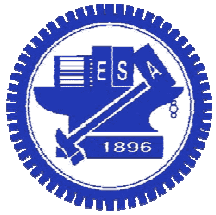
Registers and Counters



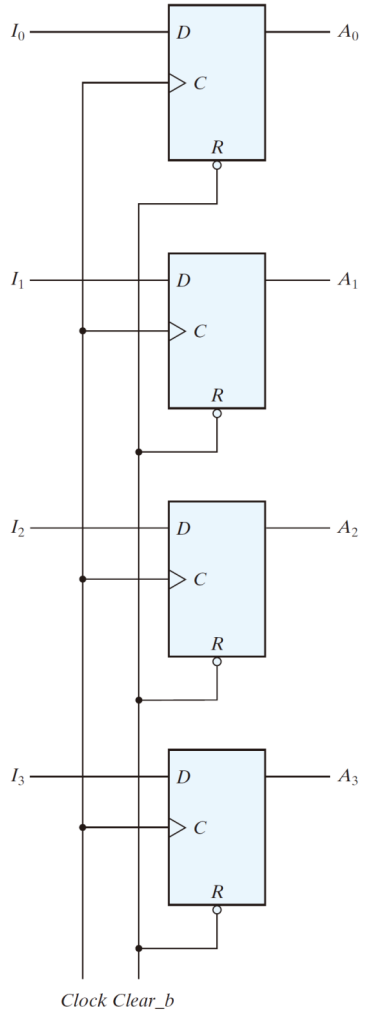
Chun-Jen Tsai
National Chiao Tung University
11/26/12

Registers

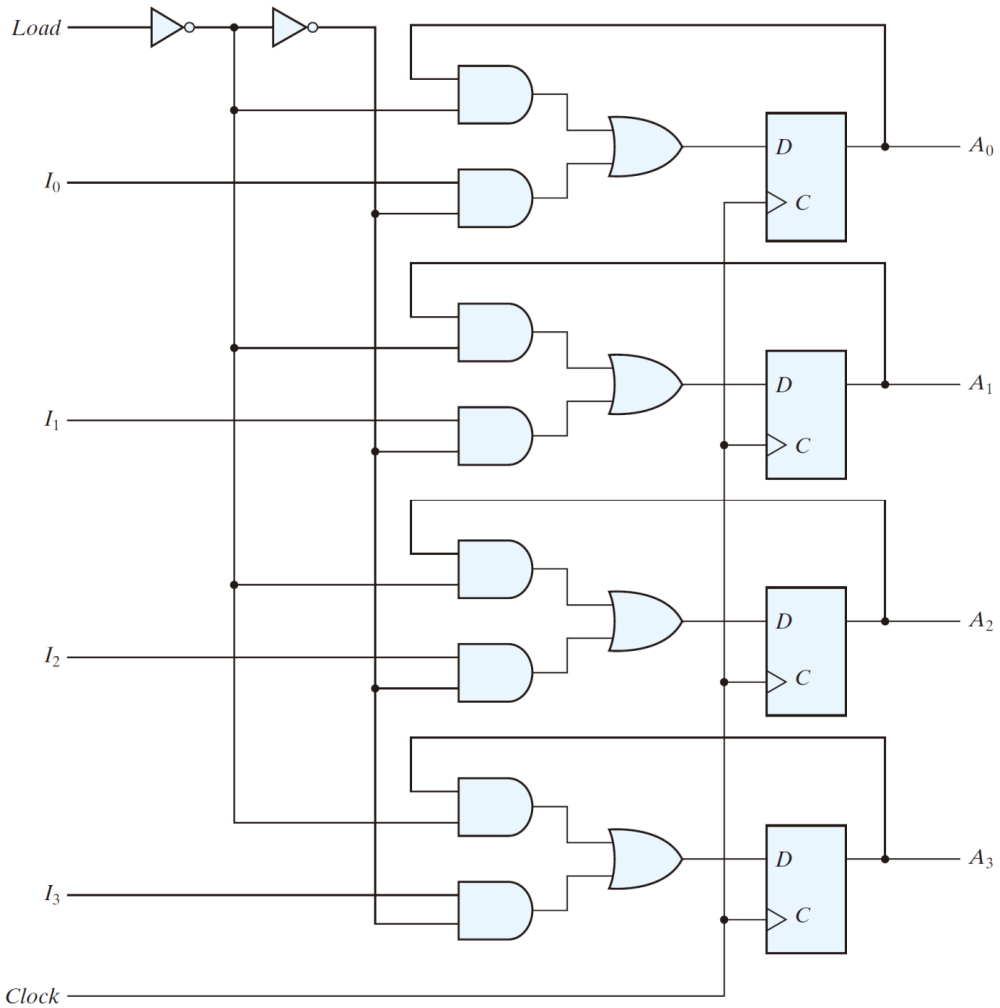
- We use single-bit flip-flops as the elementary memory device in a sequential circuit
- In practice, we need multiple bits for data storage in sequential circuits, such devices are called registers
  - A n-bit register can be constructed using an array of flip-flops or a special-purpose high-speed multi-bit memory device
  - A counter is a special-purpose register that goes through a predetermined sequence of states
- A collection of registers used in a sequential circuit is often called a “register file”

Flip-flop Implementation of Registers

- An n-bit register
  - n flip-flops capable of storing n bits of binary information
  - Example: a 4-bit register
- Problem: what if we do not want to change the state every clock cycle?

