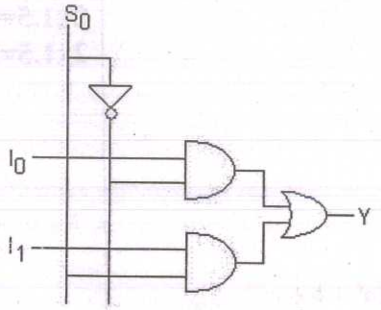
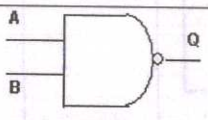
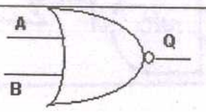
<table>
<tr><td colspan="2">Revision : 2015</td><td colspan="3">Course Code : 3133</td></tr>
<tr><td colspan="5">Course Title : DIGITAL COMPUTER PRINCIPLES</td></tr>
<tr><th>Qst. No.</th><th>Scoring Indicator</th><th>Split up score</th><th>Sub Total</th><th>Total</th></tr>
<tr><td></td><td>PART A</td><td></td><td></td><td></td></tr>
<tr><td>I(i)</td><td>NAND, NOR</td><td>2x1</td><td>2</td><td rowspan="5">10</td></tr>
<tr><td>I(ii)</td><td>Combinational Logic Circuits are memoryless digital logic circuits whose output at any instant in time depends only on the combination of its inputs.</td><td>2x1</td><td>2</td></tr>
<tr><td>I(iii)</td><td>
$Y = I_0S_0' + I_1S_0$
CIRCUIT DIAGRAM OF 2X1 MUX USING GATES</td><td>2</td><td>2</td></tr>
<tr><td>I(iv)</td><td>1. Counters
2. frequency divider
3. shift and storage registers
4. monostable multivibrator
5. timing signal generation
6. data transfer
(any two)</td><td>2x1</td><td>2</td></tr>
<tr><td>I(v)</td><td>1. Weighted Resistor DAC
2. R-2R Ladder DAC</td><td>2x1</td><td>2</td></tr>
<tr><td></td><td>PART B</td><td></td><td></td><td></td></tr>
<tr><td>II(i)</td><td>
NAND
<table><tr><th>A</th><th>B</th><th>Q</th></tr><tr><td>0</td><td>0</td><td>1</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>0</td></tr></table>

NOR
<table><tr><th>A</th><th>B</th><th>Q</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>
Logic symbols - 1.5 marks each
Truth tables - 1.5 marks each</td><td>2x1.5=3
2x1.5=3</td><td>6</td><td>42</td></tr>
</table>

II(ii)

A **minterm** is a product (AND) of all variables in the function, in direct or complemented form.

A **maxterm** is a sum (OR) of all the variables in the function, in direct or complemented form.

a	b	c	f
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

In minterms we can indicate that

$$F = m_0 + m_4 + m_7 = a'b'c' + ab'c' + abc$$

Or we can use maxterms as

$$F = M_1 \cdot M_2 \cdot M_3 \cdot M_5 \cdot M_6 = (a+b+c')(a+b'+c)(a+b'+c')(a'+b+c$$

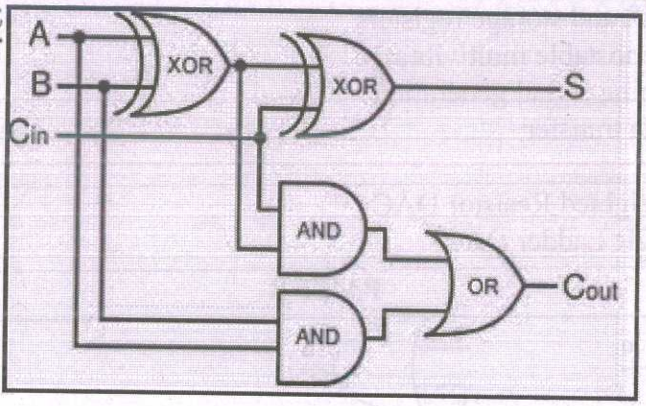
1.5 marks each for definition
1.5 marks each for examples

