Chapter 6: Registers and Counters

6-1 Registers

6-2 Shift Registers

6-3 Ripple Counters

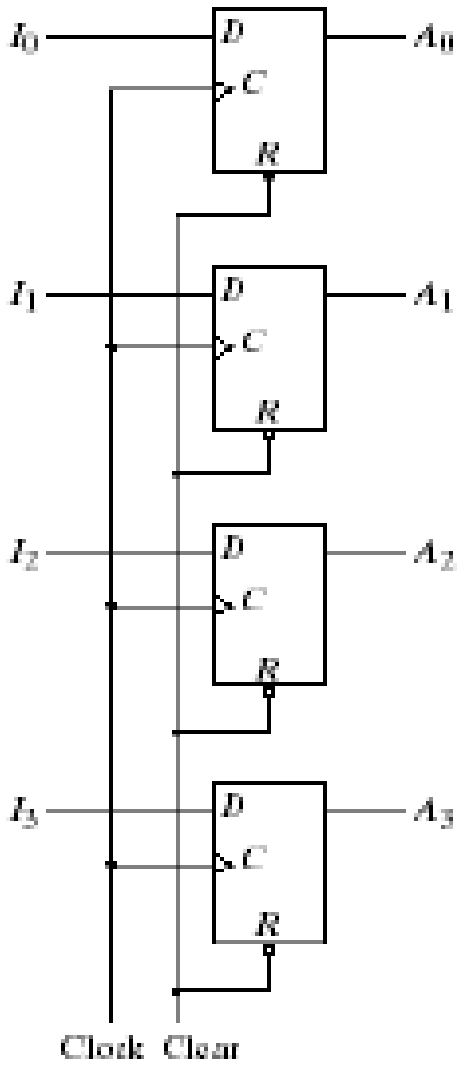
6-4 Synchronous Counters

6-5 Other Counters (Counter with Unused States and Ring Counter)

Registers

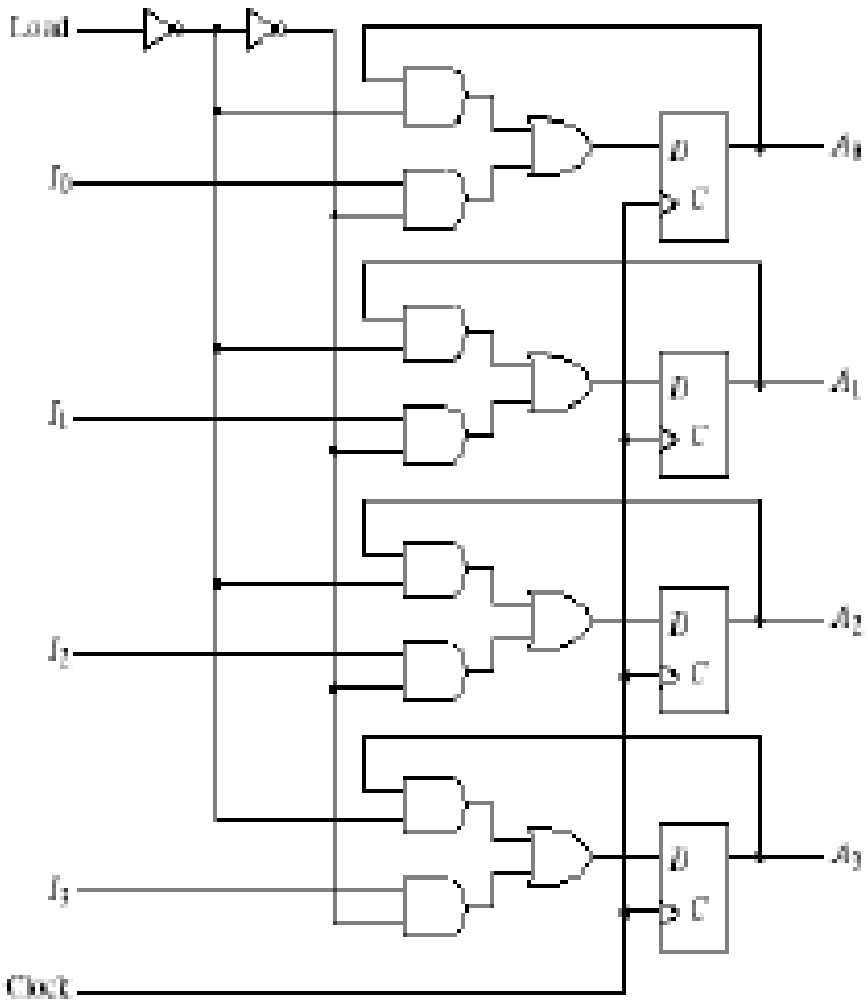
- A register is a group of flip-flops. An n-bit register consist of a group of n flip-flops capable of storing n bits of binary information.
- A counter is a register that goes through a predetermined sequence of states.
- There are different types of registers available commercially:

4-Bit Register with Clear Input



- The information is loaded in parallel.
- When *R*=0, the register will transfer to all 0's state.
- *R* input must be maintained at logic 1 during normal operations.
- A master clock generates a continuous train of clock pulses. The clock pulses are applied to all flip-flops and registers in the system. The clock pulses enable flip-flops and registers to take new inputs.
- The clock can be inhibited from reaching the register by controlling the clock input signal with an enabling gate. This may generate synchronization problems.

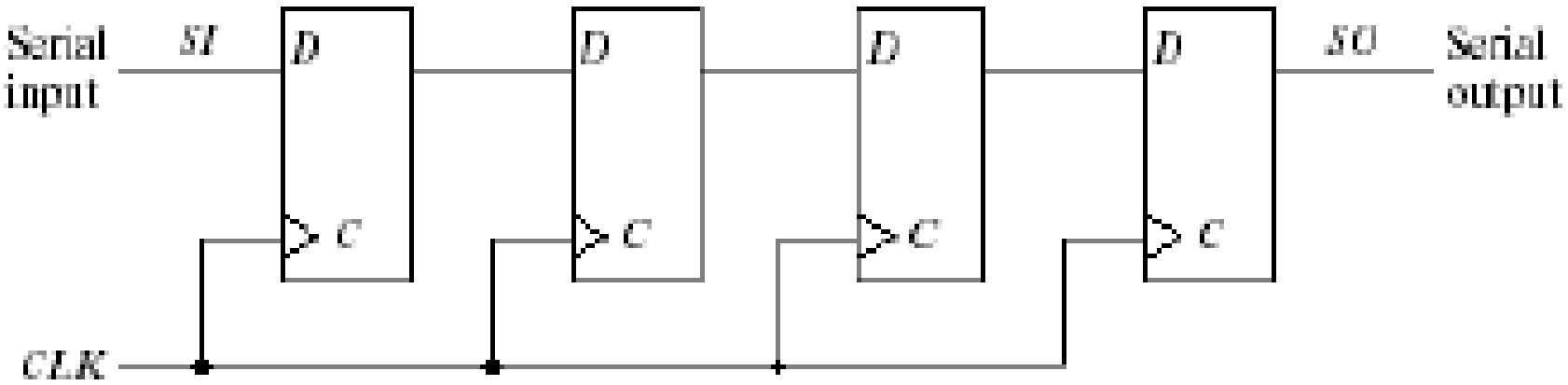
4-Bit Register with Parallel Load Control Input



- The load input to the register determines the action to be taken with each clock pulse.
- When the load input is 1, the data in the four inputs are transferred into the register with the next positive edge of the clock.
- When the load input is 0, the outputs of the flip-flops are connected to their inputs.