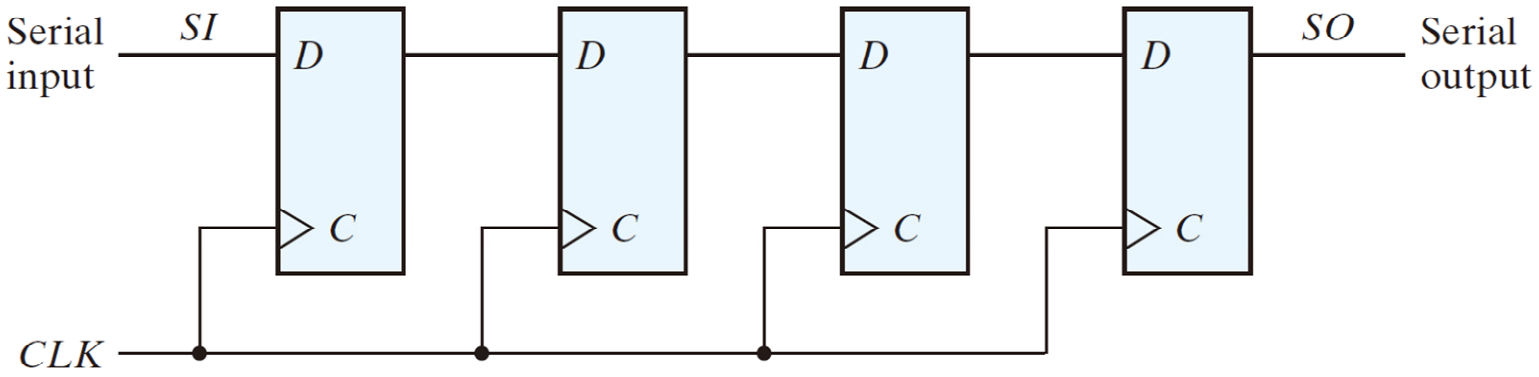


A 4-bit Register with Parallel Load



Shift Registers

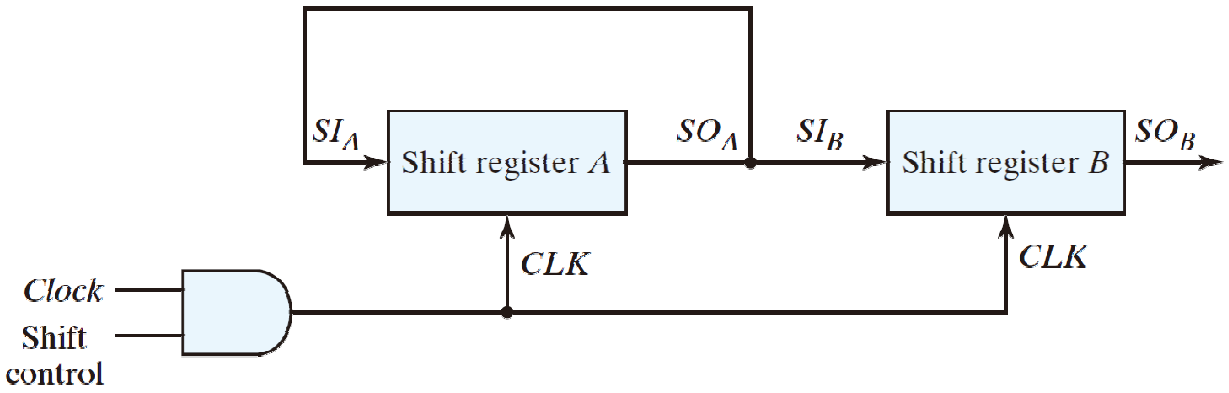
- Shift register
  - a register capable of shifting its binary information in one or both directions
- Example: 4-bit shift register
  - What is it good for?



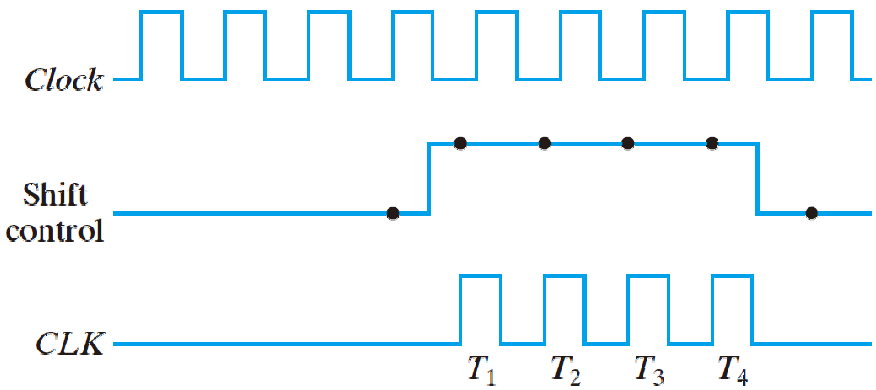
Data Transfer between Registers

- There are two type of data transfer from register to register:
- Serial transfer:
  - Information is transferred one bit at a time
  - shifts the bits out of the source register into the destination register
- Parallel transfer:
  - All the bits of the register are transferred at the same time

Serial Transfer from reg A to reg B



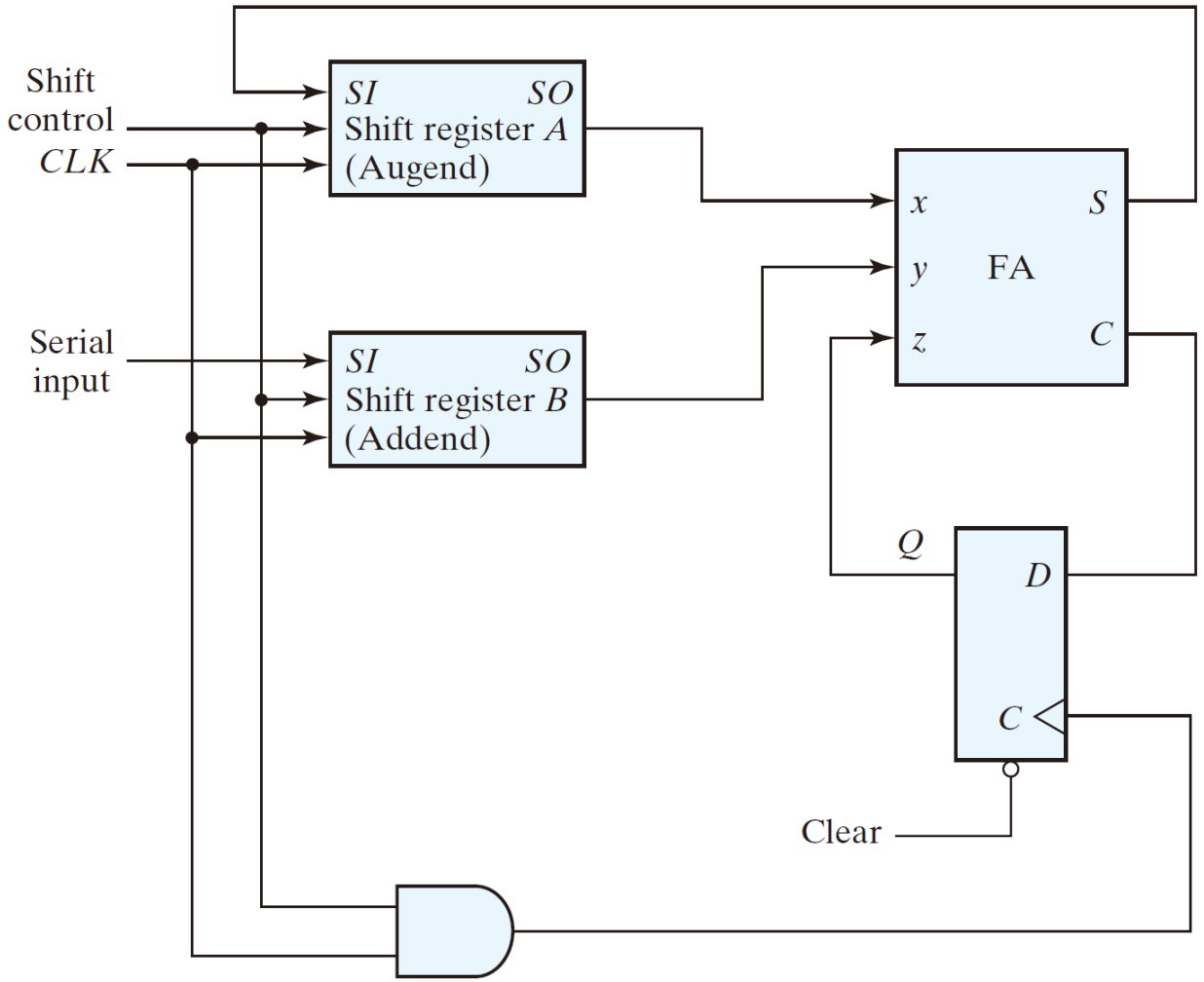
(a) Block diagram



(b) Timing diagram

Clock	Shift Reg A				Shift Reg B			
Initial value	1	0	1	1	0	0	1	0
After T_1	1	1	0	1	1	0	0	1
After T_2	1	1	1	0	1	1	0	0
After T_3	0	1	1	1	0	1	1	0
After T_4	1	0	1	1	1	0	1	1

Serial Addition using D Flip-Flops



Serial Adder using JK Flip-Flops (1/2)