2x1.5=3
2x1.5=3

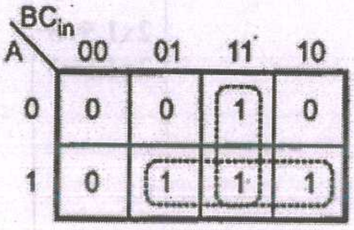
6

II(iii)

A	B	Cin	Sum	Cout
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



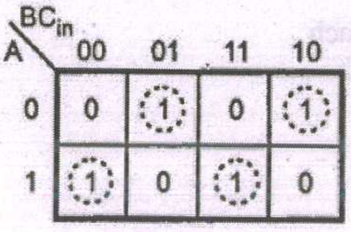
For Carry (C_{out})



A \ BC_{in}	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$C_{out} = AB + A\,C_{in} + B\,C_{in}$

For Sum



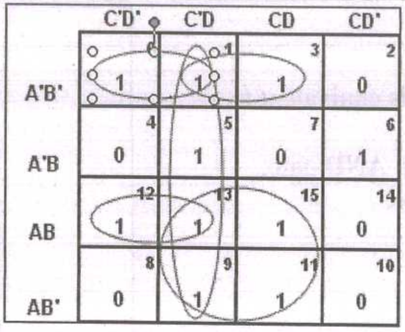
A \ BC_{in}	00	01	11	10
0	0	1	0	1
1	1	0	1	0

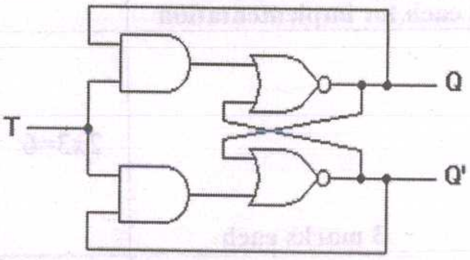
$\text{Sum} = \overline{A}\,\overline{B}C_{in} + \overline{A}B\overline{C}_{in} + A\overline{B}\,\overline{C}_{in} + ABC_{in}$

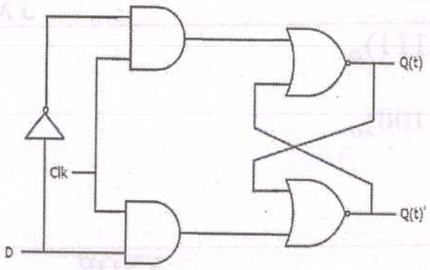
1 Mark for truth table
2 Marks for Karnaugh map
1 Marks for simplification and reduced form
2 mark for diagram

1+2+1+2=6

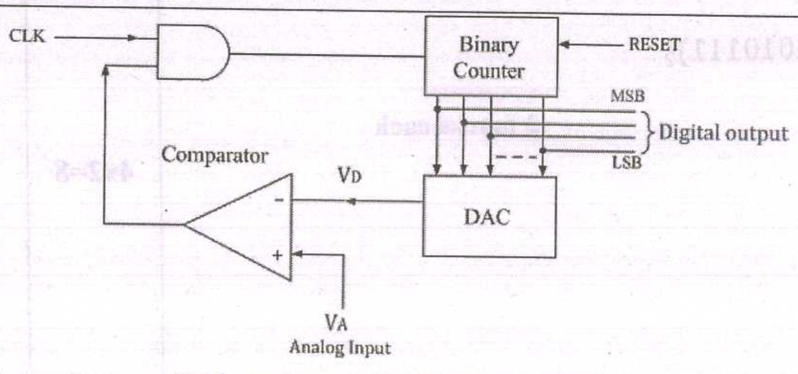
6

<table>
<tr>
<td>II(iv)</td>
<td>


<table>
<tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th><th>F</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>2</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>3</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>4</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>5</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>6</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>7</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>8</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>9</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>10</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>11</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>12</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>13</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>14</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>15</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table>
F = C'D + AD + A'B'C' + A'B'D + ABC' + A'BCD'
2 mark for truth table formation
2 marks for Kmap formation
2 mark for simplification
</td>
<td>2+2+2=
6</td>
<td>6</td>
</tr>
<tr>
<td>II(v)</td>
<td>