Shift Registers

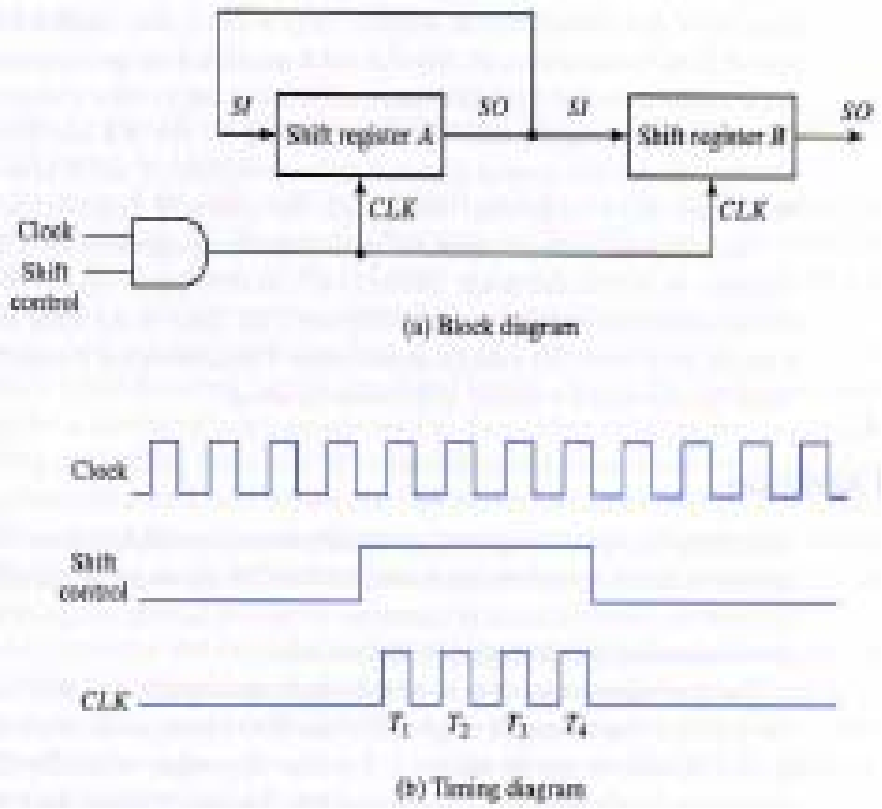
- **Registers that capable of shifting their binary information in one or both directions.**



- **Each clock pulse shifts the content of the register one bit position to the right.**

Serial Transfer

- **A digital system is said to operate in a serial mode when information is transferred and manipulated one bit at a time.**
- **Example 1:The serial transfer of information from register *A* to register *B* is done with shift registers.**



Serial Transfer

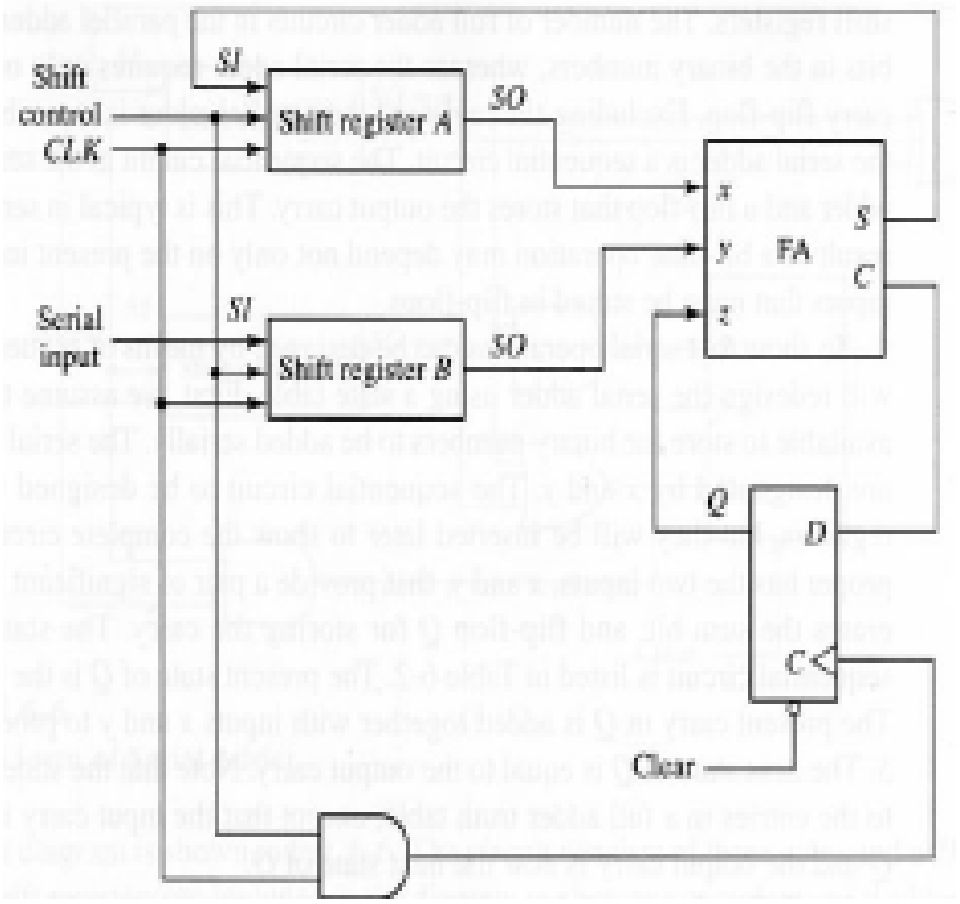
Serial-Transfer Example

Timing Pulse	Shift Register A	Shift Register 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 Transfer

