

# **Sequential Circuit (Counter Design)**

# Overview

## Counters

- Introduction: Counters
- Asynchronous (Ripple) Counters
- Asynchronous Counters with MOD number  $< 2^n$
- Asynchronous Down Counters
- Cascading Asynchronous Counters

## **Cont...**

- Synchronous (Parallel) Counters
- Up/Down Synchronous Counters
- Designing Synchronous Counters
- Decoding A Counter
- Counters with Parallel Load

## **Cont...**

- Synchronous (Parallel) Counters
- Up/Down Synchronous Counters
- Designing Synchronous Counters
- Decoding A Counter
- Counters with Parallel Load
- Bidirectional Shift Registers
- An Application – Serial Addition
- Shift Register Counters
  - ❖ Ring Counters
  - ❖ Johnson Counters
- Random-Access Memory (RAM)

## Introduction: Counters

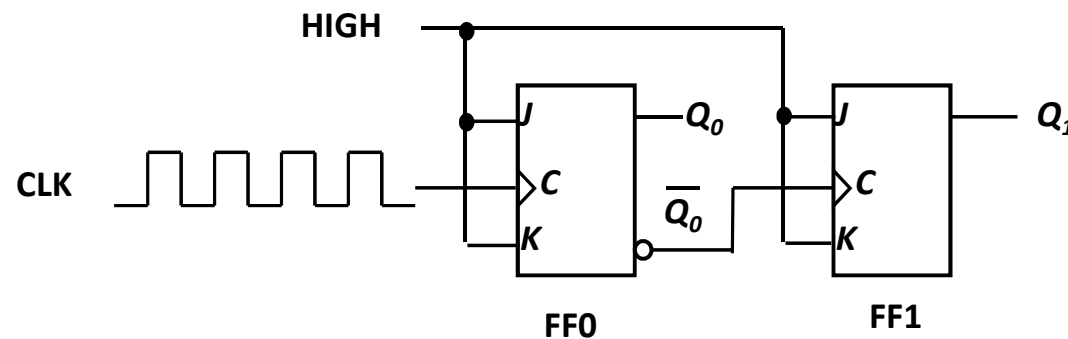
- **Counters** are circuits that cycle through a specified number of states.
- Two types of counters:
  - ❖ synchronous (parallel) counters
  - ❖ asynchronous (ripple) counters
- Ripple counters allow some flip-flop outputs to be used as a source of clock for other flip-flops.
- Synchronous counters apply the same clock to all flip-flops.

## Asynchronous (Ripple) Counters

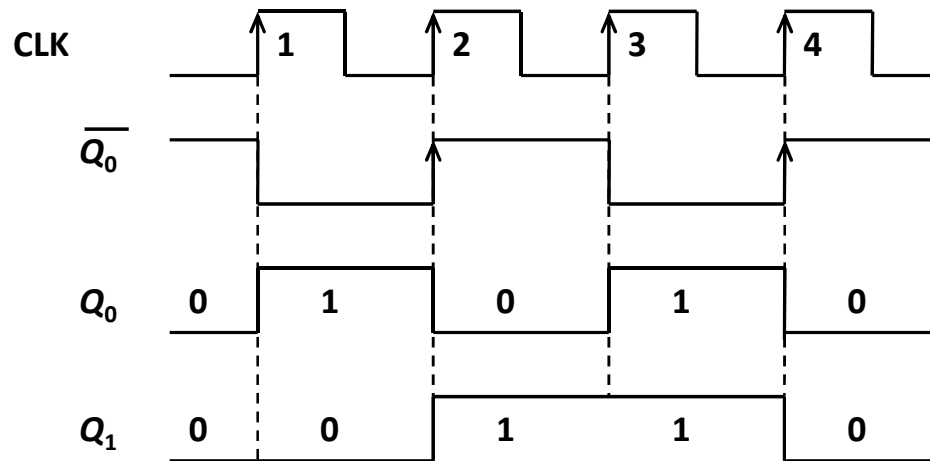
- **Asynchronous counters:** the flip-flops do not change states at exactly the same time as they do not have a common clock pulse.
- Also known as **ripple counters**, as the input clock pulse “ripples” through the counter – cumulative delay is a drawback.
- $n$  flip-flops  $\rightarrow$  a MOD (modulus)  $2^n$  counter. (Note: A MOD- $x$  counter cycles through  $x$  states.)
- Output of the last flip-flop (MSB) divides the input clock frequency by the MOD number of the counter, hence a counter is also a *frequency divider*.

