CBCS SCHEME

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

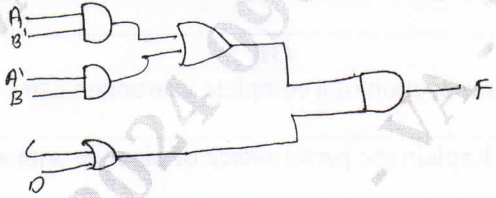
BCS302

Third Semester B.E./B.Tech. Degree Examination, Dec.2023/Jan.2024

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.	Obtain a minimum product of sums with a Karnaugh map. $F(w, x, y, z) = x' z' + wyz + w' y' z' + x' y$.	10	L3	CO1
	b.	Find the minimum sum of products for each function using a Karnaugh map i) $F_1(a, b, c) = M_0 + M_2 + M_5 + M_6$ ii) $F_2(d, e, f) = \Sigma m(0, 1, 2, 4)$ iii) $F_3(r, s, t) = rt' + r's' + r's$	10	L3	CO1
		OR			
Q.2	a.	Identify the prime implicants and essential prime implicants of the following functions: i) $f(A, B, C, D) = \Sigma\,(1, 3, 4, 5, 10, 11, 12, 13, 14, 15)$ ii) $f(W, X, Y, Z) = \Sigma(0, 1, 2, 5, 7, 8, 10, 15)$.	10	L3	CO1
	b.	Write the verilog code for the given expression using dataflow and behavioral model where $Y = (AB' + A'B)(CB + AD)(AB'C + AC)$.	5	L2	CO1
	c.	Write the verilog code and time diagram for the given circuit with propagation delay where the AND, OR gate has a delay of 30ns and 10ns.  Fig.Q.2(c)	5	L2	CO1
		Module – 2			
Q.3	a.	What is Latch? With neat diagram, explain S-R latch using NOR gate. Derive characteristics equation.	10	L3	CO2
	b.	What is priority encoder? Design 4:2 priority encoder with necessary diagrams.	10	L3	CO2
		OR			
Q.4	a.	Design and explain four bit adder with carry look ahead.	10	L3	CO2
	b.	What is multiplexer? Design 9:1 mux using 2:1 mux.	10	L3	CO2

					BCS302
		Module – 3			
Q.5	**a.**	Explain four types of operation performed by computer with an example.	**10**	**L2**	**CO3**
	b.	Show how below expression will be executed in one address, two address zero address and three address processor in an accumulator organization $X = (A * B) + (C * D)$.	**10**	**L1**	**CO3**
		OR			
Q.6	**a.**	What is addressing mode? Explain different types of addressing mode with an examples.	**10**	**L2**	**CO3**
	b.	With a neat diagram, explain basic operational concepts of a computer.	**10**	**L2**	**CO3**
		Module – 4			
Q.7	**a.**	Explain the following with respect to interrupts with diagram. i) Vector interrupt ii) Interrupt nesting iii) Simultaneous request.	**10**	**L2**	**CO3**
	b.	Explain Direct Memory Access with a neat diagram.	**10**	**L2**	**CO3**
		OR			
Q.8	**a.**	What is Bus arbitration? Explain different types of bus arbitration.	**10**	**L2**	**CO3**
	b.	Discuss different types of mapping functions of coaches.	**10**	**L2**	**CO3**
		Module – 5			
Q.9	**a.**	Draw and explain the single-bus organization of the data path inside a processor.	**10**	**L2**	**CO4**
	b.	List out the actions needed to execute the instruction ADD (R3), R1 write and explain the sequence of control steps for the execution of the same.	**10**	**L2**	**CO4**
		OR			
Q.10	**a.**	Analyze how does execution of a complete instruction carry out.	**10**	**L4**	**CO4**
	b.	What is pipeline? Explain the performance of pipeline with an example.	**10**	**L4**	**CO4**

* * * * *

Digital Design and computer organization

BCS 302 Dec 2023/ Jan 2024

Module-1

Q1. a) obtain a minimum product of sums with a Karnaugh map

$F(w,x,y,z) = x'z' + wyz + w'y'z' + x'y$

$= \cancel{x'z'}$ $= w'x'y'z' + wxyz + w'x'y'z' + w'x'yz'$

$w'x'yz'$ $wxyz$ $w'xy'z'$ $w'x'yz$

$wx'y'z'$ $wx'yz'$

$wx'yz'$ $wx'yz$

$= \underline{m_0} + \underline{m_2} + m_8 + \underline{m_{10}} + \underline{m_{11}} + m_{15} + \underline{m_0} + m_4$

