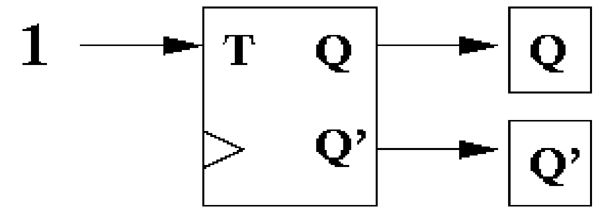
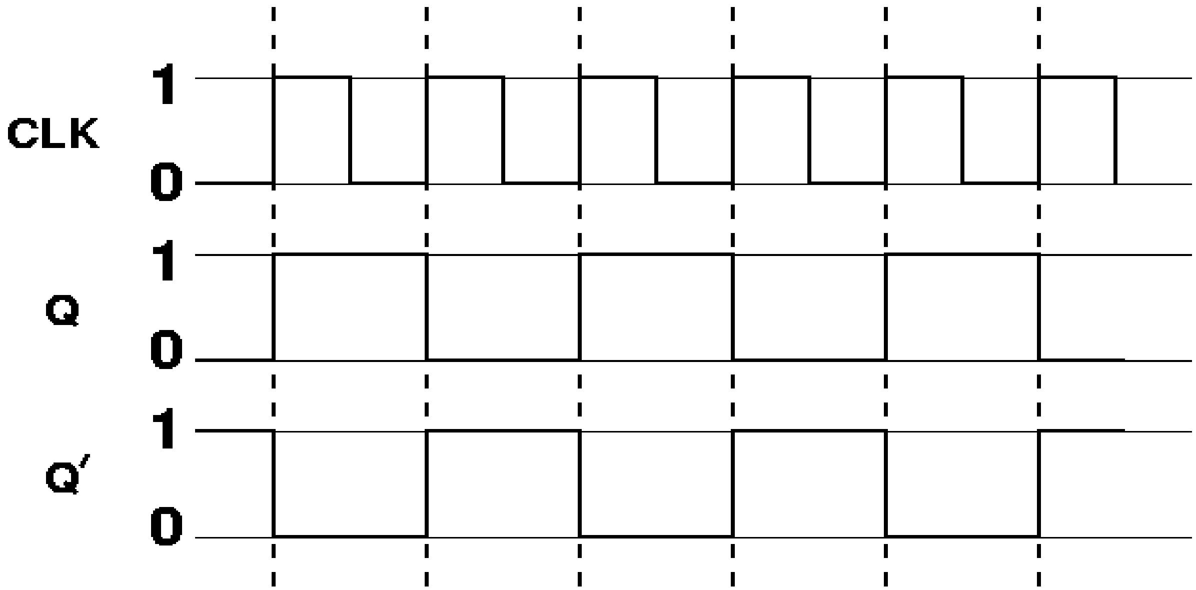
# Counter

**Counter increments an unsigned binary value from 0 to N**
**Consider a T flip-flop with hardwired input of 1:**



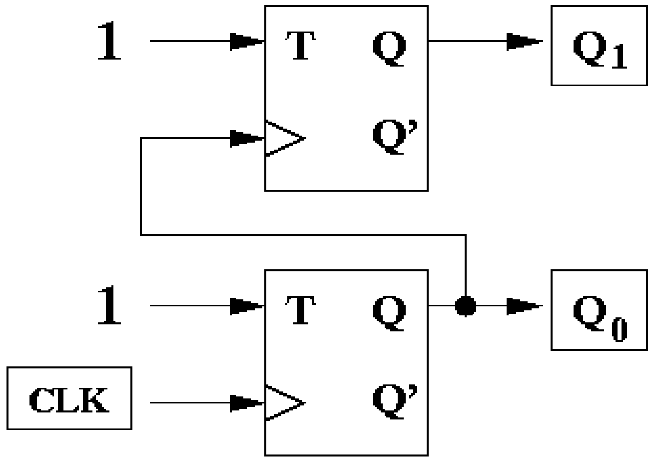

**The behavior can be represented by a timing diagram:**




**Value of Q toggles at each positive clock edge.**
**Notice that if the clock period is t, the period of the output Q**
**is exactly double the clock period, or 2t.**

# Counter

**Now use the output of the first flip-flop as the clock input of another T flip-flop:**




**What will be the period of the second flip-flop output $Q_1$?**

**If we keep repeating this N times, the period of the Nth output will be $2^N t$**

**How does this help build a counter?**

## Counter

**Consider what it means to count in binary:**

| $x_2$ | $x_1$ | $x_0$ | value |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 2 |
| 0 | 1 | 1 | 3 |
| 1 | 0 | 0 | 4 |
| 1 | 0 | 1 | 5 |
| 1 | 1 | 0 | 6 |
| 1 | 1 | 1 | 7 |