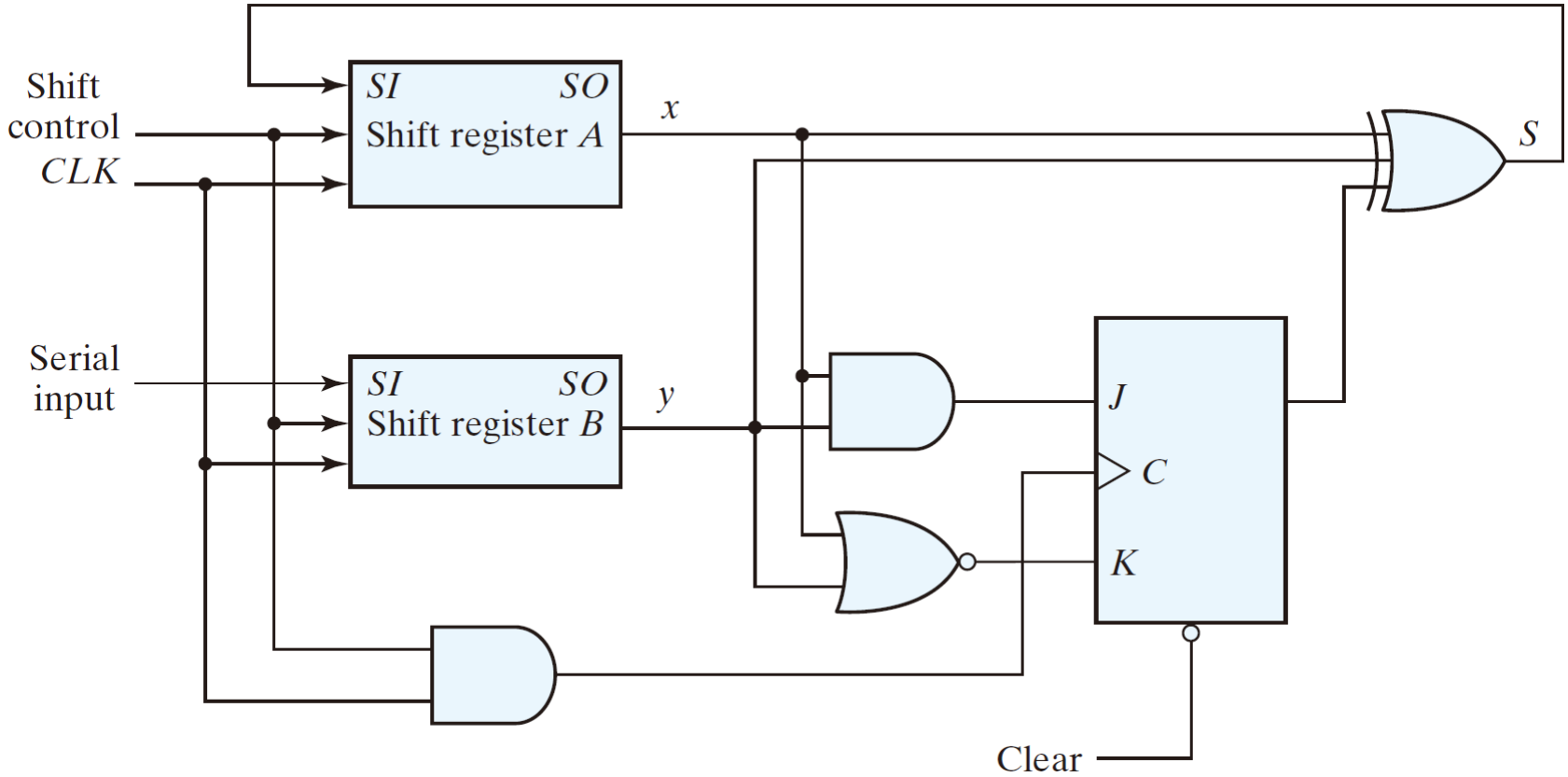
- State Table:

Present State	Inputs		Next State	Output	Flip-Flop Inputs	
Q	x	y	Q	S	J_Q	K_Q
0	0	0	0	0	0	X
0	0	1	0	1	0	X
0	1	0	0	1	0	X
0	1	1	1	0	1	X
1	0	0	0	1	X	1
1	0	1	1	0	X	0
1	1	0	1	0	X	0
1	1	1	1	1	X	0

- State functions:
  - $J_Q = x\,y$
  - $K_Q = x'\,y' = (x + y)'$
  - $S = x \oplus y \oplus Q$

Serial Adder using JK Flip-Flops (2/2)

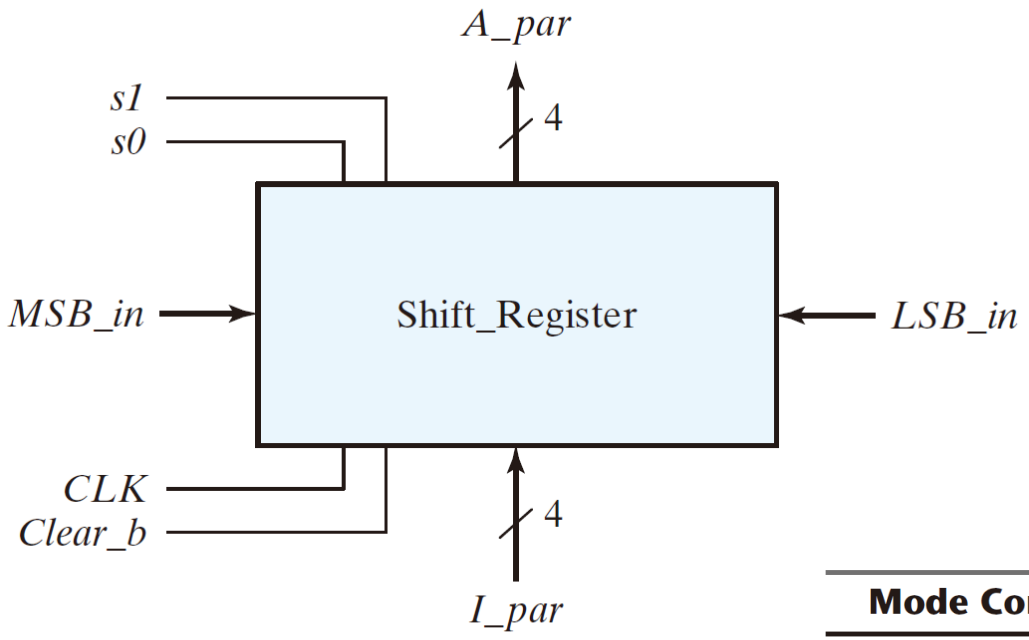
- Logic Diagram:



Universal Shift Register

- A universal shift register is a bi-directional shift register with parallel load/out capabilities
- I/O ports of a universal shift register
  - A “clear” input to clear the register to 0.
  - A “clock” input to synchronize the operations.
  - A “control” input to enable/disable the shift right/left, or n-bit parallel load operations.
  - n parallel output lines.

