1. **a) What is use of buffers?**

Ans: The Buffer Register prevents the high speed processor from being locked to a slow I/O device during a sequence of data transfer or reduces speed mismatch between faster and slower devices.

b) Write basic performance equation.

Ans: **T=(N*S)/R**

Where, T→Performance Parameter
R→Clock Rate in cycles/sec
N→Actual number of instruction execution
S→Average number of basic steps needed to execute one machine instruction.

c) What is an interrupt?

Ans: An interrupt is a signal to processor generated by hardware or software indicating an event that needs immediate attention.

d) How control memory works?

Ans: Control Memory is the storage in the microprogrammed control unit to store the microprogram(sequence of micro instructions).

e) What is role of micro programmed control?

Ans: A microprogrammed control unit is a relatively simple logic circuit that is capable of sequencing through microinstructions and (2) generating control signals to execute each microinstruction.

f) Explain indirect register addressing modes.

Ans: Register indirect addressing means that the location of an operand is held in a register. It is also called indexed addressing or base addressing.

g) State the meaning of locality of reference.

Ans: Many instructions in localized areas of the program are executed repeatedly during some time period. This behaviour manifests itself in two ways: temporal and spatial.

h) Define virtual memory.

Ans: virtual memory is a memory management technique creates the illusion to users of a very large main memory. Also a Technique that automatically move program and data blocks into the physical main memory when they are required for execution is called the Virtual Memory.

i) How cache memory helps to improve the performance considerations of computer.

Ans: It is used to reduce the average time to access data from the main memory. The cache is a smaller and faster memory which stores copies of the data from frequently used main memory locations.

j) What is advantage of data transfer using DMA?

Ans: High transfer rates and fewer CPU cycles for each transfer due to reduced intervention of processor.

k) What is the throughput of pipeline processor?

Ans: The number of instructions execute by pipelined processor per unit of time.

i)Define USB.

Ans: A Universal Serial Bus (**USB**) is a common interface that enables communication between devices and a host controller such as a personal computer (PC).

UNIT I

2. a) Write about basic operational concepts of computer. **6M**

Ans:

IR: The instruction register (IR) holds the instruction that is currently being executed.

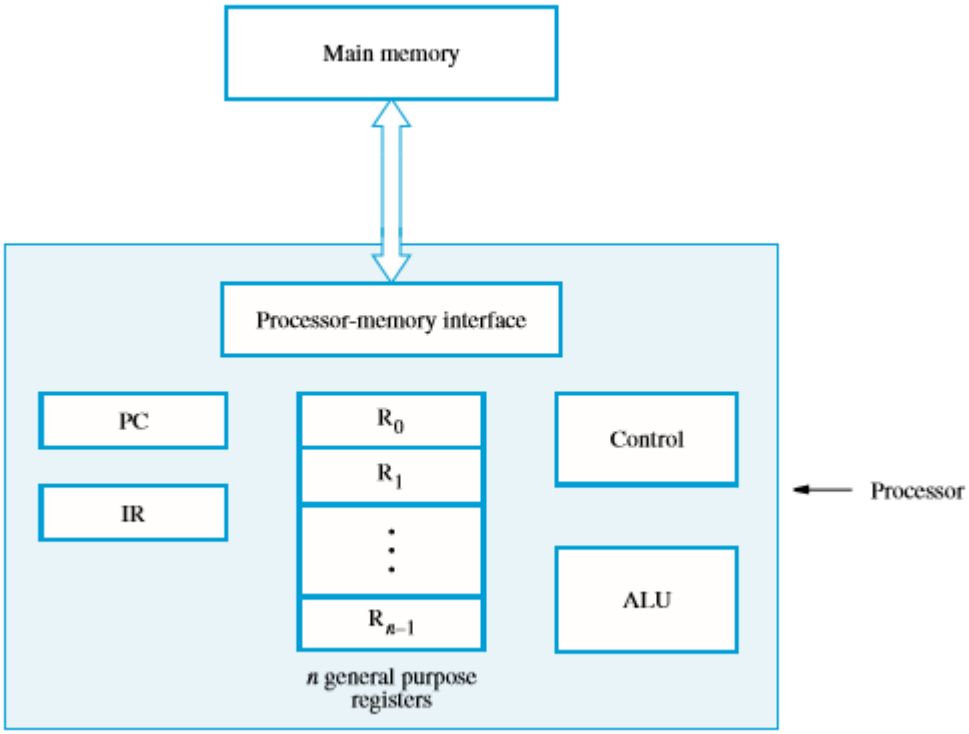
MAR: The Memory Address Register (MAR) It holds the memory location of data that needs to be accessed.

PC: It contains the memory address of the next instruction to be fetched and executed

---------→**2M**

If an operand that resides in the memory is required for an instruction, it is fetched by sending its address to the memory and initiating a Read operation. When the operand has been fetched from the memory, it is transferred to a processor register. After operands have been fetched in this way, the ALU can perform a desired arithmetic operation, such as Add, on the values in processor registers. The result is sent to a processor register. If the result is to be written into the memory with a Store instruction, it is transferred from the processor register to the memory, along with the address of the location where the result is to be stored, then a Write operation is initiated.

-------→ **2M**



---------→**2M**

2. b) Discuss about shift and rotate instructions with neat sketch. **6M**

Ans: Shift Instructions (Logical or Arithmetic) + Rotate Instructions **[3M+3M]**

