Digital Design

Revision for T2

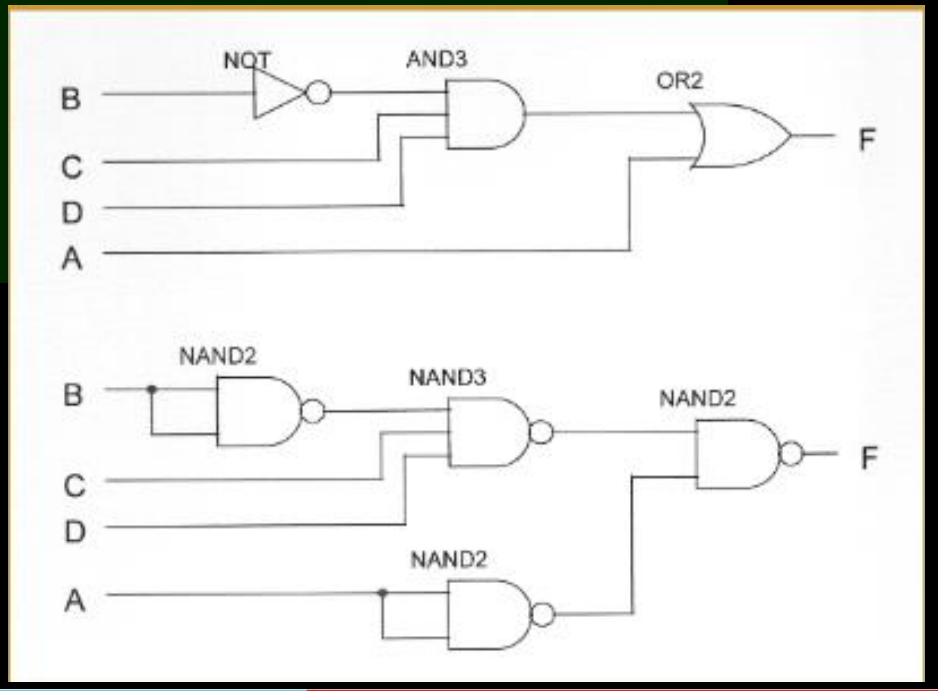
NAND Implementation

AB \ CD	00	01	11	10
00	0	0	1	0
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

B'CD

A

$F = A + B'CD$



NOR Implementation

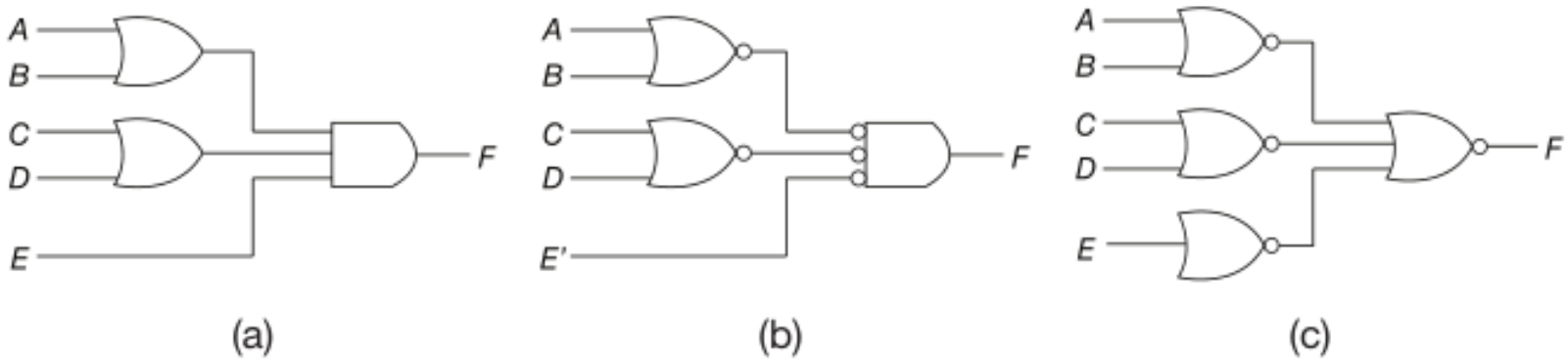


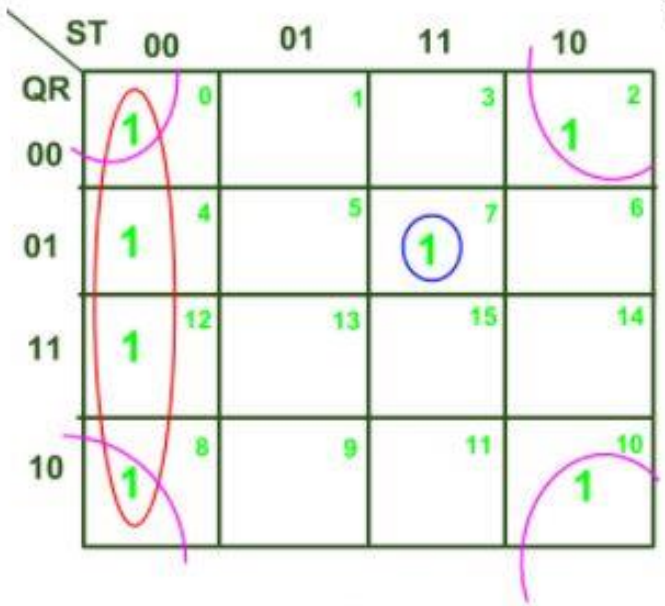
Figure 3.20 Three ways to implement $F = (A + B)(C + D)E$

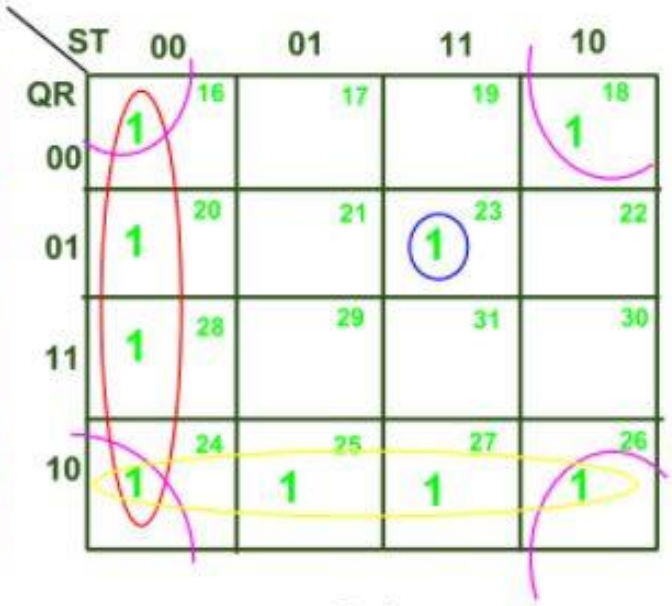
Minimization of Boolean Expression by 5 Variable K-Map (SOP form)

I. Solving SOP function –

For clear understanding, let us solve the example of SOP function minimization of 5 Variable K-Map using the following expression :

$$\sum m(0, 2, 4, 7, 8, 10, 12, 16, 18, 20, 23, 24, 25, 26, 27, 28)$$





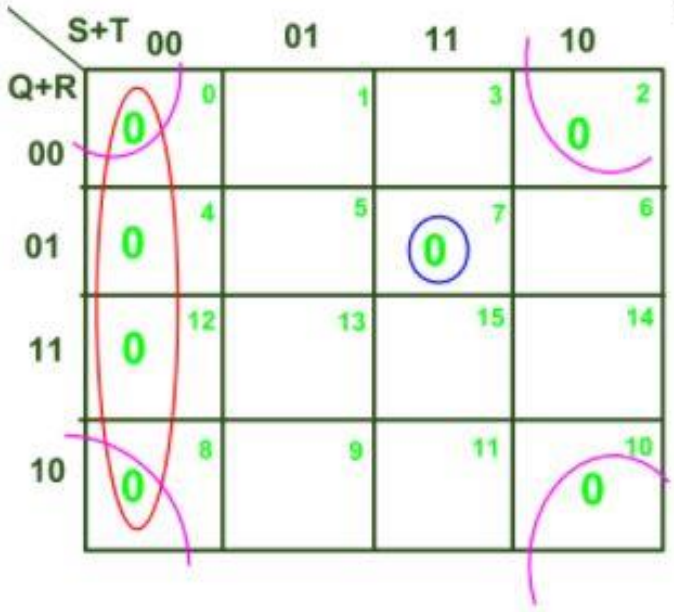
F(PQRST) = S'T' +Q'RST + PQR' +R'T'

