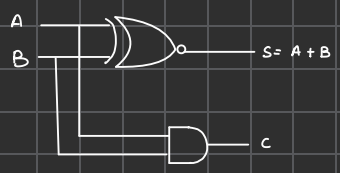
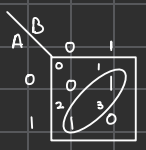
Half Adder/Full

	A	B	S	C	
0	0	0	0	0	← 0
1	0	1	1	0	← 1
2	1	0	1	0	← 2
3	1	1	0	1	← 3

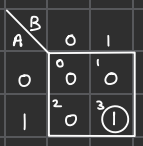


Sum



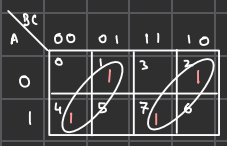
$S \rightarrow A \oplus B$

Carry



$C = A.B$

	A	B	C	S	C
0	0	0	0	0	0
1	0	0	1	1	0 ←
2	0	1	0	1	0 ←
3	0	1	1	0	1 ←
4	1	0	0	1	0 ←
5	1	0	1	0	1 ←
6	1	1	0	0	1 ←
7	1	1	1	1	1 ←



$A \oplus B'C + A + BC'$

$= A \oplus B \oplus C$

$Sum = A'B'C + A'BC' + AB'C' + ABC$

$C(A'B' + AB) + C'(A'B + AB')$

$C\{(A'+A).(A'+B).(B'+A).(B'+B)\} + C'(A \oplus B)$

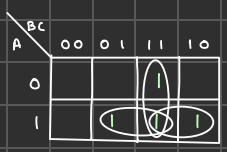
$C\{(A'+B).(B'+A)\} + C'(A \oplus B)$

$C\{(A'B + AB')'\} + C'(A \oplus B)$

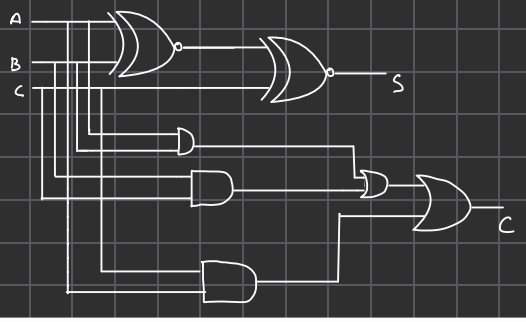
$C(A \oplus B)' + C'(A \oplus B)$

$(A \oplus B) \oplus C$

Carry

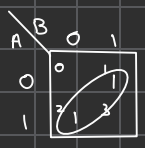


$C_r = BC + AC + AB$

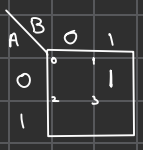


Half Subtractor

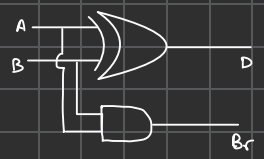
A	B	D	Br
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0



$A \oplus B$

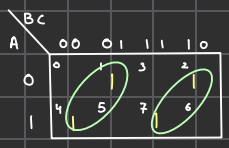


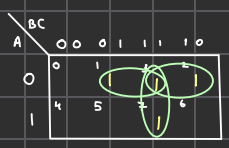
$\bar{A}B$



Full Subtractor

	A	B	C	D	Br	
0	0	0	0	0	0	
1	0	0	1	1	1	←
2	0	1	0	1	1	←
3	0	1	1	0	1	←
4	1	0	0	1	0	
5	1	0	1	0	0	
6	1	1	0	0	0	
7	1	1	1	1	1	←





$A \oplus B'C + A \oplus BC'$

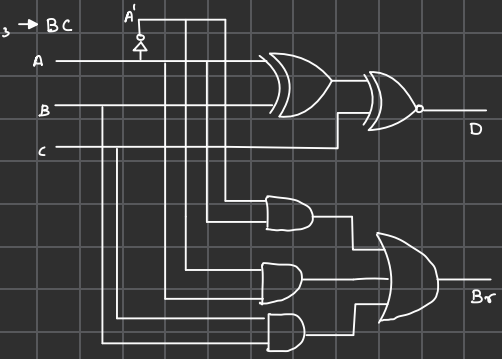
$A \oplus (B'C + BC')$

$A \oplus B \oplus C$

$G_1 \rightarrow A'C$

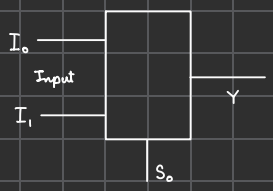
$G_2 \rightarrow A'B$

$G_3 \rightarrow BC$



Multiplexers

2:1 MUX



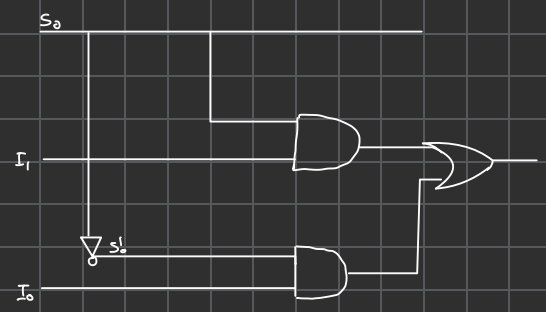
	S_0	I_0	I_1	Y
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1 ←
3	0	1	1	1 ←
4	1	0	0	0
5	1	0	1	1 ←
6	1	1	0	0
7	1	1	1	1 ←

