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

CMRIT
CMR INSTITUTE OF TECHNOLOGY, BENGALURU.
ACCREDITED WITH A+ GRADE BY NAAC

Internal Assessment Test 1 – Sept. 2018

Sub:	Digital Electronics					Sub Code:	17EC34	Branch:	ECE/TCE
Date:	08/09/2018	Duration:	90 mins	Max Marks:	50	Sem / Sec:	III – A, B, C		OBE

Answer any FIVE FULL Questions

		MARKS	CO	RBT
1	Define: i) Literal ii) Canonical SOP iii) Maxterm iv) Essential Prime Implicant v) Prime Implicant Convert the following Boolean function into: X=f (a, b, c) = (a' + b) (b + c') minterm canonical form. Y=f (x, y, z) = x + x' z' (y + z') maxterm canonical form.	[10]	CO1	L2
2	Identify essential prime implicants of following function using k-map. Y= ∑ (1,3,5,7,8,10,12,13,14) + ∑d (4,6,15) Y= π (0,1,4,5,8,9,11) +π d (2, 10)	[10]	CO1	L3
3	Why do we need minimization? State the advantages of minimization. Implement Full adder and Full Subtractor using NAND gates only.	[10]	CO1	L3
4	Find minimal sum for following Boolean function using Quine-McClusky method: f(a,b,c,d) = ∑m(7,9,12,13,14,15) + d.c(4,11)	[10]	CO1	L3

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

CMRIT
CMR INSTITUTE OF TECHNOLOGY, BENGALURU.
ACCREDITED WITH A+ GRADE BY NAAC

Internal Assessment Test 1 – Sept. 2018

Sub:	Digital Electronics					Sub Code:	17EC34	Branch:	ECE/TCE
Date:	08/09/2018	Duration:	90 mins	Max Marks:	50	Sem / Sec:	III – A, B, C		OBE

Answer any FIVE FULL Questions

		MARKS	CO	RBT
1	Define: i) Literal ii) Canonical SOP iii) Maxterm iv) Essential Prime Implicant v) Prime Implicant Convert the following Boolean function into: X=f (a, b, c) = (a' + b)(b + c') minterm canonical form Y=f (x, y, z) = x + x' z' (y + z') maxterm canonical form	[10]	CO1	L2
2	Identify essential prime implicants of following function using k-map. Y= ∑ (1,3,5,7,8,10,12,13,14) + ∑d (4,6,15) Y= π (0,1,4,5,8,9,11) +π d (2, 10)	[10]	CO1	L3
3	Why do we need minimization? State the advantages of minimization. Implement Full adder and Full Subtractor using NAND gates only.	[10]	CO1	L3
4	Find minimal sum for following Boolean function using Quine-McClusky method: f(a,b,c,d) = ∑m(7,9,12,13,14,15) + d.c(4,11)	[10]	CO1	L3

5. Why is Q-M Method preferred to K-Maps? Why is Gray Code used in K-Maps? Implement Full Subtractor using 4x1 Multiplexers.	[10]	CO1	L4
6. Find E.P.I and solve using Quine-McClusky method: $Y(P,Q,R,S) = \pi M\ (0,1,4,5,9,13)$	[10]	CO1	L3
7. Define combinational circuits. State the difference between combinational and sequential circuits with examples. Define multiplexer. Draw the internal diagram of 8:1 mux.	[10]	CO2	L4

5. Why is Q-M Method preferred to K-Maps? Why is Gray Code used in K-Maps? Implement Full Subtractor using 4x1 Multiplexers.	[10]	CO1	L4
6. Find E.P.I and solve using Quine-McClusky method: $Y(P,Q,R,S) = \pi M\ (0,1,4,5,9,13)$	[10]	CO1	L3
7. Define combinational circuits. State the difference between combinational and sequential circuits with examples. Define multiplexer. Draw the internal diagram of 8:1 mux.	[10]	CO2	L4

CMRIT

IAT-1 SOLUTION

Sub:- Digital Electronics

Sub Code:- 17EC 34

Ques 1 - (i) Literal:- A literal is a single variable within a term which may or may not be complemented.

