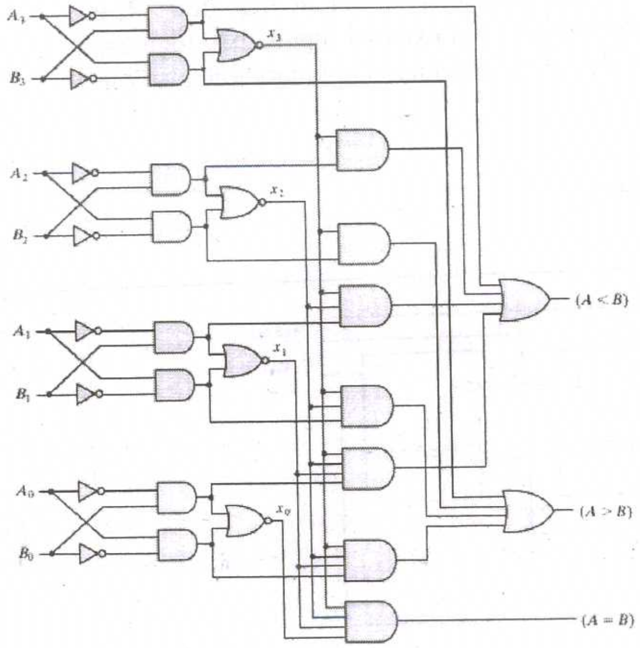
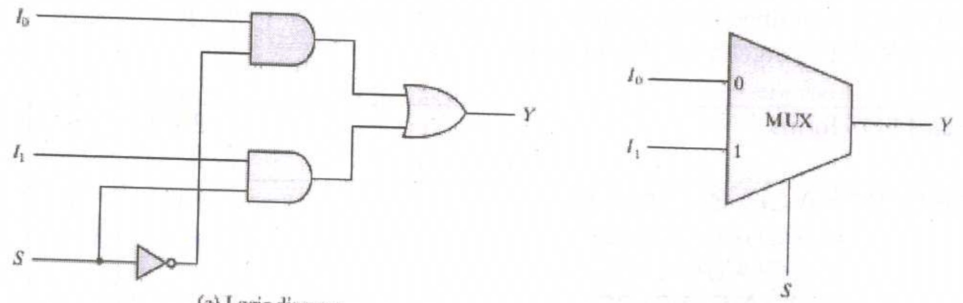
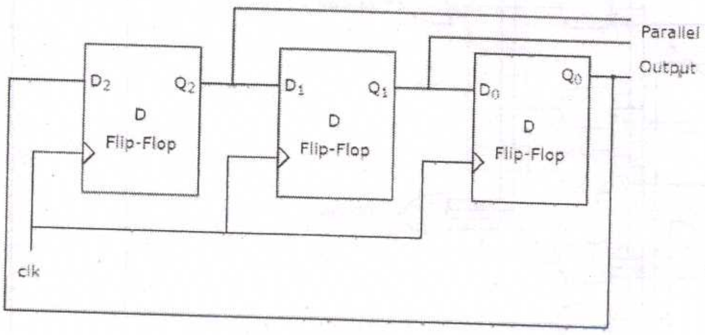
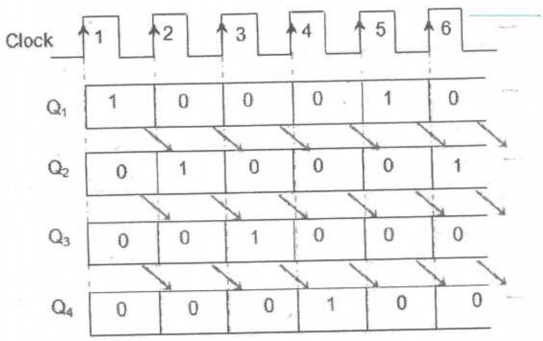
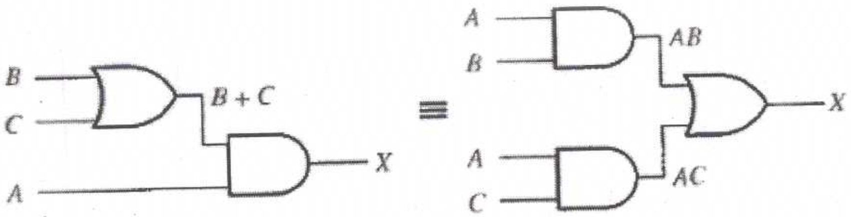
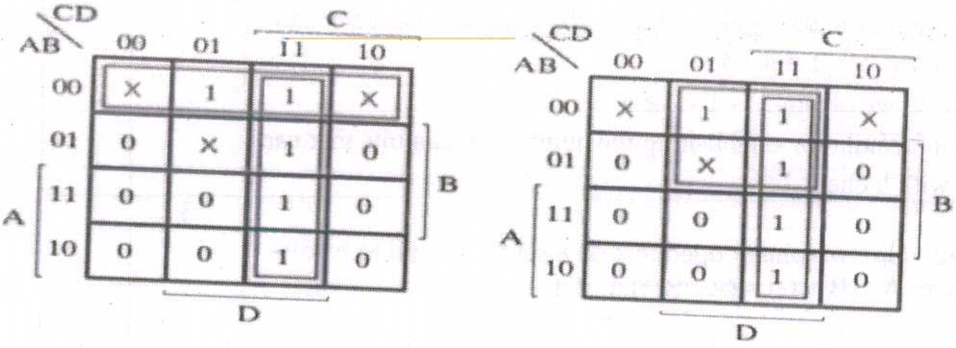
Scoring Indicators Code TED (15) - 3133