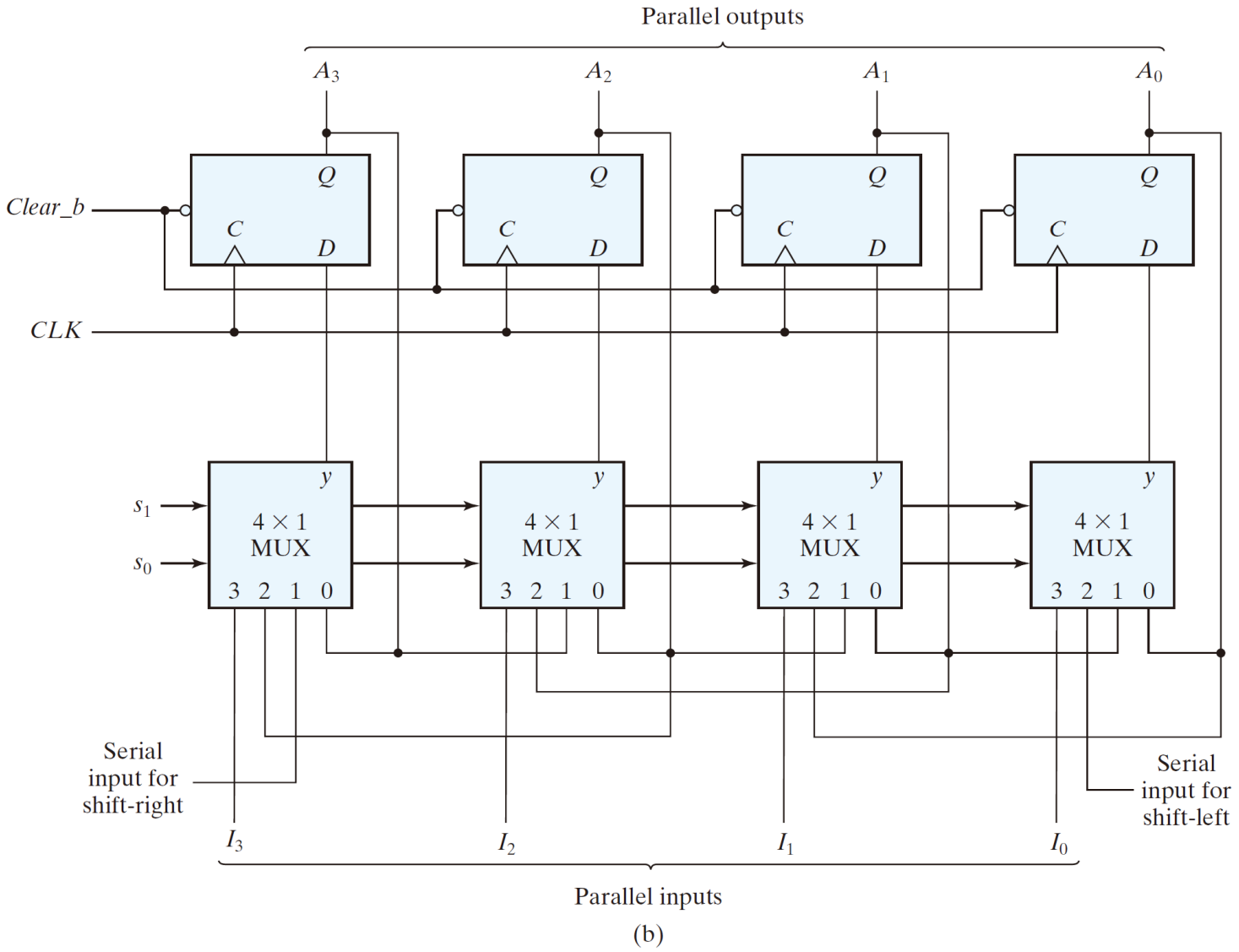
Example: 4-bit Universal Shift-Reg.



(a)

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

Logic Diagram



(b)

Ripple Counter

- A counter is a register that goes through a prescribed sequence of states upon the application of input pulses.
- The input pulses may be a clock pulses or a signal originating from some external source.
- The sequence of states may follow the binary number sequence (e.g. a binary counter) or any other sequence of states.

Categories of Counters

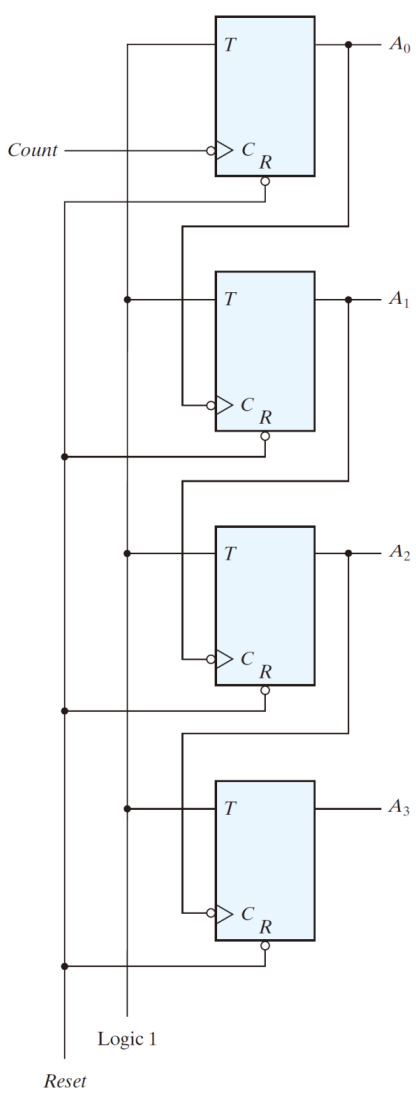
- Ripple counters:
  - The flip-flop output transition serves as a source for triggering other flip-flops
  - No common clock pulse (asynchronous)
- Synchronous counters:
  - The clock input (CLK) of all flip-flops receive a common clock

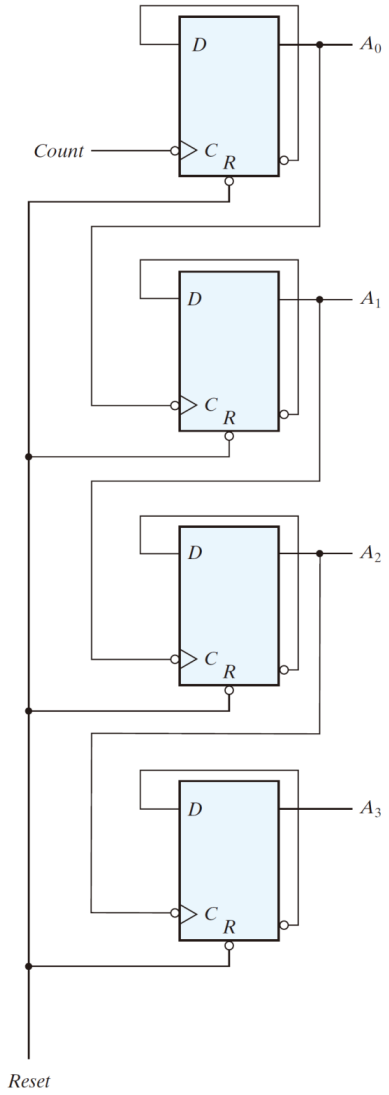
Example: Binary Ripple Counter

- A binary ripple counter can be done easily with T or D flip-flops:

Binary Count Sequence

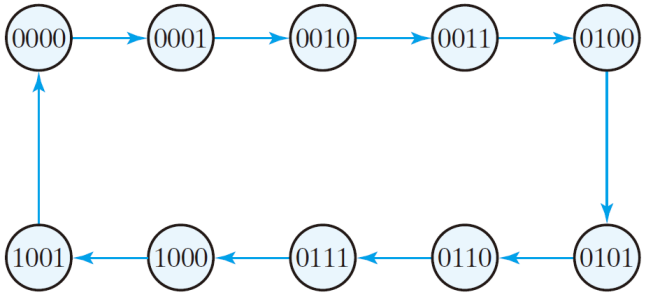
A_3	A_2	A_1	A_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0

