CMPE 212
Principles of Digital Design

Lecture 20

Modular Sequential Logic

April 6, 2016

www.csee.umbc.edu/~younis/CMPE212/CMPE212.htm

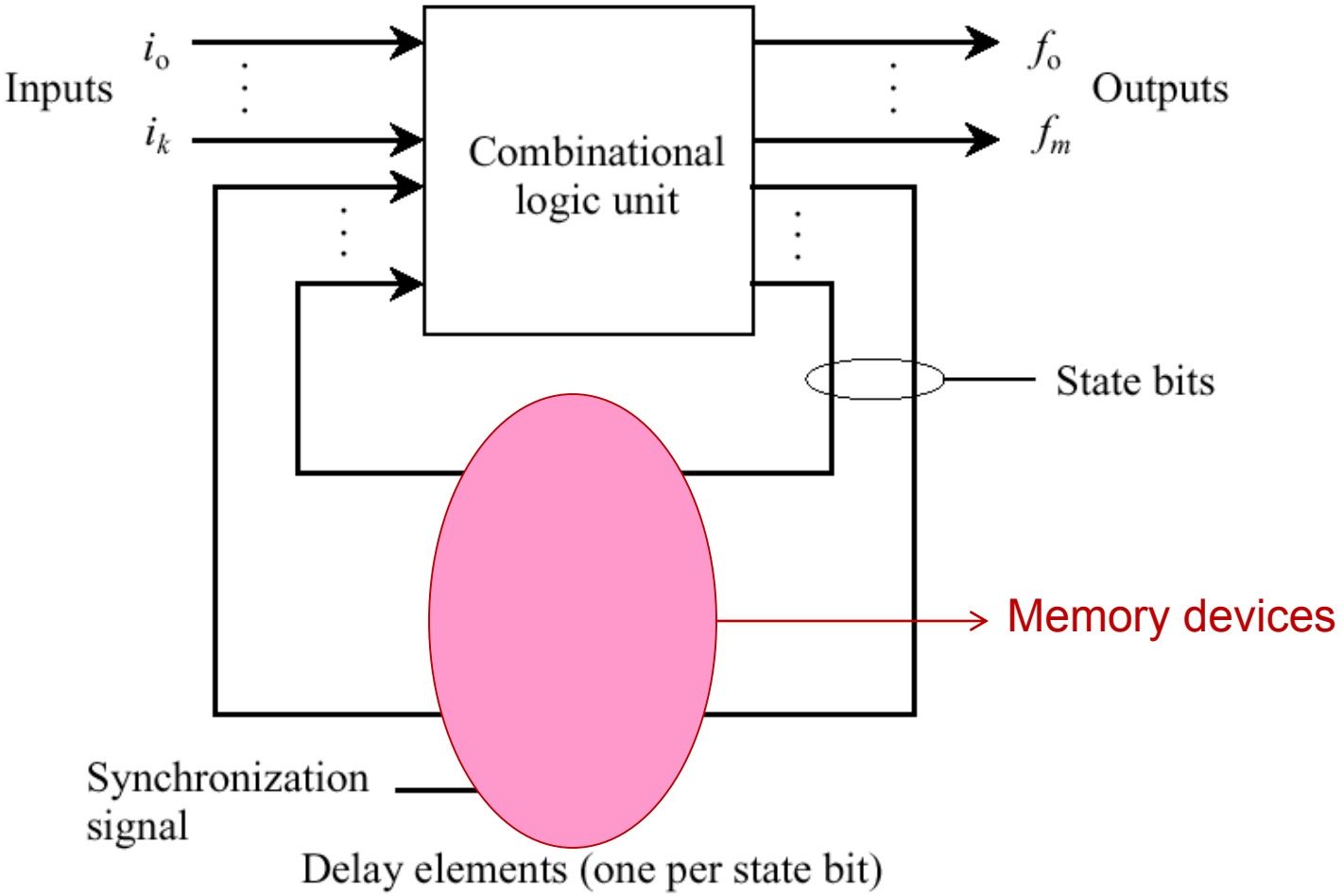
Lecture's Overview

- ***Previous Lecture***
  - ➔Introduction to sequential circuits
    (Concept, effect of feedback on combinational circuits, finite state machines, example applications)
  - ➔ Memory devices
    (S-R Latches, D-Latches, J-K and Toggle latches)
  - ➔ Flip-Flops
    (Role of clock, timing behavior diagrams)
  - ➔ Clocked sequential circuits
    (master-slave flip flops, level and edge triggered flip flops)
- ***This Lecture***
  - ➔ Modular Sequential Logic
    - Registers
    - Shift Registers
    - Synchronous and asynchronous counters

Popular Sequential Logic Modules

- The combinational logic circuits we have been studying so far have no memory ➔ The outputs always follow the inputs
- These are referred to as finite state machines, because they can have at most a finite number of states
- A finite state machine responds to an input by generating an output that is function of that input and the current state
- Finite state machine are commonly used for the design of digital controllers (typically called sequencer)
- Sequential circuits combines both combinational logic and memory devices
- Like combinational logic, there exist sequential logic modules that are available as MSI components, e.g., 74XX series
- Shift registers and counters are the most popular sequential logic modules that are used in digital designs