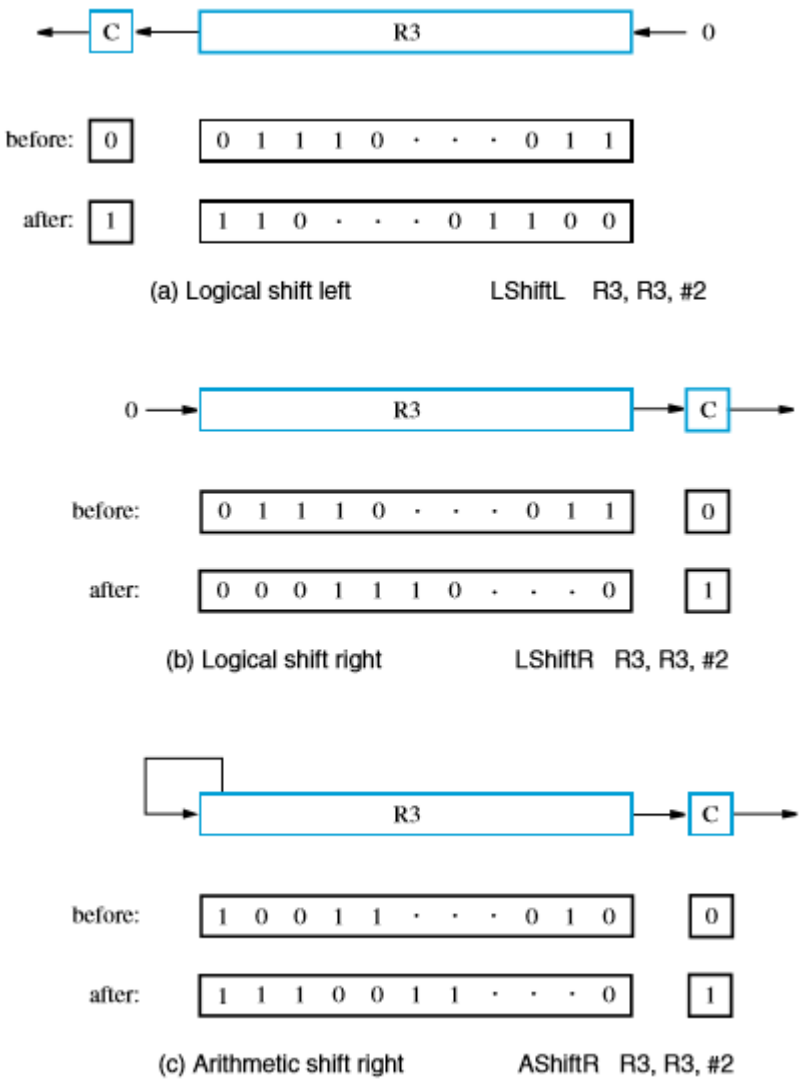
Shift and Rotate Instructions:

Logical Shifts : Two logical shift instructions are needed, one for shifting left (LShiftL) and another for shifting right (LShiftR). These instructions shift an operand over a number of bit positions specified in a count operand contained in the instruction. The general form of a Logical shift-left instruction is

LShiftL Ri,R j, count

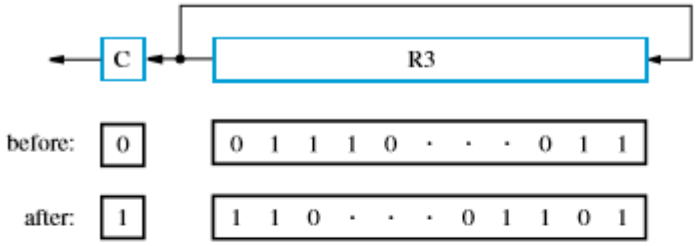
which shifts the contents of register Rj left by a number of bit positions given by the count operand, and places the result in register Ri, without changing the contents of Rj.

Arithmetic Shifts: On a right shift the sign bit must be repeated as the fill-in bit for the vacated position as a requirement of the 2's-complement representation for numbers. This requirement when shifting right distinguishes arithmetic shifts from logical shifts in which the fill-in bit is always 0.

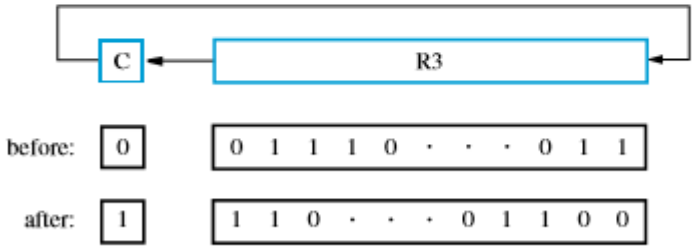


----------→**3M**

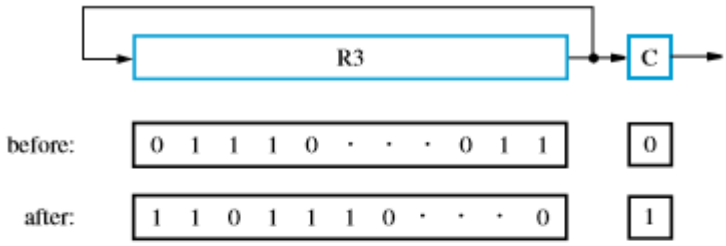
Rotate Operations: In the shift operations, the bits shifted out of the operand are lost, except for the last bit shifted out which is retained in the Carry flag C. For situations where it is desirable to preserve all of the bits, rotate instructions may be used instead.



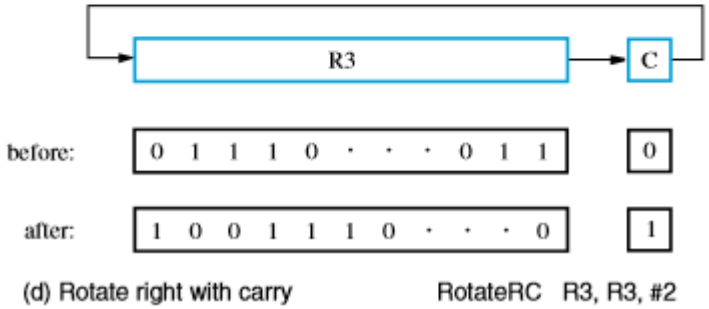
(a) Rotate left without carry RotateL R3, R3, #2



(b) Rotate left with carry RotateLC R3, R3, #2



(c) Rotate right without carry RotateR R3, R3, #2



(d) Rotate right with carry RotateRC R3, R3, #2

----------→**3M**

3. a) Demonstrate the direct and indirect addressing mode of basic computer taking a suitable example. **6M**

Ans: Direct Mode+ Indirect Mode **[3M+3M]**

Absolute Mode (Direct Mode):

The operand is in new location. The address of this location is given explicitly in the instruction.

Eg: MOVE LOC,R2

The above instruction uses the register and absolute mode. The processor register is the temporary storage where the data in the register are accessed using register mode. The absolute mode can represent global variables in the program.

Mode	Assembler Syntax	Addressing Function
Absolute mode	LOC	EA=LOC

Where EA-Effective Address

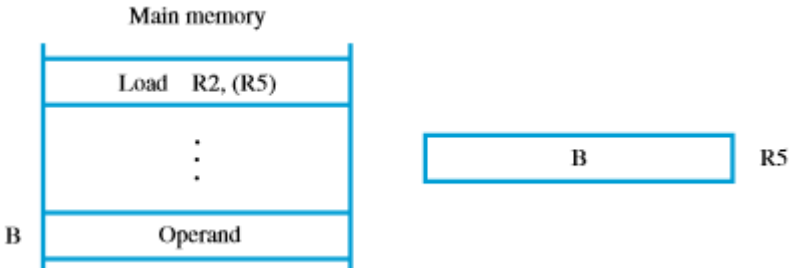
Constants:

Address and data constants can be represented in assembly language using Immediate Mode.

----------→**3M**

Indirect addressing mode:

The effective address of the operand is the contents of a register that is specified in the instruction.



Address of an operand (B) is stored into R1 register. If we want this operand, we can get it through register R1 (indirection).

The register or new location that contains the address of an operand is called the pointer.

