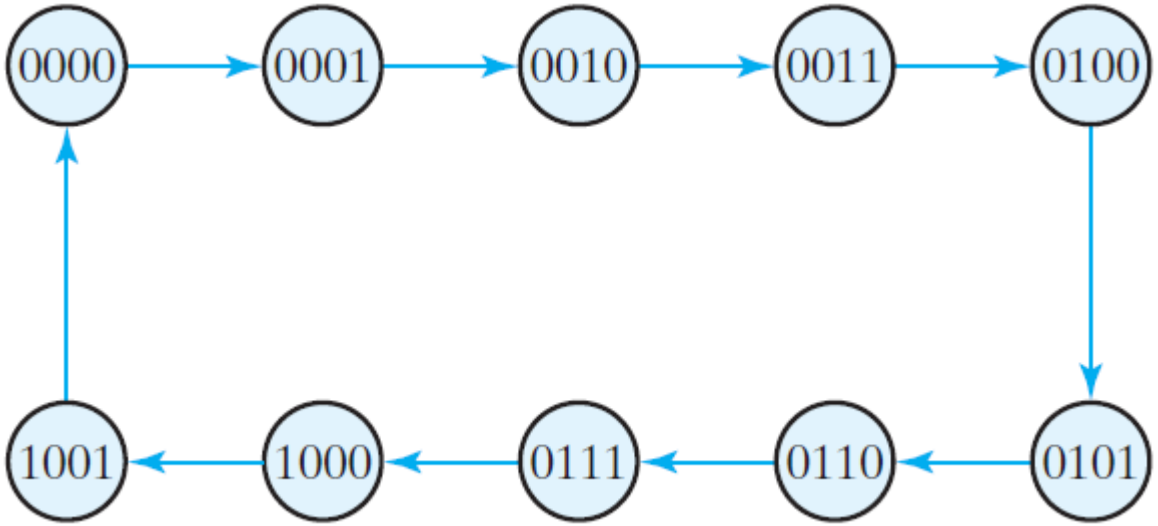
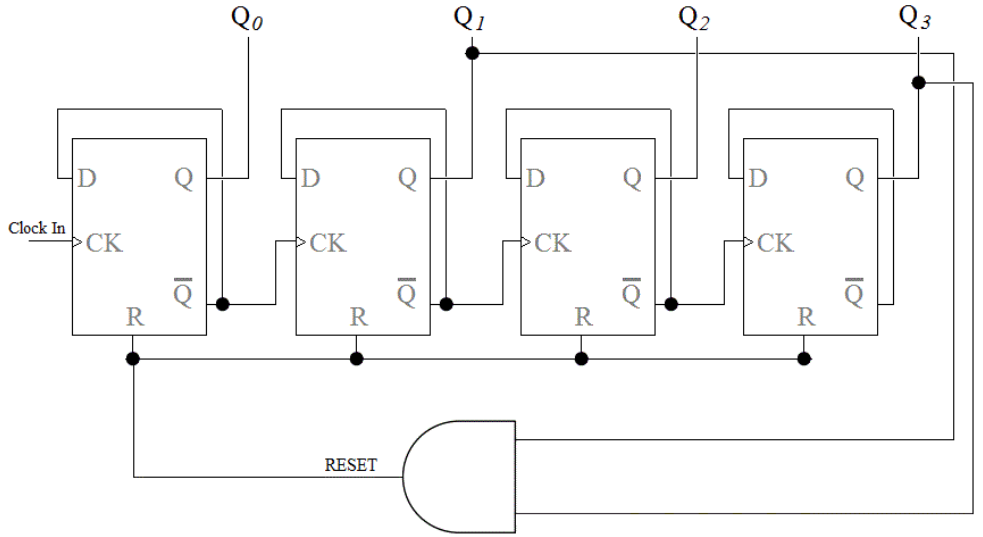
Lecture 22 – Registers and Counters 3