S_0
0 I_0
1 I_1

K-Map of Y

S_0 \ I_0I_1	00	01	11	10
0	0	1	3	2
1	4	5	7	6

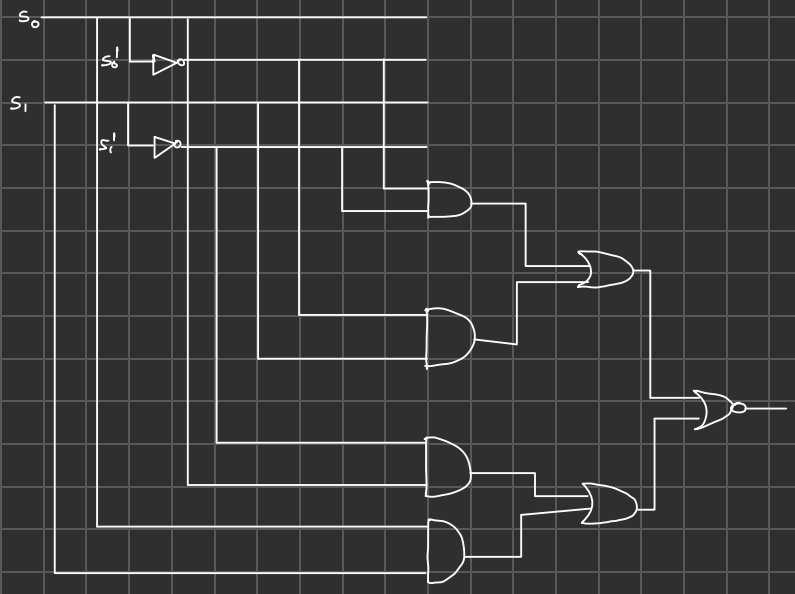
$S_0' I_0 + S_0 I_1$



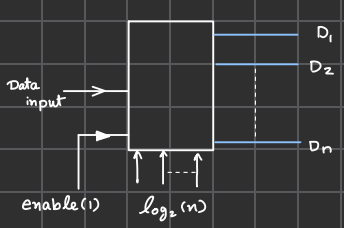
4:1 MUX

S_0	S_1	Im
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

$S_0' S_1' I_0 + S_0' S_1 I_1 + S_0 S_1' I_2 + S_0 S_1 I_3$



Demultiplexer



one input → many output

$n = 2^m$ ← select line

decision line

Truth table for demux

E	S_0	Y_0	Y_1	
0	0	0	0	don't care
0	1	0	0	condition
1	0	I_0	0	
1	1	0	I_0	

$Y_0 = E \bar{S}_0 I_0$

$Y_1 = E S_0 I_0$

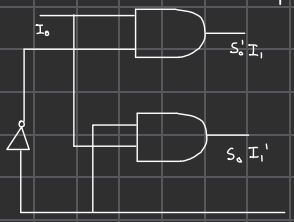
E	S_0	S_1	Y_0	Y_1	Y_2	Y_3
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	I_0	0	0	0
1	0	1	0	I_0	0	0
1	1	0	0	0	I_0	0
1	1	1	0	0	0	I_0

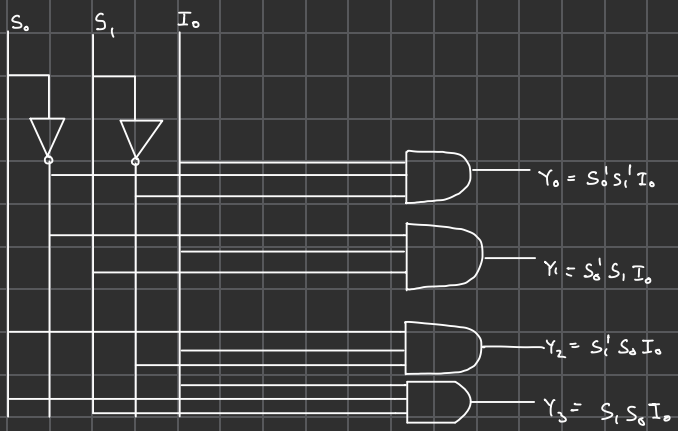
$Y_0 = S_0' S_1' I_0$

$Y_1 = S_0' S_1 I_0$

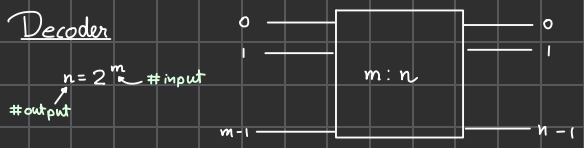
$Y_2 = S_0 S_1' I_0$

$Y_3 = S_0 S_1 I_0$





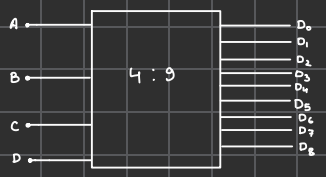
Decoder



$n = 2^m$ ← # input

output

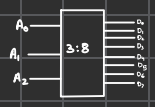
3:8 decoder



A	B	C	D	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8
0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	×	×	×	×	×	×	×	×	×
1	0	1	0	×	×	×	×	×	×	×	×	×
1	0	1	1	×	×	×	×	×	×	×	×	×
1	1	0	0	×	×	×	×	×	×	×	×	×
1	1	0	1	×	×	×	×	×	×	×	×	×
1	1	1	0	×	×	×	×	×	×	×	×	×
1	1	1	1	×	×	×	×	×	×	×	×	×

