KLS Vishwanathrao Deshpande Institute of Technology

(Accredited by NAAC with "A" Grade)

(Approved by AICTE, New Delhi, Affiliated to VTU, Belagavi)
(Recognized Under Section 2(f) by UGC, New Delhi)
Udyog Vidya Nagar, Haliyal - 581 329, Dist.: Uttara Kannada
www.klsvdit.edu.in | principal@klsvdit.edu.in | hodece@klsvdit.edu.in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

University / Model Question Paper

Scheme & Solution

Faculty Name	:	Prof. RAGHAVENDRA N
Course Name	:	Computer Organization & Architecture
Course Code	:	BEC306C
Year of Question Paper	:	Dec 2024 / Jan 2025
Date of Submission	:	04/07/2025

Faculty Member

HoD
04.07.2025

Dean (Acad.)

CBCS SCHEME

USN | | | | | | | | | | |

BEC306C

Third Semester B.E./B.Tech. Degree Examination, Dec.2024/Jan.2025
Computer Organization and Architecture

Time: 3 hrs. Max. Marks: 100

Note: 1. Answer any FIVE full questions, choosing ONE full question from each module.
2. M : Marks , L: Bloom's level , C: Course outcomes.

			M	L	C
		Module – 1			
Q.1	a.	With neat diagram explain connection between the processor and memory.	10	L1	CO1
	b.	Write the difference between little endian and big endian memory assignments.	05	L1	CO1
	c.	Write a short note on basic performance equation.	05	L1	CO1
		OR			
Q.2	a.	Describe the concept of branching with an example program of instruction execution.	10	L1	CO1
	b.	Represent the following decimal values as signed 7-bit numbers using sign and magnitude, signed 1's complement and signed 2's complement formats. – 55, +51, 8, – 27, – 39, +43, – 10, 62	05	L2	CO1
	c.	Write a short note on memory operations.	05	L1	CO1
		Module – 2			
Q.3	a.	What is an addressing mode? Explain any four types of addressing modes, with suitable example.	10	L1	CO2
	b.	Write a program to compute the sum of test scores of all the students in the three tests. Store the corresponding sums in memory.	10	L2	CO2
		OR			
Q.4	a.	Explain the Rotate and Shift instructions with an example.	10	L1	CO2
	b.	Define subroutine. Explain subroutine linkage using a link register.	05	L1	CO2
	c.	What are assembler directives? Explain any two directives.	05	L1	CO2
		Module – 3			
Q.5	a.	Define I/O interface? Explain I/O interface to connect an input device to the bus with neat diagram.	10	L1	CO3
	b.	What is interrupt? Discuss interrupt I/O method for data transfer.	05	L1	CO3
	c.	Describe two methods of handling multiple devices.	05	L1	CO3
		OR			
Q.6	a.	Explain the use of DMA controllers in a computer system with neat diagram.	10	L1	CO3
	b.	Write a note on Bus Arbitration.	10	L1	CO3
		Module – 4			
Q.7	a.	Explain the organization of 1K×1 memory chip.	10	L1	CO4
	b.	Write a note on : (i) Static memories (ii) Cache memory	10	L1	CO4
		OR			
Q.8	a.	Explain the Magnetic disk principles.	10	L1	CO4
	b.	Draw and explain the internal organization of 2M×8 asynchronous DRAM chip.	10	L2	CO4
		Module – 5			
Q.9	a.	Discuss with neat diagram the single bus organization of data path inside a processor.	10	L1	CO5
	b.	What are the actions required to execute a complete instruction ADD (R_2), R_1	10	L1	CO5
		OR			
Q.10	a.	Draw and explain multiple bus organization of CPU.	10	L1	CO5
	b.	Draw and explain organization of the control unit to allow conditional branching in the microprogram.	10	L1	CO5

* * * * *

Head of the Department
Dept. of Electronic & Communication Engg.
KLS V.D.I.T., HALIYAL (U.K.)

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGG. KLS VDIT HALIYAL

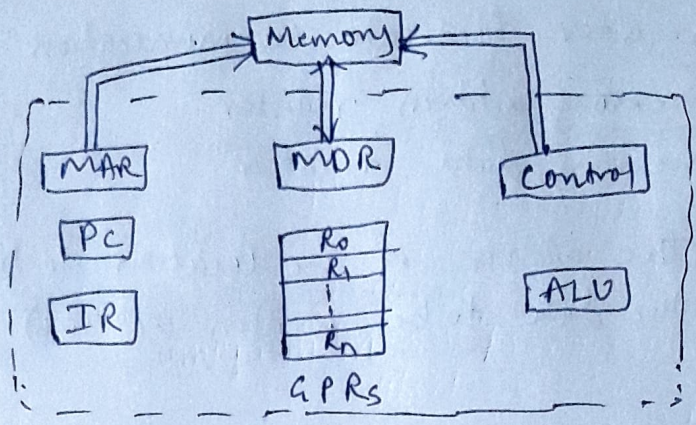
Computer Organization and Architecture

Dec 2024 / Jan. 2025

Module-1

Q.1 a) With neat diagram explain connection between the Processor & Memory.

Ans:



3M

- The computer accepts information in the form of programs & data through an I/p unit and stores it in memory
- Information stored in the memory is fetched, under program control, into an ALU where it is processed
- Processed information leaves the computer through an output unit
- All activities inside the machine are directed by control unit.
- In addition to the ALU & CU, processor contains a number of registers used for different purposes.
- IR - Instruction register holds the instruction that is currently being executed.
- GPRs - General purpose registers are used to hold data or intermediate results.

Head of the Department
Dept. of Electronic & Communication Engg.
KLS V.D.I.T., HALIYAL (U.K.)

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGG. KLS VDIT HALIYAL

- PC - Program Counter keeps track of the instruction from execution of a program. It contains the memory address of the next instruction to be fetch & executed.
- It contains 'n' general purpose registers which is used to store data and intermediate result during instruction execution.
- Two major register facilitate communication with memory.
  1) MAR - memory address register
  2) MDR - memory data register.
- MAR holds the address of the location to be accessed. 4M
- MDR holds the data to be written into (or) read out from the memory.

