VLSI ASSIGNMENT

NAME : SIVARANJANI GANESH

ROLL NO: 2015105024

CLASS : BE (ECE)

SUBJECT : VLSI DESIGN

TOPIC : CHAPTER 6 - 1, 2, 3a, 4 and attachment

SUBMISSION DATE : 12/04/2020

(1) Implement the equation $X = ((\bar{A}+\bar{B})(\bar{C}+\bar{D}+\bar{E}+\bar{F})\bar{G}$ using Complementary CMOS. Size the devices so that output resistance is the same as that of an inverter with an NMOS $W/L=2$ and PMOS $W/L=6$. Which input pattern would give the worst and best equivalent pull-up or pull-down resistance?

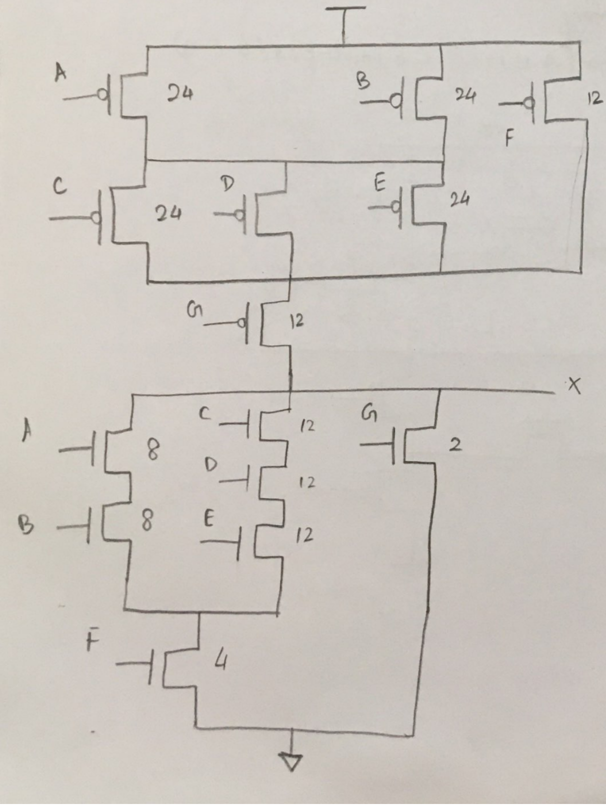
Sol^n :

Rewriting the output expression:

$$x = ((\bar{A}+\bar{B})(\bar{C}+\bar{D}+\bar{E})+\bar{F})\bar{G}$$

$$= ((\overline{AB} + \overline{CDE)\,F)}+G$$

The pullup network is dual of the pull down network.



The Worst Case pull-up resistance occurs whenever a single path exists from the output node to Vdd. Examples of vectors for the worst cases are ABCDEFG = 1111100 and 0101110. The best case pull-up resistance occurs when ABCDEFG = 0000000.

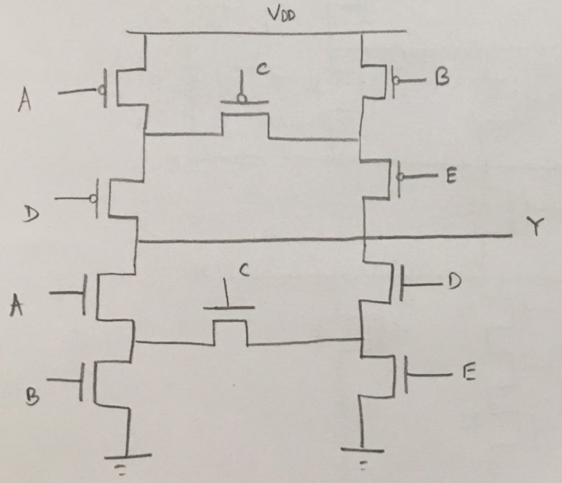
The Worst case pull down resistance occurs whenever a single path exists from the output node to GND. Examples of vectors for the worst case are ABCDEFG = 0000001 and 0011110.