Classical Model of a Finite State Machine

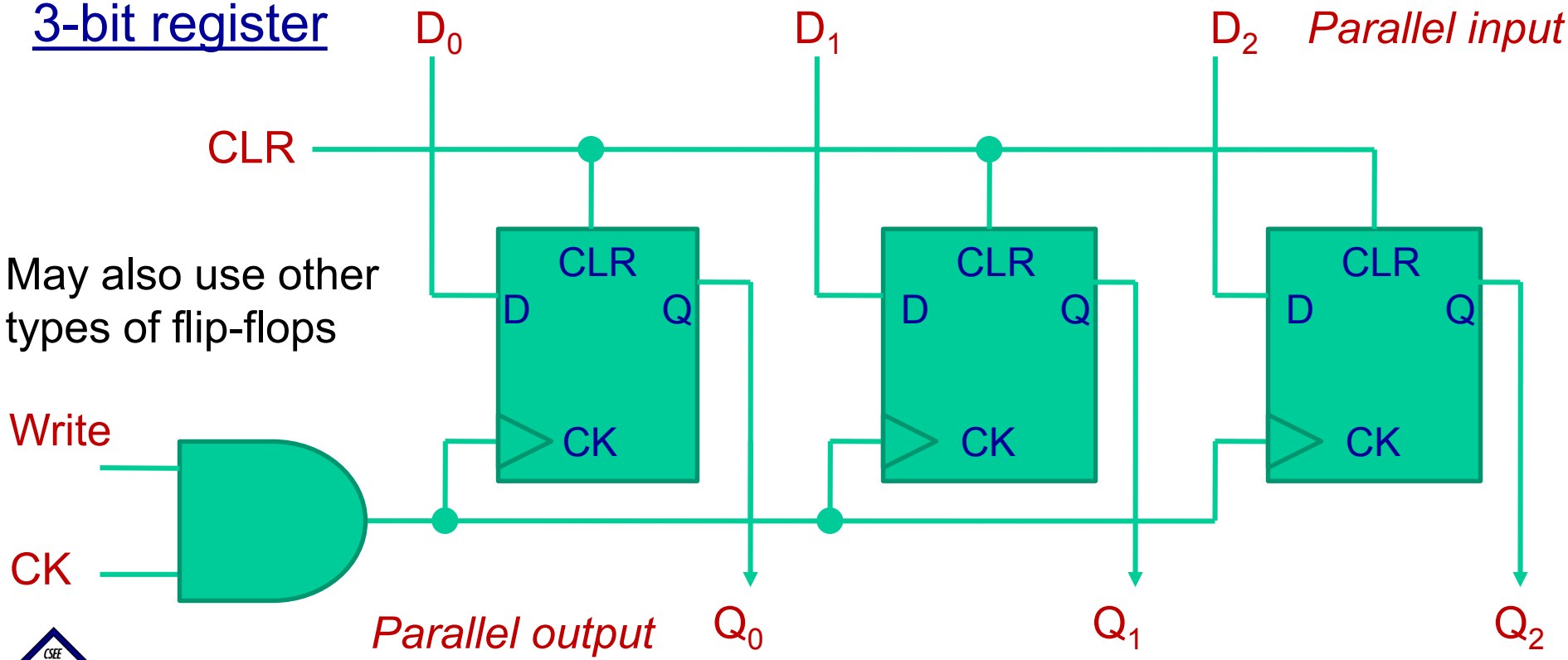


A FSM is composed of a combinational logic unit and delay elements (called latches or flip-flops) in a feedback path, which maintains state information

* Slide is courtesy of M. Murdocca and V. Heuring

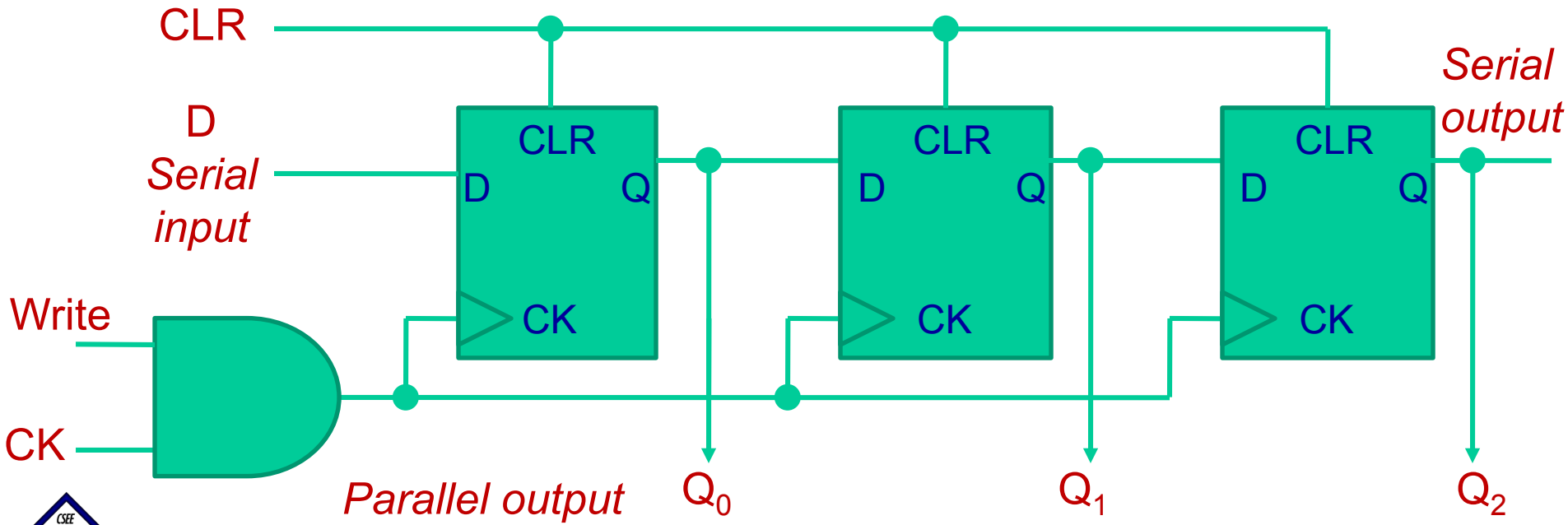
Register

- A register is composed of a set of flip-flops that stores parallel data, i.e., latches all bits at the same time at the clock edge
- Parallel output, i.e., all bits are available at the same time
- The register size is categorized by the number of stored bits

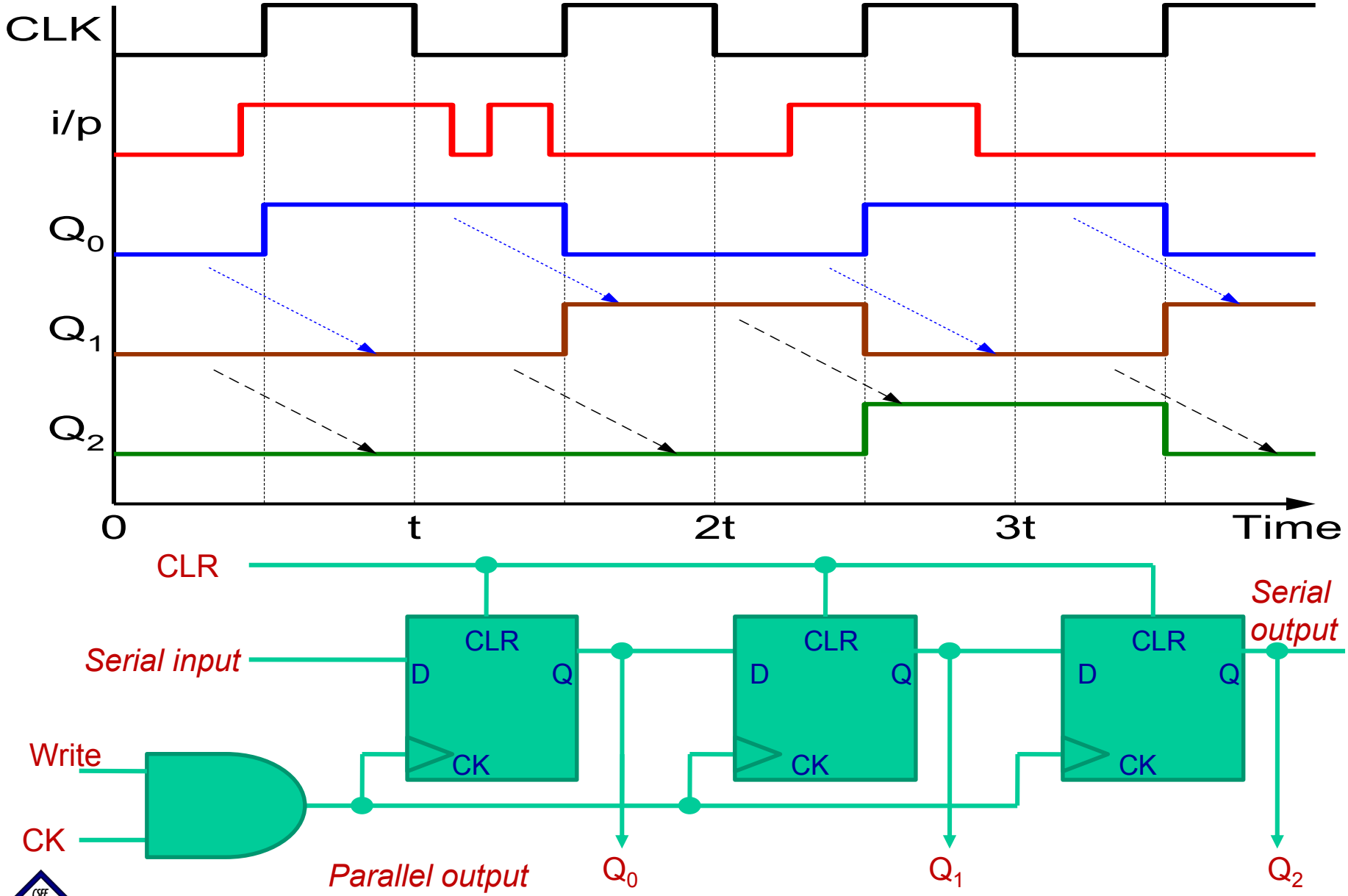


Shift Registers

- A shift register stores serial data bits and provides both serial and parallel output
- The serial output is simply a delayed bit sequence of the serial input with a delay that is proportional to the size of the register
- Numerous standard MSI devices are available, e.g., 7491A, 74164, etc. Some may not provide all features though
- The shift can be to the right or the left, with option to preload

