Counters

Counter

* Terminal count is max value counter can reach

* How fast a counter reaches a terminal count is a function of the clock

* Resolution of the counter is also a function of the clock

For a 4 bit counter using a clock of 100 Hz:

- How fast will it reach the terminal count?
- How much change in time does each count represent?

For a 24 bit counter with a clock of 3 MHz:

- How fast will it reach the terminal count?
- How much change in time does each count represent?
- How many counts are in 1 second?

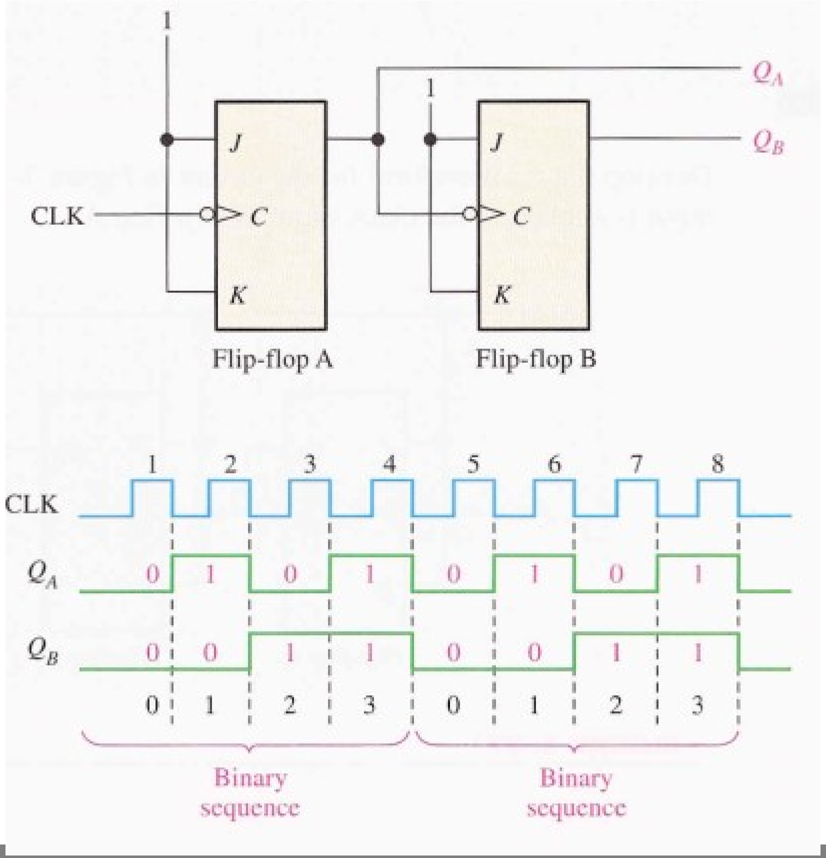
Counter

* A higher bit counter can give more counts, which can allow for more time to be represented with the counter (i.e. delays)

* A faster clock will give greater resolution in time of the counter output

Counter

- Flip flops connected together can perform a counting operation
  - Each clock cycle increases the count
  - Each flip flop is a bit in the count
- 2 – bit counter
  - Counts 0 to 3, has 2 flip flops
- 4 – bit counter
  - Counts 0 to 15, has 4 flip flops
- Counters roll over at terminal count

