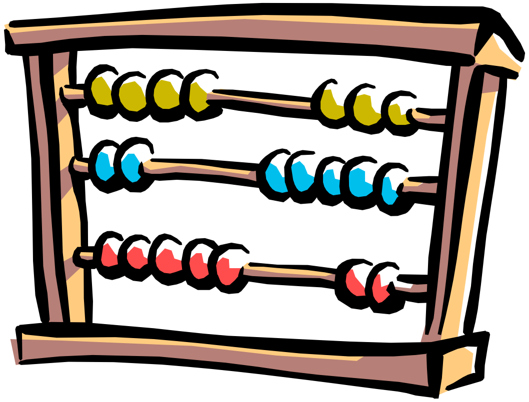
LECTURE 11

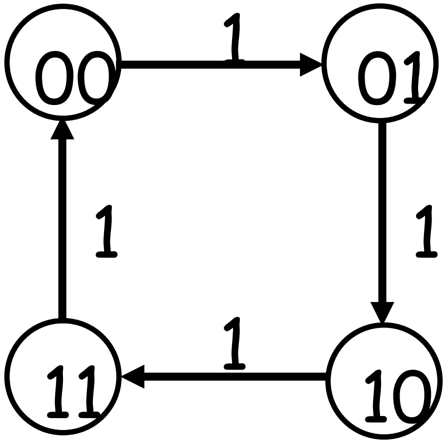
Digital Logic Families

Counters

- Counters are a specific type of sequential circuit.
- Like registers, the state, or the flip-flop values themselves, serves as the "output."
- The output value increases by one on each clock cycle.
- After the largest value, the output "wraps around" back to 0.
- Using two bits, we'd get something like this:



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



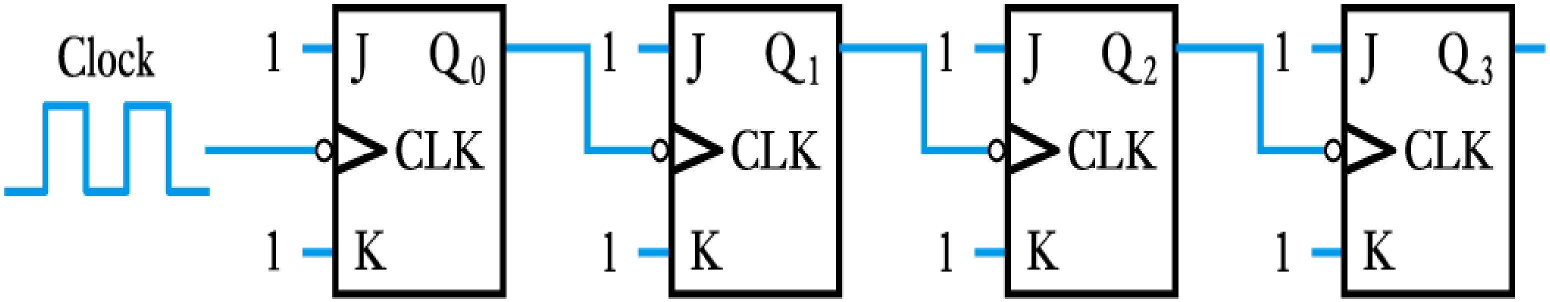
Benefits of counters

- Counters can act as simple clocks to keep track of “time.”
- You may need to record how many times something has happened.
  - How many bits have been sent or received?
  - How many steps have been performed in some computation?
- All processors contain a program counter, or PC.
  - Programs consist of a list of instructions that are to be executed one after another (for the most part).
  - The PC keeps track of the instruction currently being executed.
  - The PC increments once on each clock cycle, and the next program instruction is then executed.
- In digital logic and computing, a **counter** is a device which stores (and sometimes displays) the number of times a particular event or process has occurred, often in relationship to a clock signal.

