Scoring Indicators

Paper Set A -3134(2021)

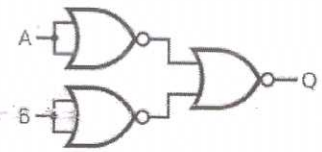
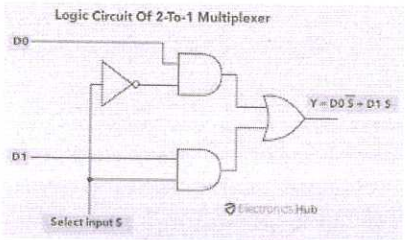
DIGITAL COMPUTER FUNDAMENTALS

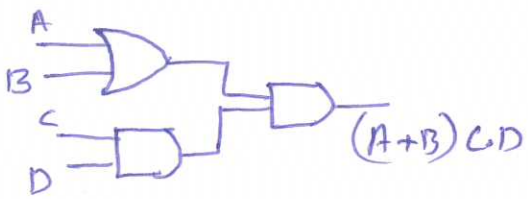
<table>
<tr><th>Q No</th><th>Scoring Indicators</th><th>Split score</th><th>Sub Total</th><th>Total score</th></tr>
<tr><td></td><td>PART A</td><td></td><td></td><td>9</td></tr>
<tr><td>I. 1</td><td>$0010\ 1000_2$</td><td>1</td><td>1</td><td rowspan="10">9</td></tr>
<tr><td>I. 2</td><td>01111_2</td><td>1</td><td>1</td></tr>
<tr><td>I. 3</td><td>1. Sign magnitude form
2. 2's complement form</td><td>Any one</td><td>1</td></tr>
<tr><td>I. 4</td><td>NAND and NOR Gates</td><td>0.5 + 0.5</td><td>1</td></tr>
<tr><td>I. 5</td><td><table><tr><th>A</th><th>B</th><th>Output</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td><td>1</td><td>1</td></tr>
<tr><td>I. 6</td><td>Binary Coded Decimal</td><td>1</td><td>1</td></tr>
<tr><td>I. 7</td><td>Preset and Clear</td><td>0.5+0.5</td><td>1</td></tr>
<tr><td>I. 8</td><td>SUM = A'B + AB'
Carry = A.B</td><td>½ + ½</td><td>1</td></tr>
<tr><td>I. 9</td><td>sequential logic is a type of logic circuit whose output depends on the present value of its input signals as well as the present state.</td><td>1</td><td>1</td></tr>
<tr><td></td><td>PART B</td><td></td><td></td><td>24</td></tr>
<tr><td>II. 1</td><td>25_{10} = 0010 0101 48_{10} = 0100 1000 BCD conversion – 1 marks
Result = 0010 0101 +
0100 1000

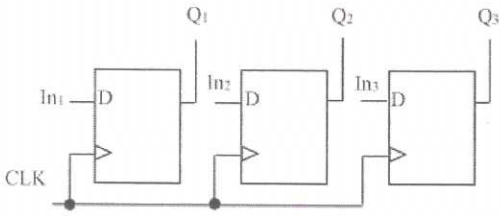
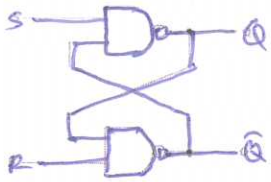
0110 1101 + Addition – 2 marks
0110