Operation

- Program is stored in memory, PC is set to point to the 1st instruction of program. The content of PC transferred to MAR & read control signal is sent to the memory, the information is read out of memory & loaded into the MDR. Further the content of MDR is transferred to IR register.
- If the operation needs to be performed by ALU it is necessary to obtain the required operands. Operands may be in memory or it may be available in general purpose register. If operand is in memory, then address of that memory is placed in MAR and once, the read signal is issued then operand is fetched from the data memory & placed in MDR & finally it is transferred from MDR to ALU.
- If the result of this operation needs to be stored in memory, then result is sent to MDR. The address of memory location, where data needs to be stored is placed in MAR & write signal is initiated. At the

Some time the content of PC is memorized, so that PC points to next instruction. 3M

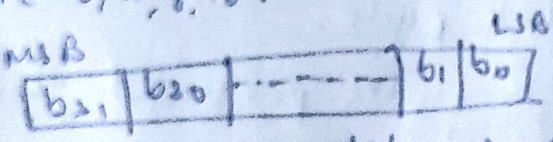
Q.1.b) Write the difference between Little endian and big endian memory assignments.

Ans There are two ways that byte addresses are assigned across word address

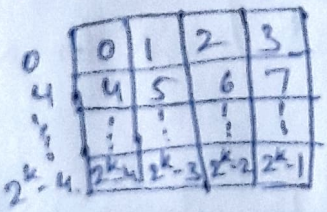
1) Big - endian assignment
2) Little - endian assignment

1) Big endian assignment

Consider word length of computer of 32-bit & word address are 0, 4, 8, 16 ....

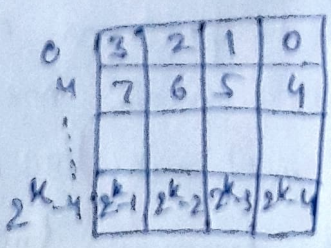


Big endian of the data is MSB & it is stored at lower byte address, i.e., storing of data starts from MSB to LSB



2) Little endian assignment

Little endian means little end of data is LSB & it is stored at lower byte address i.e., storing of data starts from LSB 5M



Q.1.c) Write a short note on basic performance equation

Ans - Let T be the processor time required to execute a program that has been prepared in some high-level language.

- The computer generates a machine language object program that corresponds to the source program.
- Assume that complete execution of program requires the execution of N machine language instructions. The number N is actual number of instructions and is not necessarily equal to the number of machine instructions of object program. Suppose that the average number of basic steps needed to execute one machine instruction is where each basic step is completed in one clock cycle.
- If the clock rate is R cycles per second then program execution time is given by

$$T = \frac{N S}{R}$$

This is referred as basic performance equation.

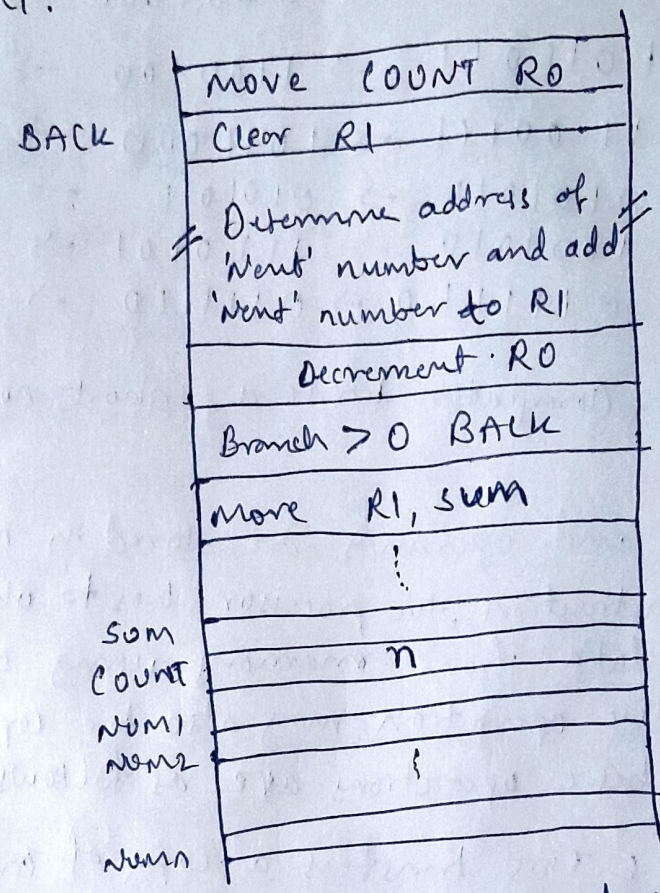
5M

Q.2 a) Describe the concept of branching with an example program of instruction execution.

Ans

Consider a program to add a list of 'n' numbers. Writing separate add instructions to perform addition is not only cumbersome but also requires more memory space to store. Instead of using separate Add instruction, we can use single Add instruction inside a program loop as shown below

The sequence of instruction inside the loop are executed as many times as needed. During each pass, the address of the next number on the list is determined the data at that location is fetched and then added to register R1.



3M

Let 'n' be stored in COUNT. R0 is used as a counter. Decrement decreases the contents of R0 by 1 for every pass through the loop. Execution of loop is repeated as long as R0 is greater than zero.

7M

Q.26). Represent the following decimal values as signed 7-bit numbers using sign and magnitude, signed 1's complement and signed 2's complement formats.

Ans

Decimal	Signed 7-bit number	1's complement	2's complement
-55	1110111	1001000	1001001
+51	0110011	0110011	0110011
.8	0001000	0001000	0001000
-27	1011011	1100100	1000101
-39	1100111	1011000	1011001
+43	0101011	0101011	0101011
-10	1001010	1110101	1110110
62	0111110	0111110	0111110

5M

8.2 C ⇒ Explain Computer Write a short note on memory operations

Ans ⇒ Instructions and operands are stored in memory. To execute an instruction, the processor has to obtain the instruction and data from memory using control signals. The result of the operation may also be copied to the memory. Two basic operations are as follows.

1) load (Read): This transfers a copy of contents from memory location to processor register.
- Processor sends the address of the memory location
- processor issues a Read control signal to memory to request the data.
- Memory sends the data to the processor
- Data gets stored in processor register

2) Store (Write): This transfers the information from the processor to the memory location specified in instruction
→ Processor sends the address of the memory location
→ A write signal is issued by the processor
- The content of the processor register is written into specified memory location.

5M

Module-2

Q.3 a) What is an addressing mode? Explain any four types of addressing modes, with suitable example.

→ Addressing modes: The different ways of in which location of an operand is specified in an instruction.

Four types of addressing modes are.

1) Immediate mode 2) Register mode
3) Direct mode 4) Indirect mode — 2M

1) Immediate Mode: In immediate mode the data is present in the instruction itself. It is identified by '#' sign. Eg:- Move #200, R0
After execution of instruction, R0 = 200. — 2M

2) Register mode: In register mode both source & distination are registers. In this mode data is available in one of the general purpose registers.
Eg:- Move R0, R1 — 2M
Data in R0 is copied to R1 Register.