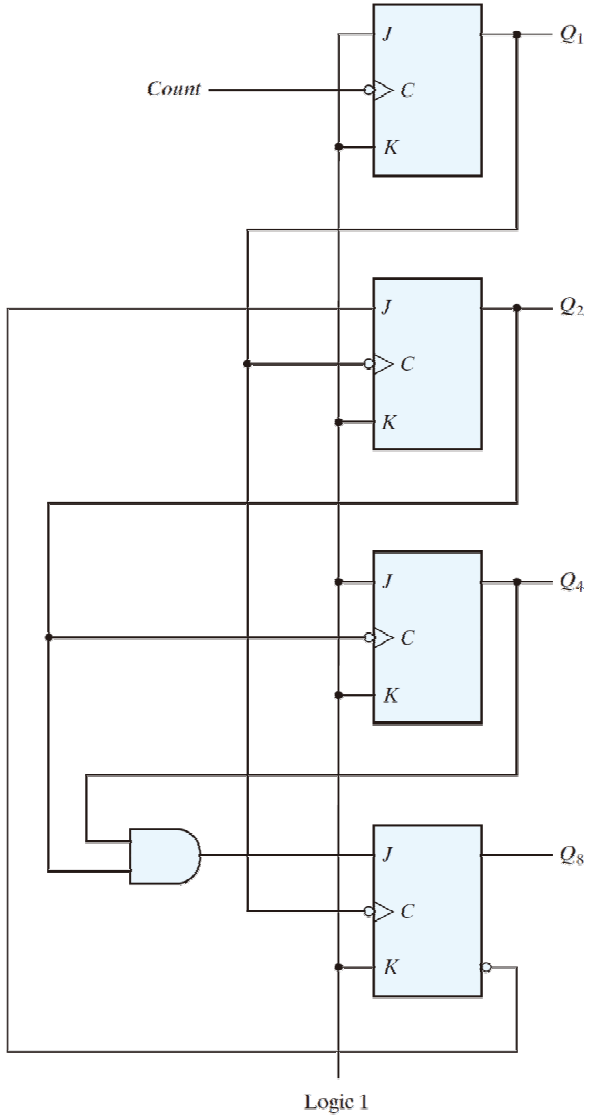




Example: BCD Counter

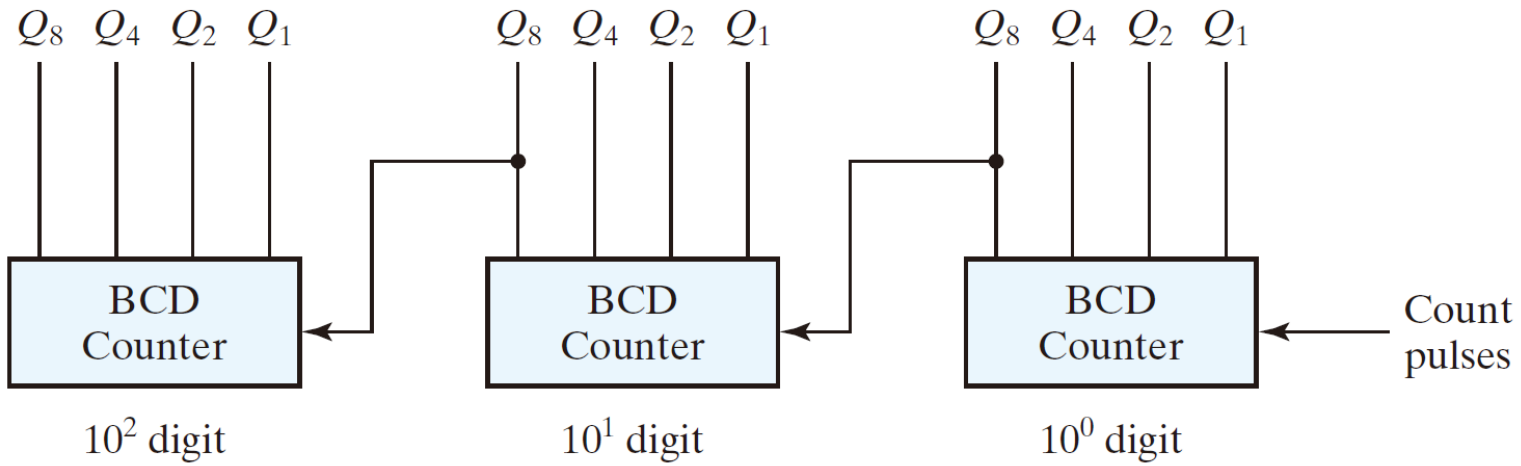
- The state and logic diagram of a decimal BCD counter:





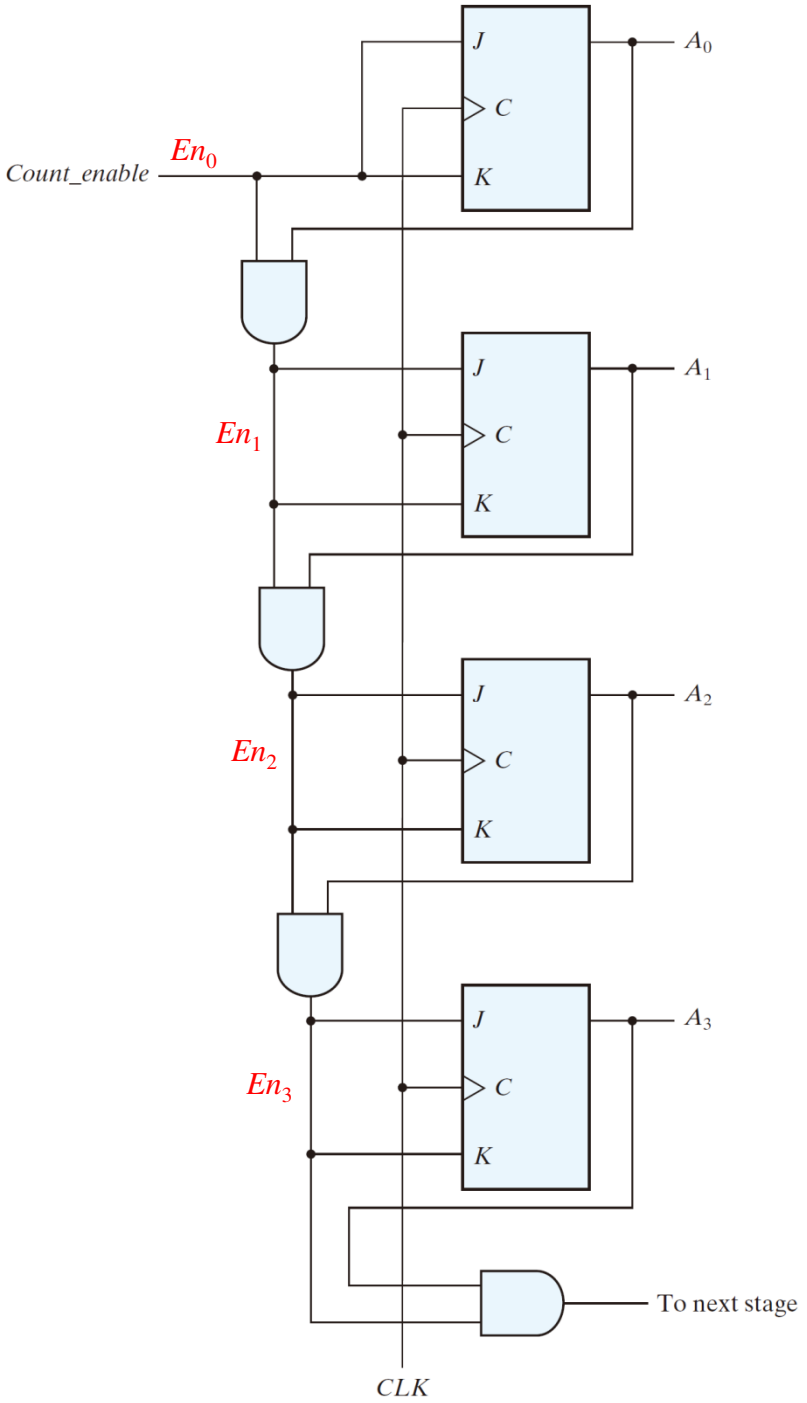
Multi-digit Decimal BCD Counter

- BCD counter can be chained together to form a multi-digit BCD counter



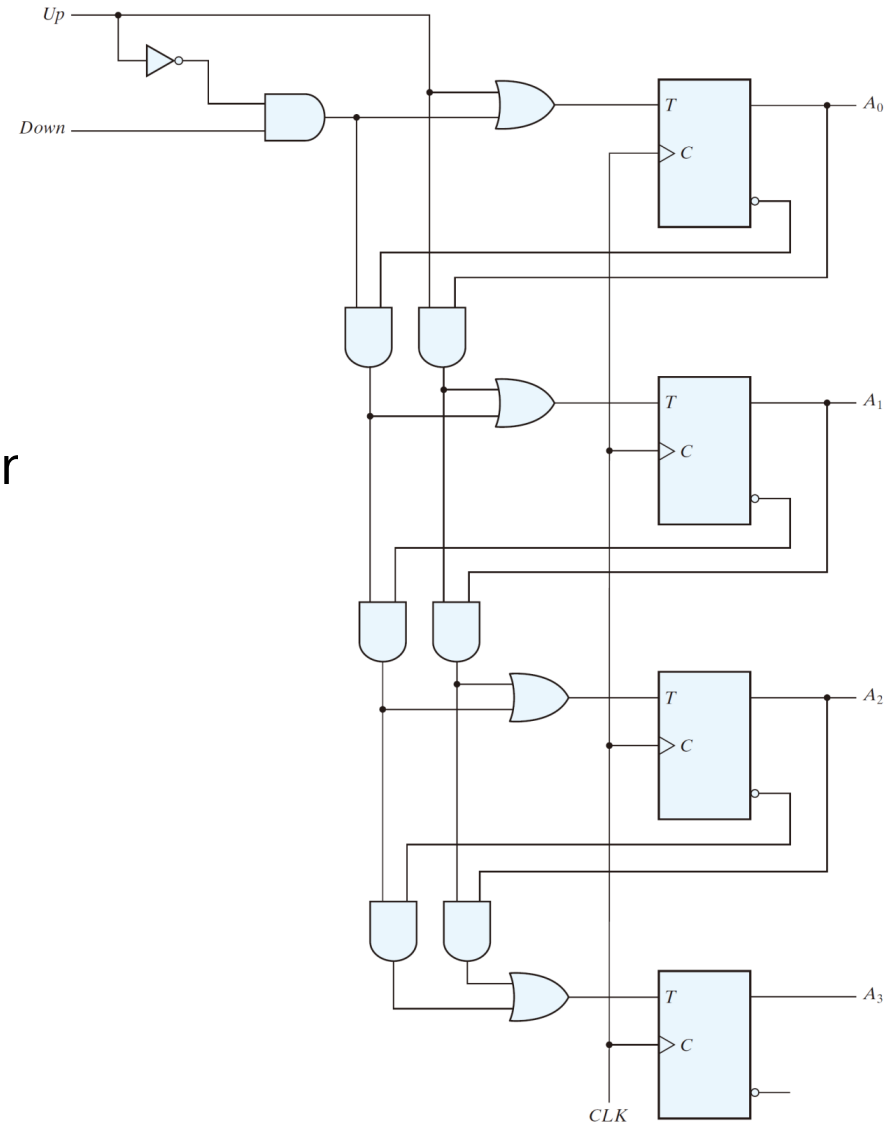
Sync. Counters

- A synchronous counter uses a common clock triggers all flip-flops simultaneously
- Note that:
  - $En_1 = En_0 A_0$
  - $En_2 = En_0 A_0 A_1$
  - $En_3 = En_0 A_0 A_1 A_2$



Up/Down Binary Counters

- A up/down counters can perform counting in both directions
- Counting control:
  - U/D = 1/0 → counts up
  - U/D = 0/1 → counts dowr
  - U/D = 0/0 → pause
  - U/D = 1/1 → counts up



BCD Counters

- A BCD counter counts from 0 → 9, repeatedly

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

- Simplified input functions for T flip-flops:

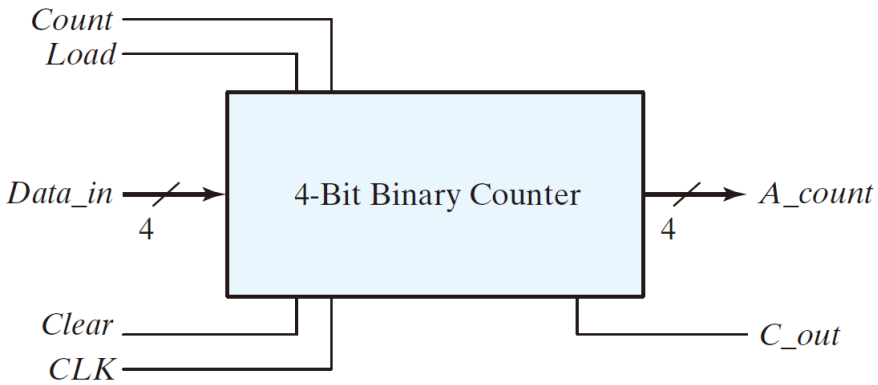
$$T_{Q1} = 1$$
$$T_{Q2} = Q'_8 Q_1$$
$$T_{Q4} = Q_2 Q_1$$
$$T_{Q8} = Q_8 Q_1 + Q_4 Q_2 Q_1$$
$$y = Q_8 Q_1$$

Binary Counter w/ Parallel Load (1/2)

- Counters often requires initialization before counting
- The function table for a counter with parallel load:

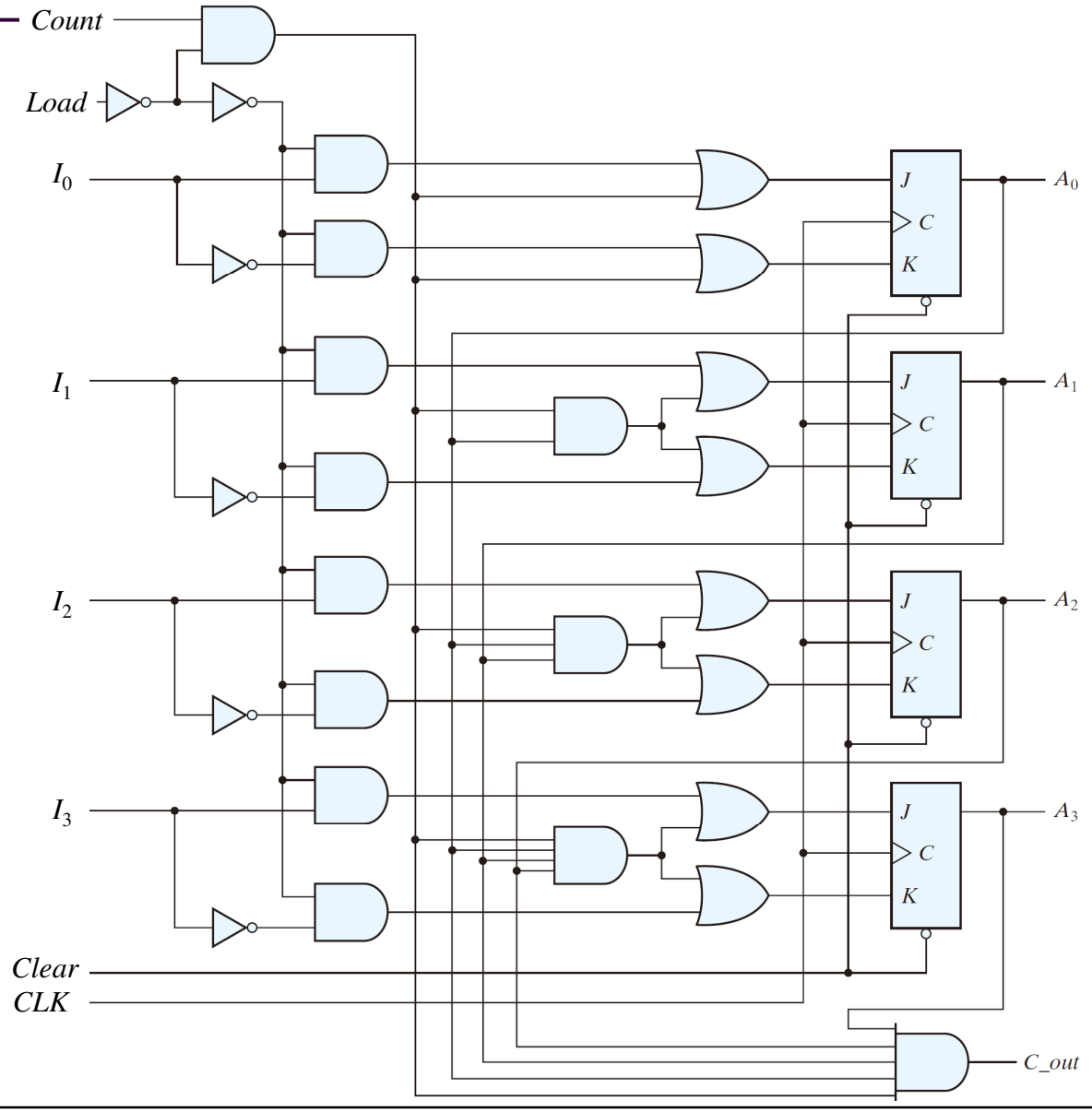
Clear	CLK	Load	Count	Function
0	X	X	X	Clear to 0
1	↑	1	X	Load inputs
1	↑	0	1	Count next binary state
1	↑	0	0	No change

