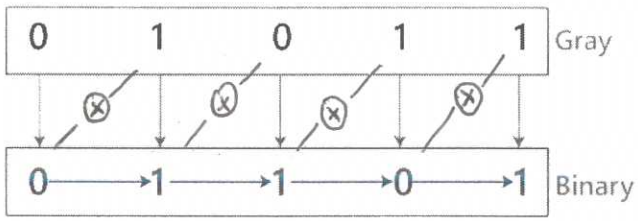
305

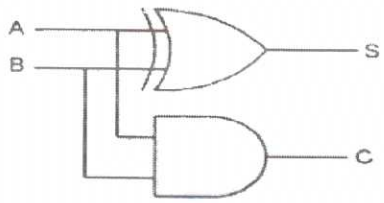
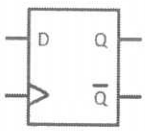
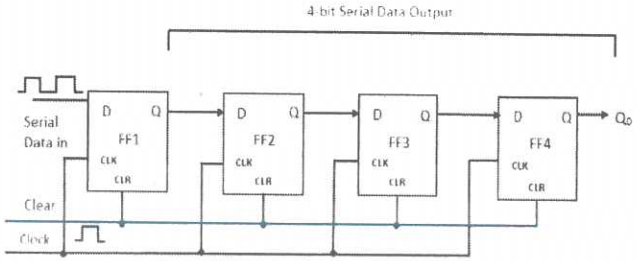
13

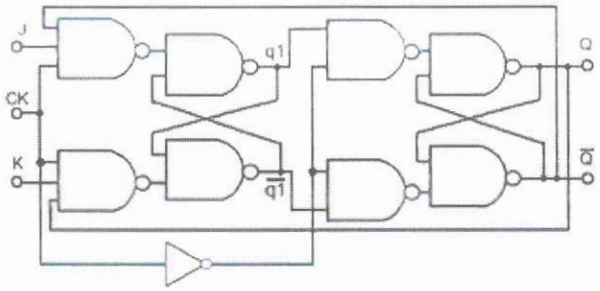
Scoring Indicators

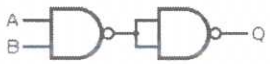
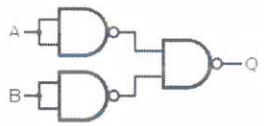
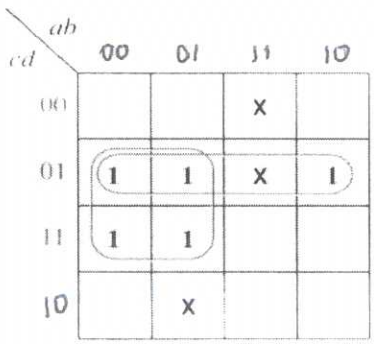
COURSE NAME: DIGITAL ELECTRONICS

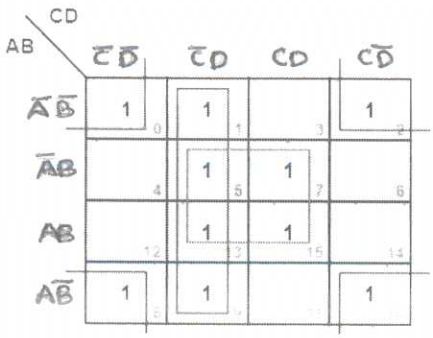
COURSE CODE: 3044 (21) **QID: 2110220223**

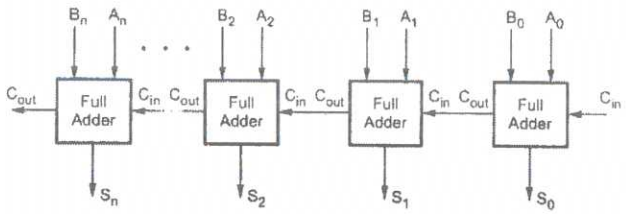
Q.NO	Scoring Indicators	Split Score	Sub Total	Total Score
	PART A			**9**
I.1	Hexadecimal system	1	1	1
I.2	01010	1	1	1
I.3	Emitter Coupled Logic (ECL)	1	1	1
I.4	2	1	1	1
I.5	Toggle	1	1	1
I.6	Combinational Circuit	1	1	1
I.7	3	1	1	1
I.8	the total number of states = $2^4 = 16$ states	1	1	1
I.9	Electrically Erasable Programmable Read-only Memory	1	1	1
	PART B			**24**
II.1	6 5 0 E + 0 8 C 1 6 D C F	3	3	3
II.2	$\overline{A\,B} = \overline{A} + \overline{B}$ $\overline{A + B} = \overline{A}\,\overline{B}$	(1.5*2)	3	3
II.3		3	3	3