3) Direct Mode: In direct mode, memory address of the operand is explicitly specified in the instruction
Eg:- Move NUM, R2.
where NUM is the address of memory which holds the data. — 2M

4) Indirect mode - In Indirect mode, the memory address of the operand is not directly specified rather memory address of the operand is specified in one of the register.

eg:- MOVE (R0), R2 2M

w.r.t above example R0 register holds the memory address, where data is present and that data is transfered to R2 register

Q. 3 b) Write a program to compute the sum of test scores of all the students in the three tests. store the corresponding sums in memory.

→ Program

Label	Instructions	Remarks
	MOVE N, R4	R4 ← N
	MOVE #LIST, R0	R0 ← LIST
	CLEAR R1	R1 ← 0
	CLEAR R2	R2 ← 0
	CLEAR R3	R3 ← 0
BACK :	Add 4(R0), R1	
	Add 8(R0), R2	
	Add 12(R0), R3	
	Add #16, R0	
	Decrement R4	
	Branch > 0. BACK	
	MOVE R1, SUM1	
	MOVE R2, SUM2	
	MOVE R3, SUM3	

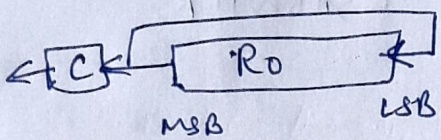
10M

Q.4
a) Explain the Rotate and shift instructions with an example.

→ Rotate instructions

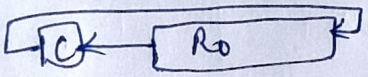
Bits are shifted out of the operand are lost when we use shift instruction. However there are situations like arithmetic operations, where the bits are to be preserved for future use. This is accomplished by using Rotate instructions.

a) Rotate Left without carry (Rotate L)



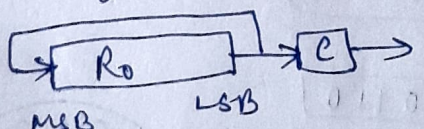
before [0] [1110010]
after [1] [1100101]

b) Rotate left with carry (Rotate RL)



before [0] [01110011]
after [1] [11001100]

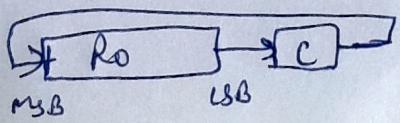
c) Rotate right without carry Rotate R



Eg:- Rotate R #1, R0

Before [01110011] [0]
After [10111001] [1]

d) Rotate right with carry (Rotate RC)



eg: RotateRC #2, R0

before | 0 1 1 1 0 0 1 1 | 0 |

After | 1 0 0 1 1 1 0 0 | 1 |

5M

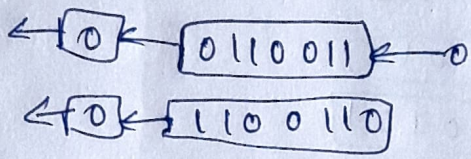
Shift instructions

Many Arithmetic operations like multiplication and division require shifting of operands.

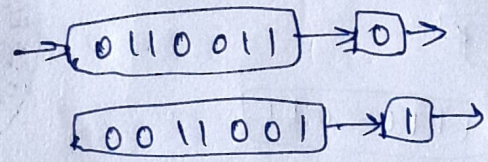
Logical shift → LShiftL, LShiftR

arithmetic shift → AShiftL, AShiftR

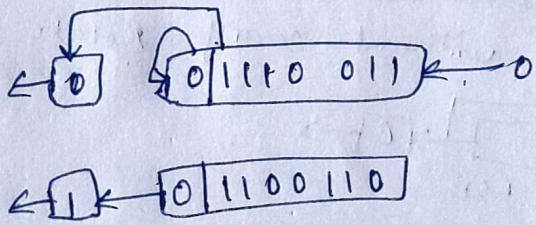
1) LShiftL #1, R0



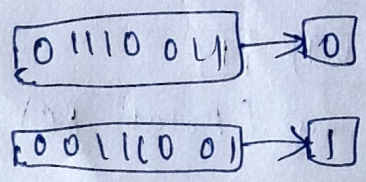
2) LShiftR #1, R0



3) AShiftL #1, R0



4) AShiftR #1, R0



5M

g) Restate

4b) Define subroutine. Explain subroutine linkage using a link register

→ A substack, consisting of a set of instructions, which is executed many times is called as subroutines

A subroutine linkage method is used to save the return address in the special register called as linkage register.

When the subroutine completes its stack, the return instruction returns to the calling program by using the content of link register

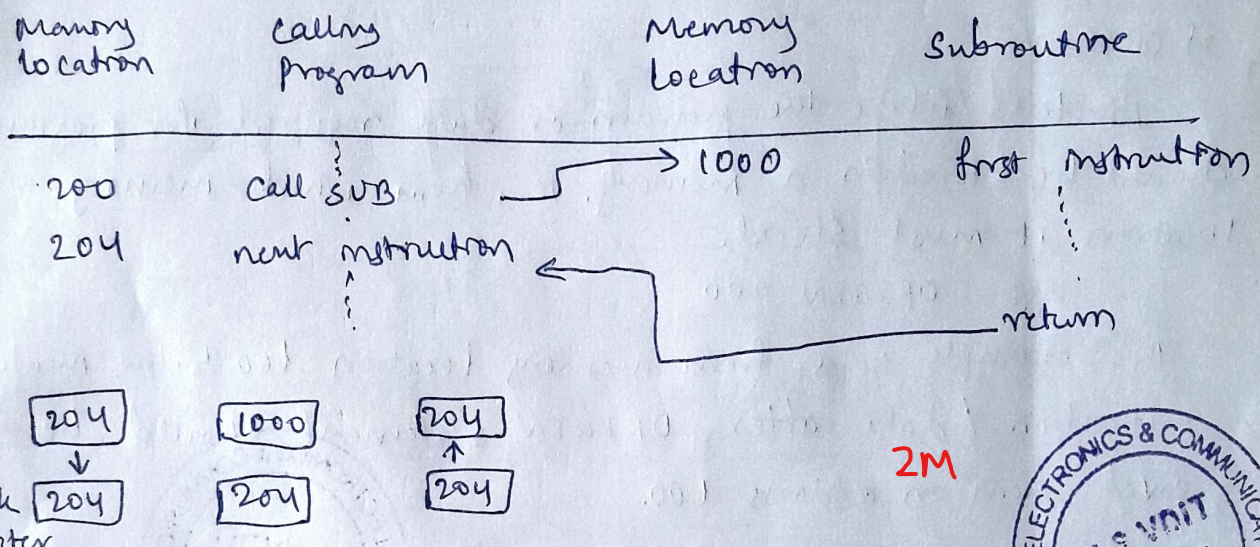


Fig: Subroutine linkage using link register.