(ii) Canonical SOP:- Boolean functions expressed as sum of product terms such that each product term contains all the variables is termed as canonical SOP form.

(iii) Maxterm:- Maxterm is sum of all literals.

(iv) Essential Prime Implicant:- A prime implicant on a Karnaugh map which covers at least one minterm which is not covered by any other prime implicant.

<V> Prime Implicant :→ Prime Implicant is a product term obtained by combining the maximum possible number of adjacent squares in the map.

$X = f(a,b,c) = (a'+b)(b+c')$ → Convert into minterm Canonical form

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

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

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

$= 100 \quad 101 \quad 001 \quad 101$

$= \pi M(1,4,5)$

$= \Sigma m(0,2,3,6,7)$

$= \bar{a}\bar{b}\bar{c} + \bar{a}b\bar{c} + \bar{a}bc + ab\bar{c} + abc$

~~$Y = f(a,b,c) =$~~

$Y = f(x, y, z) = x + x'z'(y + z')$ - convert into maxterm canonical form.

$= x + x'z'y + x'z'z'$

$= x + x'z'y + x'z'$

$= x \cdot (y + y')(z + z') + x'z'y + x'z'(y + y')$

$= (xy + xy')(z + z') + x'z'y + x'z'y + x'z'y'$

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

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

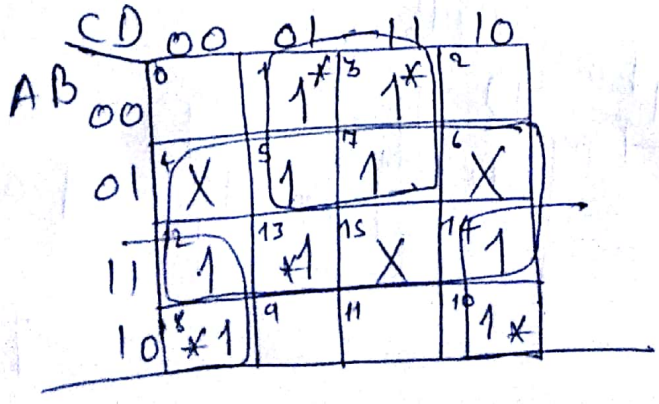
111 110 101 100 010 000

$= \Sigma m(0, 2, 4, 5, 6, 7)$

$= \pi M(1, 3)$

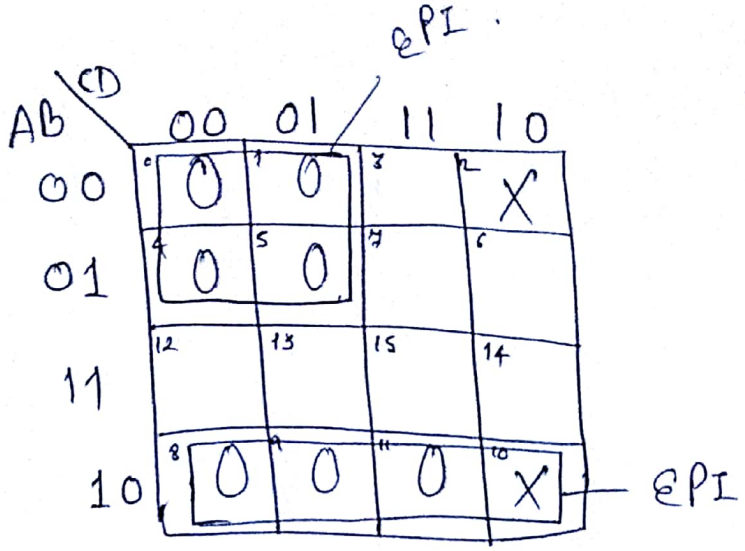
$= (x + y + z')(x + y' + z')$

Ques2 $Y = \Sigma(1, 3, 5, 7, 8, 10, 12, 13, 14) + d(4, 6, 15)$



EPI = B, AD', A'D

Ques2. $Y = \pi(0,1,4,5,8,9,11) + \pi d(2,10)$



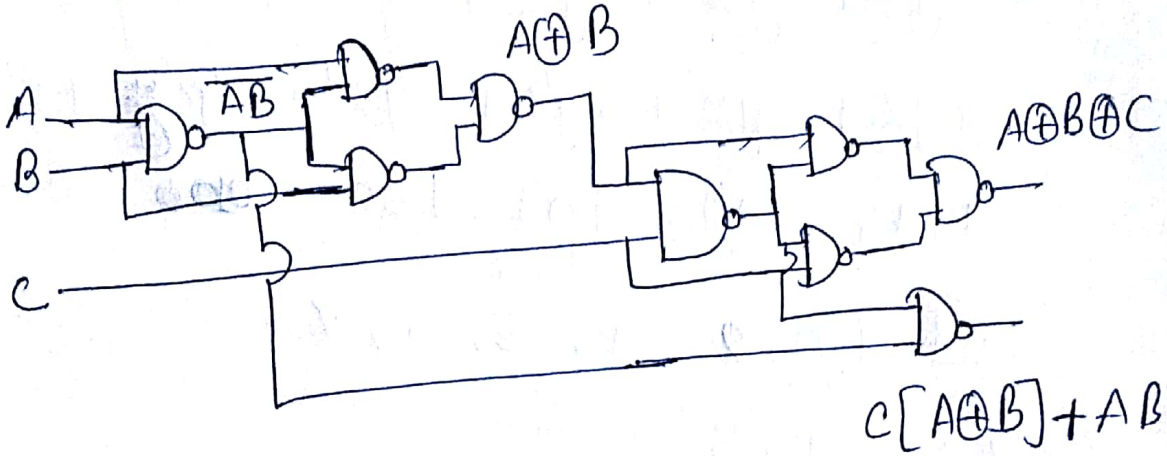
EPI = (A+C), (A'+B)

Ques 5. Why we need minimization.

→ It reduces number of logic gates required.
→ It increases the speed of operation.
→ It decreases power dissipation.
→ Complexity of circuit becomes less.

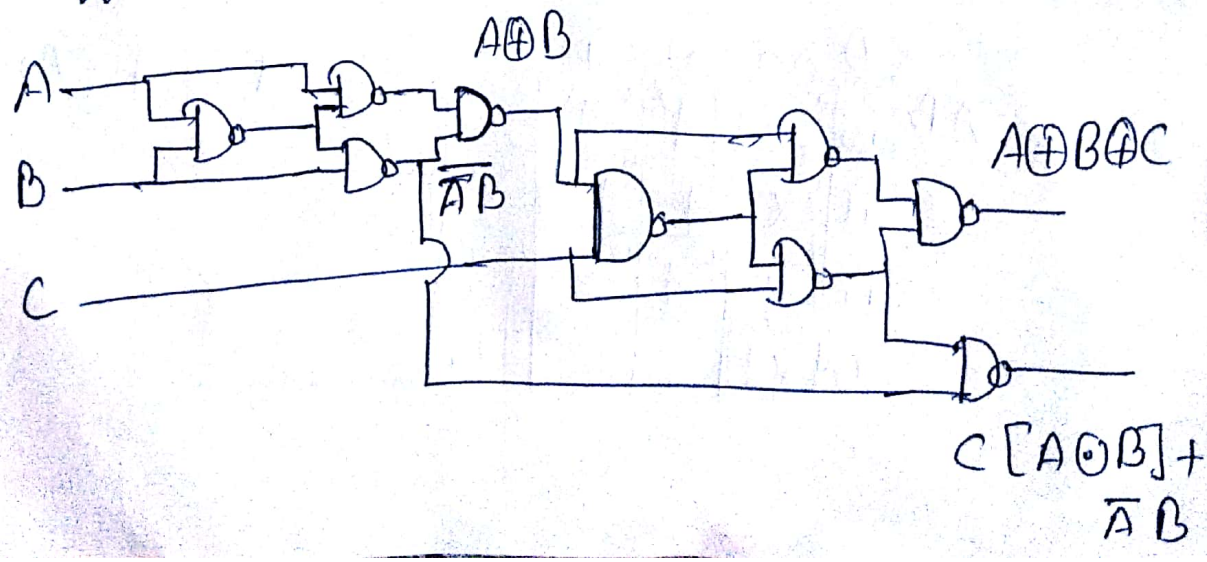
Implementation of full Adder using NAND gates.