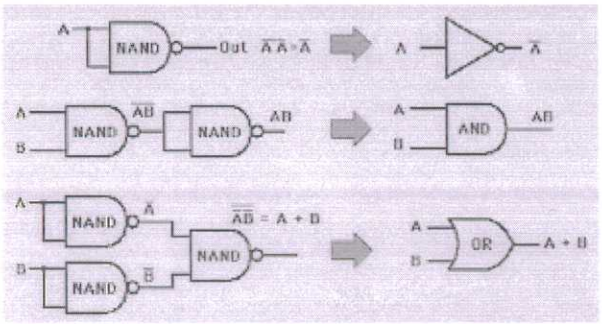
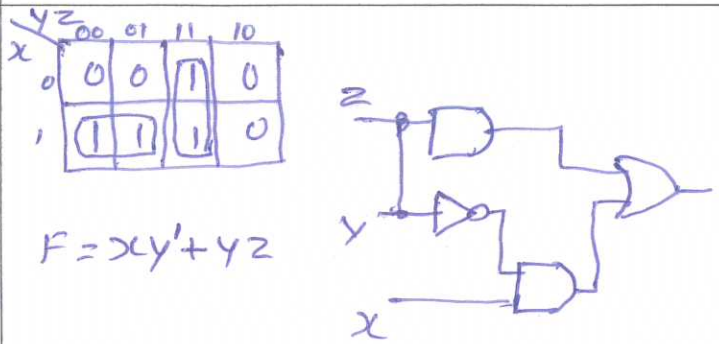
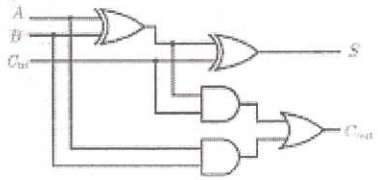
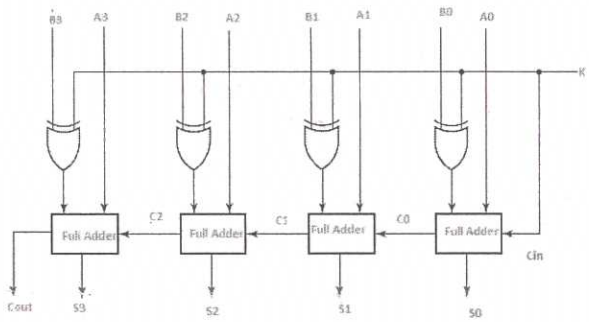
0111 0011 = 73_{10}</td><td>1 + 2</td><td>3</td><td>3</td></tr>
<tr><td>II.2</td><td>A Gray code is an encoding of numbers so that adjacent numbers have a single digit differing by 1.
It is an unweighted code</td><td>3</td><td>3</td><td>3</td></tr>
<tr><td>II.3</td><td>De Morgan's First Law :states that the complement of the product of two or more variables is corresponding to the sum of the complement of each variable.</td><td>1.5 + 1.5</td><td>3</td><td>3</td></tr>
</table>

1/π

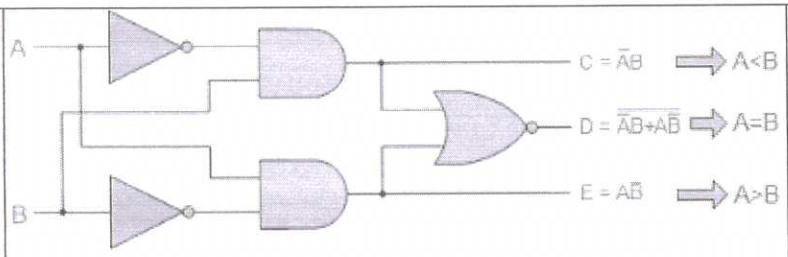
<table>
<tr><td></td><td>Second Law: The complement of the sum of all the terms is equal to the product of the complement of each term.
(or write the Boolean equation)</td><td></td><td></td><td></td></tr>
<tr><td>II.4</td><td>
<table>
<tr><th>X</th><th>Y</th><th>Z</th><th>F (output)</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</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>
<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.5</td><td>X+XY = X(1+Y)
= X(1)
= X = RHS</td><td>3</td><td>3</td><td>3</td></tr>
<tr><td>II.6</td><td> 
</td><td>3</td><td>3</td><td>3</td></tr>
<tr><td>II.7</td><td> 

select input output
0 D0
1 D1</td><td>2+1</td><td>3</td><td>3</td></tr>
<tr><td>II.8</td><td> 
</td><td>3</td><td>3</td><td>3</td></tr>
</table>

