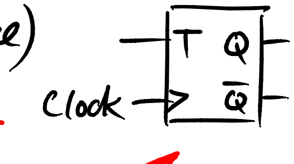
5.5, 7-9, 12 T-ff and counters

Counting in binary

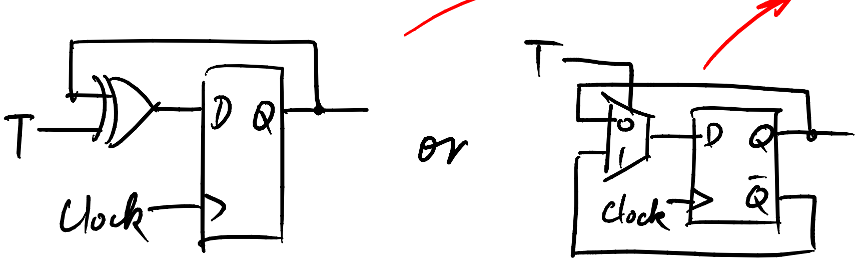
```
0 0 0
0 0 1
0 1 0
0 1 1
1 0 0
1 0 1
  ⋮
```

We want to build a toggle flip-flop (T-ff)

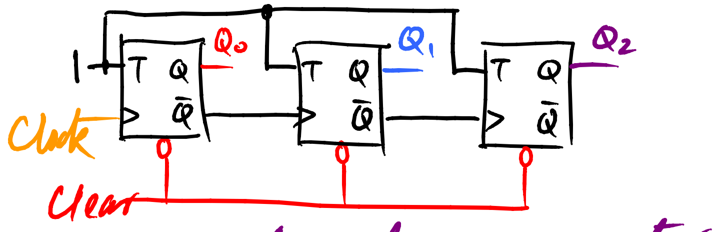
T	Q(t+1)
0	Q(t) (no change, hold old value)
1	$\overline{Q}(t)$ (toggle)



start with a D-ff



make a 3-bit up counter out of T-ff

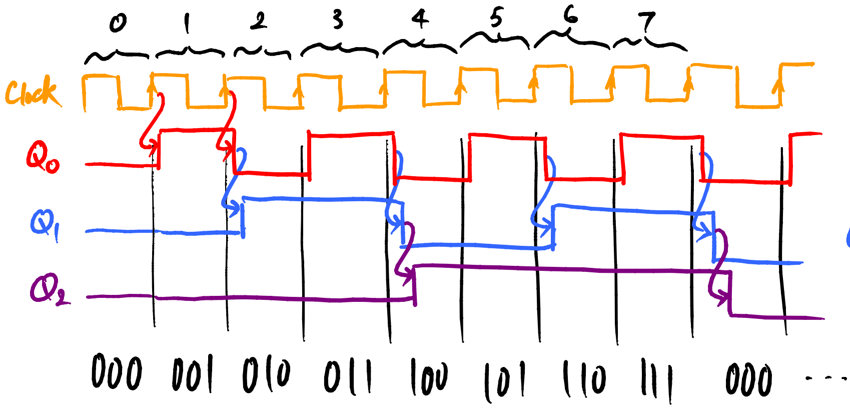


Asynchronous counter

Counting sequence

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

Note: Q_0 toggles at ½ clock frequency.
Q_1 toggles at ½ of Q_0's frequency
Q_2 toggles at ½ of Q_1's frequency