The Best case pull-down resistance occurs with ABCDEFG = 1111111.

(2) Implement the following expression in a full static CMOS logic using no more than 10 transistors.

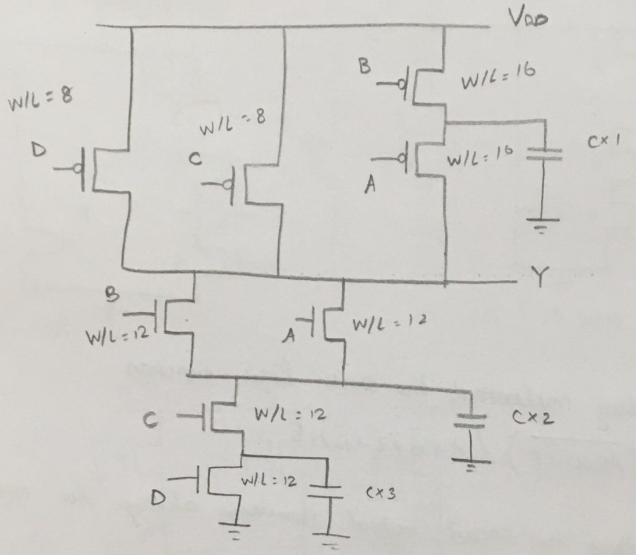
$$Y = (A \cdot B) + (A \cdot C \cdot E) + (D \cdot E) + (D \cdot C \cdot B)$$

50%

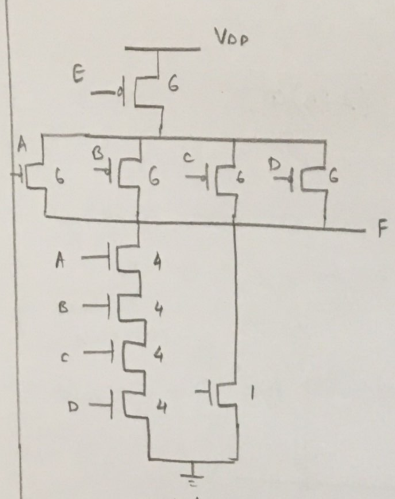


30) What is the logic function implemented by the CMOS transistor n/w? Size the NMOS and PMOS devices so that the output resistance is the same as that of an inverter with an NMOS W/L = 4 and PMOS W/L = 8

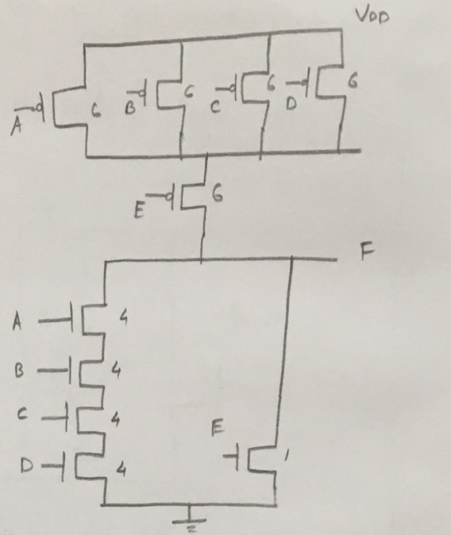
Sol^n: The logic function is $Y = \overline{(A+B)CD}$.



4a) Do the following circuits implement the same logic functions? If yes what is the logic function? If no, give Boolean expression for both circuits.



circuit A



Sol^n: Yes, they implement the same logic function

$F = (\overline{ABCD + E}) = (\bar{A} + \bar{B} + \bar{C} + \bar{D}) . \bar{E}$

(b) Will these two circuit output resistance always be equal to each other?

NO

(c) Will these two circuits rise and fall times always be equal to each other? Why or why not?