2M

Call instruction performs the following operations.

1) Store the content of PC in the link register.
2) Branch to target address specified in the call instruction.

Return instruction performs the following operations

Branch to address contained in link register.

3M

4c) what are assembler directives? Explain any two directives

→ Assembler directives are the commands given to assembler

It gives direction to the assembler how the part of the program must be treated. They are called as Pseudo codes hence they are never executed by the processors. 2M

Assembler Directives

1) EQU – It is an equate directive

eg: SUM EQU 200

This directive equates or assigns value to the name, ie, 200 value is assigned to SUM. Now in program wherever SUM appears then it is replaced by 200.

2) ORIGIN

It tells where the program or data required for program should be stored in a memory. ie, from which memory location it must stored.

eg: ORIGIN 200 3M

This example says from memory location 100 below mentioned instruction / data after ORIGIN statement should be saved from an address 100.

MODULE - 3

5 a) Define I/O interface? Explain I/O interface to connect an input device to the bus, with neat diagram.

→ In order to have an efficient and organized data transfer, an intermediary is required between processor and I/O devices. This is called as I/O interface.

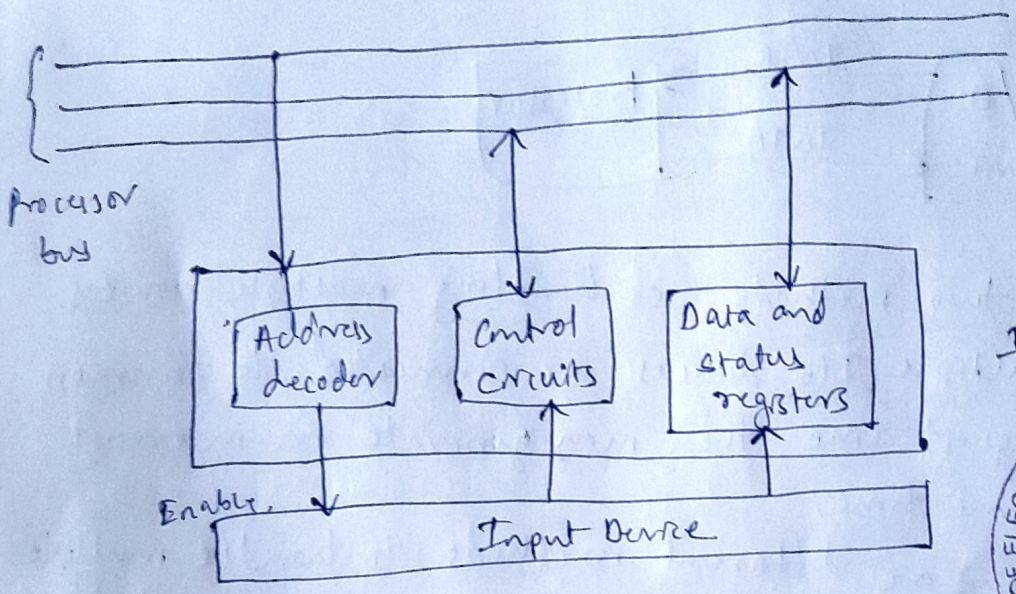
The basic functional blocks of an I/O interface are address decoder, control circuits, Data and status registers.

In addition depending on functionality & application a small RAM/ROM and timer can be incorporated.

Figure below shows a simple I/O interface to connect an input device to the bus.

The address decoder decodes the address sent on bus. The data register hold the data being transferred. The status register contains flags to indicate conditions like buffer full, buffer empty etc..

7M



3M

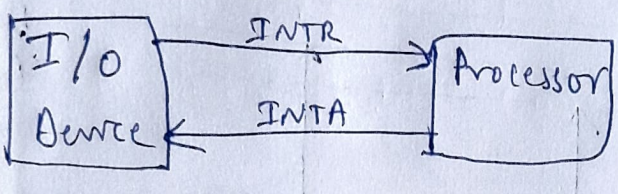
I/O interface.

5b) What is interrupt? Discuss interrupt I/O method for data transfer.

→ In order to achieve synchronization, as soon as I/O devices becomes ready to transfer or receive the data, it sends a special signal over the bus to the processor. This mechanism is called as interrupt.

The routine or sub program executed in response to an interrupt is called the Interrupt Service Routine (ISR).

Once the interrupt signal comes from the device, the processor has to inform the device that its request has been recognized and will be serviced soon. This is indicated by another hardware control signal on the processor bus called interrupt acknowledge signal. This is shown in the schematic below. After receiving the interrupt-acknowledge signal, the external device will de activate the request signal.



5M

5 c) Describe two methods of handling multiple devices.

→ When multiple I/O devices are connected to common interrupt request line, they may generate an interrupt at the same time.

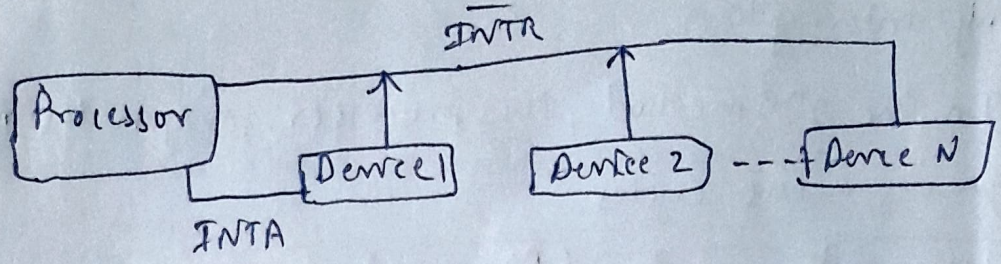
Hence there are different methods to handle multiple device connected to common interrupt request line.

1) Polling the IRQ-bit in processor register.
2) Vectored interrupt
3) Daisy chain.

Vectored Interrupt

A device requesting an interrupt can send a special code to the processor over the bus. By this processor can identify the device even though several requests may come. The code may represent the starting address of the ISR for that device 2M

Daisy chain



In this method, the device that is electrically closest to the processor gets the service by blocking INTA. If device i does not requires any service then INTA signal will be passed to next device. 3M

6 a) Explain the use of DMA controllers in a computer system with neat diagram.

