

COUNTERS

Why do we need counters?

Counters in digital circuits may be used for 3 functions:

- **Timing:** Building a precision digital clock is an example where a low frequency (10 Hz) clock cannot be achieved with a crystal oscillator.
- **Sequencing:** Starting of a rocket motor is an example where the energizing of fuel pumps, ignition, etc. must follow a critical sequence.
- **Counting:** Measuring the flow of traffic on a road is an application in which the total number of vehicles passing a certain point needs to be counted.

COUNTERS (continued)

A counter is a register that goes through a sequence of states.

Counter categories:

1. Ripple counters
 2. Synchronous counters
- **Ripple counters:** The flip-flop's output transition triggers other flip-flops.
 - **Synchronous counters:** A common clock triggers all flip-flops simultaneously rather than one at a time in succession as in ripple counters.

BINARY RIPPLE COUNTER

- A binary ripple counter consists of a series connection of complementing flip-flops → the output of each flip-flop is connected to the *C* input of the next higher-order flip-flop.

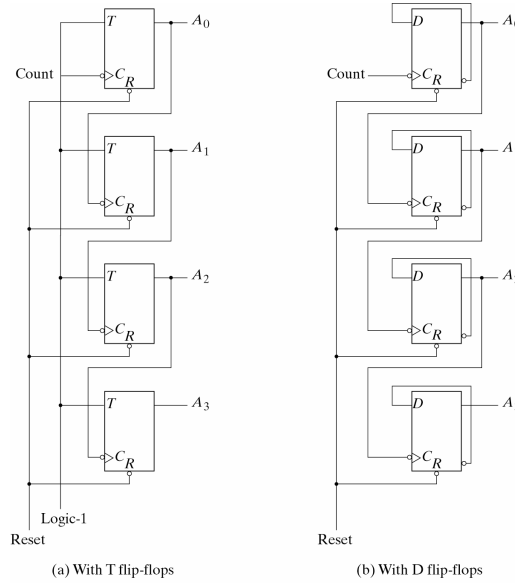
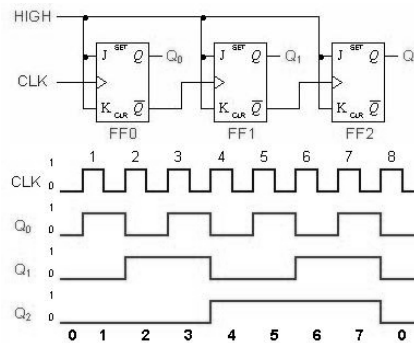


Fig. 6-8 4-Bit Binary Ripple Counter

BINARY RIPPLE COUNTER

Q_3	Q_2	Q_1	Q_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0



Q_0 is complemented with the count pulse. Since Q_0 goes from 1 to 0, it triggers Q_1 and complements it. As a result, Q_1 goes from 1 → 0, which in turn complements Q_2 changing it from 0 → 1. Q_2 does not trigger Q_3 because Q_2 produces a positive transition. The flip-flops change one bit at a time in succession and the signal propagates through the counter in a ripple fashion from one stage to the next.

PROBLEMS WITH RIPPLE COUNTERS

- *Asynchronous or ripple counters* are arranged in such a way that the output of one flip flop changes the state of the next. In a long chain of ripple counter stages, the last flip flop changes its state considerably later than the first FF due to propagation delays in each stage. Problems occur if this delay is longer than the response time of other logic elements connected to the circuit.
- *Synchronous counters* overcome the problems of propagation delay and erroneous intermediate states. In this type of counter all the FF clock inputs are wired together, so the transitions of all stages occur simultaneously.

SYNCHRONOUS COUNTERS

Synchronous counters are different from ripple counters in that the clock is applied to the inputs of all flip-flops, which triggers all flip-flops at the same time.

If $T = 0$ or $J = K = 0$, the flip-flop does not change state.

If $T = 1$ or $J = K = 1$, the flip-flop complements.

Suppose for a 4-bit counter $A_3A_2A_1A_0 = 0011$, the next count is 0100.

- A_0 is always complemented.
- A_1 is complemented because the present state of $A_0 = 1$.
- A_2 is complemented because the present state of $A_1A_0 = 11$.
- A_3 is not complemented because the present state of $A_2A_1A_0 = 011$.

4-BIT SYNCHRONOUS COUNTER

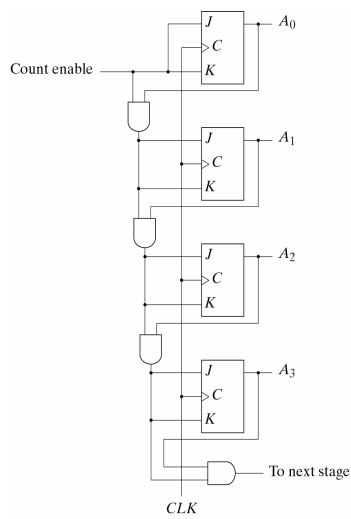


Fig. 6-12 4-Bit Synchronous Binary Counter

Flip-Flop Excitation table:

$Q(t)$	$Q(t+1)$	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

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

- If the enable is 0 and, all J and K inputs are 0 and the clock does not change the state of counter.
- The first stage A_0 has its J and $K = 1$ if enable = 1.
- The other J and K are equal to 1 if all previous least significant stages are equal to 1. The chain of AND gates generates the required logic for the J and K inputs in each stage.

Note that Synchronous counters have a regular pattern.