No. circuit B appears optimized for the case where the transistor with input E is on critical path. ∴ if input E arrives later circuit B will be faster than circuit A since the internal node will already be charged and only the output capacitance needs to be switched. Consider an input combination E, A, B, C, D are all low. circuit A has only one body affected device while circuit B has four. since the associated rise in V_t and fall in output resistance affects only one resistor in circuit A, but four parallel resistors in circuit B, we expect a difference in the timing waveform.

5a) $t_{setup} = 3 * t_{pd-inv} + t_{pd-tx}$

where t_{pd-inv} ⇒ propogation delay of each inverter

t_{pd-tx} ⇒ propogation delay of transmision gate

$t_{cq} = t_{pd-inv} + t_{pd-tx}$

$t_{hold} = 0$

2) Dynamic edge triggured register

$t_{su} = t_{pd-inv} + t_{pd-tx}$

$t_{hold} = 0$

$t_{cq} = 2 * t_{pd-inv} + t_{pd-tx}$

Positive edge triggered register

$t_{su} = t_{pd-tx}$, $t_{pd} = 2 * t_{pd-inv} + t_{pd-tx}$; $t_{hold} = 0$

Static register with reduced clock load :

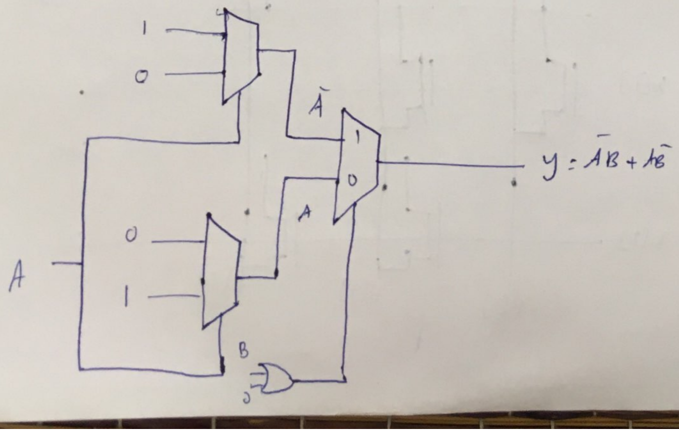
$t_{su} = t_{pd-tx}$

$t_{pd} = t_{pd-tx} + 2 \times t_{pd-inv}$

$t_{hold} = 0$

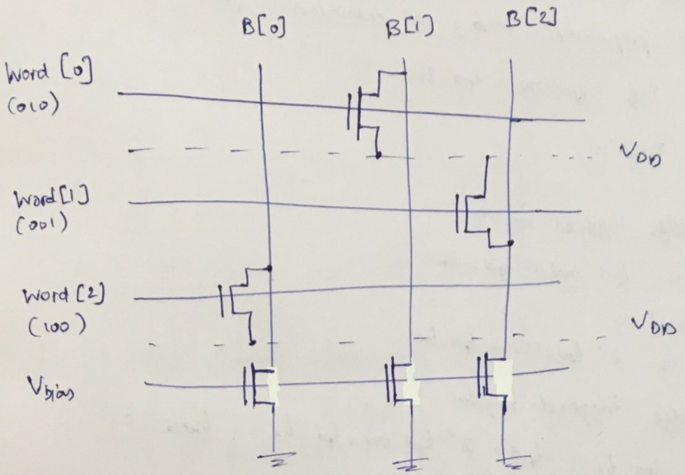
3) 2 input XOR using ACT 1 logic

$y = A\bar{B} + \bar{A}B$

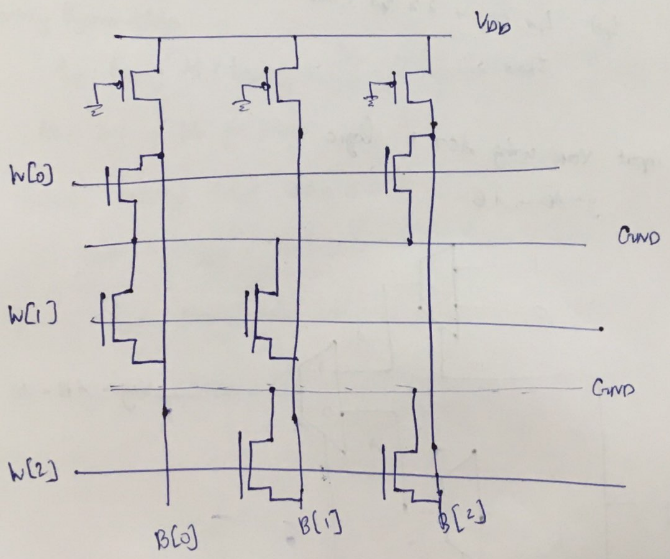