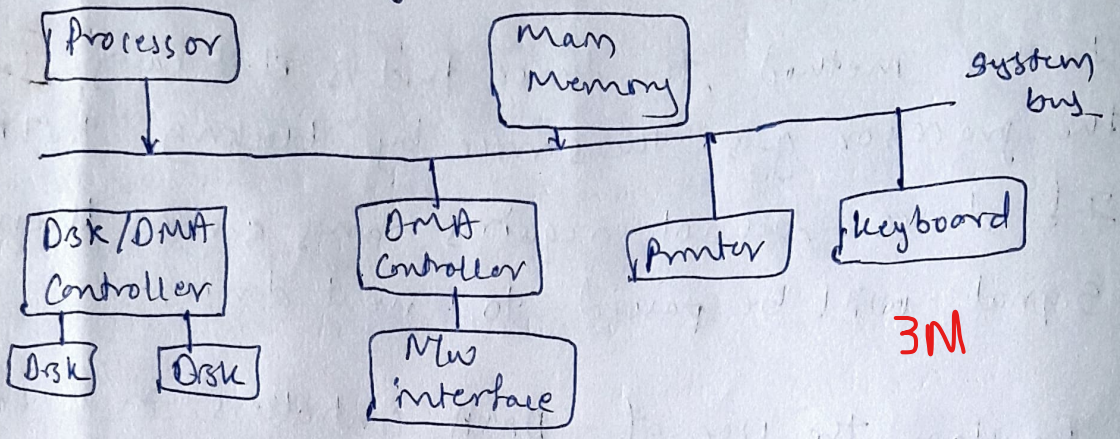
→ - A DMA controller connects a high speed network to the computer bus. The disk controller which controls two disks, also has DMA capability and provides two DMA channels. It can perform two simultaneous DMA operations.

- To start DMA transfer, a routine in OS writes memory address, word count & function to perform information into the registers.
- Now DMA controller proceeds independently to implement the function.
- When DMA transfer is completed, this information is recorded in the status and control register of the DMA controller.

- There are two ways in which DMA operation can be carried out.
- In the 1st method the DMA controller is given exclusive 7M

access to the memory to transfer a data without any interruption.

- In the 2nd method, the processor originates most memory access cycles.



3M

fig: use of DMA controller in a computer system.

6b) write a note on Bus Arbitration.

→ A conflict may arise if both processor and DMA controller try to use the bus at the same time, for memory.

- To resolve these conflicts a special circuit is used called bus arbiter to coordinate.

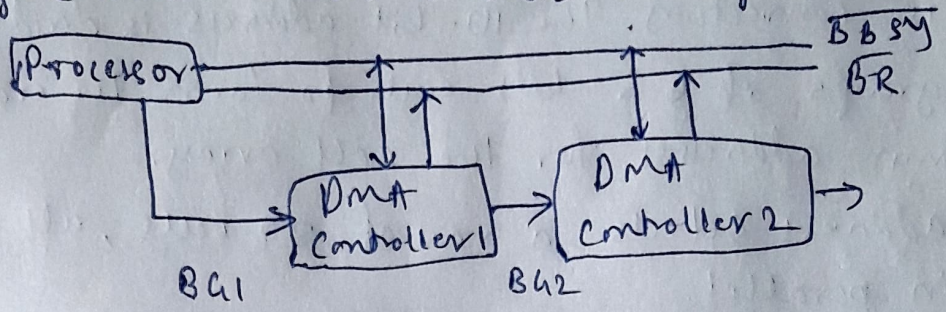
- Device that is allowed to initiate data transfers on the bus is called bus master

- Bus arbitration is the process by which the next device to become the bus master is selected and bus mastership is transferred to it.

There are 2 approaches 1) Centralized 2) Distributed

Centralized

- In this a single bus arbiter performs the required arbitration.
- Figure below shows a basic arrangement



3M

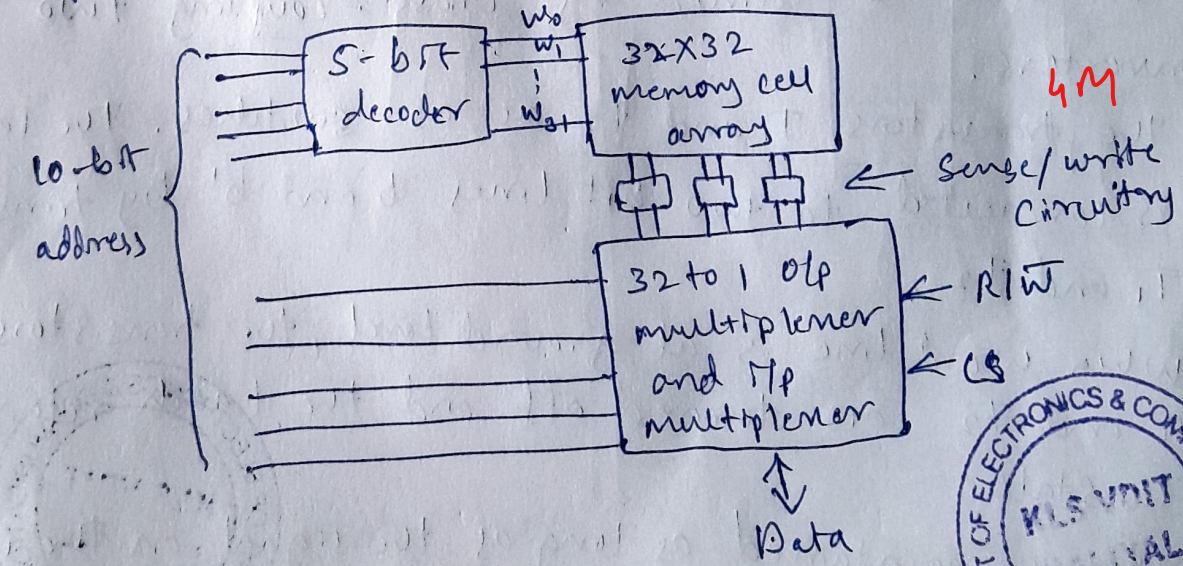
- Processor is the bus master unless it grants the bus mastership to DMA.
- DMA controller activates $\overline{BR}$ to request to become Bus master.
- Processor activates the Bus Grant signal (BG) indicating the DMA controller that they may use Bus when it becomes free.

