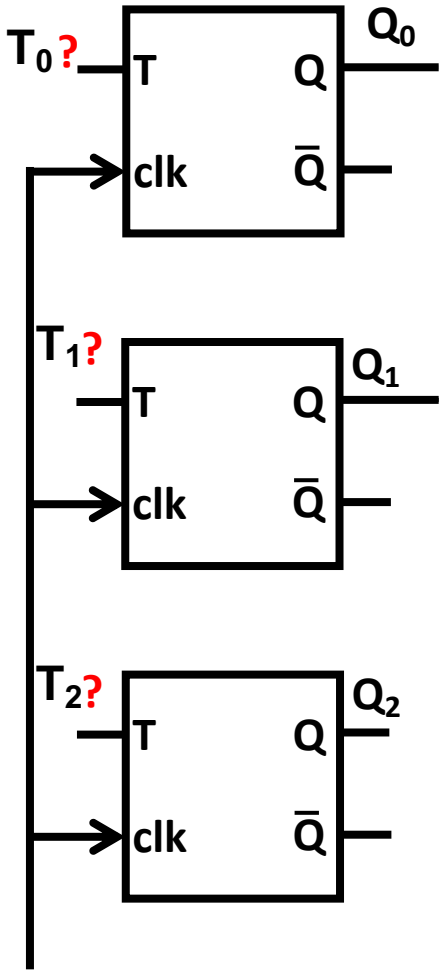
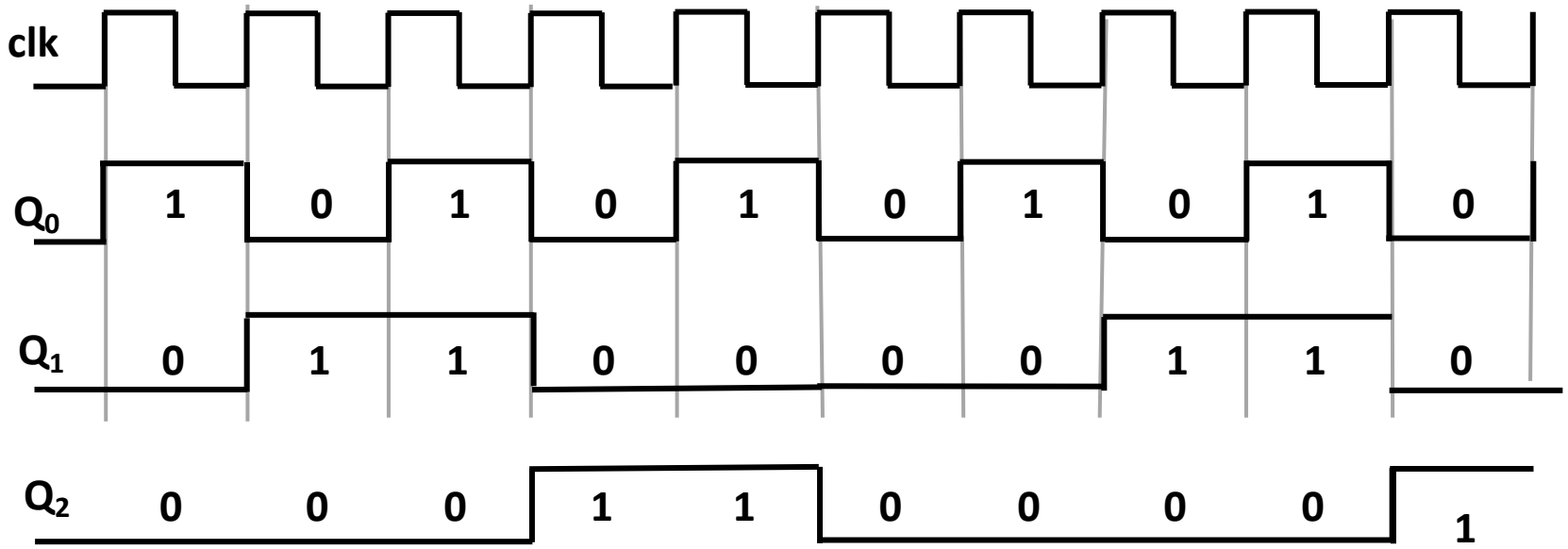
Synchronous Modulo-6 UP Counter using T-FF:





Modulo 6 Synchronous Counter using T-FF:

We require to count: 000 – 001 – 010 – 011 – 100 – 101 – 000 – 001 – 010 .....

State Transition Table:

PS	NS	Q_2 PS	Q_1 PS	Q_0 PS	Q_2 NS	Q_1 NS	Q_0 NS	T_2	T_1	T_0
000	001	0	0	0	0	0	1	0	0	1
001	010	0	0	1	0	1	0	0	1	1
010	011	0	1	0	0	1	1	0	0	1
011	100	0	1	1	1	0	0	1	1	1
100	101	1	0	0	1	0	1	0	0	1
101	000	1	0	1	0	0	0	1	0	1

Q_2Q_1 \ Q_0	0	1
00	0	0
01	0	1
11	ϕ	ϕ
10	0	1

T_2

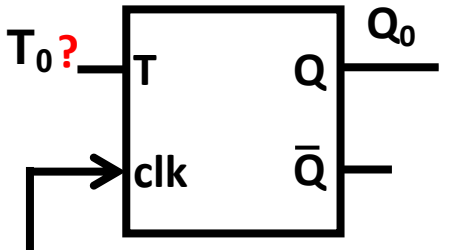
Q_2Q_1 \ Q_0	0	1
00	0	1
01	0	1
11	ϕ	ϕ
10	0	0

T_1

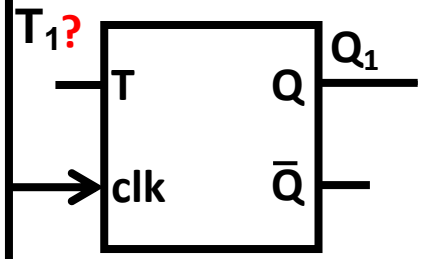
Q_2Q_1 \ Q_0	0	1
00	1	1
01	1	1
11	ϕ	ϕ
10	1	1

T_0

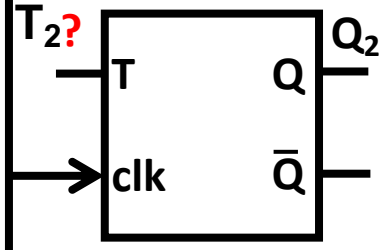
Synchronous Modulo-6 UP Counter using T-FF:



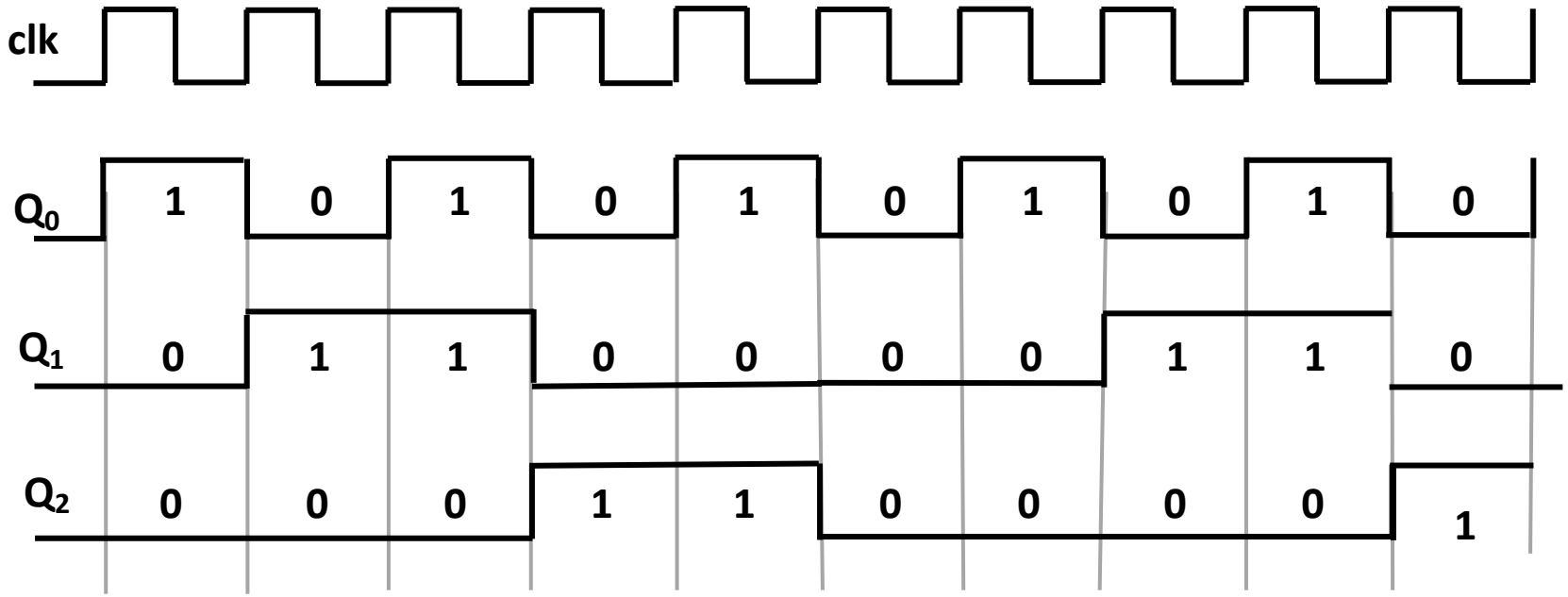
$T_0 = 1$



$T_1 = \overline{Q_2} Q_0$



$T_2 = Q_2 Q_0 + Q_1 Q_0$



Q_2Q_1 \ Q_0	0	1
00	0	0
01	0	1
11	ϕ	ϕ
10	0	1

T_2

Q_2Q_1 \ Q_0	0	1
00	0	1
01	0	1
11	ϕ	ϕ
10	0	0

T_1

Q_2Q_1 \ Q_0	0	1
00	1	1
01	1	1
11	ϕ	ϕ
10	1	1

T_0

Modulo 6 Synchronous Counter using D-FF:

We require to count: 000 – 001 – 010 – 011 – 100 – 101 – 000 – 001 – 010 .....

State Transition Table:

PS	NS	Q_2 PS	Q_1 PS	Q_0 PS	Q_2 NS	Q_1 NS	Q_0 NS	D_2	D_1	D_0
000	001	0	0	0	0	0	1	0	0	1
001	010	0	0	1	0	1	0	0	1	0
010	011	0	1	0	0	1	1	0	1	1
011	100	0	1	1	1	0	0	1	0	0
100	101	1	0	0	1	0	1	1	0	1
101	000	1	0	1	0	0	0	0	0	0

Q_2Q_1 \ Q_0	0	1
00	1	0
01	1	0
11	ϕ	ϕ
10	1	0

$D_0 = \overline{Q_0}$

Q_2Q_1 \ Q_0	0	1
00	0	1
01	1	0
11	ϕ	ϕ
10	0	0

$D_1 = Q_1\overline{Q_0} + \overline{Q_2}\,\overline{Q_1}Q_0$

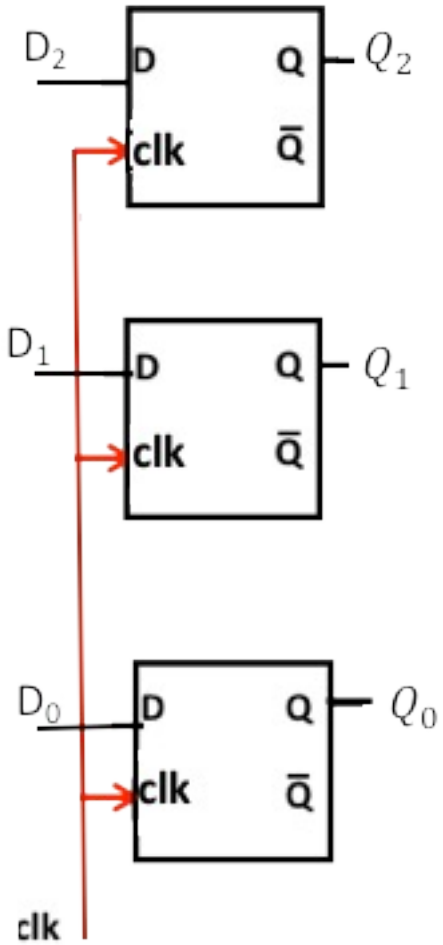
4. 3 i/p AND

Reduced ?