- I/O ports



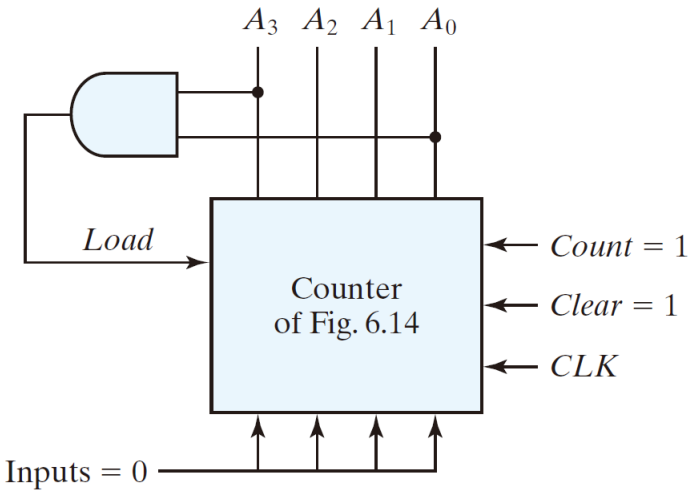
Binary Counter w/ Parallel Load (2/2)

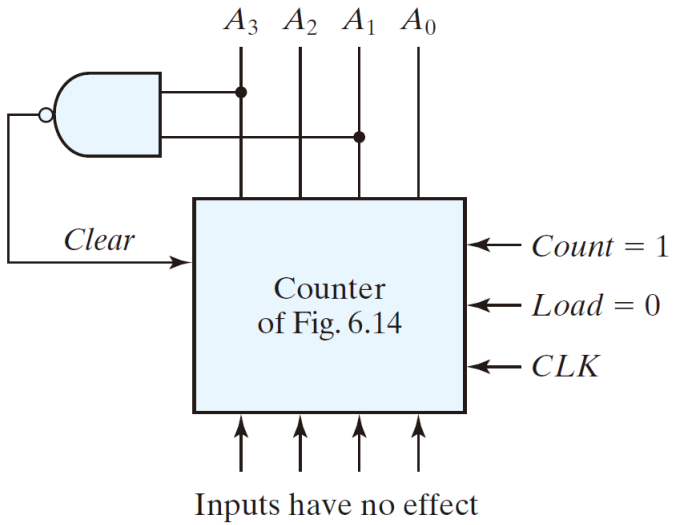
- Logic diagram:



Count Sequence Control

- A counter with parallel load can be used to perform counting for different count sequence
- Example: two ways to emulate BCD sequences:





Comments on Counter Design

- A counter can be designed to generate any desired sequence of states
  - Example: a divide-by-N counter (modulo-N counter) goes through a repeated sequence of N states; the sequence may follow the binary count or may be any arbitrary sequence
- Handling unused states
  - A counter composed of n flip-flops can represent 2^n states
  - Not all states are used; unused states may be treated as
    - Assigned specific next states $\rightarrow$ circuit behavior is deterministic
    - Don't-care conditions $\rightarrow$ circuit behavior should not have a trap!

Self-Correcting Counter (1/2)

- A self-correcting counter ensures that when it enters one of its unused states, it eventually goes into one of the valid states after one or more clock pulses
- Example: a counter with 2 unused states – 011 & 111

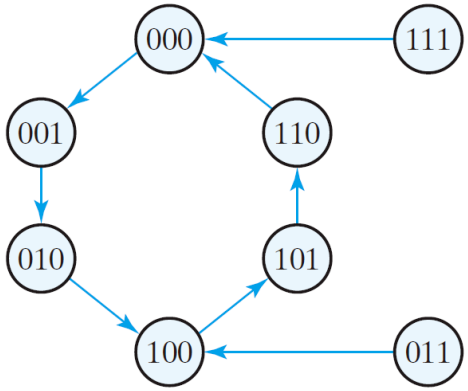
Present State			Next State			Flip-Flop Inputs					
A	B	C	A	B	C	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	1	0	0	1	X	X	1	0	X
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	1	1	0	X	0	1	X	X	1
1	1	0	0	0	0	X	1	X	1	0	X

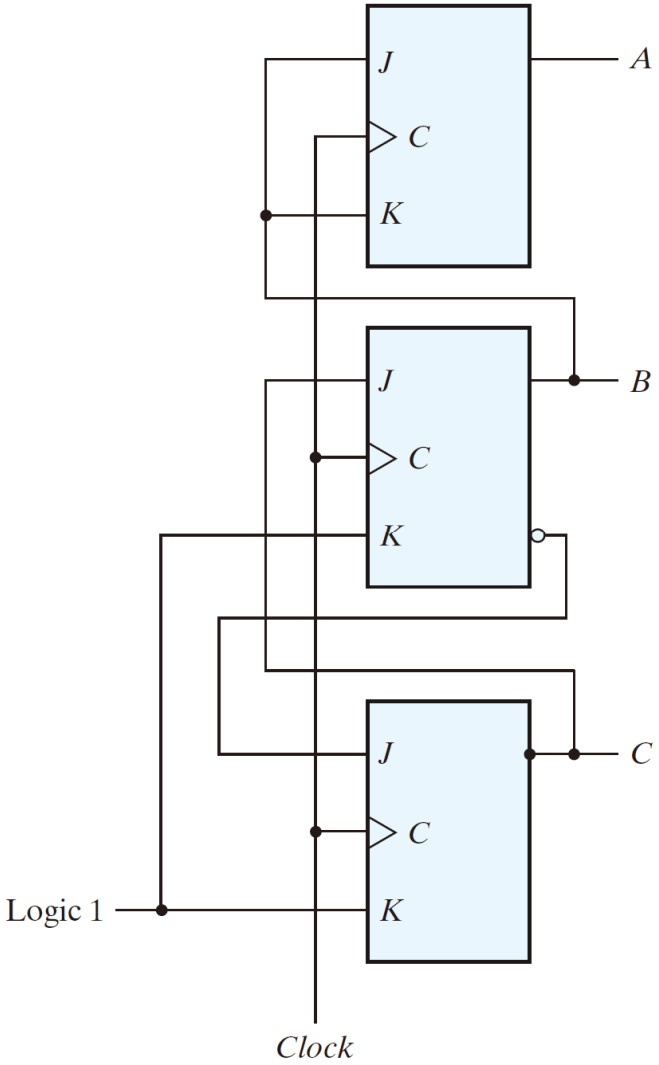
set X to '1' for simplification!