Mode	**Assembler Syntax**	**Addressing Function**
Indirect	Ri , LOC	EA=[Ri] or EA=[LOC]

----------→**3M**

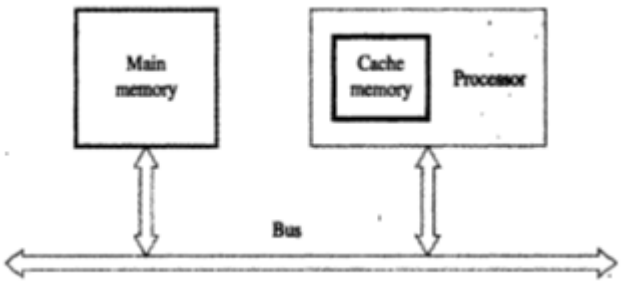
3. b) Explain about performance considerations of computer. 6M

Ans: Cache Memory+ Any Five considerations [1M+5M]

For best performance, it is necessary to design the compiler, machine instruction set and hardware in a co-ordinate way.

Elapsed Time is the total time required to execute the program is called the elapsed time. It depends on all the units in computer system.

Processor Time is the period in which the processor is active is called the processor time.
It depends on hardware involved in the execution of the instruction.



A Program will be executed faster if the movement of instruction and data between the main memory and the processor is minimized, which is achieved by using the Cache.

Cache memory------→1M

- **Processor clock:**

Clock--→The Processor circuits are controlled by a timing signal called a clock.

Clock Cycle→The cycle defines a regular time interval called clock cycle.

$$\text{Clock Rate, } R = 1/P$$

Where, P-→Length of one clock cycle.

- **Basic Performance Equation:**

$$T = (N*S)/R$$

Where, T→Performance Parameter

R→Clock Rate in cycles/sec

Clock Rate,R =1/P

N→Actual number of instruction execution

S→Average number of basic steps needed to execute one machine instruction.

To achieve high performance,

N,S<R

- **Pipelining and Superscalar operation:**

Pipelining: A Substantial improvement in performance can be achieved by overlapping the execution of successive instruction using a technique called pipelining.

Superscalar Execution: It is possible to start the execution of several instruction in every clock cycles (ie)several instruction can be executed in parallel by creating parallel paths. This mode of operation is called the Superscalar execution.

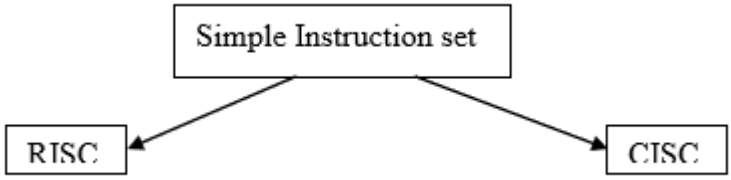
- **Clock Rate:**

There are 2 possibilities to increase the clock rate(R).They are,

Improving the integrated Chip(IC) technology makes logical circuits faster. Reduce the amount of processing done in one basic step also helps to reduce the clock period P.

- **Instruction Set: CISC AND RISC:**

The Complex instruction combined with pipelining would achieve the best performance. It is much easier to implement the efficient pipelining in processor with simple instruction set.



Reduced Instruction Set Computer)	(Complex Instruction Set Computer)
It is the design of the instruction set of a processor with simple instruction	It is the design of the instruction set of a processor with complex instruction.

- **Functions of Compiler:**

The compiler re-arranges the program instruction to achieve better performance. The high quality compiler must be closely linked to the processor architecture to reduce the total number of clock cycles.

- **Performance Measurement:**

The Performance Measure is the time it takes a computer to execute a given benchmark. A non-profit organization called SPEC (System Performance Evaluation Corporation) selects and publishes representative application program.

$$\text{SPEC rating} = \frac{\text{Running time on reference computer}}{\text{Running time on computer under test}}$$

The Overall SPEC rating for the computer is given by,

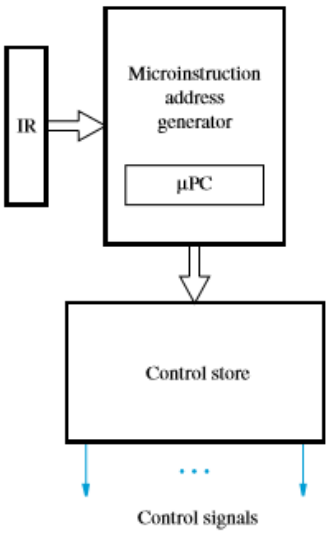
$$\text{SPEC rating} = \left(\prod_{i=1}^{n} \text{SPEC}i \right)^{1/n}$$

UNIT-II

4. a) Write about micro programmed control in detail. 6M

Ans: : Explanation+ Diagram+ control signals Grouping [2M+1M+3M]

Control signals are generated for each execution step based on the instruction in the IR. In hardwired control, these signals are generated by circuits that interpret the contents of the IR as well as the timing signals derived from a step counter. Instead of employing such circuits, it is possible to use a "software" approach, in which the desired setting of the control signals in each step is determined by a program stored in a special memory. The control program is called a micro program to distinguish it from the program being executed by the processor. The micro program is stored on the processor chip in a small and fast memory called the micro program memory or the control store.



------→**1M**

It consists of a microinstruction address generator, which generates the address to be used for reading microinstructions from the control store. The address generator uses a micro program counter, µPC, to keep track of control store addresses when reading microinstructions from successive locations.

During step 2 , the microinstruction address generator decodes the instruction in the IR to obtain the starting address of the corresponding micro routine and loads that address into the µPC. This is the address that will be used in the following clock cycle to read the control word corresponding to step 3. As execution proceeds, the microinstruction address generator increments the µPC to read micro instructions from successive locations in the control store. One bit in the micro instruction, which we will call End, is used to mark the last microinstruction in a given micro routine. When End is equal to 1, as would be the case in step 3 and step 7 , the address generator returns to the microinstruction corresponding to step 1, which causes a new machine instruction to be fetched.

--------→**2M**

Microinstruction

A simple way to structure microinstructions is to assign one bit position to each control signal required in the CPU.

Grouping of control signals --------→**2M**

Grouping technique is used to reduce the number of bits in the microinstruction. Gating signals: IN and OUT signals Control signals: Read, Write, clear A, Set carry in, continue operation, end, etc. ALU signals: Add, Sub,etc;

There are 46 signals and hence each microinstruction will have 46 bits. It is not at all necessary to use all 46 bits for every microinstruction because by using grouping of control signals we minimize number of bits for microinstruction.