## Asynchronous (Ripple) Counters

- Example: 2-bit ripple binary counter.
- Output of one flip-flop is connected to the clock input of the next more-significant flip-flop.



# Timing Diagram



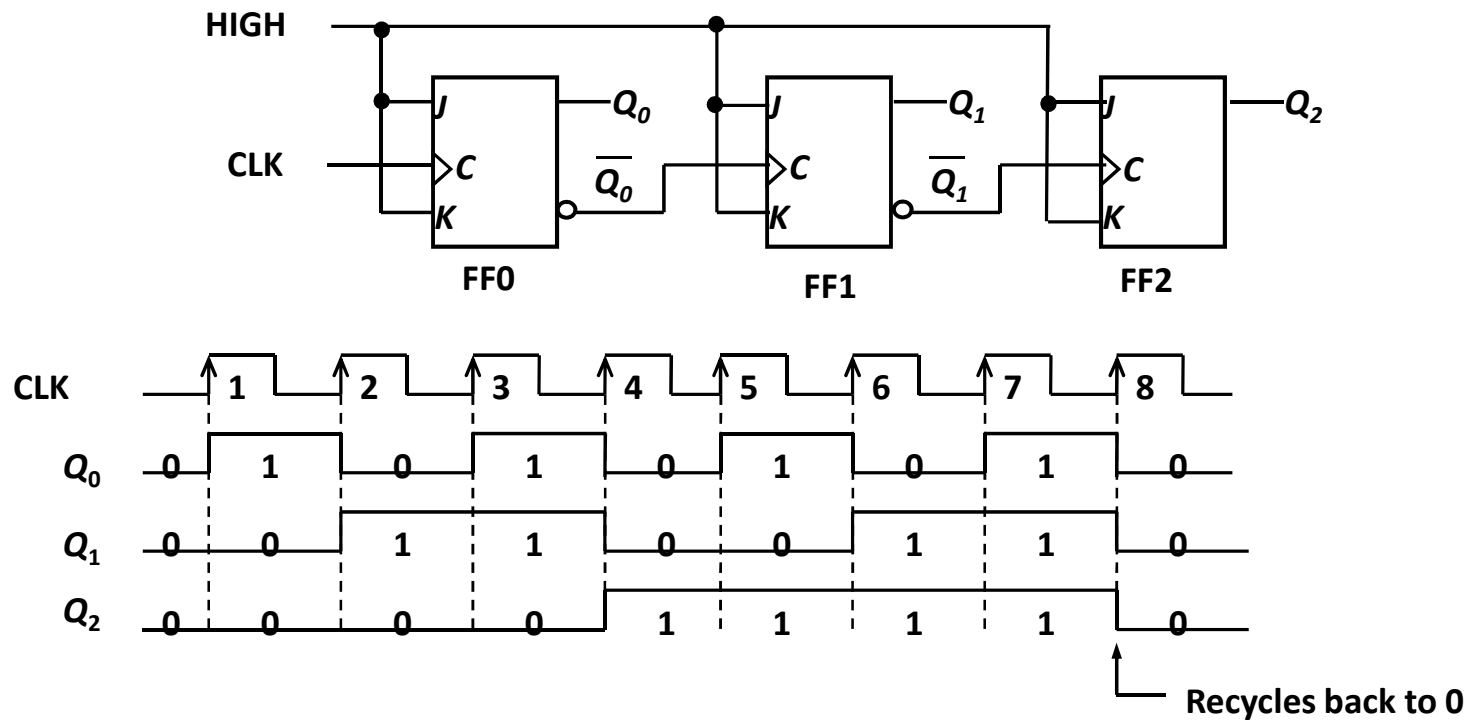
Timing diagram