Chapter 6

Ripple counter - BCD

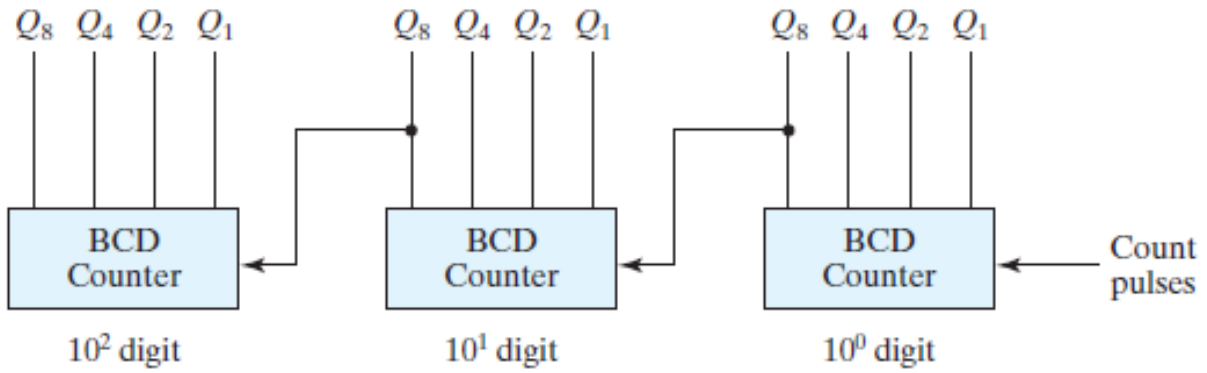
- A decimal counter follows a sequence of 10 states and returns to 0 after the count of 9
- Such a counter must have at least four flip-flops to represent each decimal digit, since a decimal digit is represented by a binary code with at least four bits
- The sequence of states in a decimal counter is dictated by the binary code used to represent a decimal digit
- A decimal counter is similar to a binary counter, except that the state after 1001 (the code for decimal digit 9) is 0000 (the code for decimal digit 0)





Ripple counter - BCD

- A decade counter counts from 0 to 9
- To count in decimal from 0 to 99, we need a two-decade counter
- To count from 0 to 999, we need a three-decade counter
- Multiple decade counters can be constructed by connecting BCD counters in cascade, one for each decade
- The inputs to the second and third decades come from Q_8 of the previous decade
- When Q_8 in one decade goes from 1 to 0, it triggers the count for the next higher order decade while its own decade goes from 9 to 0

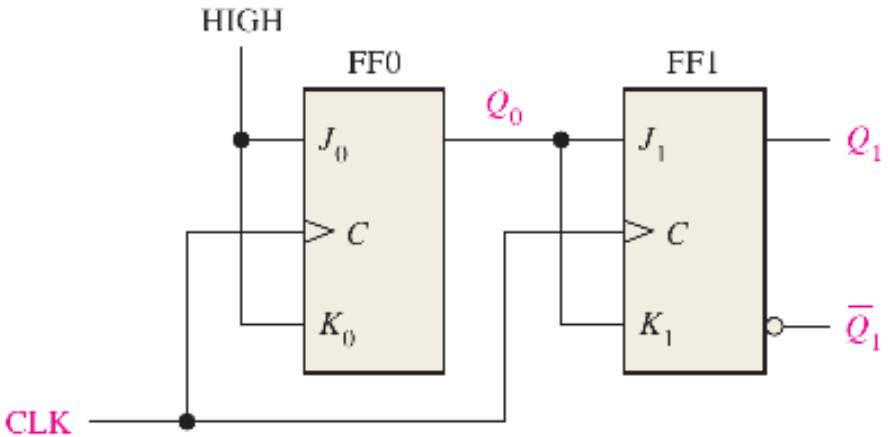


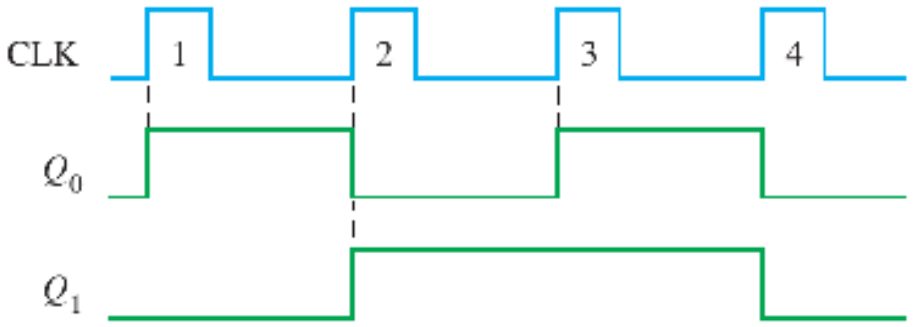
Synchronous counter - binary

- Synchronous counters are different from ripple counters in that clock pulses are applied to the inputs of all flip-flops
- A common clock triggers all flip-flops simultaneously, rather than one at a time in succession as in a ripple counter
- The decision whether a flip-flop is to be complemented is determined from the values of the data inputs, such as *T* or *J* and *K* at the time of the clock edge
- 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

