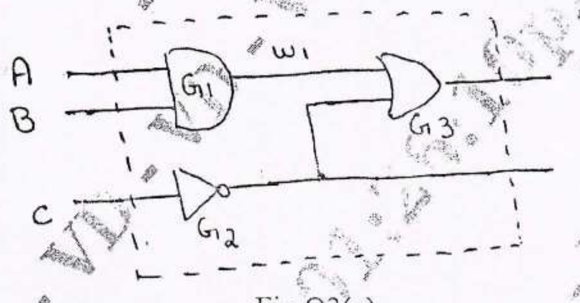
CBCS SCHEME

USN □□□□□□□□□□ **BCS302**

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

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.

		Module – 1	M	L	C
Q.1	**a.**	Determine the complement of the following function: (i) F = xy′ + x′y (ii) F = x′yz′ + x′y′z	06	L3	CO1
	b.	Describe map method for three variables.	04	L2	CO1
	c.	Apply K map technique to simplify the following function: (i) F(x, y, z) = Σ(0, 2, 4, 5, 6) (ii) F(x, y, z) = x′y + yz′ + y′z′	10	L3	CO1
		OR			
Q.2	**a.**	Apply K map technique to simplify the function: F(w, x, y, z) = Σ(1, 3, 7, 11, 15) and d(w, x, y, z) = Σ(0, 2, 5)	06	L3	CO1
	b.	Determine all the prime implicants for the Boolean function F and also determine which are essential F(w, x, y, z) = Σ(0, 2, 4, 5, 6, 7, 8, 10, 13, 15)	10	L3	CO1
	c.	Develop a verilog gate-level description of the circuit shown in Fig.Q2(c).  Fig.Q2(c)	04	L3	CO1
		Module – 2			
Q.3	**a.**	Explain the combinational circuit design procedure with code conversion example.	10	L2	CO2
	b.	Design a full adder circuit. Also develop data flow verilog model for full adder.	10	L3	CO2
		OR			
Q.4	**a.**	Describe 4 × 1 MUX with block diagram and truth table. Also develop a behavioral model verilog code for 4 × 1 MUX.	10	L2	CO2
	b.	What are storage elements? Explain the working of SR and D latch along with logic diagram and function table.	10	L2	CO2
		Module – 3			
Q.5	**a.**	Explain the basic operational concepts between the processor and memory.	10	L2	CO3
	b.	Describe the following: (i) Processor clock (ii) Basic performance equation (iii) Clock rate (iv) SPEC rating	10	L2	CO3
		OR			
Q.6	**a.**	Define addressing mode. Explain any four types of addressing mode with example.	10	L2	CO3

	b.	Mention four types of operations to be performed by instructions in a computer. Explain the basic types of instruction formats to carry out. $C \leftarrow [A] + [B]$	10	L2	CO3
		Module – 4			
Q.7	a.	With a neat diagram, explain the concept of accessing I/O devices.	10	L2	CO4
	b.	What is bus arbitration? Explain centralized and distributed arbitration method with a neat diagram.	10	L2	CO4
		OR			
Q.8	a.	With neat sketches, explain various methods for handling multiple interrupts requests raised by multiple devices.	10	L2	CO4
	b.	What is cache memory? Explain any two mapping function of cache memory.	10	L2	CO4
		Module – 5			
Q.9	a.	Draw the single bus architecture and write the control sequence for execution of instruction ADD (R_3), R_1.	10	L3	CO5
	b.	With suitable diagram, explain the concept of register transfer and fetching of word from memory.	10	L2	CO5
		OR			
Q.10	a.	With a neat diagram, explain the flow of 4-stage pipeline operation.	10	L2	CO5
	b.	Explain the role of cache memory and pipeline performance.	10	L2	CO5

* * * * *

KLS VDIT HALIYAL

Department of Computer Science and Engineering

Sem : 3

Subject : Digital Design and Computer Organization

Subject Code : BCS302

Subject Teacher : Prof. Shree Gowri S S – Sri
Prof. Nirmala Ganiger – Ganiger

Academic Year : 2024-25

HOD

Dr. Venkatesh Shankar

HOD
Computer Science & Engineering
KLS Vishwanathrao Deshpande
Institute of Technology, Haliyal

Q1 i) $F = x\bar{y} + \bar{x}y$ ii) $F = \bar{x}y\bar{z} + \bar{x}\bar{y}z$

1a

i) $F = x\bar{y} + \bar{x}y$

The dual of F is $(x+\bar{y})(\bar{x}+y)$

Complement each literal. $(\bar{x}+y)(x+\bar{y})$

or

$F = \overline{x\bar{y} + \bar{x}y}$

$= \overline{x\bar{y}} + \overline{\bar{x}y}$

$= (\bar{x}+\bar{\bar{y}})(\bar{\bar{x}}+\bar{y})$

$= (\bar{x}+y)(x+\bar{y})$

ii) $F = \bar{x}y\bar{z} + \bar{x}\bar{y}z$

The dual of F is $(\bar{x}+y+\bar{z})(\bar{x}+\bar{y}+z)$

complement of each literal $(x+\bar{y}+z)(x+y+\bar{z})$

or

$F = \overline{(\bar{x}y\bar{z} + \bar{x}\bar{y}z)}$

$= \overline{(\bar{x}y\bar{z})} \cdot \overline{(\bar{x}\bar{y}z)}$

$= (x+\bar{y}+z)(x+y+\bar{z})$

1b) Three variable K-map: There are eight minterms for three binary variable

∴ The map consists of 8 square

The minterms are arranged not in binary

binary sequence similar to the graycode.

=> characteristic of this only one bit change in value from one adjacent column to next

$x \backslash yz$	$\bar{y}\bar{z}$	$\bar{y}z$	yz	$\bar{y}z$
$\bar{x}$	m_0	m_1	m_3	m_2
x	~~3u~~ m_4	m_5	m_7	m_6

Cell 0 is adjacent to 1,4,2

Cell 2 is adjacent to 3,6,0

Cell 4 is adjacent to 0,5,6

Cell 6 is adjacent to 2,7,4

The cell number have been arranged that physically adjacent cells are also logically adjacent

$x \backslash yz$	00	01	11	10
0	0	1	3 1	2
1	4 1	5	7 1	6 1

1c) i) $F(x, y, z) = \Sigma(0,2,4,5,6)$

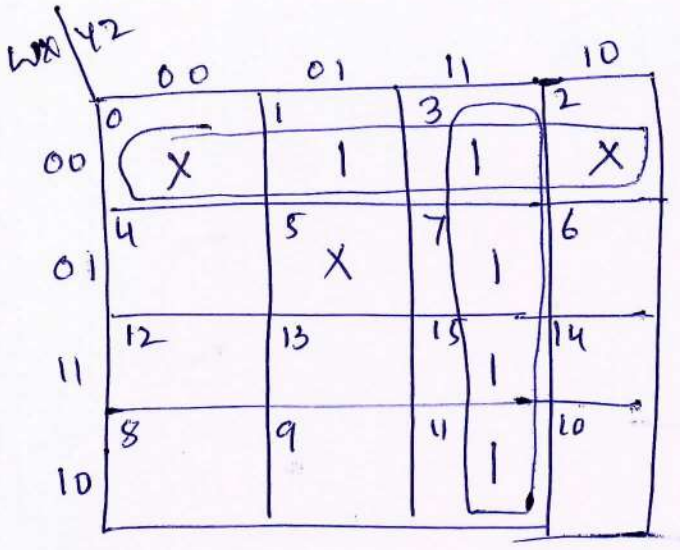
$x \backslash yz$	00	01	11	10
0	0 1	1	3	2 1
1	4 1	5 1	7	6 1

$$F = \bar{z} + x\bar{y}$$

ii) $F(x, y, z) = \bar{x}y + y\bar{z} + \bar{y}\bar{z}$

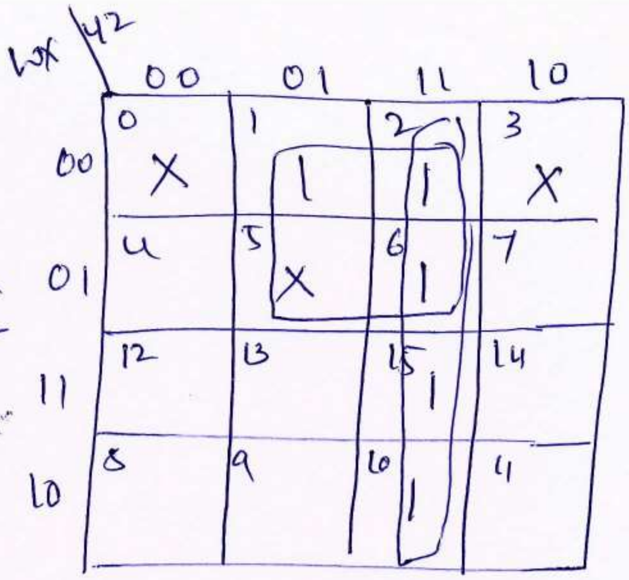
$x \backslash yz$	$\bar{y}\bar{z}$	$\bar{y}z$	yz	$y\bar{z}$
$\bar{x}$	0 1	1	2 1	3 1
x	4 1	5	7	6 1

