it service or not.
2.	During polling processor is busy and therefore, have serious and decremental effect on system throughput.	In interrupt driven I/O, the processor is allowed to execute its instructions in sequence and only stop to service I/O device when it is told to do so by the device itself. This increases system throughput.
3.	It is implemented without interrupt hardware support.	It is implemented using interrupt hardware support.

4.	It does not depend on interrupt status.	Interrupt must be enabled to proces s interrupt driven I/O.
5.	It does not need initialization of stack.	It needs initialization of stack.
6.	System throughput decreases as number of I/O devices connected in the system increases.	System throughput does not depend on number of I/O devices connected in the system.

Q.15 State the drawbacks of programmed I/O and interrupt driven I/O.

Ans. : 1.The I/O transfer rate is limited by the speed with which the CPU can test and service a device.

2. The time that the CPU spends testing I/O device status and executing a number of instructions for I/O data transfers can often be better spent on other processing tasks.

14.3 : Direct Memory Access (DMA)

Q.16 Explain the working of DMA controller.

☞ [SPPU : June-22, Marks 8]

Ans. : • DMA controlled data transfer is used for large data transfer. For example to read bulk amount data from disk to main memory.

- To read a block of data from the disk processor sends a series of commands to the disk controller device telling it to search and read the desired block of data from the disk.
- When disk controller is ready to transfer first byte of data from disk, it sends DMA request DRQ signal to the DMA controller.
- Then DMA controller sends a hold request HRQ, signal to the processor HOLD input. The processor responds this HOLD signal by floating its buses and sending out a hold acknowledge signal HLDA, to the DMA controller.
- When the DMA controller receives the HLDA signal, it takes the control of system bus.

When DMA controller gets control of the buses, it sends the memory address where the first byte of data from the disk is to be written. It

also sends a DMA acknowledge, DACK signal to the disk controller device telling it to get ready to output the byte.

- Finally, it asserts both the I/O read and memory write signals on the control bus. Asserting the I/O read signal enables the disk controller to output the byte of data from the disk on the data bus and asserting the memory write signal enables the addressed memory to accept data from the data bus. In this technique data is transferred directly from the disk controller to the memory location without passing through the processor or the DMA controller.
- Thus, the CPU is involved only at the beginning and end of the transfer.
- After completion of data transfer, the HOLD signal is deasserted to give control of all buses back to the processor.

Q.17 Give comparison between I/O program controlled transfer and DMA transfer.

Ans. :

Sr. No.	I/O program controlled transfer	DMA transfer
1.	It is software controlled data transfer	Hardware controlled data transfer.
2.	Data transfer speed is low.	Data transfer speed is high.
3.	CPU is involved in the transfer.	CPU is not involved in the transfer.
4.	Extra hardware is not required.	DMA controller is required to carry-out data transfer.
5.	During data transfer data is routed through processor	During data transfer data does not routed through processor.

Q.18 Draw and explain the block diagram of typical DMA controller.

OR Write a short note on direct memory access (DMA).

OR What is direct memory access ? Explain. Give block diagram of circuitry required for direct memory access. ☞ [SPPU : June-22, Marks 8]

Ans. : DMA stands for Direct Memory Access. It is a hardware controlled data transfer. DMA controller is used to carry-out data transfer. During data transfer data is not routed through processor.

For performing the DMA operation, the basic blocks required in a DMA channel/controller are shown in Fig. Q.18.1.

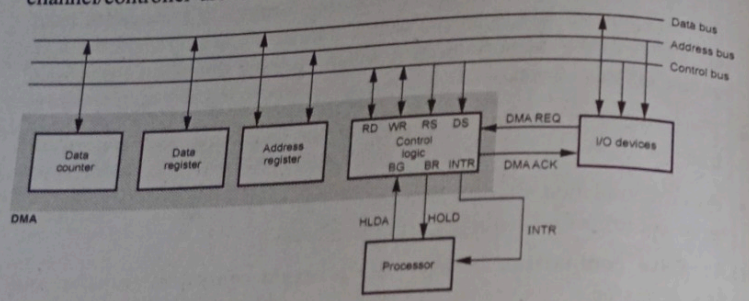


Fig. Q.18.1 Typical DMA block diagram

- DMA controller communicates with the CPU via the data bus and control lines.
- The registers in DMA are selected by the CPU through the address bus by enabling the DS (DMA select) and RS (Register select) inputs.
- The RD (Read) and WR (write) inputs are bidirectional.
- When the BG (bus grant) input is 0, the CPU can communicate with the DMA registers through the data bus to read from or write the DMA registers RD and WR signals are input signals for DMA.
- When BG = 1, the CPU has relinquished the buses and the DMA can communicate directly with the memory by specifying an address in the address bus and activating the RD or WR signals (RD and WR are now output signals for DMA). DMA consists of data count, data register, address register and control logic.
- Data counter register stores the number which gives the number data transfers to be done in one DMA cycle. It is automatically decremented after each word transfer.

- Data register acts as buffer whereas address register initially holds the starting address of the device. Actually, it stores the address of the next word to be transferred. It is automatically incremented or decremented after each word transfer.
- After each transfer, data counter is tested for zero. When the data count reaches zero, the DMA transfer halts.
- The DMA controller is normally provided with an interrupts capability, in which case it sends an interrupt to processor to signal the end of the I/O data transfer.

END...