don't care

3:8 Decoder



A_0	A_1	A_2	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

$D_0 = A_0' A_1' A_2'$

$D_1 = A_0' A_1' A_2$

$D_2 = A_0' A_1 A_2'$

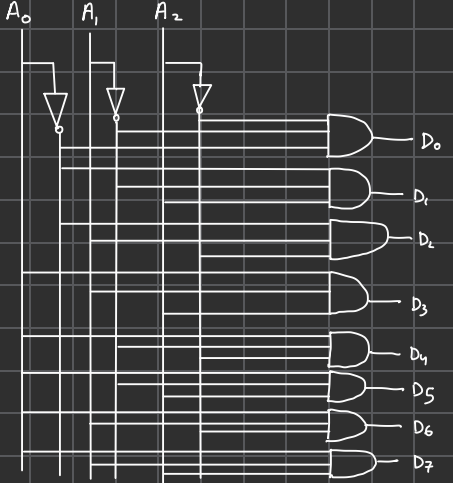
$D_3 = A_0' A_1 A_2$

$D_4 = A_0 A_1' A_2'$

$D_5 = A_0 A_1' A_2$

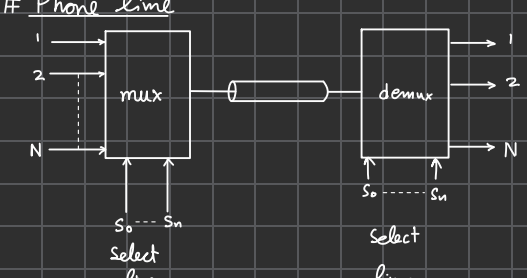
$D_6 = A_0 A_1 A_2'$

$D_7 = A_0 A_1 A_2$



Application for mux-demux

Phone line

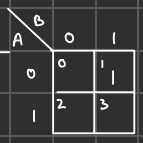


Comparator

1-bit Comparator

A	B	A=B	A>B	A<B
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

① $A<B$



$Y_1 = A'B$

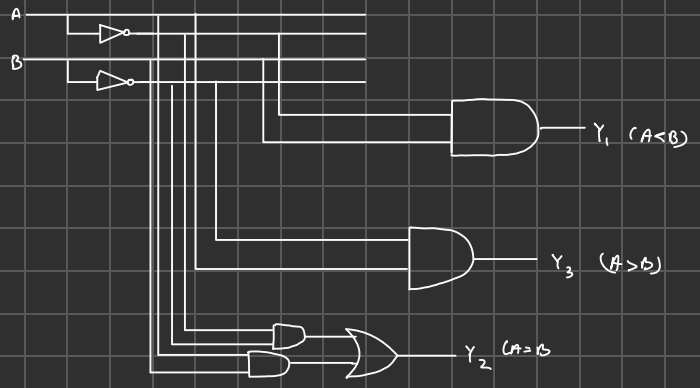
② $A=B$

$Y_2 = \overline{A \oplus B}$ or

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

③ $A>B$

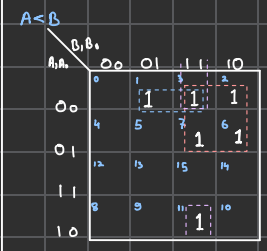
$Y_3 = AB'$



2-bit Comparator

	A_1	A_0	B_1	B_0	A<B	A=B	A>B
0	0	0	0	0	0	1	0
1	0	0	0	1	1	0	0
2	0	0	1	0	1	0	0
3	0	0	1	1	1	0	0
4	0	1	0	0	0	0	1
5	0	1	0	1	0	1	0
6	0	1	1	0	1	0	0
7	0	1	1	1	1	0	0
8	1	0	0	0	0	0	1
9	1	0	0	1	0	0	1
10	1	0	1	0	0	1	0
11	1	0	1	1	1	0	0
12	1	1	0	0	0	0	1
13	1	1	0	1	0	0	1
14	1	1	1	0	0	0	1
15	1	1	1	1	0	1	0

