PATRIoT lab

Lecture 23 – Registers and Counters 3

Dr. Aftab M. Hussain,

Assistant Professor, PATRIoT Lab, CVEST

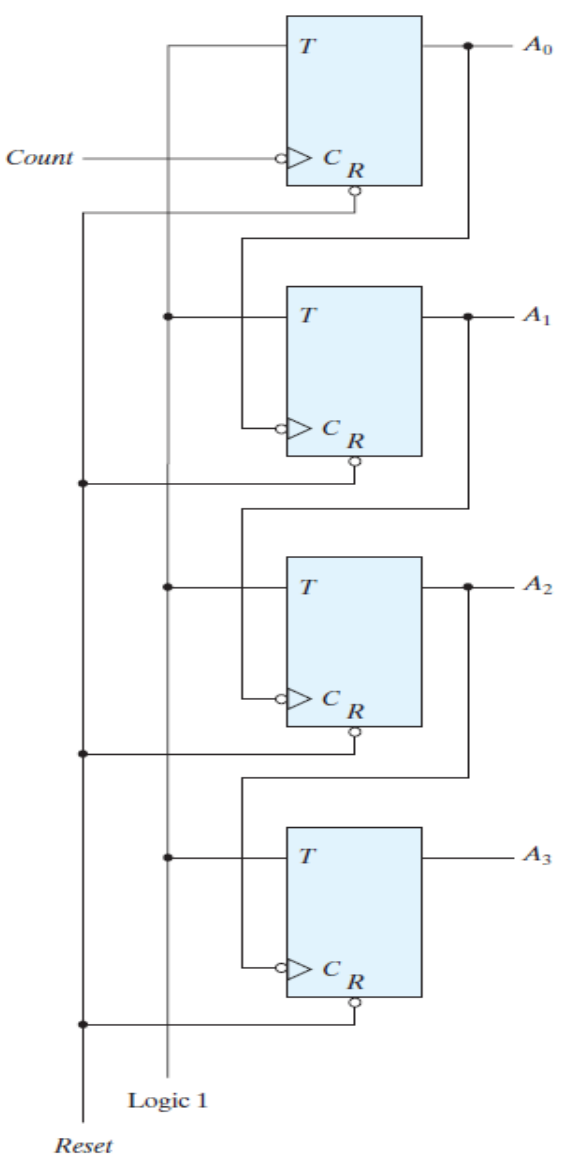
Chapter 6

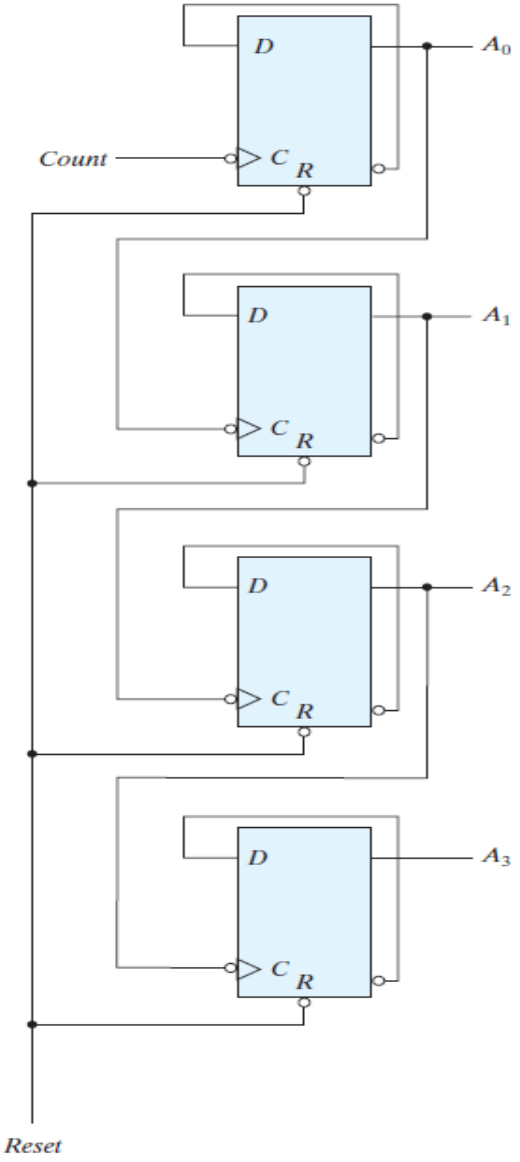
Counters

- A register that goes through a prescribed sequence of states upon the application of input pulses is called a *counter*
- The input pulses may be clock pulses, or they may originate from some external source and may occur at a fixed interval of time or at random
- The sequence of states may follow the binary number sequence or any other sequence of states
- Counters are available in two categories: ripple counters and synchronous counters
- In a ripple counter, a flip-flop output transition serves as a source for triggering other flip-flops