2a)



wx\yz	00	01	11	10
00	0 X	1 1	3 1	2 X
01	4	5 X	7 1	6
11	12	13	15 1	14
10	8	9	11 1	10

Or

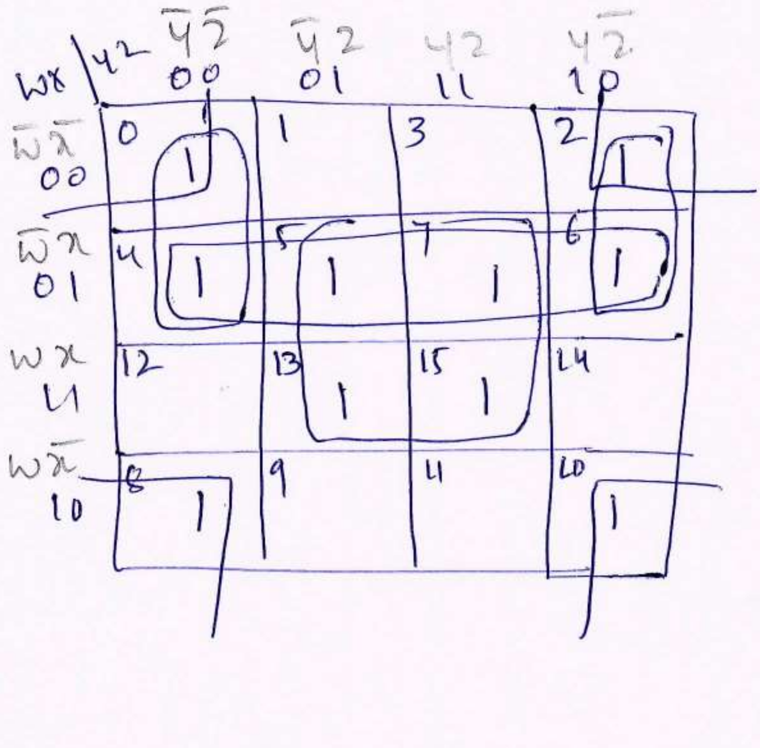


wx\yz	00	01	11	10
00	0 X	1 1	2 1	3 X
01	4	5 X	6 1	7
11	12	13	15 1	14
10	8	9	10 1	11

$F = yz + \bar{w}\bar{x}$

$F = yz + \bar{w}z$

2b)



wx\yz	$\bar{y}\bar{z}$ 00	$\bar{y}z$ 01	yz 11	$y\bar{z}$ 10
$\bar{w}\bar{x}$ 00	0 1	1	3	2 1
$\bar{w}x$ 01	4 1	5 1	7 1	6 1
wx 11	12	13 1	15 1	14
$w\bar{x}$ 10	8 1	9	11	10 1

Prime implicant
$\bar{x}\bar{z}$, $\bar{w}x$, $\bar{w}\bar{y}\bar{z}$, $\bar{w}y\bar{z}$, xz

Essential prime implicant · xz, $\bar{x}\bar{z}$

2c)

```
Module and-or-gate (
    input A, B, C;
    output D, E );
);
   wire W1;
   and G1 (W1, A, B);
   not G2 (E , C);
   or  G3 (D, W1, E);
end module
```

3a The design of combination circuit start from specification of the design objective and culminate in logic circuit diagram.

code conversion Example

The availability of large variety of codes for some discrete element of information result in use of different code by different digital system. Thus, code converter is circuit that makes two system compatible even though each uses a different binary code.

Example: BCD to excess 3 code Converter.

Input BCD				output Excess-3 code			
A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	Φ
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	Φ	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	Φ	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

wx \ yz	00	01	11	10
00	0 1	1	3	2 1
01	4 1	5	7	6 1
11	12 X	13 X	15 X	14 X
10	8 1	9	11 X	10 X