2 Marks each for diagram
1 mark each for explanation
</td>
<td>2x2=4
1x2=2</td>
<td>6</td>
</tr>
<tr>
<td>II(vi)</td>
<td>
1.In synchronous counter, all flip flops are triggered with same clock simultaneously. In asynchronous counter, different flip flops are triggered with different clock, not simultaneously.
2.Synchronous Counter is faster than asynchronous counter in operation. Asynchronous Counter is slower than synchronous counter in operation.
3.Synchronous Counter does not produce any decoding errors. Asynchronous Counter produces decoding error.
4.Synchronous Counter is also called Serial Counter. Asynchronous Counter is also called Parallel Counter.
5.Synchronous Counter designing as well implementation are complex due to increasing the number of states. Asynchronous Counter designing as well as implementation is very easy.
6.Synchronous Counter will operate in any desired count sequence. Asynchronous Counter will operate only in fixed count sequence (UP/DOWN).

1.5 Marks each (Any 4 points)
</td>
<td>4x1.5=6</td>
<td>6</td>
</tr>
<tr>
<td>II(vii)</td>
<td>


A counter type ADC produces a digital output, which is approximately equal to the analog input by using counter operation internally.
2 marks for diagram., 4 marks for explanation
</td>
<td>2+4=6</td>
<td>6</td>
</tr>
</table>

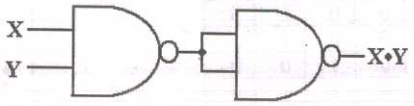
PART C

III(a)

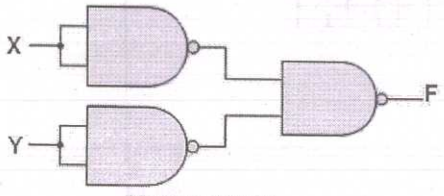
Universal property

a NAND gate with both of its inputs driven by the same signal is equivalent to a **NOT** gate

A NAND gate whose output is complemented is equivalent to an **AND** gate, and a NAND gate with complemented inputs acts as an **OR** gate.



AND using NAND



OR using NAND

3 marks for Universal property
3 Marks each for implementation

3+
2x3=9

9

15

III(b)

$(1010)_b = (1111)_g$

$(1010)_g = (1100)_b$

3 marks each

2x3=6

6

IV(a)

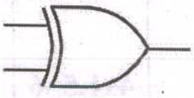
XOR

X	Y	X⊕Y
0	0	0
0	1	1
1	0	1
1	1	0

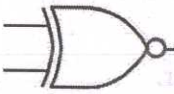
XNOR

X	Y	$\overline{X \oplus Y}$
0	0	1
0	1	0
1	0	0
1	1	1

XOR Symbol



XNOR Symbol



2 marks for symbol
2 marks for truth table
1.5 marks each for explanation

2+2+2x
1.5=7

7

15

IV(b)

$(632)_8 = (19A)_H$
$(E0B3)_H = (57523)_D$
$(32.46)_{10} = (1000000.0111010111)_2$
$(83)_O = (123)_8$

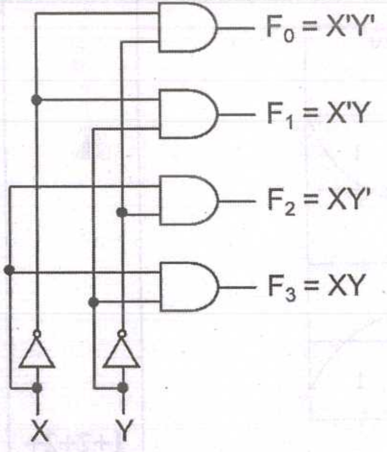
2 marks each

4x2=8

8

V(a)