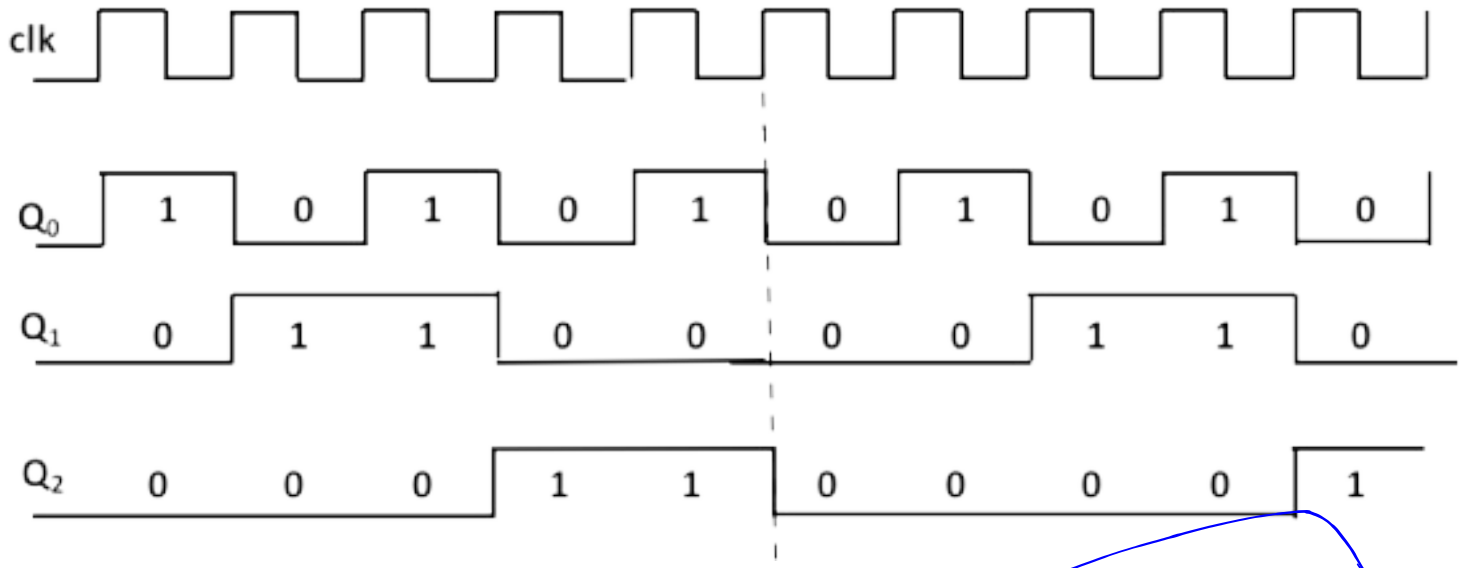
Q_2Q_1 \ Q_0	0	1
00	0	0
01	0	1
11	ϕ	ϕ
10	1	0

$D_2 = Q_2\overline{Q_0} + Q_1Q_0$

Modulo 6 Synchronous Counter using D-FF:



$$D_0 = \overline{Q_0} \qquad D_1 = Q_1\overline{Q_0} + \overline{Q_2}\,\overline{Q_1}Q_0 \qquad D_2 = Q_2\overline{Q_0} + Q_1Q_0$$



Programmable 3-bit Counter

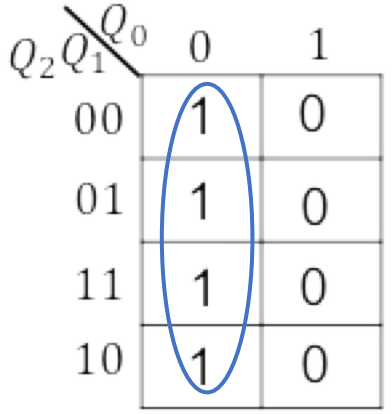
3 bit Counter ⇒ programmable to mod 5, 6, 7 or 8 ⇐ 3 bits.

Variable length 3-bit synchronous counter using DFF and Load:

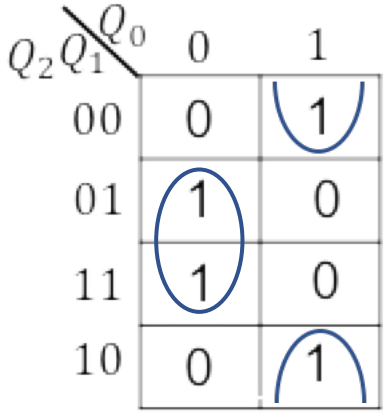
Step 1: Design of a 3-bit synchronous counter using DFF

Load ? 0 / Load = 000 / 100 ← Mod5, 101 — Mod6, 110 ← Mod7, 111 ← Mod8

PS	NS	Q_2 PS	Q_1 PS	Q_0 PS	Q_2 NS	Q_1 NS	Q_0 NS	D_2	D_1	D_0
000	001	0	0	0	0	0	1	0	0	1
001	010	0	0	1	0	1	0	0	1	0
010	011	0	1	0	0	1	1	0	1	1
011	100	0	1	1	1	0	0	1	0	0
100	101	1	0	0	1	0	1	1	0	1
101	110	1	0	1	1	1	0	1	1	0
110	111	1	1	0	1	1	1	1	1	1
111	000	1	1	1	0	0	0	0	0	0



