Book : Digital System Design (Using VHDL)
By Charles H. Roth
Lizy Kurian John.

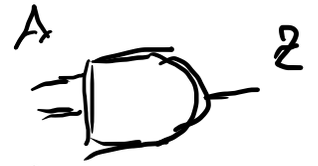
Weightage : 50% (Theory) + 50% (Labs)

Theory: 40% Mid Sem, 60% End Sem.

Labs: When you come back to the Insti. (ASAP)

Course Pre-Requisites : Assuming BEC + DLD + CO/CA

BASIC:

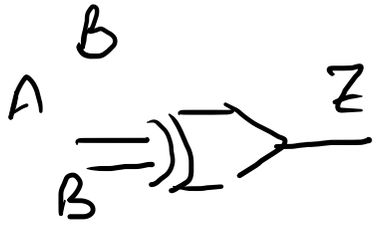


$Z = A \cdot B$ = A and B.

OR = $A \cap B$ = A OR B = A + B

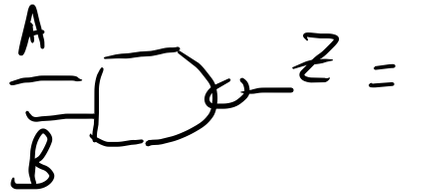
$Z = \overline{A}$ = !A = A'

= NOT (A).

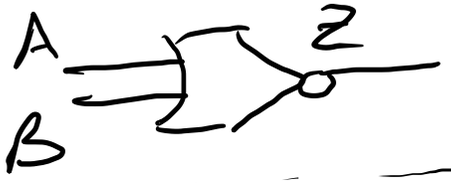


$Z = A \oplus B$ = A XOR B

Universal Gates



Z = A NAND B = $(A \cdot B)'$



Z = A NOR B = $(A + B)'$

$Z = \overline{A}$

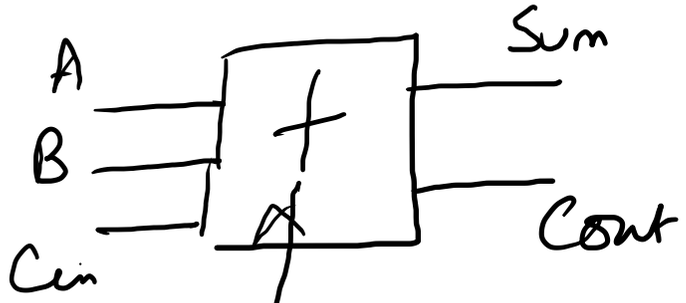
MUX, EXOR, DFF, Latches, memories, Adders, Controllers

↓

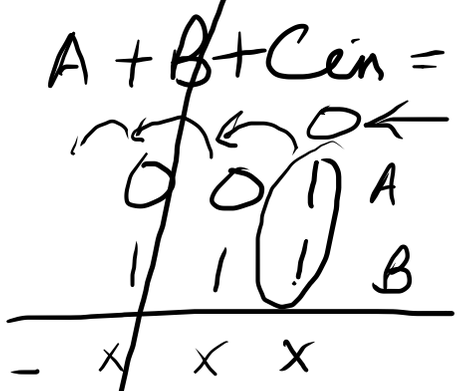
MPs, ...

