ECN 104

Digital Logic Design

Spring 2023

Tanmoy Pramanik

http://ece.iitr.ac.in/tanmoy_pramanik

Acknowledgment: Content mostly taken from textbook by Mano.

Registers with Parallel Load

Registers:

- Series of flip-flops that can hold binary data for processing
- Holds data until power is turned off
- Need simple options like loading new data/holding the data for multiple clock cycles etc.

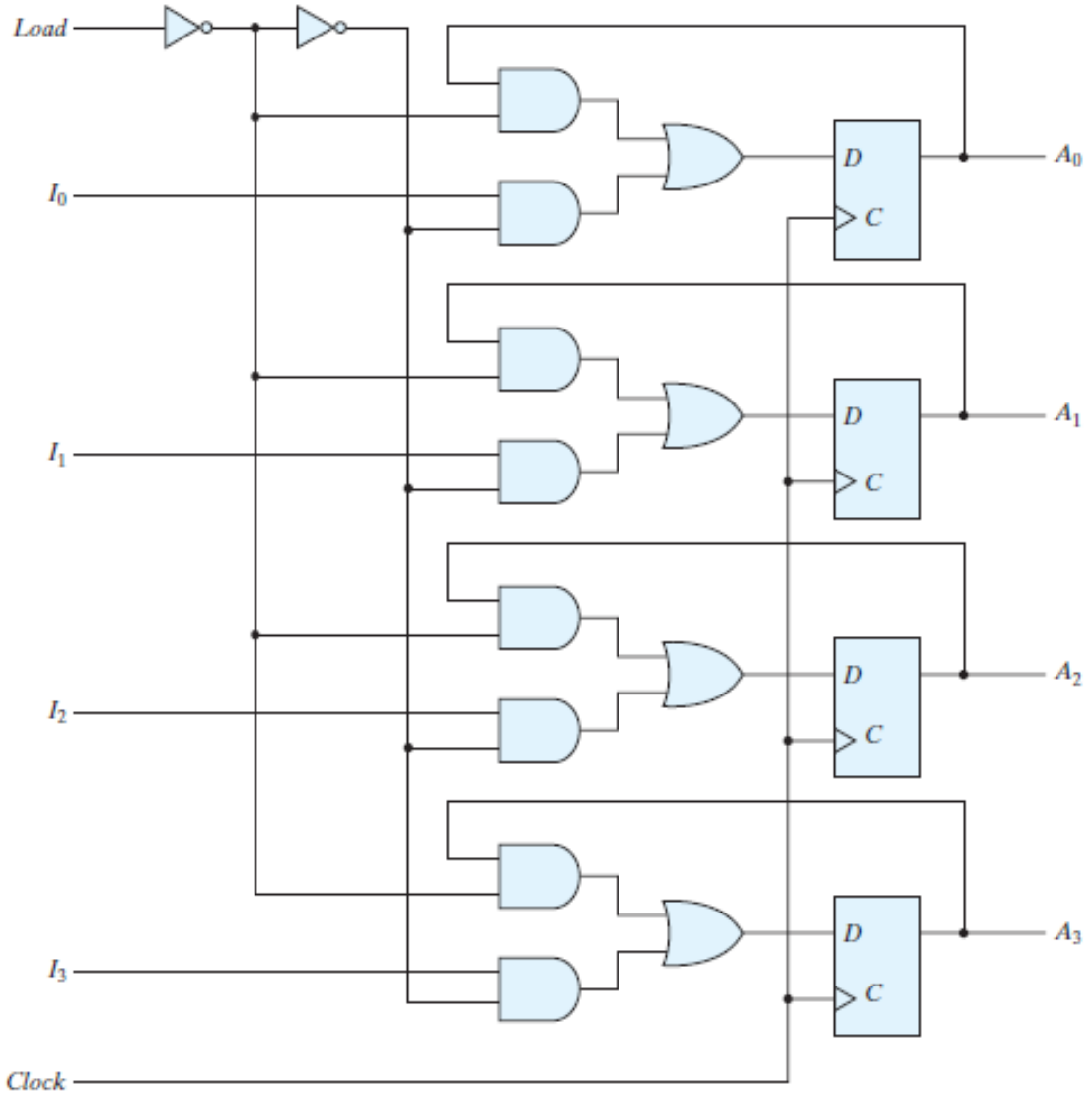