$z = \overline{D}$

wx \ yz	00	01	11	10
00	1		1	
01	1		1	
11	X	X	X	X
10	1		X	X

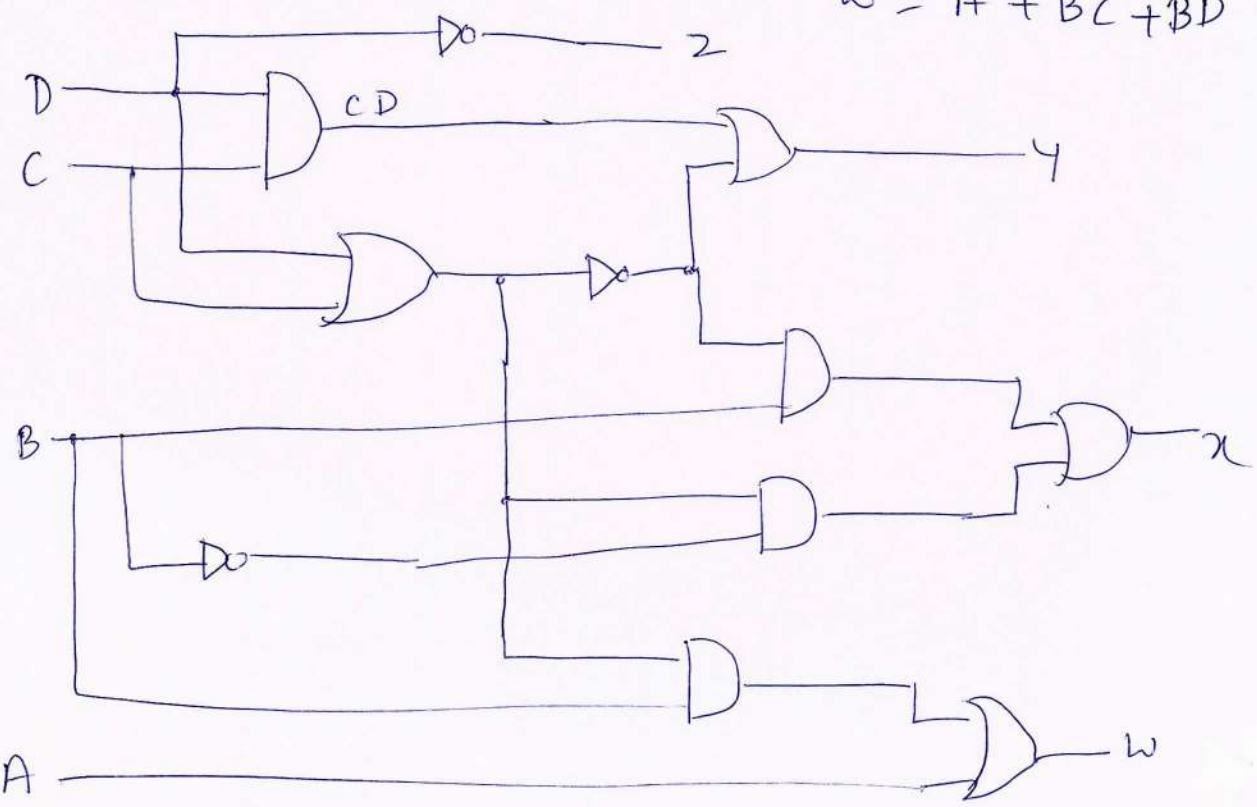
$y = CD + \overline{C}\,\overline{D}$

AB \ CD	00	01	11	10
00	0	1 1	3 1	2 1
01	4 1	5	7	6
11	12 X	13 X	15 X	14 X
10	8	9 1	11 X	10 X

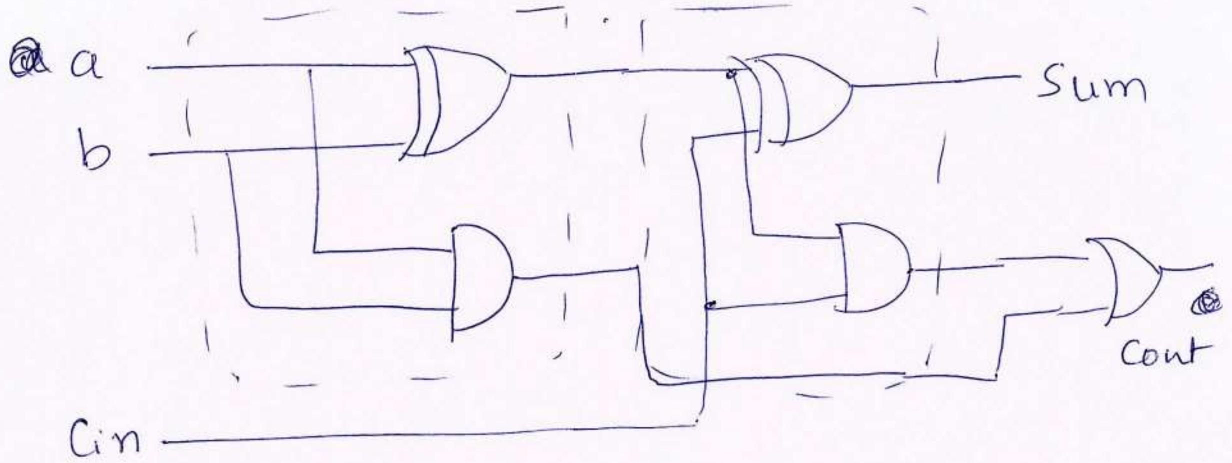
$x = \overline{B}C + \overline{B}D + B\overline{C}\,\overline{D}$

wx \ yz	00	01	11	10
00	0	1	3	2
01	4	5 1	7 1	6 1
11	12 X	13 X	15 X	14 X
10	8 1	9 1	11 X	10 X

$w = A + BC + BD$



3b) Full Adder



$Sum = a \oplus b \oplus Cin$

$Cout = (a \oplus b)Cin + ab = ab + bc + ca.$

```
Module fadd (sum, Cout, a, b, cin)
input a, b, Cin
output Sum, cout;
assign sum = a ^ b ^ c;
assign Cout = (a&b) | (b&Cin) | (cin&a);
endmodule
```

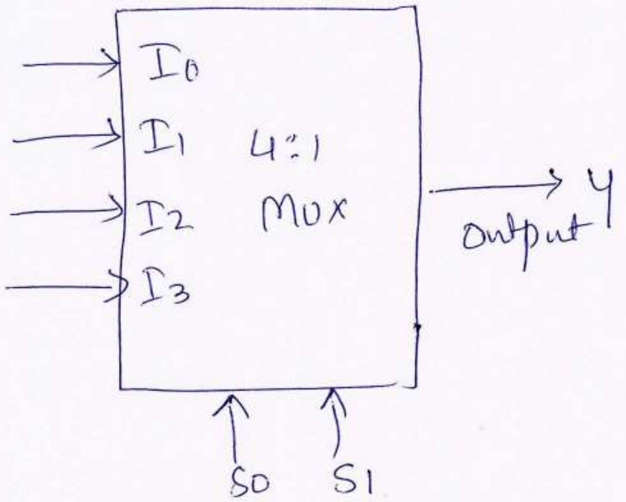
4a) Multiplexer is a combinational circuit that select binary information from one of many inputs lines and directs it to a single output line

* A selection of a particular input lines whose bit combination determine which input is selected.

2^n input n selector 1, output

4(ii) It has 4 input and single output

⇒ The output is selected based on one of 4 inputs which is based on selection inputs.



S1	S0	Y
0	0	I0
0	1	I1
1	0	I2
1	1	I3

```
module mul-4-to-1 (I0, I1, I2, I3, S0, S1, Y)
    input wire I0, I1, I2, I3;
    intput wire S0, S1;
    output regout;
    always @ (I0, I2, I3, I1, S0, S1)
    begin
     case (S0|S1)
     2'b00 : out = I0;
     2'b01 : out = I1;
     2'b10 : out = I2;
     2'b11 : out = I3;
     endcase
    end module;
```

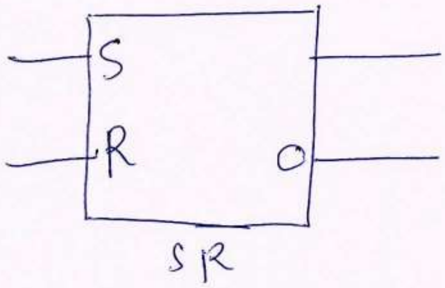
4b) Storage elements are devices capable of storing binary information. Binary information stored in these elements at any given time defines the state of sequential circuit at that time.

SR Latch: SR Latch has two inputs S(set) and R (Reset). It has two output Q & $\overline{Q}$ SR Latch constructed with two cross coupled NOR gate or NAND gate.

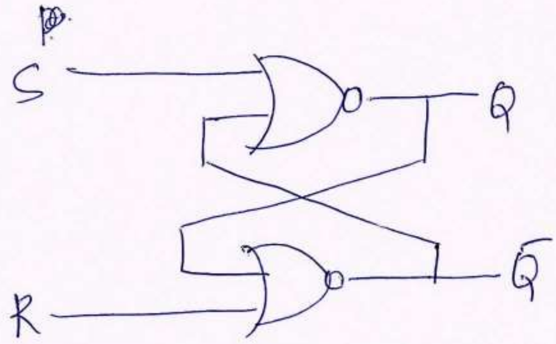
* Latch has two state. when output $Q=1$ and $\overline{Q}=0$ the latch is said to be in set state.

* when $Q=0$ & $\overline{Q}=1$ it is in Reset ~~also~~ state.

SR Latch with NOR gate.



SR

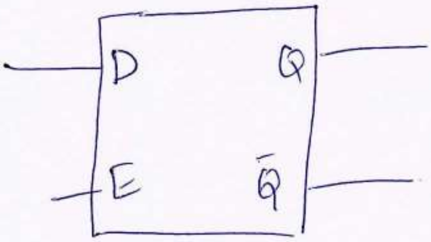


S	R	Q	$\overline{Q}$	
1	0	1	0	
0	0	1	0	(after $S=1, R=0$
0	1	0	1	
1	0	1	0	(after $S=0, R=1$)
1	1	0	0	forbidden

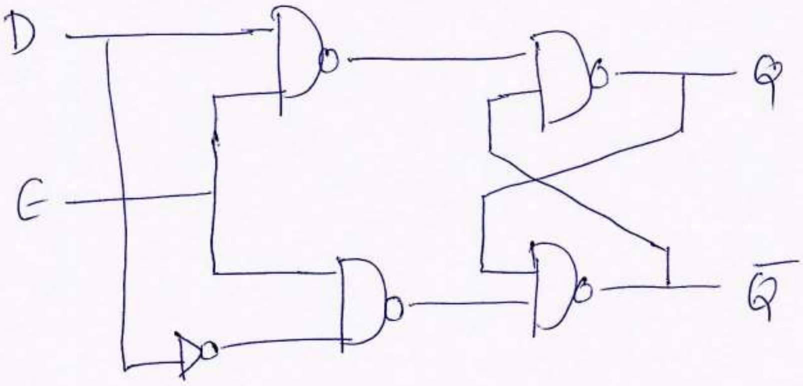
D Latch : D latch can store a bit value 1 or 0 when its enable pin high, the value on the D pin will be stored on the Q output.

* The D latch is logic circuit most frequently used for storing data in digital system.

* It is based on S-R latch, but it does'nt have an 'undefined' or "invalid" state problem.



E	D	Q	
0	x	No change	
1	0	Q = 0	Reset
1	1	Q = 1	Set



.5a Basic operational Concept

=) Instruction are accessed from memory to the procenor one by one and executed.

steps for instruction execution.

& Fetch instruction from memory into IR.