Karnaugh Map :-

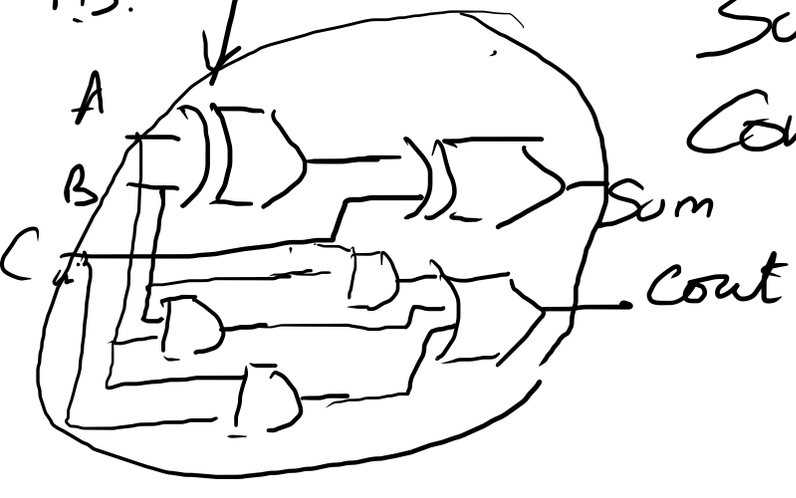
Full Adder



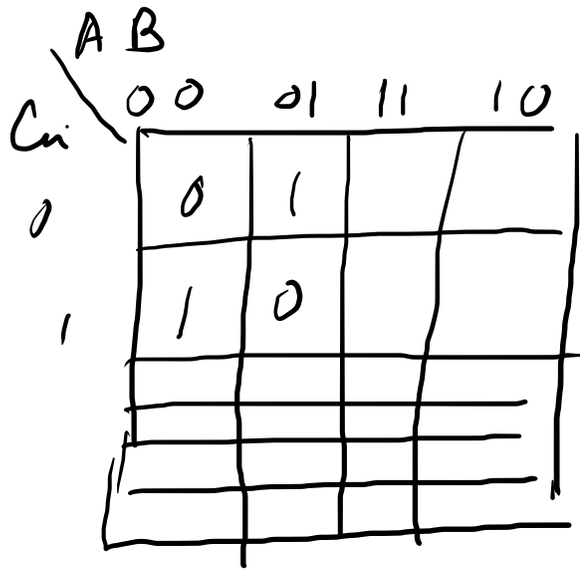
$A + B + C_{in} = Sum + Cout$.



P/s. do not work



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

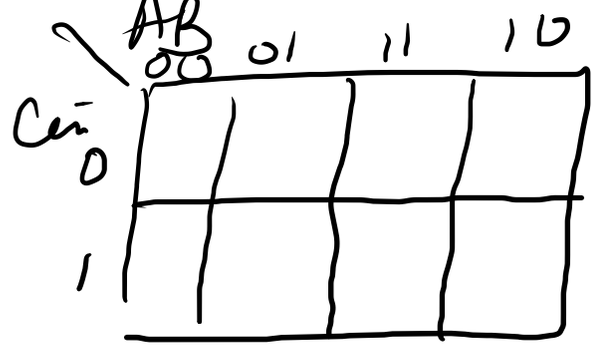


$Sum = A'B'C_{in} + A'BC_{in}' + AB'C_{in}' + ABC_{in}$

$= A \oplus B \oplus C$

$Cout = A'BC_{in} + AB'C_{in} + ABC_{in}' + ABC_{in}$

$= AB + AC_{in} + BC_{in}$



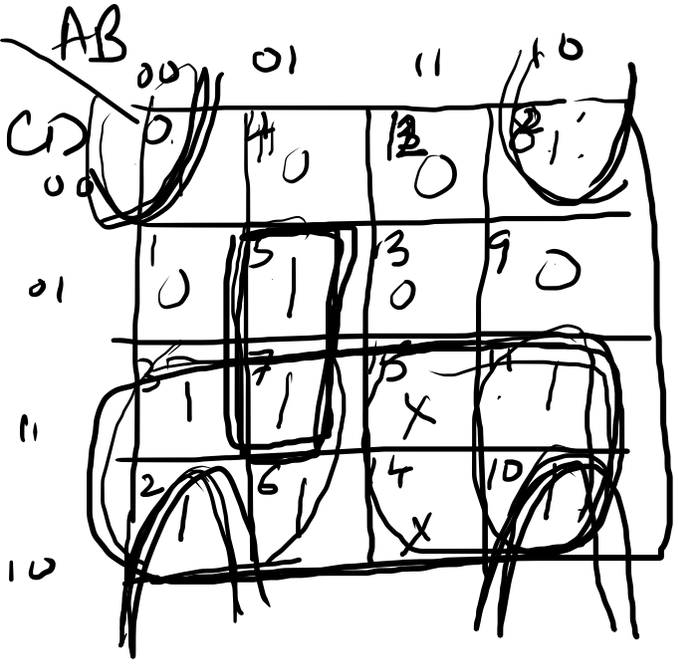
Cout =

4 - variable K-map:

$$f = \sum m(0,2,3,5,6,7,8,10,11) + \sum d(14,15)$$

$$= C + B'D' + A'BD \quad \leftarrow \text{SOP (Sum of product)}$$

(Product)