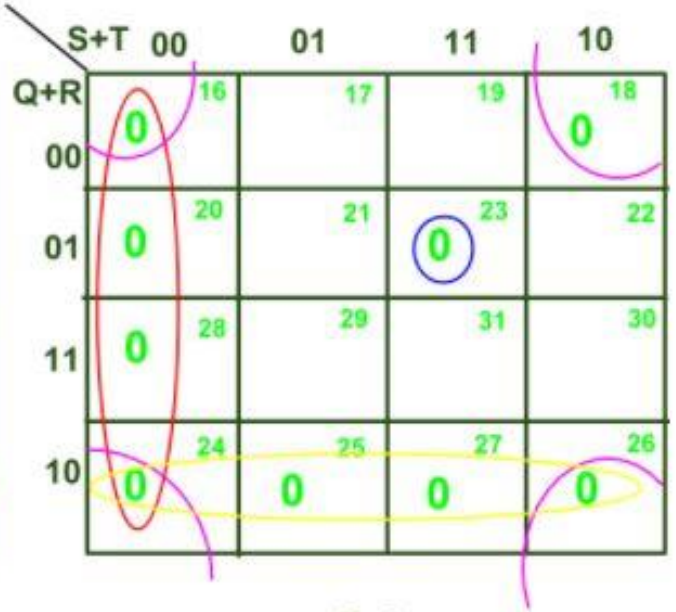
Minimization of Boolean Expression by 5 Variable K-Map (POS form)

II. Solving POS function –

Now, let us solve the example of POS function minimization of 5 Variable K-Map using the following expression :

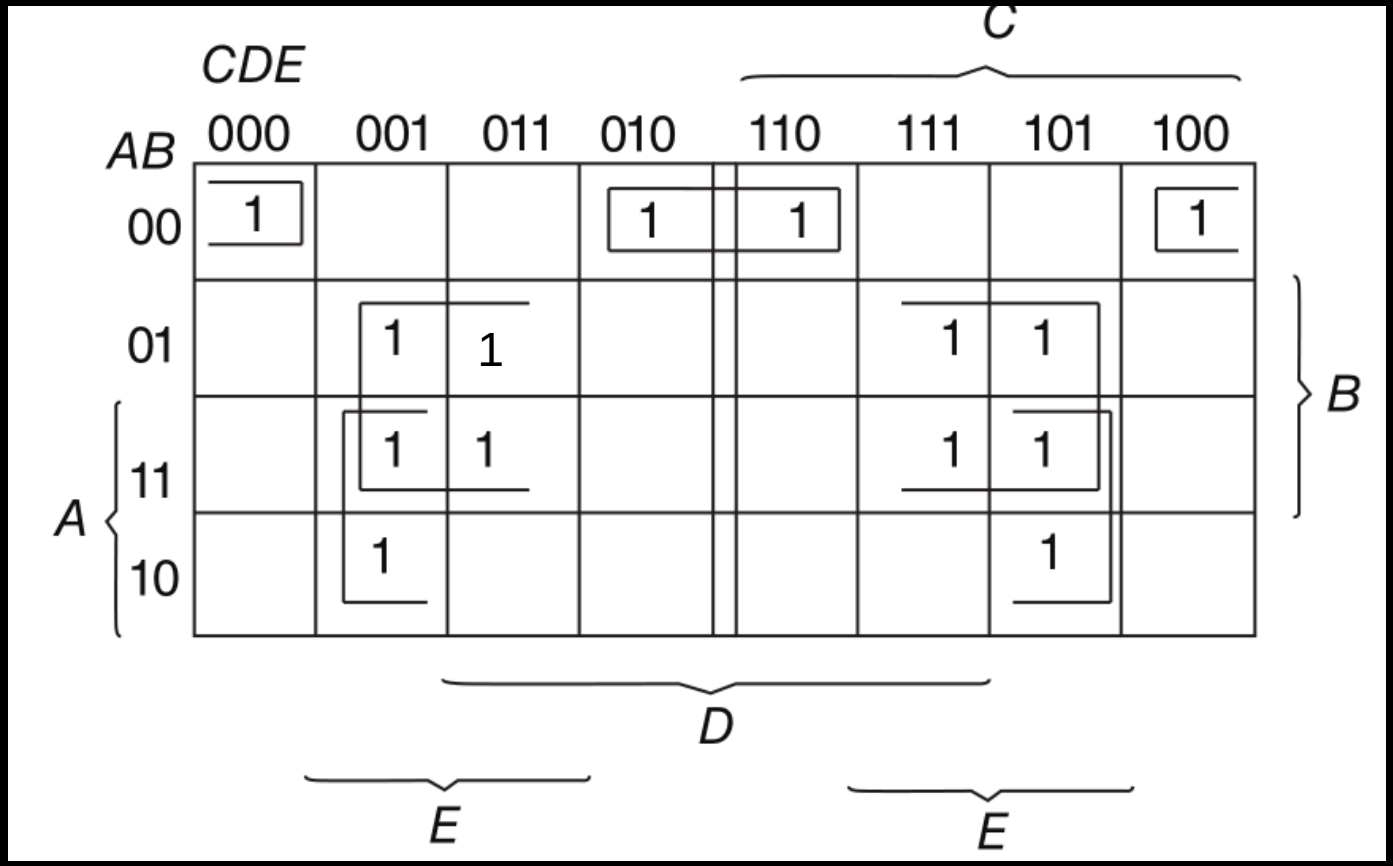
$$\prod M(0, 2, 4, 7, 8, 10, 12, 16, 18, 20, 23, 24, 25, 26, 27, 28)$$





f(PQRST) = (S+T) . (R+T) . (Q+R'+S'+T') . (P'+Q'+R)

$F(A, B, C, D, E) = \sum(0, 2, 4, 6, 9, 11, 13, 15, 17, 21, 25, 27, 29, 31)$



$BE + AD'E + A'B'E'$

Prime Implicants

	Column I	Column II	Column III
group 0	0 0000✓	0,1 000- ✓	0,1,8,9 -00- ⟵ D = x'y'
group 1	1 0001✓	0,2 00-0 ✓	0,2,8,10 -0-0 ⟵ E = x'z'
	2 0010✓	0,8 -000 ✓	2,6,10,14 --10 ⟵ F = yz'
	8 1000✓	1,5 0-01 ⟵ A = w'y'z	
group 2	5 0101✓	1,9 -001 ✓	
	6 0110✓	2,6 0-10 ✓	
	9 1001 ✓	2,10 -010 ✓	
	10 1010✓	8,9 100- ✓	
group 3	7 0111✓	8,10 10-0 ✓	
	14 1110✓	5,7 01-1 ⟵ B = w'xz	
		6,7 011- ⟵ C = w'xy	
		6,14 -110 ✓	
		10,14 1-10 ✓	