II.9	 Inputs are given to In1, In2 and In3 . Outputs are Q1,Q2,Q3, When clock is given, inputs are transferred to output simultaniously. **(figure 2 marks Explanation 1 mark)**	2+1	3	3
II.10	 Alternative figures are also available.	3	3	3
	III PART C			42
III 1a	**$11011 = 1x2^4 + 1x2^3 + 1x\ 2^2 + 1 = 27_{10} = 1B_{16}$**	1+1	2	7
III 1b	**$11A_{16} = 1x16^2 + 1x16 + 10 = 256 + 16 + 10 = 282_{10}$**	2	2	
III 1c	**$242_{10} = 11\ 110\ 010_2 = 362_8$**	1.5 +1.5	3	
III 2 a	Error-detecting codes are a sequence of numbers generated by specific procedures for detecting errors in data that has been transmitted over computer networks. The parity check is done by adding an extra bit, called parity bit, to the data to make the number of 1s either even or odd depending upon the type of parity. The parity check is suitable for single bit error detection only. The two types of parity checking are **Even Parity**– Here the total number of bits in the message is made even. **Odd Parity** – Here the total number of bits in the message is made odd.	1 2 1	4	7
2 b	binary-coded decimal (BCD) is a class of binary encodings of decimal numbers where each digit is represented by a fixed number of bits, usually four. 1000 + 0101 = 1101 since this is > 1001 add 0110 $1101+0110 = 10011 = 13_{10}$	1 2	3	

3		2 + 2.5 + 2.5	7	7
4	 **K map = 2marks, Grouping 2 mark, logic diagram 3 marks**	2+ 2+ 3	7	7
5	 	Inputs			Outputs	
A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1 | **Truth table 4 marks , diagram 3 marks**	4 + 3	7	7
6	 When input K = 0, the circuit add two 4 bit values. The bits B3,B2,B1 and B0 are transferred to the second inputf of Full adders with out changing. So the circuit acts as an adder. When K = 1, the XOR gates complements the B inputs and Cin also becomes 1 , So the 2's complement of B3,B2,B1,B0 is added by the full adder. So it works as a subtractor	4 + 3	7	7
7		6+ 1	7	7

4/7



A	B	A<B	A=B	A>B
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

From the truth table A<B = A'B , A>B = AB' and A equal to B is A XOR B. The circuit produces three outputs according to the input condition.

Figure + truth table 6 marks

Exlanation 1 marks

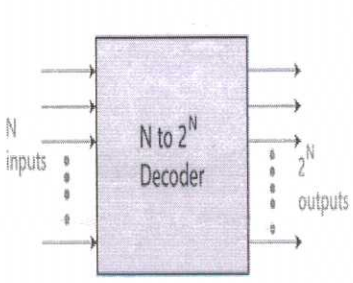
8

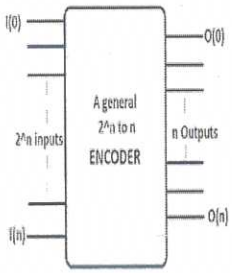
Decoder:

The combinational circuit that change the binary information into 2^N output lines is known as Decoders. The binary information is passed in the form of N input lines. The output lines define the 2^N-bit code for the binary information.

Encoder

An Encoder is a combinational circuit that performs the reverse operation of Decoder.It has maximum of 2^n input lines and 'n' output lines, hence it encodes the information from 2^n inputs into an n-bit code.





Example 1 marks

Binary to decimal decoder,

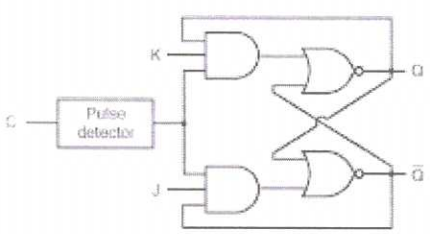
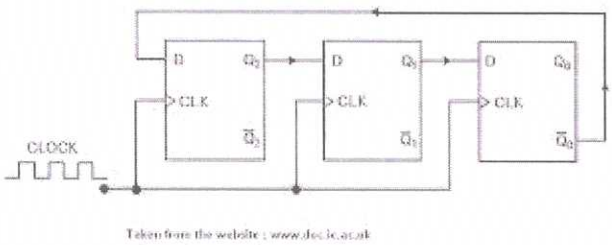
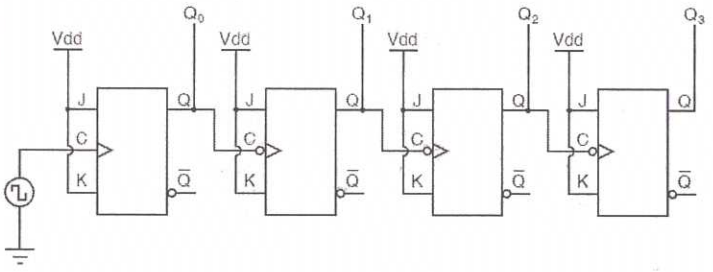
decimal to binary encoders