<table>
<tr>
<td>II.4</td>
<td>The half adder circuit has two inputs: A and B, which add two input digits and generates a carry and a sum.
S = A ⊕B
C =AB
<table><tr><th colspan="2">Inputs</th><th colspan="2">Outputs</th></tr><tr><th>A</th><th>B</th><th>SUM</th><th>CARRY</th></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td></tr></table>  </td>
<td>Fig:1
Exp :2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>II.5</td>
<td>D Flip-flop
Symbol
 
Table of truth
<table><tr><th>clk</th><th>D</th><th>Q</th><th>Q̄</th></tr><tr><td>0</td><td>0</td><td>Q</td><td>Q̄</td></tr><tr><td>0</td><td>1</td><td>Q</td><td>Q̄</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td></tr></table>•</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>II.6</td>
<td>Serial-in to Serial-out (SISO) – the data is shifted serially “IN” and “OUT” of the register, one bit at a time in either a left or right direction under clock control.
  </td>
<td>Fig:2
Exp:1</td>
<td>3</td>
<td>3</td>
</tr>
</table>

<table>
<tr><td>II.7</td><td>
<table>
<tr><th>Sr. No.</th><th>Combinational circuits</th><th>Sequential circuits</th></tr>
<tr><td>1.</td><td>In combinational circuits, the output variables are at all times dependent on the combination of input variables.</td><td>In sequential circuits, the output variables dependent not only on the present input variables but they also depend up on the past history of these input variables.</td></tr>
<tr><td>2.</td><td>Memory unit is not required in combinational circuits.</td><td>Memory unit is required to store the past history of input variables in the sequential circuit.</td></tr>
<tr><td>3.</td><td>Combinational circuits are faster in speed because the delay between input and output is due to propagation delay of gates.</td><td>Sequential circuits are slower than the combinational circuits.</td></tr>
<tr><td>4.</td><td>Combinational circuits are easy to design.</td><td>Sequential circuits are comparatively harder to design.</td></tr>
<tr><td>5.</td><td>Parallel adder is a combinational circuit.</td><td>Serial adder is a sequential circuit.</td></tr>
</table>
</td><td>Any 3
(3*1)</td><td>3</td><td>3</td></tr>
<tr><td>II.8</td><td> 
</td><td>Fig :3</td><td>3</td><td>3</td></tr>
<tr><td>II.9</td><td>
<table>
<tr><th>Synchronous Counter</th><th>Asynchronous Counter</th></tr>
<tr><td>All flip flops are triggered with same clock.</td><td>Different clock is applied to different flip flops.</td></tr>
<tr><td>It is faster.</td><td>It is lower</td></tr>
<tr><td>Design is complex.</td><td>I Design is relatively easy.</td></tr>
<tr><td>Decoding errors not present.</td><td>Decoding errors present.</td></tr>
<tr><td>Any required sequence can be designed</td><td>Only fixed sequence can be designed.</td></tr>
</table>
</td><td>Any 3
(3*1)</td><td>3</td><td>3</td></tr>
<tr><td>II.10</td><td>Random Access Memory (RAM) is used to store the programs and data being used by the CPU in real-time. The data on the random-access memory can be read, written, and erased any number of times. RAM is a hardware element where the data</td><td>3</td><td>3</td><td>3</td></tr>
</table>

	being currently used is stored. It is a volatile memory. Types of RAM: 1. **Static RAM**, or (SRAM) which stores a bit of data using the state of a six-transistor memory cell. 2. **Dynamic RAM**, or (DRAM) which stores a bit data using a pair of transistor and capacitor which constitute a DRAM memory cell.			
	PART C			**42**
III.1	Universal Gates: A universal gate is a gate which can implement any Boolean function without need to use any other gate type **Implementing NOT, AND & OR using only NAND Gate** **NOT**  **AND**  **OR** 	Fig :3*2 Exp :1	7	7
III.2	 **F = a'd + c'd**	Fig:4 Ans: 3	7	7