A, B, C, D, E, F are Prime Implicants

Essential Prime Implicants

2. Find Essential Prime Implicants

The Minterms 9 and 14 are covered by single terms

Prime Implicants		Minterms									
		0	**1**	**2**	**5**	**6**	**7**	**8**	**9**	**10**	**14**
(1,5)	A=w'y'z		X		X						
(5,7)	B=w'xz				X		X				
(6,7)	C=w'xy					X	X				
(0,1,8,9)	D=x'y'	X	X					X	X		
(0,2,8,10)	E=x'z'	X		X				X		X	
(2,6,10,14)	F=yz'			X		X				X	X

$$f(w,x,y,z) = \sum(0,1,2,5,6,7,8,9,10,14)$$

yz' and **x'y'**

Essential Prime Implicants

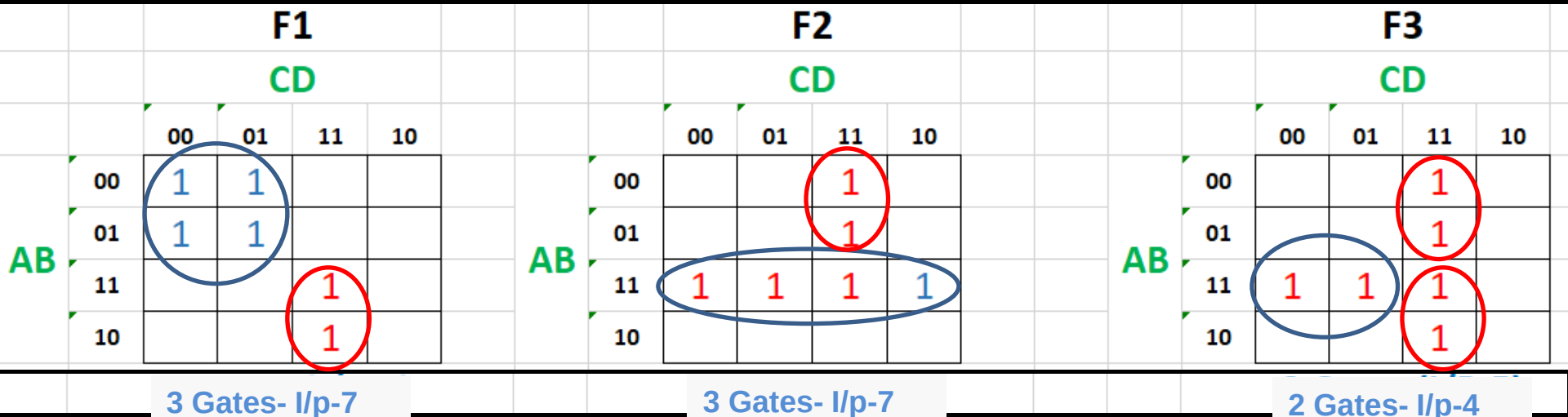
Static Glitch Elimination

a \ bc	00	01	11	10
0	0	1	0	0
1	0	1	1	1

$f=ab+b'c + ac$

The extra product term does not include the changing input variable, and therefore serves to prevent possible momentary output glitches due to this variable.

Multi-Output Circuit Optimization

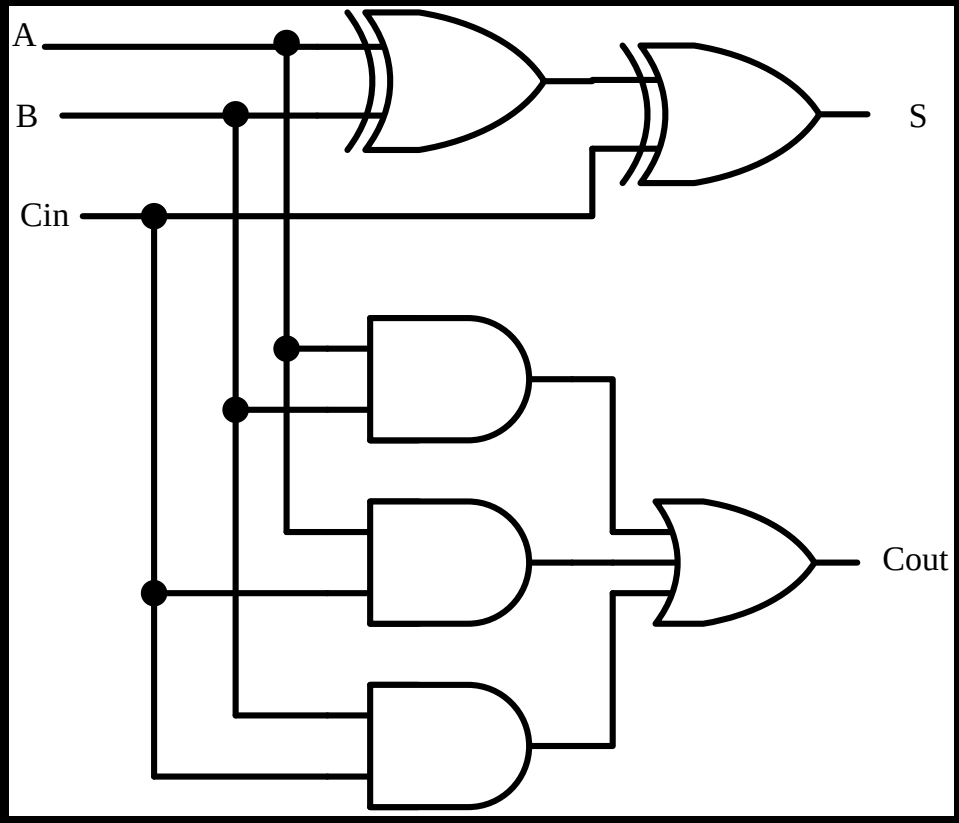


Full Adder

Two level logic

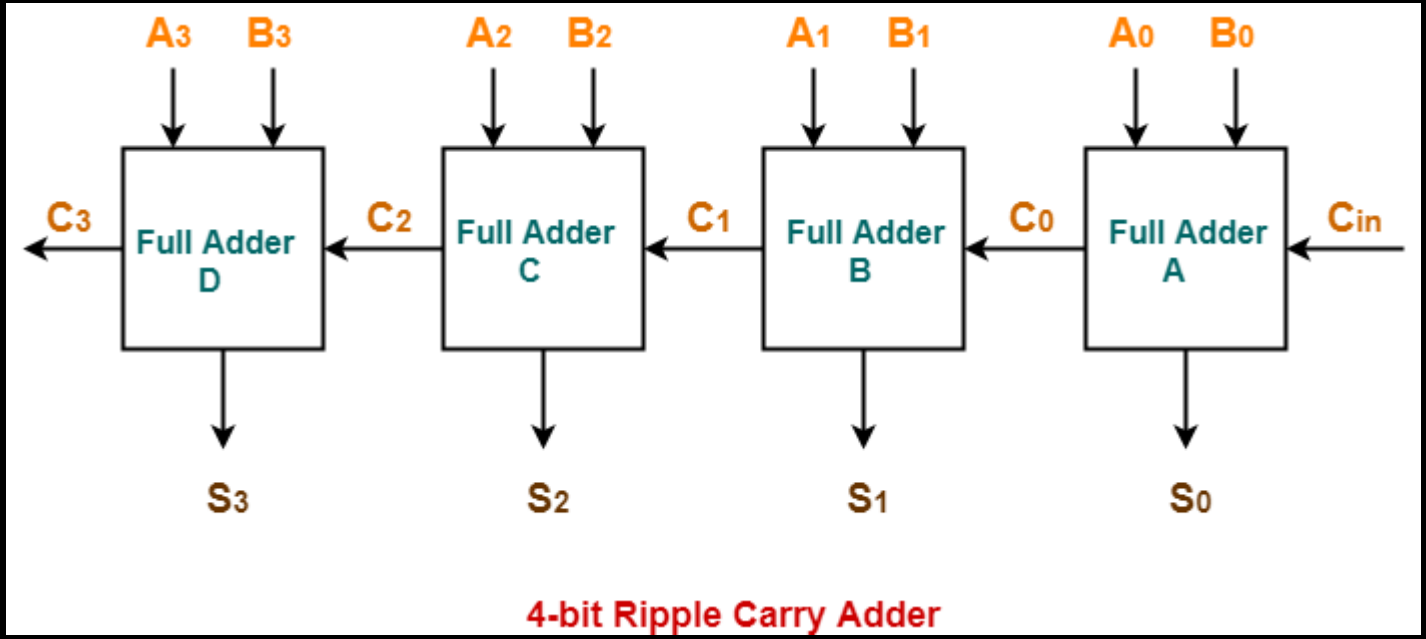
Output delay

- A, B Cin → Sum
- Delay through 2 xor gates
- $t_s = 2\ t_{xor}$
- A, B Cin → Cout
- Delay through an AND gate and an OR gate
- $t_{Cout} = t_{and} + t_{or}$



Ripple Carry Adder

This is called Ripple Carry Adder ,because of the construction with full adders are connected in cascade.