Microinstruction

F1	F2	F3	F4	F5

F1 (4 bits)	F2 (3 bits)	F3 (3 bits)	F4 (4 bits)	F5 (2 bits)
0000: No transfer	000: No transfer	000: No transfer	0000: Add	00: No action
0001: PC_{out}	001: PC_{in}	001: MAR_{in}	0001: Sub	01: Read
0010: MDR_{out}	010: IR_{in}	010: MDR_{in}	⋮	10: Write
0011: Z_{out}	011: Z_{in}	011: $TEMP_{in}$		
0100: $R0_{out}$	100: $R0_{in}$	100: Y_{in}	1111: XOR	
0101: $R1_{out}$	101: $R1_{in}$		16 ALU functions	
0110: $R2_{out}$	110: $R2_{in}$			
0111: $R3_{out}$	111: $R3_{in}$			
1010: $TEMP_{out}$				
1011: $Offset_{out}$				

F6	F7	F8	...

F6 (1 bit)	F7 (1 bit)	F8 (1 bit)
0: SelectY	0: No action	0: Continue
1: Select4	1: WMFC	1: End

Techniques of grouping of control signals **---------→1M**

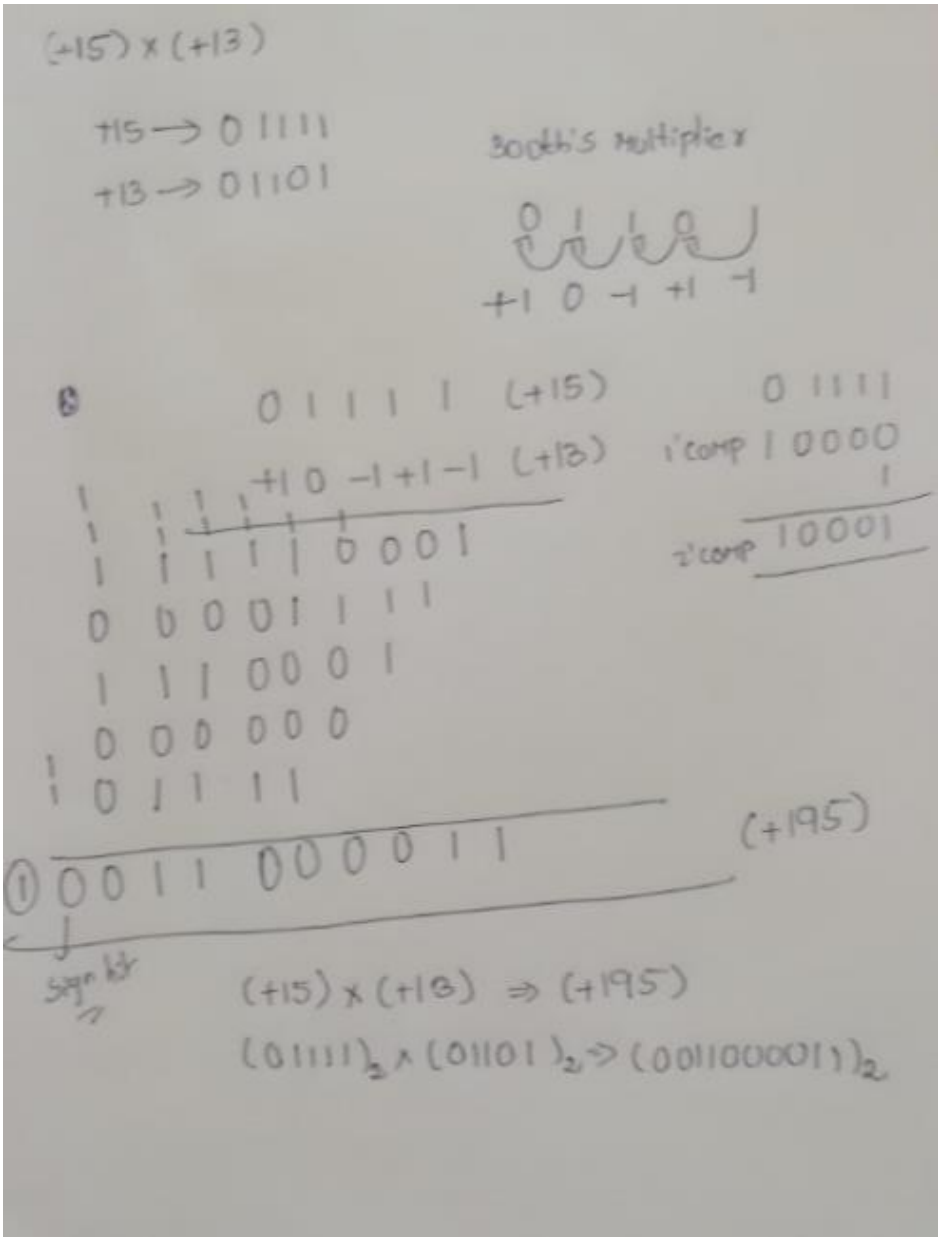
The grouping of control signal can be done either by using technique called vertical organization or by using technique called vertical organization or by using technique called horizontal organization.

Vertical organization Highly encoded scheme that can be compact codes to specify only a small number of control functions in each microinstruction are referred to as a vertical organization.

Horizontal organization The minimally encoded scheme, in which resources can be controlled with a single instruction is called a horizontal organization.

4 b) Show the step by step multiplication process using booth's algorithm. When the following binary numbers are multiplied. Assume 5-bit registers that holds signed numbers: (+15) * (+13). 6M

Ans: Procedure + Solution [5M+1M]



5.a) Explain the difference between hardwired control and micro programmed control. Is it possible to have a hardwired control associated with a control memory? 6M

Ans: Any five differences + Justification [5M+1M]

Attribute	Hardwired Control	Microprogrammed Control
Speed	Fast	Slow
Control functions	Implemented in hardware	Implemented in software
Flexibility	Not flexible to accommodate new system specifications or new instructions	More flexible, to accommodate new system specification or new instructions redesign is required
Ability to handle large/complex instruction sets	Difficult	Easier
Ability to support operating systems and diagnostic features	Very difficult	Easy
Design process	Complicated	Orderly and systematic
Applications	Mostly RISC microprocessors	Mainframes, some microprocessors
Instructionset size	Usually under 100 instructions	Usually over 100 instructions
ROM size	-	2K to 10K by 20-400 bit microinstructions
Chip area efficiency	Uses least area	Uses more area

----------→5M

Yes, it is possible. The control unit whose control signals are generated by the hardware through a sequence of instructions is called a hardwired control unit. The control logic of a hardwired control is implemented with gates, flip flops, decoders etc.

----------→1M

5. b) Explain single and double precision format of a floating point number representation.