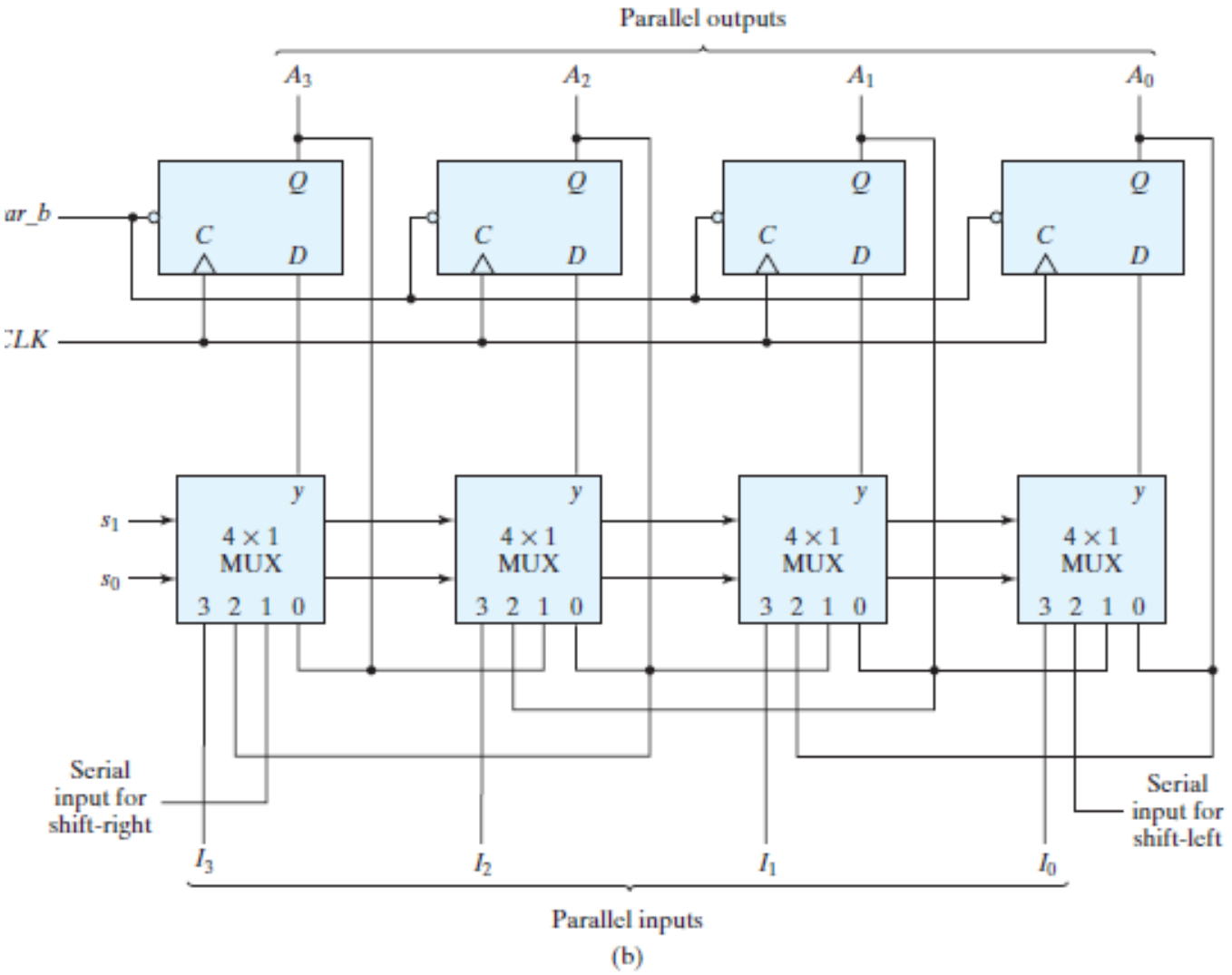
FIGURE 6.2
Four-bit register with parallel load

Universal Shift Register

Desired operations:

Table 6.3
Function Table for the Register of Fig. 6.7

Mode Control		
s_1	s_0	Register Operation
0	0	No change
0	1	Shift right
1	0	Shift left
1	1	Parallel load



Useful for various purposes

- Shifting is multiplying or diving by 2
- Convert serial data stream to parallel data for processing etc.