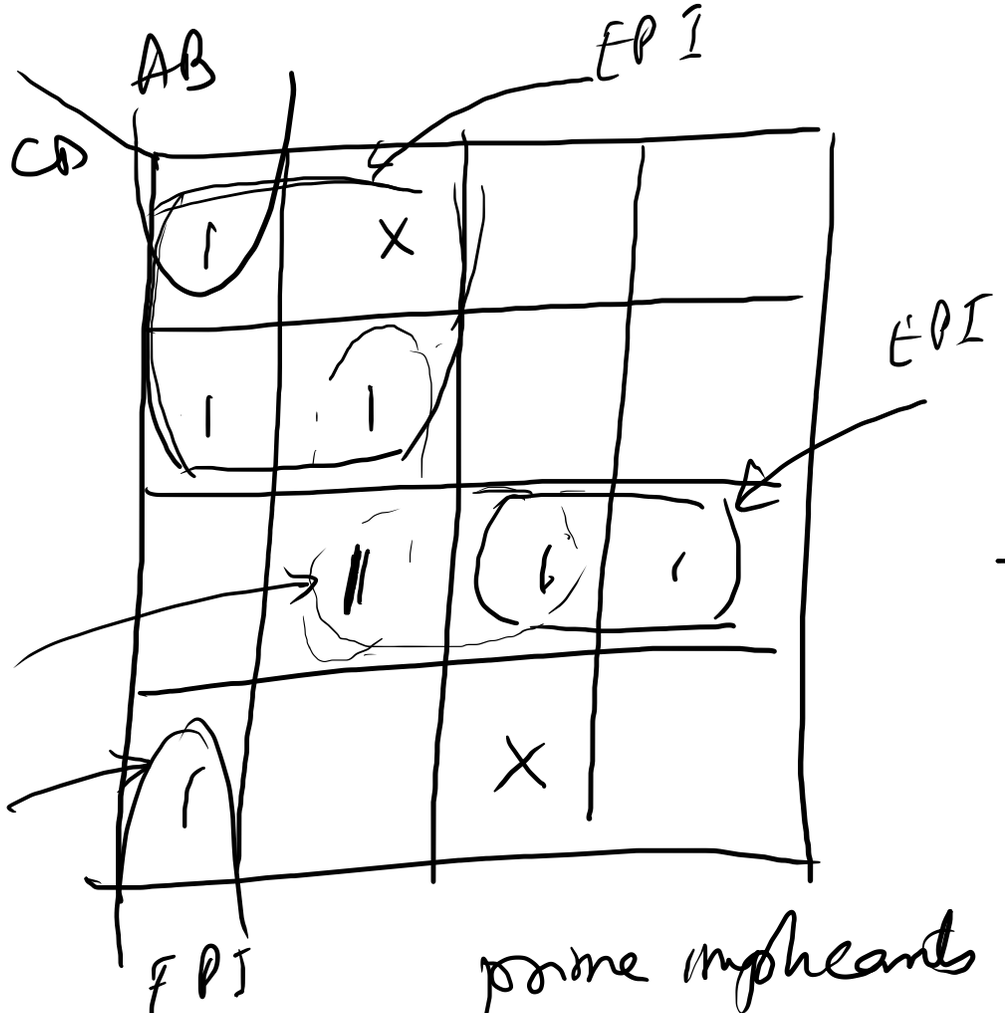


$F' = BC'D' + B'C'D + AB$ POS

$= (B'+C+D).(B+C+D').(A'+B')$

Sum

Product of sums.