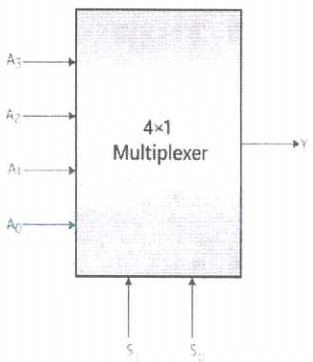
III.3	 **F = C'D +BD+B'D'**	Fig :4 Ans :3	7	7
III.4	The binary coded decimal (BCD) is a type of binary code used to represent a given decimal number in an equivalent binary form. The most common BCD code is the 8421 BCD code. 8, 4, 2 and 1 represent the weights of different bits in the four-bit groups, starting from the (MSB) most significant bit (to extreme left) and proceeding towards the least significant (LSB) bit. Decimal | Binay (BCD) 8 4 2 1 0 | 0 0 0 0 1 | 0 0 0 1 2 | 0 0 1 0 3 | 0 0 1 1 4 | 0 1 0 0 5 | 0 1 0 1 6 | 0 1 1 0 7 | 0 1 1 1 8 | 1 0 0 0 9 | 1 0 0 1	7	7	7
III.5	A circuit , consisting of n full adders , that will add n-bit binary numbers. The output consists of n sum bits and a carry bit. Cout of one full adder is connected to Cin of the next full adder. The number of full adders used will depend on the number of bits in the binary digits which require to be added	Fig :4 Exp:3		



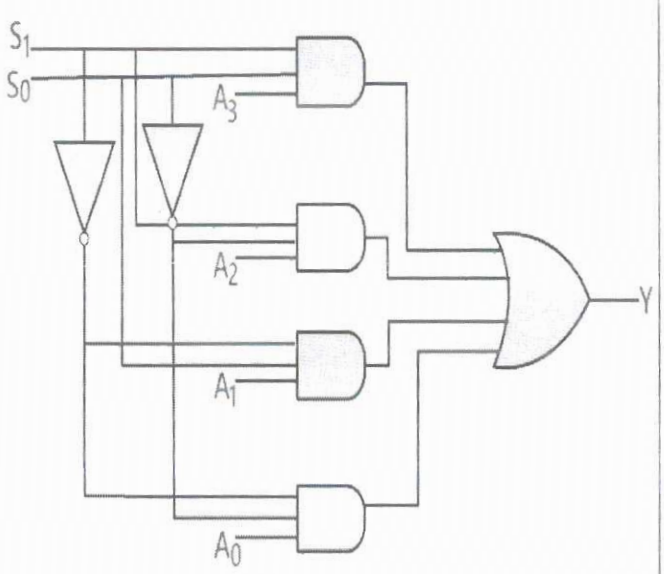
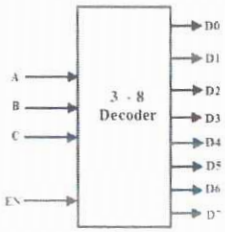
Block diagram of n-bit parallel adder

The bits are added with full-adder.Starting from the least significant position to form the sum and carry. The input carry Cin in the least significant position must be zero. The value Cout of each FA is transferred into the input carry Cin of the next full-adder that adds the bits one higher significant position to left.The sum bits are thus generated starting from the rightmost position and are available as soon as the corresponding previous carry bit is generated.

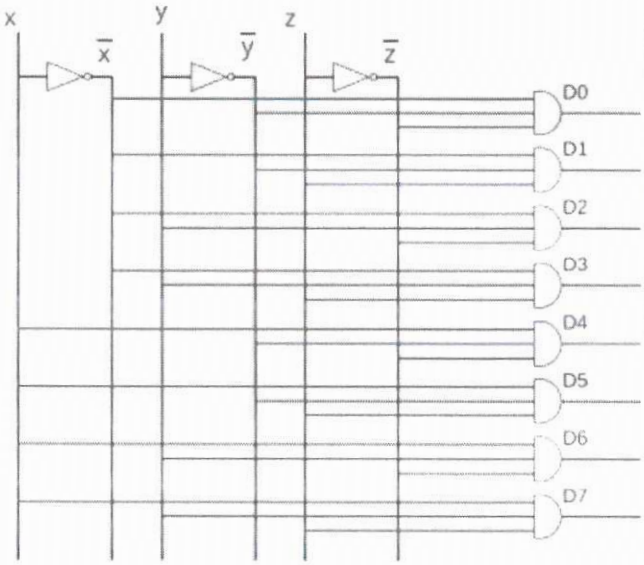
III.6	A multiplexer is a combinational circuit that has 2^n input lines and a single output line. In the 4×1 multiplexer, there is a total of four inputs, i.e., A_0, A_1, A_2, and A_3, 2 selection lines, i.e., S_0 and S_1 and single output, i.e., Y. On the basis of the combination of inputs that are present at the selection lines S_0 and S_1, one of these 4 inputs are connected to the output	Fig :3 Exp:4	7	7