- In a synchronous counter, the *C* inputs of all flip-flops receive the common clock

Ripple counter - binary

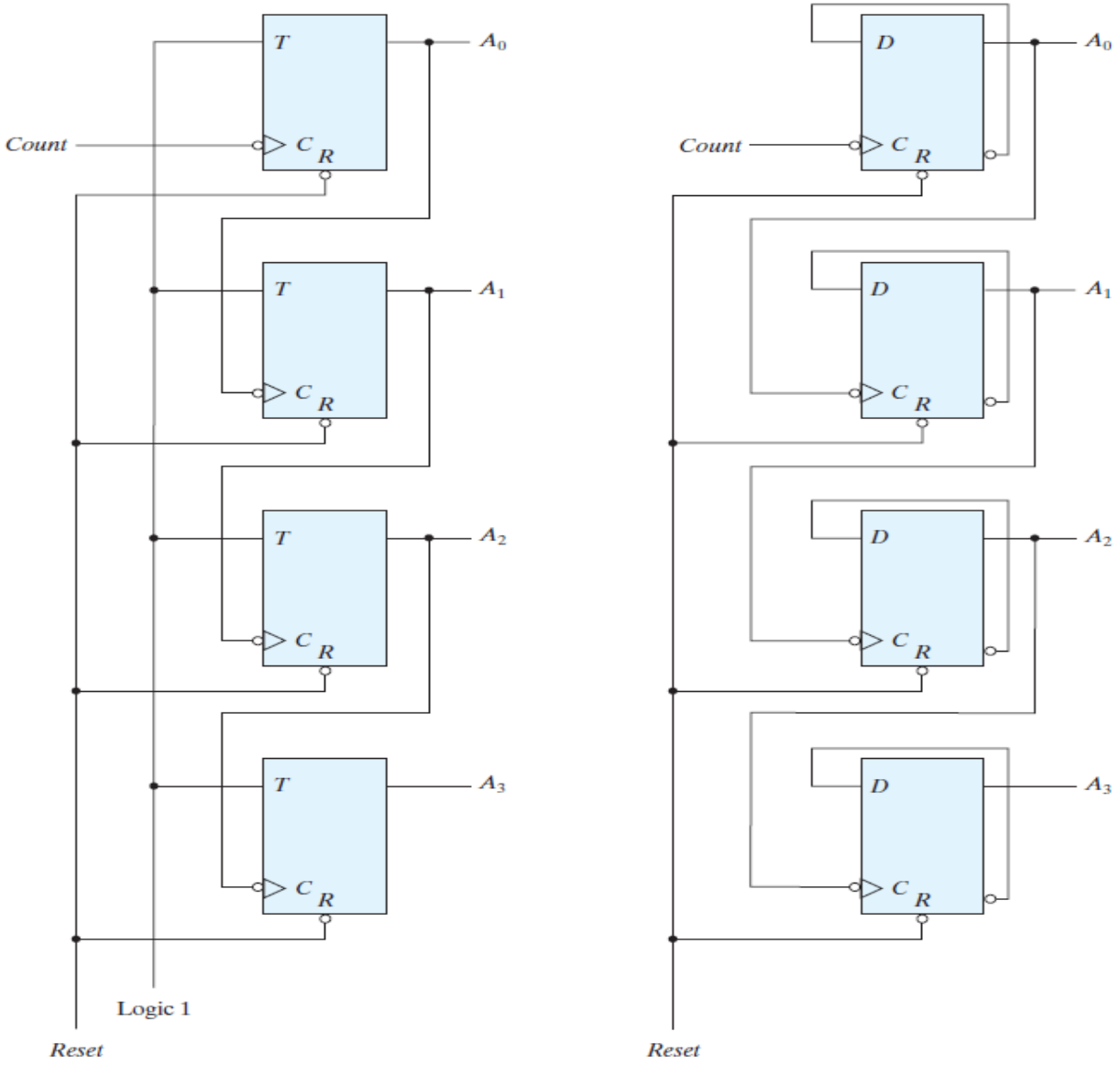
- A binary ripple counter consists of a series connection of complementing flip-flops, with the output of each flip-flop connected to the *C* input of the next higher order flip-flop
- The flip-flop holding the least significant bit receives the incoming count pulses
- A complementing flip-flop can be obtained from a *JK* flip-flop with the *J* and *K* inputs tied together or from a *T* flip-flop
- A third possibility is to use a *D* flip-flop with the complement output connected to the *D* input
- In this way, the *D* input is always the complement of the present state, and the next clock pulse will cause the flip-flop to complement