A<B



$G_1 = A_1' A_0' B_0$

$G_2 = B_1 B_0 A_0'$

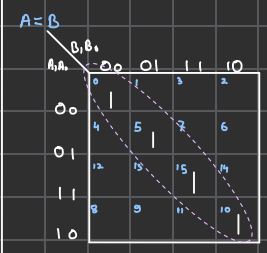
$G_3 =$

0	0	1	1
0	0	1	0
0	1	1	1
0	1	1	0

$Y_1 = A_1' A_0' B_0 + B_1 B_0 A_0' + A_1' B_1$

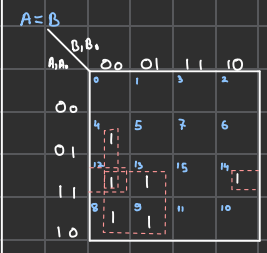
$= A_1' B_1$

A=B



$Y_2 = \overline{A_1 \oplus B_1 \oplus A_0 \oplus B_0}$

A=B



$G_1 = A_0 B_1' B_0'$

$G_2 = A_0 A_1 B_0'$

$G_3 = A_1 B_1'$

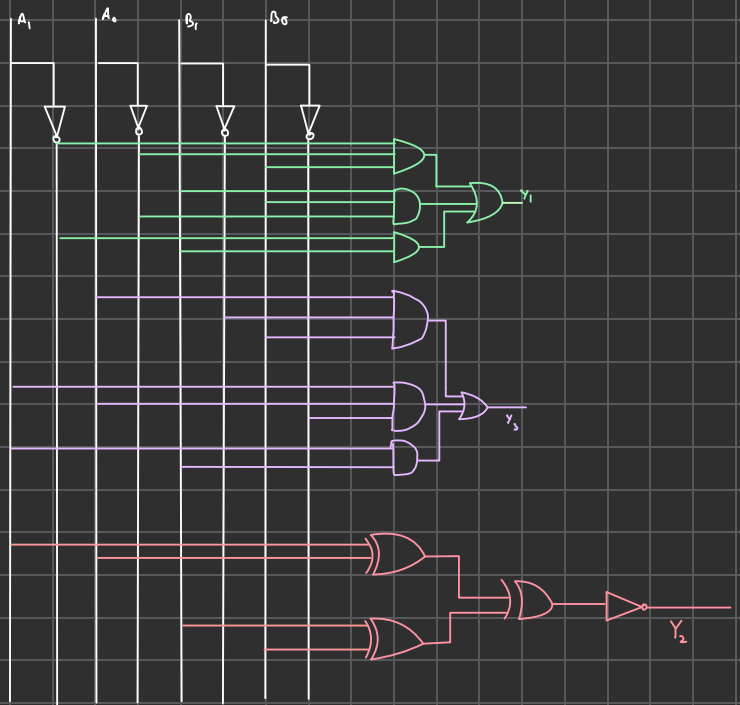
1	1	0	0
1	1	0	1
1	0	0	0
1	0	0	1

$Y = A_0 \bar{B}_1 B_0 + A_1 \bar{B}_1 + A_1 A_0 \bar{B}_0$

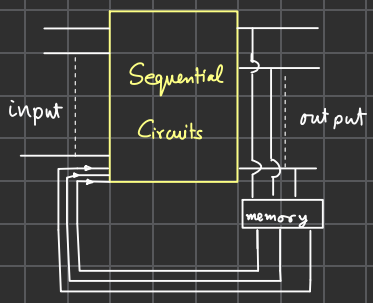
$Y_1 = A_1' A_0' B_0 + B_1 B_0 A_0' + A_1' B_1$

$Y_2 = \overline{A_1 \oplus B_1 \oplus A_0 \oplus B_0}$

$Y_3 = A_0 \bar{B}_1 B_0 + A_1 \bar{B}_1 + A_1 A_0 \bar{B}_0$



Sequential Circuit

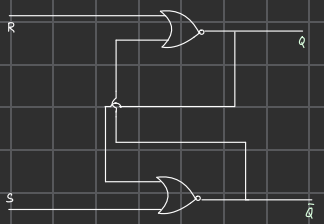


Important Sequential Circuit

- ✓ Latch
- ✓ Counter
- ✓ Flip-flop
- ✓ register

Basic Storage Element

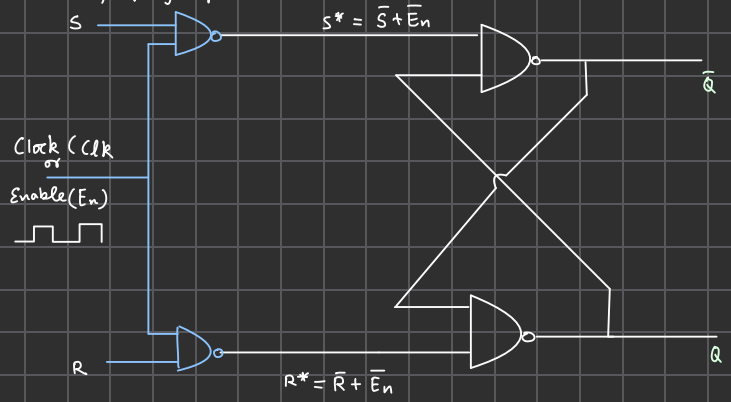
S-R Latch (Set - Reset)



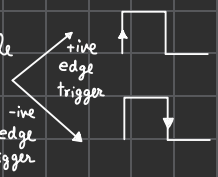
S	R	Q	$\bar{Q}$
0	0	Q_{n-1}	$\bar{Q}_{n-1}$
0	1	0	1
1	0	1	0
1	1	Invalid Output	

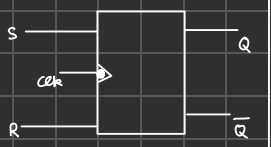
combinational circuit with capacity to store 1