➢Example 2:Serial Addition

- The two binary numbers are stored in two shift registers.
- Bits are added one pair at a time using a full adder.
- The carry out is transferred to a *D* flip-flop.
- The result will be stored in shift register *A*.
- The circuit can be used to sum a set of numbers.
- Register *B* can be used to transfer a new binary number.
- Initially, register *A* and carry flip-flop are cleared to 0.



Serial Transfer

- **Example 3:Serial Adder using a *JK* flip-flop We need to use 2 shift registers to store two numbers and one *JK* flip-flop to store the carry out.**

Flip-Flop Excitation Tables

$Q(t)$	$Q(t+1)$	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

(a) *JK*

State Table for Serial Adder

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

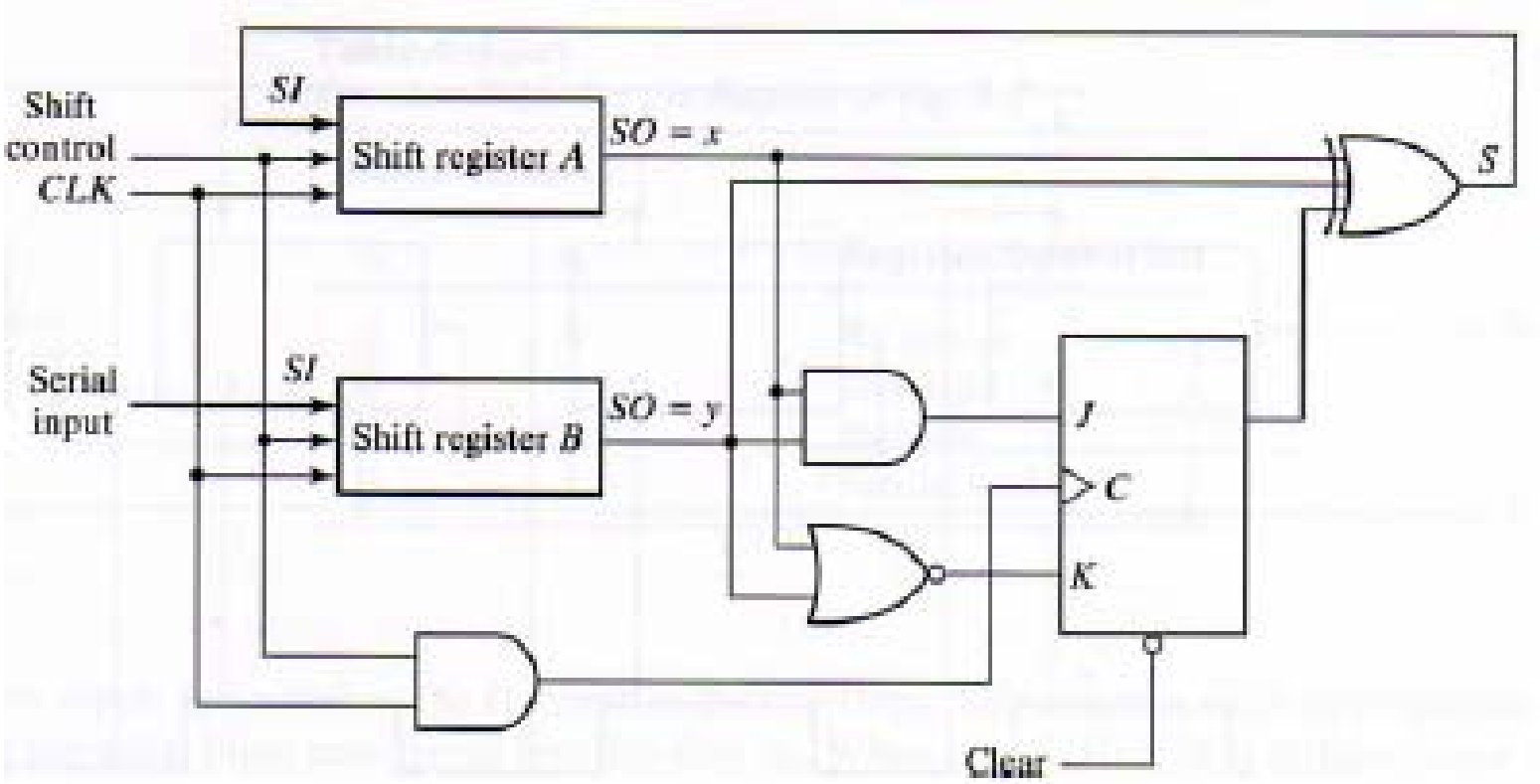
We need to use the maps to find S, J_Q, and K_Q.