6M

Ans: Explanation+ Representations [3M+3M]

The basic IEEE format is a 32-bit representation, shown in Figure The leftmost bit represents the sign, S, for the number. The next 8bits, E^1, represent the signed exponent of the scale factor (with an implied base of 2), and the remaining 23 bits, M, are the fractional part of the significant bits. The full 24-bit string, B, of significant bits, called the mantissa, always has a leading 1, with the binary point immediately to its right. Therefore, the mantissa

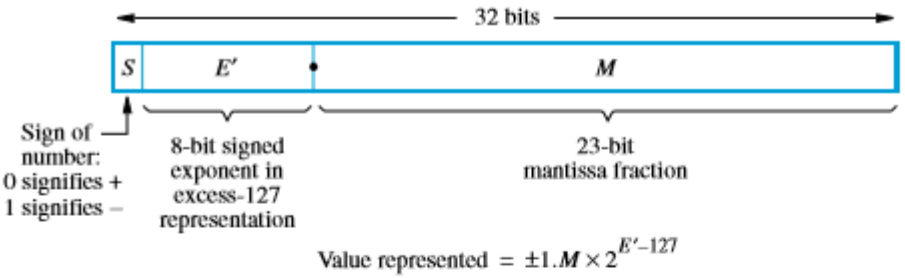
$$B=1.M =1.b-1b-2 \ldots b-23$$

Instead of the actual signed exponent, E, the value stored in the exponent field is an unsigned integer E

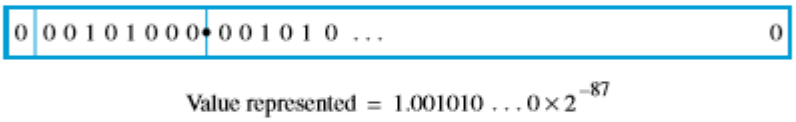
=E+127. This is called the excess-127 format. Thus, E is in the range 0≤E≤255. The end values of this range, 0 and 255.

The 32-bit standard representation in Figure (a) is called a single-precision representation because it occupies a single 32-bit word. The scale factor has a range of 2−126 to 2+127, which is approximately equal to 10±38. The 24-bit mantissa provides approximately the same precision as a 7-digit decimal value.

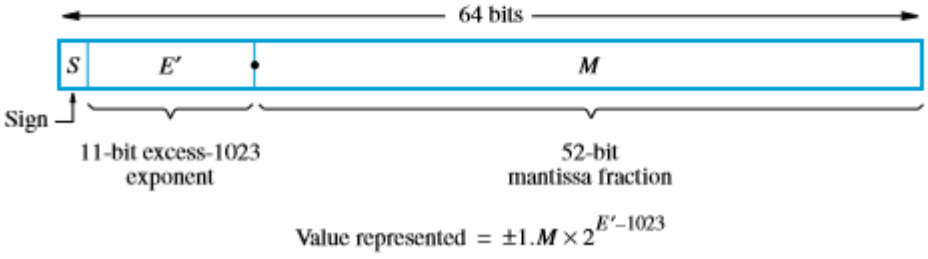
To provide more precision and range for floating-point numbers, the IEEE standard also specifies a double-precision format, as showninFigure9.26c. The double-precision format has increased exponent and mantissa ranges. The 11-bit excess-1023 exponent E has the range 1≤E≤2046 for normal values, with 0 and 2047 used to indicate special values, as before. Thus, the actual exponent E is in the range−1022≤E ≤1023, providing scale factors of 2−1022 to 21023 (approximately 10±308). The 53-bit mantissa provides a precision equivalent to about 16 decimal digits. ----------→**3M**



(a) Single precision



(b) Example of a single-precision number



(c) Double precision

----------→**3M**

UNIT-III

6. a) What are the attributes used in design of memory hierarchy? Show the memory hierarchy organization. 8M

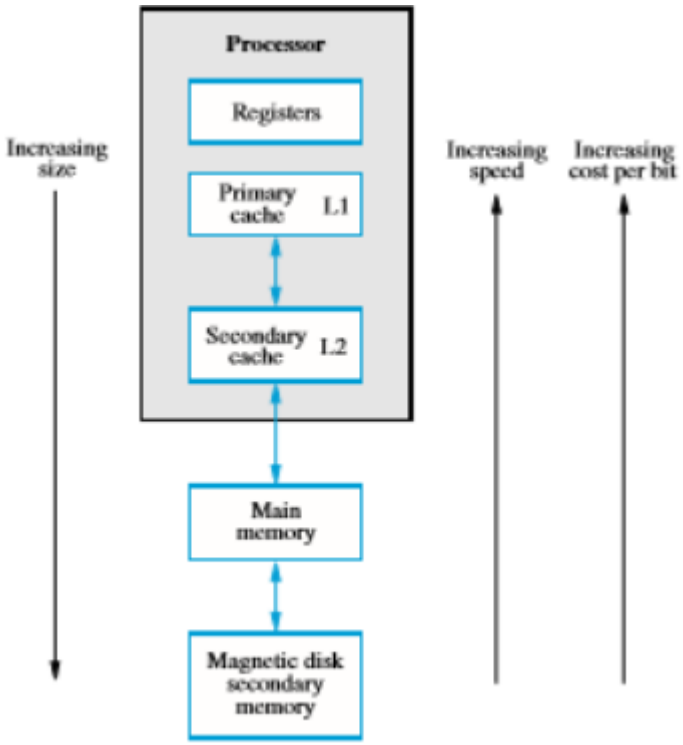
Ans: Attributes+ Diagram+ Explanation [1M+2M+5M]

The attributes used in design of memory hierarchy are:

Speed, Size and cost --------→**1M**

Although dynamic memory units with gigabyte capacities can be implemented at a reasonable cost, the affordable size is still small compared to the demands of large programs with voluminous data. A solution is provided by using secondary storage, mainly magnetic disks, to provide the required memory space. Disks are available at a reasonable cost, and they are used extensively in computer systems. However, they are much slower than semiconductor memory units.

In summary, a very large amount of cost-effective storage can be provided by magnetic disks, and a large and considerably faster, yet affordable, main memory can be built with dynamic RAM technology. This leaves the more expensive and much faster static RAM technology to be used in smaller units where speed is of the essence, such as in cache memories.



----------→**2M**

The fastest access is to data held in processor registers. Therefore, if we consider the registers to be part of the memory hierarchy, then the processor registers are at the top in terms of speed of access.

At the next level of the hierarchy is a relatively small amount of memory that can be implemented directly on the processor chip. This memory, called a processor cache, holds copies of the instructions and data stored in a much larger memory that is provided externally.

The primary cache is referred to as the level 1 (L1) cache. A larger, and hence somewhat slower, secondary cache is placed between the primary cache and the rest of the memory. It is referred to as the level 2 (L2) cache. Often, the L2 cache is also housed on the processor chip.

The next level in the hierarchy is the main memory. This is a large memory implemented using dynamic memory components, typically assembled in memory modules such as DIMMs.

Disk devices provide a very large amount of inexpensive memory, and they are widely used as secondary storage in computer systems. They are very slow compared to the main memory.