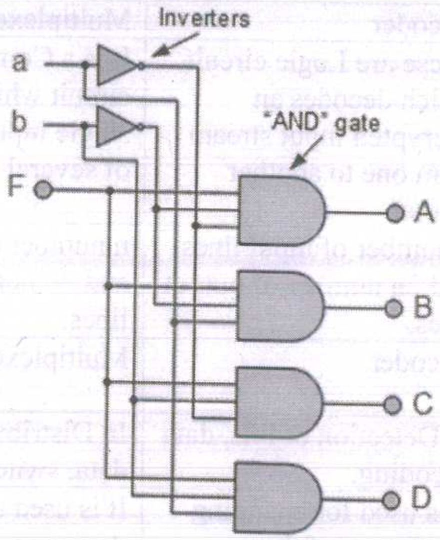
	Criteria	Decoder	Multiplexer
1.	Basic	These are Logic circuit which decodes an encrypted input stream from one to another format.	It is a Combination circuit which routes a single input signal to one of several output signals.
2.	Input/output	n number of input lines and 2n number of output lines.	n number of select lines and 2n number of output lines.
3.	Inverse of	Encoder.	Multiplexer.
4.	Application	In Detection of bits, data encoding.	In Distribution of the data, switching.
5.	Use	It is used for changing the format of the instruction in the machine specific language.	It is used as a routing device to route the data coming from one signal into multiple signals.
6.	Select Lines	Not contains.	Contains.
7.	Implementation	Majorly implemented in the networking application.	Employed in data-intensive applications where data need to be changed into another form.



Decoder

4+4=8	8	15

4 Channel Demultiplexer using Logic Gates



demux

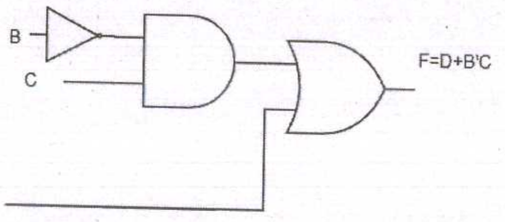
1 marks for each points (any four)(4x1=4)
1 marks each for example (4x1=4)

V(b)

	A	B	C	D	F
0	A'	B'	C'	D'	0
1	A'	B'	C'	D	1
2	A'	B'	C	D'	1
3	A'	B'	C	D	1
4	A'	B	C'	D'	0
5	A'	B	C'	D	1
6	A'	B	C	D'	0
7	A'	B	C	D	1
8	A	B'	C'	D'	0
9	A	B'	C'	D	1
10	A	B'	C	D'	1
11	A	B'	C'	D	1
12	A	B'	C	D'	0
13	A	B	C'	D	1
14	A	B	C	D'	0
15	A	B	C	D	1

CD AB	00	01	11	10
00		1	1	1
01		1	1	
11		1	1	
10		1	1	1

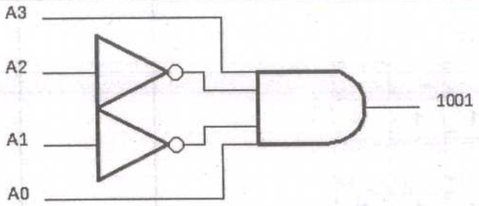
F = D + B′C



1 mark for truth table
2 marks for K map
2 marks for simplified expression
2 marks for circuit diagram

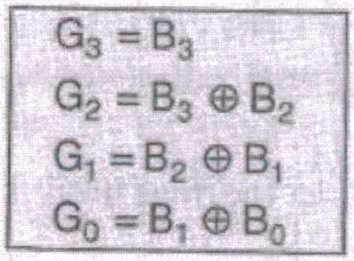
1+2+2+2=7

7

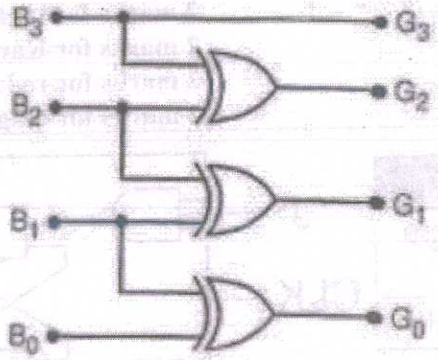
<table>
<tr>
<td>VI(a)</td>
<td>


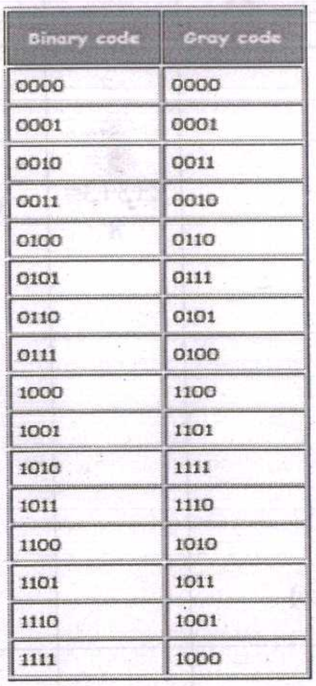
a binary decoder is a combinational logic circuit that converts binary information from the n coded inputs to a maximum of 2^n unique outputs.