$J_Q = xy$

$K_Q = x'y' = (x+y)'$

$S = x \oplus y \oplus z$

Serial Transfer



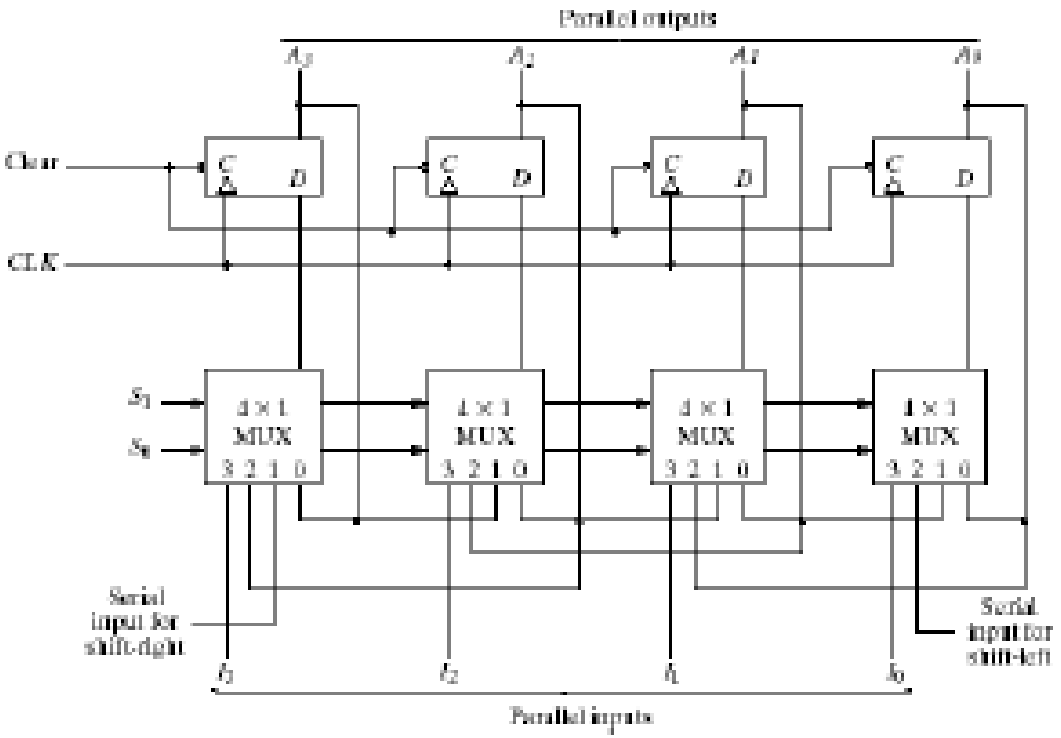
Serial Adder using a *JK* flip-flop

Universal Shift Register

- **A register with the following operations and characteristics:**
  1. **A clear control to clear the register to 0**
  2. **A shift-right control**
  3. **A shift-left control**
  4. **A parallel-load control**
  5. **nparallel output lines.**
  6. **A control state that leaves the information in the register unchanged in the presence of the clock.**