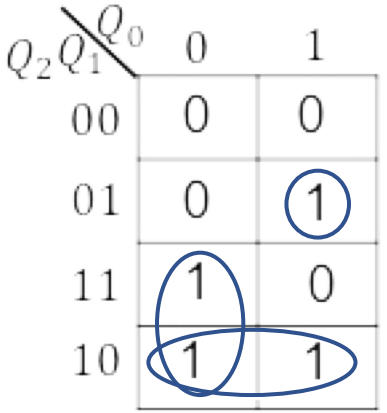
$$D_0 = \overline{Q_0}$$



3 gates

$$D_1 = Q_1\overline{Q_0} + \overline{Q_1}Q_0$$

$$= Q_1 \oplus Q_0$$ ← 1 gate



Q_1, $\overline{Q_1}$

4 gates

$$D_2 = Q_2\overline{Q_1} + Q_2\overline{Q_0} + \overline{Q_2}Q_1Q_0$$

$$= Q_2(\overline{Q_1} + \overline{Q_0}) + \overline{Q_2}Q_1Q_0$$

$$= Q_2 \oplus Q_1Q_0$$ ← 3 gates

Reduces the gates

Step 1: Design of a 3-bit synchronous counter using DFF

$$D_0 = \overline{Q_0}$$

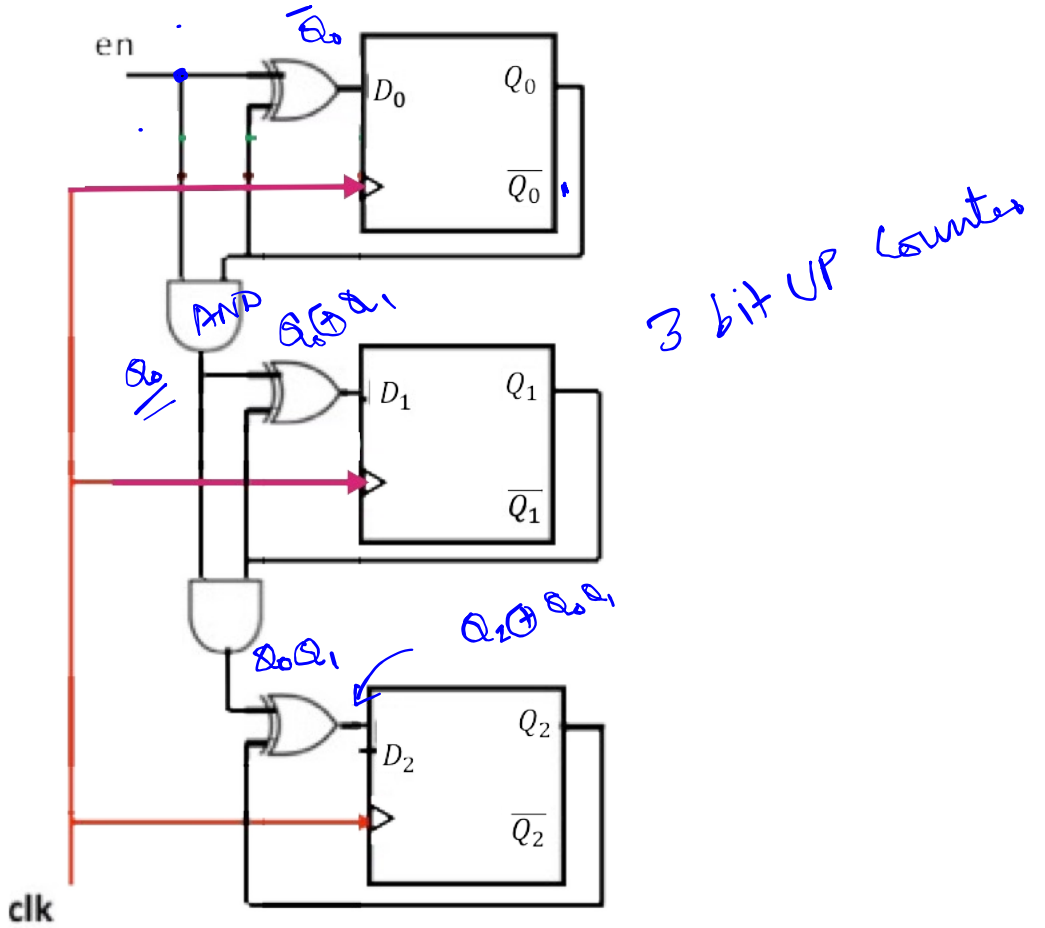
$$D_1 = Q_1\overline{Q_0} + \overline{Q_1}Q_0$$

$$= Q_1 \oplus Q_0$$

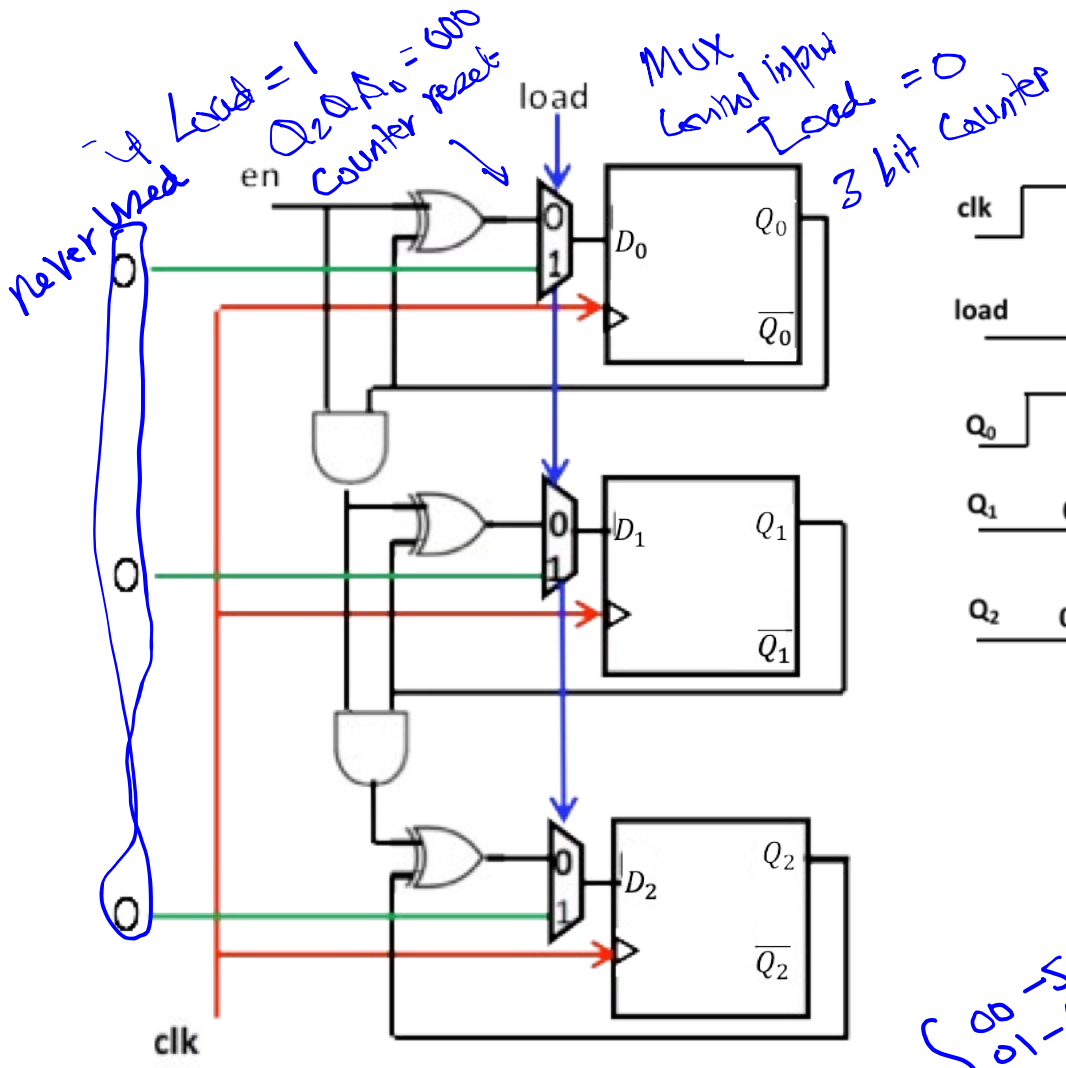
$$D_2 = Q_2\overline{Q_1} + Q_2\overline{Q_0} + \overline{Q_2}Q_1Q_0$$