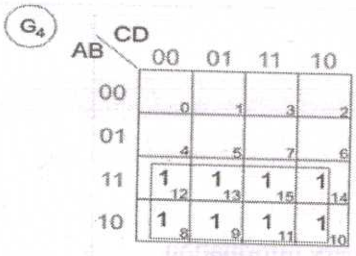
Decoding is necessary in applications such as data multiplexing, 7 segment display and memory address decoding. The example decoder circuit would be an AND gate because the output of an AND gate is "High" (1) only when all its inputs are "High."

2 marks for definition
2 marks for explanation
3 marks for diagram
</td>
<td>2+2+3=
7</td>
<td>7</td>
<td rowspan="2">15</td>
</tr>
<tr>
<td>VI(b)</td>
<td>


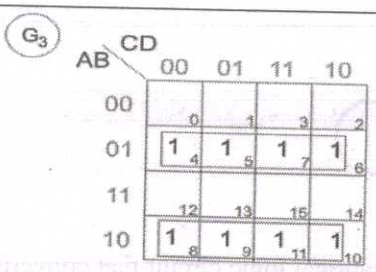
$G_3 = B_3$
$G_2 = B_3 \oplus B_2$
$G_1 = B_2 \oplus B_1$
$G_0 = B_1 \oplus B_0$



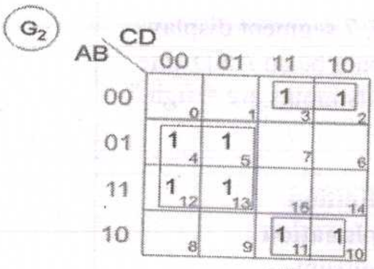

<table>
<tr><th>Binary code</th><th>Gray code</th></tr>
<tr><td>0000</td><td>0000</td></tr>
<tr><td>0001</td><td>0001</td></tr>
<tr><td>0010</td><td>0011</td></tr>
<tr><td>0011</td><td>0010</td></tr>
<tr><td>0100</td><td>0110</td></tr>
<tr><td>0101</td><td>0111</td></tr>
<tr><td>0110</td><td>0101</td></tr>
<tr><td>0111</td><td>0100</td></tr>
<tr><td>1000</td><td>1100</td></tr>
<tr><td>1001</td><td>1101</td></tr>
<tr><td>1010</td><td>1111</td></tr>
<tr><td>1011</td><td>1110</td></tr>
<tr><td>1100</td><td>1010</td></tr>
<tr><td>1101</td><td>1011</td></tr>
<tr><td>1110</td><td>1001</td></tr>
<tr><td>1111</td><td>1000</td></tr>
</table>
</td>
<td>2+2+2+
2=8</td>
<td>8</td>
</tr>
</table>