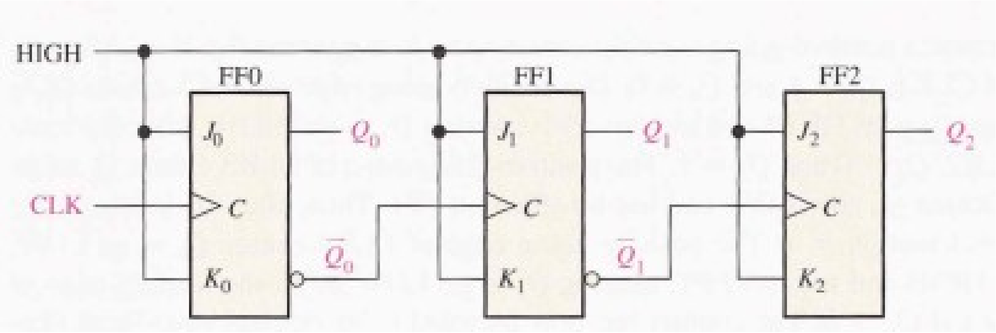


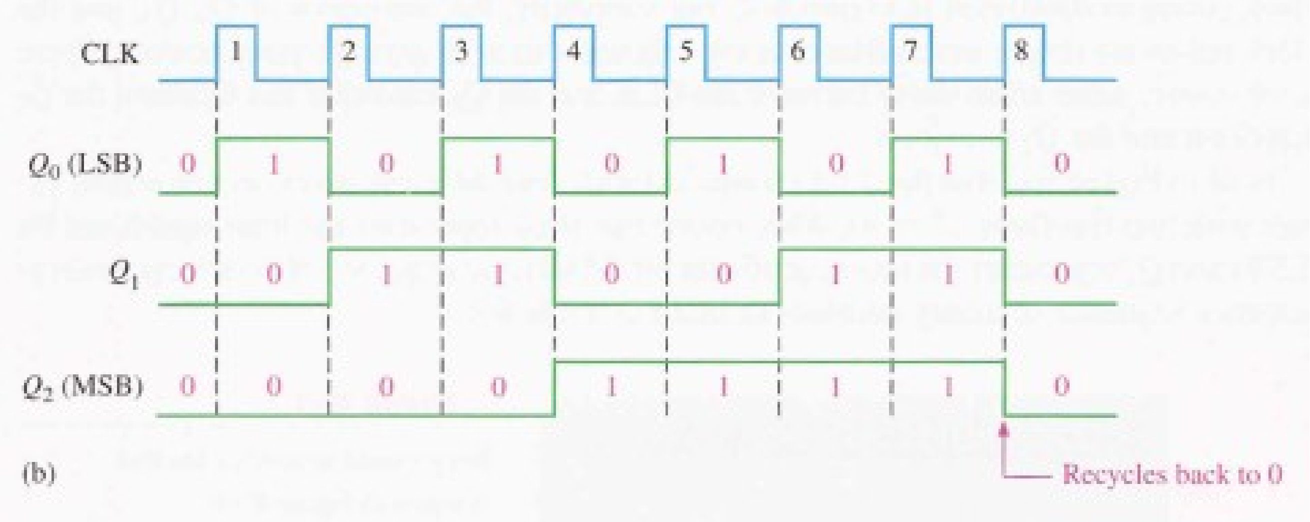
Asynchronous Counters

* Counter where the flip flops do not share a common clock

- Clock is connected to the LSB flip flop only, output of flip flop is the output of counter and the input to the next clock
- Implemented with a series of T flip flops (or equivalent using JK)

3 – bit Counter





Max Frequency

* The maximum frequency is determined by the propagation delay of the flip flops and the number of flip flops used.

* The clock can not change faster than the time it takes for the last flip flop (MSB) output of the counter to change

A 4 – bit async. counter is using flip flops that have a propagation delay of 10 ns. What is the fastest clock the counter can use?

Example

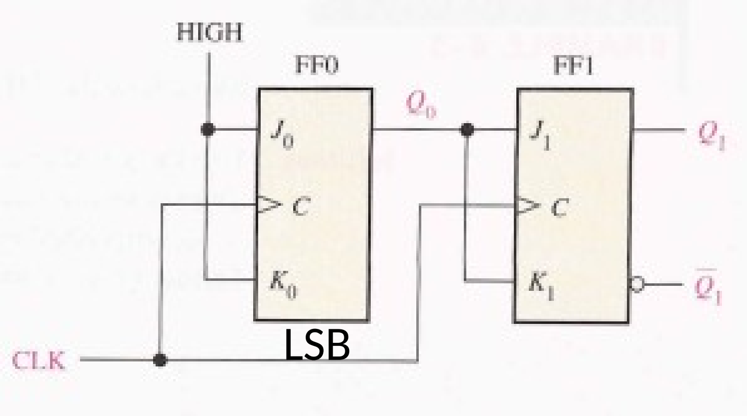
* Using T flip flops, design a 5 – bit asynchronous counter

- ◦ Assuming the propagation delay of each flip flop is 20 ns, what is the max frequency of the counter?
- ◦ Running the counter at max frequency, what is the resolution of the counter?
- ◦ How fast will the terminal count be reached at max frequency?