3x3 ROM for 010, 001, 100

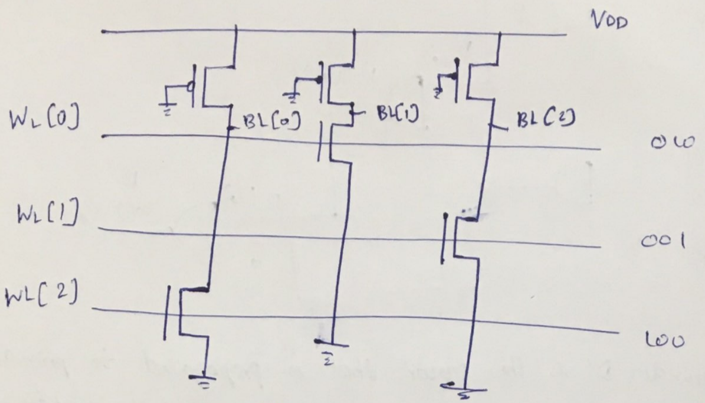
(1) MOS OR ROM



2) MOS-NOR



MOS- NAND ROM



7) Four bit Ripple carry adder

$t_{pd} \Rightarrow (N-1)\, t_{carry} + t_{sum}$

N - No of i/p bits

(ii) Manchester carry chain adder:

$$t_p = 0.69 \sum_{i=1}^{N} C_i \left(\sum_{j=1}^{i} R_j \right) = 0.69 \left(\frac{N(N+1)}{2} \right) RC$$

(iii) Carry Bypass adder

$$t_p = t_{su} + M \times t_{carry} + \left(\frac{N}{M} - 1 \right) t_{bypass} + (M-1)\, t_{carry} + t_{sum}$$

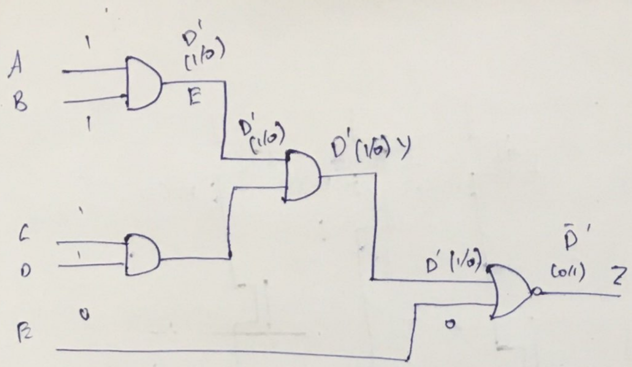
M - No. of bits per block

(iv) Linear carry select adder:

$$t_{add} = t_{setup} + M \times t_{carry} + \left(\frac{N}{M} \right) \times t_{mux} + t_{sum}$$

(v) Square Root carry select adder:

$$t_{add} = t_{setup} + M^{*} t_{carry} + \left(\sqrt{2N} \right)^{*} t_{MUX} + t_{sum}$$



⇒ Consider D' is the error that is propogated to primary output

→ Activate the s-to fault at node E and to justify errors Substitute the fault free value and work backward from the fault origin to primary inputs (PI's) and the input values (A and B).

⇒ Work forward from fault to Po, setting input to propogate D' towards Z.

⇒ work backwards to justify C and D and again work forward from y and set input 'E' case that fault is propogated to (Po) Z.

⇒ A=1; B=1, C=1, D=1, E=0