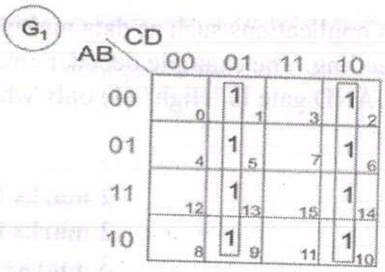
$G_4 = A$



$G_3 = \overline{A}B + A\overline{B} = A \oplus B$



$G_2 = B\overline{C} + \overline{B}C = B \oplus C$

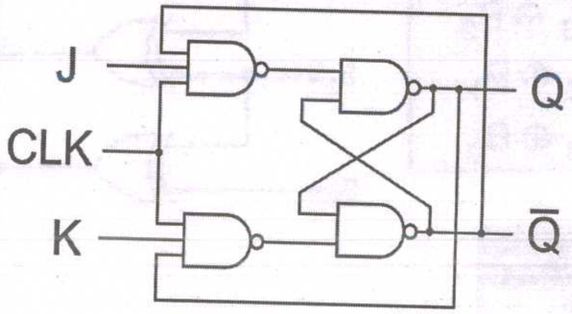


$G_1 = \overline{C}D + C\overline{D} = C \oplus D$

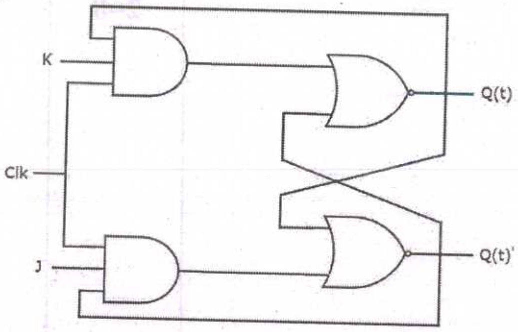
2 marks for truth table
2 marks for Karnaugh map
2 marks for reduced equation
2 marks for diagram

VII(a)

Clk	J	K	Q	Q'	State
1	0	0	Q	Q'	No change in state
1	0	1	0	1	Resets Q to 0
1	1	0	1	0	Sets Q to 1
1	1	1	-	-	Toggles



Or



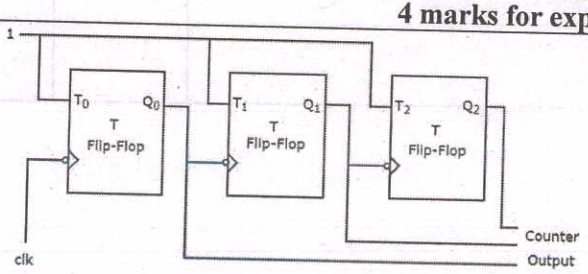
2 Marks for truth table
2 marks for diagram (any one of the above)
4 marks for explanation

2+3+3= 8

8

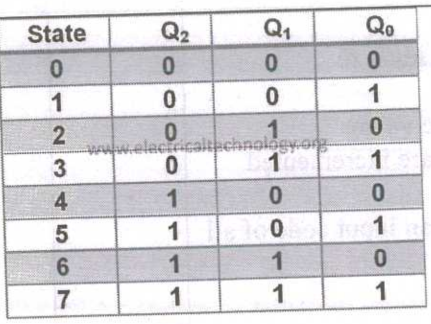
15

VII(b)



2+1+4= 7

7

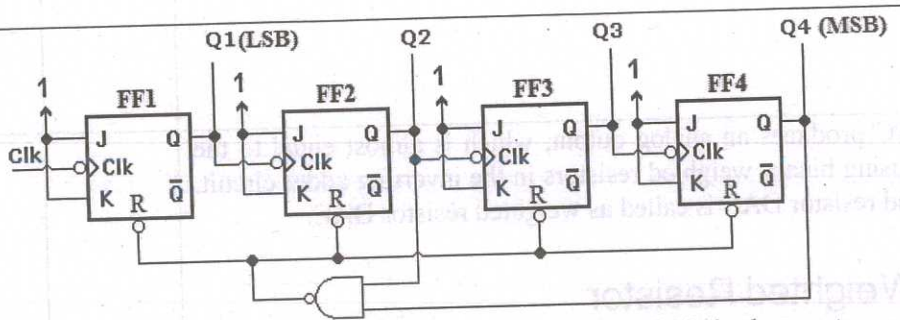


State	Q_2	Q_1	Q_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

The output of first T flip-flop is applied as clock signal for second T flip-flop. So, the output of second T flip-flop toggles for every negative edge of output of first T flip-flop. Similarly, the output of third T flip-flop toggles for every negative edge of output of second T flip-flop, since the output of second T flip-flop acts as the clock signal for third T flip-flop.

2 mark for diagram
1 mark for truth table
4 mark for explanation

VIII(a)



Counters with ten states in their sequence are called mod 10 ripple counter or decade counter. To obtain a truncated sequence, it is necessary to force the counter to recycle before going through all of its normal states. Hence the decade counter must recycle back to the 0000 state after 1001 state.