$$= Q_2(\overline{Q_1} + \overline{Q_0}) + \overline{Q_2}Q_1Q_0$$

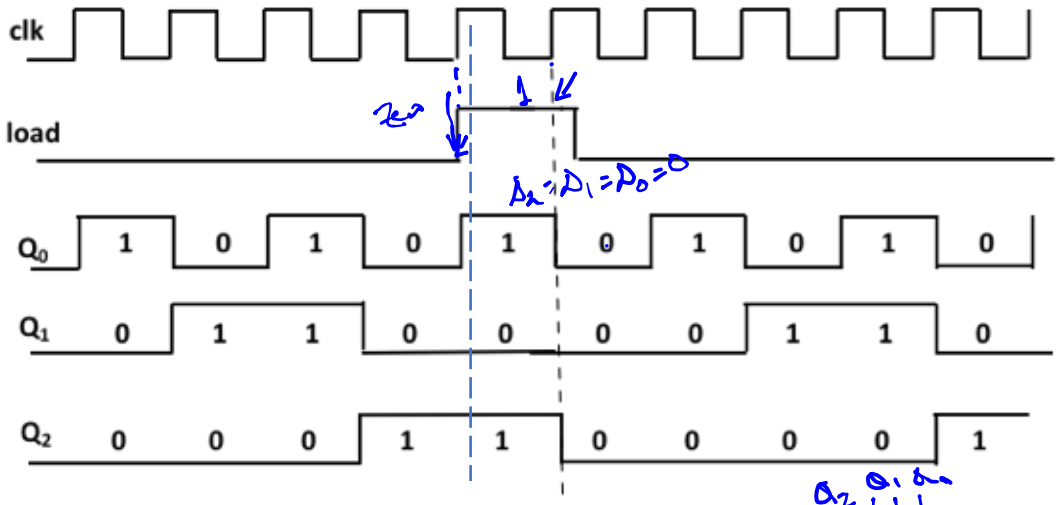
$$= Q_2 \oplus Q_1Q_0$$



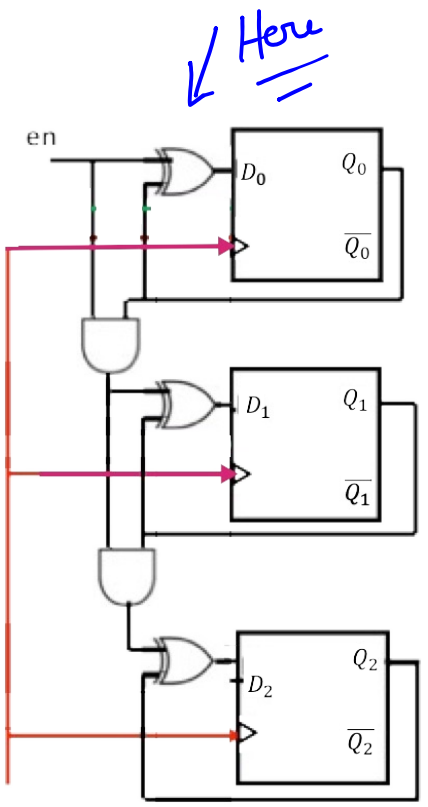
Step 2: Synchronous Modulo-6 UP Counter with Load using D-FF:



$$\text{Load} = Q_2\overline{Q_1}Q_0$$



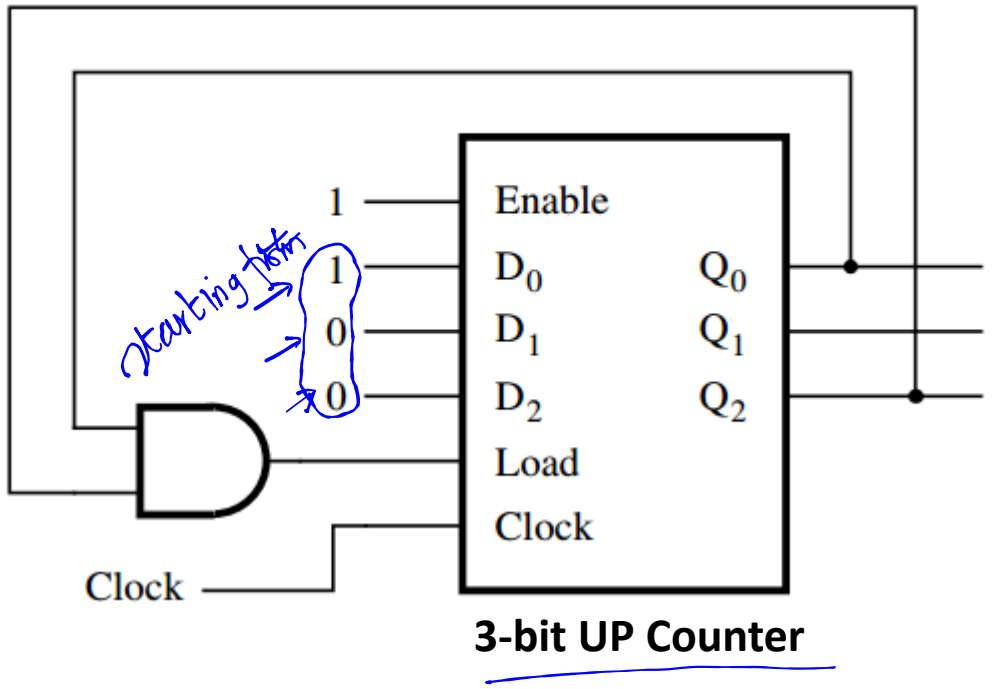
Load	Count
$Q_2\overline{Q_1}\,\overline{Q_0}$	Mod 5
$Q_2\overline{Q_1}Q_0$	Mod 6
$Q_2Q_1\overline{Q_0}$	Mod 7
0 or $Q_2Q_1Q_0$	Mod 8