QnNo	Scoring Indicators	Split score	Total score
I. 1.	101	1x2	2
2.	An AND term in a standard SOP expression in which all variables are present either in normal form or in complemented form.	1x2	2
3.	Storage elements that operate with signal levels (rather than signal transitions) are referred to as latches.	1x2	2
4.	Resolution of a D/A converter is defined as the smallest change that can occur in the analog output as a result of a change in the digital input..	1x2	2
5.	11000	1x2	2
II. 1.	Standard SOP & standard POS forms	2 x 3	6
2.	$A'C + A'B + AB'C + BC = A'C + A'B + C(AB' + B)$ $= A'C + A'B + C(A+B)$ $= A'C + A'B + AC + BC$ $= (A'+A)C + A'B + BC$ $= C + A'B + BC$ $= C(1+B) + A'B$ $= C + A'B$	6	6
3.	 A_3 B_3 x_3 A_2 B_2 x_2 A_1 B_1 x_1 A_0 B_0 x_0 $(A < B)$ $(A > B)$ $(A = B)$	3 x 2	6

4.	A multiplexer is a combinational circuit that selects binary information from one of many input lines and directs it to a single output line. The selection of a particular input line is controlled by a set of selection lines. Normally, there are 2 n input lines and n selection lines whose bit combinations determine which input is selected. A two-to-one-line multiplexer connects one of two 1-bit sources to a common destination, as shown in Fig (a). The circuit has two data input lines, one output line, and one selection line S . When S=0, the upper AND gate is enabled and I $_0$ has a path to the output. When S = 1, the lower AND gate is enabled and I $_1$ has a path to the output.  (a) Logic diagram (b) Block diagram	Def. 2, Eg. with fig 4 2 + 4	6
5.	A sequential circuit consists of a combinational circuit to which storage elements are connected to form a feedback path. The storage elements are devices capable of storing binary information. The binary information stored in these elements at any given time defines the state of the sequential circuit at that time. The sequential circuit receives binary information from external inputs that, together with the present state of the storage elements, determine the binary value of the outputs. Comparison with combinational circuits	3 3	6
6.	 Parallel Output D2 Q2 D1 Q1 D0 Q0 D Flip-Flop clk	3	6