* Decode instruction

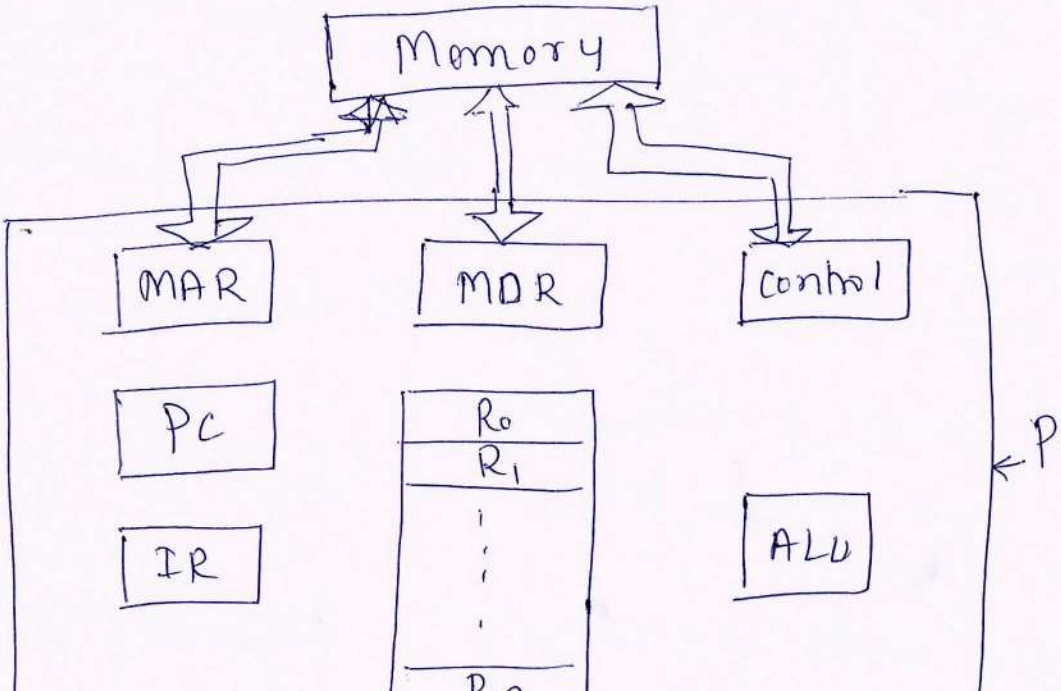
* Accen memory opeland

* Accen Registr opeland

* perform operation according to operation. code

* store result in destination memory location

Connection Between memory andprocenor



IR : Instruction Registr which holds instruction to be executed, it notices control unit, which generate timing signal the control various operations in execution of instruction

PC: program countes : It is special purpose of. registes used to hold the addren of next instruc tion to executed.

=) the content of PC are incremented by 1 or 2 or 4 during execution of current instruction

General purpose Registes: Registes is group of flip flop., it in storage element.

=) It is used store data temporarly during execution of program.

=) It can used as pointer ~~as~~ to memory.

MAR: Memory addren Registes which holds addren of location to accened.

=) It establish connection between memory and procemor.

=) It stores addren of memory location.

MDR: Memory data Registes.

which contain data to be read or written to addren location.

=) It also establishes connection between memory and procenor.

=) It stores content of memory location (data) written or read from memory

control unit: It control data transfes opelation between memory and procenor.

It control data transfes between I/o & procenor

ALU ; Arithmetic logic unit ; It perform arithmetic logical operation on given data.

steps for Reading instruction.

$[PC] \rightarrow MAR \rightarrow memory \rightarrow MDR \rightarrow IR.$

cp (Read Signal).

5b) i) processor Clock : processor circuit are controlled by timing signal called clock.

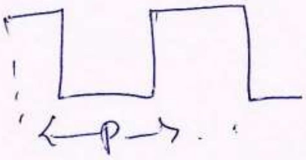
=) clock cycle are regular time interval.

=) processor devide action to be performed into sequence of steps.

=) Each step is completed in one clock cycle.

p → length of one clock cycle

R → clock rate $R = \frac{1}{P}$



Measured cycles per second or Hertz (Hz)

500 million cps is 500 Hz.

1250 million cps is 1.25 GHz.

$$P = \frac{1}{R} = \frac{1}{500000000} = 2 \text{ nanaseconds.}$$

$$P = \frac{1}{1.25 \times 10^9} = 0.8 \text{ ns.}$$

ii) Basic Performance Equation.

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

=) Compiler geneeate machine language object Program, Let anume

=) complet execution of program require of N machine language instruction

=) N is actual no. of instn execution

=) S is aveeage numbel of basic steps needed to execute one machine instruction.

=) Each basic step is completed in one clock cycle.

=) R is clock rate.

Execution time is $T = \frac{N \times S}{R}$.

high performance => reduced value of T

-) Reducing N & S, increasing R

iii) Clock Rate:

possibility of increasing Clock Rate R.

* Improving IC technology makes logic circuit faster. This reduce P & Hence increase R, this reduce time needed to complete basic step.

* Reducing amount of processing done in one basic step.

(iv) SPEC Rate: Computer community adpoted the idea of measuring computer performance using benchmark program.

=) Bench mark program is tool or script designed to measure the performance of specific component of system.

=) standard program must be used.

=) performance measure is time take to execute given benchmark

=) A not profit organisation called system performane Evalution corporation (SPEC) select & publis appl$\underline{n}$ program for different application domain

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

SPEC Rating so means computer under test so time faster on particular Benchmark.

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

n is number of program unit.

6a) Addressing mode is rule for specifying how to . interpret or modify an instruction address field. before accessing the opeland.

(ii) Type of Addressing mode.

i) Registes Mode; The opeland is the content of a processor register. register name specifies. the address of registes.

Eg; MOVE R_1, R_0

Content of R_1 registes moved to registes R_0

ii) Absolute Mode; The opeland is in memory location. The address of this location is specified in the instruction. Afts called direct mode.

Eg: MOVE LOC, R2

the content of memory location LOC is moved to registes R2

iii) Immediate Mode; The opeland is given explicitly in the instruction value of source opeland is specified.

Eg: MOVE #200., R3

The direct value is moved to the registes R3, any one of opeland is value in immediate mode.

iv) Indirect Mode; Effective addren of opeeand is the content of registes or memory location whose addren appear in instruction

=) indirection is denoted by palanthesis around the name of registes or memory location.

Eg: ADD (R1), Ro

indirection addrening mode through GPR. Add content of memory location pointed by R1 to the content of Ro and place the result in Ro.

6b. Mention any foue types of opeeations to be peefoemed by instructions in a compute. Explain the basic types of instruction foemat to carry out — 10 marks.

Ans:- Foue types of opeeations are Addition, K.c ~~Theee address instruction~~ Subtraction, multiplication, Division

Basic types of instruction foemat are :-

$C = A + B$

1) Add A, B, C → Theee address instruction

2) Add A, B ; $B \leftarrow [A] + [B]$

Move B, C ; $C \leftarrow [B]$

→ Two address instruction

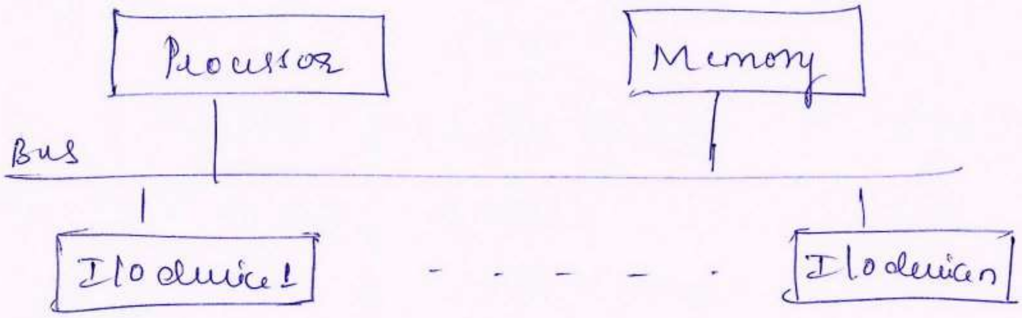
3) Store C = A + B using one address instruction

```
load  A   ;  Acc ← A
Add   B   ;  Acc ← Acc + B
Store C   ;  C ← Acc
```

Module – 4

7a. with a neat diagram, explain the concept of accessing I/O devices — 10Marks.