Serial Adder Circuit

- Area-efficient but slow adder

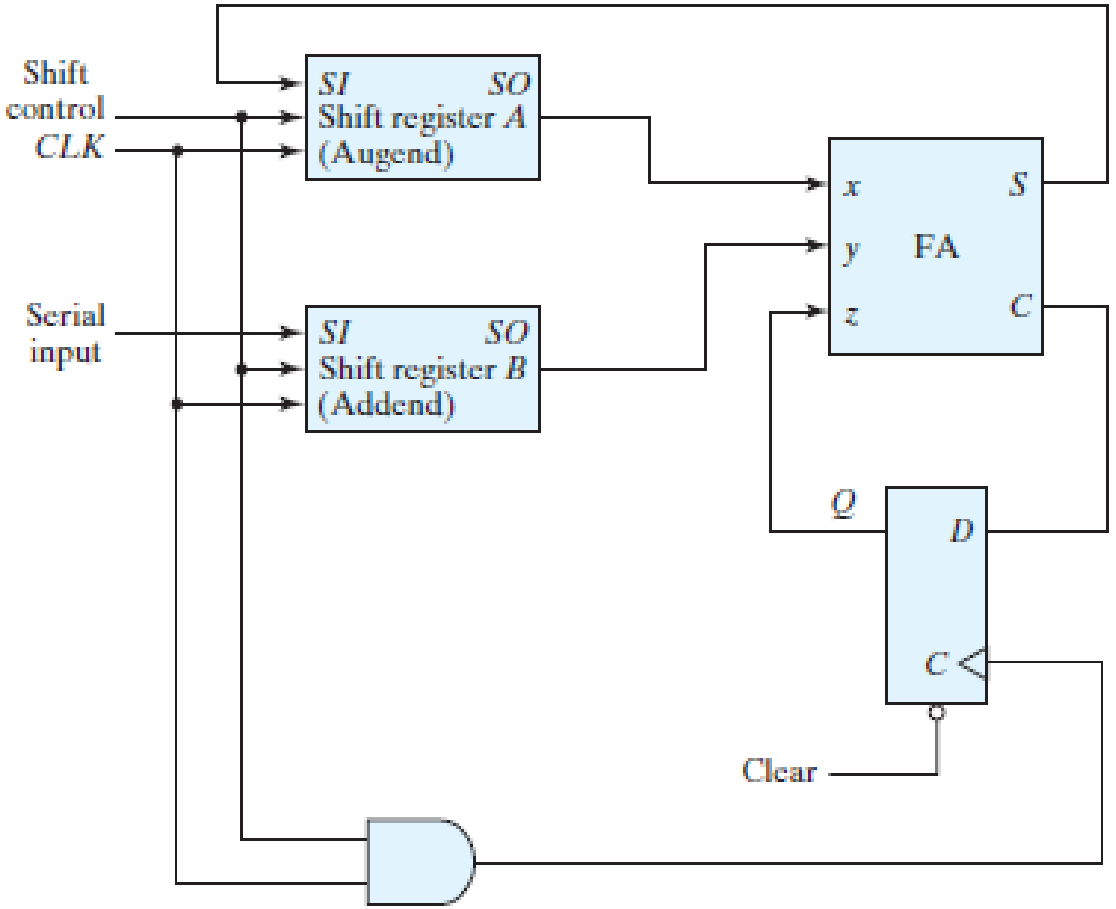


FIGURE 6.5
Serial adder

Synchronous Counters

- Counts the number of clock pulses
- Each flip-flop is connected to the clock signal

- Types
  - UP/DOWN Binary counter