7M

Module-4

Q.7 a) Explain the organization of 1K×1 memory chip

→



4M

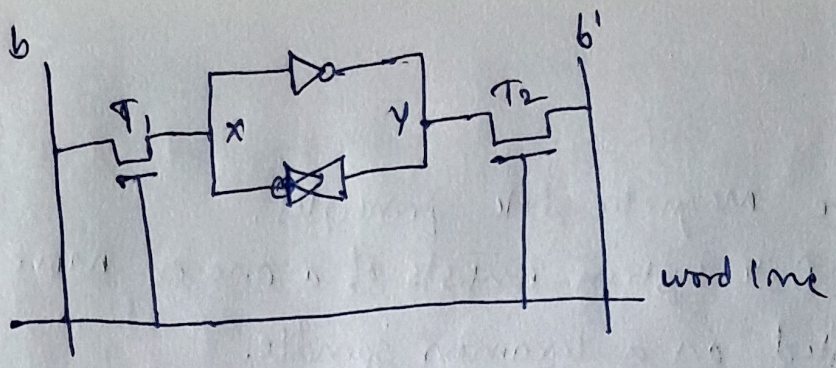
- Memory chip of 1kx1 can be used to reduce the pin count and increase the memory size.
- Here 1k cells are arranged as 1kx1 format which requires 15 connections. The 10-bit address is divided into 2 groups of 5-bits each to form the row and column addresses for the cell array.
- A row address selects a row of 32 cells, all accessed in parallel.
- But according to the column address, only one of these cells is connected to the external data line by the o/p multiplexer and i/p multiplexer.

6M

7b) Write a note on a) static memory b) cache memory.

-s Static memories

- A static memory cell is capable of retaining its state as long as power is applied. A simple static RAM cell may be implemented using a latch.
- The latch is formed by cross-coupling two inverters.
- The transistors, T_1 and T_2 act as switches. The latch is connected to two bit lines b and b' by transistors T_1 and T_2.
- When word line is at ground level, the transistors T_1 and T_2 are turned off and the latch retains its state.
- This is maintained as long as the signal on the word line is not changed and power is applied to the circuit.



SM

Cache memory

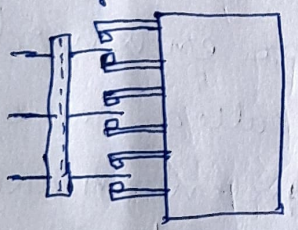
- Since main memory is slow, a special type of memory called cache memory is designed that reduces the time needed to access info.
- There are 2 types of caches.
  1) A primary cache – is always located in the processor chip, referred to as level 1 (L1) cache.
  2) Larger secondary cache – called level 2 (L2) cache, is placed between the processor and main memory.
- Another L3 cache is also implemented in new systems.
- cache memory uses a property known as locality of reference, which works in 2 ways – temporal and spacial.
- The smallest unit of data that can be exchanged between main memory and cache is called cache line.
- When a memory word that is not present in cache is referenced, the block containing the required operands has to be brought into cache from main memory. SM
- If the cache is full, then the cache control hardware must decide which block should be removed to create space for the new block.

8 a). Explain the magnetic disk principles.

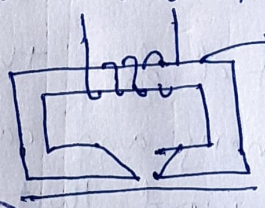
→ Magnetic Disk system consists of a one or more disks mounted on a common spindle.

- A thin magnetic film is deposited on each disk usually on both sides.
- The disk are placed in a rotary disk or drive. So that magnetized surfaces move in close proximity to read/write heads.
- The disk rotate at a uniform speed.
- Each head consists of magnetic yoke and a magnetizing coil. This causes the magnetization of the film in the area immediately underneath the head. to switch to a direction parallel to the applied field.
- The same head can be used for reading the stored info
- Magnetic hard drive is a one of the example of secondary storage device and similarly the other examples are floppy disk and RAID Disk.

7M



Read/write head

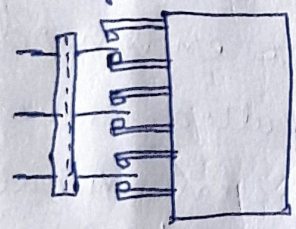


3M

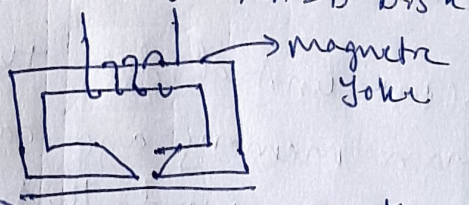
8 a). Explain the magnetic disk principles.

→ Magnetic Disk system consists of a one or more disks mounted on a common spindle.

- A thin magnetic film is deposited on each disk usually on both sides.
- The disk are placed in a rotary disk or drive. So that magnetized surfaces move in close proximity to read/write heads.
- The disk rotate at a uniform speed.
- Each head consists of magnetic yoke and a magnetizing coil. This causes the magnetization of the film in the area immediately underneath the head. to switch to a direction parallel to the applied field.
- The same head can be used for reading the stored info
- Magnetic hard drive is a one of the example of secondary storage device and similarly the other examples are floppy disk and RAID Disk.