Classifications of Counters

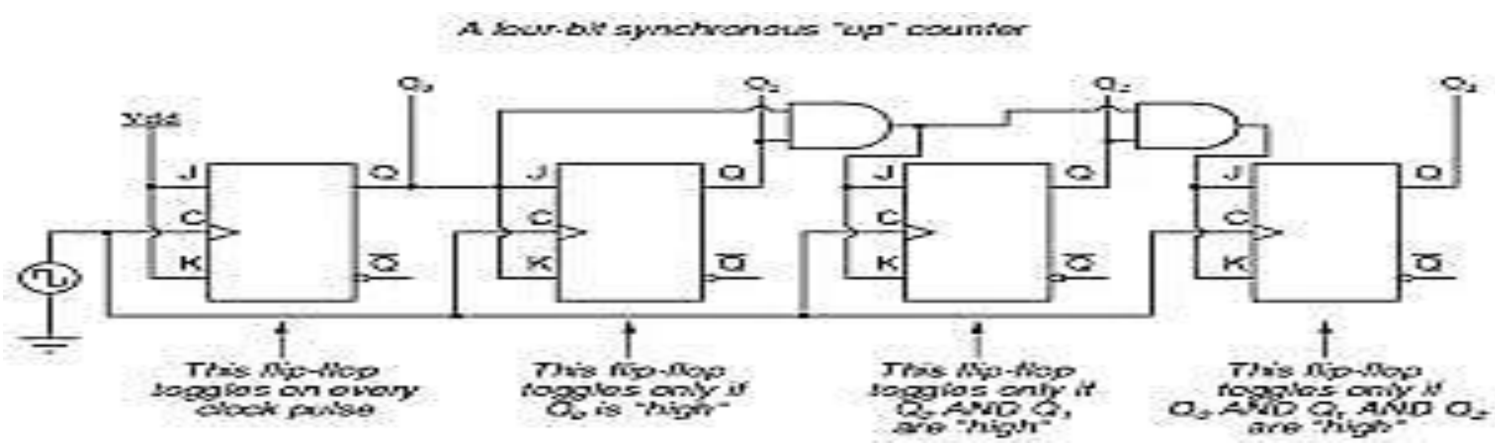
Asynchronous Counters

- Only the first flip-flop is clocked by an external clock. All subsequent flip-flops are clocked by the output of the preceding flip-flop.means output of previous flip-flop is connected to clock input of next flip flop.
- Asynchronous counters are slower than synchronous counters because of the delay in the transmission of the pulses from flip-flop to flip-flop.
- Asynchronous counters are also **called *ripple-counters*** because of the way the clock pulse ripples it way through the flip-flops.



Synchronous Counters

- All flip-flops are clocked simultaneously by an external clock. Means clock input of all flip flops are connected to same external clock.
- Synchronous counters are faster than asynchronous counters because of the simultaneous clocking.
- Synchronous counters are an example of *state machine* design because they have a set of states and a set of transition rules for moving between those states after each clocked event.



- The number of flip-flops determines the count limit or number of states.

  (STATES = $2^{\textit{\# of flip flops}}$) States / Modulus / Flip-Flops

- The number of states used is called the *MODULUS.*

- For example, a Modulus-12 counter would count from 0 (0000) to 11 (1011) and requires four flip-flops (16 states - 12 used).

❖Electronic counters -- Examples

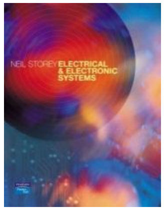
1. Up/down counter – counts both up and down, under command of a control input