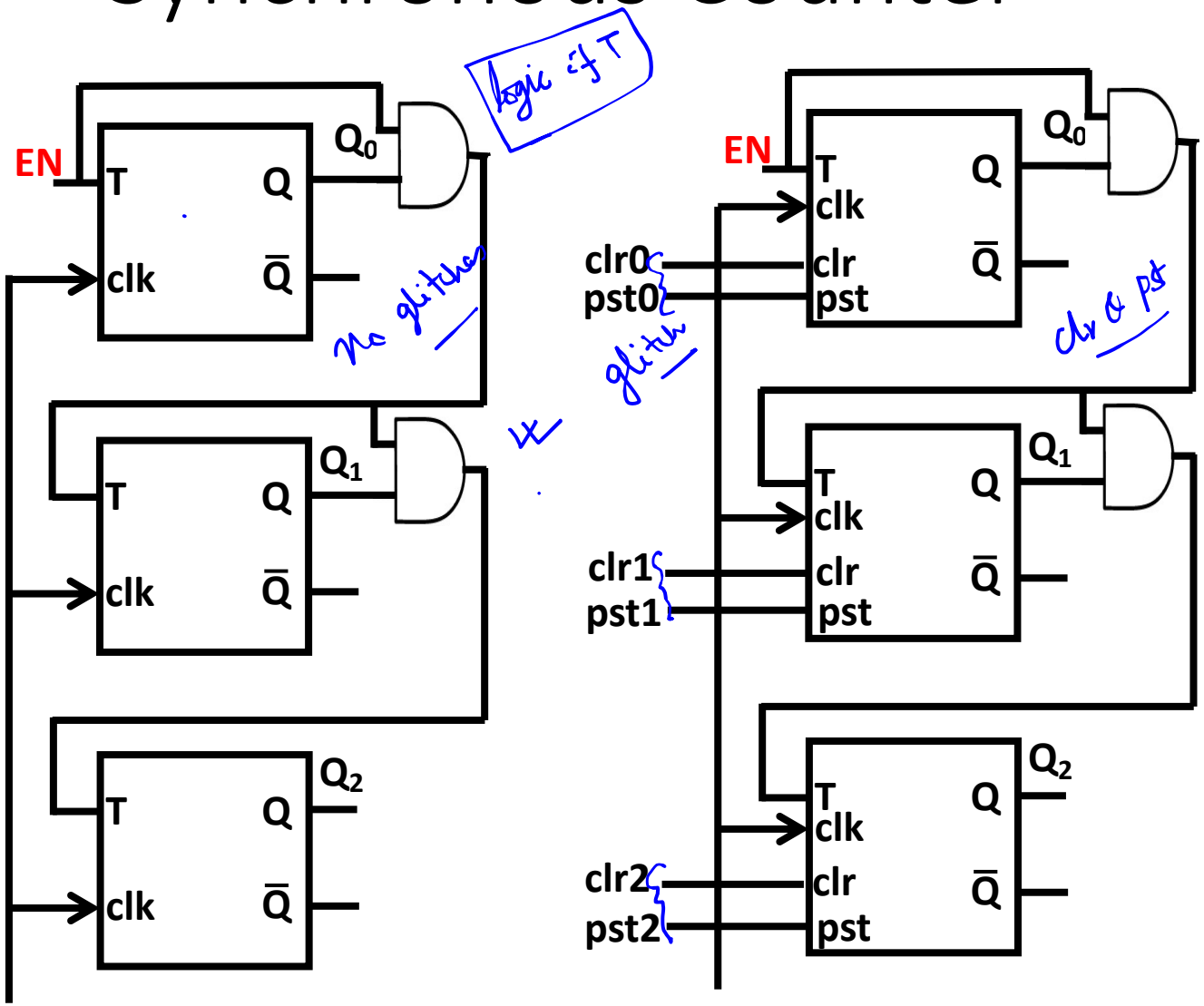
When Load = 0
3-bit counter

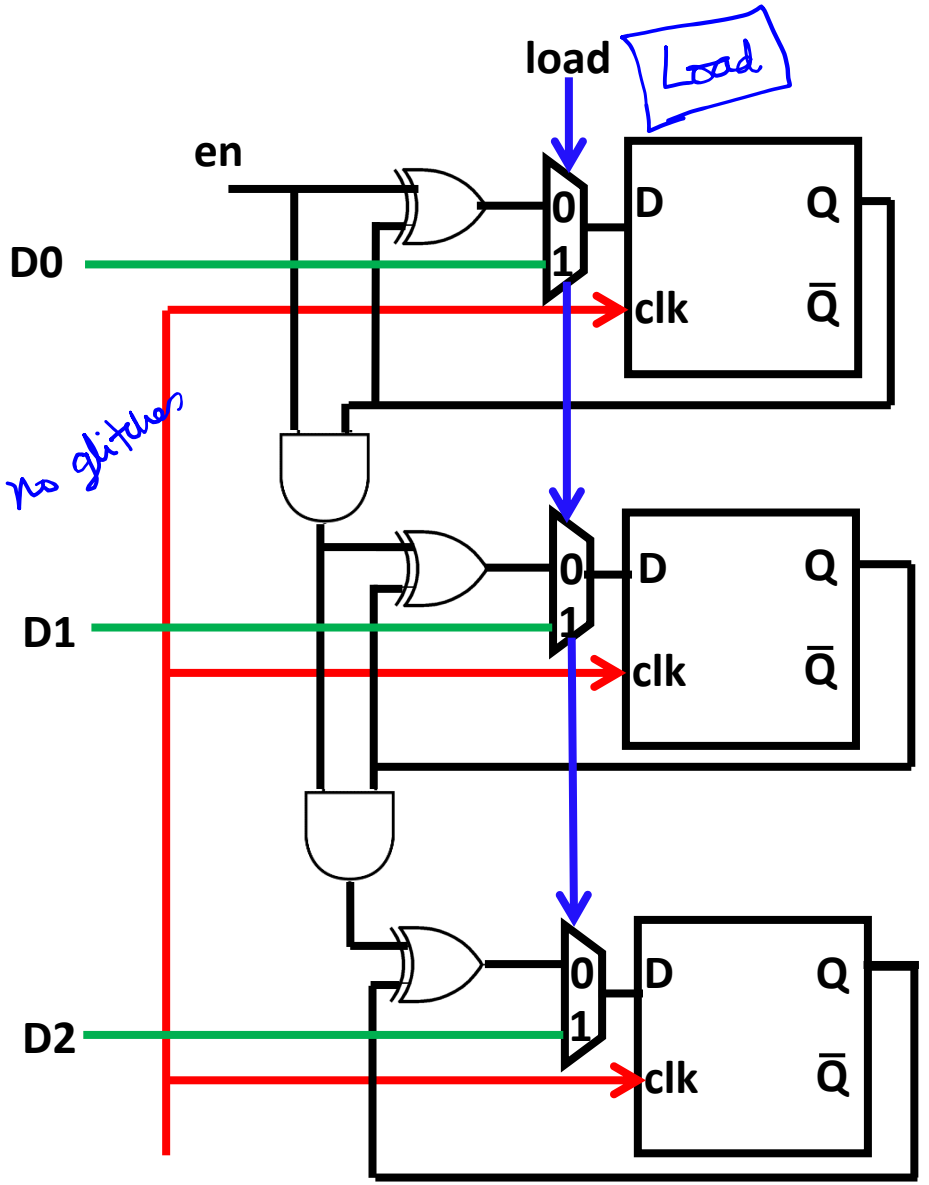
HW

- Draw the timing diagram of the following circuit

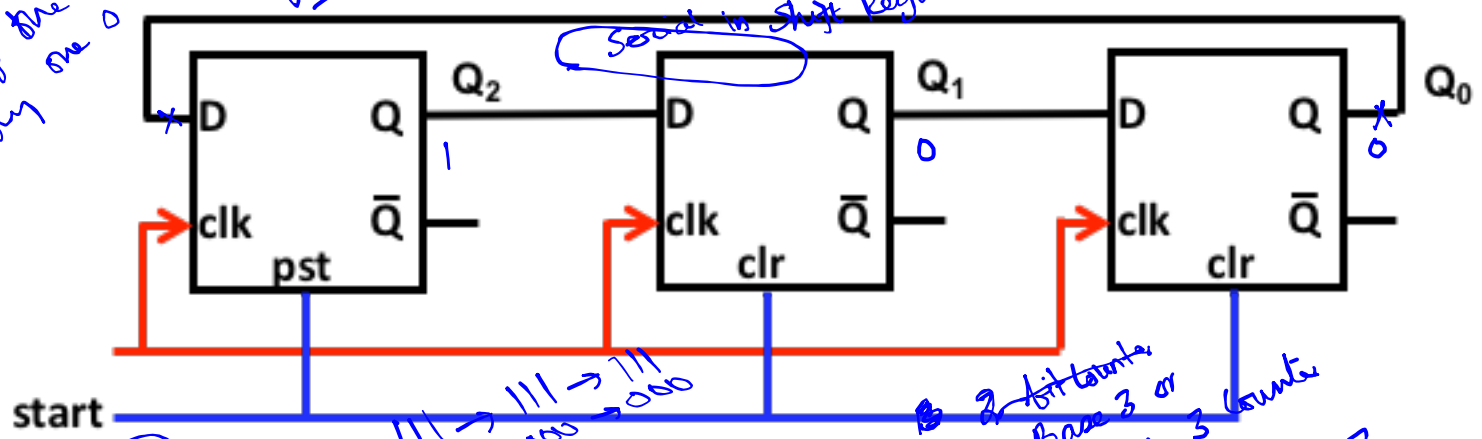


Synchronous Counter





Ring Counter:



Q_2	Q_1	Q_0
1	0	0
0	1	0
0	0	1
1	0	0
0	1	0