Synchronous Counters

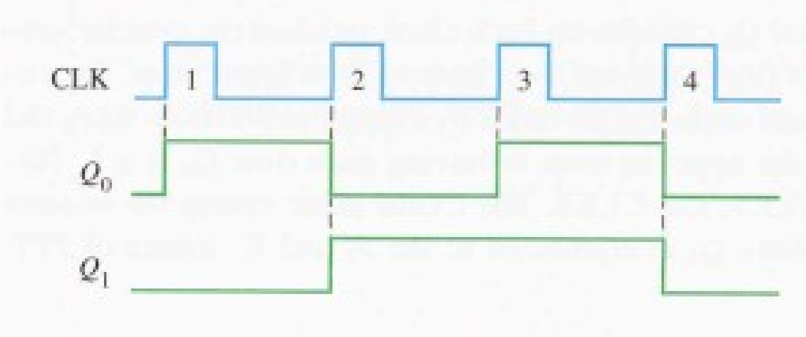
* Main difference is that all flip flops share the same clock and output of previous flip flop is input to the next one.

- Still implemented connecting T flip flops (or equivalent) together



LSB

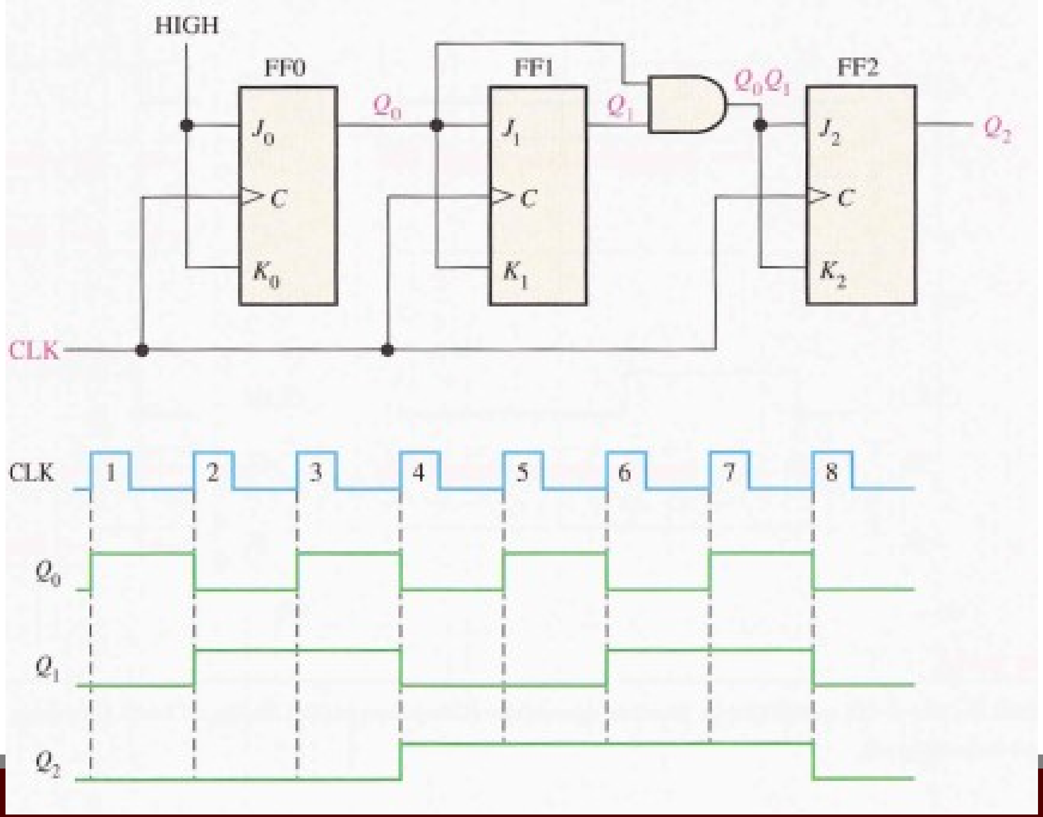
2 – bit counter



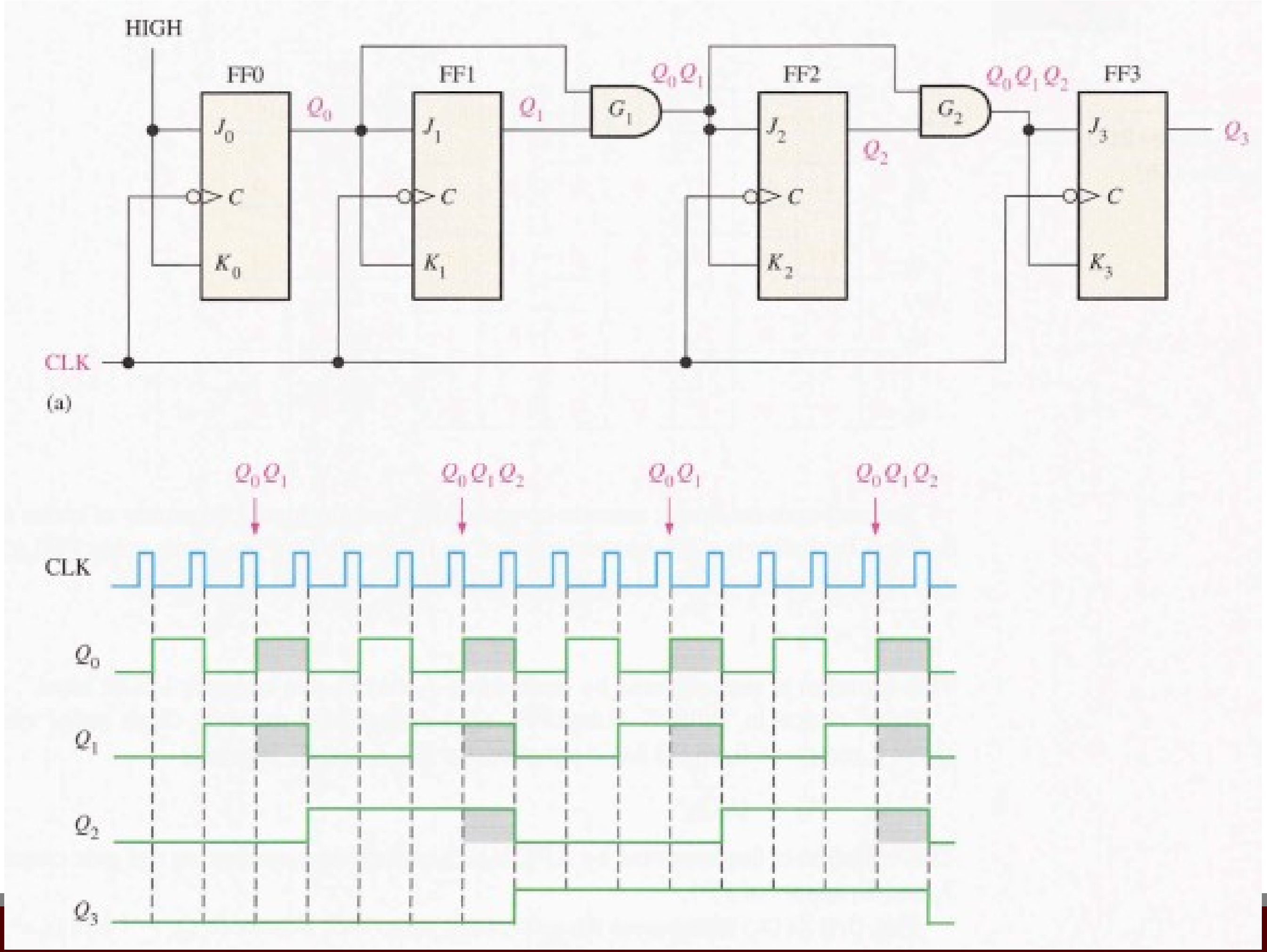
Creating Larger Sync. Counters

* Can not simply add another flip flop

* Each additional digit will have an AND gate (except LSB & MSB flip flop) that ANDs the flip flop output and input