$+ \underline{m_2} + m_3 + \underline{m_{10}} + \underline{m_{11}}$

$= \Sigma m(0, 2, 3, 4, 8, 10, 11, 15)$ minimum SOP

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

1 Quad, 3 duals.

$= x'z' + w'y'z' + w'x'y + wyz$

minimum pos

$F(w,x,y,z) = \Pi M(1, 5, 6, 7, 9, 12, 13, 14)$

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

1 quad, 2 dual

$= y'z + w'xy + wxz'$

b) Find the minimum Sum of products for each function using a Karnaugh map.

(i) $F_1(a,b,c) = M_0 + M_2 + M_5 + M_6$

a \ bc	00	01	11	10
0	0_0	1_1	1_3	0_2
1	1_4	0_5	1_7	0_6

1 single, 2 duals.

$= (b'+c)(a+c)(a'+b+c')$

(ii) $F_2(d,e,f) = \Sigma m(0,1,2,4)$

d \ ef	00	01	11	10
0	1_0	1_1	0_3	1_2
1	1_4	0_5	0_7	0_6

$= e'f' + d'e' + d'f'$

(iii) $F_3(r,s,t) = rt' + r's' + r's$

$= rs't' + r's't' + r'st'$
$\quad rst' \quad r's't \quad r'st$

$= 100 + 000 + 010$
$\quad 110 \quad 001 \quad 011$

$= m_4 + m_0 + m_2 + m_6 + m_1 + m_3$

$= \Sigma m(0,1,2,3,4,6)$

r \ st	00	01	11	10
0	1_0	1_1	1_3	1_2
1	1_4	0_5	0_7	1_6

2 quad

$= r' + t'$

2. a) Identify prime implicants and essential prime implicants of the following functions.

i) $f(A, B, C, D) = \Sigma(1, 3, 4, 5, 10, 11, 12, 13, 14, 15)$

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

PI

$= B'C' + AC + AB + A'B'D$

EPI

$= B'C' + AC + A'B'D$

=

ii) $f(w, x, y, z) = \Sigma(0, 1, 2, 5, 7, 8, 10, 15)$

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

PI

$= x'z' + w'y'z + xyz + w'x'y' + w'xy'$

EPI

$= x'z' + w'y'z + xyz$

b) write the Verilog code for the given expression using dataflow and behavioral model where

$y = (AB' + A'B)(CB + AD)(AB'C + AC)$

Dataflow

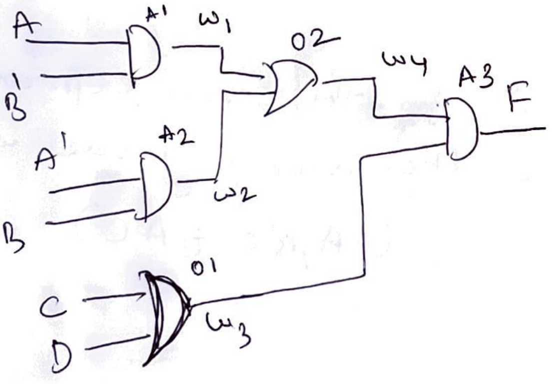
```
model df_y (A, B, C, D, y);
  input A, B, C, D;
  output y;
  assign y = (((A && (!B)) || ((!A) && B)) &&
             ((C && B) || (A && D)) &&
             ((A && (!B) && C) || (A && C)))

  end module
```

Endmodule

Behavioral model

c. write the verilog code and time diagram for the given circuit with propagation delay where the AND, OR gate has a delay of 30ns and 10ns.



```
module stru_F (F, A, B', A', B, C, D);
  input A, B', A', B, C, D;
    output F;
  wire w1, w2, w3, w4;
  and A1 (w1, A, B');
  and A2 (w2, A', B);
  and A3 (F, w3, w4);
  or O1 (w3, C, D);
  or O2 (w4, w1, w2);
end module.
```

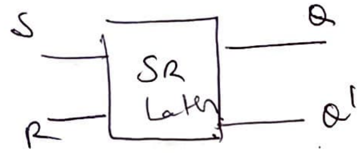
Timing diagram

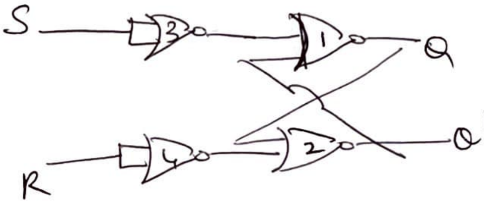
Module-2

Q3 a) What is latch? With neat diagram, explain S-R latch using NOR gate. Derive characteristic equation.