0	0	1

N-FF -> N Pattern generator

000 & 111 are forbidden

SR = 11 is forbidden

2N patterns with 3 FF

Realize this

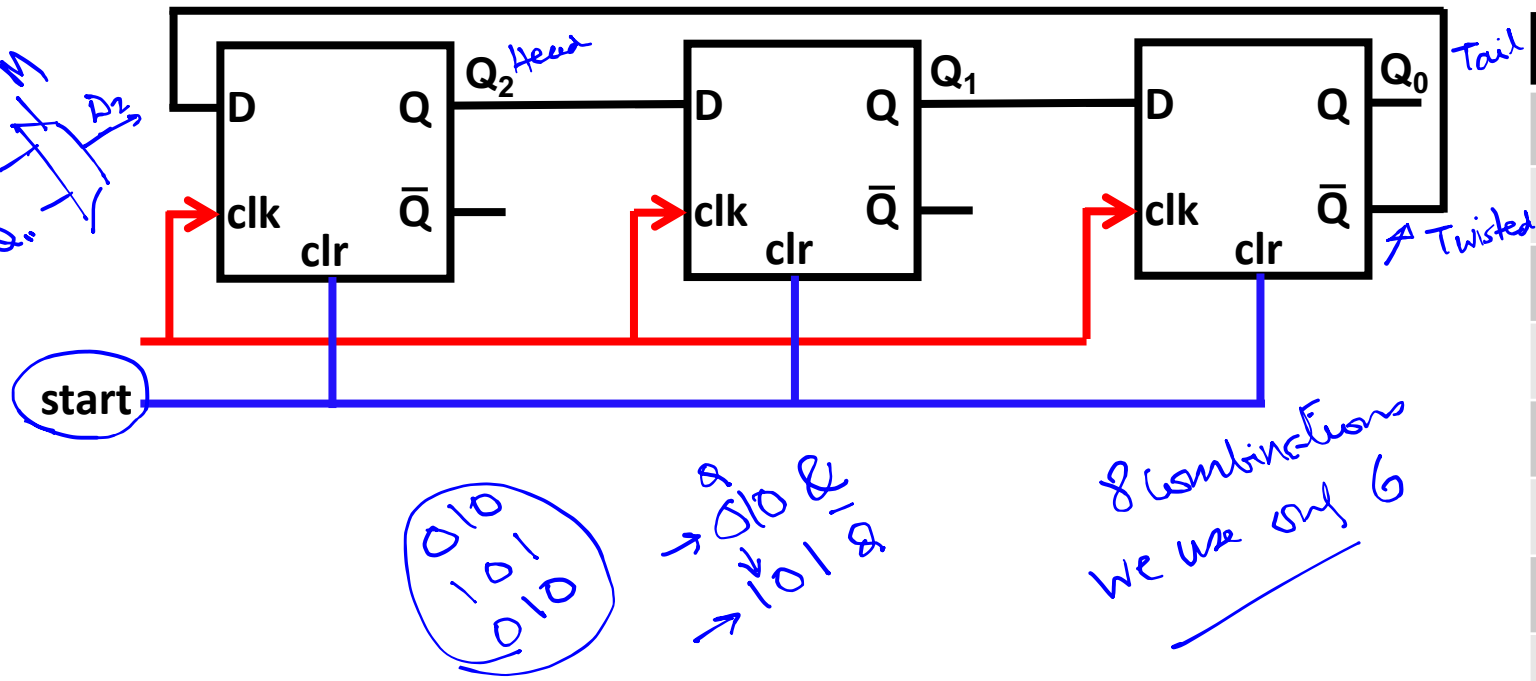
$Q_2 = \overline{Q_0}$

100 → 110 → 111 → 011 → 001 → 000 → 100

6 patterns with 3 FF

Johnson Counter (Twisted Tail Ring Counter):