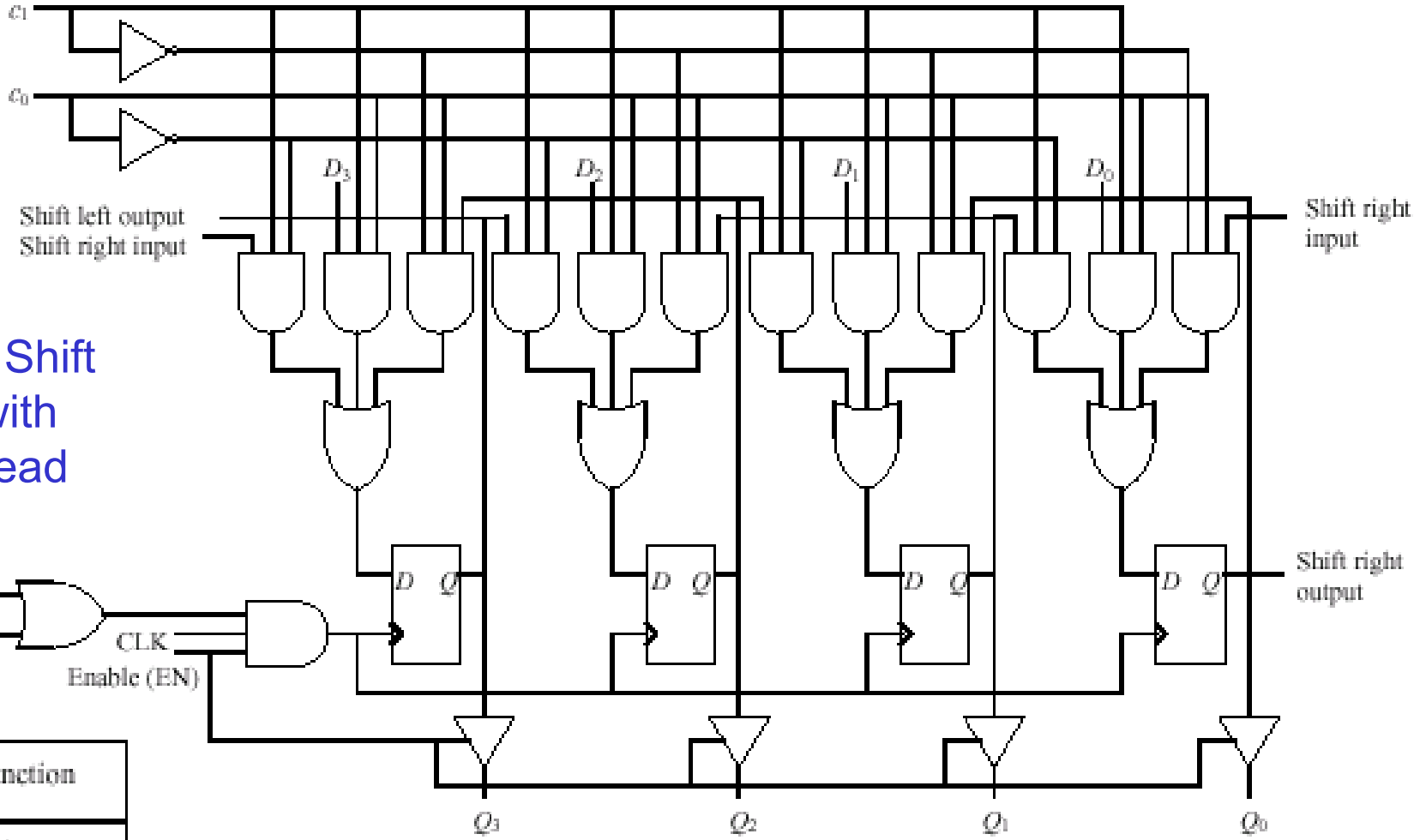


Timing Diagram Shift-Right

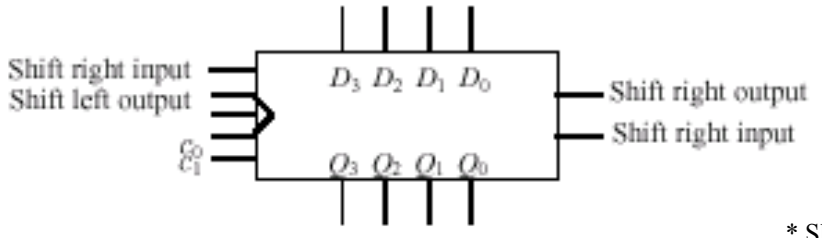


Left-Right Shift Reg. with Parallel Load

Left-Right Shift Register with Parallel Read and Write



Control c_1	Control c_0	Function
0	0	No change
0	1	Shift left
1	0	Shift right
1	1	Parallel load