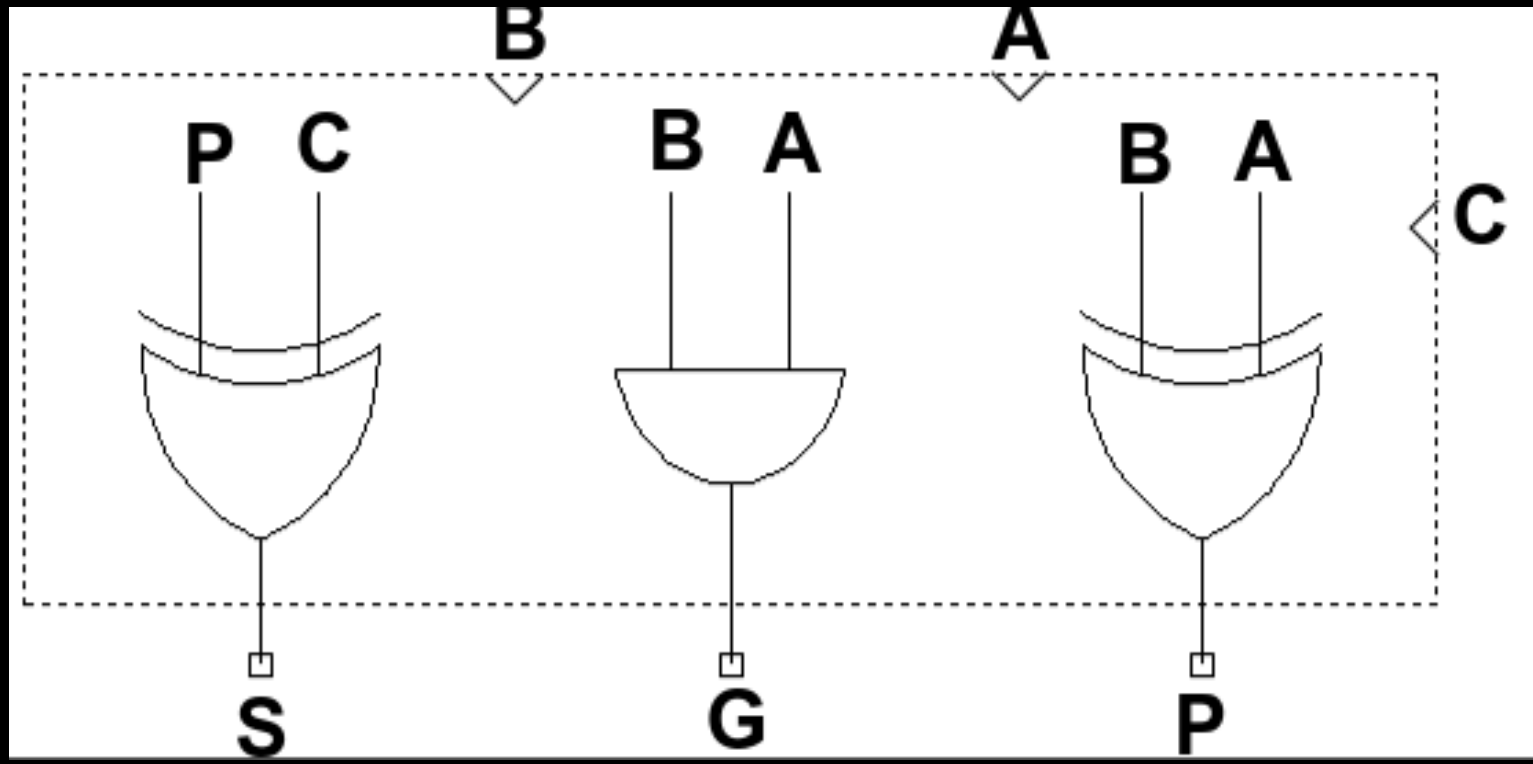


4-bit Ripple Carry Adder

Delay= 4 X Full Adder Delay

Carry Look-Ahead Adder

1-bit CLA



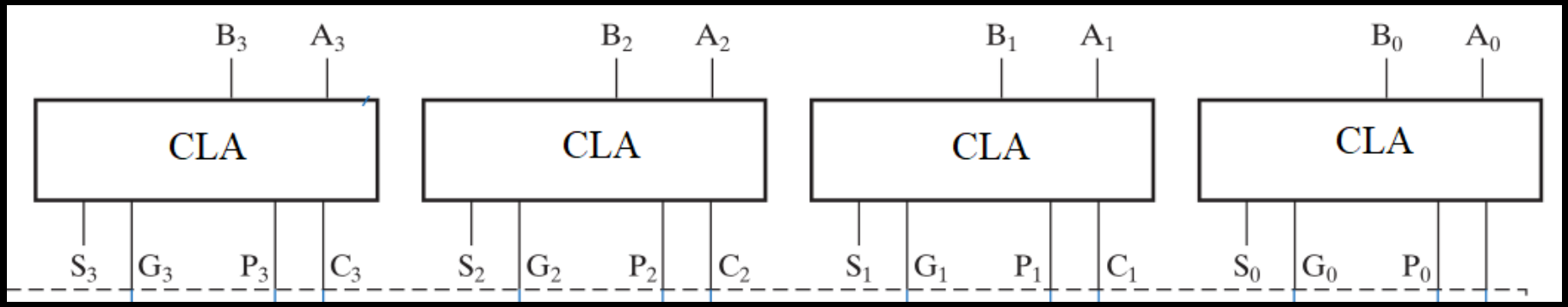
Carry Look-Ahead Adder

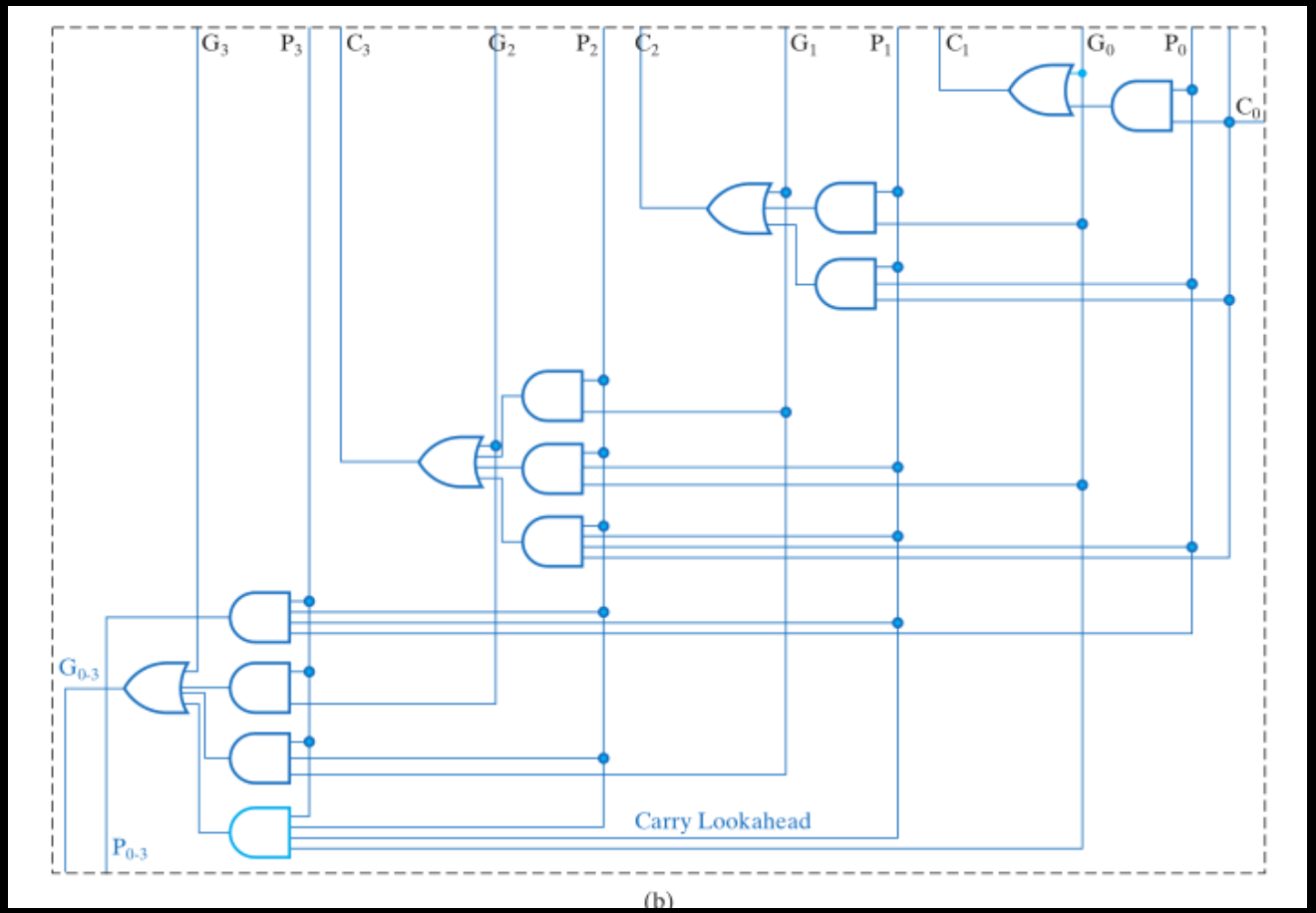
$$C_1=G_0+P_0C_0$$

$$C_2=G_1+P_1G_0+P_1P_0C_0$$

$$C_3=G_2+P_2G_1+ P_2P_1G_0 +P_2P_1P_0C_0$$