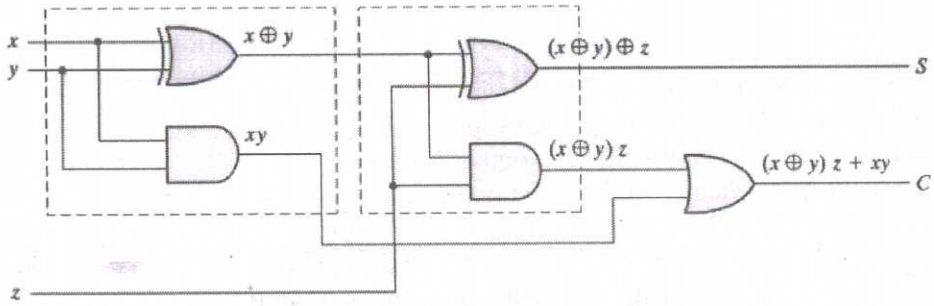
	 Clock 1 2 3 4 5 6 Q1 1 0 0 0 1 0 Q2 0 1 0 0 0 1 Q3 0 0 1 0 0 0 Q4 0 0 0 1 0 0		
7.	In the Hamming code, k parity bits are added to an n bit data word, forming a new word of n + k bits. The bit positions are numbered in sequence from 1 to n + k. Those positions numbered as a power of 2 are reserved for the parity bits. The remaining bits are the data bits. The code can be used with words of any length In general, the Hamming code consists of k check bits and n data bits, for a total of n + k bits. The syndrome value C consists of k bits and has a range of 2 k values between 0 and 2 k - 1. One of these values, usually zero, is used to indicate that no error was detected, leaving 2 k – 1 values to indicate which of the n + k bits was in error. Each of these 2 k - 1 values can be used to uniquely describe a bit in error. Therefore, the range of k must be equal to or greater than n + k, giving the relationship $2^k - 1 \geq n + k$ Solving for n in terms of k, we obtain $2^k - 1 - k \geq n$ This relationship gives a formula for establishing the number of data bits that can be used in conjunction with k check bits.	1x6	6
III. a)	Distributive law: If . and + are two binary operators on a set S, . is said to be distributive over + whenever A . (B + C) = (A . B)+(A . C)  B C B + C A X ≡ A B AB X A C AC X=A(B+C) X=AB+AC	3 5	8
III			

b)	Truth table of expression A'B'C + A'BC + AB'C + ABC	7	7
IV. a)	A + A'B = (A + AB) + A'B (A = A + AB) = (AA + AB) + A'B (A = AA) =AA +AB +AA' +A'B (AA' = 0) = (A + A')(A + B) (Factoring) = 1. (A + B) (A + A' = 1) =A + B	8	8
IV b)	Introduce missing variables in each term. D is absent in the first term, and A in the second term. Final expression is (A' + B + C + D)(A' + B + C + D') (A + B' + C + D')(A' + B' + C + D') (A + B + C' + D')	7	7
V (a)	Functions that have unspecified outputs for some input combinations are called incompletely specified functions . In most applications, we simply don't care what value is assumed by the function for the unspecified minterms. For this reason, unspecified minterms of a function are called don't-care conditions. These don't-care conditions can be used on a map to provide further simplification of the Boolean expression.  CD AB 00 01 11 10 C 00 X 1 1 X 01 0 X 1 0 11 0 0 1 0 10 0 0 1 0 A B D CD AB 00 01 11 10 C 00 X 1 1 X 01 0 X 1 0 11 0 0 1 0 10 0 0 1 0 A B D So, F = A'D+CD	4 6	10
V (b)	Full Adder		

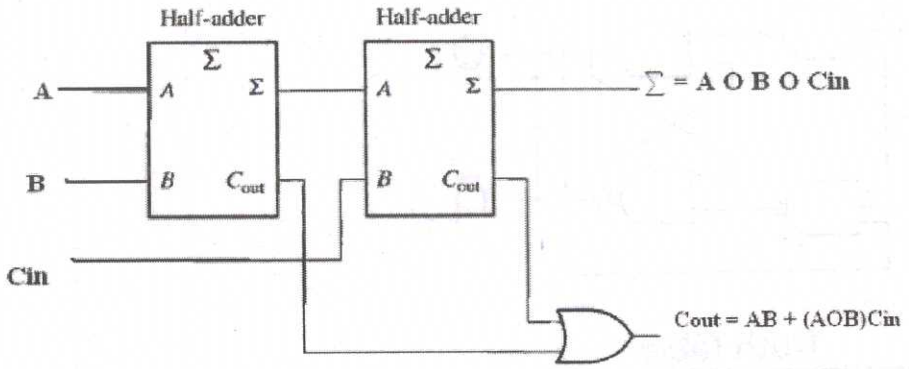
<table>
<tr>
<td></td>
<td>

(Truth Table)

<table>
<tr><th>x</th><th>y</th><th>z</th><th>C</th><th>S</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table>