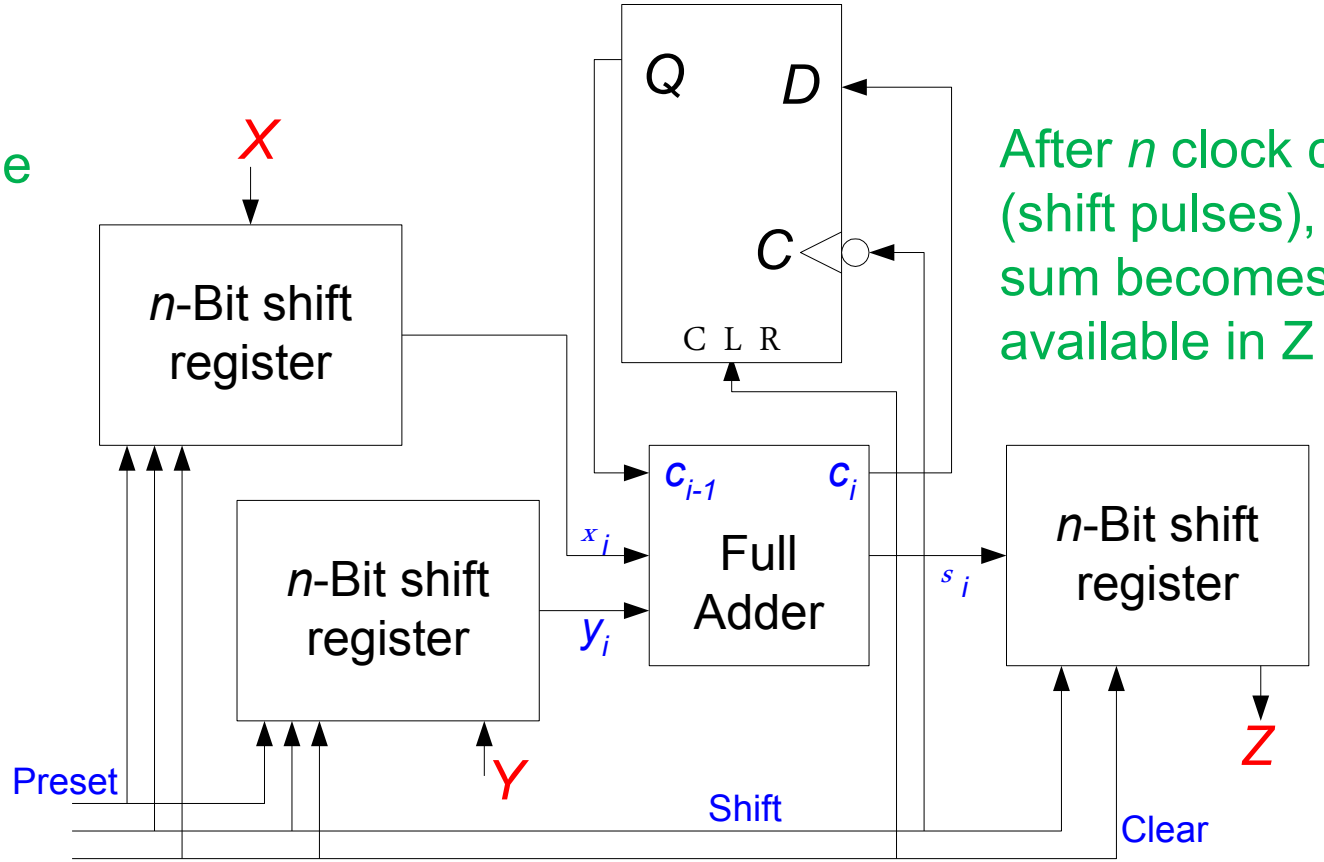
* Slide is courtesy of M. Murdocca and V. Heuring

Applications of Shift Registers

Load X and Y in parallel to the corresponding shift registers

Clear, Preset and shift are control lines

After *n* clock cycles (shift pulses), the sum becomes available in Z



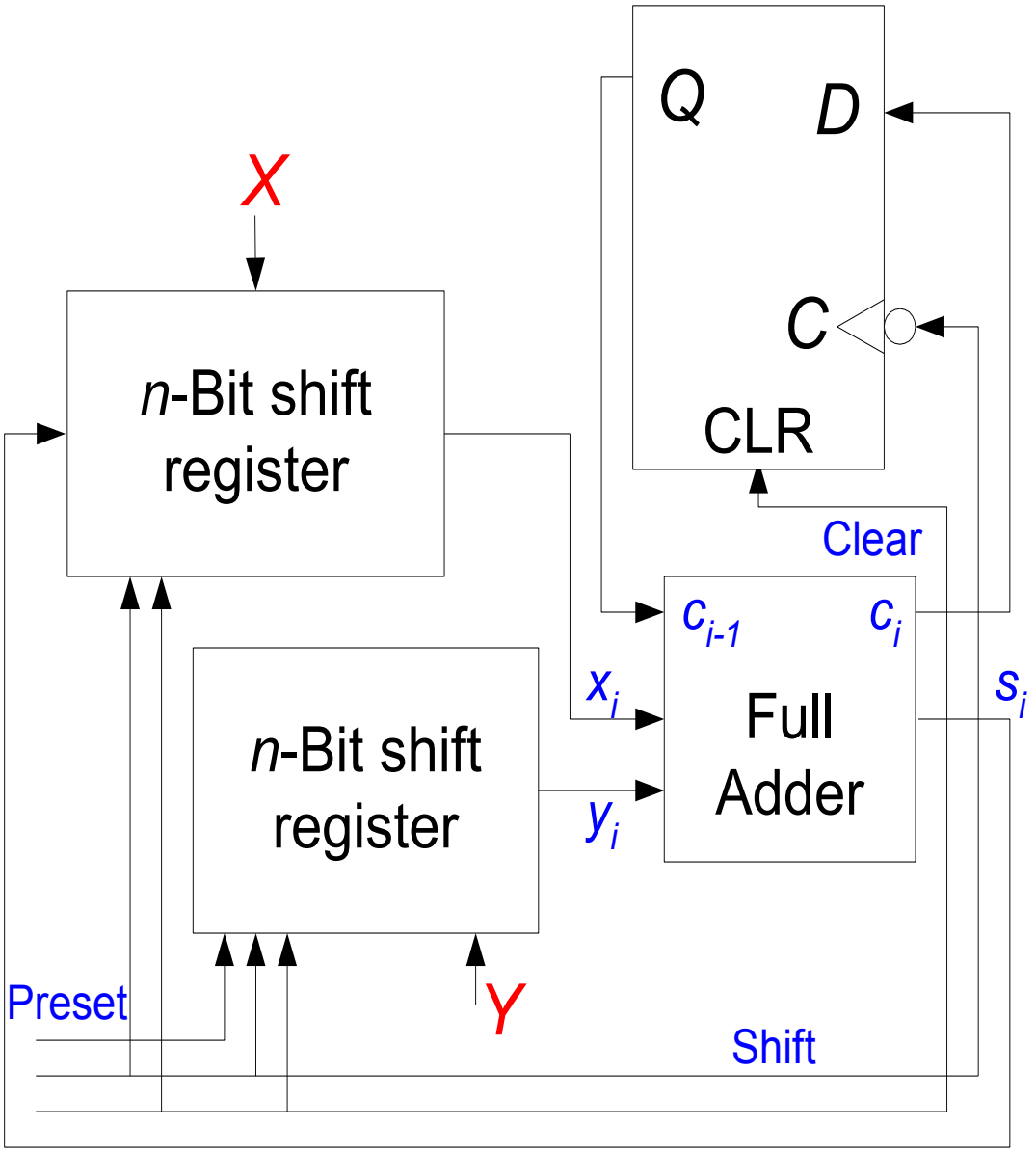
❶ Serial Adder:

1. Shift right to add one bit at time using the full adder and save in shift register that corresponds to Z
2. Store the carry-out in cycle "*i*" in the D flip-flop to be used in the clock cycle "*i+1*" as carry-in for the next bit