2. Ring counter – formed by a shift register with feedback connection in a ring

3. Johnson counter – a *twisted* ring counter

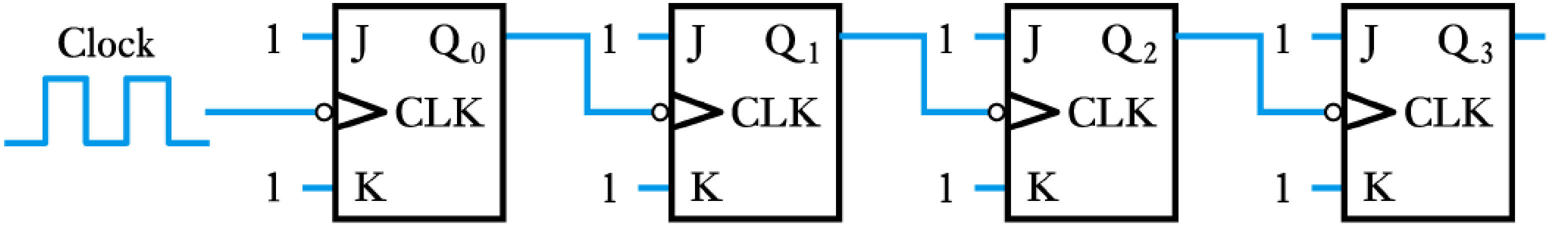
4. Cascaded counter

5. Decade Counter

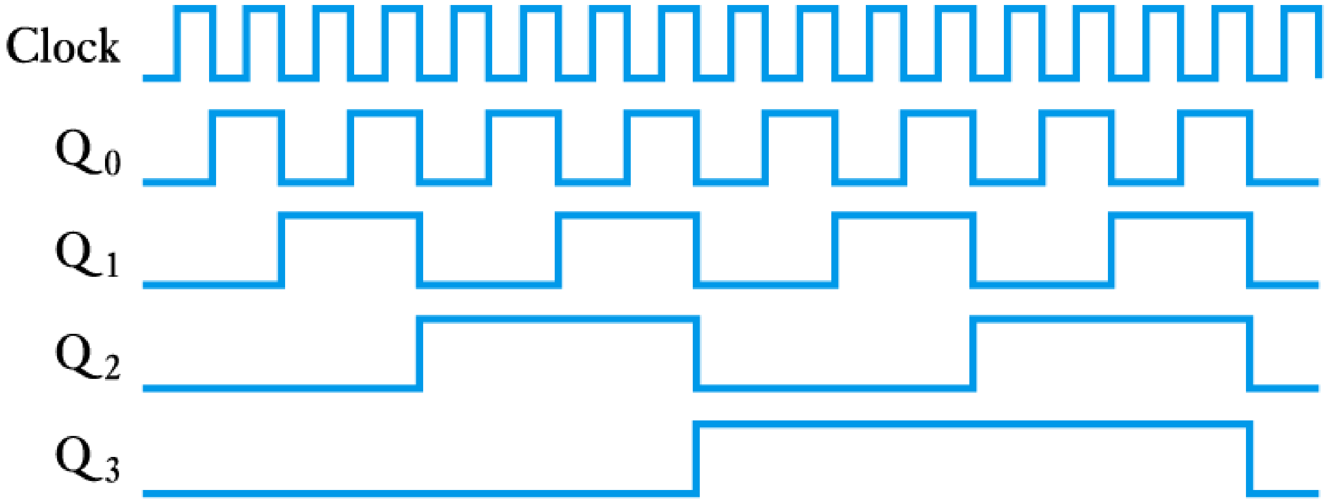


Asynchronous Counters

- **Asynchronous Counter/Ripple counters**
  - can be constructed using several flip flops
  - consider the following arrangement
  - with $J = K = 1$ each flip flop toggles on the falling edge of its clock input



- Each
  the fr
  previ

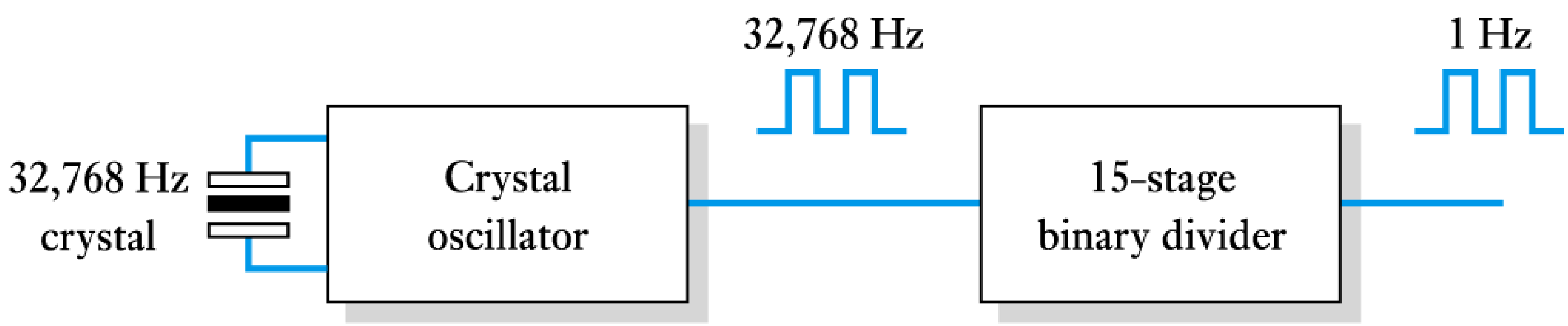


  - acts as a frequency divider
  - divides frequency by 2^n (n is the number of stages)