  - BCD Counter
  - Arbitrary sequence - Modulo-N counters
  - Timing Generation - Ring counter
  - Johnson counter

Up/Down Counter

- Simple UP counter logic

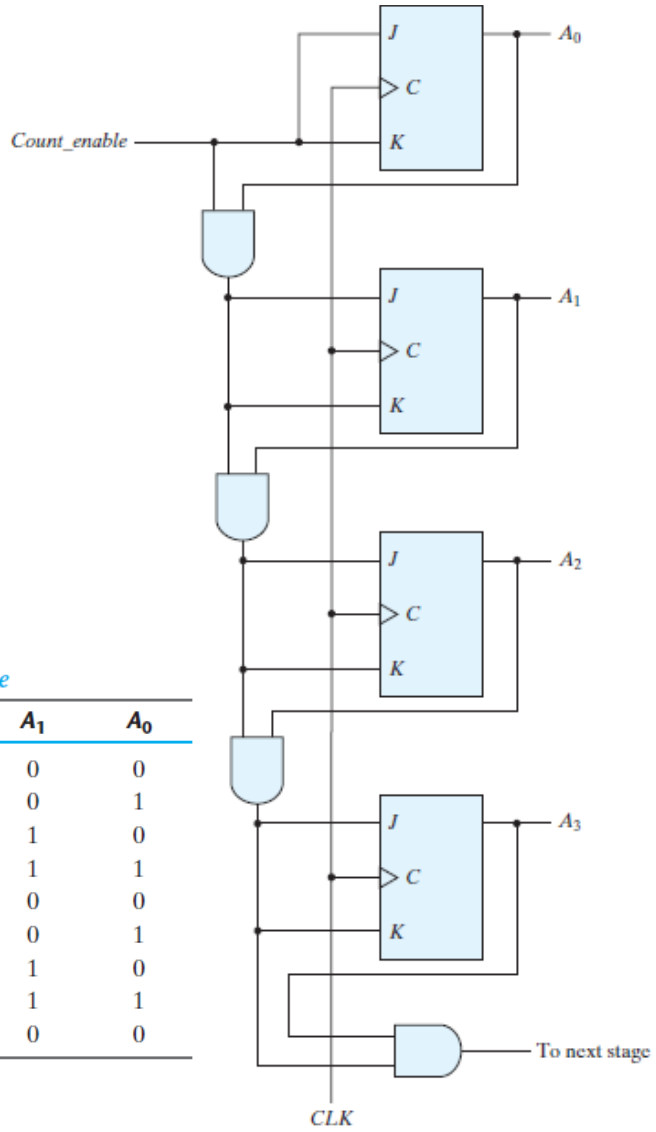
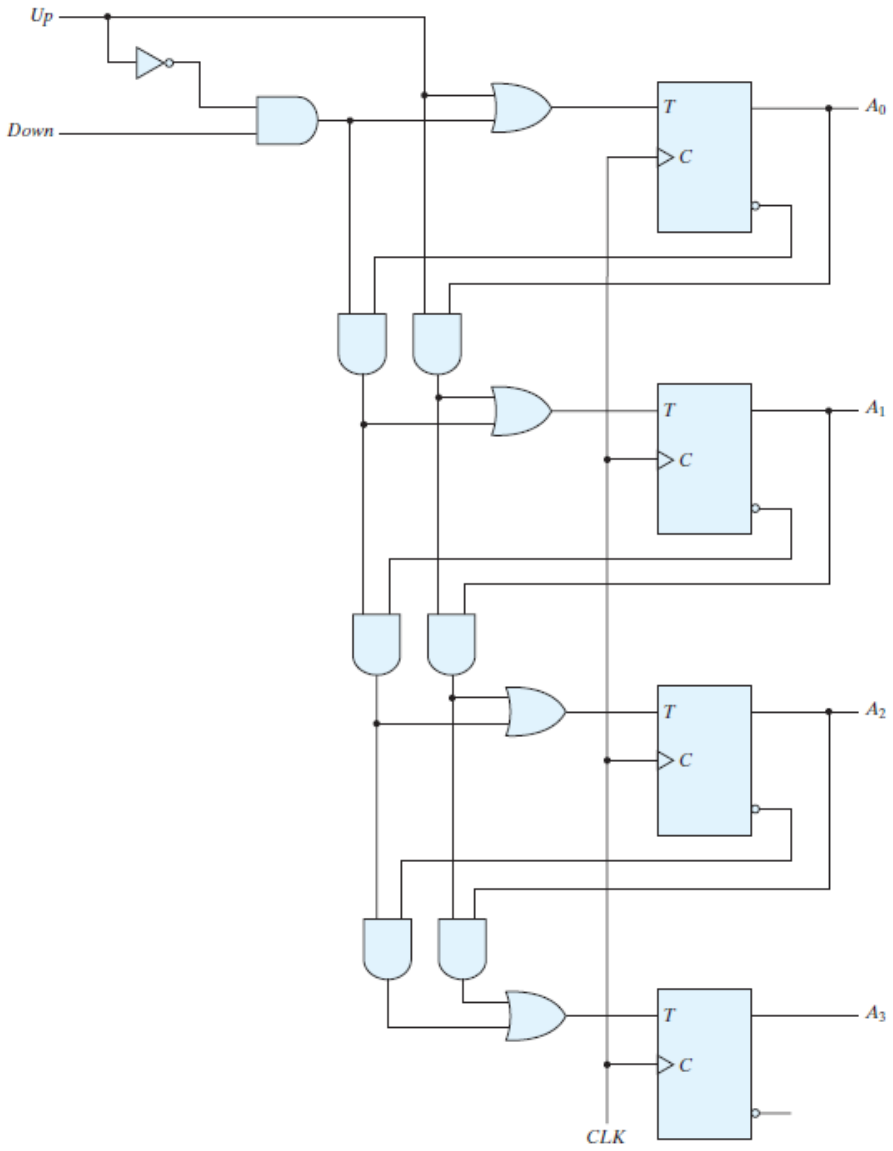