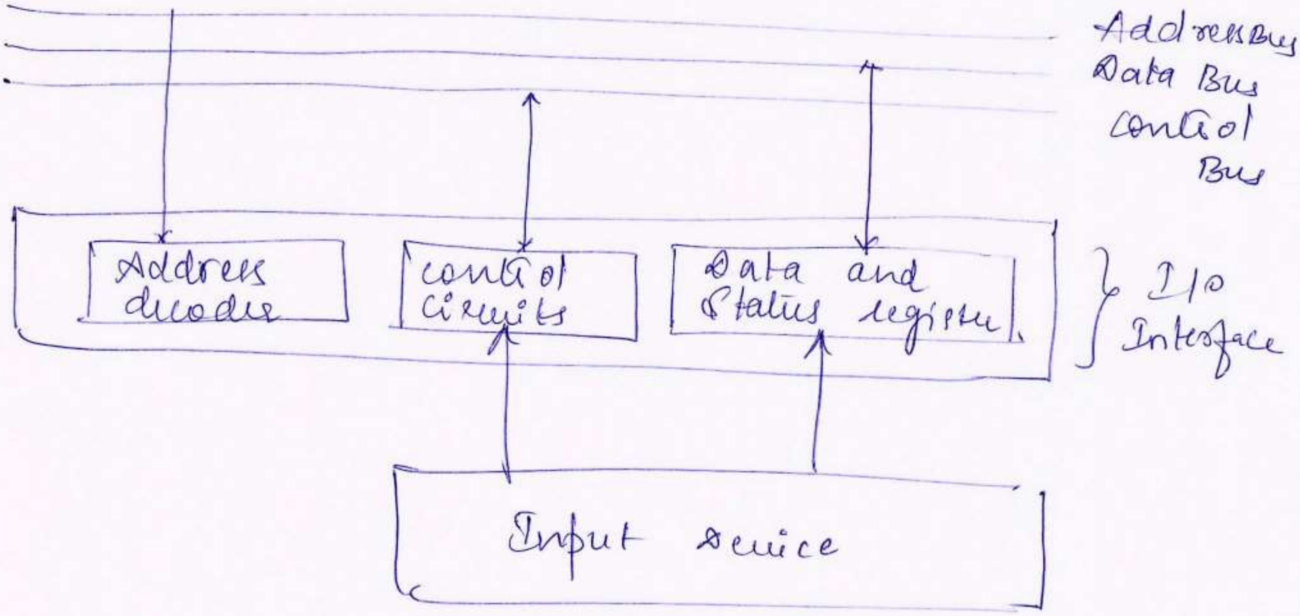
Sol:-



I/O devices can be connected to computer using single bus structure. Bus enables all devices connected to exchange information. There are 3-types of bus 1) Address Bus 2) Data Bus 3) control Bus.

Each I/O device is assigned with unique set of addresses. when processor place the address on address bus, the device that recognize this address responds to command issued on the control bus. Normally processor requests for read/write operation & the requested information is transferred on the data bus.

Hardware required to connect I/o device to bus.



Address Decoder :- Address decoder enables the I/o device to recognize its address when this address appears on the address lines.

Data & Status Register :- Data register holds the data being transferred to or from the processor. Status register contains relevant information to indicate the operation of the I/o device.

Control circuitry :- required to co-ordinate the I/o devices.

7b. What is bus arbitration? Explain centralized and distributed arbitration method with a neat diagram —10 Marks

Sol:- The one device that initate the data transfer on the bus at any given time is called bus Master.

Bus Arbitration:- It is the process by which the next device going to become bus master and Bust Mastership is transferred to it.
There are two approaches for bus Arbitration
1) Centralized Arbitration
2) Distributed Arbitration.

1) Centralized Arbitration:- The Bus Arbiter may be the processor of seperate unit connected to the bus

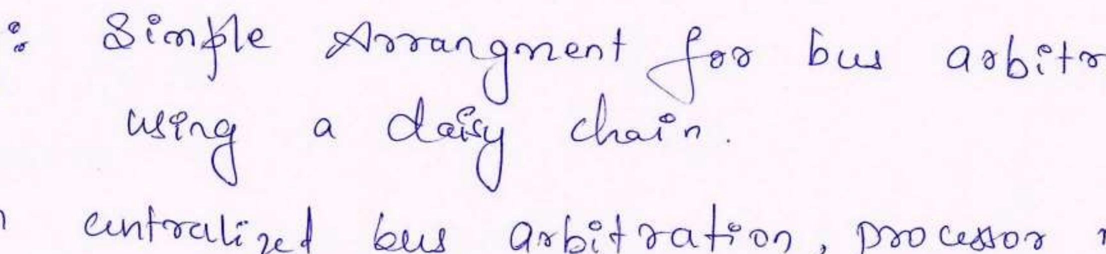


fig: Simple Arrangment for bus arbitration using a daisy chain.

In centralized bus arbitration, processor may be the bus arbiter & it makes the decision who is going to become the bus master

DMA controller indicates that it needs to become bus master by activating Bus-request line BR.

The signal on the Bus-request line is logical OR of the bus requests from all the devices connected to it. when Bus request is activated, the processor activates the Bus grant signal BG1, indicating to DMA controllers that they may use the bus when it becomes free.

Distributed Arbitration :- Distributed Arbitration means all devices waiting use the bus have equal responsibility in carrying out the arbitration process, without using central arbitrer. The bus is assigned a 4-bit identification number. when one or more devices request the bus they assert the start arbitration signal & place their 4-bit ID number on four open collector line $\overline{ARB0}$ through $\overline{ARB3}$

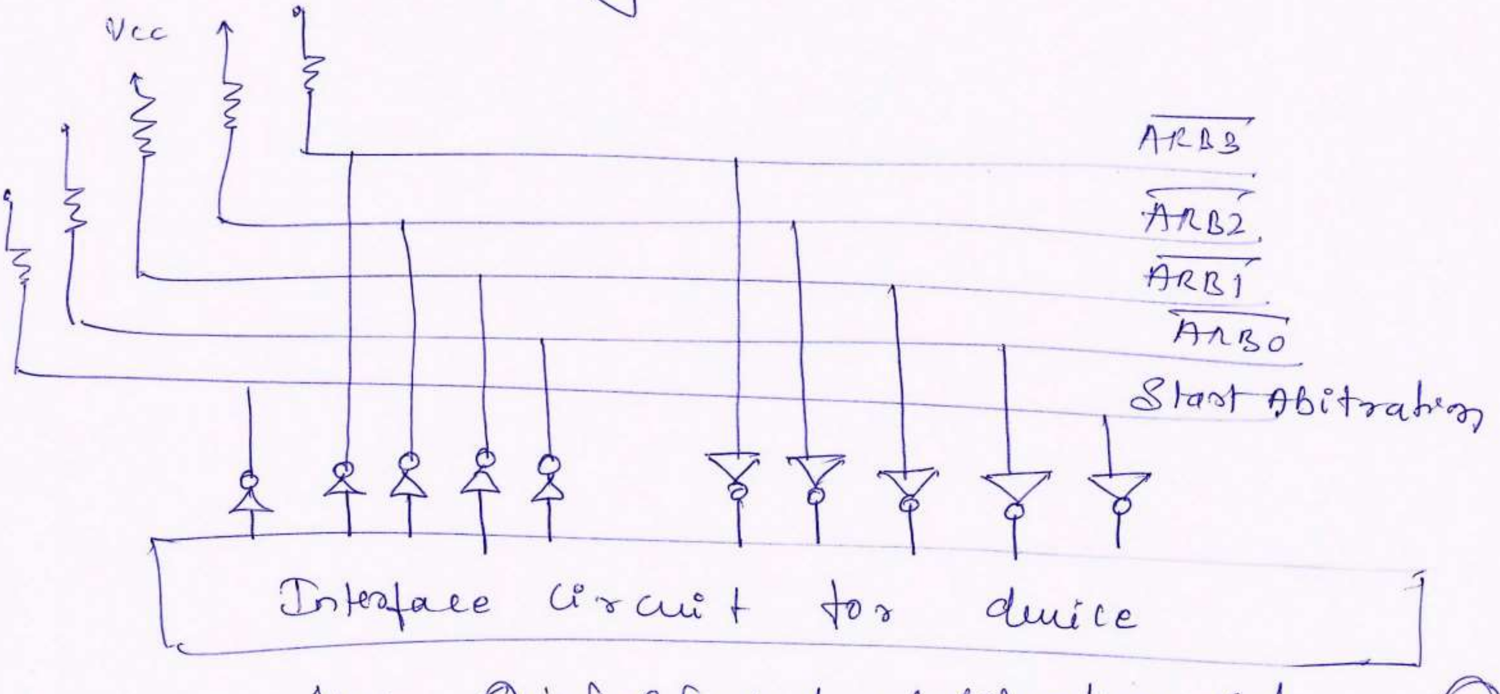


fig: Distributed Arbitration Scheme.