----------→**5M**

6. b) Contrast cache memory and virtual memory. **4M**

Ans: Any four differences

	Virtual Memory	Cache Memory
Definition	Virtual memory is an abstraction of the main memory. It extends the available memory of the computer by storing the inactive parts of the content RAM on a disk. It fetches it back to the RAM when the content is required.	Cache memory is used to store frequently accessed data in order to quickly access the data whenever it is required. They both are conceptually the same; however they mainly differ in the matter of implementation.
Purpose	It extends the memory capacity of a computer beyond the one that is installed.	It reduces the amount of time needed to access the data.
Speed	It operates in the millisecond range.	It operates in the nanosecond range.
Control mechanism	Managed by the operating system	Managed automatically by the hardware
Component	It is a part of the hard drive (secondary storage).	Located on the processor itself

7. a) Show the four stages of pipeline and explain how it helps to improve the performance of computer. **6M**

Ans: Pipelining diagram+ explanation+ Justification **[2M+2M+2M]**

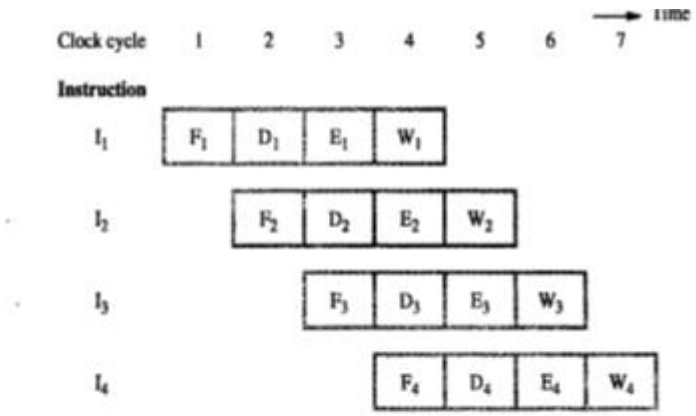
Hardware organization An inter-stage storage buffer, B1, is needed to hold the information being passed from one stage to the next. New information is loaded into this buffer at the end of each clock cycle.

F Fetch: read the instruction from the memory

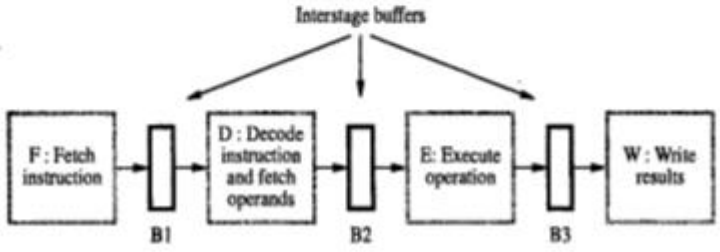
D Decode: decode the instruction and fetch the source operand(s)

E Execute: perform the operation specified by the instruction

W Write: store the result in the destination location



(a) Instruction execution divided into four steps



(b) Hardware organization

----------→**2M**

Pipelined execution:

In the first clock cycle, the fetch unit fetches an instruction I1 (step F1) and stores it in buffer B1 at the end of the clock cycle.

In the second clock cycle the instruction fetch unit proceeds with the fetch operation for instruction I2 (step F2).

Meanwhile, the execution unit performs the operation specified by instruction I1,which is available to it in buffer B1 (step E1).

By the end of the second clock cycle, the execution of instruction I1 is completed and instruction I2 is available.

Instruction I2 is stored in B1, replacing I1, which is no longer needed.

Step E2 is performed by the execution unit during the third clock cycle, while instruction I3 is being fetched by the fetch unit.

These units must be capable of performing their tasks simultaneously and without interfering with one another.

Information is passed from one unit to the next through a storage buffer.

During clock cycle 4, the information in the buffers is as follows:

Buffer B1 holds instruction I3, which was fetched in cycle 3 and is being decoded by the instruction-decoding unit.

Buffer B2 holds both the source operands for instruction I2 and the specifications of the operation to be performed. This is the information produced by the decoding hardware in cycle 3.

-The buffer also holds the information needed for the write step of instruction I2(step W2).

- Even though it is not needed by stage E, this information must be passed on to stage W in the following clock cycle to enable that stage to perform the required write operation.

Buffer B3 holds the results produced by the execution unit and the destination information for instruction I1.

----------→3M

Performance is improved with pipelined execution compared to sequential execution. In the above diagram, it takes 7 clock cycles to execute four instructions where sequential execution need more clock cycles. So, performance is improved with pipelined processors. **----------→1M**

7. b) What is Hazard? Explain about data Hazard in detail. 6M

Ans: Definition+ Effect on pipelining + Diagram [1M+3M+2M]

Def: A data hazard is any condition in which either the source or the destination operands of an instruction are not available at the time expected in the pipeline. As a result some operation has to be delayed, and the pipeline stalls.

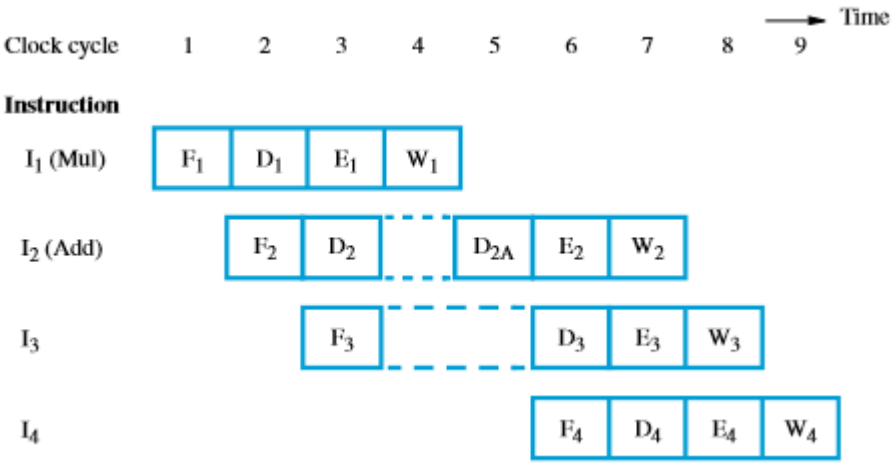
--------→1M

Assume that A=5, and consider the following two operations:

$A \leftarrow 3+A$

$B \leftarrow 4 \times A$

When these operations are performed in the order given, the result is B = 32. But if they are performed concurrently, the value of A used in computing B would be the original value, 5, leading to an incorrect result.