A sum of one, two, four or eight adjacent 1's on a map represents a prime-implicant if it cannot be combined with another group of 1's to eliminate a variable.

If is essential if it contains a 1 that is not contained in any other prime implicant.

prime implicants : $A'C'$, ACD, $A'B'D'$, $A'BD$, BCD

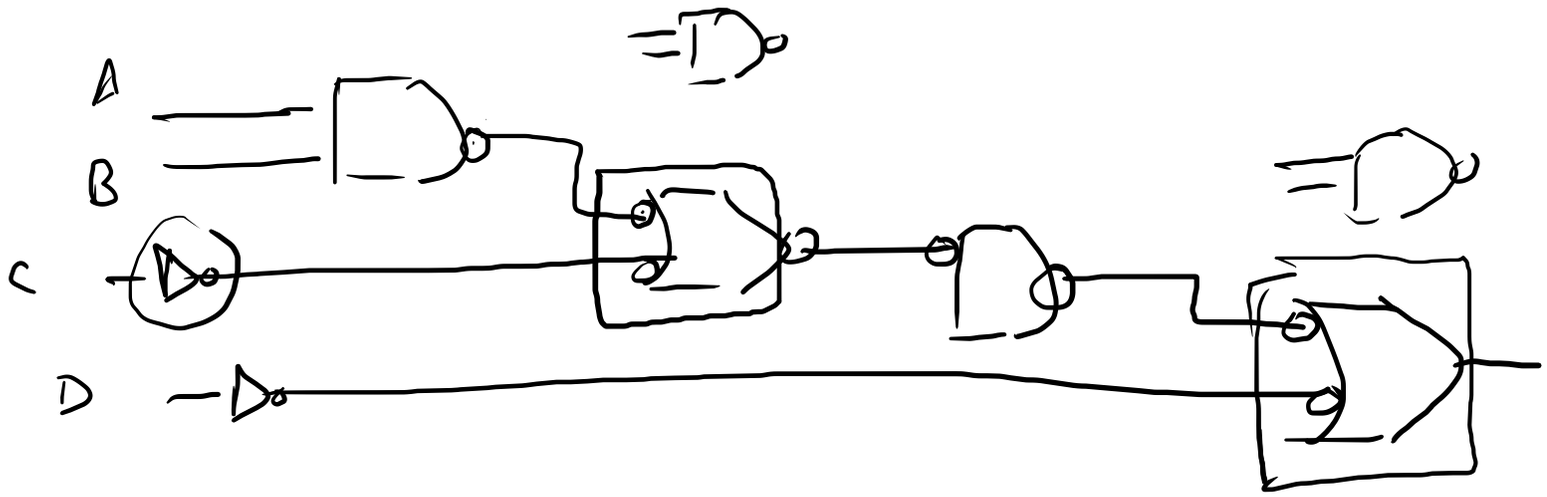