OR

8 a. with neat sketches, explain various methods for handling multiple interrupts requests raised by multiple devices —— 10 Marks

Sol:- The various methods for handling multiple interrupt request raised by multiple devices are :-

1) Vectored interrupt :- Vectored- interrupt is code supplied by the I/O device may represent the starting address of ISR and its length is 4 to 8- bits or it may be the address of memory where the address of ISR is located. This arrangment implies the address of ISR for given device may always locate at same location. The starting address of ISR is in a memory address & it is called as interrupt vector and the processor reads this address & loads it into the PC. Interrupt vector code is sent to the processor through the data bus.

2) Multiple level priority organization

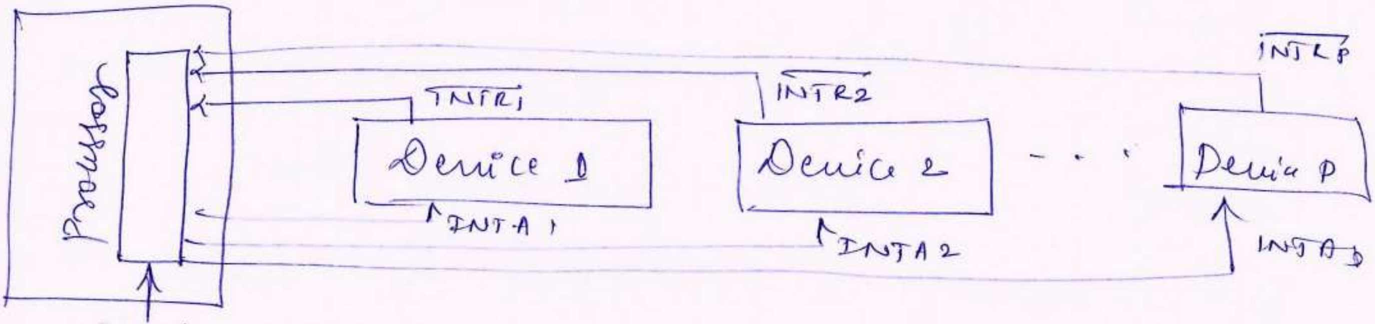


fig: Implementation of interrupt priority

Multiple priority scheme can be implemented using seperate interrupt request and interrupt - acknowledge lines for each device. Each of the interrupt request lines is assigned a different priority level. Interrupt requests received over these lines are sent to the priority arbitration circuit in the processor. A request is accepted only if it has a higher priority than that current assigned to the processor.

Q6. what is cache memory? Explain any two mapping function of cache memory — 10 Marks.

Sol:- Cache memory is an efficient solution which essentially makes the main memory appear to processor to be faster than it really is.

Mapping function is to map main memory to cache memory. There are 3 - types of mapping function.

1) Direct Mapping
2) Associative Mapping.
3) Set - Associative mapping

1) Direct Mapping :- let us consider, cache can contain 128 blocks where each block is of 16 words ∴ 128 × 16 = 2048K words.

Main memory is addressable by 16-bit address ∴ 2^{16} = 64K words

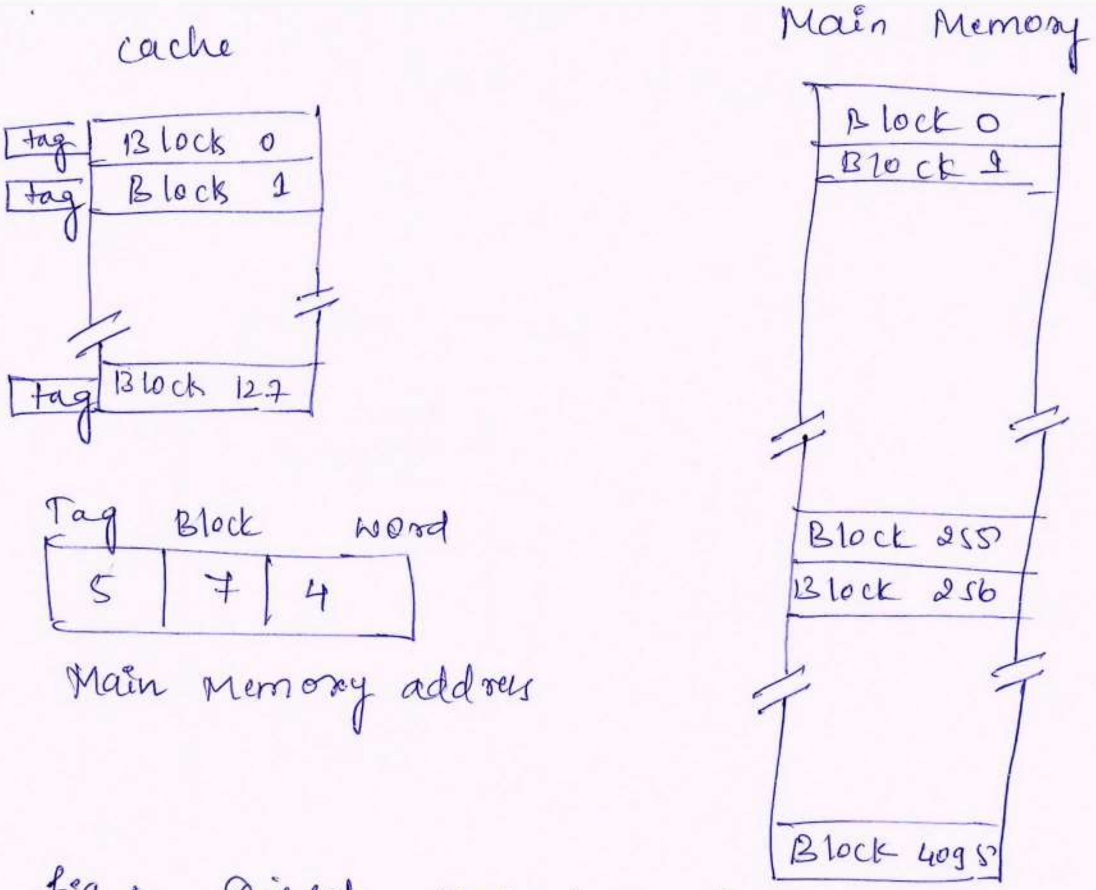


fig : Direct mapped cache

Direct - mapping is simplest - mapping technique In this mapping modulo operation takes place. The size cache memory in terms of Block is 128 ∴ The block that needs to be placed in cache memory is block number modulo 128

for ex:- block 0, 128, 256, ... of main memory operation block 0 modulo 128

block 128 modulo 128

block 256 modulo 128 & so on

The main memory address is divided into three fields tag, block & word.

word field is of 4-bits $8 - 2^4 = 16$ ∴ 1-block contains 16 words, this selects one of the word in the 16 - word

Blocks field is of 7-bit :- 7-bit is a blocks field. which determines the position of blocks in cache memory