N-FF -> 2*N* Pattern generator



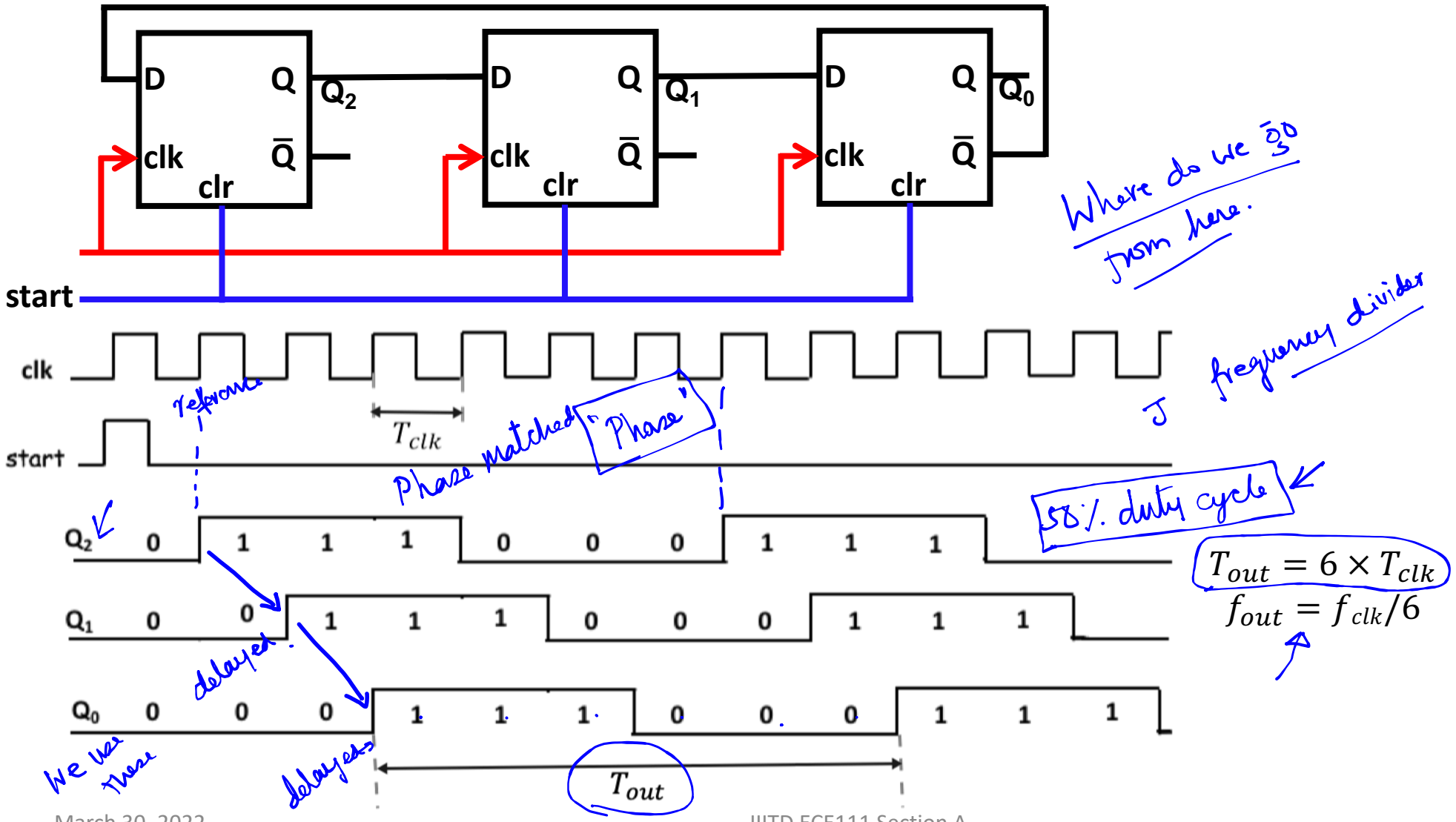
Q_2		Q_1	Q_0
0	0	0	0
1	1	0	0
1	2	1	0
1	3	1	1
0	4	1	1
0	5	0	1
0		0	0
1		0	0
1		1	0

forbidden

Check what happens when FFs are initialised to 010 or 101 or 100

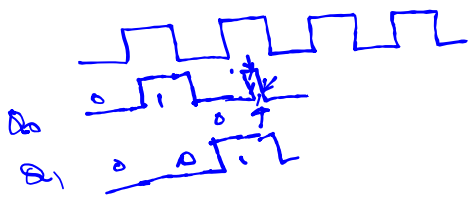
ensured

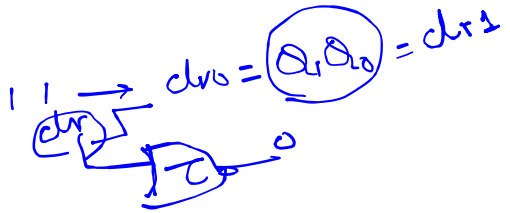
Johnson Counter as Frequency Divider:



Q_2	Q_1	Q_0
0	0	0
1	0	0
1	1	0
1	1	1
0	1	1
0	0	1
0	0	0
1	0	0
1	1	0
1	1	1
0	1	1
0	0	1
0	0	0
1	0	0
1	1	0
1	1	1

What Next…





- Ripple counters have limitation of clock frequency. Synchronous counters do not have that limitation.
- Modulo-K counters with asynchronous preset/clear has glitch which may not be acceptable in some applications. One approach is to use D-FF with parallel load capability (Explore and get back in the next class)
- How to design circuits/counters with variable steps: 2->4-> 5->7->2
- How to design circuits which can generate any pattern?
- How to design circuits/counters for clock division other than power-of-two
- How to design frequency division circuit which can generate desired clock frequency with control over the duty cycle?