EPI : $A'C'$, ACD, $A'B'D'$

$$F = A'C' + ACD + A'B'D' + A'BD$$

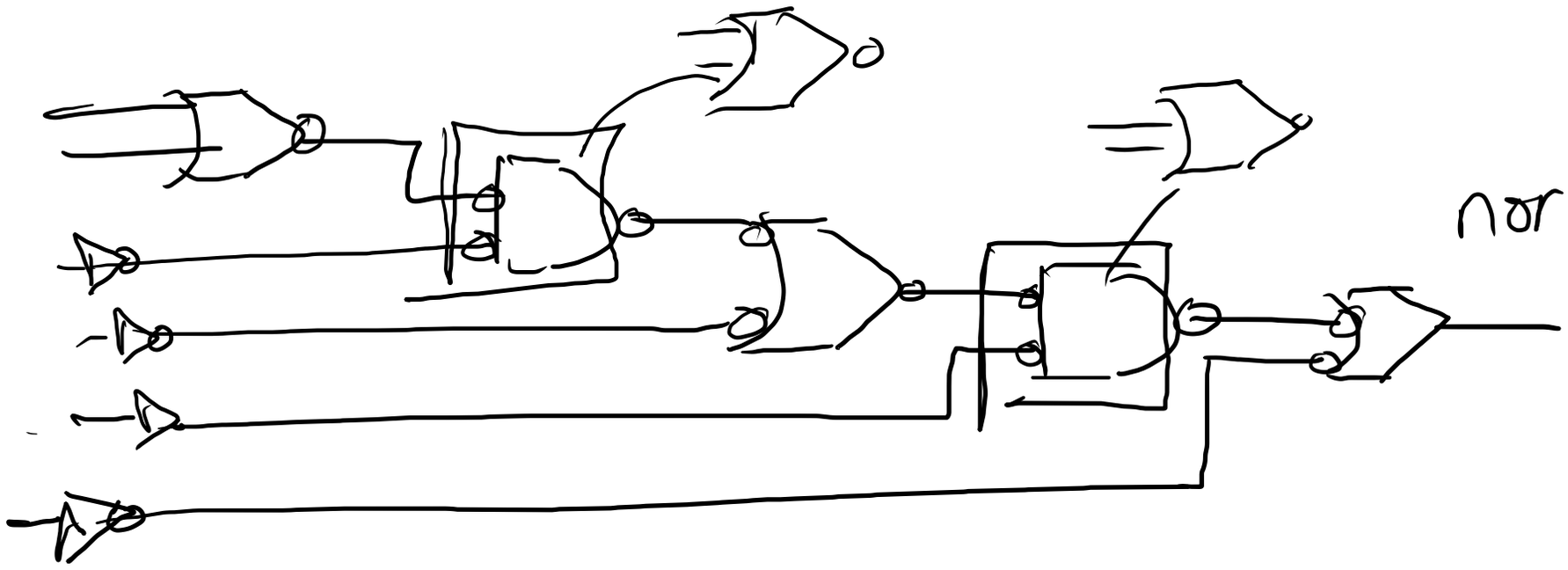
or

$$A'C' + ACD + A'B'D' + BCD$$

$(A+B)' = A' \cdot B'$

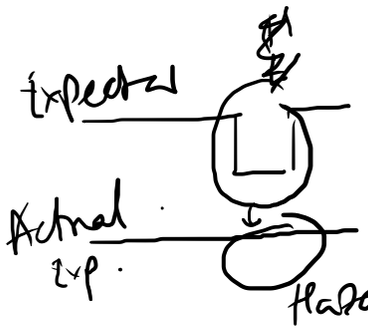


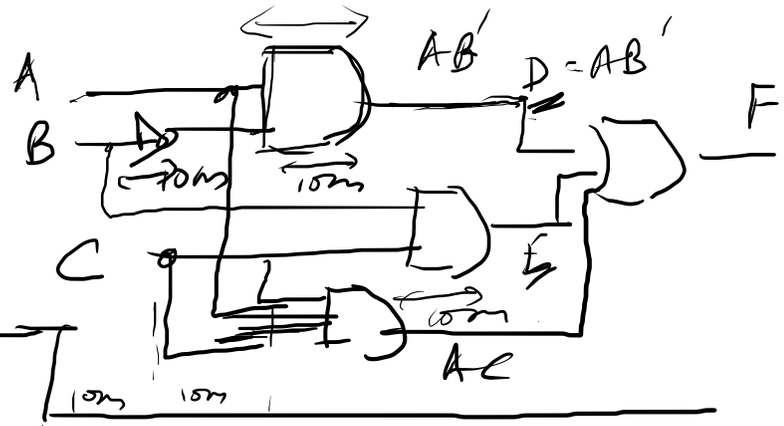
nand gate realization



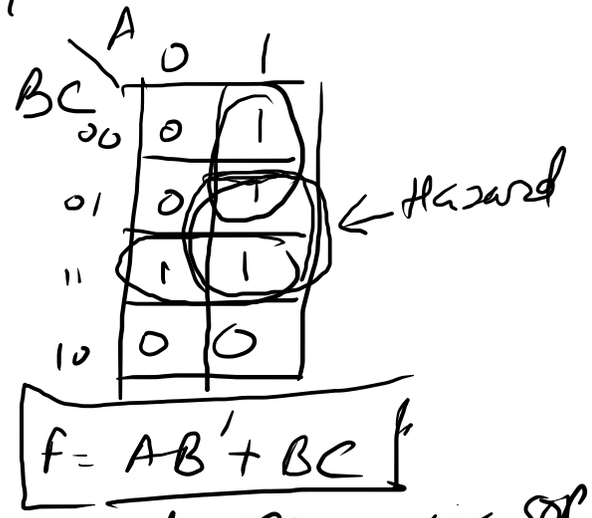