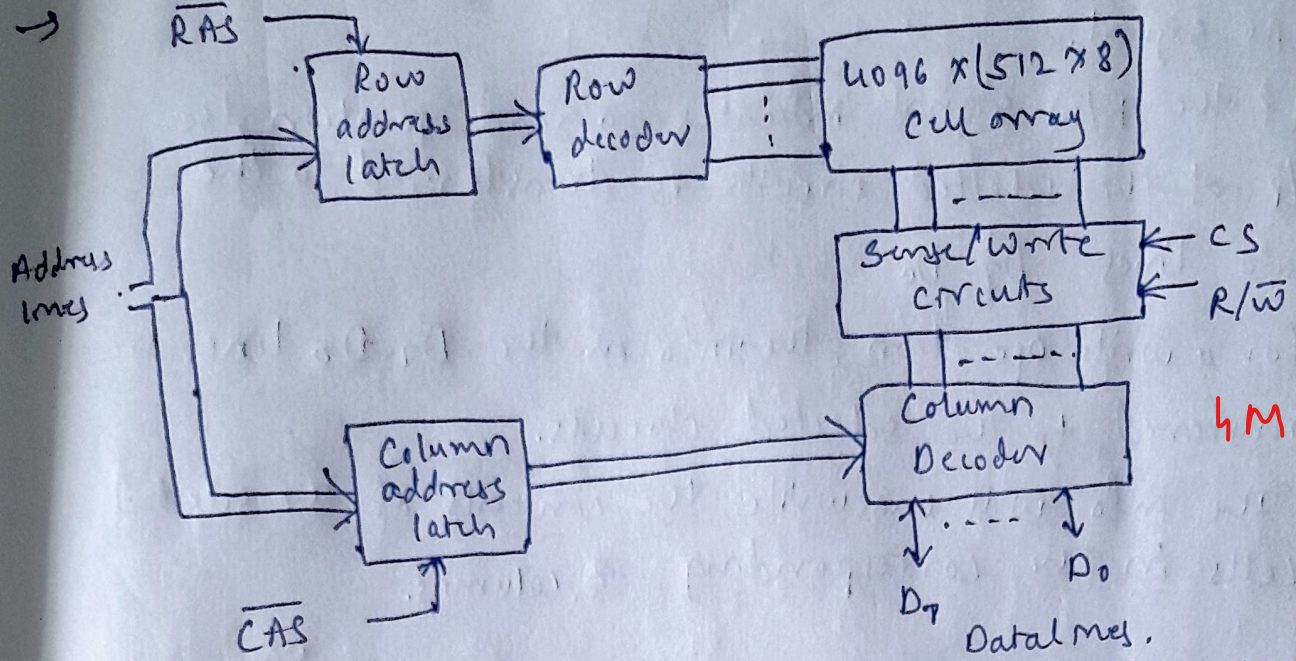


Read/write head



8 b) Draw and explain the internal organization of 2M×8 asynchronous DRAM chip.

→



4M

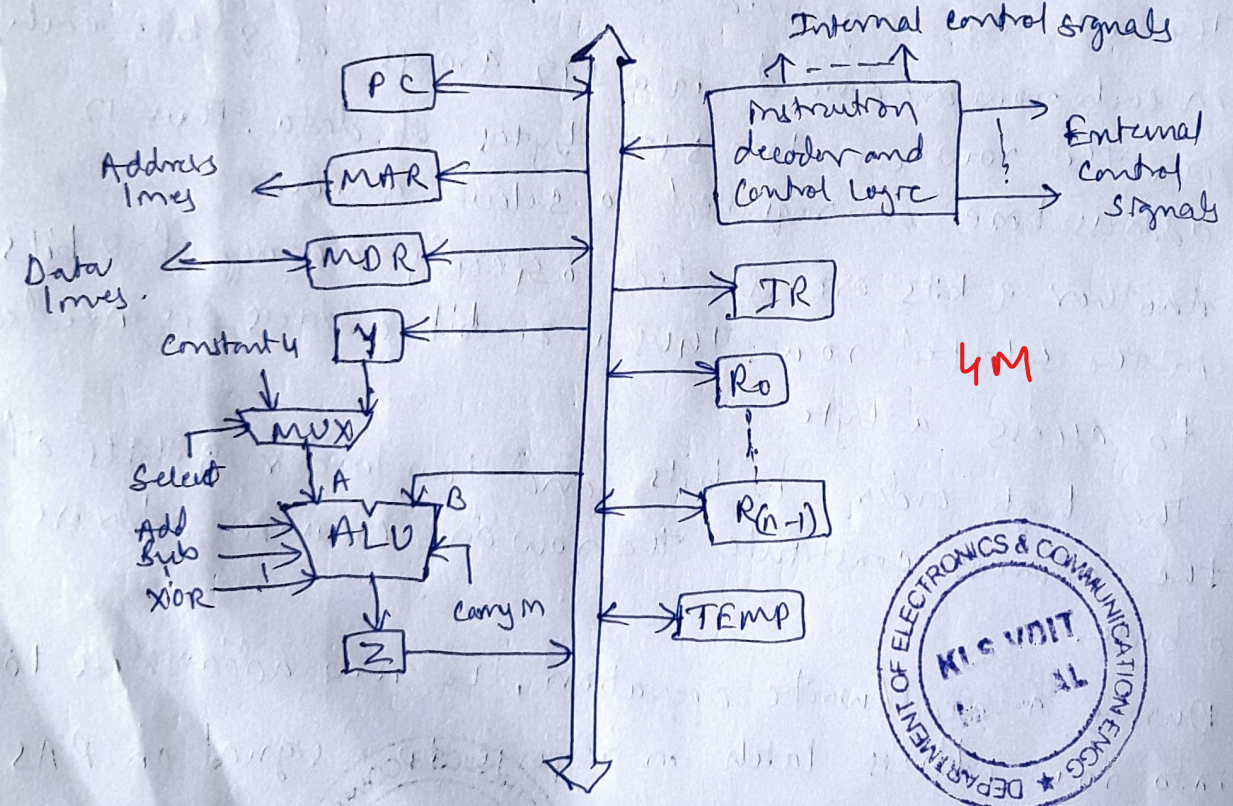
- The above figure shows the organization of a 16M bit DRAM structured as 2M×8 memory
- The cells are organized as 4K×4K array. The 4K cells in each row are divided into 512 groups of 8-bits each, i.e. the row can store 512 bytes of data. Thus 12 address lines are required to select a row.
- another 9 bits are needed to specify a group of 8 bits in the selected row. Thus a 21-bit address is needed to access a byte
- The high order 12 bits and low order 9 bits of the address constitute the row and column address of a byte.
- During a read/write operation, the row address is loaded into row address latch in response to a signal on RAS input of the chip.

- Column address is loaded into column address latch using LAS signal.
- If the R/$\overline{W}$ signal indicates a read operation, the O/P of the selected circuits is transferred to the data lines $D_7 - D_0$
- For a write operation, the info on the $D_7 - D_0$ lines are transferred to the selected circuits. 6M
- The info will ones write the contents of the selected cells in the corresponding 8 columns.

Module-5

Q-9 a). Discuss with neat diagram the single-bus organization of data path inside a processor.

→



4M

fig: Single bus organization

- The instructions of a program are loaded in sequential locations in main memory.
- To execute program, processor fetches one instruction at a time and performs the operations specified.
- Processor uses PC to track the address of next instruction
- Processor performs following actions.
  1) fetch the contents of memory locations pointed to by PC
  2) They are loaded into IR
  3) Increment the content of PC to get next word.
  4) Carry out operations specified in instruction in IR.
- If instruction is more than one word, steps 1 & 2 & 3 must be repeated, which are called as fetch phase.
- Step 4 is called as execution phase
- The functional units like PC, MAR, MDR, R0, R1 etc — are connected to the internal bus. 6M
- MAR and MDR are also connected to the external bus.
- The multiplexer MUX selects either the o/p of registers or a constant value 4. This is provided as I/p A of the ALU.

q6) What are the actions required to execute a complete instruction
ADD (R2), R1

→ Following actions are needed
1) Fetch the instruction
2) Fetch the first operand from memory
3) Perform addition
4) Store the result in R1 4M