4 – Bit Counter



Max Frequency

* The max frequency of the counter is determined by the max frequency that the flip flop clocks can handle

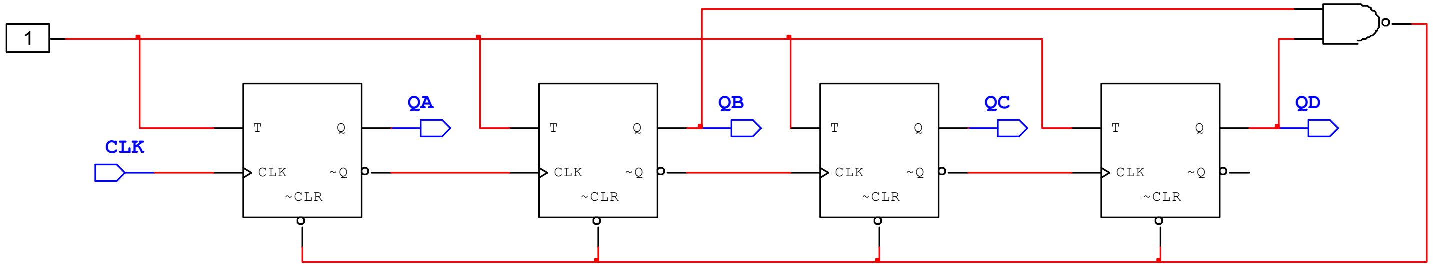
  - There is no ripple effect from clock

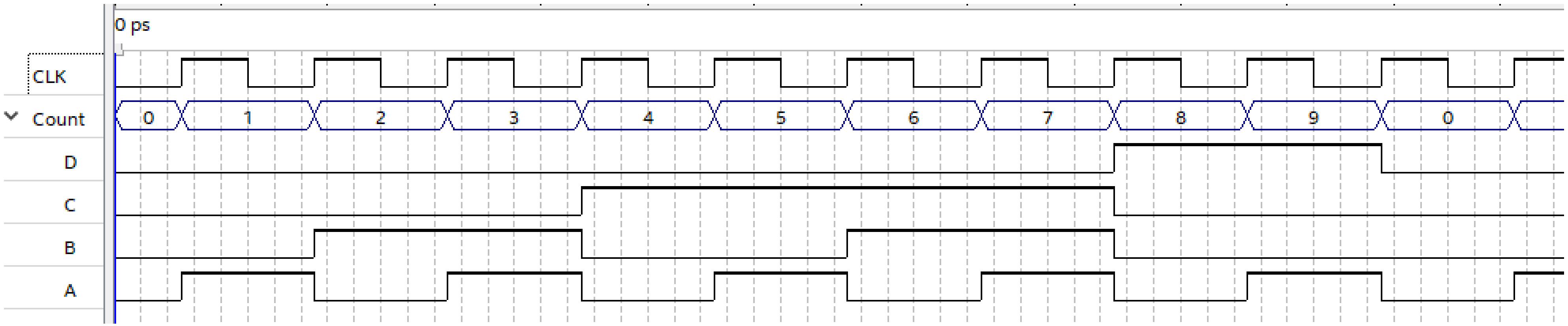
Example

* Using T flip flops, design a 5 – bit synchronous counter

Decade & MOD counters

- Decade counter only counts 10 values, decimal 0 to decimal 9
- Uses NAND gate to reset flip flops