$00 \rightarrow 01 \rightarrow 10 \rightarrow 11 \rightarrow 00 \dots$



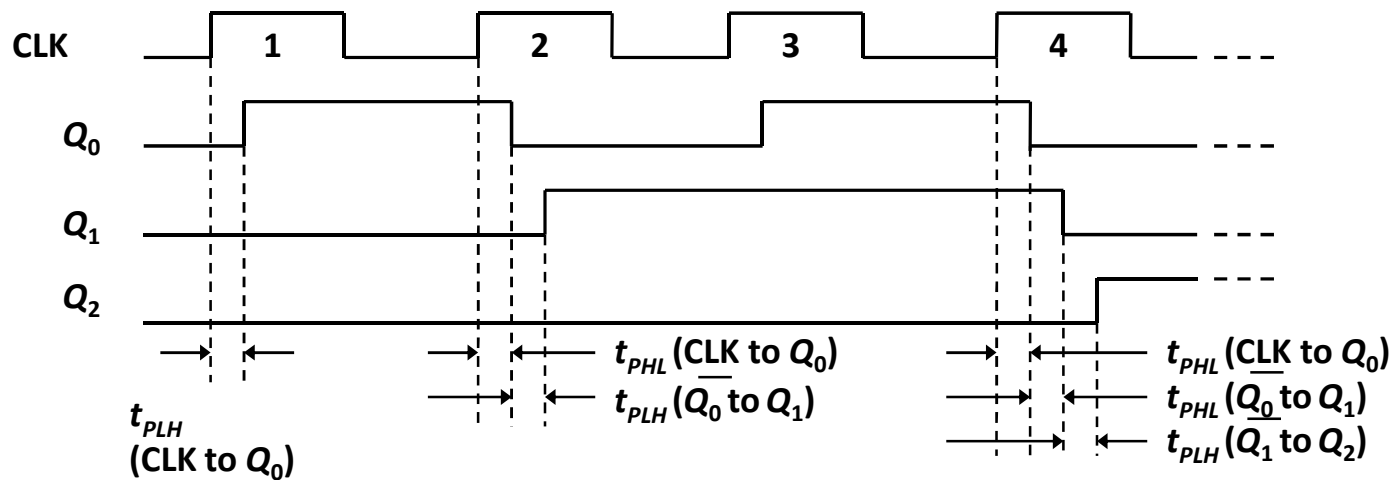
# Asynchronous (Ripple) Counters

- Example: 3-bit ripple binary counter.



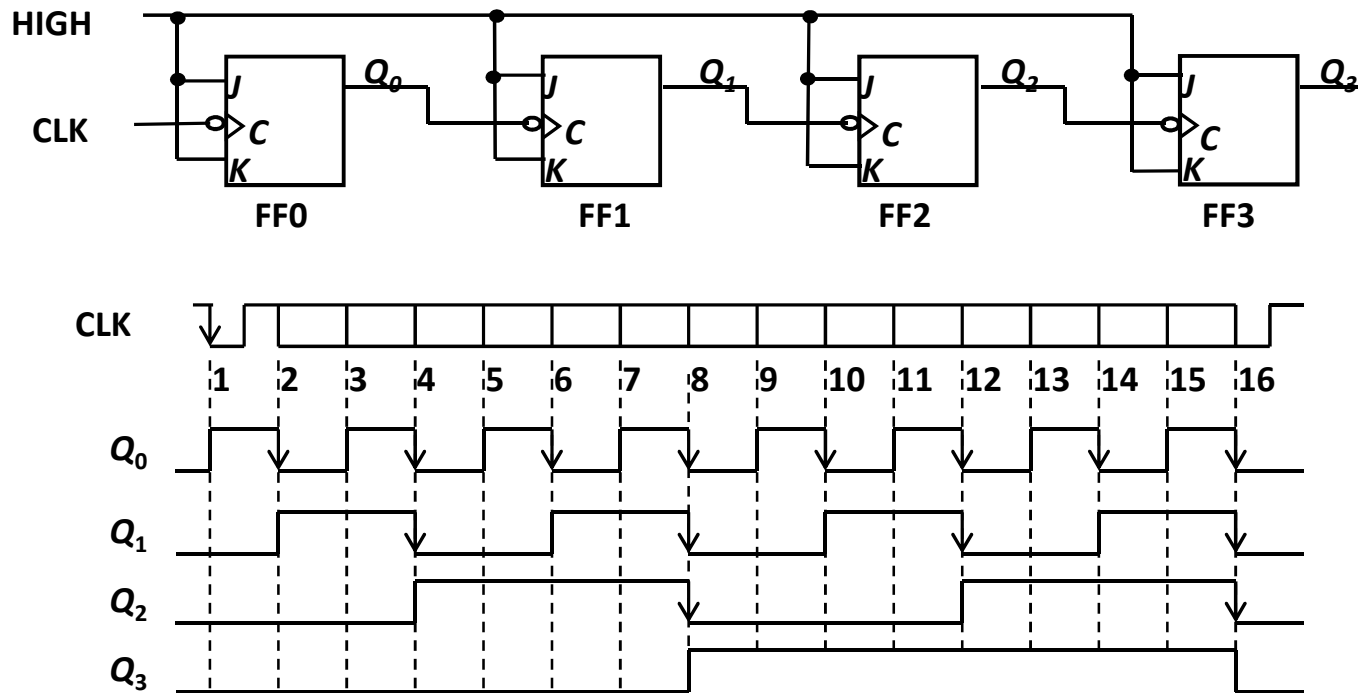
## Cont...