Table 6.4
Binary Count Sequence

A_3	A_2	A_1	A_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0

- Control knobs to count UP/DOWN or do nothing



BCD counter with T-flipflops

- Counts sequence from 0-9 in BCD coded format

Table 6.5
State Table for BCD Counter

Present State				Next State				Output	Flip-Flop Inputs			
Q_8	Q_4	Q_2	Q_1	Q_8	Q_4	Q_2	Q_1	y	TQ_8	TQ_4	TQ_2	TQ_1
0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	1	0	0	1	0	0	0	0	1	1
0	0	1	0	0	0	1	1	0	0	0	0	1
0	0	1	1	0	1	0	0	0	0	1	1	1
0	1	0	0	0	1	0	1	0	0	0	0	1
0	1	0	1	0	1	1	0	0	0	0	1	1
0	1	1	0	0	1	1	1	0	0	0	0	1
0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	0	0	1	0	0	0	0	1
1	0	0	1	0	0	0	0	1	1	0	0	1

Counter with parallel load/store

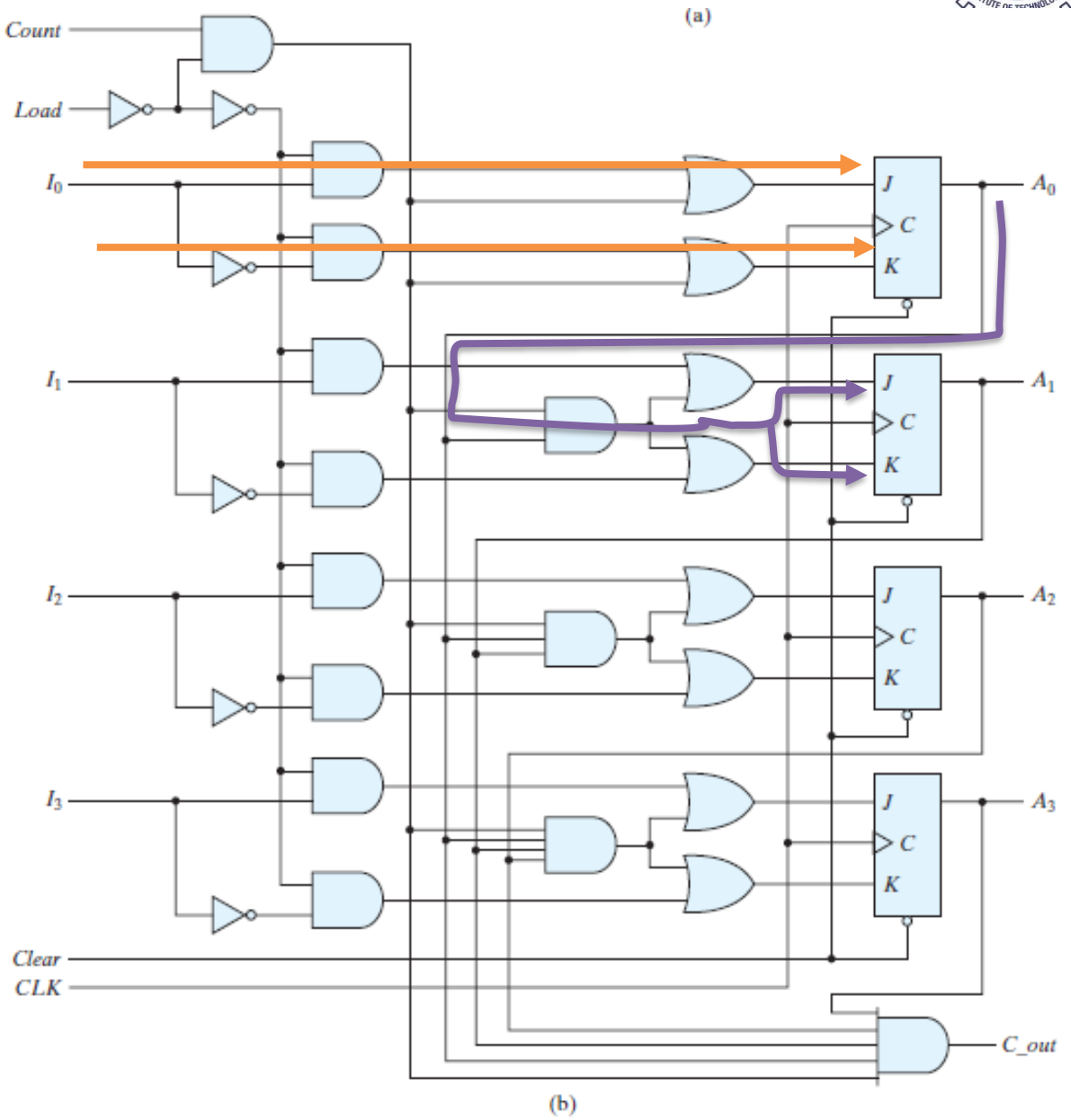
- Combination of counter + register functionality

If Count =0 and Load = 1

If Count =1 and Load = 0

If Count =0 and Load = 1

Hold state



Modulo-N counter

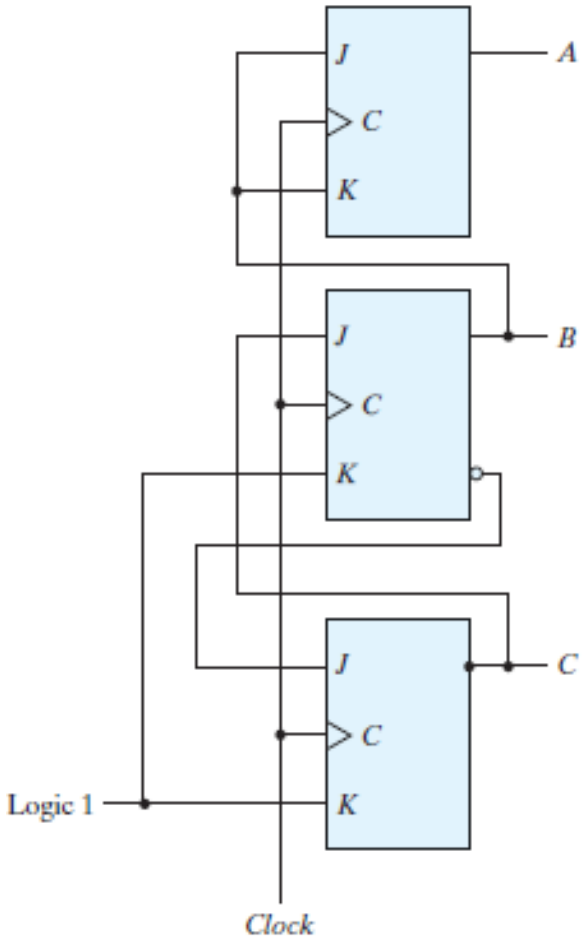
- A counter that goes through a repeated sequence of *N* states is called Modulo-N counter.
- MOD 3 : 00→01→10→00
- MOD 6: 0→1→2→3→4→5→0
- BCD counter is MOD 10 counter

- In general for "MOD N" counter you need *n* flip-flops such that $2^n \geq N$

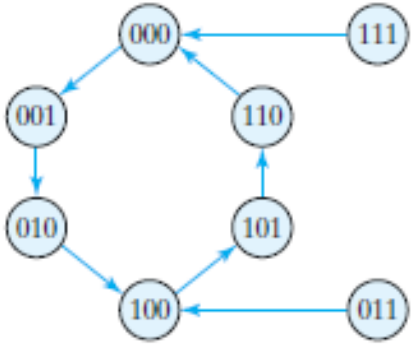
- The naming is also valid for asynchronous counters