Sum = $A \oplus B \oplus C$ Carry = $C[A \oplus B] + AB$



Implementation of full subtracter using NAND gates

Difference = $A \oplus B \oplus C$, Borrow = $C[A \odot B] + \overline{A}B$



Ques 4. $f(a,b,c,d) = \Sigma m(7,9,12,13,14,15) + d(4,11)$

Step 1

	Binary Value
4	0100
7	0111
9	1001
11	1011
12	1100
13	1101
14	1110
15	1111

Step 2

Group	Binary Value
4	0100 ✓
9	1001 ✓
12	1100 ✓
7	0111 ✓
11	1011 ✓
13	1101 ✓
14	1110 ✓
15	1111 ✓

Step 3

Group	Binary Value
4, 12	-100 PI X
9, 11	10-1 ✓
9, 13	1-01 ✓
12, 13	110- ✓
12, 14	11-0 ✓
7, 15	-111 PI X
11, 15	1-11 ✓
13, 15	11-1 ✓
14, 15	111- ✓

Step 4

Group	Binary Value
9,11,13,15	1--1
9,11,13,15	1--1 — AD (PI)
12,13,14,15	11--
12,14,13,15	11-- — AB (PI)

fig 6

PI	7	9	12	13	14	15.
BC'D'			X			
BCD	(X)					X
AD		(X)		X		X
AB			X	X	(X)	X

$f = AB + AD + BCD$

Ques.5 :→ Why is Q-M method preferred to K-maps?

Ans → If the variables are more than five then the simplification using K-map becomes too difficult. Quine - Mc clusky method is based on automatic or computer driven simplification routine, so this method can be programmed in system for minimization of complex function involving large number of variables. K-map is graphical method of pairing minterms hence it is quite tough to program in system as it ~~is~~ depends on visualization

Ques 5- Why is Gray Code used in K-maps?

Ans- Gray Code sequence only changes one binary bit as we go from one number to the next in the sequence, unlike binary. That means that adjacent cells will only vary by one bit, or boolean variable. This is what we need to organize the outputs of a logic function so that we may view commonality.

Ques 6. $Y(P,Q,R,S) = \pi M(0,1,4,5,9,13)$

Step 1	Binary Value
0	0000
1	0001
4	0100
5	0101
9	1001
13	1101

Step 2:-

Groups	Binary Value
0	0000 ✓
1	0001 ✓
4	0100 ✓
5	0101 ✓
9	1001 ✓
13	1101 ✓

Step 3

Group	Binary Value
0, 1	000- ✓
0, 4	0-00 ✓
4, 5	010- ✓
1, 9	-001 ✓
1, 5	0-01 ✓
5, 13	-101 ✓
9, 13	1-01 ✓

Step 4

Group	Binary Value
0, 1, 4, 5	0-0- → 0-0-
0, 4, 1, 5	0-0- → 0-0-
1, 5, 9, 13	--01 → --01
1, 9, 5, 13	--01 → --01

flipkart

PI chart.

	0	1	4	5	9	13
(P+R)	X	X	X	X		
(R+S')		X		X	X	X

EPI → (P+R), (R+S')

final expression

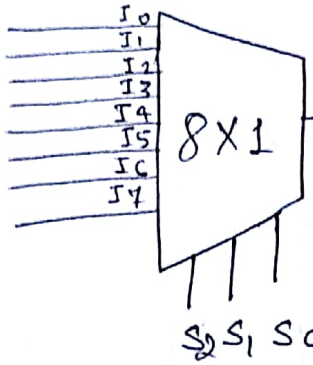
Ques 4 Combination circuits:→ Logic circuits without feedback from output to the input, constructed from a functionally complete gate set, are said to be combinational.