nor gate realization

Hazards: In response to an input change and for some combination of propagation delays a ckt o/p may momentarily go to 0'(1) when it should stay at 1·(0) (State '1') / (State '0')

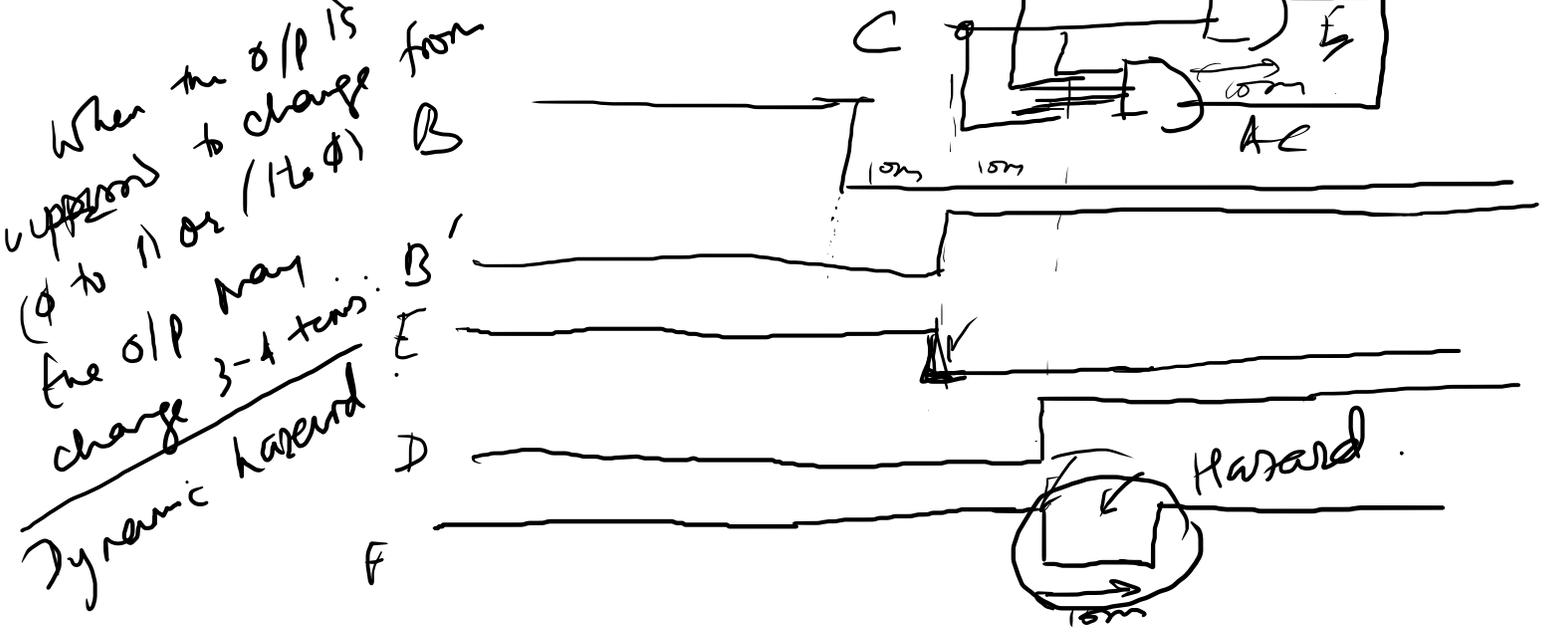




Assume
A = C = 1



When the o/p is supposed to change from (0 to 1 or (1 to 0)