Counter with unused states

- "Outside interference during its operation may cause the circuit to enter one of the unused states." How to resume normal operation?
- If the unused states are treated as don't-care conditions, then once the circuit is designed, it must be investigated to determine the effect of the unused states. "

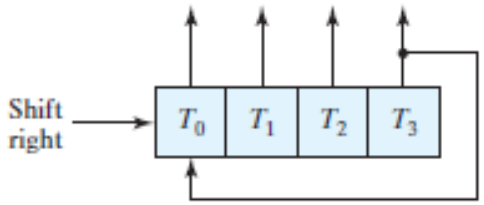


(a) Logic circuit diagram

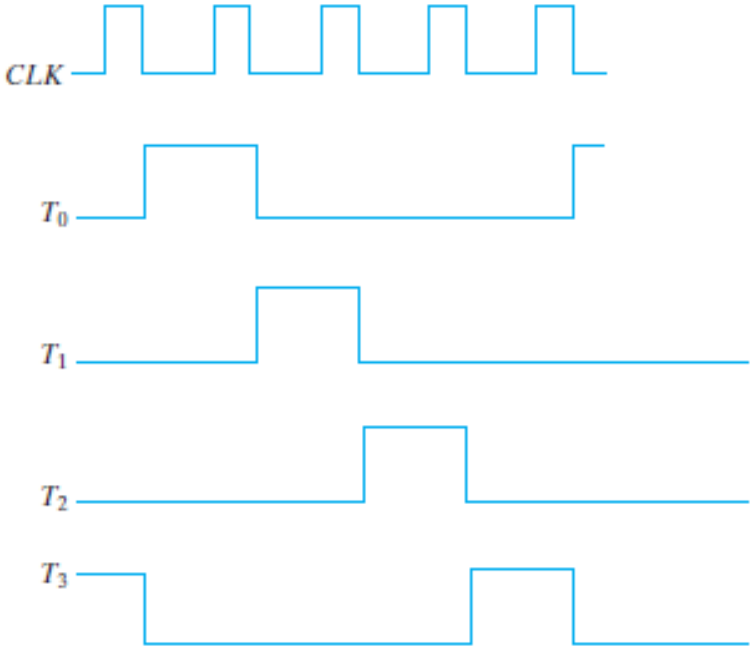


(b) State transition diagram

Ring Counter – Timing Signal Generation

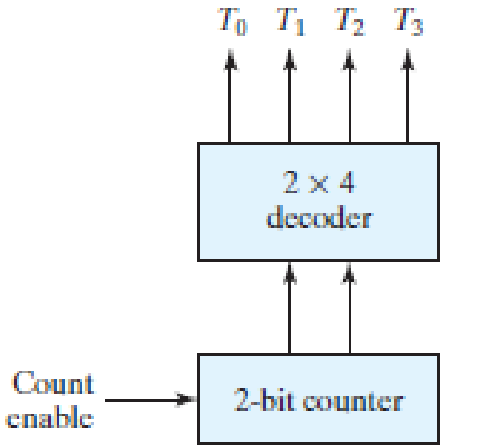


(a) Ring-counter (initial value = 1000)



(b) Sequence of four timing signals

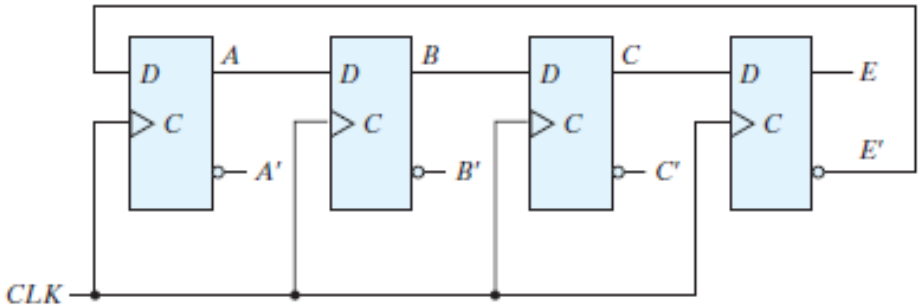
- All D flipflops
- Similar to the universal shift register with shift right activated
- Useful to generate timing signals – for example, for memory operations.
  - Alternative Design:



(c) Counter and decoder

Johnson Counter

- "A *k* -bit ring counter circulates a single bit among the flip-flops to provide *k* distinguishable states. The number of states can be doubled if the shift register is connected as a ***switch-tail*** ring counter. A switch-tail ring counter is a circular shift register with the complemented output of the last flip-flop connected to the input of the first flip-flop."



(a) Four-stage switch-tail ring counter

Sequence number	Flip-flop outputs A	B	C	E	AND gate required for output
1	0	0	0	0	$A'E'$
2	1	0	0	0	AB'
3	1	1	0	0	BC'
4	1	1	1	0	CE'
5	1	1	1	1	AE
6	0	1	1	1	$A'B$
7	0	0	1	1	$B'C$
8	0	0	0	1	$C'E$

(b) Count sequence and required decoding

k -bit counter → 2*k* states (with decoders made of 2-input gates only)

How do you derive the decoding logic?

What happens if the counter goes to the incorrect state?