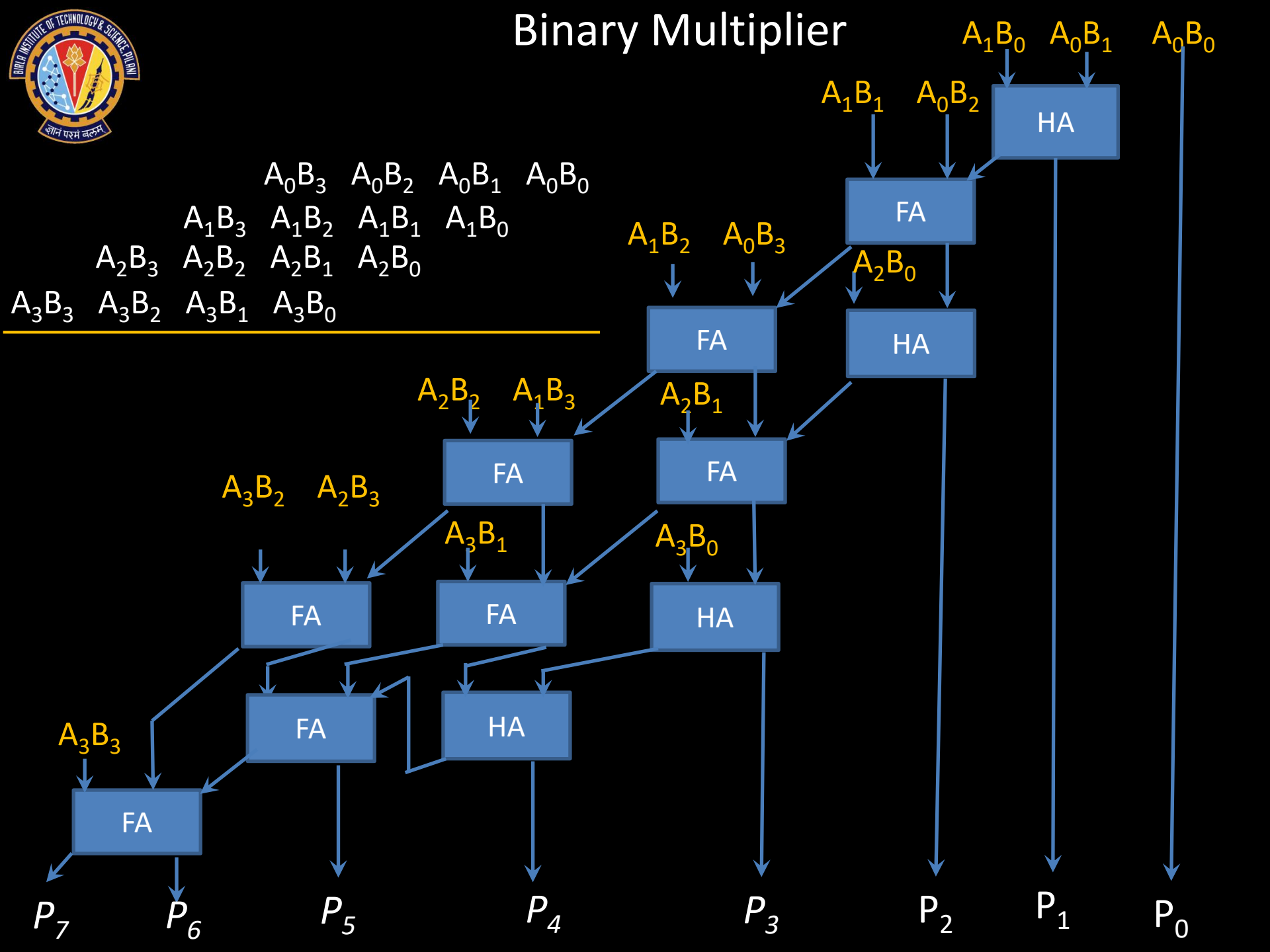
$$C_4=G_3+P_3G_2+ P_3P_2G_1 +P_3P_2P_1G_0 +P_3P_2P_1P_0C_0$$


B3
A3
B2
A2
B1
A1
B0
A0
CLA
CLA
CLA
CLA
S3
G3
P3
C3
S2
G2
P2
C2
S1
G1
P1
C1
S0
G0
P0

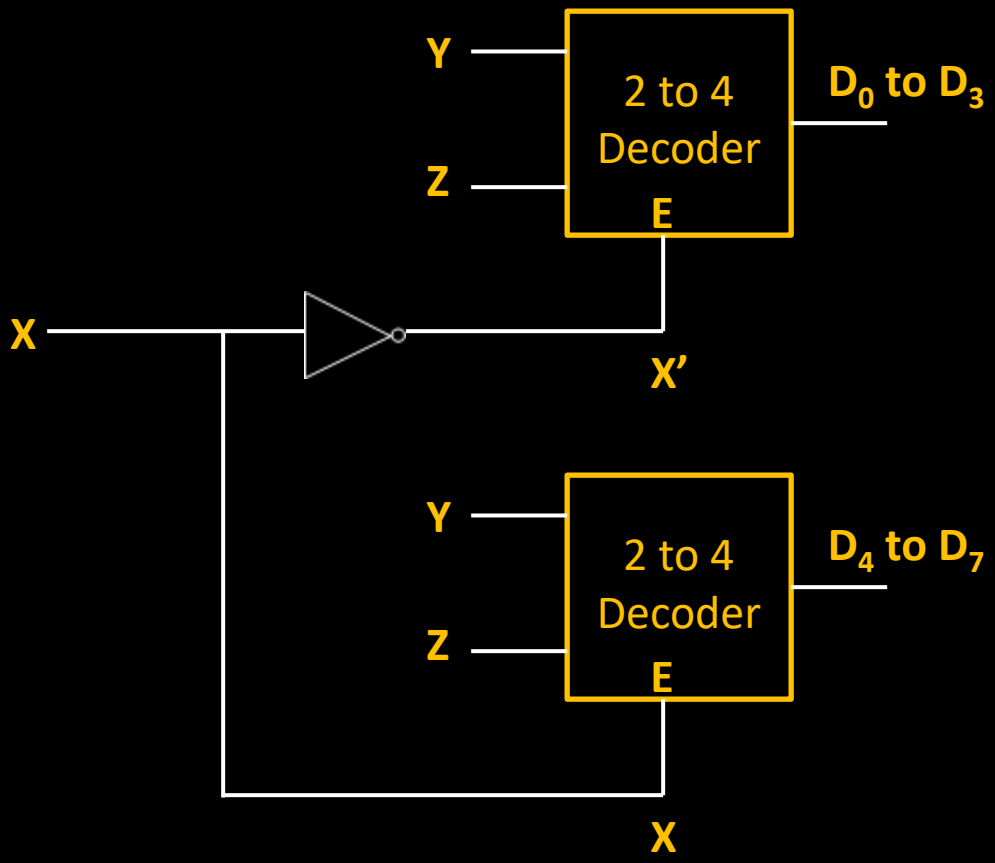

G3
P3
C3
G2
P2
C2
G1
P1
C1
G0
P0
C0
G0-3
P0-3
Carry Lookahead

(b)