- Application of a frequency divider

Clock generator for a digital watch

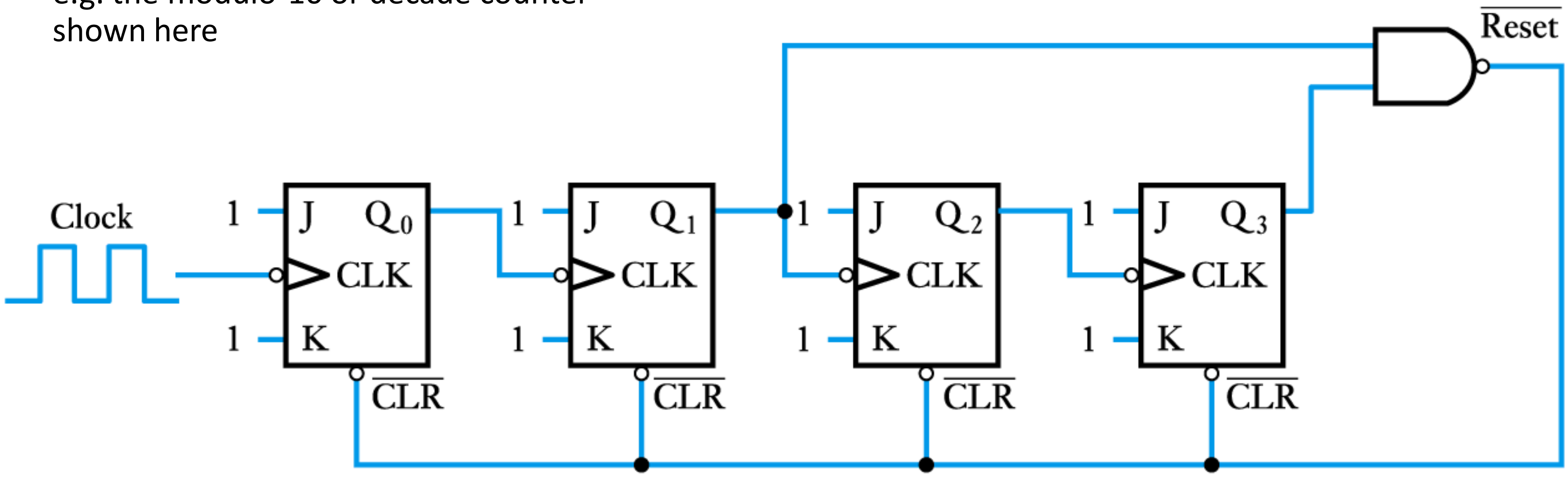
  - 15-stage counter divides signal from a crystal oscillator by 32,768 to produce a 1 Hz signal to drive stepper



- Consider the pattern on the outputs of the counter as shown – displayed on the right
- the outputs count in binary from 0 to 2^n-1 and then repeat
    - the circuit acts as a **modulo-2^n counter**
    - since the counting process propagates from one bistable to the next this is called a **ripple counter**
    - circuit shown is a **4-bit** or **modulo-16** (or **mod-16**) ripple counter

Number of clock pulses	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1
16	0	0	0	0
17	0	0	0	1
18	0	0	1	0
19	0	0	1	1
20	0	1	0	0

- **Modulo-*N* counters**
  - by using an appropriate number of stages the earlier counter can count modulo any power of 2
  - to count to any other base we add reset circuitry
  - e.g. the modulo-10 or decade counter shown here



- **Down and up/down Counters**
  - a slight modification to the earlier circuit will produce a counter that counts from 2^n-1 to 0 and then restarts
  - this is a **down counter**
  - a further modification can produce an **up/down counter** which counts up or down depending on the state of a control line (usually labelled)
    - when this is 1 the counter counts up
    - when this is 0 the counter counts down