Combinational circuits	Sequential circuits
⇒ Present output depends only on present input.	⇒ Present output depends only on present input and previous output.
⇒ No feedback	⇒ with feedback
⇒ It has no memory block	⇒ It has memory
⇒ Eg :- Adders, Encoders, Decoders, Mux, etc.	⇒ Eg ⇒ flip-flop Register, Counter.

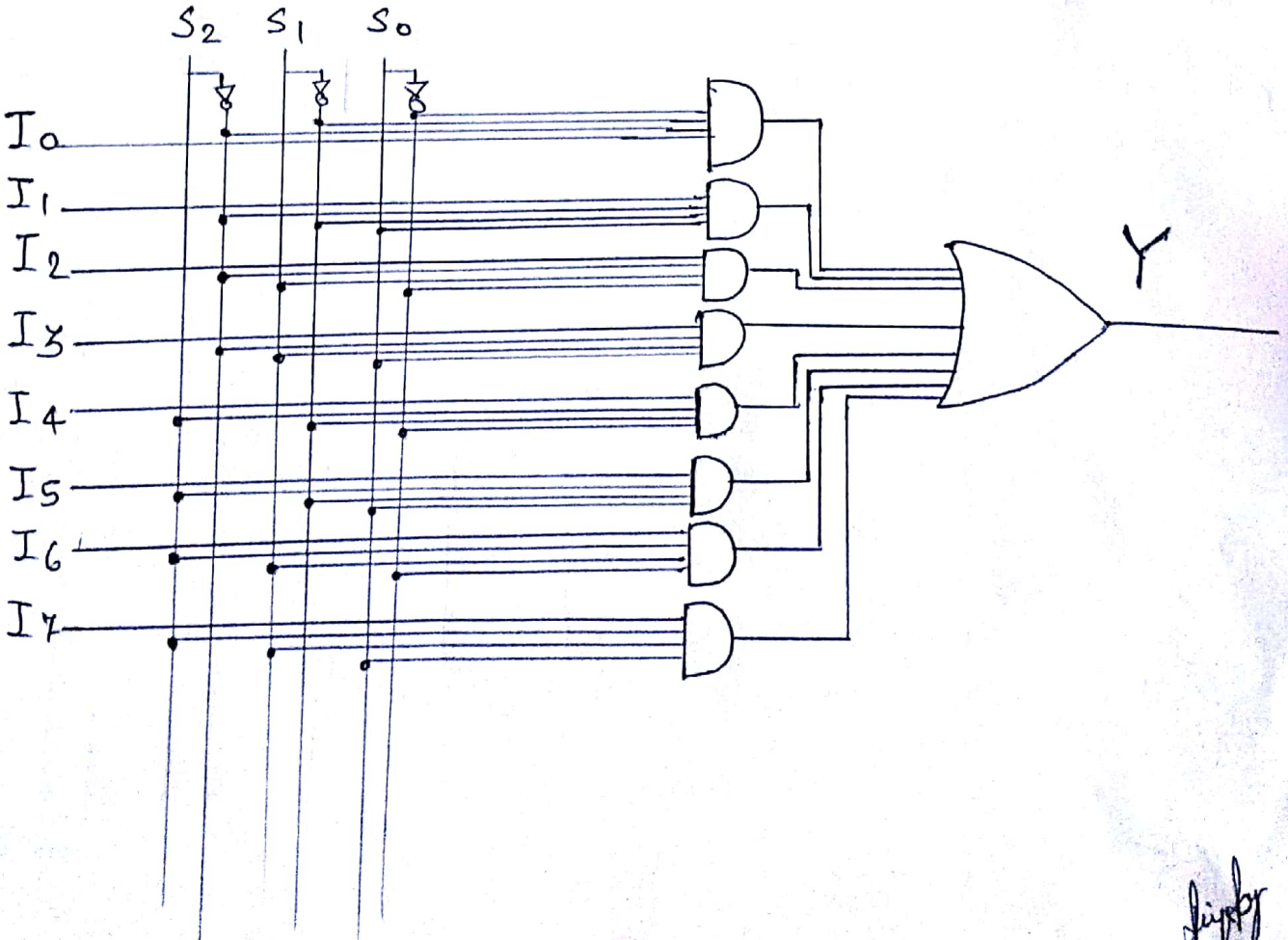
Multiplexer :→ A multiplexer is a combinational circuit that connects one of n inputs to a single output line, so that the logical value of the input is transferred to the output.