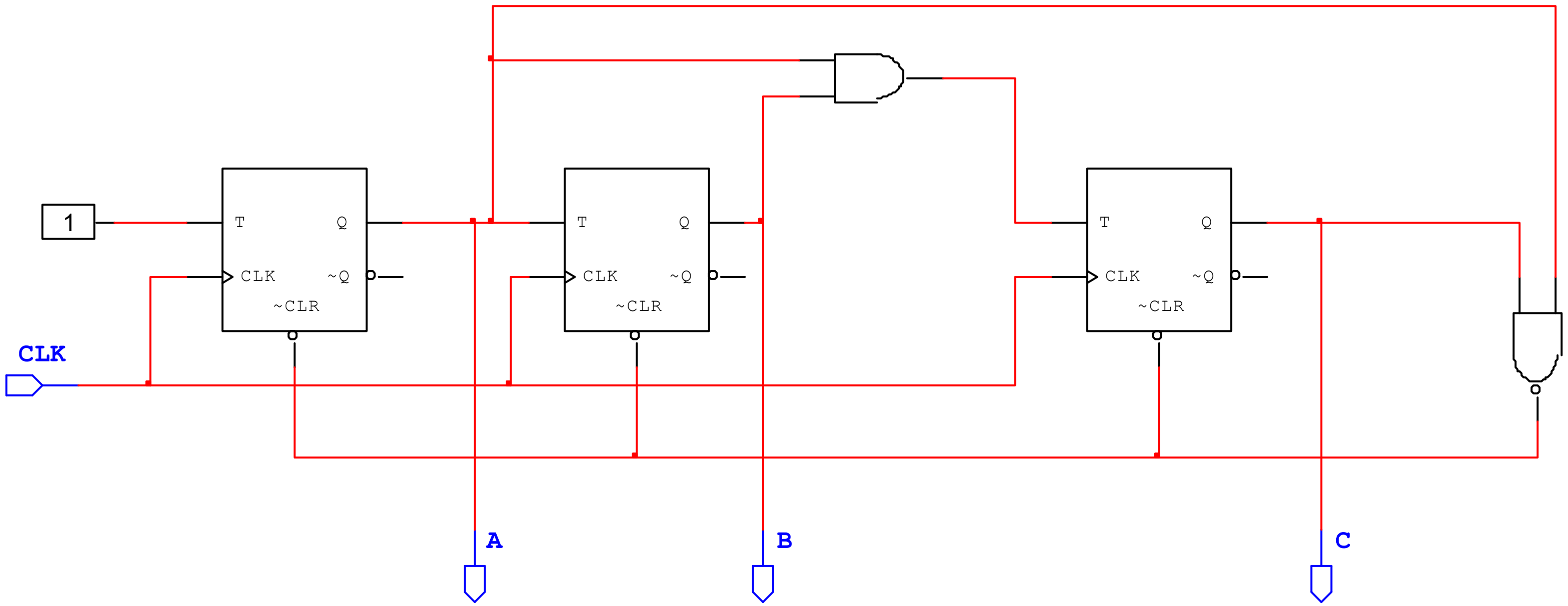


Modulus Counter

- Decade counter is a modulus or MOD 10 counter
- A MOD *n* counter only contains *n* counts
  - MOD 4 counts 0 to 3
  - MOD 12 counts 0 to 11
- Counters are reset once a certain count is reached
- How would an asynchronous MOD 12 counter be constructed with T flip flops & active HIGH CLK?

Synchronous MOD Counter

- Same concept of resetting flip flops when creating synchronous MOD counters
- What is the MOD of the counter below?

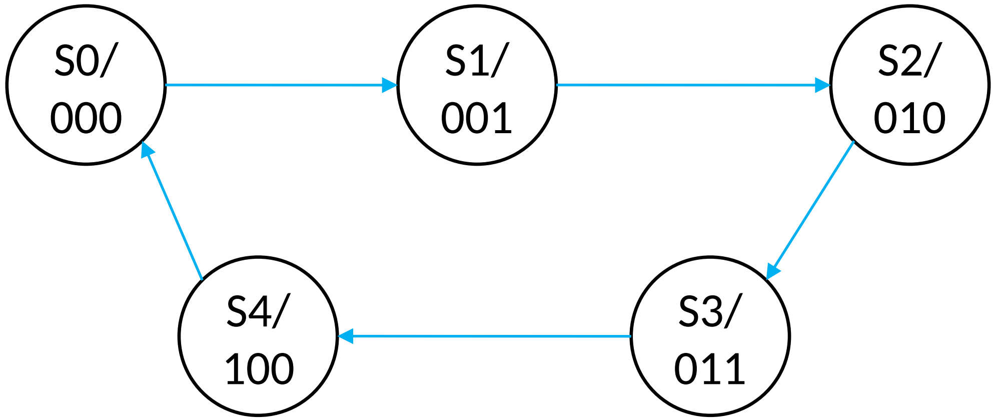


MOD Counter Design Steps

- Determine if counter is async. or sync.
- Find how many flip flops are needed.
- Determine count that will reset the flip flops
- Draw circuit