Universal Shift Register

- 4-Bit Universal Shift Register



- Function Table

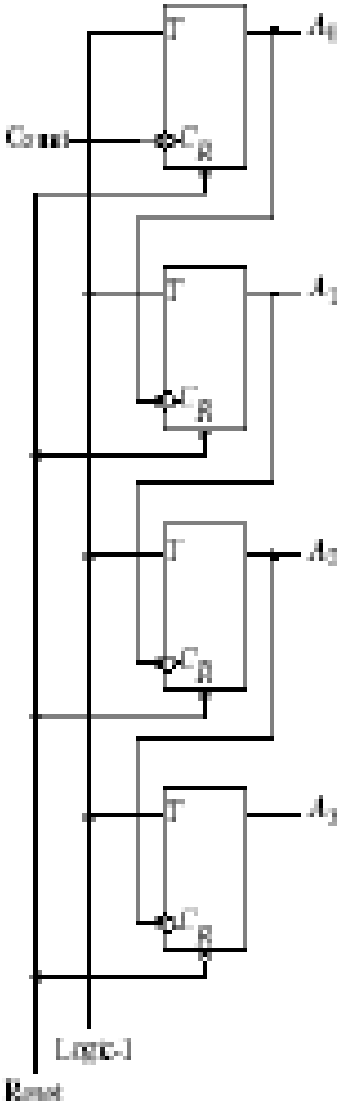
Mode Control		
S_1	S_0	Register Operation
0	0	No change
0	1	Shift right
1	0	Shift left
1	1	Parallel load

6.3 Ripple Counters Page 253

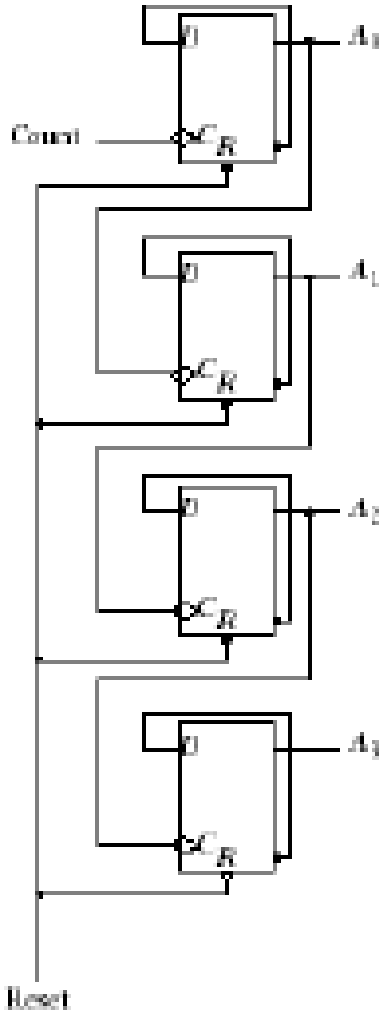
- A counter is a register that goes through a predetermined sequence of states upon the application of input pulses.
- Input Pulses Sources could be: Clock pulses or Pluses originate from internal or external source
- Input Pulses can occur at a fixed interval of time or at random
- A binary counter: A counter that follows the binary number sequence.
- An n-bit binary counter consists of n flip-flops
  - Can count in binary from 0 to 2^n-1
- Two Categories:
  - Ripple counters: the flip-flop output transition serves as a source for triggering other flip-flops.
  - Synchronous counters: all flip-flops receive the common clock.

Binary Ripple Counters

- The count starts with binary 0 and increments by one with each count pulse input. After the count of 15, the counter starts again from 0.
- Consist of a series connection of complementing flip-flops.
- The flip-flop holding the least significant bit receives the incoming count pulses.
- All *T* inputs of all the flip-flops are connected to a logic 1.
- Each flip-flop complement if the signal in its *C* input goes through a negative transition.
- Any pulse from the counter input complements A0.
- Every time A0 goes from 1 to 0, it complements A1, every time A1 goes from 1 to 0, it complements A2, and every time A2goes from 1 to 0, it complements A3.



Binary Ripple Counter using *D* Flip-Flops



- For each *D* flip-flop, the complement output connected to the *D* input.

- **Question:**

- How can we implement a binary ripple counter using *JK* flip-flops?