Self-Correcting Counter (2/2)

- The flip-flop input equations:
  - $J_A = B, \quad K_A = B$
  - $J_B = C, \quad K_B = 1$
  - $J_C = B', \quad K_C = 1$

- The state and logic diagrams:

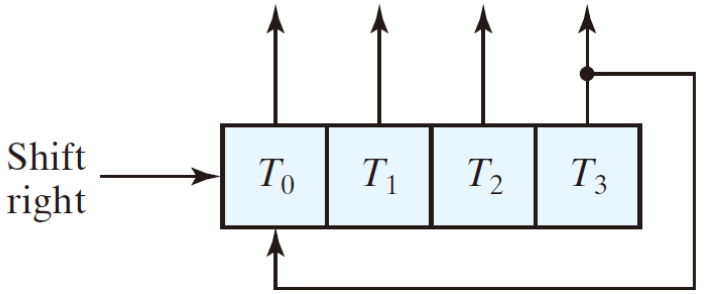




Ring Counter

- A ring counter is a circular shift register with only one flip-flop being set at any particular time, all others are cleared (initial value = 1 0 0 … 0)
- The single bit is shifted from one flip-flop to the next to produce the sequence of timing signals
- Example: 4-bit ring counter

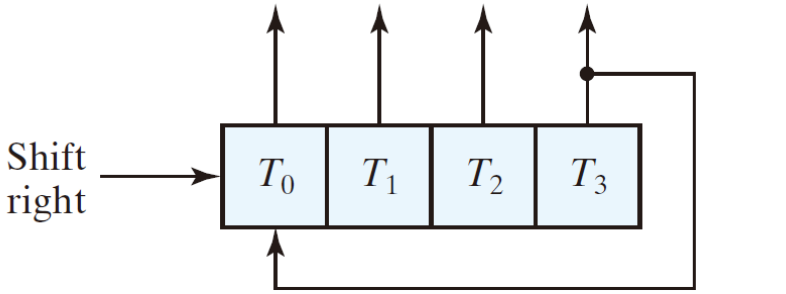
A_3	A_2	A_1	A_0
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1
1	0	0	0



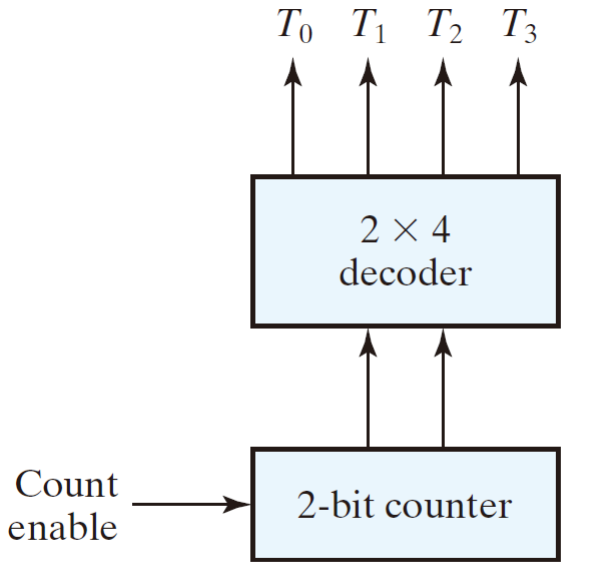
(a) Ring-counter (initial value = 1000)

Application of Ring Counters (1/2)

- Ring counters may be used to generate timing signals to control the sequence of operations in a digital system
- Approaches for generation of 2^n timing signals
  1. a shift register with 2^n flip-flops
  2. an n-bit binary counter together with an n-to-2^n-line decoder



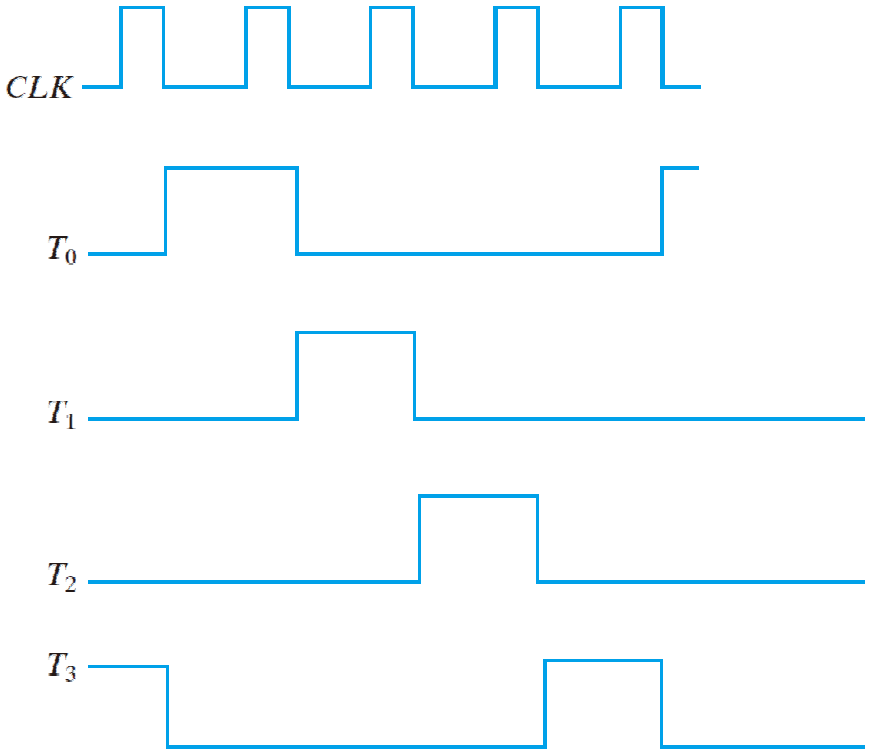
Ring-counter (initial value = 1000)



Counter and decoder

Application of Ring Counters (2/2)

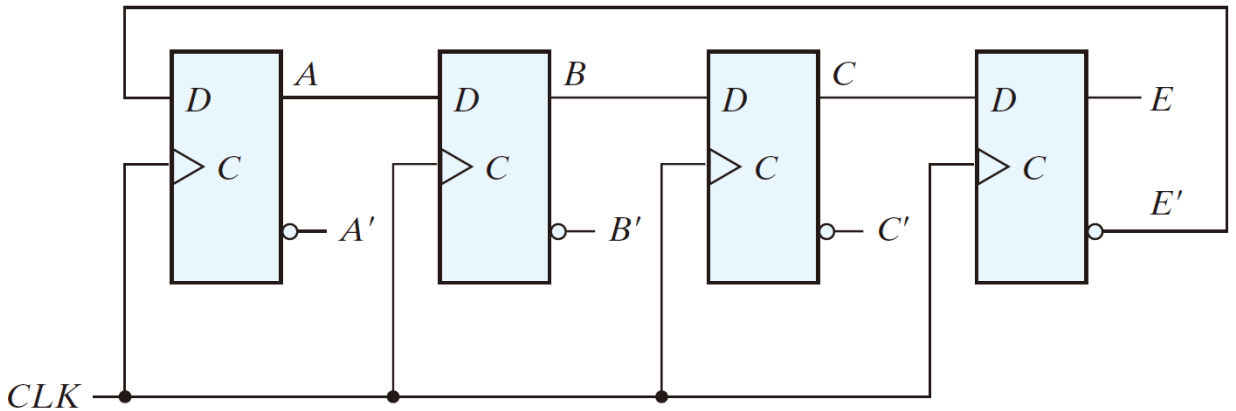
- The resulting timing signals:



Sequence of four timing signals

Switch-tail Ring Counter

- A k-bit ring counter circulates a single bit among the flip-flops to provide k distinguishable states.
- A switch-tail ring counter use a circular shift register to provide twice the states of normal a ring counter
  - a k-bit switch-tail ring counter will go through a sequence of $2k$ distinguishable states (initial value = $0\ 0\ \ldots\ 0$).



Four-stage switch-tail ring counter

Johnson Counters (1/2)

- A Johnson counter is a k-bit switch-tail ring counter plus $2k$ decoding gates
  - Each decoding gate has 2 inputs.
  - It provides $2k$ timing signals; only one is enabled at a time.
- Example: a 4-bit Johnson counter

Sequence number	Flip-flop outputs				Decoding Logic ($2k$ gates total)
	A	B	C	E	
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	$C'E$

Count sequence and required decoding

Johnson Counters (2/2)

- If a Johnson counter enters an unused state, it will persist to circulate in the invalid states and never find its way back to a valid state.
  - The invalid states happens when three consecutive flip-flops have the value $0–1–0$ or $1–0–1$, a possible correcting scheme is to force the state of the third flip-flop to reset to 0 by the correction logic: $D_C = (A + C)\,B$
- Johnson counters can provide k timing sequences using only $k/2$ flip-flops and k decoding gates.