S-R flip flop



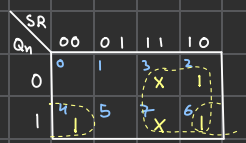
Flop → 50% duty cycle or edge trigger





S	R	Q_n	Q_{n+1}	Q'_{n+1}
0	0	X	Q_n	Q_n
0	1	X	0	1
1	0	X	1	0
1	1	X	Invalid o/p	Invalid o/p

	Q_n	S	R	Q_{n+1}	Q'_{n+1}	
0	0	0	0	0	0	
1	0	0	1	0	1	
2	0	1	0	1	0	←
3	0	1	1	X	X	←
4	1	0	0	1	0	←
5	1	0	1	0	1	
6	1	1	0	1	0	←
7	1	1	1	X	X	←



G_1

Q_n	S	R
0	1	1
0	1	0
1	1	1
1	1	0

→ S

$G_2 \rightarrow Q_n R'$

$Q_{n+1} = S + Q_n R'$

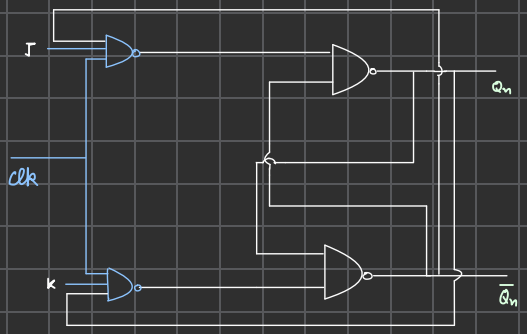
Excitation Table

Q_n	Q_{n+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

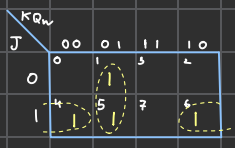
Toggle 1 0 1 0.....
0 1 0 1.....

↳ memory toggle.

J-K Flip flop



J	K	Q_n	Q_{n+1}	Q'_{n+1}	
0	0	0	0	1	memory state (Q_n)
0	0	1	1	0	
0	1	0	0	1	Reset
0	1	1	0	1	
1	0	0	1	0	Set
1	0	1	1	0	
1	1	0	1	0	Toggle (Q_n')
1	1	1	0	1	



$Q_{n+1} = K' Q_n + J Q_n'$

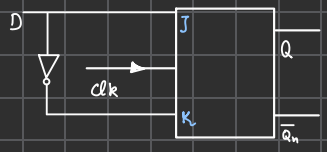
Excitation Table

Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

J	K	Q_n	Q_{n+1}	Q_{n+1}
0	0	X	Q_n	Q_n
0	1	X	0	1
1	0	X	1	0
1	1	X	Q_n'	Q_n'

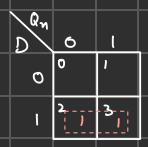
Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

D-flip flop # widely used as buffer

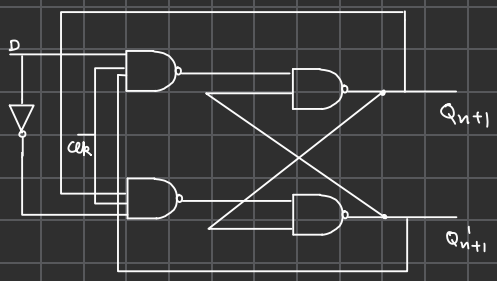


clk	D	Q	$\bar{Q}$
0	×	memory	memory
1	1	1	0
1	0	0	1

D	Q_n	Q_{n+1}	$\overline{Q_{n+1}}$
0	0	0	1
0	1	0	1
1	0	1	0
1	1	1	0



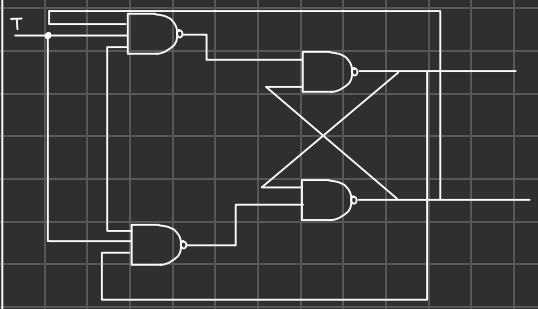
$Q_{n+1} = D$



Excitation Table

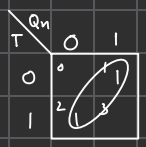
Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