State Machine Counter Design

- Use a state diagram to create synchronous MOD counters



MOD 5 Counter State Diagram

Present State			Next State		
A	B	C	A	B	C
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

MOD 5 Counter Design

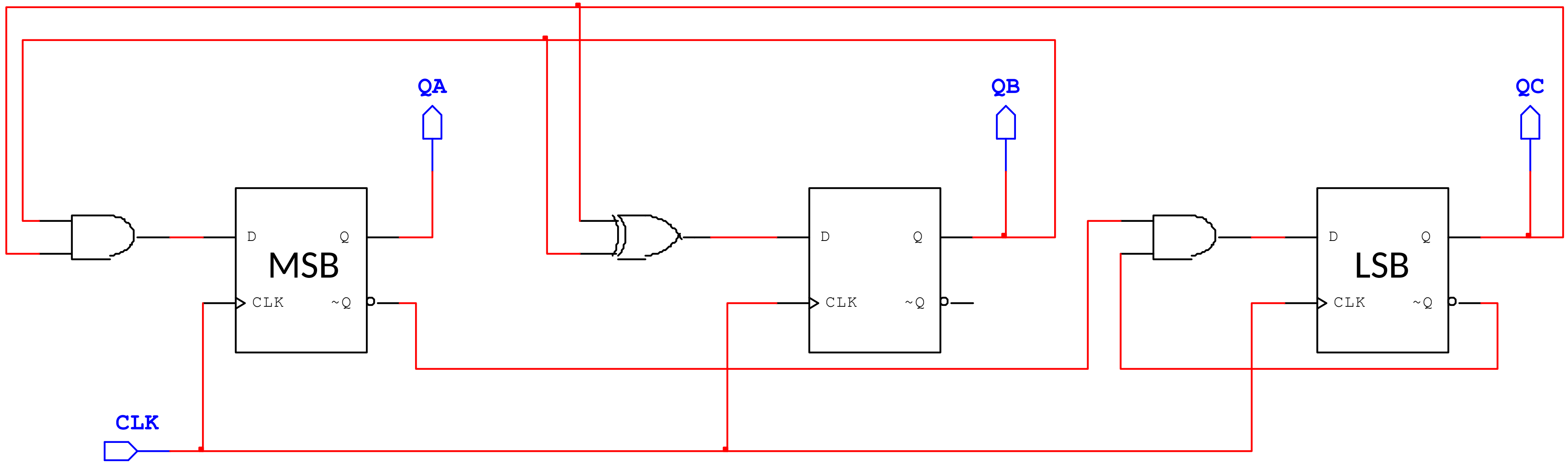
Present State			Next State		
A	B	C	A	B	C
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	x	x	x
1	1	0	x	x	x
1	1	1	x	x	x

A \ BC	00	01	11	10
0				
1				

A \ BC	00	01	11	10
0				
1				

A \ BC	00	01	11	10
0				
1				

MOD 5 Counter Design

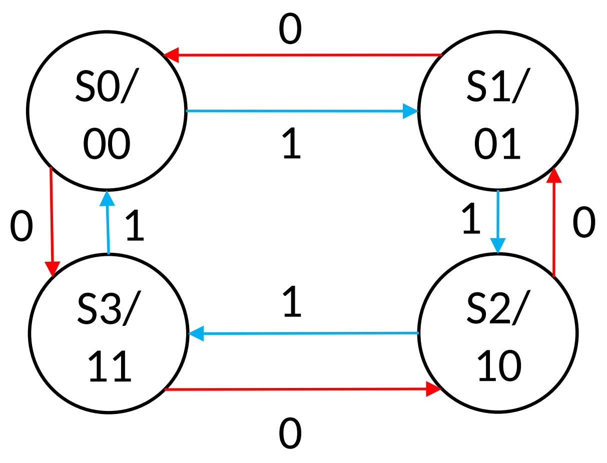


Up/Down Counter

* Use state diagram to design a 2 bit up down counter. It has one input that determines if the count is up (0,1,2,3) or down (3,2,1,0). If the input is HIGH, the counter counts up. If the input is LOW, the counter counts down.