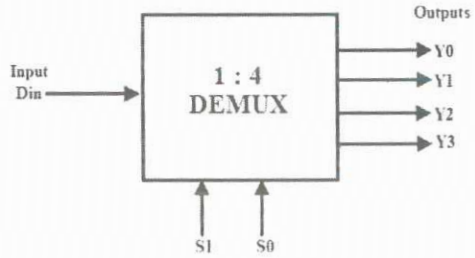


INPUTS		Output
S_1	S_0	Y
0	0	A_0
0	1	A_1
1	0	A_2
1	1	A_3

	The logical expression of the term Y is as follows: $Y=S_1' S_0' A_0+S_1' S_0 A_1+S_1 S_0' A_2+S_1 S_0 A_3$  			
III.7	It is a combinational logic circuit that receives the n input lines and generates a maximum of 2n unique output lines.  	Fig:3 Exp:4	7	7

Inputs			Outputs							
x	y	z	D0	D1	D2	D3	D4	D5	D6	D7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

	$D_0(x, y, z) = \bar{x}\,\bar{y}\,\bar{z}$ $D_1(x, y, z) = \bar{x}\,\bar{y}\,z$ $D_2(x, y, z) = \bar{x}\,y\,\bar{z}$ $D_3(x, y, z) = \bar{x}\,y\,z$ $D_4(x, y, z) = x\,\bar{y}\,\bar{z}$ $D_5(x, y, z) = x\,\bar{y}\,z$ $D_6(x, y, z) = x\,y\,\bar{z}$ $D_7(x, y, z) = x\,y\,z$ 			
III.8	A Demultiplexer is a combinational logic circuit that receives the information on a single input line and transmits the same information over one of 'n' possible output lines. A 1-to-4 demultiplexer has a single input (D), two selection lines (S1 and S0) and four outputs (Y0 to Y3). The input data goes to any one of the four outputs at a given time for a particular combination of select lines.			



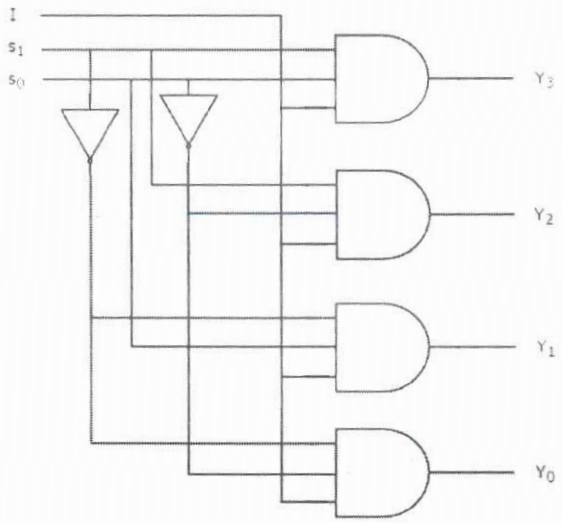
Data Input	Select Inputs		Outputs			
D	S_1	S_0	Y_3	Y_2	Y_1	Y_0
D	0	0	0	0	0	D
D	0	1	0	0	D	0
D	1	0	0	D	0	0
D	1	1	D	0	0	0

$Y0 = \overline{S1}\ \overline{S0}\ D$

$Y1 = \overline{S1}\ S0\ D$

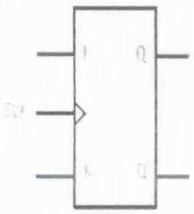
$Y2 = S1\ \overline{S0}\ D$

$Y3 = S1\ S0\ D$



III.9	JK flip-flop has two inputs. And using the two inputs the flip-flop can be set, reset, hold (memory) or toggled. Unlike the SR flip-flop, in the JK flip-flop, when both inputs J and K are 1 then the output of the flip-flop toggles	Fig:3 Exp:4		