Binary Multiplier



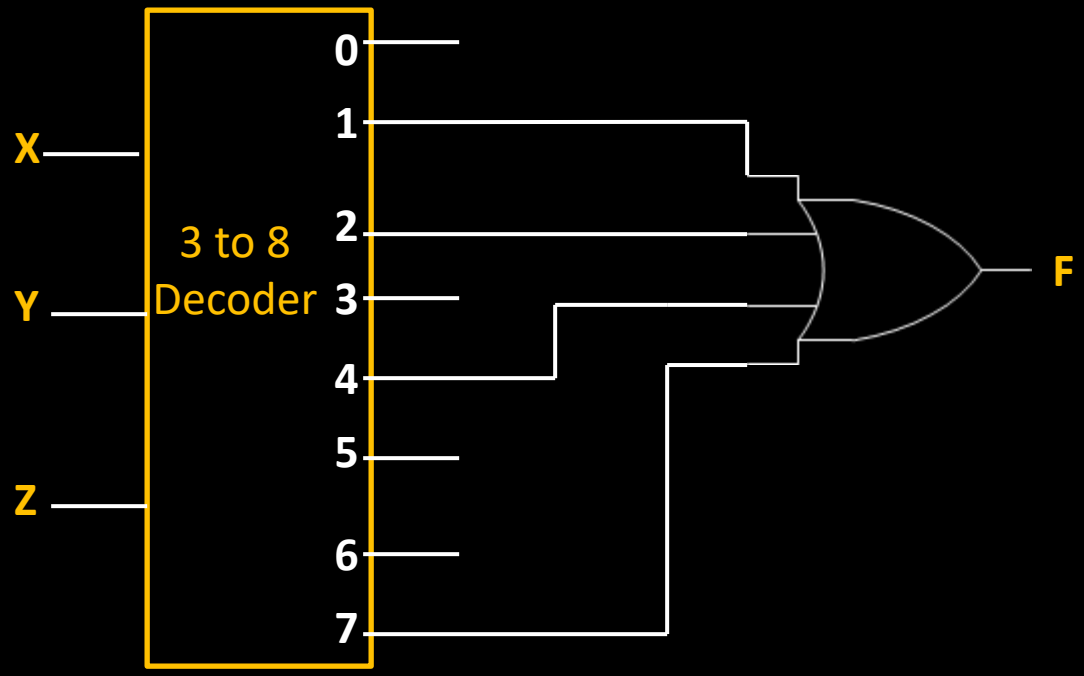
Decoder



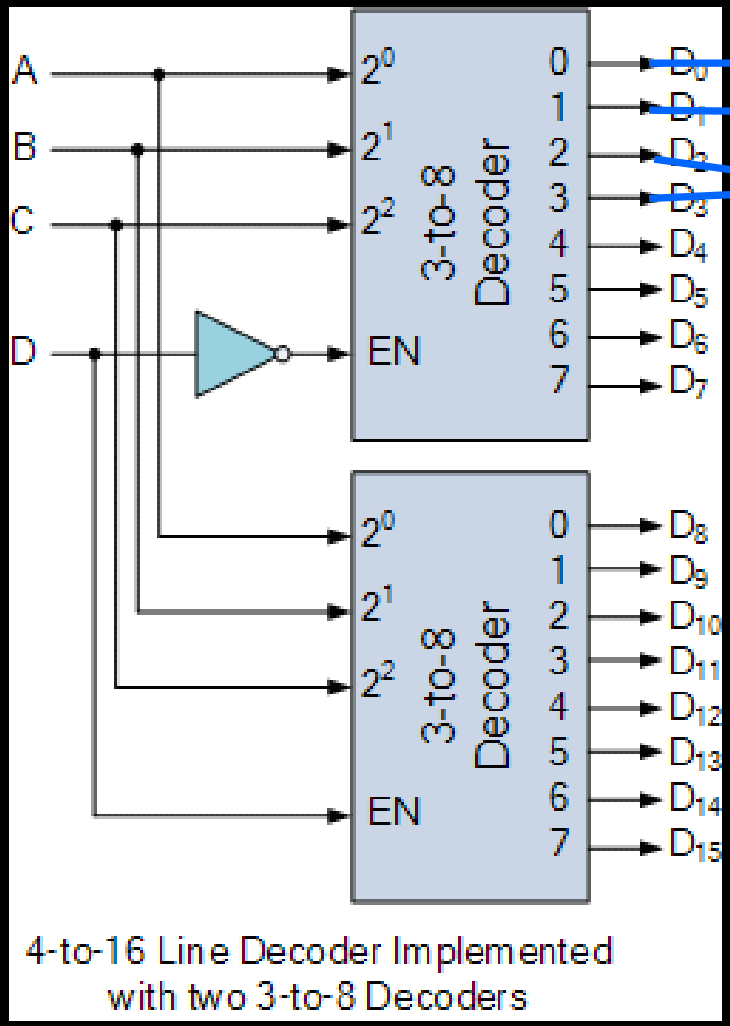
Decoder

Implementing function using Decoders

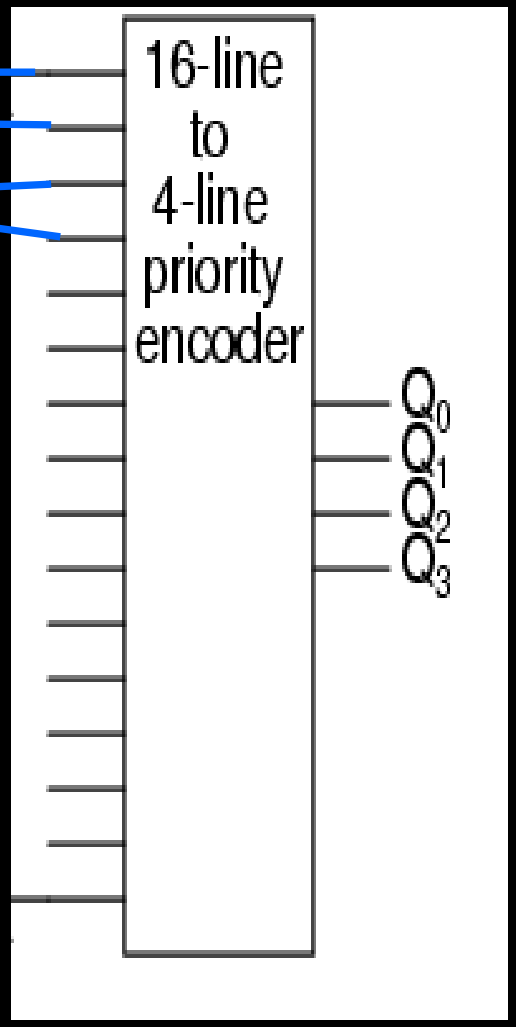
$F = \sum (1, 2, 4, 7) = X'Y'Z + X'YZ' + XY'Z' + XYZ$