Step	Action
1	PCout, MARM, Read, selecty, Add, ZM
2	Zout, PCM, YM, WMFC
3	MDRout, IRM
4	R3out, MARM, Read
5	R1out, YM, WMFC
6	MDRout, selectY, Add, ZM
7	Zout, R1M, End

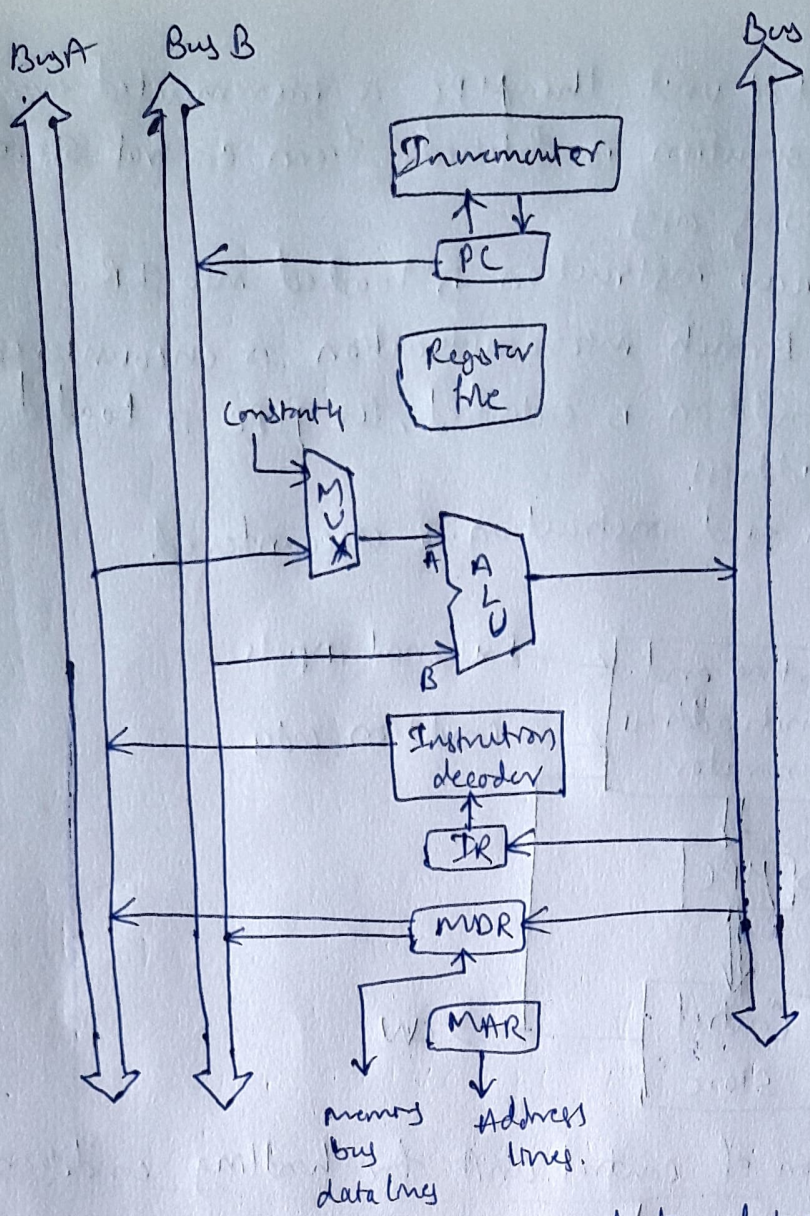
6M

10 a) Draw and explain multiple bus organization of CPU

→ Multiple bus or 3 bus organization of the processor consists of 3 internal bus in the processor.

- The figure below shows three bus structure used to connect the registers and the ALU of the processor.
- All GPR's are combined into the single block called the register file.
- In VLSI technology the most efficient way to implement a number of registers is in the form of an array of memory.
- The register file is said to have 3 ports
- There are 2 O/P allowing the contents of two different registers to be accessed simultaneously and have their contents placed on bus A and B.
- The 3rd port allows the data on the bus C to be loaded into a third register during the same clock cycle
- Bus A and B are used to transfer the source operands to the A and B inputs of the ALU.
- A second feature is the introduction of the incrementor unit, which is used to increment PC by 4

6M



4M

Three bus organization of the datapath inside a processor

10 b) Draw and explain organization of the control unit to allow conditional branching in the microprogram.

→ Micro instructions are to be used to handle unconditional or conditional branching.

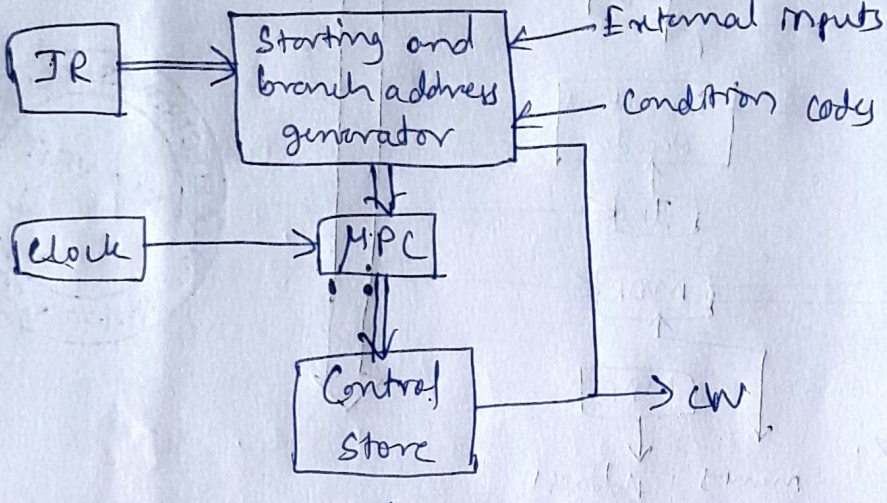
- These micro instructions specify the branch address. They also indicate which external inputs, conditional codes or bits of the IR should be checked as a condition for branching.
- The modified microprogrammed control unit is shown below.
- Starting and branch address generator block accepts inputs from external inputs, condition codes as well as from IR.

- In this control unit, the MPC is incremented everytime a new instruction is fetched from control store, except in the following cases.

a) When a new instruction is loaded into IR. 6M

b) When a Branch microinstruction is encountered and the branch condition is satisfied, the MPC is loaded with the branch address

c) When an end instruction is encountered.



4M

organization of control unit for handling conditional branching

XX The End XX

Head of the Department
Dept. of Electronic & Communication Engg.
KLS V.D.I.T., HALIYAL (U.K.)