Synchronous counter - binary

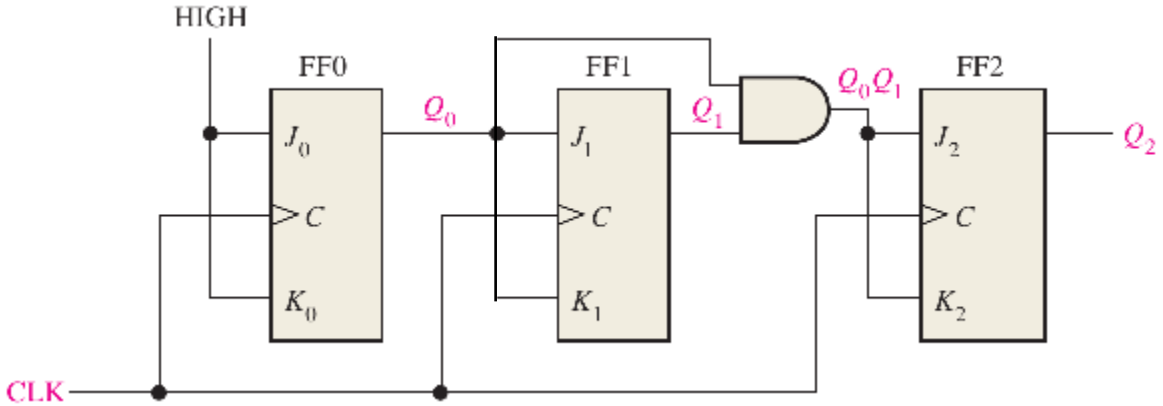
2-bit synchronous binary counter

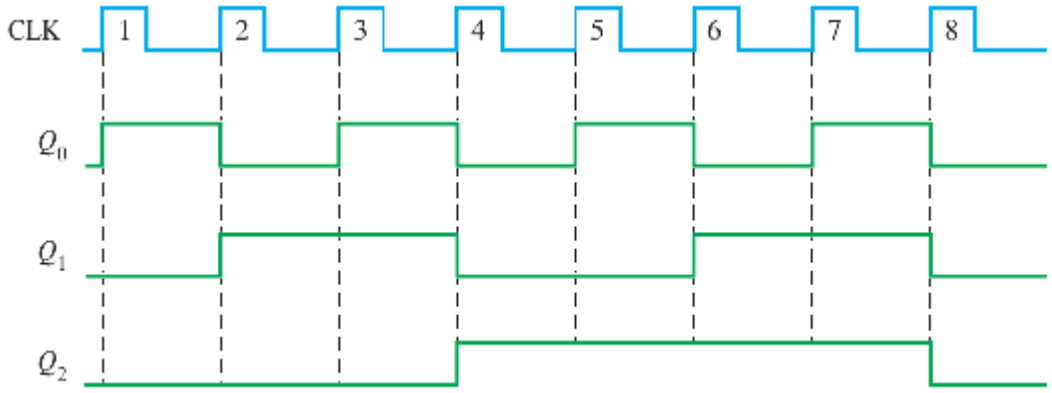




Synchronous counter - binary

3-bit synchronous binary counter





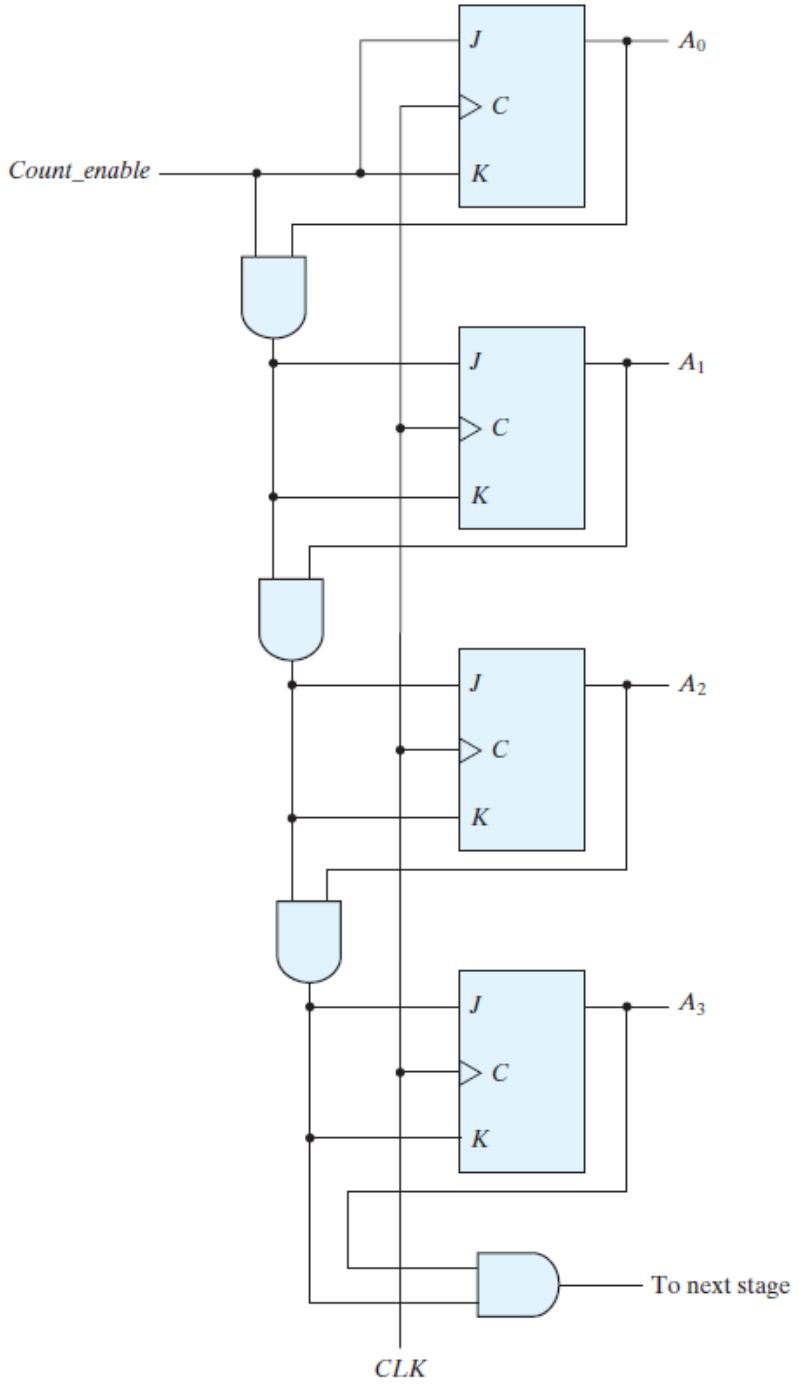
Clock Pulse	Q_2	Q_1	Q_0
Initially	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8 (recycles)	0	0	0

Synchronous counter - binary

- The design of a synchronous binary counter is very simple
- In a synchronous binary counter, the flip-flop in the least significant position is complemented with every pulse
- *A flip-flop in any other position is complemented when all the bits in the lower significant positions are equal to 1*
- For example, if the present state of a four-bit counter is $A_3A_2A_1A_0 = 0011$, 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 $A_1A_0 = 11$
- However, A_3 is not complemented, because the present state of $A_2A_1A_0 = 011$, which does not give an all-1's condition