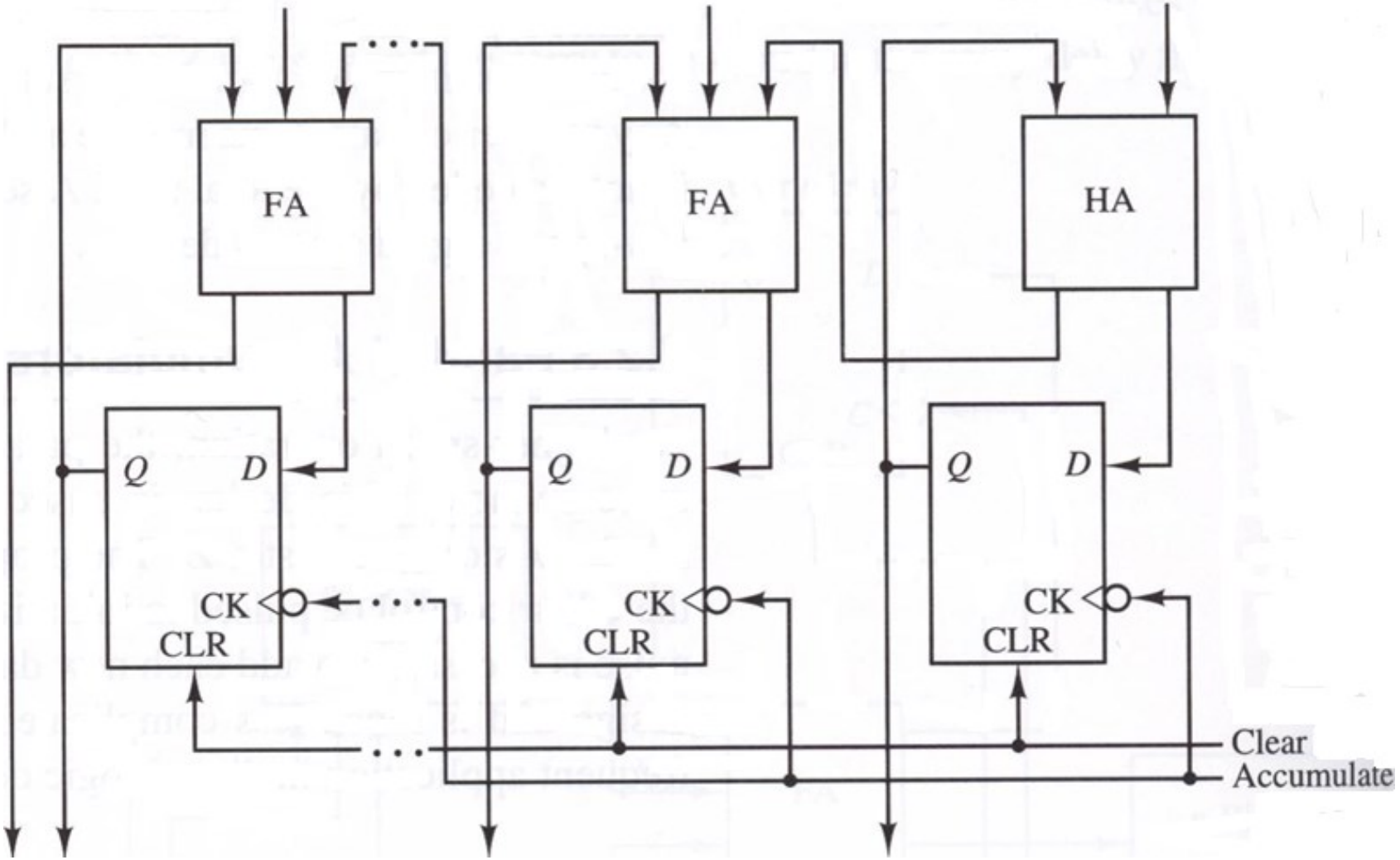
Applications of Shift Registers

❷ Serial Accumulator:

- Simply a serial adder for which the serial output is reused as a serial input to the accumulator
- The shift right frees up the left most significant bit in X to host the least significant bit of Z.
- After *n* iterations, Z will be stored in X
- Preset line is not used for X (need a different signal to reset the accumulator)
- The rest of the circuit is similar to the serial adder

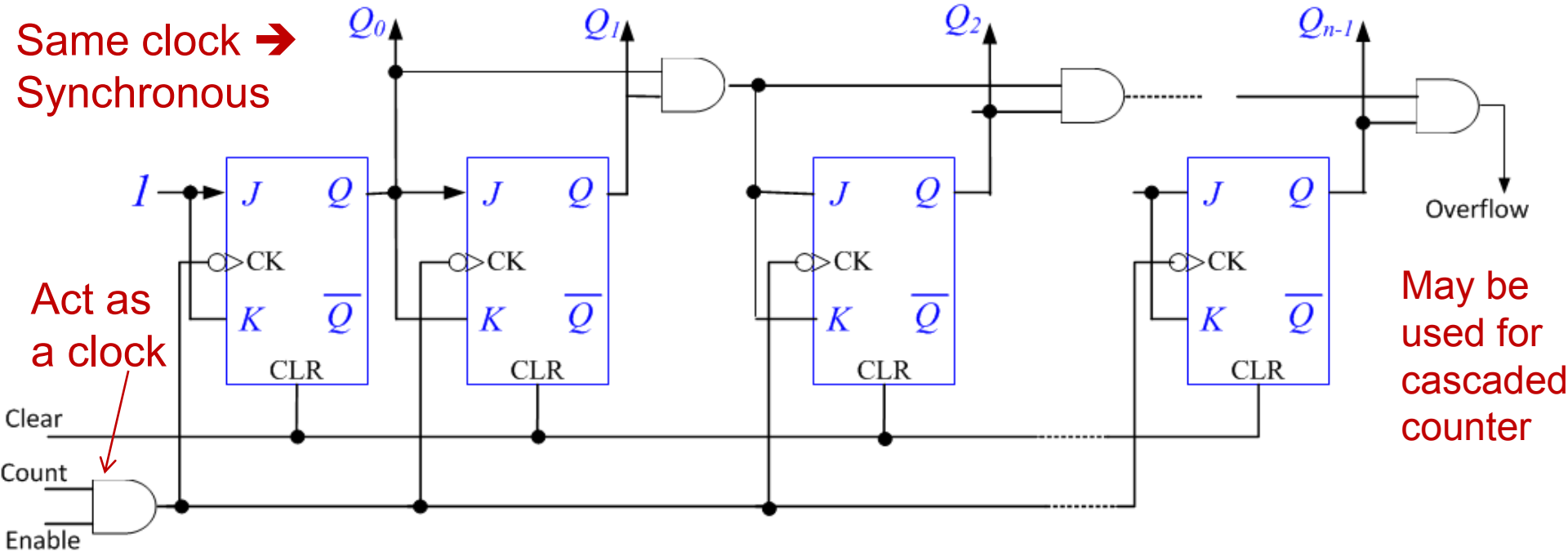


Parallel Accumulator Design

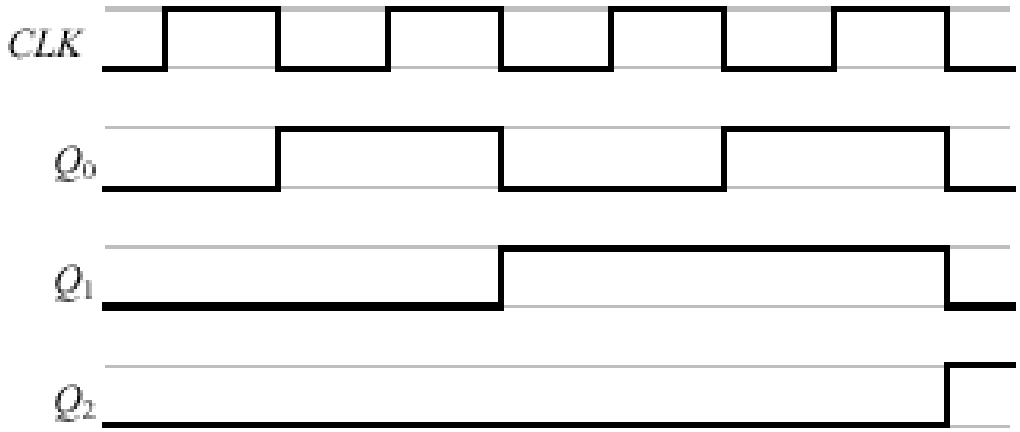


- No shift registers, just a register is used to store the accumulated values
- Faster operation (adding *n*-bits rather than one bit at a time)