T-flip flop



Truth Table

T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

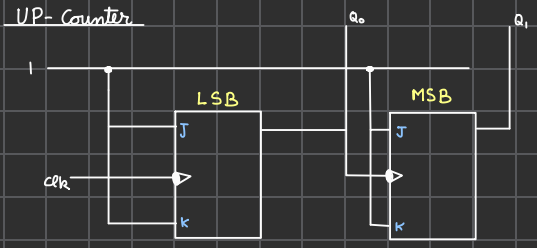


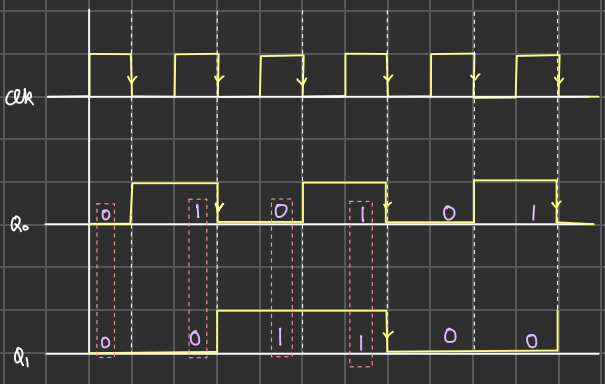
$T \oplus Q_n$

Asynchronous Counter using T-flip flop

2-bit Counter : -ive Edge trigger

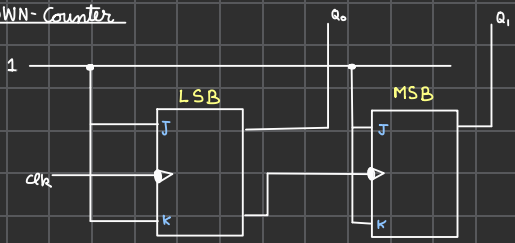
UP-Counter

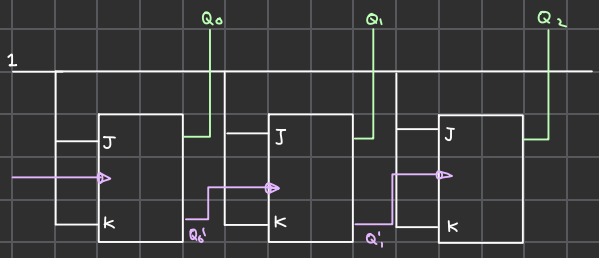




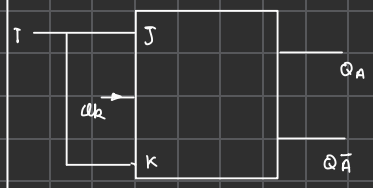
clk	Q_1	Q_2	
0	0	0	0
1	0	1	1
2	1	0	2
3	1	1	3
4	0	0	0
5	0	1	1