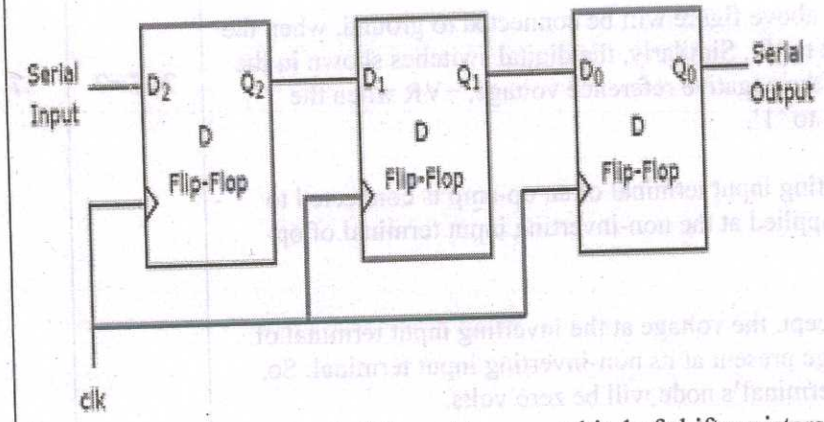
Note that the method of achieving the recycling after the count 1001 is to decade the count 1010 with a NAND gate and connecting the output of the NAND gate to the $\overline{CLR}$ inputs of the flip flops.

2 mark for diagram
6 mark for explanation

2+6=8 8

15

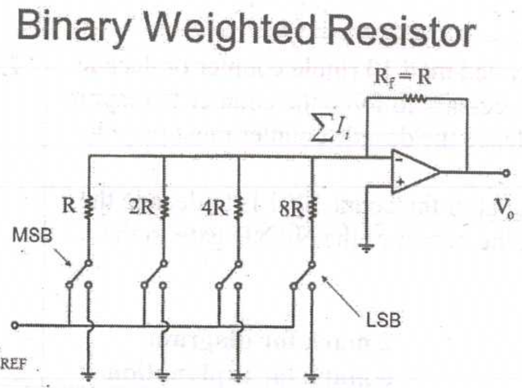
VIII(b)

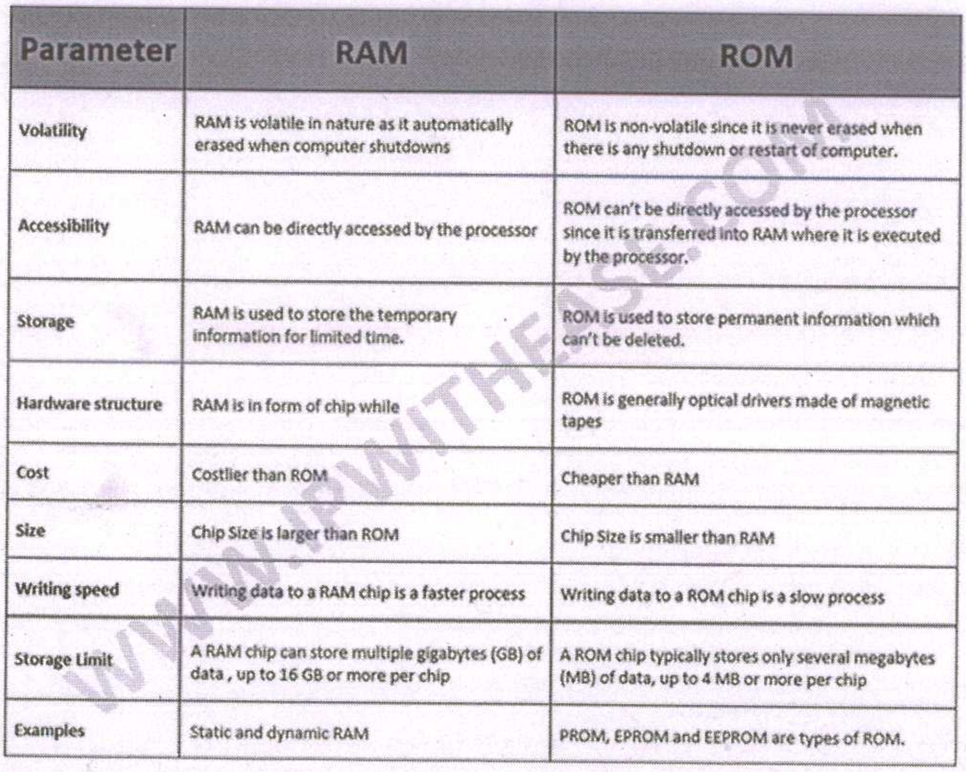


Serial In Serial Out (SISO) shift registers are a kind of shift registers where both data loading as well as data retrieval to/from the shift register occurs in serial-mode.