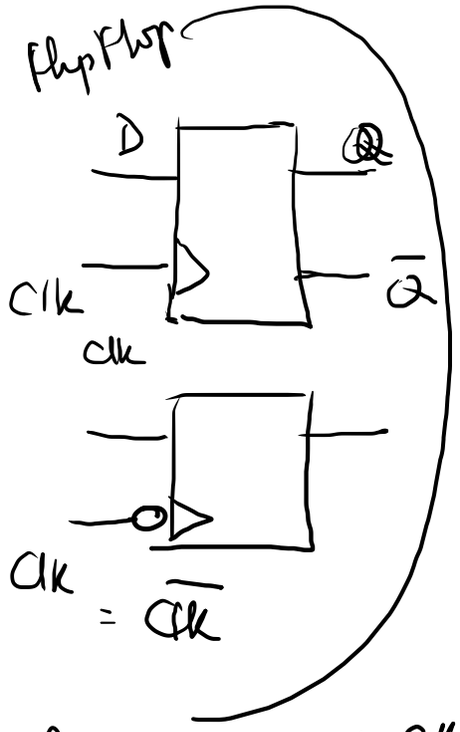


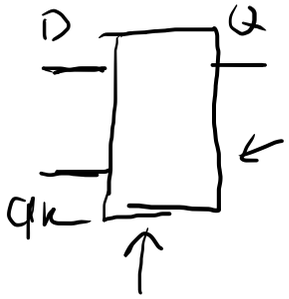
the o/p may change 3-4 times

Dynamic hazard

Static '1' hazard occurs in SOP implementation when 2 minterms differing by only 1 i/p variable are not covered by same product term. F = AB' + BC + AC

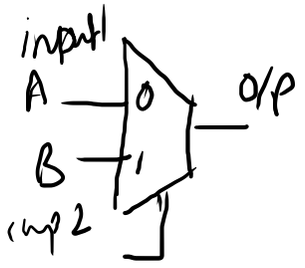
Latch.





?

Latch.



JK ← Next state Q = C.S.

$Q^+ = JQ' + K'Q$

(Next) State Equations.

D flipflop – o/p follows the i/p.

Q^+ o/p = 0 at Clk ↓

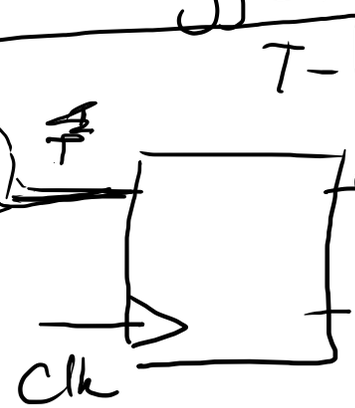
D	Q	Q^+
0	0	0
0	1	0
1	0	1
1	1	1

Next state

$Q^+ = D$

Toggle Flip Flop:

T-FlipFlop.

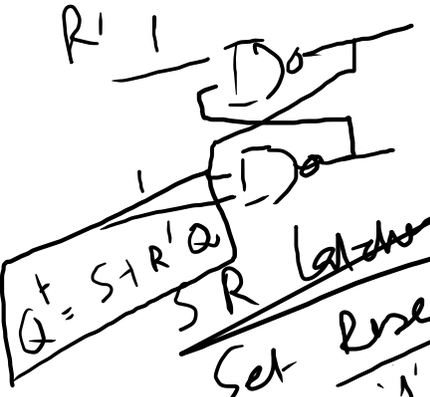


Q	T	Q	Q^+
0	0		0
0	1		1
1	0		1
1	1		0

$Q^+ = Q$

Toggles

$Q^+ = QT' + \bar{Q}T = Q \oplus T$



SR Latches

Set Reset

1 1

(11) State not allowed

Clocked condition:

0 → Set/Reset

Race condition

(11) X