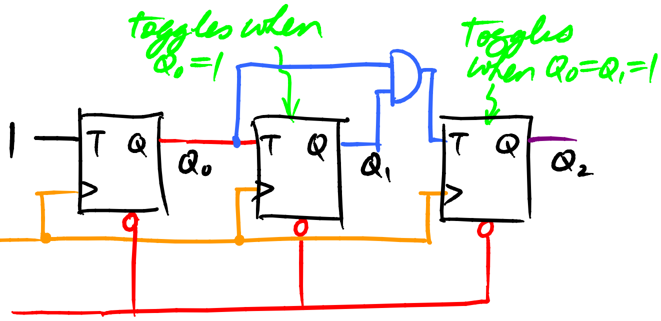


Q_0 toggles at the ↑ of clock

Q_1 toggles at the ↓ of Q_0

Q_2 toggles at the ↓ of Q_1

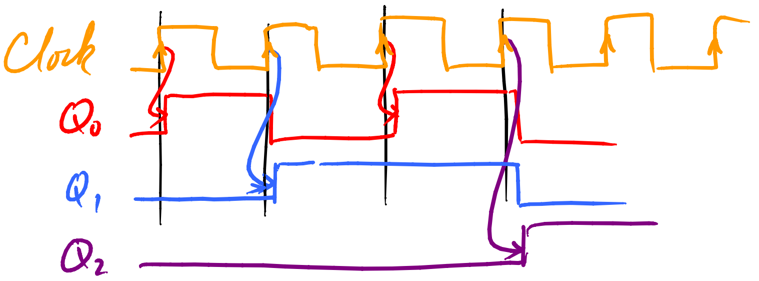
Synchronous up-counter



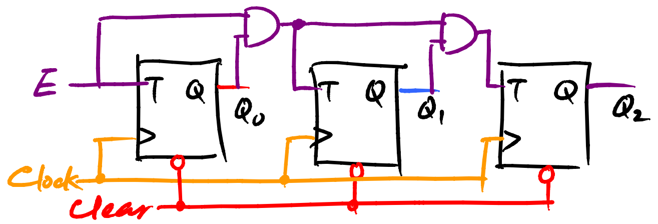
Clock cycle	Count ($Q_2Q_1Q_0$)
0	0 0 0
1	0 0 1
2	0 1 0
3	0 1 1
4	1 0 0
5	1 0 1
6	1 1 0
7	1 1 1
8	0 0 0 Repeats

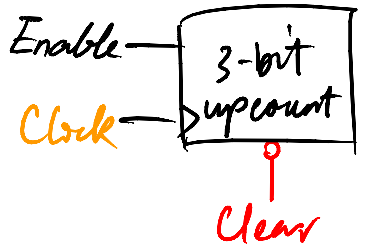
Timing Diagram of a synchronous up-counter

all Q's change values only at the Clock ↑

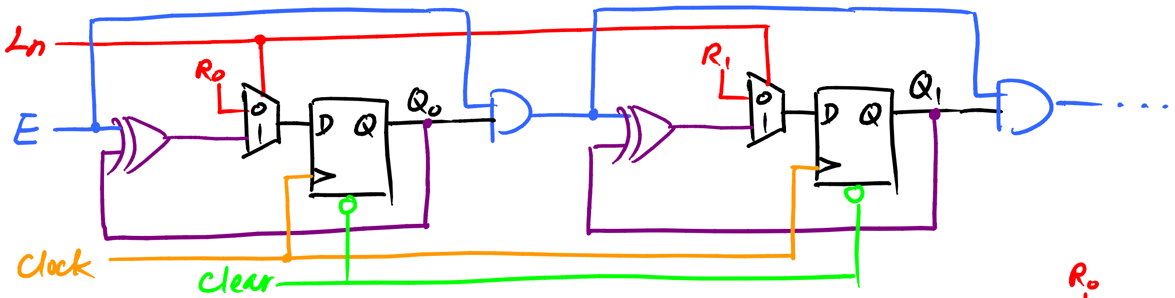


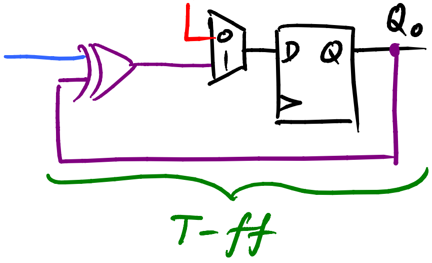
→ Add an Enable(E) input to the synchronous up counter



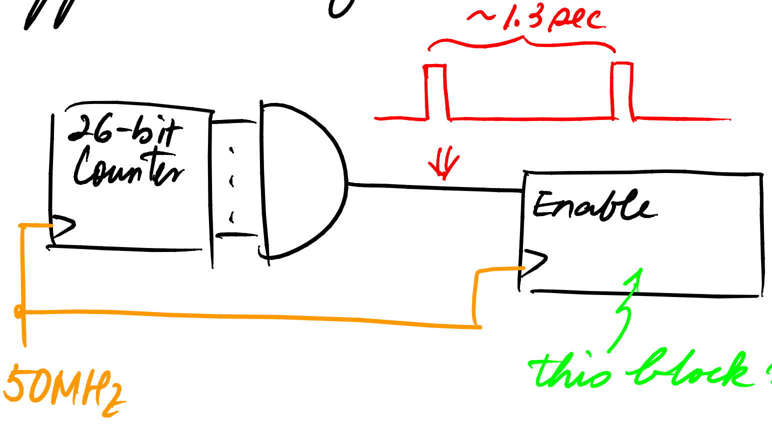


→ Add a parallel load (L_n) input to the synchronous up counter (for convenience, show D-ff in the circuit diagram)





Application of counters



$$(2^{26}-1)\frac{1}{50\times10^6} \simeq 1.3 \text{ sec.}$$

this block will be enabled every 1.3 sec.