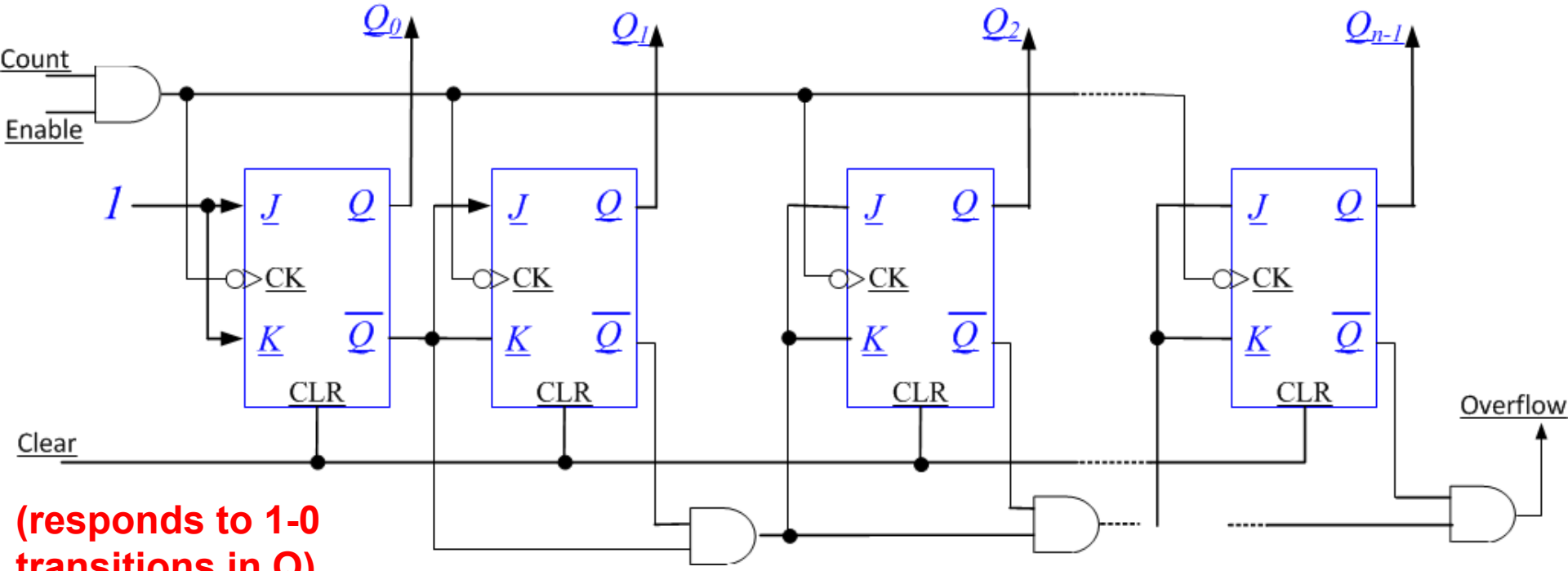
Synchronous Binary Counters



- Synchronous counters start at zero and increment every clock cycle (until 2^n-1 and then recycle)
- T flip-flops, implemented as J-K's, are used to toggle the next flip-flop when its output is 1



Counting Down



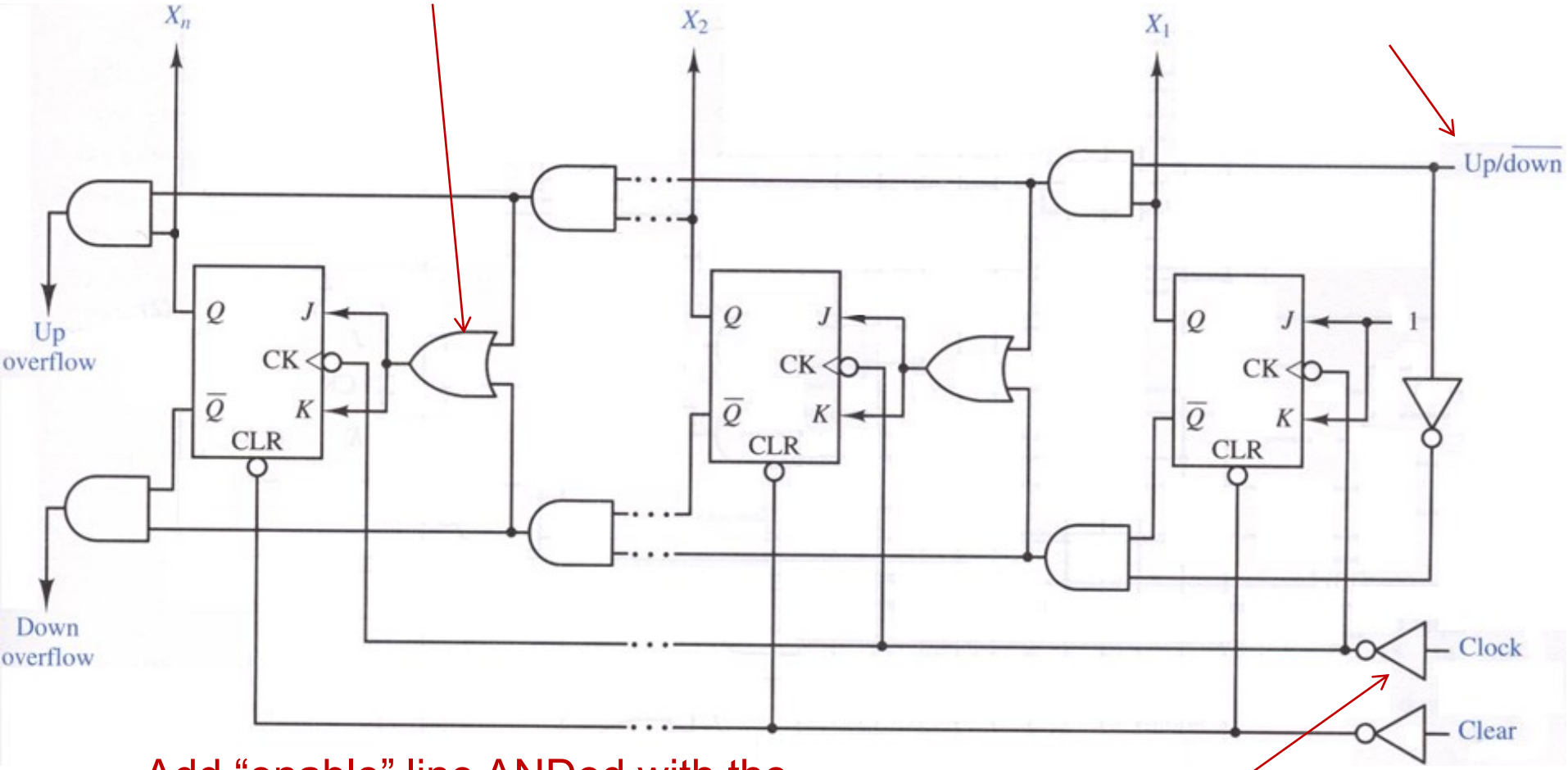
(responds to 1-0 transitions in Q)