Latches are digital circuits that store single bit of information and hold its value until it is updated by new input signals. They are used in digital systems as temporary storage elements to store binary information.

SR-latch





T.T.

S	R	Q	Q'	
0	0	?	?	not allowed
0	1	0	1	reset
1	0	1	0	set.
1	1	No change		

Chc table

Qt	S	R	Q_{t+1}	Q_{t+1}'
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	0	1
1	0	0	0	0
1	0	1	0	1
1	1	0	1	0
1	1	1	1	0

K-Map

Q_{t+1}

Qt \ SR	00	01	11	10
0	0 $_0$	0 $_1$	0 $_3$	1 $_2$
1	0 $_4$	0 $_5$	1 $_7$	1 $_6$

$Q_{t+1} = SR' + Q_t S$

Q_{t+1}'

Qt \ SR	00	01	11	10
0	0	1	1	0
1	0	1	0	0

$Q_{t+1}' = S'R + Q_t' R$

b) what is priority Encoder ? Design 4:2 priority Encoder with necessary diagrams.

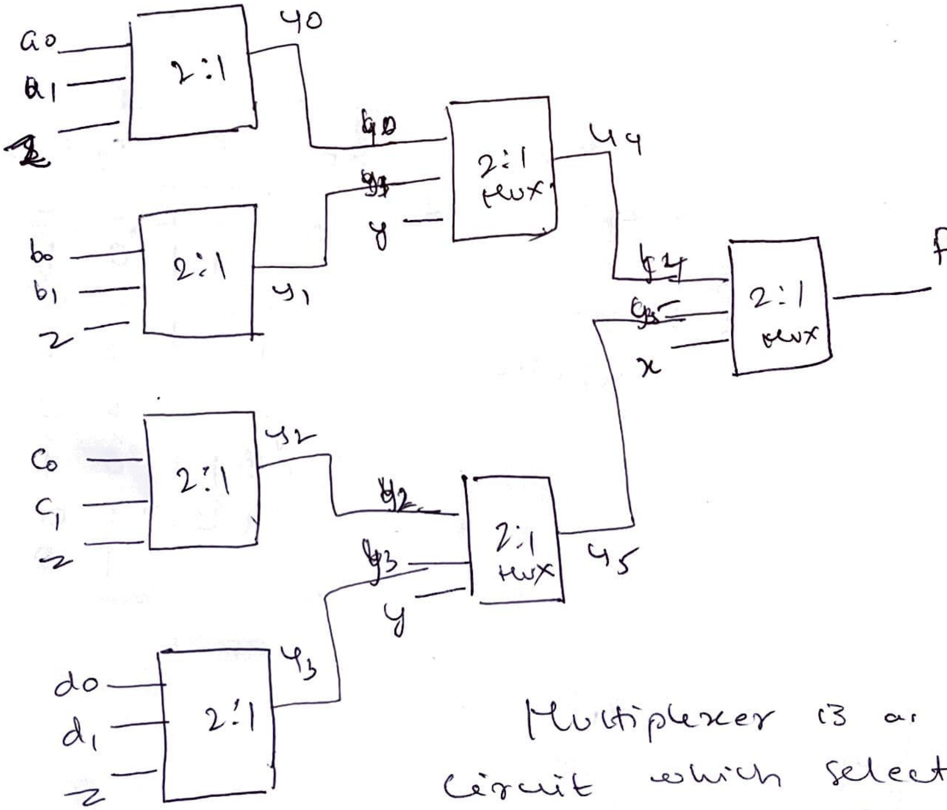
refer Page No. 42
model paper 2 . 4 (a).

Q 4 a) Design and Explain Four bit adder with carry look ahead.