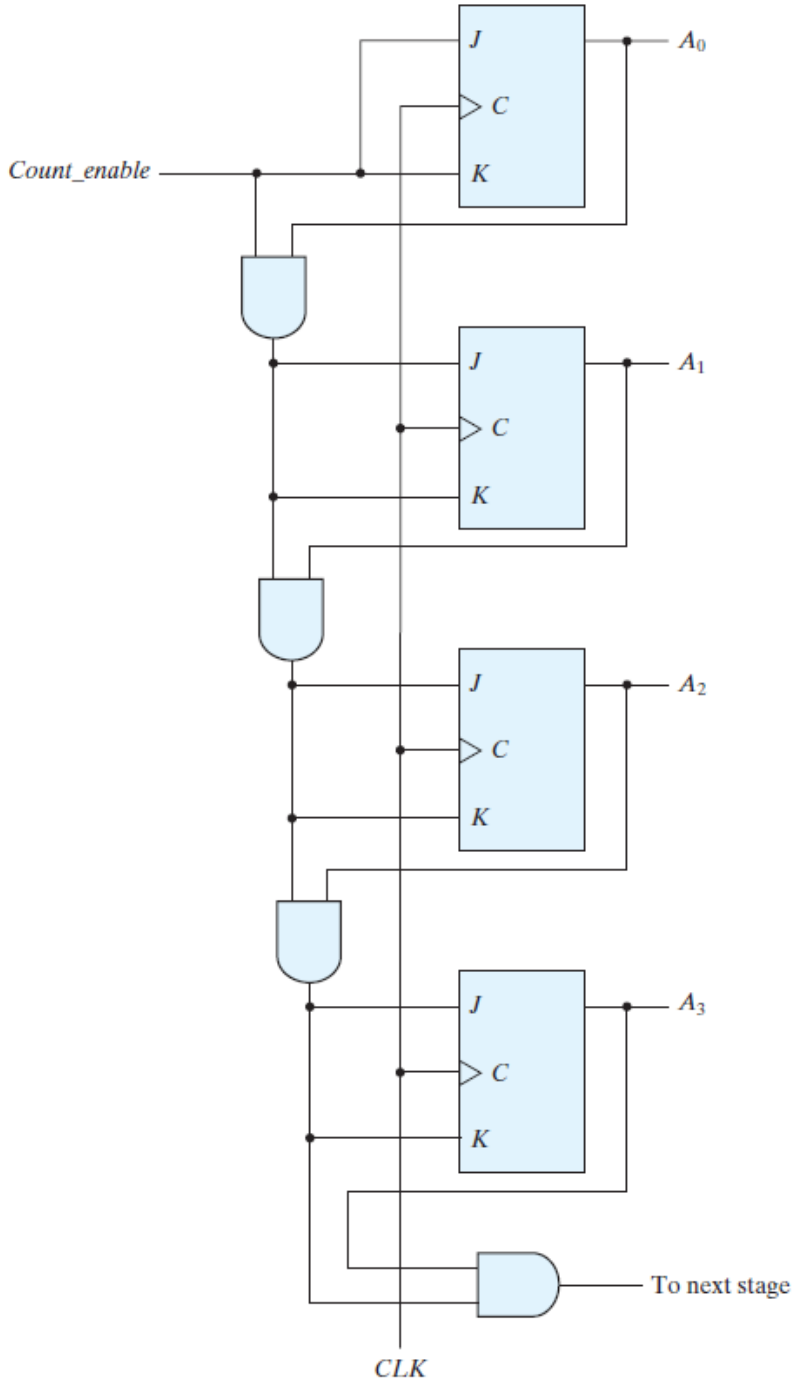
Synchronous counter - binary

- Synchronous binary counters have a regular pattern and can be constructed with complementing flip-flops and gates
- The regular pattern can be seen from the four-bit counter
- The *C* inputs of all flip-flops are connected to a common clock
- The counter is enabled by *Count_enable*
- If the enable input is 0, all *J* and *K* inputs are equal to 0 and the clock does not change the state of the counter
- The first stage, A_0, has its *J* and *K* equal to 1 if the counter is enabled
- The other *J* and *K* inputs are equal to 1 if all previous least significant stages are equal to 1 and the count is enabled



Synchronous counter - binary

- Note that the flip-flops trigger on the positive edge of the clock
- The synchronous counter can be triggered with either the positive or the negative clock edge
- The complementing flip-flops in a binary counter can be of either the *JK* type, the *T* type, or the *D* type with XOR gates



Synchronous counter – down counter

- A synchronous countdown binary counter goes through the binary states in reverse order, from 1111 down to 0000 and back to 1111 to repeat the count
- It is possible to design a countdown counter in the usual manner, but the result is predictable by inspection of the downward binary count
- The bit in the least significant position is complemented with each pulse
- *A bit in any other position is complemented if all lower significant bits are equal to 0*
- For example, the next state after the present state of 0100 is 0011
- The second significant bit is complemented because the first bit was 0
- The third significant bit is complemented because the first two bits were equal to 0
- But the fourth bit does not change, because not all lower significant bits are equal to 0
- A countdown binary counter can be constructed as a regular counter, except that the inputs to the AND gates must come from the complemented outputs, instead of the normal outputs, of the previous flip-flops

Synchronous counter – up/down counter

- Let us see if we can design a counter that counts up/down based on a control input
- The counter should count up if U=1, D=0; down if U=0, D=1; no change if U=D=0; and up if U=D=1
- We can cascade this logic to make the synchronous up/down counter

Synchronous counter – up/down counter

- Let us see if we can design a counter that counts up/down based on an input
- The counter should count up if U=1, D=0; down if U=0, D=1; no change if U=D=0; and up if U=D=1
- We can cascade this logic to make the synchronous up/down counter