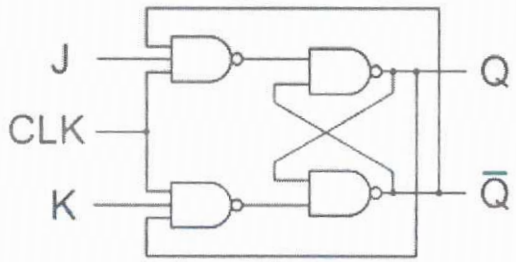
Symbol



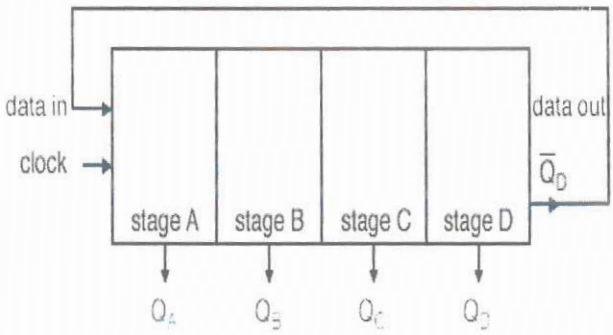
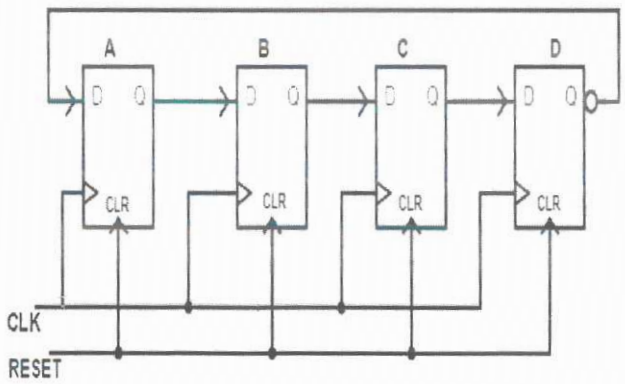
Truth Table

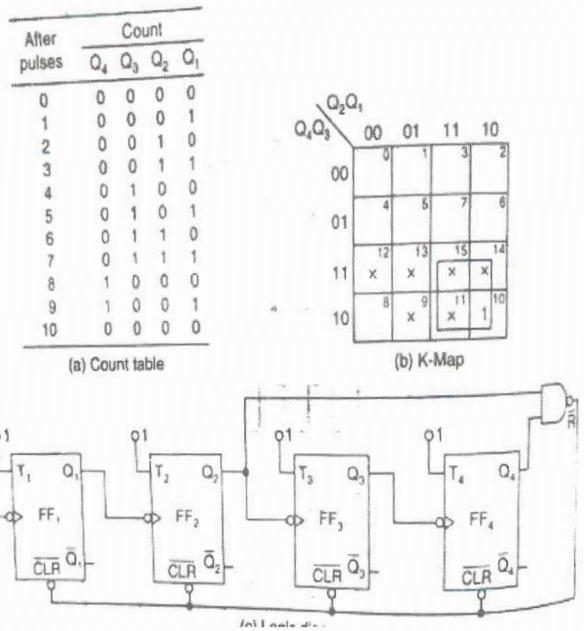
CLK	J	K	Q_{n+1}
↑	0	0	Q_n
↑	0	1	0
↑	1	0	1
↑	1	1	Q_n'

In the JK flip-flop, at the rising edge of the clock, when **J = 0 and K = 0** then flip-flop **retains (holds) the current state**. When **J = 0 and K = 1**, then flop-flop **resets to 0**. When **J = 1 and K = 0**, then flip-flop **sets the output to 1**. And when **J = 1 and K = 1** then **output of the flip-flop toggles**. When the clock signal is low, then irrespective of the value of J and K inputs, the flop-flop retains the present state



III.10	A ring counter is also known as SISO (serial in serial out) shift register counter, where the output of the flip flop is connected to the input of the flip flop which acts as a ring counter	Fig:3 Exp:4	7	7

	<table><tr><th>Q_A</th><th>Q_B</th><th>Q_C</th><th>Q_D</th></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td colspan="4">repeat</td></tr></table>  Ring Counter: shift register output fed back to input 			
III.11	MOD 10 asynchronous counter counts from 0000 to 1001. Rest of the states are invalid. To design the combinational circuit of valid states, following truth table and K-map is drawn	Fig :3 Exp:4	7	7

				
III.12	ROM ROM stands for **Read Only Memory**. The memory from which can only be read but cannot write on it. This type of memory is non-volatile. The information is stored permanently . The single-write, multiple-read nature of ROM makes it most suitable to store boot-up programs and the firmware applications **Programmable Read Only Memory (PROM)** **PROMs** are blank ROMs and contain no pre-recorded instructions. These allow the user to program it only once, using PROM programmers, after which their contents can never be changed. This is because, a typical PROM with all 1's in it will be programmed to have 0's by blowing-off the fuses using high voltage pulses, wherever required. As this is an irreversible process, PROMs are one-time programmable.	7	7	7

	Erasable and Programmable Read Only Memory (EPROM) These devices comprise of an array of transistors characterized by floating-gates, which are programmed one-by-one with the help of high-voltage pulses. However, such a programmed **EPROM** can be erased by exposing them to strong ultra-violet (UV) light for certain duration, after which they can be programmed once again. Nevertheless, multiple-erase operations cause wearing out of the device due to which they lack the unlimited reprogramming ability. These kind of memory devices are generally used to store firmware programs which demand frequent upgrades.			