- Propagation delays in an asynchronous (ripple-clocked) binary counter.
- If the accumulated delay is greater than the clock pulse, some counter states may be misrepresented!



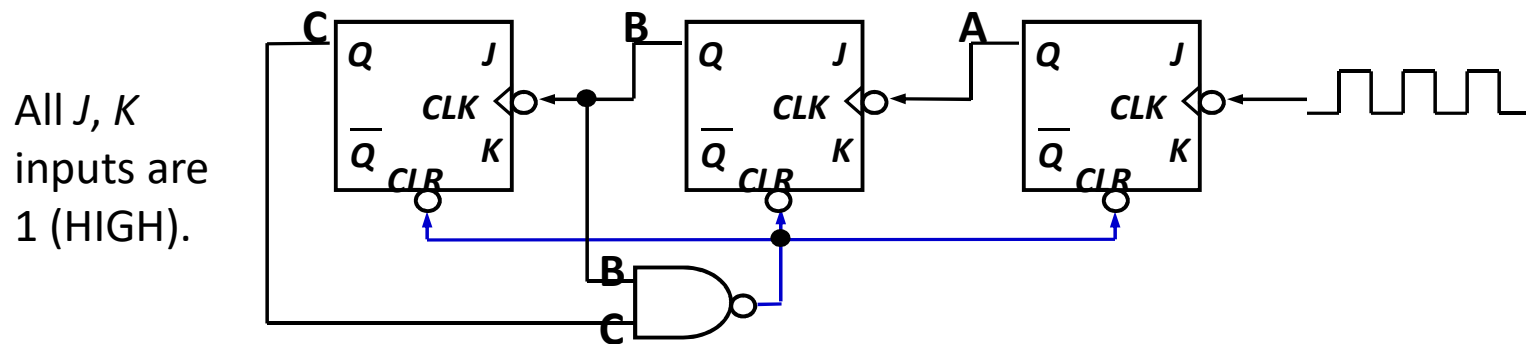
## Cont...

- Example: 4-bit ripple binary counter (negative-edge triggered).



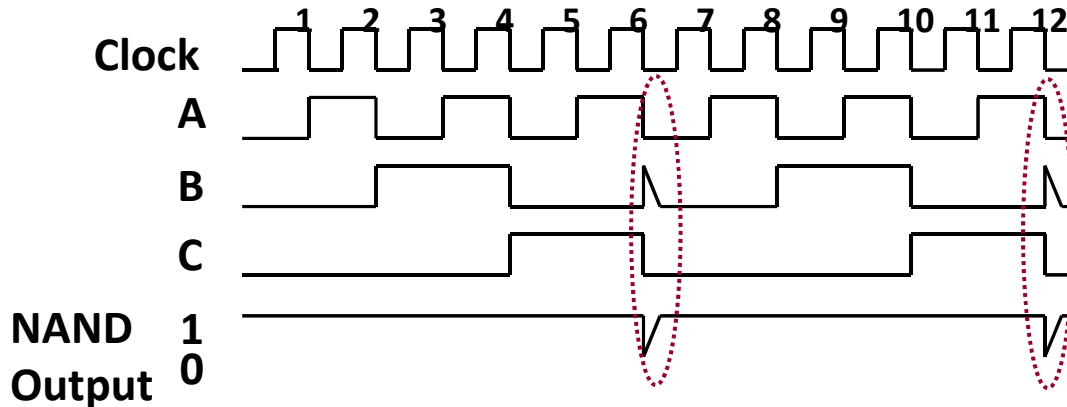
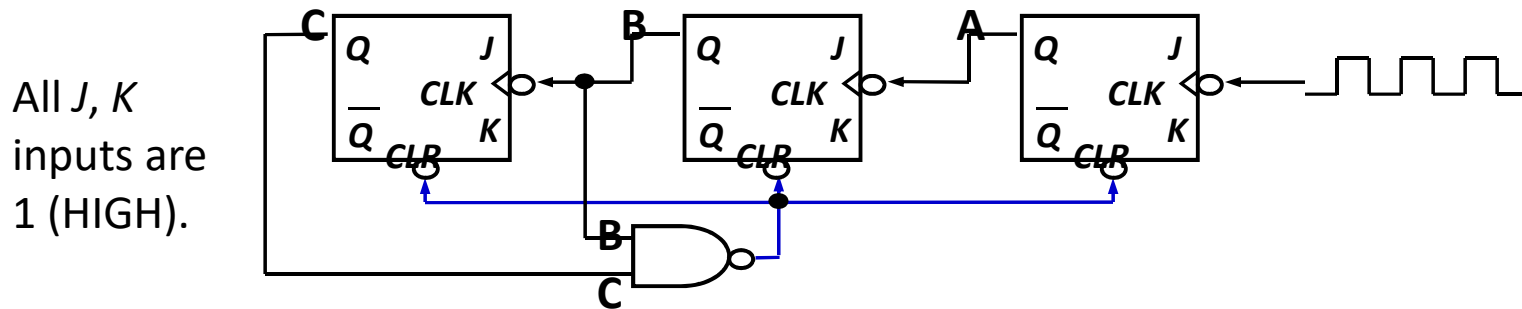
## Asyn. Counters with MOD no. $< 2^n$