- Counter starts at 2^n-1 and decrements every clock cycle until reaching zero and then resets back to 2^n-1)
- T flip-flops, implemented as J-K's, are used to toggle the next flip-flop when its output is 1.
- Connecting to J and K to $\overline{Q}$ instead of Q, triggers the transition on a change of $\overline{Q}$ in, i.e. change of Q from 0 to 1,

Synchronous Up/Down Counters

Factoring 1-0 or 0-1 transitions depends on the selection of the up or down mode

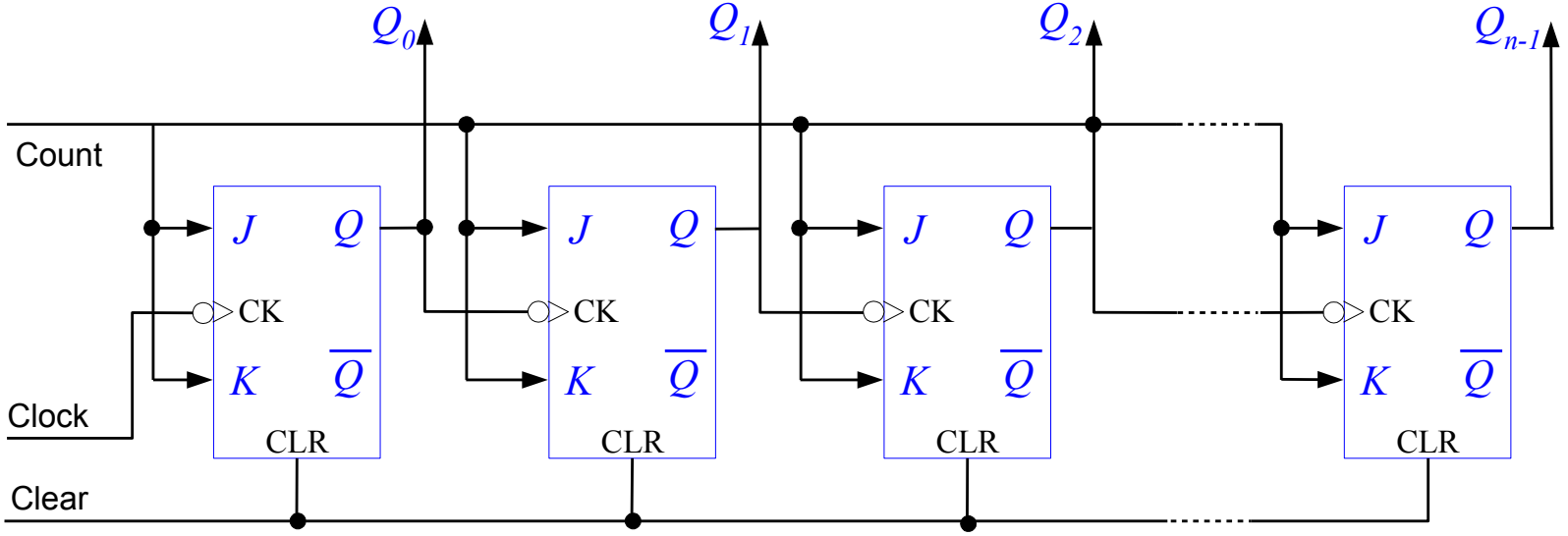
The counter is either counting up or down



Add "enable" line ANDed with the clock in order to freeze the state

Synchronous design

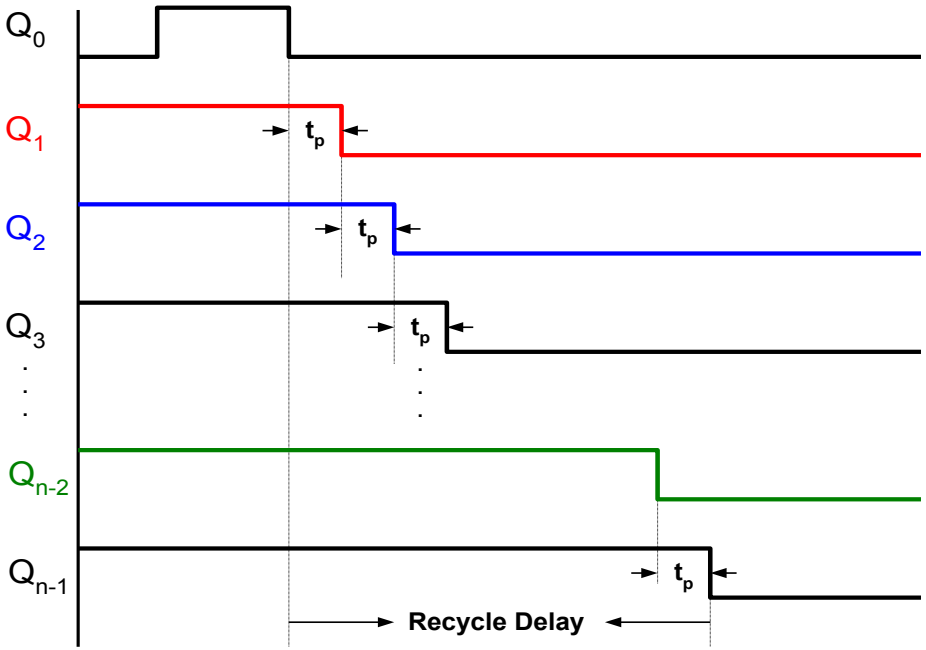
Asynchronous Binary Counters



Changes propagate from one flip-flop to the next

Asynchronous ➔ state change is not controlled by a synchronizing clock (ready-go operation)

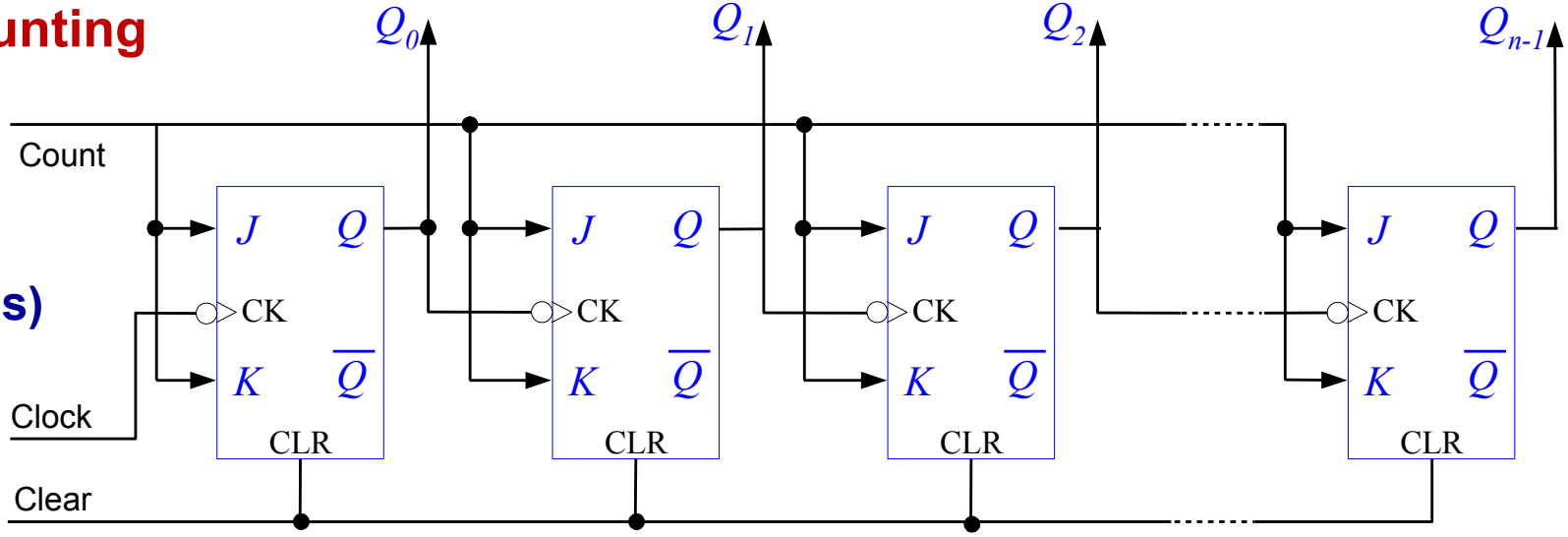
- Delay is not homogenous and one has to determine the worst case delay to avoid reading the wrong output
- Recycling back to zero is not instantaneous, *(n-1)t_p*