Asynchronous/Ripple counters

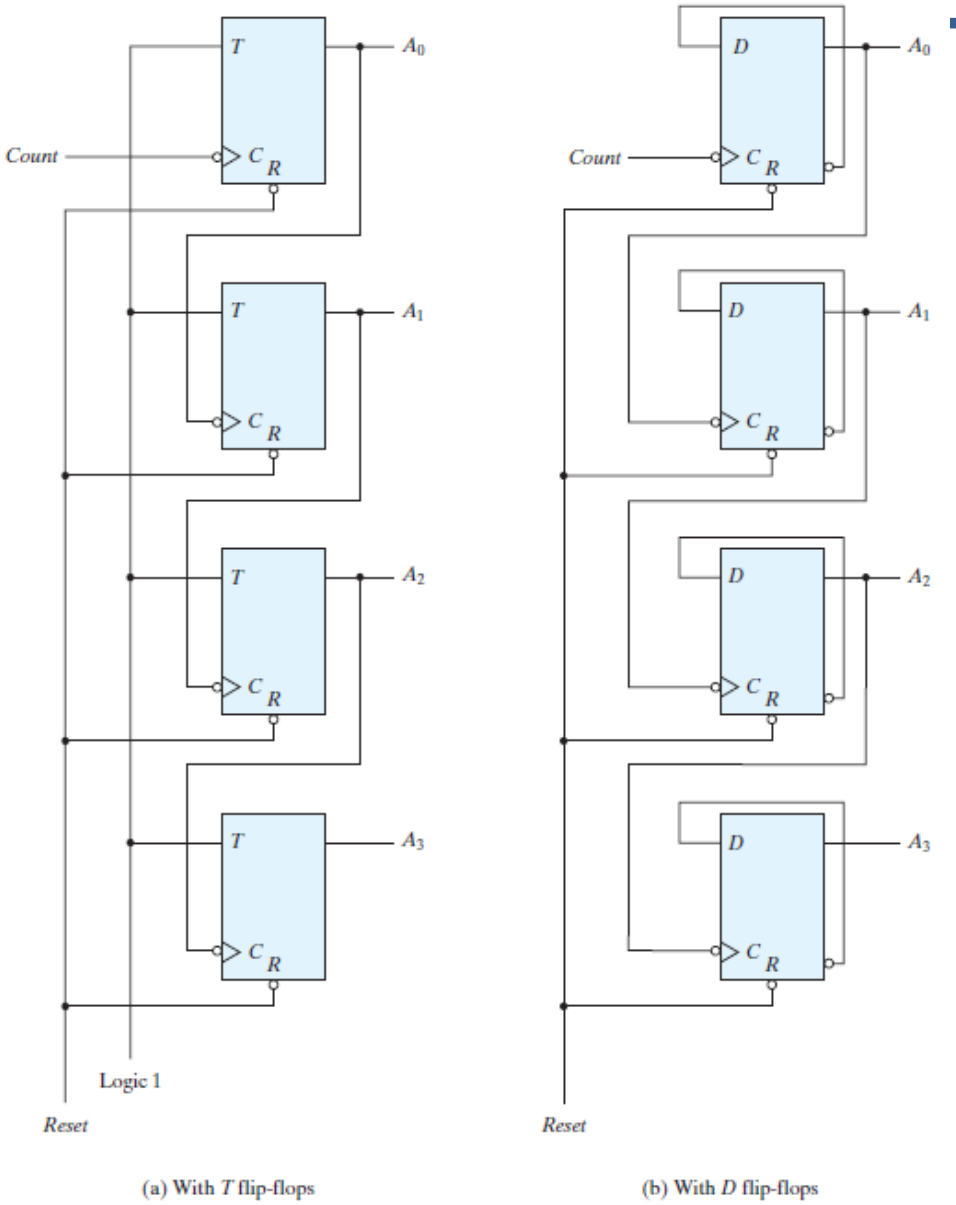
- Count events that may happen at random time intervals
- Use flip-flop transitions as a clock to the subsequent stages
- No common clock

Table 6.4
Binary Count Sequence

A_3	A_2	A_1	A_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0

- Observe the transitions and design the logic
- Every negative transition triggers the next higher level