----------→2M

The data dependency just described arises when the destination of one instruction is used as a source in the next instruction. For example, the two instructions

Mul R2,R3,R4
Add R5,R4,R6

give rise to a data dependency. The result of the multiply instruction is placed into register R4, which in turn is one of the two source operands of the Add instruction. Assuming that the multiply operation takes one clock cycle to complete, execution would proceed as shown in Figure above. As the Decode unit decodes the Add instruction in cycle 3, it realizes that R4 is used as a source operand. Hence, the D step of that instruction cannot be completed until the W step of the multiply instruction has been completed. Completion of step D2 must be delayed to clock cycle 5, and is shown as step D2A in the figure. Instruction I3 is fetched in cycle 3, but its decoding must be delayed because step D3 cannot precede D2. Hence, pipelined execution is stalled for two cycles.

---------→3M

UNIT IV

8.a) In most computers an interrupt is recognized only after the execution of the instruction. Consider the possibility of acknowledging the interrupt at any time during the execution of the instruction. Discuss the difficulty that may arise. 6M

Ans:

We would have to save numerous pieces of system state to accomplish this:

1. The value of the instruction step counter.
2. The values in temporary registers.
3. The state of memory operations.

The CPU could probably be designed to save the step counter and temporary registers into a set of ``save'' registers, which the interrupt handler could push onto a stack and later restore to resume the instruction.

Actually, this is a common situation. Not with interrupts per se, but with page faults in virtual memory systems. Consider the situation where an instruction is fetched just fine, but that instruction references an operand that's not in main memory. A page fault trap is immediately generated to fetch the page into main memory.

8.b) What is the difference between program controlled I/O transfer and DMA transfer. 6M

Ans: Any six relevant differences

Direct Memory Access and Programmed Input/Output, DMA and PIO respectively, are two ways of transferring information in electronic devices; more famously in computers and other like devices. PIO is an older method that has since been replaced by DMA in most applications due to certain advantages. DMA is newer and better than PIO in many ways and many devices now use mainly DMA with only minimal PIO support for compatibility and to establish DMA modes.

The primary disadvantage of PIO, and the main reason for the advent of DMA, is the toll it takes on the CPU. With PIO, the CPU is responsible for moving the data from one point to another. The faster the transfer speed, the busier the CPU becomes; producing a major bottleneck in the performance of the computer. DMA does not operate the same way as PIO does. The CPU does not facilitate the transfer of information, leaving it free to do other tasks regardless of the rate of transfer of information. This means that the CPU is not a factor when it comes to considering the maximum rate of transfer.

As previously stated, some devices are capable of working with PIO and DMA despite using only DMA. PIO is used whenever there are problems with DMA. Whenever an error threshold is reached, the device automatically switches to PIO mode for a more consistent operation.

Although DMA is superior to PIO in many ways, it is still in use in many devices. The circuitry needed for a PIO controller is much simpler, therefore cheaper, when compared to DMA. In devices where a high speed of transfer is not necessary and in simple ones, it is more cost effective to use PIO rather than DMA. That is why PIO is still in use despite being bested by DMA in almost all aspects. One example of devices that still use PIO is CompactFlash. There are even new PIO modes that were designed for CompactFlash.

When choosing which mode to use with your hard drives, DMA would always produce better performance than PIO. It is better to let the system choose automatically though as it would automatically choose the better one at that your hardware can work with.

9.a) What is need for DMA? Explain the working of DMA. Also mention its advantages. 8M

Ans: DMA + Bus arbitration [4M+4M]

Direct memory access (DMA) is a method that allows an input/output (I/O) device to send or receive data directly to or from the main **memory**, bypassing the CPU to speed up **memory** operations.

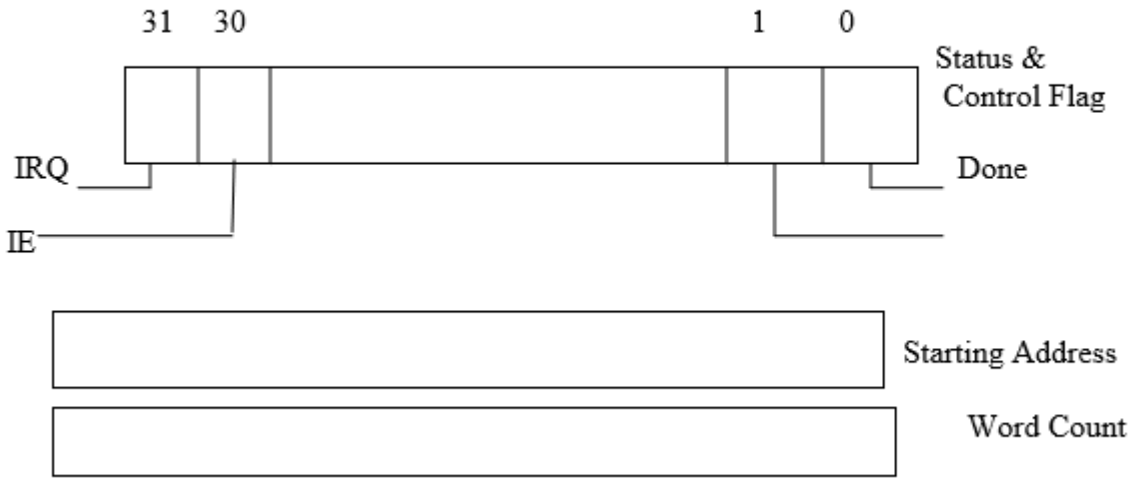
A special control unit may be provided to allow the transfer of large block of data at high speed directly between the external device and main memory , without continuous intervention by the processor. This approach is called DMA. DMA transfers are performed by a control circuit called the DMA Controller. To initiate the transfer of a block of words , the processor sends,

- Starting address
- Number of words in the block
- Direction of transfer.

When a block of data is transferred , the DMA controller increment the memory address for successive words and keep track of number of words and it also informs the processor by raising an interrupt signal.

While DMA control is taking place, the program requested the transfer cannot continue and the processor can be used to execute another program. After DMA transfer is completed, the processor returns to the program that requested the transfer. ----------→**2M**

Fig:Registes in a DMA Interface



----------→**1M**

Cycle Stealing:

Requests by DMA devices for using the bus are having higher priority than processor requests . Top priority is given to high speed peripherals such as , □ Disk □ High speed Network Interface and Graphics display device.

Since the processor originates most memory access cycles, the DMA controller can be said to steal the memory cycles from the processor. This interviewing technique is called Cycle stealing.