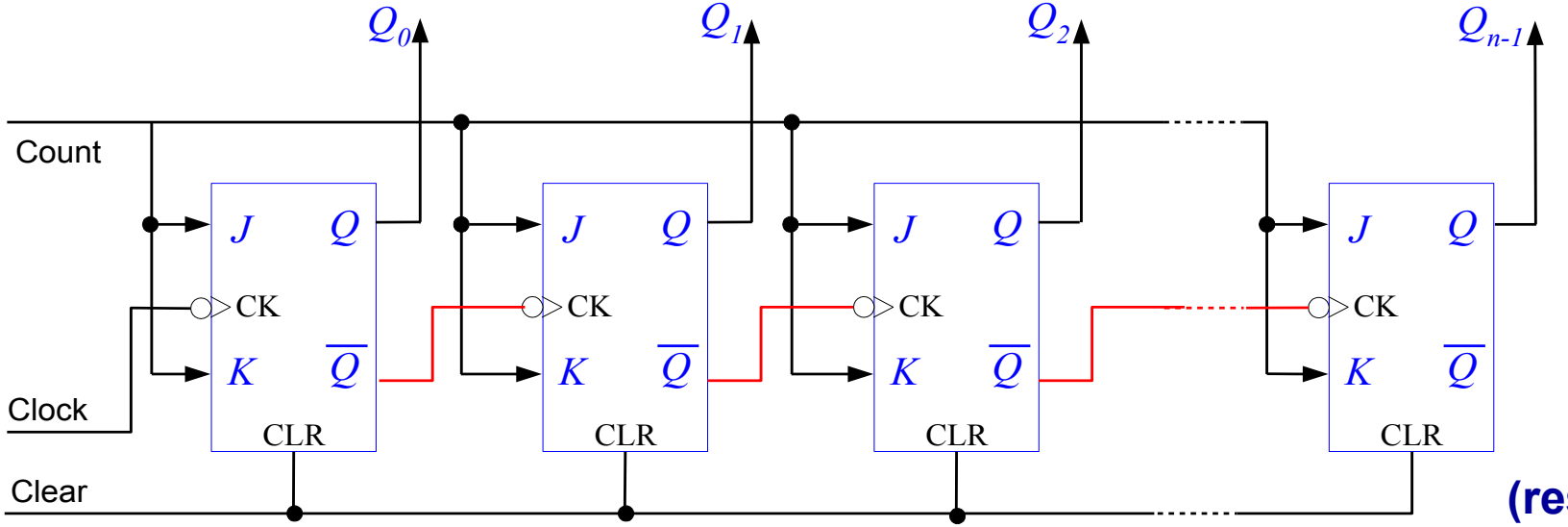


Down Counters

Forward counting

(responds to 0-1 transitions)



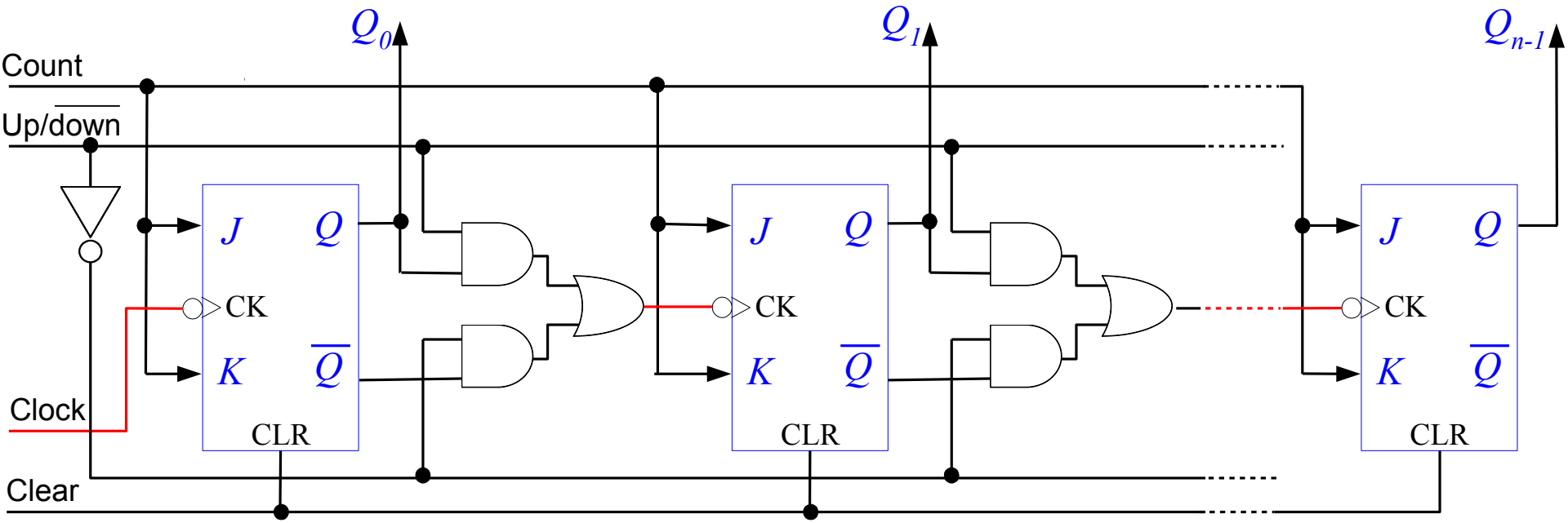


Backward counting

(responds to 1-0 transitions)

Need parallel load or preset line (not shown)

Asynchronous Up/Down Counters



- Counters are very popular module in sequential logic circuit
- An up/down counter is a simple finite state machine in which transitions from a particular state can be to the next (count-up), the previous (count-down) or the initial (CLR) states
- Generally synchronous design is much easier than asynchronous design; yet may not be as efficient

Conclusion

❑ *Summary*

➔Register and Shift Registers
(Design, supporting right and lift shift, parallel load)

➔ Applications of Shift Registers
(Serial adders, serial accumulators, parallel accumulators)

➔Binary counters
(synchronous and asynchronous counters, Down and up/down counters)

❑ *Next Lecture*

➔Modular Sequential Logic

- Modulo-N counters
- Ring counters
- Multiple sequence counters

Reading assignment: Sections 7.1 – 7.3 in the textbook