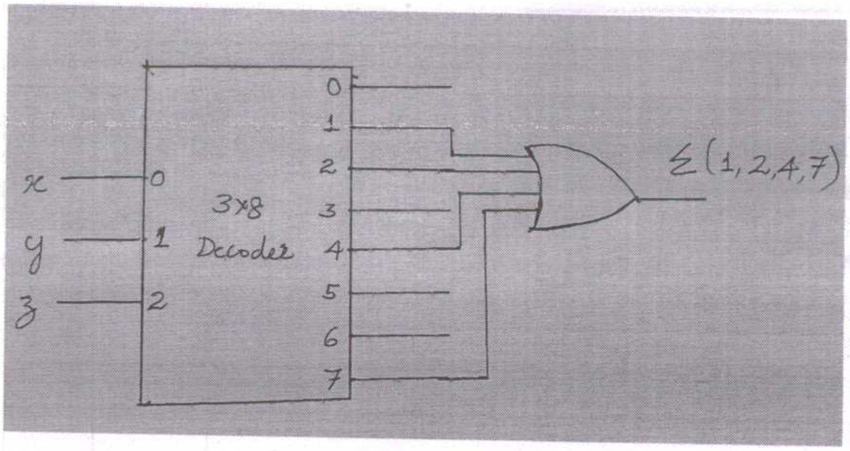
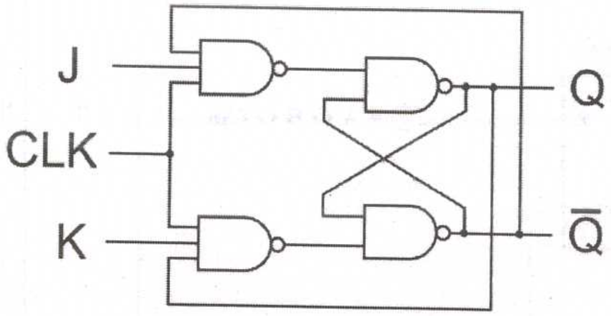


</td>
<td>2
3</td>
<td>5</td>
</tr>
<tr>
<td>VI
a)</td>
<td>

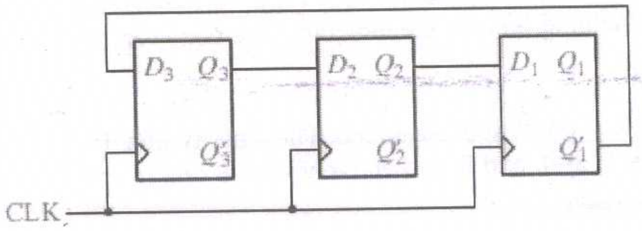


Full adder construction using two half adders

</td>
<td>8</td>
<td>8</td>
</tr>
</table>

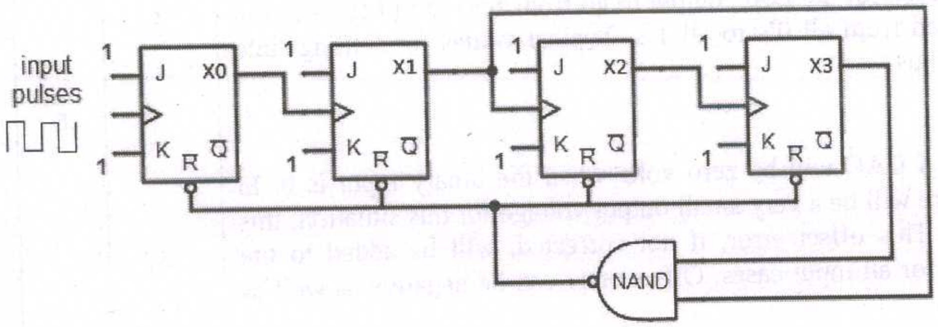
<table>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td>VI.
b)</td><td>
</td><td>7</td><td>7</td></tr>
<tr><td>VII.
a)</td><td>The J-K flip-flop is an extended version of the S-R flip-flop. The J-K flip-flop has three inputs—J, K, and the clock (CK). The J input corresponds to S, and K corresponds to R. That is, if J = 1 and K = 0, the flip-flop output is set to Q = 1 after the active clock edge; and if K = 1 and J = 0, the flip-flop output is reset to Q = 0 after the active edge. Unlike the S-R flip-flop, a 1 input may be applied simultaneously to J and K, in which case the flip-flop changes state after the active clock edge. When J = K = 1, the active edge will cause Q to change from 0 to 1, or from 1 to 0.</td><td>3</td><td>9</td></tr>
<tr><td></td><td>
</td><td>3</td><td></td></tr>
<tr><td></td><td>Truth Table
<table>
<tr><th>J</th><th>K</th><th>CLK</th><th>Q</th></tr>
<tr><td>0</td><td>0</td><td>↑</td><td>Q_0(no change)</td></tr>
<tr><td>1</td><td>0</td><td>↑</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>↑</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>↑</td><td>$\bar{Q}_0$ (toggles)</td></tr>
</table></td><td>3</td><td></td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td>VII.
b)</td><td>