- States may be skipped resulting in a **truncated sequence**.
- Technique: force counter to *recycle before going through all of the states* in the binary sequence.
- Example: Given the following circuit, determine the counting sequence (and hence the modulus no.)



# Cont...

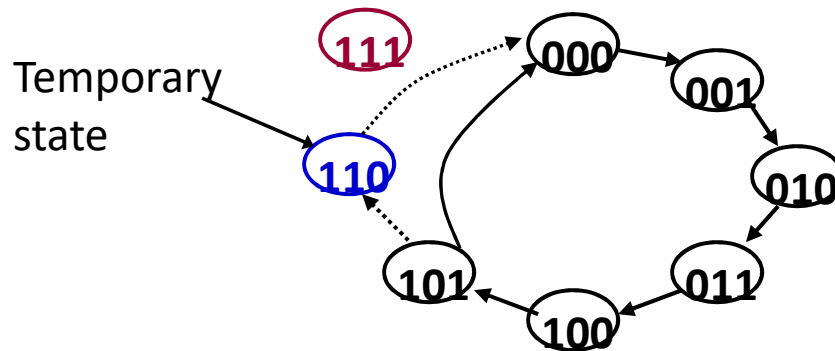
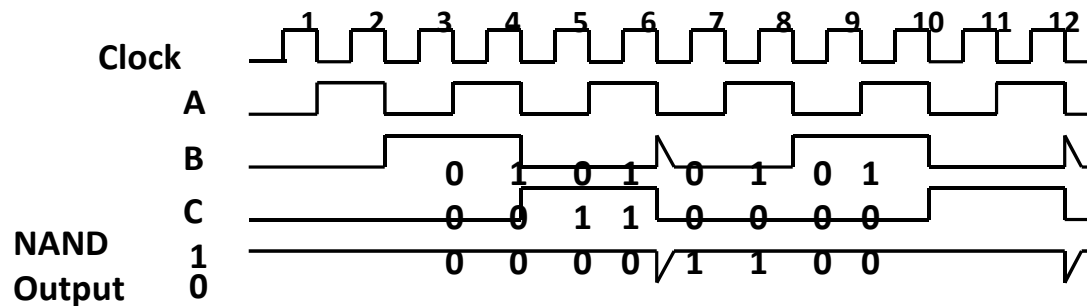
## Example (cont'd):



**MOD-6 counter**  
produced by clearing  
(a MOD-8 binary  
counter) when count  
of six (110) occurs.

## Asyn. Counters with MOD no. $< 2^n$

Example (cont'd): Counting sequence of circuit (in CBA order).