Binary Ripple counter – count UP

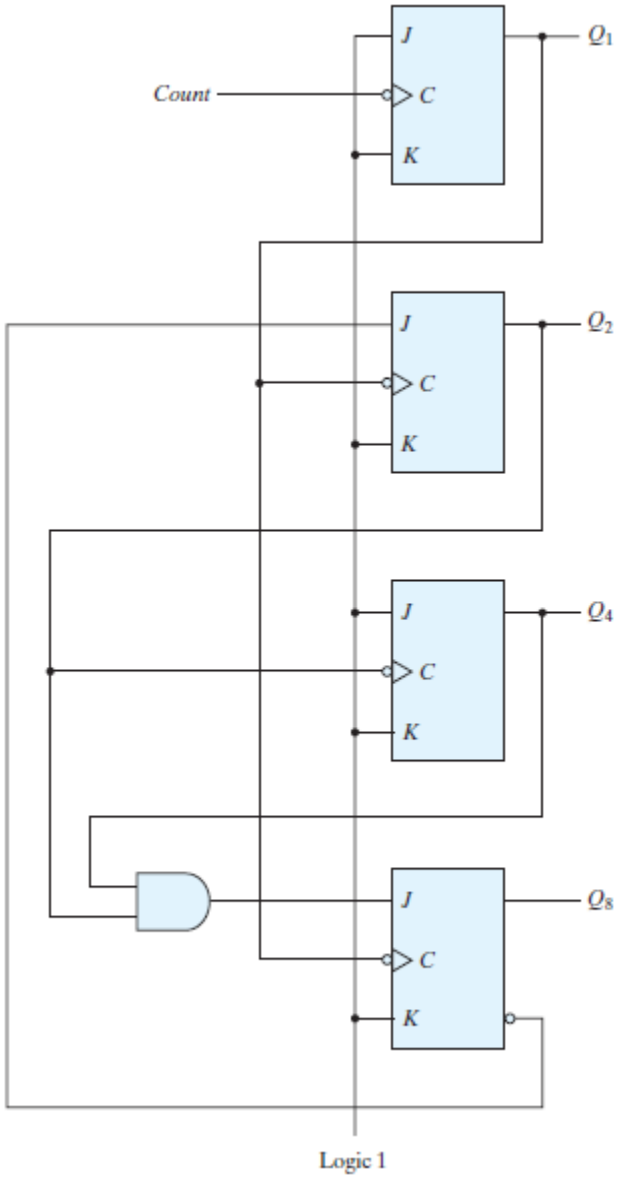


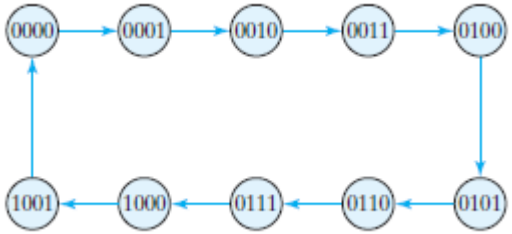
(a) With *T* flip-flops

(b) With *D* flip-flops

- What if all the flip-flops are positive-edge triggered?
- How to make MOD-6 ripple counter?
- What is the limitation for successfully counting an event?

BCD/Decade Ripple Counter





- How to design the logic?
- BCD Ripple counters can be easily cascaded

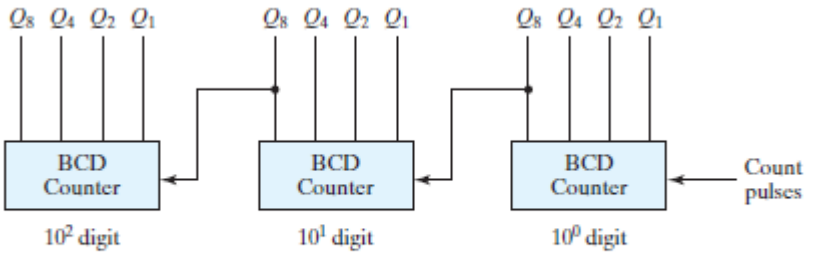
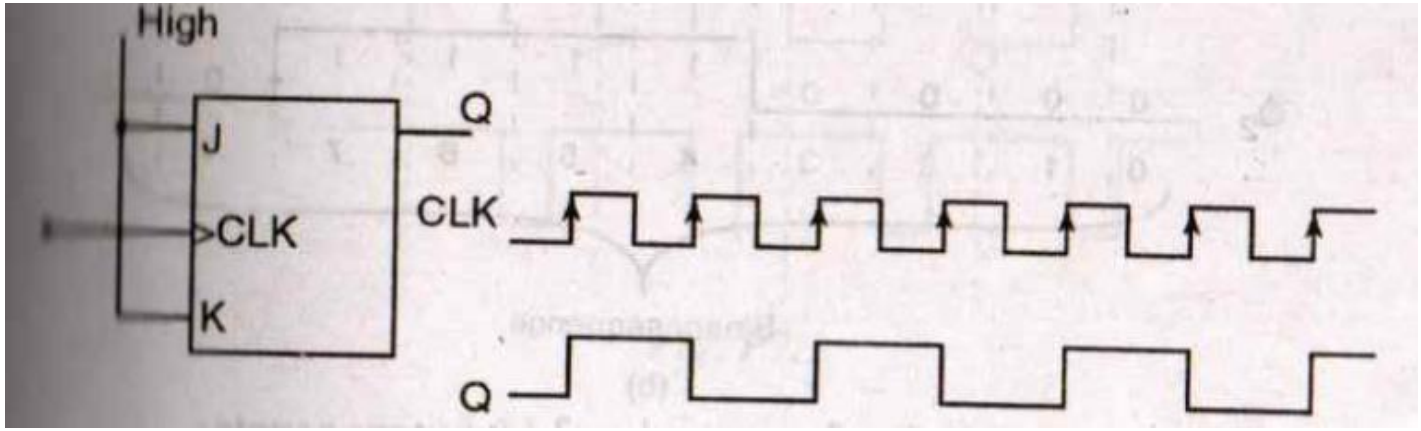


FIGURE 6.11
Block diagram of a three-decade decimal BCD counter

Frequency Division by Flipflop

- CLK frequency can be divided by 2 multiple times by passing through a flipflop

Divide by 2



Divide by 4