No of positive edge of Clock	Serial Input = Q_0	Q_2(MSB)	Q_1	Q_0(LSB)
0	-	0	0	0
1	1	1	0	0
2	1	1	1	0
3	1	1	1	1
4	0	0	1	1
5	0	0	0	1
6	0	0	0	0

</td><td>3
3</td><td>6</td></tr>
<tr><td>VIII
a)</td><td>



</td><td>9</td><td>9</td></tr>
<tr><td>VIII
(b)</td><td>1) SISO Shift Register acts as a delay element.
2) SIPO Shift register converts serial data into parallel data
3) PISO shift register is used to convert parallel data to serial data.
4) PIPO shift register is used as a temporary storage device and like SISO Shift register it acts as a delay element.</td><td>3 x 2</td><td>6</td></tr>
</table>

IX (a)	One of the most common error correcting codes used in RAMs is Hamming code. In the Hamming code, k parity bits are added to an n bit data word, forming a new word of n + k bits. The bit positions are numbered in sequence from 1 to n + k. Those positions numbered as a power of 2 (1, 2, 4, 8 etc.) are reserved for the parity bits. The remaining bits are the data bits. The code can be used with words of any length.	2	
	For the data word 11000100 (8 bits), 4 parity bits are added. The 4 parity bits, P_1, P_2, P_3, and P_4, are in positions 1, 2, 4, and 8, respectively. The 8 bits of the data word are in the remaining positions. Bit position: 12 11 10 9 8 7 6 5 4 3 2 1 1 1 0 0 P_4 0 1 0 P_3 0 P_2 P_1	3	9
	Each parity bit is calculated as follows: 1) XOR of bits (1, 3, 5, 7, 9, 11) should be 0 for even parity and 1 for odd parity 2) XOR of bits (2, 3, 5, 7, 10, 11) should be 0 for even parity and 1 for odd parity 3) XOR of bits (4, 5, 6, 7, 12) should be 0 for even parity and 1 for odd parity 4) XOR of bits (8, 9, 10, 11, 12) should be 0 for even parity and 1 for odd parity For even parity, these bits will be: $P_1 = 0$, $P_2 = 0$, $P_3 = 1$ and $P_4 = 1$ Substituting the 4 P bits in their proper positions, we obtain the 12 bit composite word.	4	
IX (b)	<u>Settling Time:</u> The operating speed of a DAC is usually specified by giving its settling time, which is the time required for the DAC output to go from zero to full scale as the binary input is changed from all 0's to all 1's. Typical values for settling time range from 50 ns to 10 μs.	3	6
	<u>Offset Error:</u> Ideally, the output of a DAC will be zero volts when the binary input is 0. In practice, however, there will be a very small output voltage for this situation; this is called offset error. This offset error, if not corrected, will be added to the expected DAC output for all input cases. Offset error can be negative as well as positive.	3	

X
(a)

(i) p(A,B,C) = AB’+A’B (ii) q(A,B,C) = AB+BC+AC

PLA Programming Table for the functions:

Product Term		Inputs			Outputs	
					(T)	(C)
		A	B	C	F_1	F_2
AB’	1	1	0	-	1	-
A’B	2	0	1	-	1	-
AB	3	1	1	-	-	1
BC	4	-	1	1	-	1
AC	5	1	-	1	-	1

5

8

Diagram

3

X
(b)

A PLD is an integrated circuit with programmable gates divided into an AND array and an OR array to provide an AND–OR sum of product implementation. PLA (Programmable Logic Array) and PAL (Programmable Array Logic) are the most common PLDs. The most flexible PLD is the PLA, in which both the AND and OR arrays can be programmed. The product terms in the AND array may be shared by any OR gate to provide the required sum of products implementation. The PAL has a programmable AND array and a fixed OR array. The AND gates are programmed to provide the product terms for the Boolean functions, which are logically summed in each OR gate.

3.5

7

A decoding circuit is essential in every memory unit to select the memory word specified by the input address. A memory with 2 k words of n bits per word requires k address lines that go into a k x 2 k decoder. Each one of the decoder outputs selects one word of n bits for reading or writing.

3.5