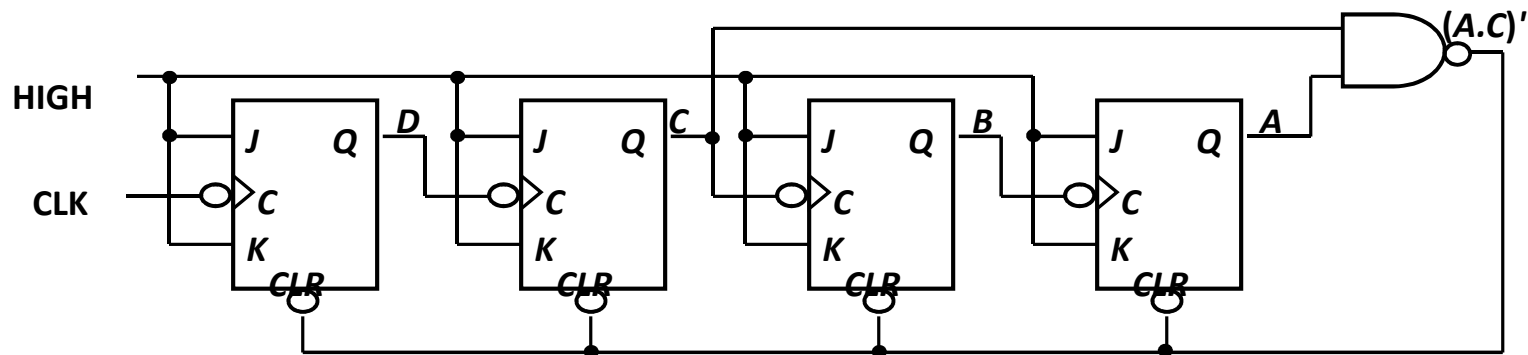


Counter is a MOD-6 counter.



## Asyn. Counters with MOD no. $< 2^n$

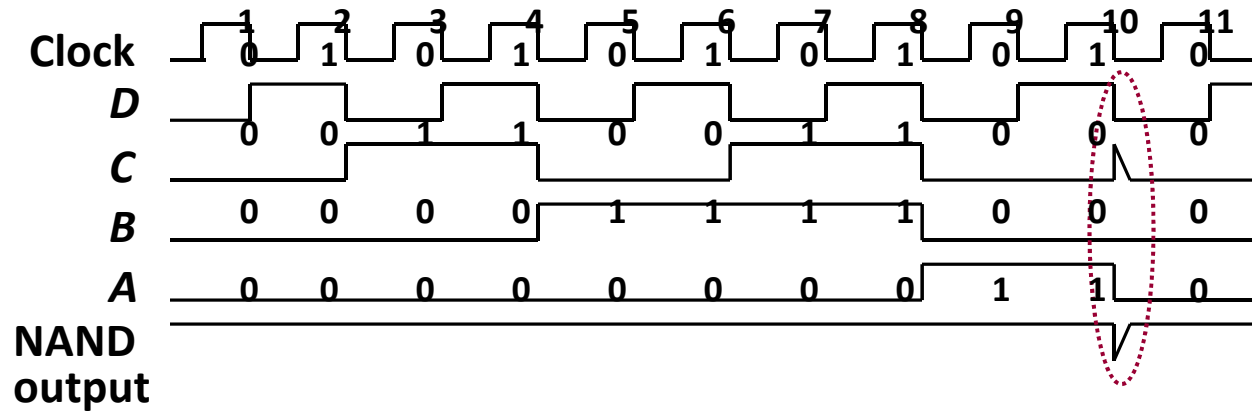
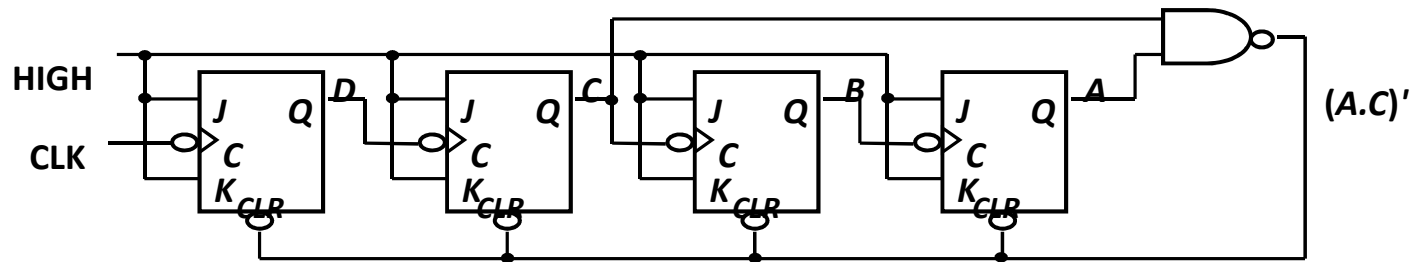
- **Decade counters** (or **BCD counters**) are counters with 10 states (modulus-10) in their sequence. They are commonly used in daily life (e.g.: utility meters, odometers, etc.).
- Design an asynchronous decade counter.





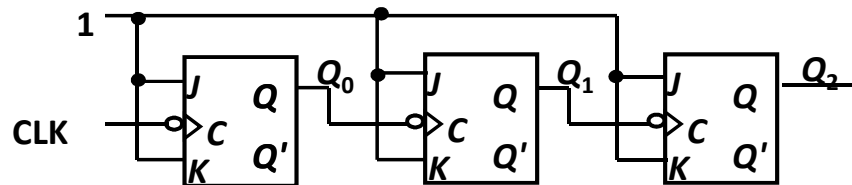
## Asyn. Counters with MOD no. $< 2^n$

- Asynchronous decade/BCD counter (cont'd).