Binary to Gray Using Encoder-Decoder



4-to-16 Line Decoder Implemented with two 3-to-8 Decoders



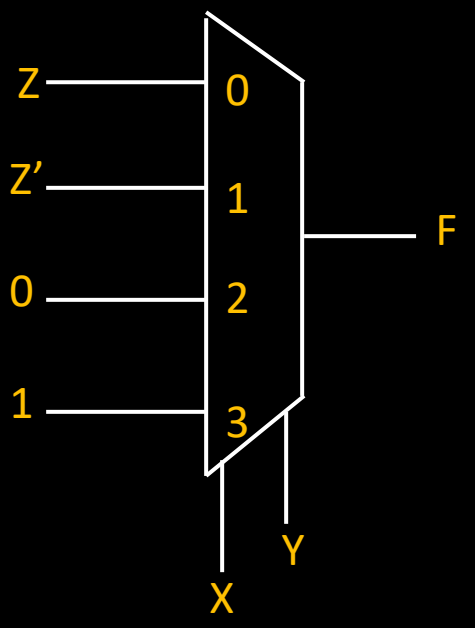
Multiplexers

Boolean Function implementation

$F(X,Y,Z) = \sum (1, 2, 6, 7)$

X	Y	Z	F	
0	0	0	0	F = Z
0	0	1	1	
0	1	0	1	F = Z’
0	1	1	0	
1	0	0	0	F = 0
1	0	1	0	
1	1	0	1	F = 1
1	1	1	1	

4:1 Multiplexer



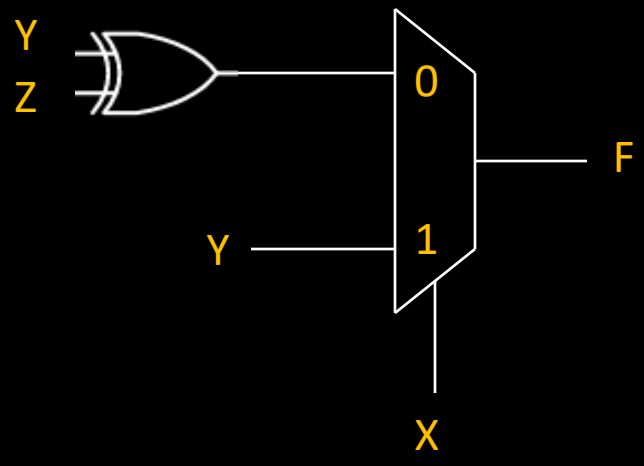
Multiplexers

Boolean Function implementation

$F(X,Y,Z) = \sum (1, 2, 6, 7)$

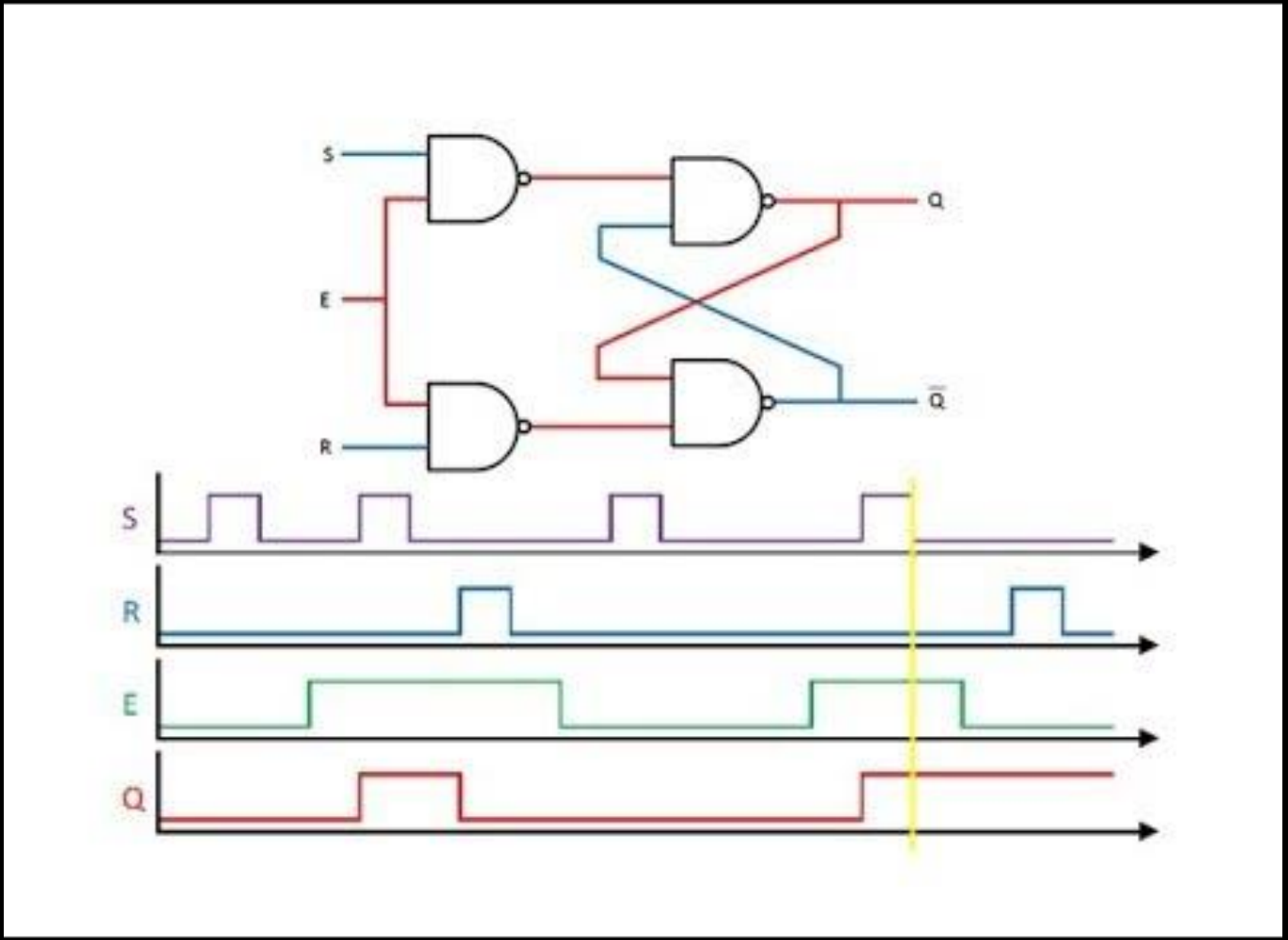
X	Y	Z	F	
0	0	0	0	
0	0	1	1	$F = Y \oplus Z$
0	1	0	1	
0	1	1	0	
1	0	0	0	
1	0	1	0	
1	1	0	1	$F = Y$
1	1	1	1	