2 marks for diagram
5 mark for explanation

2+5=7 7

IX(a)	1 **Resolution** - This is the number of bits that the DAC has. 2 **Settling Time** – The time required for the output of the DAC to settle to within+(1/2) LSB of the final value for a given digital input. 3 **Accuracy** – indicates how close the measured value is to the true value. 4 **Monotonicity** – The output value increases as the binary inputs are incremented from 1 value to the next. 5 **Offset Voltage** – or offset error is the analog output response to an input code of all zeros. **2 marks for each point(any four)**	**4x2=8**	**8**	**15**
IX(b)	A **weighted resistor** DAC produces an analog output, which is almost equal to the digital (binary) input by using binary weighted resistors in the inverting adder circuit. In short, a binary weighted resistor DAC is called as weighted resistor DAC.  The digital switches shown in the above figure will be connected to ground, when the corresponding input bits are equal to '0'. Similarly, the digital switches shown in the above figure will be connected to the negative reference voltage, −VR when the corresponding input bits are equal to '1'. In the above circuit, the non-inverting input terminal of an op-amp is connected to ground. That means zero volts is applied at the non-inverting input terminal of op-amp. According to the virtual short concept, the voltage at the inverting input terminal of opamp is same as that of the voltage present at its non-inverting input terminal. So, the voltage at the inverting input terminal's node will be zero volts. We can write the generalized output voltage equation of an N-bit binary weighted resistor DAC as shown below based on the output voltage equation of a 3-bit binary weighted resistor DAC. $=> V_0 = \frac{V_R}{2}\left\{\frac{b_{N-1}}{2^0} + \frac{b_{N-2}}{2^1} + \ldots + \frac{b_0}{2^{N-1}}\right\}$ **2 Marks for diagram**	**2+5=7**	**7**	

	5 marks for explanation			
X(a)	An **analog signal** is any continuously varying signals. Analog signals are real world signals(eg: audio signal, temperature, pressure, velocity etc..) a **digital signal** is a discrete-time signal for which not only the time but also the amplitude has discrete values. processing is much easier and cheaper in the digital systems. **Need for ADC** 1. To sample and digitally analyze signals for computation purposes 2. To digitize speed of video signals for filtering by digital circuits 3. To feed the digital signals to computers etc. **Need for DAC** 1. To display the digital output of a digital system in analog form 2. To reconstruct the analog signal 3. To synthesize the speed, video signals etc. **1 mark each for analog and digital signal definition(2x1=2)** **1.5 marks for each point (any two from each category) 4x1.5=6**	**2+6=8**	**8**	**15**
X(b)	 **1.5 marks for each point (any 5)**	**5x1.5=7**	**7**	

Parameter	RAM	ROM
Volatility	RAM is volatile in nature as it automatically erased when computer shutdowns	ROM is non-volatile since it is never erased when there is any shutdown or restart of computer.
Accessibility	RAM can be directly accessed by the processor	ROM can't be directly accessed by the processor since it is transferred into RAM where it is executed by the processor.
Storage	RAM is used to store the temporary information for limited time.	ROM is used to store permanent information which can't be deleted.
Hardware structure	RAM is in form of chip while	ROM is generally optical drivers made of magnetic tapes
Cost	Costlier than ROM	Cheaper than RAM
Size	Chip Size is larger than ROM	Chip Size is smaller than RAM
Writing speed	Writing data to a RAM chip is a faster process	Writing data to a ROM chip is a slow process
Storage Limit	A RAM chip can store multiple gigabytes (GB) of data , up to 16 GB or more per chip	A ROM chip typically stores only several megabytes (MB) of data, up to 4 MB or more per chip
Examples	Static and dynamic RAM	PROM, EPROM and EEPROM are types of ROM.