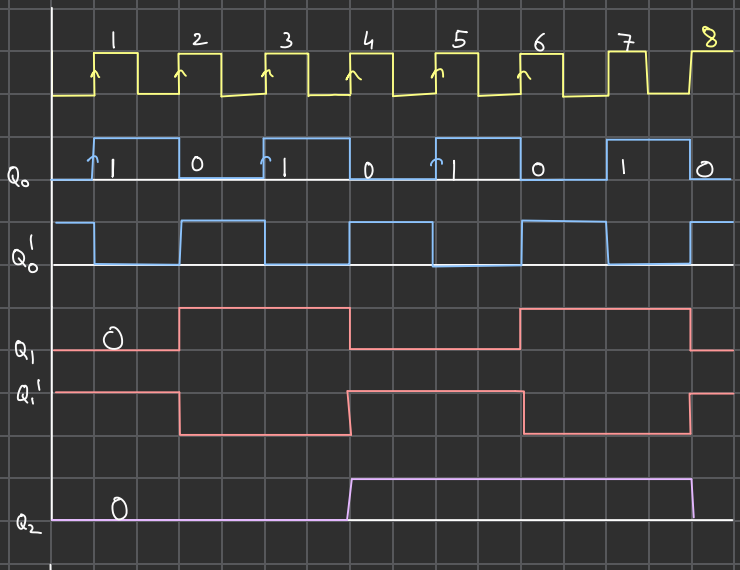
DWN-Counter





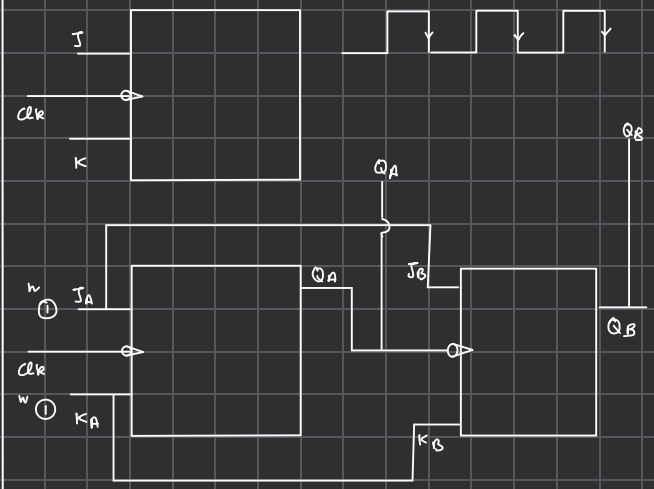
T - Flip flop

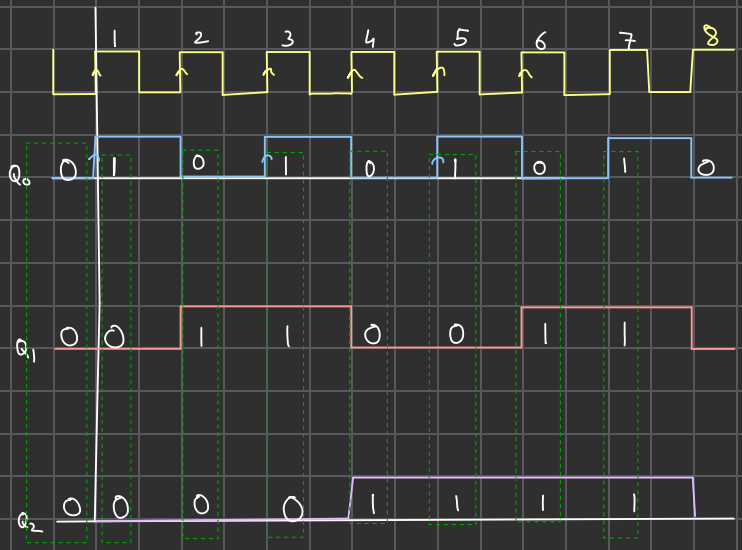


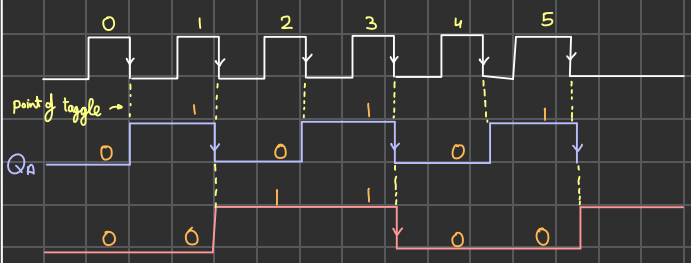


Asynchronous counter using T - Flip flop

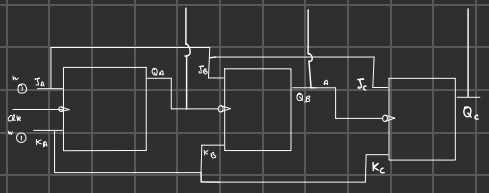
-ive edge clock →

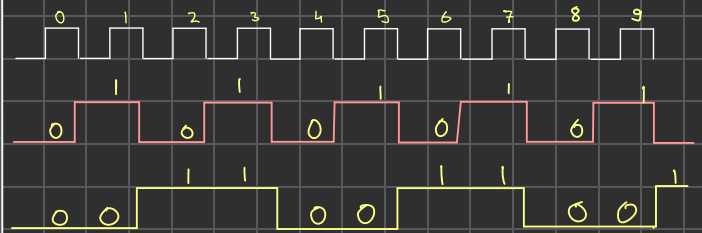






clk	Q_B	Q_A	Decimal
0	0	0	0
1	0	1	1
2	1	0	2
3	1	1	3
4	0	0	0
5	0	1	1

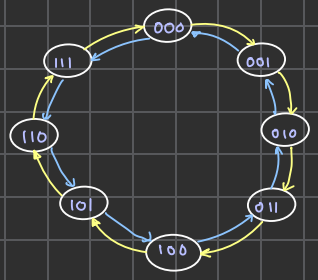




Clk	Q_C	Q_B	Q_A	Decimal
0	0	0	0	0
1	0	0	1	1
2	0	1	0	2
3	0	1	1	3
4	1	0	0	4
5	1	0	1	5
6	1	1	0	6
7	1	1	1	7
8	0	0	0	0
⋮	⋮	⋮	⋮	

Synchronous Counter

(i) up counter (ii) down counter (iii) up/down counter



Definitive T-T for J-K flip flop

T	Q_n	Q_{n+1}
0	0	1
1	1	0
0	1	0
1	0	1

Excitation Table

Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	1	0
1	0	1

Control Signal	Present Stage			Present Stage			I/p		
M	Q_C	Q_B	Q_A	Q_C	Q_B	Q_A	T_C	T_B	T_A
0 (up counter)	0	0	0	0	0	1	0	0	1
0	0	0	1	0	1	0	0	1	1
0	0	1	0	0	1	1	0	0	1
0	0	1	1	1	0	0	1	1	1
0	1	0	0	1	0	1	0	0	1
0	1	0	1	1	1	0	0	1	1
0	1	1	0	1	1	1	0	0	1
0	1	1	1	0	0	0	1	1	1
1 (down counter)	0	0	0	1	1	1	1	1	1
1	1	1	1	1	1	0	0	0	1
1	1	1	0	1	0	1	0	1	1
1	1	0	1	1	0	0	0	0	1
1	1	0	0	0	1	1	1	1	1
1	0	1	1	0	1	0	0	0	1
1	0	1	0	0	0	1	0	1	1
1	0	0	1	0	0	0	0	0	1

K- Maps

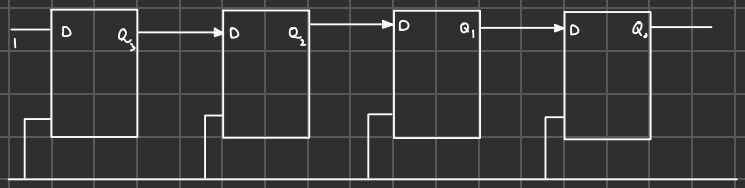
MQ_C \ Q_BQ_A	00	01	11	10
00	0	0	1	0
01	0	0	1	0
11	1	0	0	0
10	1	0	0	0