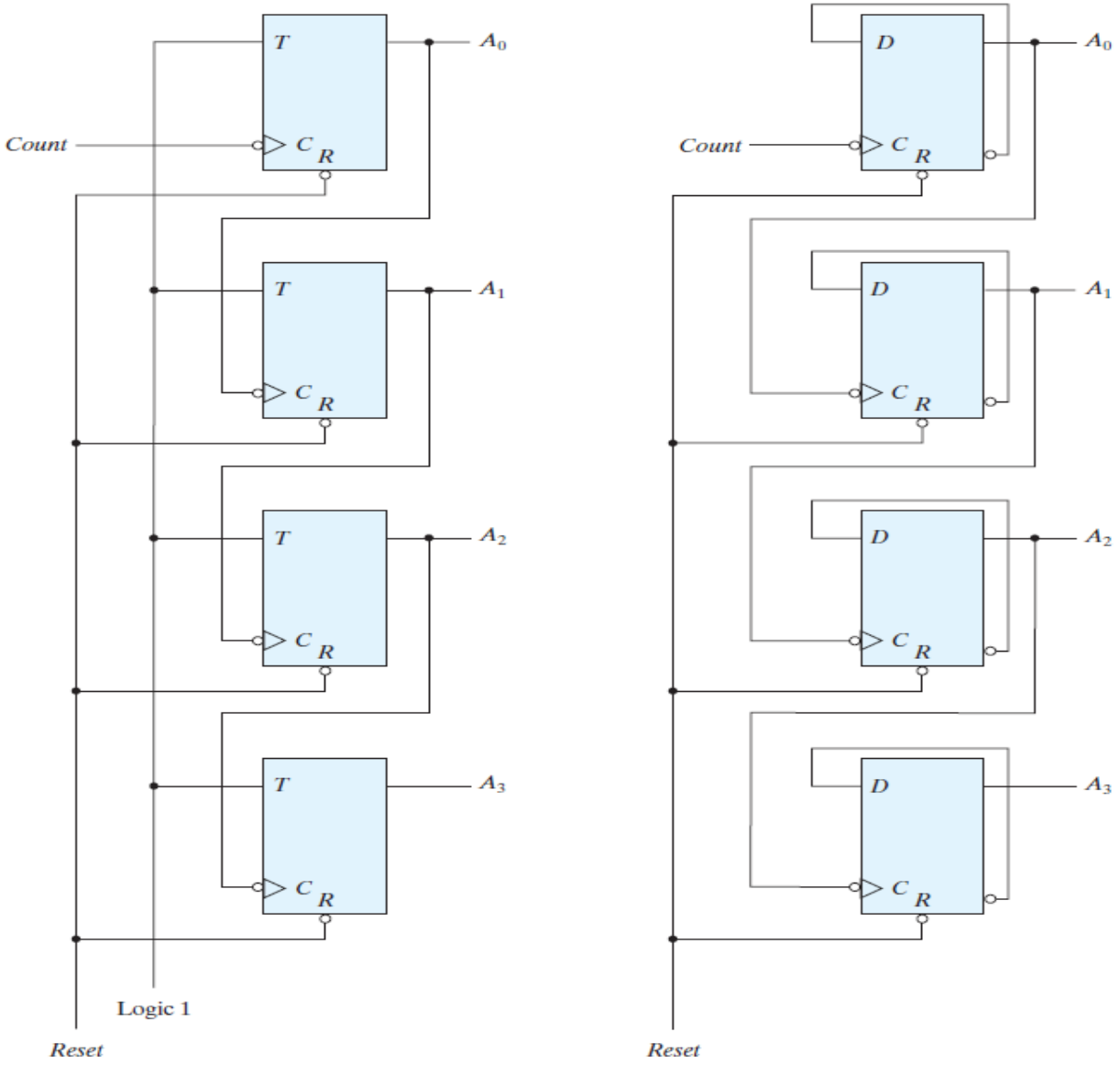
Ripple counter - binary

- The count starts with binary 0 and increments by 1 with each count pulse input
- After the count of 15, the counter goes back to 0 to repeat the count
- The least significant bit, A_0, is complemented with each count pulse input
- Every time that A_0 goes from 1 to 0, it complements A_1 and so on…



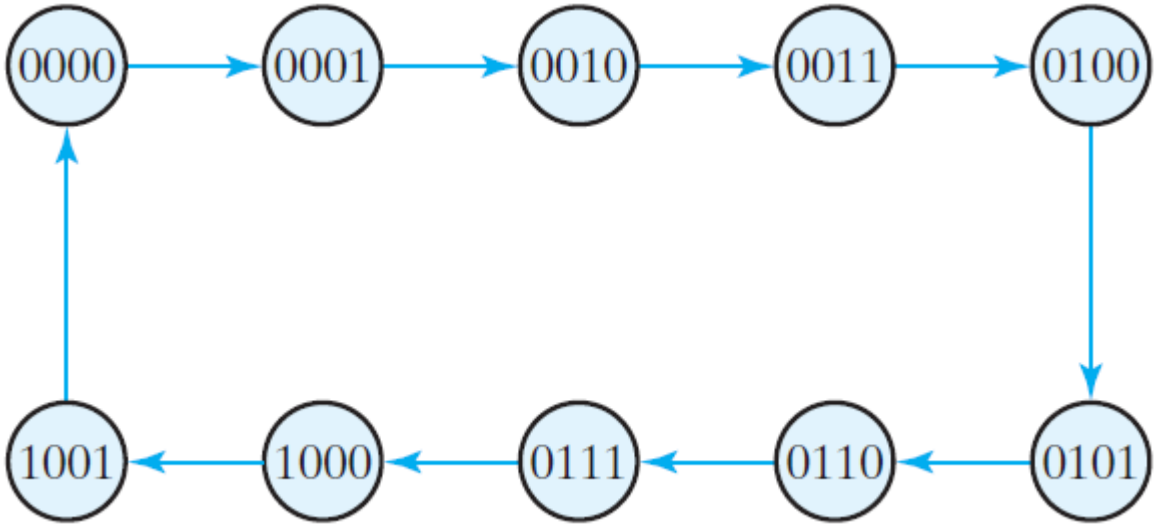
Ripple counter - binary

- For example, consider the transition from count 0011 to 0100
- A_0 is complemented with the count pulse
- Since A_0 goes from 1 to 0, it triggers A_1 and complements it
- As a result, A_1 goes from 1 to 0, which in turn complements A_2, changing it from 0 to 1
- A_2 does not trigger A_3, because A_2 produces a positive transition, and the flip-flop responds only to negative transitions