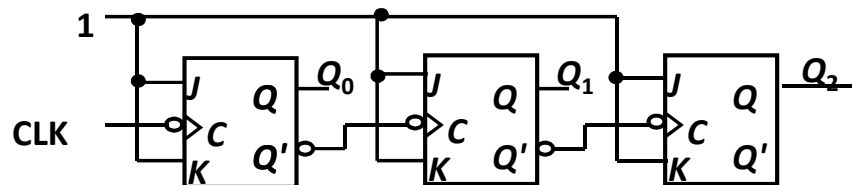


## Asynchronous Down Counters

- So far we are dealing with *up counters*. *Down counters*, on the other hand, count downward from a maximum value to zero, and repeat.
- Example: A 3-bit binary (MOD- $2^3$ ) down counter.



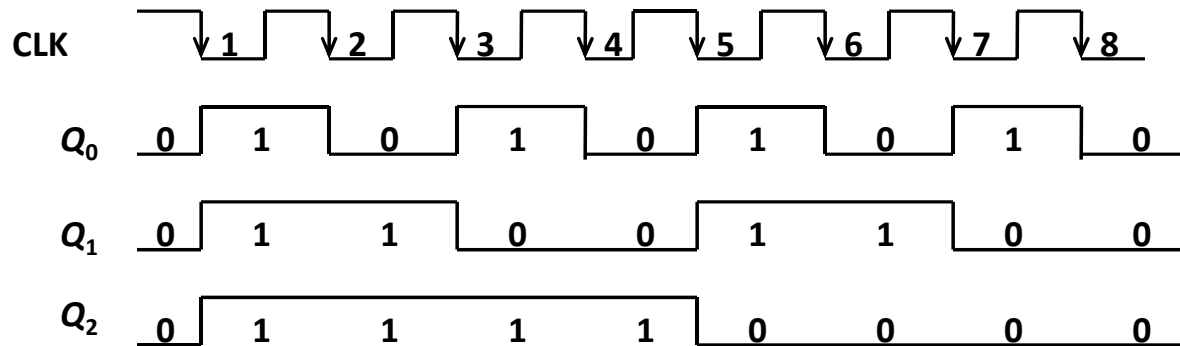
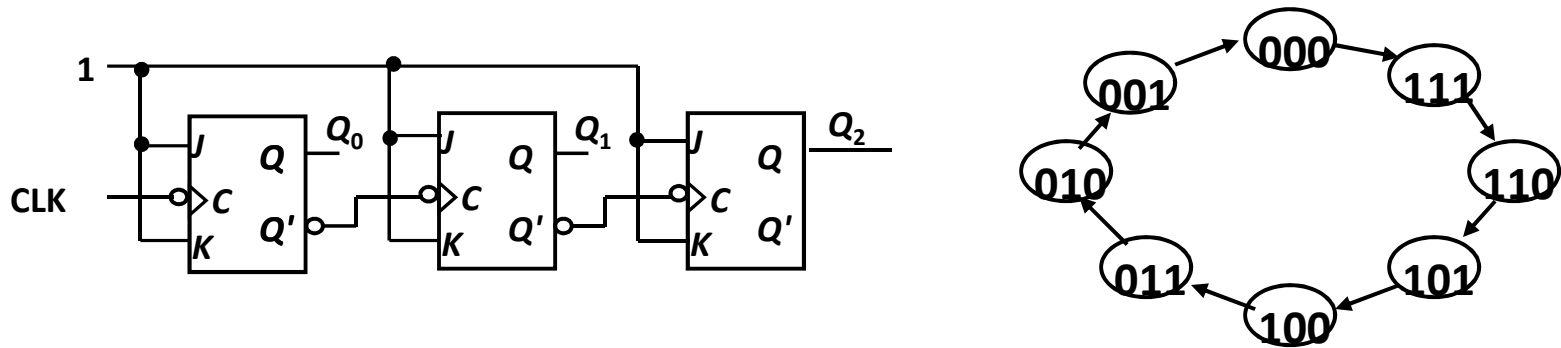
3-bit binary  
up counter