$$T_C = \bar{M} Q_A Q_B + M \bar{Q_A} \bar{Q_B}$$

Universal shift Registers : D flip flop

Modes	Loading	Reading	Total
SI SO	n	n-1	2n-1
PI PO	1	0	1

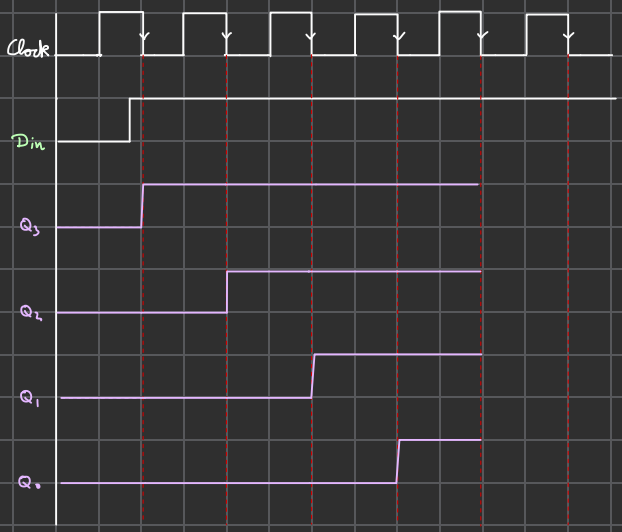
Serial In Serial Out :



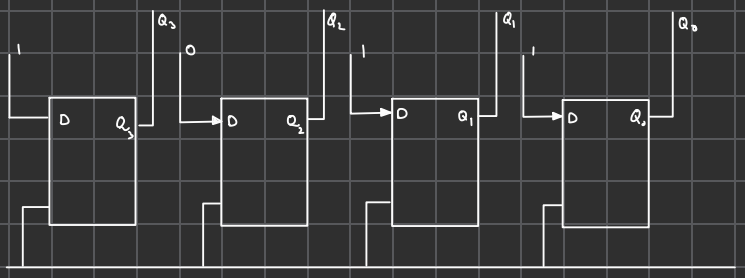
Ex 1 1 1 1

	Q_3	Q_2	Q_1	Q_0	
initial	0	0	0	0	
Clock 1	1	0	0	0	} n to load
Clock 2	1	1	0	0	
Clock 3	1	1	1	0	
Clock 4	1	1	1	1	← 1 output already present.
Clock 5	\	1	1	1	} n-1 to read
Clock 6	\	\	1	1	
Clock 7	\	\	\	1	

Din : 1 1 1 1

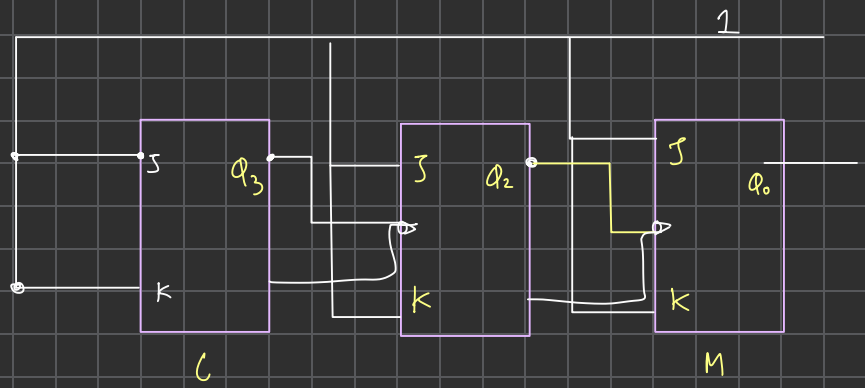


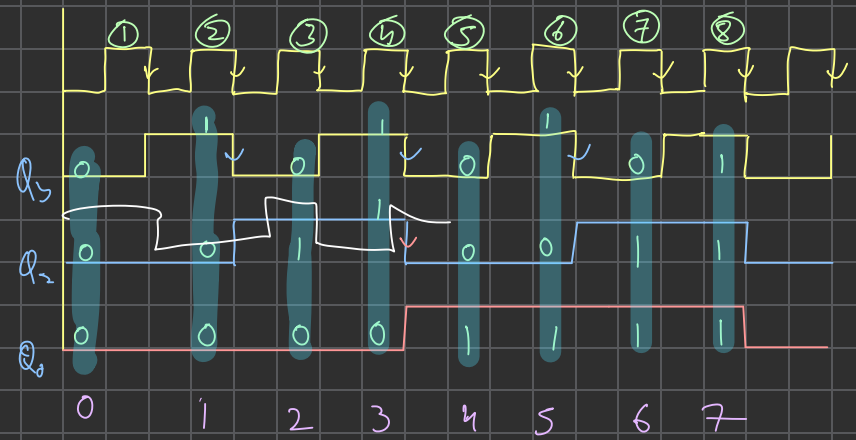
Parallel In Parallel Out :



Ex 1 0 1 1

	Q_3	Q_2	Q_1	Q_0
initial	0	0	0	0
Clock 1	1	0	1	1





clk	Q_0	Q_2	Q_3	
0	0	0	0	0
1	0	0	1	1
2	0	1	0	2
3	0	1	1	3
4	1	0	0	4
5	1	0	1	5
6	1	1	0	6
7	1	1	1	7
8	0	0	0	0
9	0	0	1	1
10	0	1	0	2