Associative Mapping

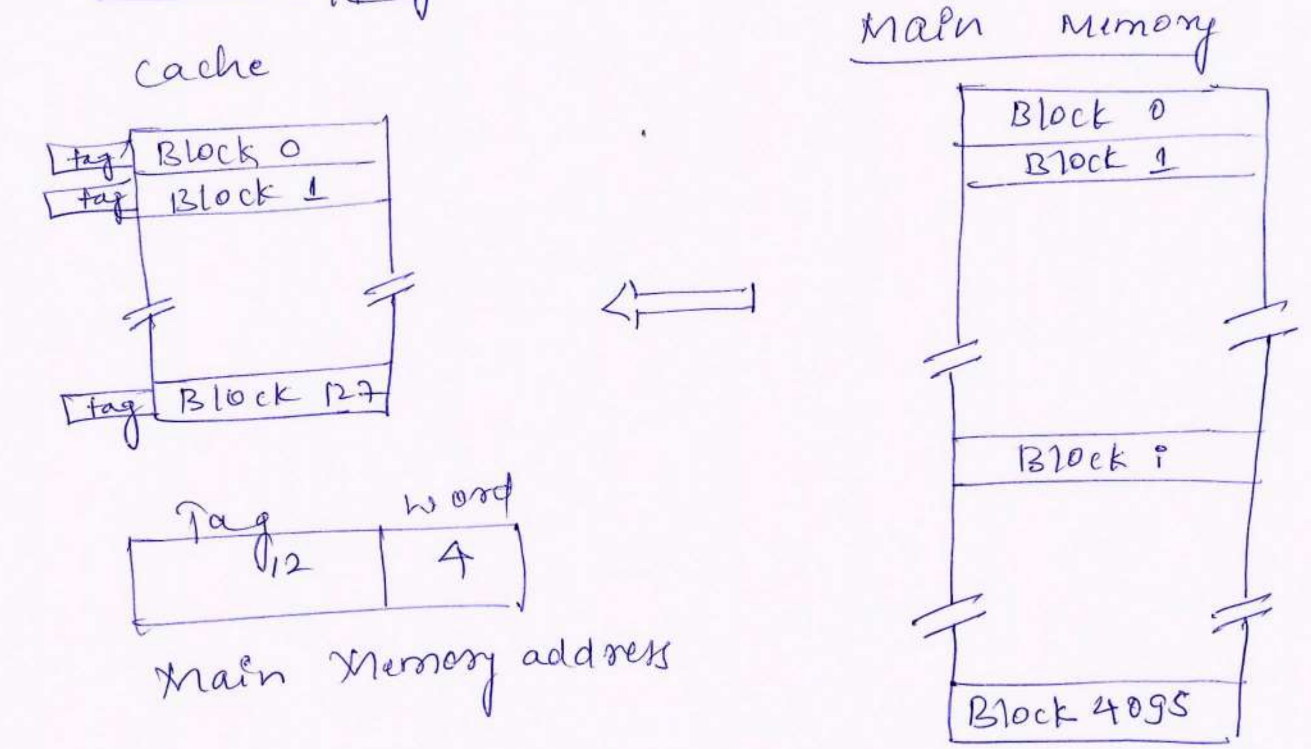


fig: Associative - Mapped cache

Associative mapping method is more flexible method, in which main memory block can be placed into any cache block position

Main memory address contains only two elements one is tag & other is word. The tag is of 12-bits The 12 tag bits are used to identify the main memory in the cache memory. when processor sends the address then MSB 12-bits are used to search the presence of that address in cache memory i.e it searches all the tags of cache memory. This is called as Associative Mapping techniques.

9 a. Draw the single bus architecture and write the control sequence for execution of instruction ADD (R3), R1 — Mark 10

Sol :-

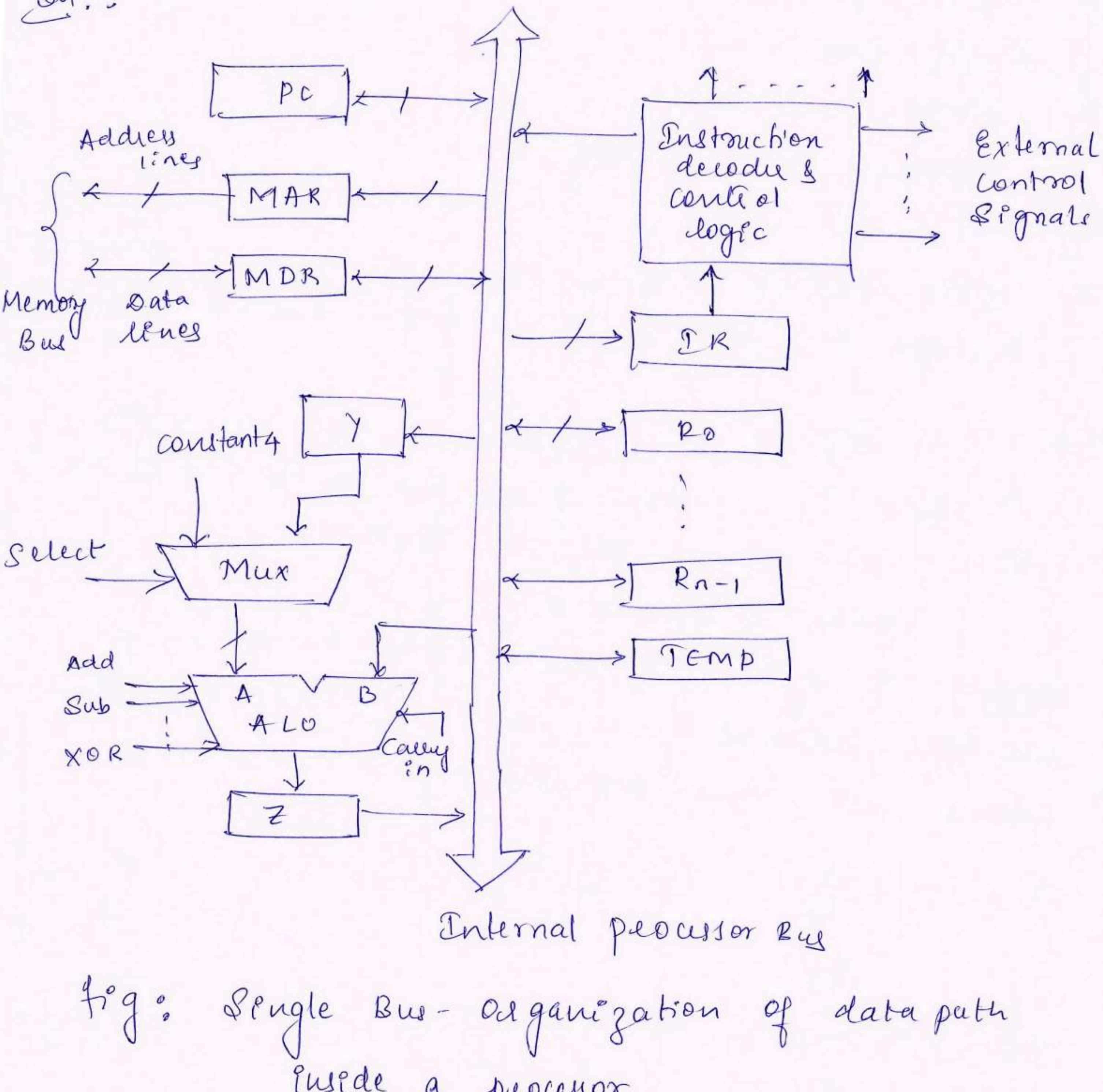


fig: Single Bus - Organization of data path inside a processor

Add (R3), R1

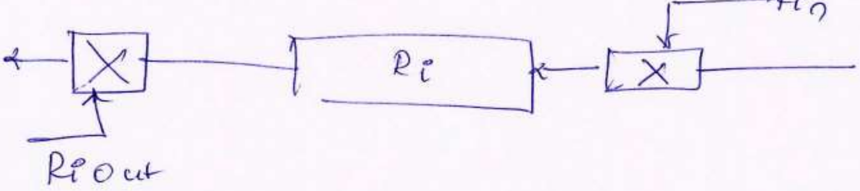
Steps

1. PC out, MARin, Read, Select4, Add, Zin
2. Zout, PCin, Yin, WMFC
3. MDR out, IRin
4. R3out, MARin, Read
5. R1out, Yin, WMFC
6. MDRout, Select Y, Add, Zin
7. Zout, R1in, End

9b. With Suitable diagram, explain the concept of register transfer and fetching of word from memory —10 marks

Sol:- Each register uses "two control signals" in order to place the contents of register on the bus & to load the data on bus to register.

Consider a register Ri where in the input & output of register Ri are connected to the bus via switches controlled by the two control signals called Riin and Riout respectively.



when Ri in = 1, the data on the data bus are loaded into Ri & when Riout = 1 the content of the register are placed on the bus.

when the control signal Riin and Riout = 0 then data cannot be transferred b/w the processor bus & registers.