3+
3+
1

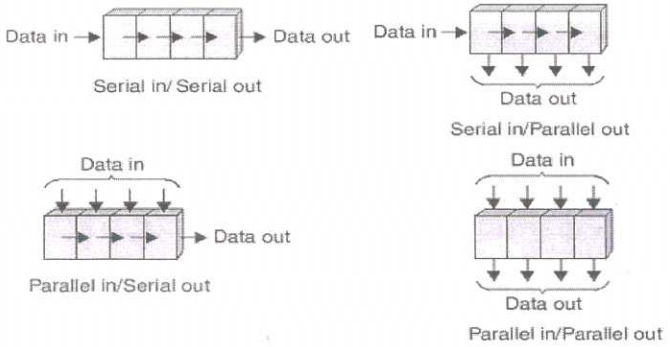
7

7

5/7

9	 **Latch means no change in the output.** **Circuit and truth table – 5 marks** **explanation 2 marks**	5+2	7	7
10	 Taken from the website : www.doc.ic.ac.uk **Figure 4 marks explanation 3 marks** A ring counter is a typical application of the Shift register. The ring counter is almost the same as the shift counter. The only change is that the output of the last flip-flop is connected to the input of the first flip-flop in the case of the ring c ounter but in the case of the shift resister it is taken as output. Except for this, all the other things are the same. No. of states in Ring counter = No. of flip-flop used	4+3	7	7
11	A four-bit "up" counter  All J and K inputs are connected to VDD so for each clock input , the flip flop toggles. Input clock is given to first flop. The output of this is the LSB. The output of previous one act as the clock of the next flipflop. So second flipflop toggles on every 2nd clock, third flip flop toggles on every 4th clock and fourth one flips on every 8th clock. So ths will produce count sequence from 0000 to 1111 and repeats. **(Figure 4 marks explanation 3 marks)**	4+3	7	7

6/7

12	 **figure 4 marks explanation 3 marks** The shift register, which allows serial input (one bit after the other through a single data line) and produces a serial output is known as Serial-In Serial-Out shift register. Since there is only one output, the data leaves the shift register one bit at a time in a serial pattern, thus the name Serial-In Serial-Out Shift Register. The shift register, which allows serial input (one bit after the other through a single data line) and produces a parallel output is known as Serial-In Parallel-Out shift register. The shift register, which allows parallel input (data is given separately to each flip flop and in a simultaneous manner) and produces a serial output is known as Parallel-In Serial-Out shift register. The shift register, which allows parallel input (data is given separately to each flip flop and in a simultaneous manner) and also produces a parallel output is known as Parallel-In parallel-Out shift register.	4+3	7	7

Blue Print - QID -2110220196

DIGITAL COMPUTER FUNDAMENTALS

TED (21) -3134 - SET A

Mark Distribution

Module	Hours/ Module (hi)	Marks/ Module (h^i/H^i) * 123 (±5%)	Type of Questions							
			Part A		Part B		Part C		Total	
			No. of questions	Marks	No. of questions	Marks	No. of questions	Marks	No. of questions	Marks
1	8	23(22-24)	2	2	2	6	4	28	7	
2	11	32(30 -34)	3	3	3	9	2	14	9	
3	12	34(32-36)	2	2	3	9	2	14	7	
4	12	34(32-36)	2	2	2	6	4	28	8	
Total	43	123	9	9	10	30	12	84	31	123

Cognitive Level Distribution

Cognitive Level	Marks	% of Marks
Remembering	8	7
Understanding	45	36
Applying	70	57
Total	123	100