3-bit binary  
down counter

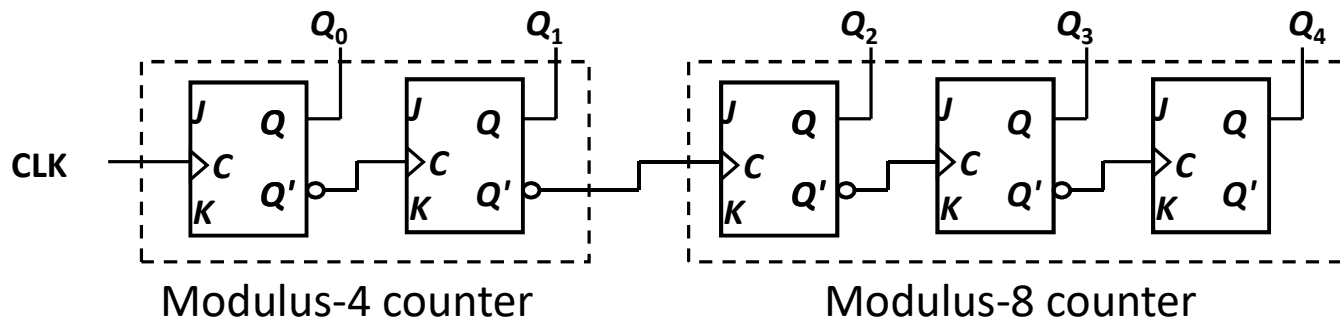
# Asynchronous Down Counters

- Example: A 3-bit binary (MOD-8) down counter.



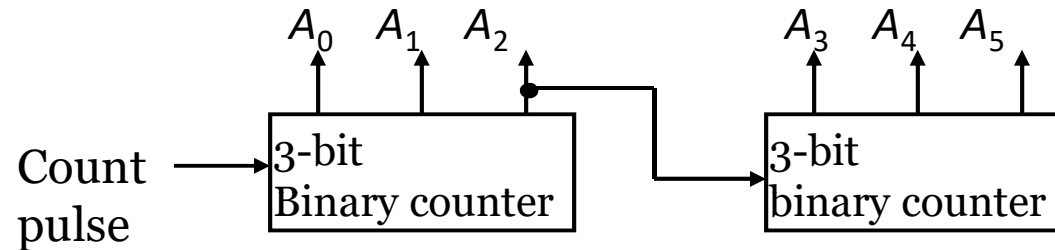
## Cascading Asynchronous Counters

- Larger asynchronous (ripple) counter can be constructed by cascading smaller ripple counters.
- Connect last-stage output of one counter to the clock input of next counter so as to achieve higher-modulus operation.
- Example: A modulus-32 ripple counter constructed from a modulus-4 counter and a modulus-8 counter.



## Cascading Asynchronous Counters

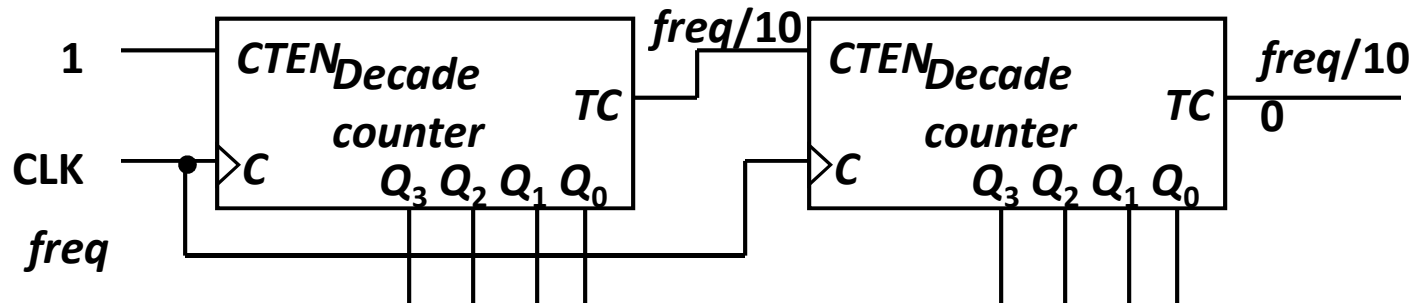
- Example: A 6-bit binary counter (counts from 0 to 63) constructed from two 3-bit counters.



$A_5$	$A_4$	$A_3$	$A_2$	$A_1$	$A_0$
0	0	0	0	0	0
0	0	0	0	0	1
0	0	0	:	:	:
0	0	0	1	1	1
0	0	1	0	0	0
0	0	1	0	0	1
:	:	:	:	:	:

## Cascading Asynchronous Counters

- If counter is not a binary counter, requires additional output.
- Example: A modulus-100 counter using two decade counters.



**$TC = 1$  when counter recycles to 0000**