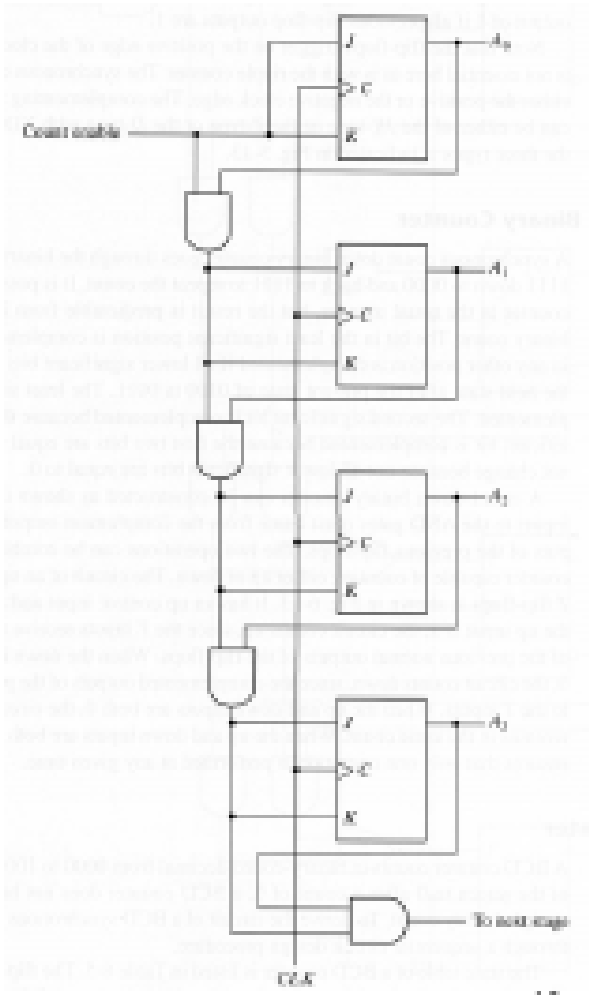
6.4 Synchronous Counters P. 258

- **Synchronous Counters are different from ripple counters in that clock pulses applied to the inputs of all flip-flops.**

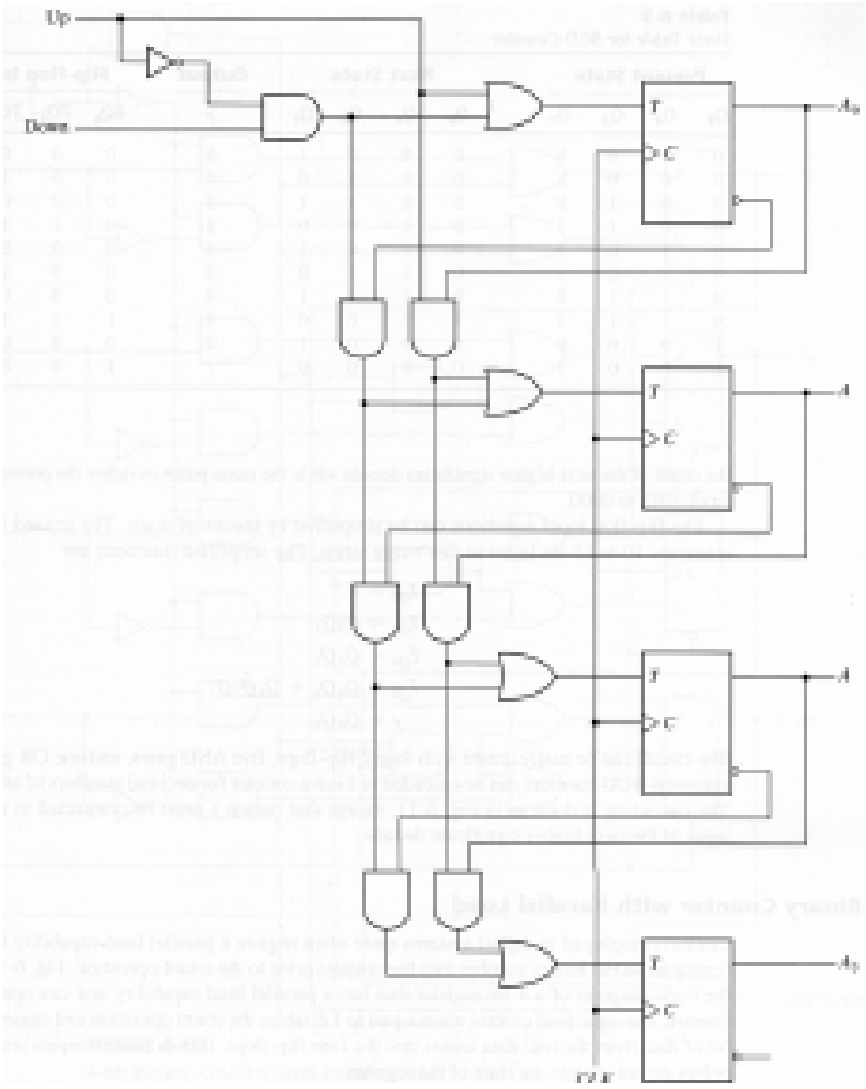
- **A common clock triggers all flip-flops simultaneously.**