**The sequence of $x_0$ values looks like a clock with period 1: 0 1 0 1 0 1 0 1**

**The sequence of $x_1$ values looks like a clock with period 2: 0 0 1 1 0 0 1 1**

**The sequence of $x_2$ values looks like a clock with period 4: 0 0 0 0 1 1 1 1**

**However, notice when $x_1$ changes relative to $x_0$:**

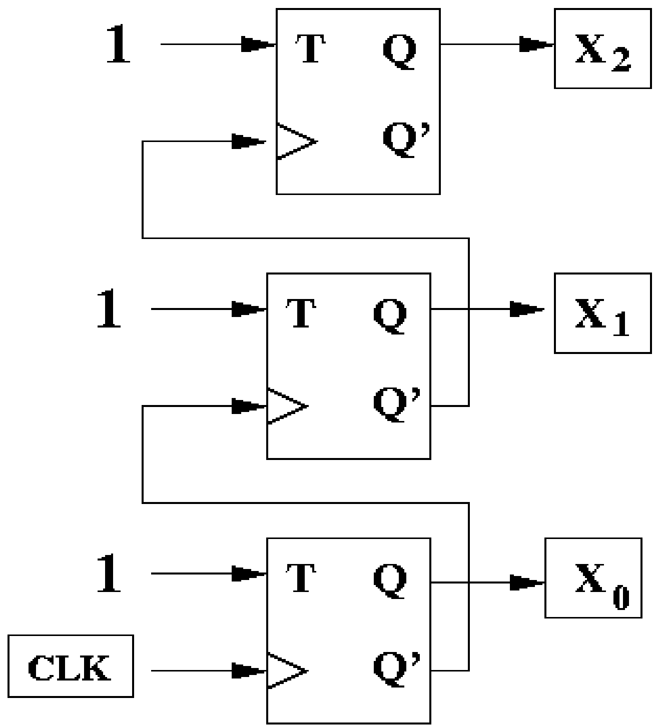
**$x_1$ goes from 0 to 1 (for example, value 1 to 2) when $x_0$ goes from 1 to 0**

**$x_1$ goes from 1 to 0 (for example, value 3 to 4) when $x_0$ goes from 1 to 0**

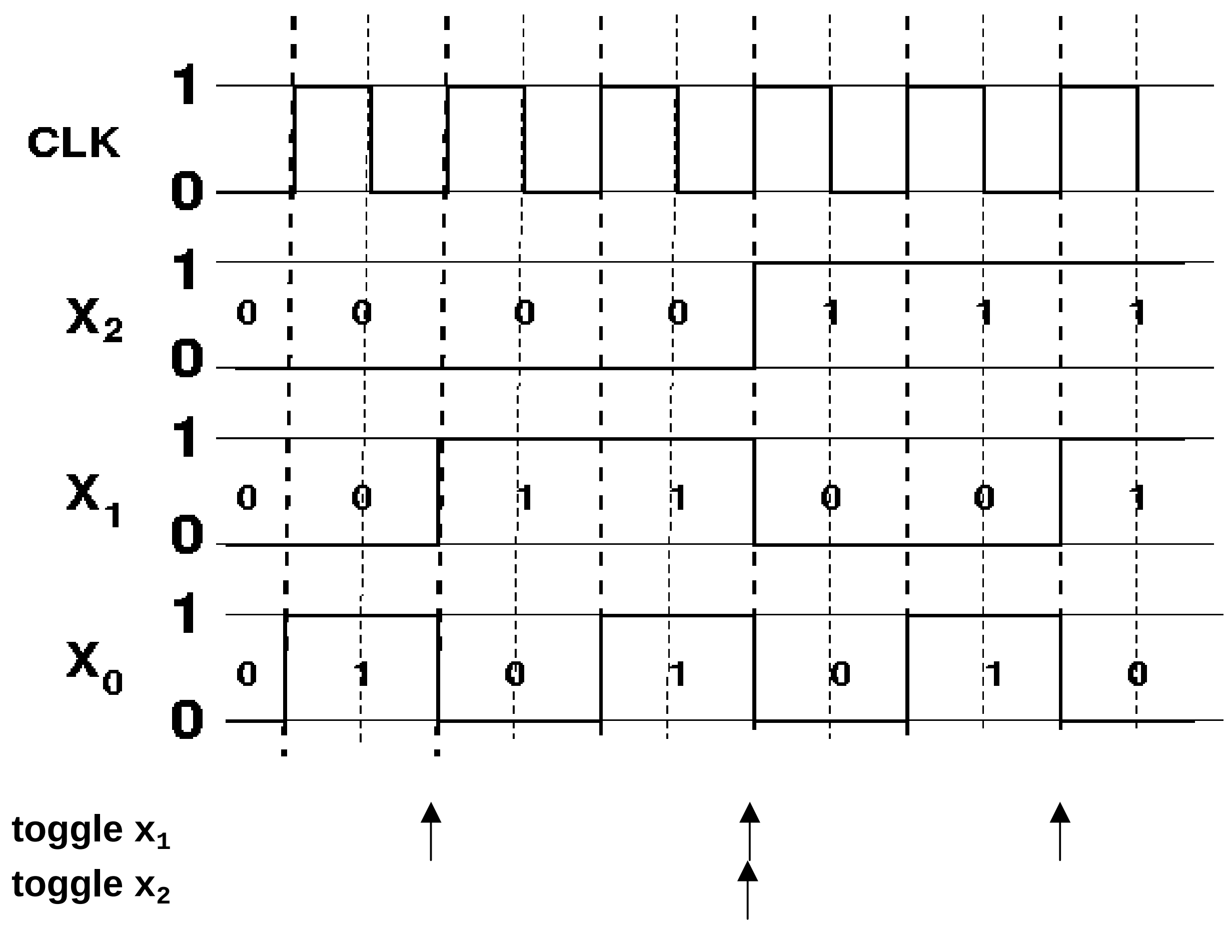
**This means that we need to toggle $x_1$ when $x_0$ is on a negative edge,**
**but we want to use positive-edge flip-flops.**

## Counter

**Toggling $Q_1$ on a negative edge of $Q_0$ is the same as toggling $Q_1$ on a positive edge of $Q_0$',**
**so connect the negated output Q' of each flip flop to the input of the next flip-flop:**



Timing diagram for this 3-bit counter:




Read values in each column from left to right:

```
000, 001, 010, . . .
```

Variation: how would we use D flip-flops instead of T flip-flops?