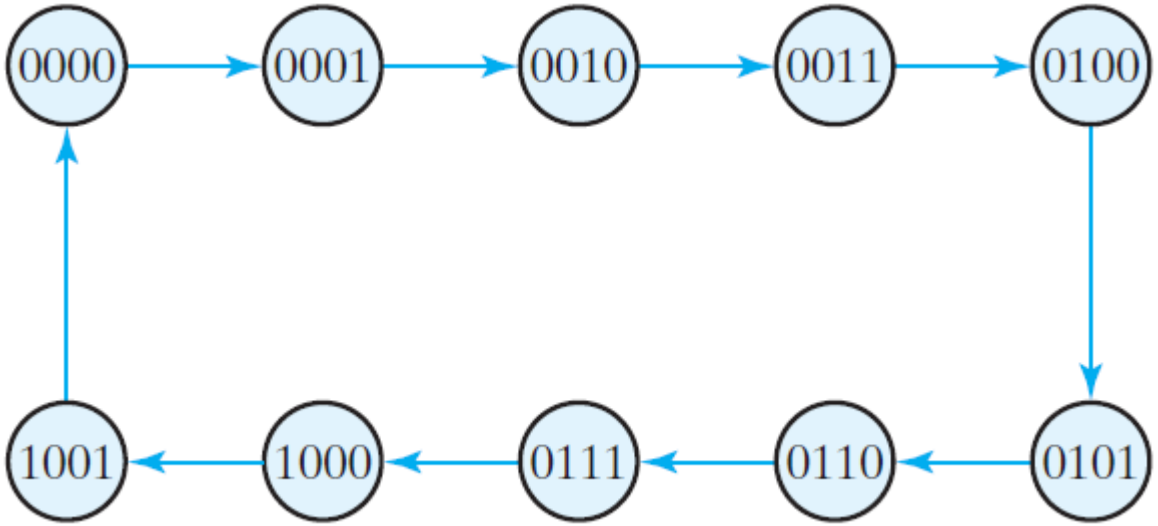
Ripple counter - BCD

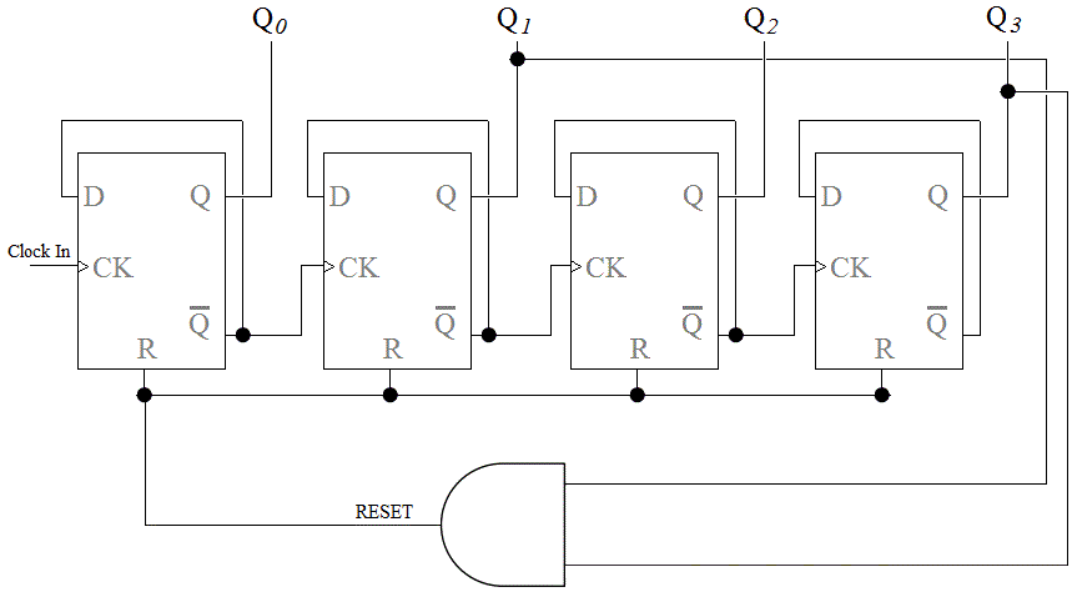
- A decimal counter follows a sequence of 10 states and returns to 0 after the count of 9
- Such a counter must have at least four flip-flops to represent each decimal digit, since a decimal digit is represented by a binary code with at least four bits
- The sequence of states in a decimal counter is 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

- We can obtain a decade counter by clearing all the flip-flops as soon as the state 1010 is obtained
- This can be done with the asynchronous input of CLR
- The condition for 1010 is checked by ANDing Q_3 and Q_1
- This is a very commonly used counter for making clocks/timer circuits
- Can we make it stop at any other number?





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 circuit
- To count from 0 to 999, we need a three-decade counter circuit
- 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