Burst Mode: The DMA controller may be given exclusive access to the main memory to transfer a block of data without interruption. This is known as Burst/Block Mode

Bus Master: The device that is allowed to initiate data transfers on the bus at any given time is called the bus master. --------→**1M**

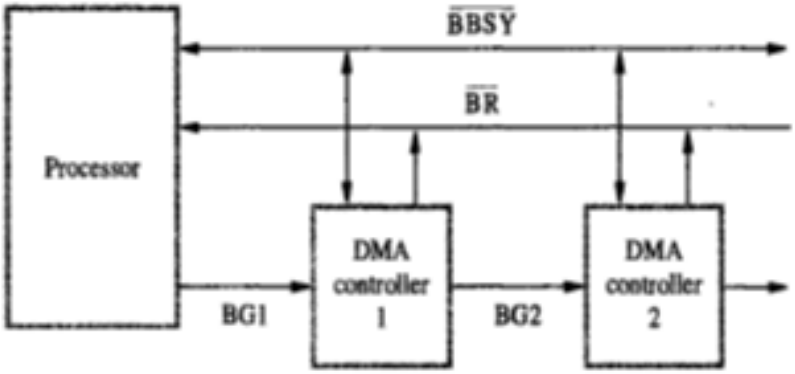
Bus Arbitration:

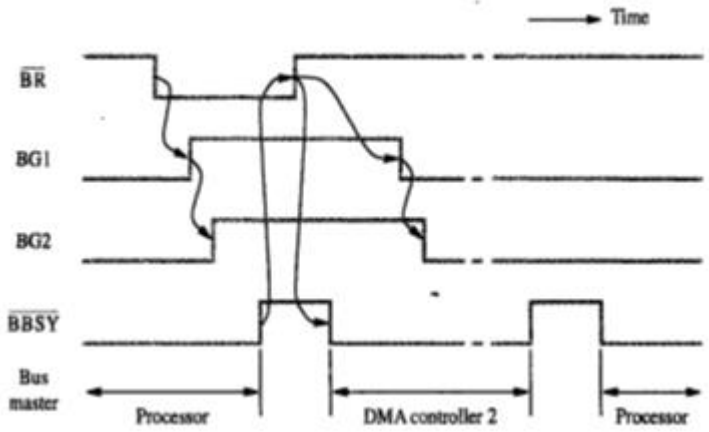
----------→**4M**

It is the process by which the next device to become the bus master is selected and the bus mastership is transferred to it. Types: There are 2 approaches to bus arbitration. They are, □ Centralized arbitration (A single bus arbiter performs arbitration) □ Distributed arbitration (all devices participate in the selection of next bus master).

Centralized Arbitration:

Here the processor is the bus master and it may grants bus mastership to one of its DMA controller. A DMA controller indicates that it needs to become the bus master by activating the Bus Request line (BR) which is an open drain line. The signal on BR is the logical OR of the bus request from all devices connected to it. When BR is activated the processor activates the Bus Grant Signal (BGI) and indicated the DMA controller that they may use the bus when it becomes free. This signal is connected to all devices using a daisy chain arrangement. If DMA requests the bus, it blocks the propagation of Grant Signal to other devices and it indicates to all devices that it is using the bus by activating open collector line, Bus Busy (BBSY).

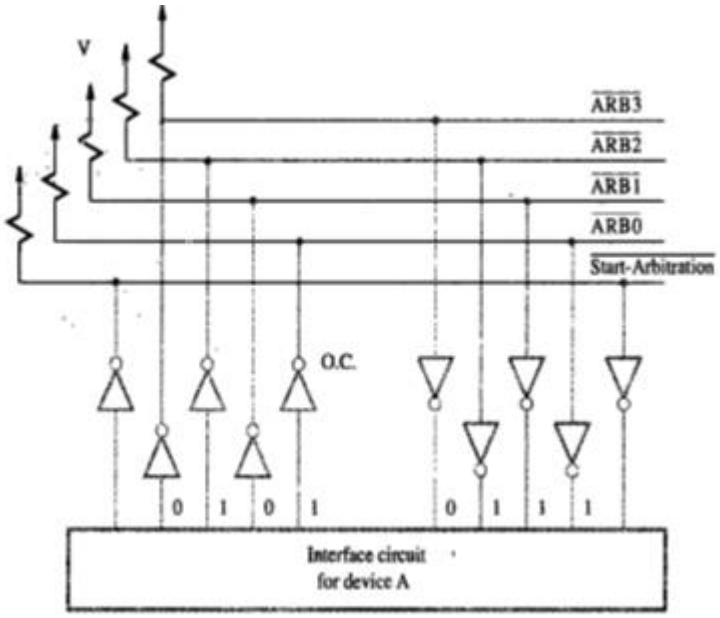




The timing diagram shows the sequence of events for the devices connected to the processor is shown. DMA controller 2 requests and acquires bus mastership and later releases the bus. During its tenture as bus master, it may perform one or more data transfer. After it releases the bus, the processor resources bus mastership

Distributed Arbitration:

It means that all devices waiting to use the bus have equal responsibility in carrying out the arbitration process.



Each device on the bus is assigned a 4 bit id. When one or more devices request the bus, they assert the Start-Arbitration signal & place their 4 bit ID number on four open collector lines, ARB0 to ARB3. A winner is selected as a result of the interaction among the signals transmitted over these lines.

The net outcome is that the code on the four lines represents the request that has the highest ID number. The drivers are of open collector type. Hence, if the i/p to one driver is equal to 1, the i/p to another driver connected to the same bus line is equal to „0"(ie. bus the is in low-voltage state).

Eg:

Assume two devices A & B have their ID 5 (0101), 6(0110) and their code is 0111. Each devices compares the pattern on the arbitration line to its own ID starting from MSB. If it detects a difference at any bit position, it disables the drivers at that bit position. It does this by placing „0" at the i/p of these drivers. In our eg. „A" detects a difference in line ARB1, hence it disables the drivers on lines ARB1 & ARB0. This causes the pattern on the arbitration line to change to 0110 which means that „B" has won the contention.

9.b) what are the events generated by SCSI controller after receive command from processor. 4M

Ans:

Consider the disk read operation, it has the following sequence of events.

The SCSI controller acting as initiator, contends process, it selects the target controller & hands over control of the bus to it.

The target starts an output operation, in response to this the initiator sends a command specifying the required read operation.

The target that it needs to perform a disk seek operation, sends a message to the initiator indicating that it will temporarily suspends the connection between them.

Then it releases the bus.

The target controller sends a command to disk drive to move the read head to the first sector involved in the requested read in a data buffer. When it is ready to begin transferring data to initiator, the target requests control of the bus. After it wins arbitration, it reselects the initiator controller, thus restoring the suspended connection.

The target transfers the controls of the data buffer to the initiator & then suspends the connection again. Data are transferred either 8 (or) 16 bits in parallel depending on the width of the bus.

The target controller sends a command to the disk drive to perform another seek operation. Then it transfers the contents of second disk sector to the initiator. At the end of this transfer, the logical connection b/w the two controller is terminated.

As the initiator controller receives the data, if stores them into main memory using DMA approach.

The SCSI controller sends an interrupt to the processor to inform it that the requested operation has been completed.

Scheme prepared by Signature of HOD,IT Dept.

Paper Evaluators:

S NO	Name of the college	Name of Examiner	signature