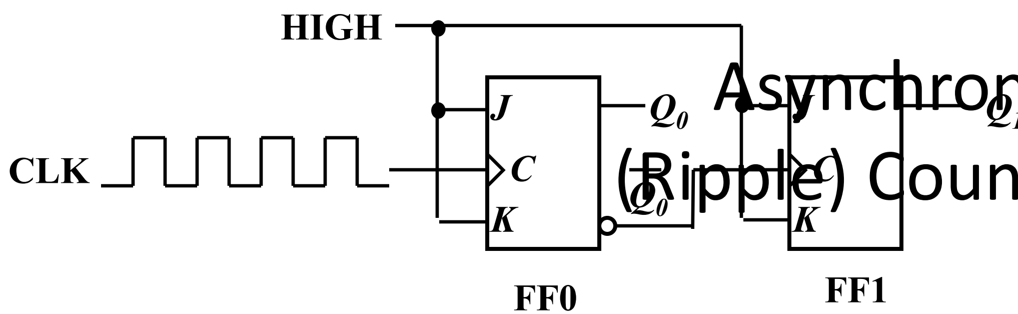
$$up/\overline{down}$$

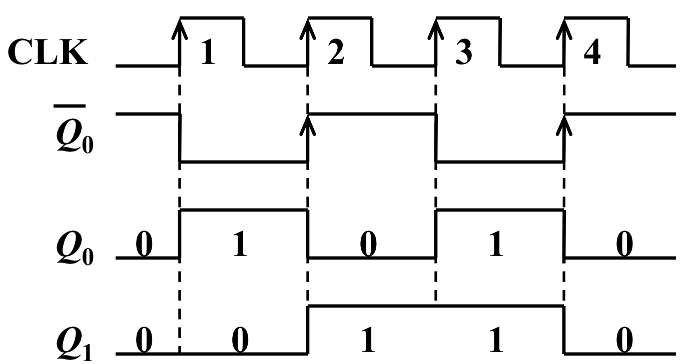
Drawbacks/Limitation of Ripple Counter

- **Propagation delay in counters**
  - while ripple counters are very simple they suffer from problems at high speed
  - since the output of one flip-flop is triggered by the change of the previous device, delays produced by each flip-flop are summed along the chain
  - the time for a single device to respond is termed its **propagation delay time** t_{PD}
  - an n-bit counter will take $n \times t_{PD}$ to respond
  - if read before this time the result will be garbled

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.