2:1 Multiplexer



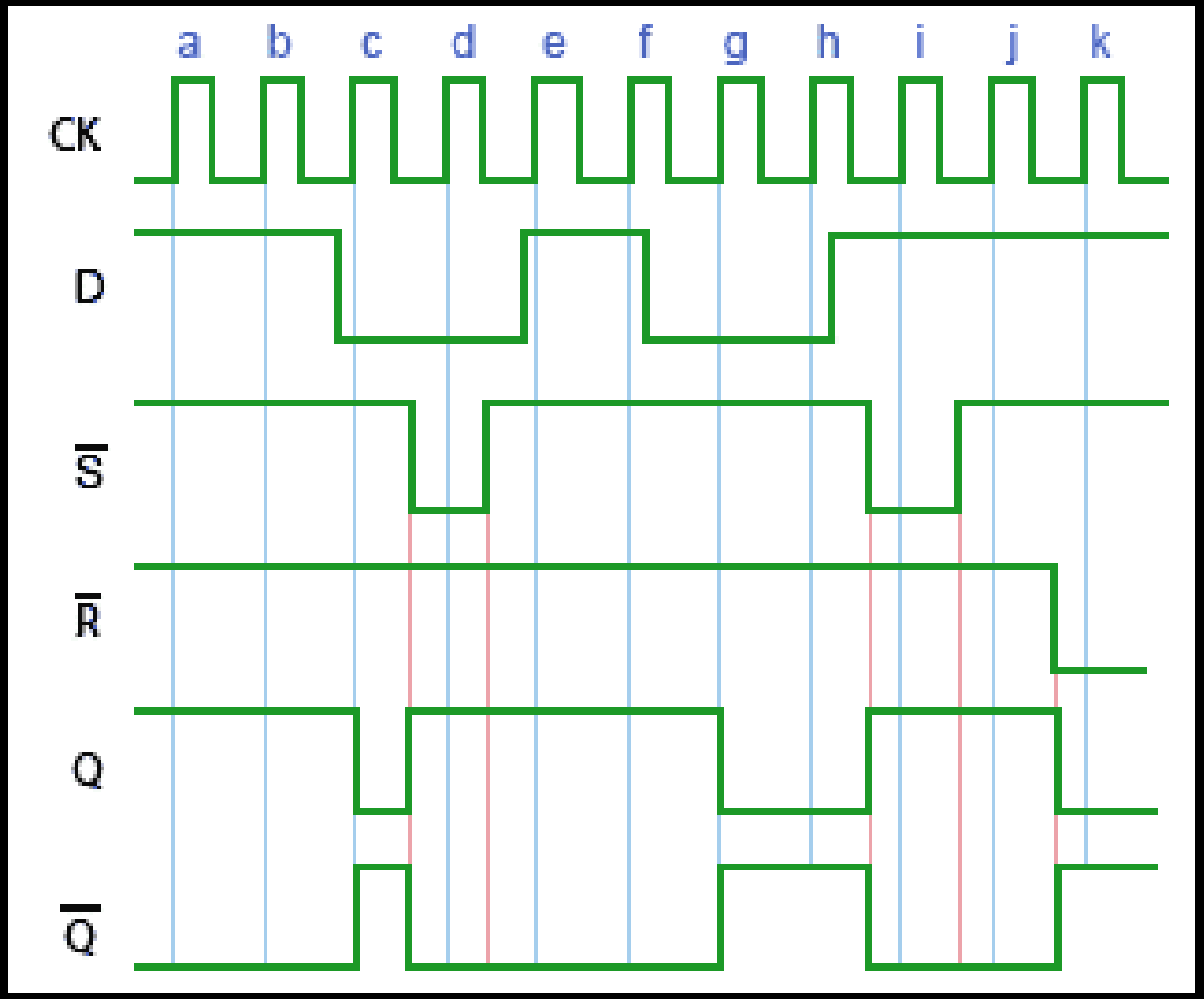
Latches

SR Latch with enable



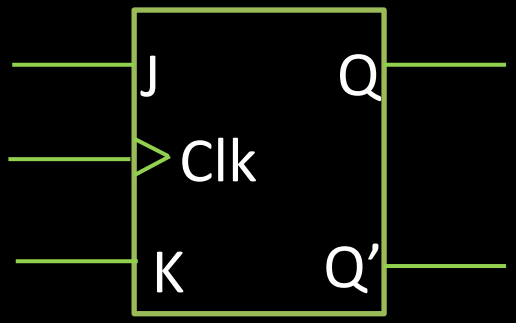
Edge Triggered D Flip-Flops

D-type positive edge triggered



Other Flip-Flops

JK Flip-flop



Characteristic Table

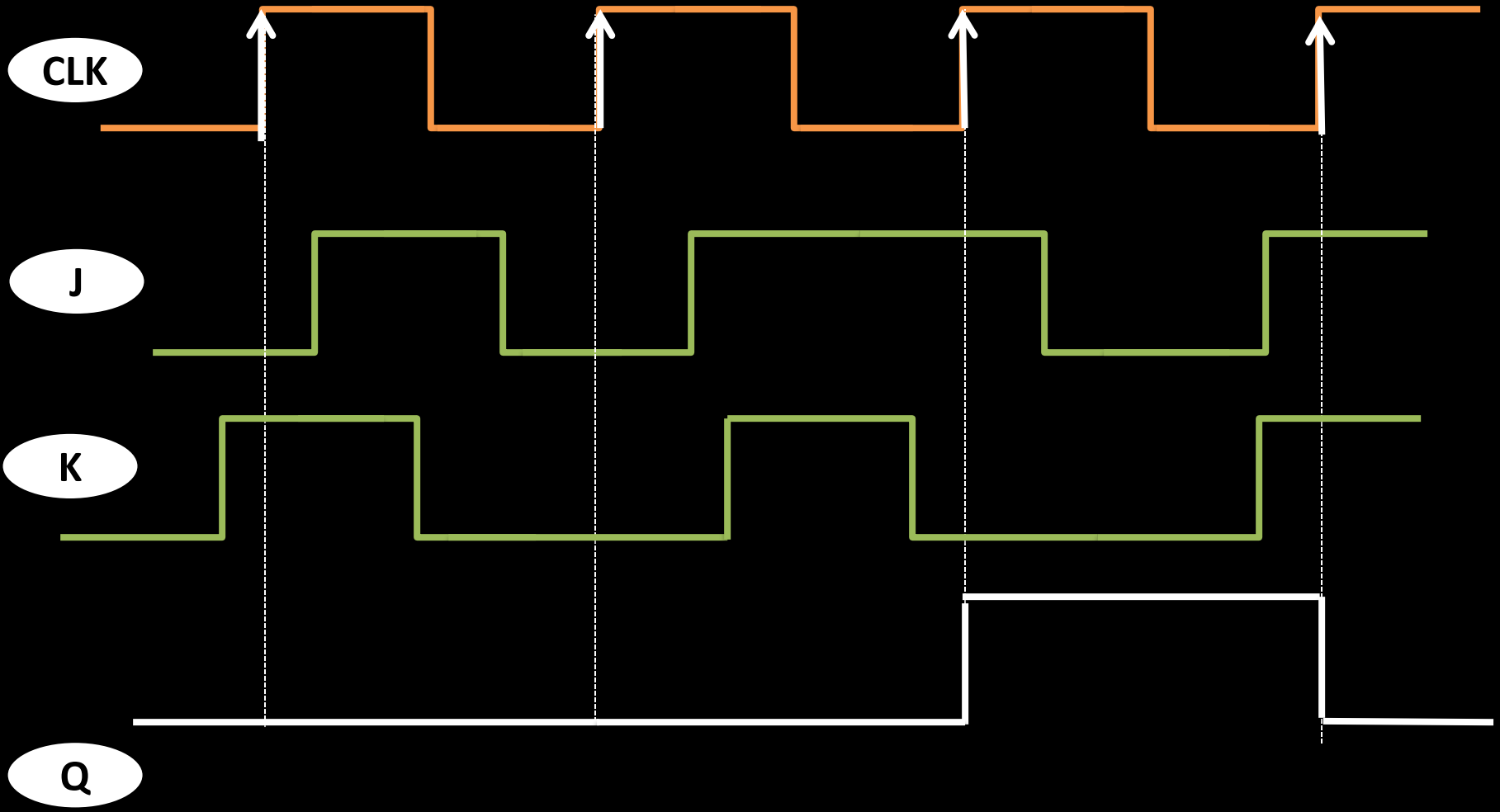
J	K	Q(t+1)	
0	0	Q(t)	No Change
0	1	0	Reset
1	0	1	Set
1	1	Q'(t)	Toggle/ Complement

Characteristic Equation??

$$Q(t+1) = JQ' + K'Q$$

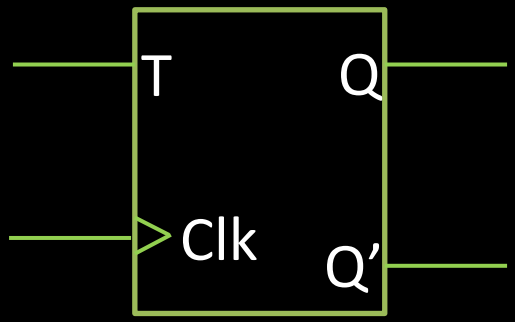
Edge Triggered JK Flip-Flops

Positive Edge triggered



Other Flip-Flops

T Flip-flop



Characteristic Table

T	Q(t+1)	
0	Q(t)	No Change
1	Q'(t)	Toggle

Characteristic Equation??

$$Q(t+1) = T \oplus Q$$

Edge Triggered T Flip-Flops

Positive Edge triggered