Adding Control Inputs

- An input can be added to a counter to enable to count up or down



Present State			Next State	
A	B	x	A	B
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

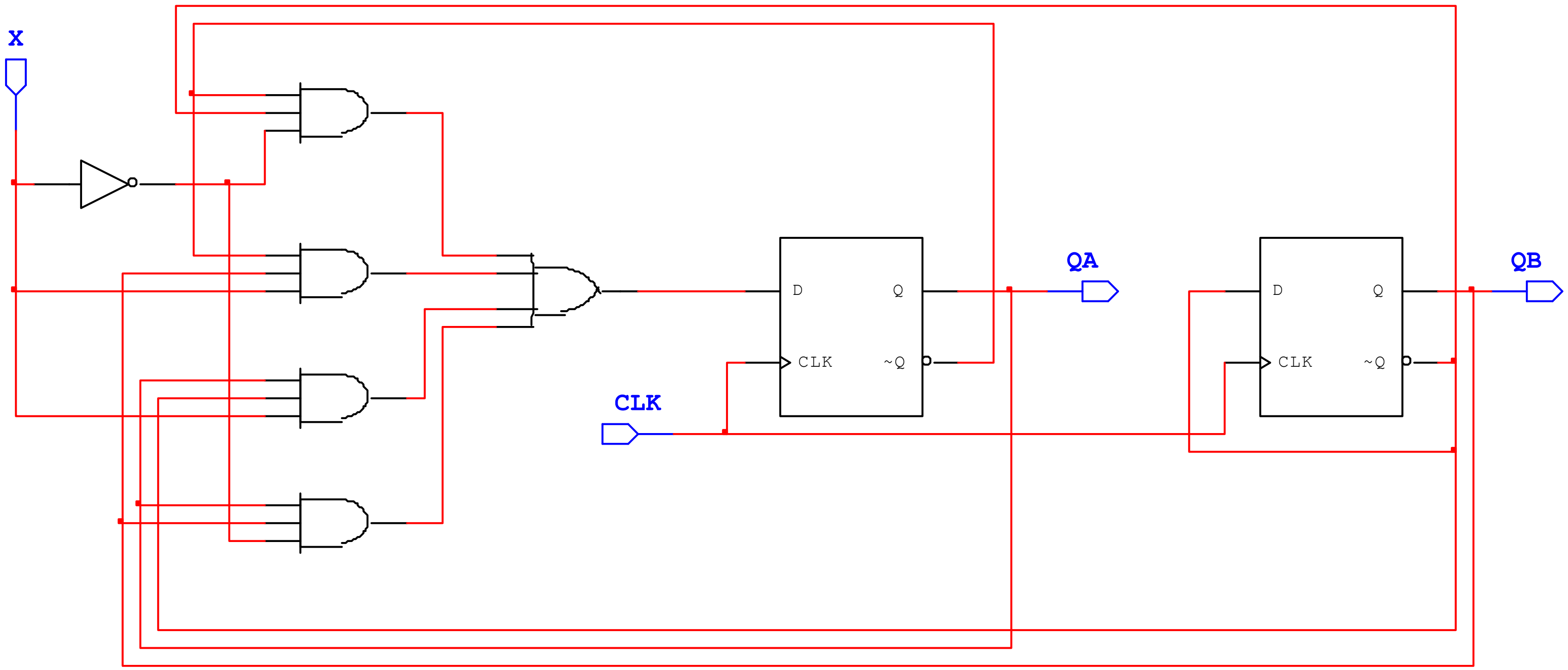
Up/Down Counter Example

Present State			Next State	
A	B	x	A	B
0	0	0	1	1
0	0	1	0	1
0	1	0	0	0
0	1	1	1	0
1	0	0	0	1
1	0	1	1	1
1	1	0	1	0
1	1	1	0	0

A \ Bx	00	01	11	10
0				
1				

A \ Bx	00	01	11	10
0				
1				

Up/Down Counter Example


X
QA
QB
CLK
D
Q
CLK
~Q
D
Q
CLK
~Q