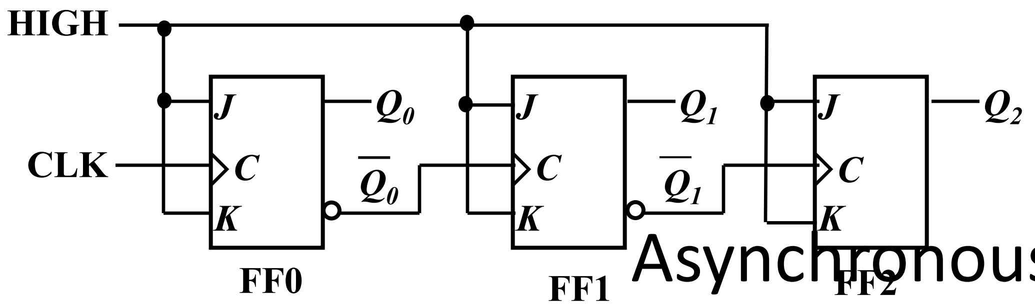


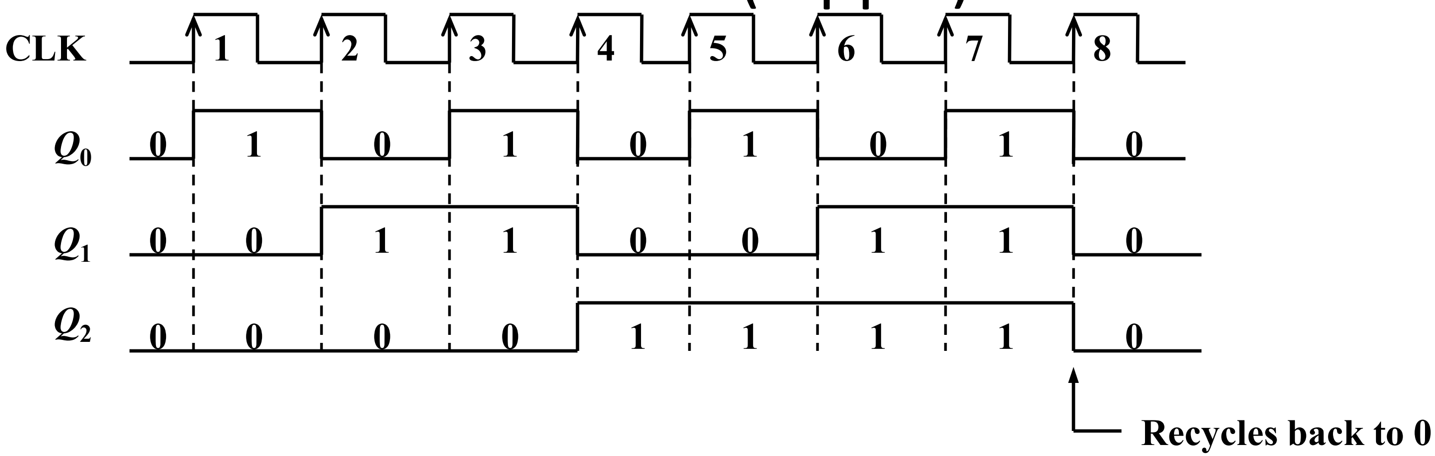
Timing diagram

$00 \rightarrow 01 \rightarrow 10 \rightarrow 11 \rightarrow 00$ ...

Asynchronous (Ripple) Counters

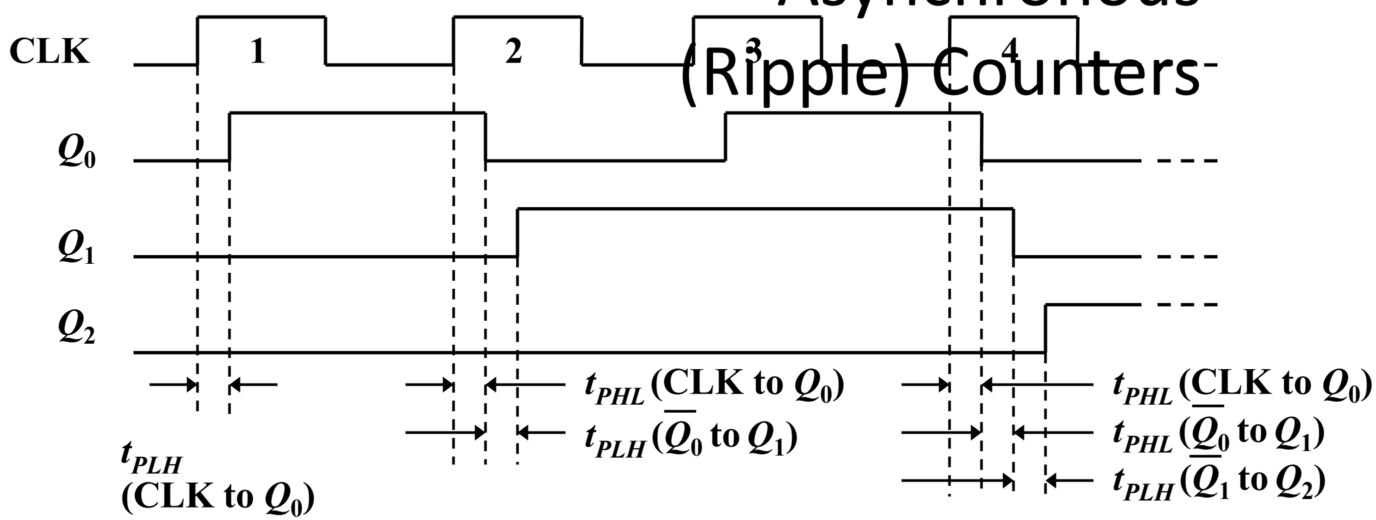
- Example: 3-bit ripple binary counter.





Asynchronous (Ripple) Counters

- 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!



Asynchronous (Ripple) Counters

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