The one of n inputs selection is determined by m' select inputs, where $n = 2^m$.

8X1 Mux Internal Diagram

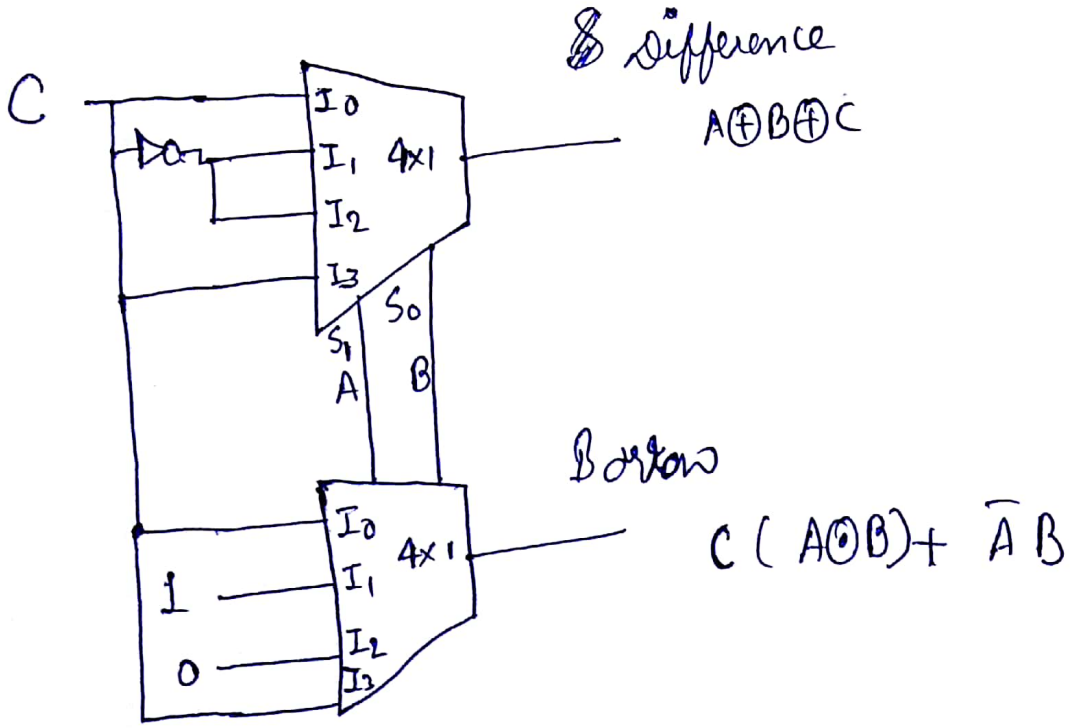


$Y = \bar{S}_2\bar{S}_1\bar{S}_0 I_0 + \bar{S}_2\bar{S}_1 S_0 I_1 + \bar{S}_2 S_1 \bar{S}_0 I_2 + \bar{S}_2 S_1 S_0 I_3 + S_2\bar{S}_1\bar{S}_0 I_4 + S_2\bar{S}_1 S_0 I_5 + S_2 S_1 \bar{S}_0 I_6 + S_2 S_1 S_0 I_7$



Q5. Implement full Subtractor using 4x1 Mux.

	A	B	C	Diff.	Borrow
I_0	0	0	0	0 → C	0 → C
	0	0	1	1	1
I_1	0	1	0	1 → $\bar{C}$	1 → 1
	0	1	1	0	1
I_2	1	0	0	1 → $\bar{C}$	0 → 0
	1	0	1	0	0
I_3	1	1	0	0 → C	0 → C
	1	1	1	1	1



Difference

$A \oplus B \oplus C$

Borrow

$C(A \odot B) + \bar{A}B$