Page No. 9
Model paper 1. 4 (a)

b) what is multiplexer ? Design 9:1 mux using 2:1 mux
8:1

x y z



Multiplexer is a Combinational Circuit which selects one of 2^n number of input lines with the help of n number of select lines and route it to only one output.

Q 5. a) Explain four types of operation performed by computer with example.

The computer is designed to perform 4 types of operations,

① Data transfer operations

② ALU operations

③ program sequencing and control

④ I/o operations

① Data Transfer operations

(a) Data transfer between two registers.

Format : opcode Source Destination

The processor uses MOV instruction to perform data transfer operation between two registers. The mathematical representation of this instruction is $R_1 \longrightarrow R_2$

Ex: MOV R_1, R_2 : R_1 & R_2 are processor registers.

This instruction transfers the contents of R_1 to R_2.

Before Execution

R_1	34h
R_2	65h

After execution

R_1	34h
R_2	34h

(b) Data transfer from memory to register

The processor uses LOAD instruction to perform data transfer operation from memory to register.

The mathematical representation of this instruction is

$[LOCA] \rightarrow ACC$, where ACC is accumulator.

Format:	opcode	operand
	LOAD	LOCA

For this instruction memory location is the Source and Accomulator is the destination.

c) Data transfer from accumulator register to memory.

The processor uses STORE instruction to perform data transfer operation accumulator register to memory location.

The mathematical representation of this instruction is $[ACC] \rightarrow LOCA$

Format:	opcode	operand
	STORE	LOCA

Ex:

For this instruction accumatator is the Source and memory location is the destination.

2. ALU operations

The Instructions are designed to perform arithmetic operations such as Addition, Subtraction, Multiplication, Division as well as logical operations such as AND, OR and NOT operations.

Ex1: ADD R_0, R_1

Adds the content of R_0 with the content of R_1 and result is placed in R_1.

$R_1 \leftarrow [R_0] + [R_1]$

Ex2: SUB R_0, R_1

Subtracts the content of R_0 from the content of R_1 and result is placed in R_1.

$R_1 \leftarrow [R_0] - [R_1]$

Ex3: AND R_0, R_1

Logically multiplies the contents of R_0 with the content of R_1 and result is stored in R_1.

$R_1 \leftarrow [R_0]$ AND $[R_1]$

Ex3: NOT R_0; It performs the function of complementation.

3. I/O operations: The instructions are designed to perform INPUT and OUTPUT operations. The processor uses MOV instruction to perform I/O operations.

The input device consists of one tempory register called as DATAIN register and output register consists of one temporary register called as DATAOUT register.

a) Input operation: It is a process of transferring one WORD of data from DATAIN register to processor register

Ex: MOV DATAIN, R0 ; R0 ← [DATAIN]

b) output operation: It is a process of transferring one WORD of data from processor register to DATAOUT register.

Ex: MOV R0, DATAOUT ; [R0] ← DATAOUT

4. program Sequencing and control

It refers to the order in which instruction in a program are executed and how the flow of control is managed.

5. b) Show how below expression will be executed in one address, two address, zero address and three address processor in an accumulator organization

$$X = (A * B) + (C * D)$$

i) 3-address

```
START:  MPY  A, B, T1 ;   [A] * [B] -> T1
        MPY  C, D, T2 ;   [C] * [D] -> T2
        ADD  T1, T2, X ;  [T1] + [T2] -> X
        HLT
```

ii) Two-address

```
START:  MPY  A, B ;   A <- [A] * [B]
        MPY  C, D ;   C <- [C] * [D]
        ADD  A, C ;   A <- [A] + [C]
        MOV  X, A ;   X <- [A]
        HLT.
```

(iii) One- address instruction

```
START:   LOAD A    ; ACC ← [A]
         MPY B     ; ACC ← [ACC] * [B]
         STORE X   ; X ← [ACC]
         LOAD C    ; ACC ← [C]
         MPY D     ; ACC ← [ACC] * [D]
         ADD X     ; ACC ← [ACC] + [X]
         STORE X   ; X ← [ACC]
         HLT.
```

(iv) Zero address instruction

```
START:   LOAD A
         LOAD B
         MPY
         LOAD C
         LOAD D
         MPY
         ADD
         STORE X
         HLT.
```