OR

10 a. with neat diagram, explain the flow of 4-stage pipeline operation — Marks 10

Sol :- Instruction is divided into 4-steps as follows

1) F - Fetch: read instruction from memory

2) D Decode: Decode the instruction reads the source operand

3) E Execute: Perform the specified operation by the instruction

4 W write : Store the result to the specified destination location.

Clock cycle	1	2	3	4	5	6	7
Instruction							
I1	F1	D1	E1	W1			
I2		F2	D2	E2	W2		
I3			F3	D3	E3	W3	
I4				F4	D4	E4	W4

fig: Instruction divided into 4 steps.

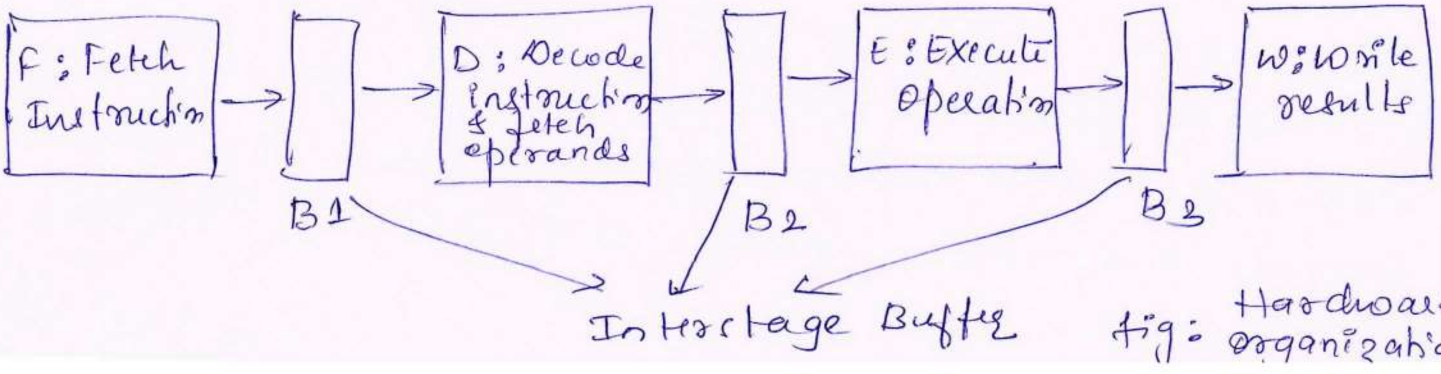


fig: Hardware organization

Fetching a word from memory

To fetch a word of information from memory the processor has to specify the address of the memory location where this information is stored & require a Read operation. The processor transfers the required address to MAR, whose output is connected to the address lines of the memory bus. At the same time, the processor uses the control lines of the memory bus to indicate that a Read operation is needed. when the requested data are received from the memory they are stored in register MDR, from where they can be transferred to other registers in the processor.

The connection for register MDR is as illustrated below

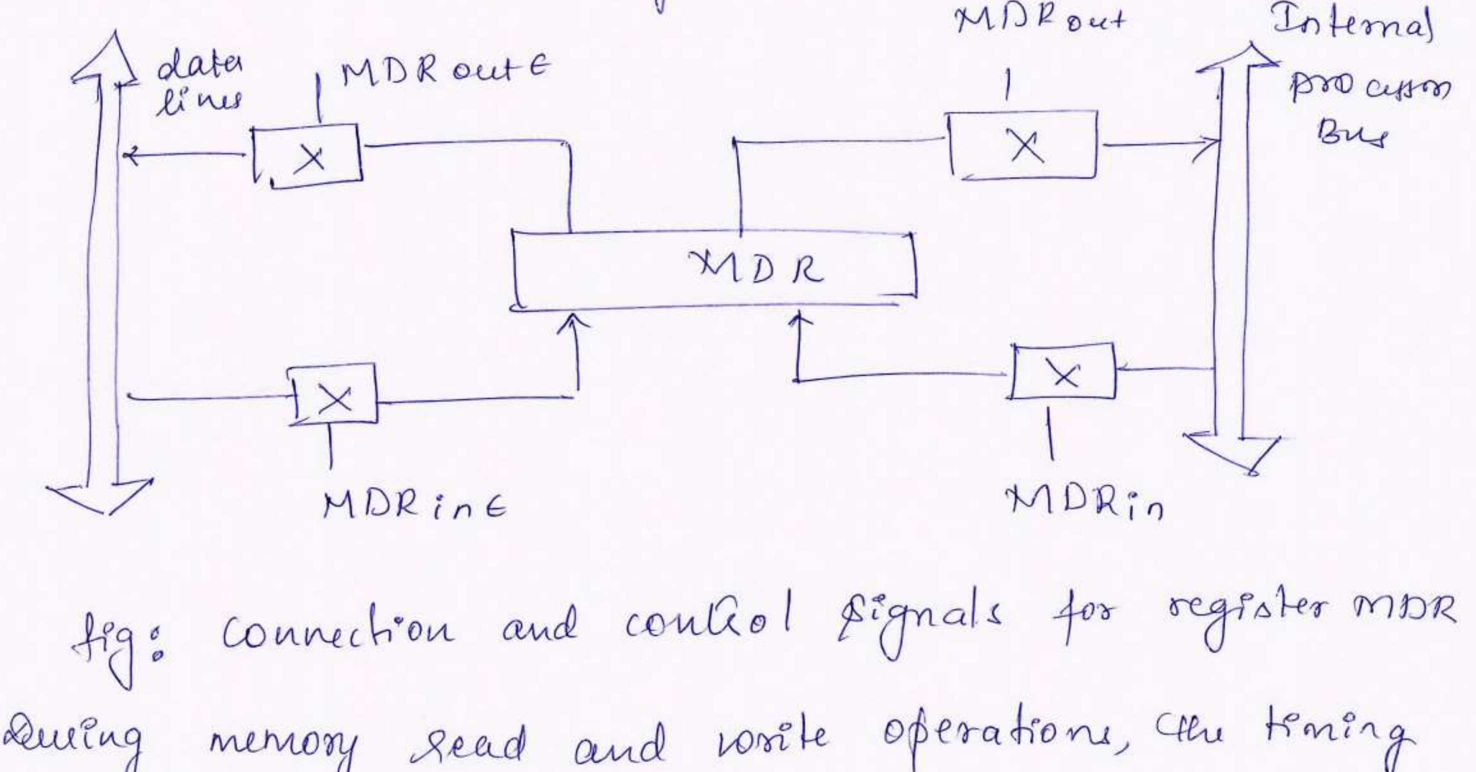


fig: connection and control signals for register MDR

During memory read and write operations, the timing of internal processor operations must be co-ordinated with the response of the addressed device on the memory bus. The processor completes one internal data transfer in one clock cycle.

In 4- stage pipeline, all the 4 - units are capable of performing their tasks simultaneously without interfering one another. Information is passed from one stage to another stage through immediate buffer.

Example, consider the things that happens at 4th clock cycle period

1. Instruction I_4 is fetched, while instruction I_3 which is fetched during clock cycle 3 is stored in buffer B1 i.e B1 contains I_3 & it is decoded by decode unit in clock cycle 4

2. B2 contains I_2 and that will be executed by execution unit during clock cycle 4. It also contains source operand & also information need to store the result after execution.

3. The executed result of I1 instruction i~~s~~ will be stored in B3 i.e result of I1 stored in destination location in clock cycle 4.

10 b. Explain the role of cache memory and pipe line performance — Marks 10

Sol :- Role of Cache Memory. Each stage in pipeline is expected to complete the operation in one clock cycle.

Hence the clock period should be sufficiently large to complete the tasks.

Pipeling is effective in improving the performance

In case of instruction fetch, it is about to access. the main memory, it requires more time to access the instruction therefore clock cycle time period should be equal or greater than a complete fetch operation.

Thus everytime if instruction needs to be fetched from the main memory then pipeling concept does not or it adds little value.

Therefore cache memory ~~at~~ solves the main memory access problem, Access time of cache memory is same as time taken by other units/stages of pipeline.

Pipe line Performance :- There might be variety of reasons, where one of the pipeline stage may not be able to complete its processing tasks for a given instruction in allotted time.

The pipeline operation may be stalled for number of clock cycles & it is called as pipeline stalling. Pipeline stalling is delay generation in processing the instruction. Any condition that causes the pipeline to stall is Hazard. There are three types of Hazard that causes the pipeline to get stalled.

They are

1) Data Hazard
2) Instruction or control Hazard
3) Structural Hazard