Synchronous Binary Counter

- A_0 is complemented with each clock pulse.
- A_1 is complemented when A_0=1.
- A_2 is complemented when A_0=1 and A_1=1.
- A_3 is complemented when A_0=1, A_1=1, and A_2=1.



Up-Down Binary Counter Fig. 6.13 P. 261



Up	Down	Function
0	0	No change
0	1	The circuit counts down
1	0	The circuit counts up
1	1	The circuit counts up

BCD Counter

- A BCD counter counts in binary-coded decimal from 0000 to 1001 and back to 0000.

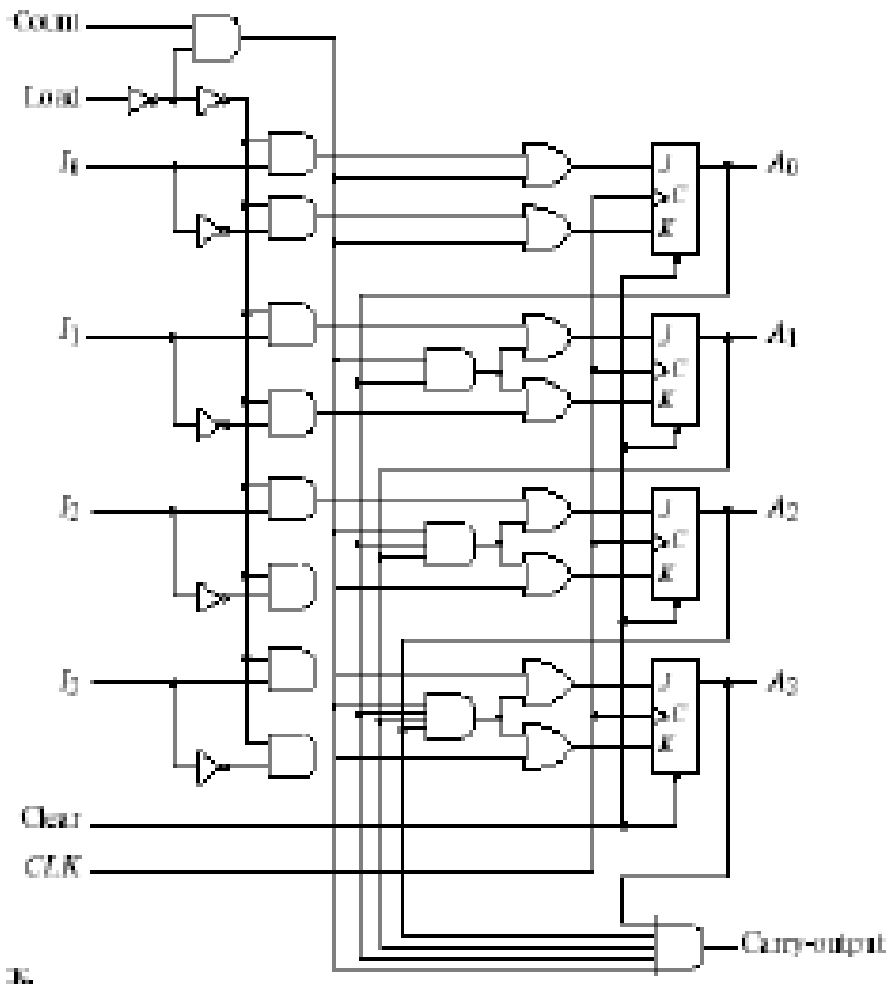
State Table for BCD Counter

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

- $T_{Q1}=1$, $T_{Q2}=Q_8'Q_1$, $T_{Q4}=Q_2Q_1$

  $T_{Q8}=Q_8Q_1+Q_4Q_2Q_1$, $y=Q_8Q_1$

Binary Counter with Parallel Load Fig. 6.14



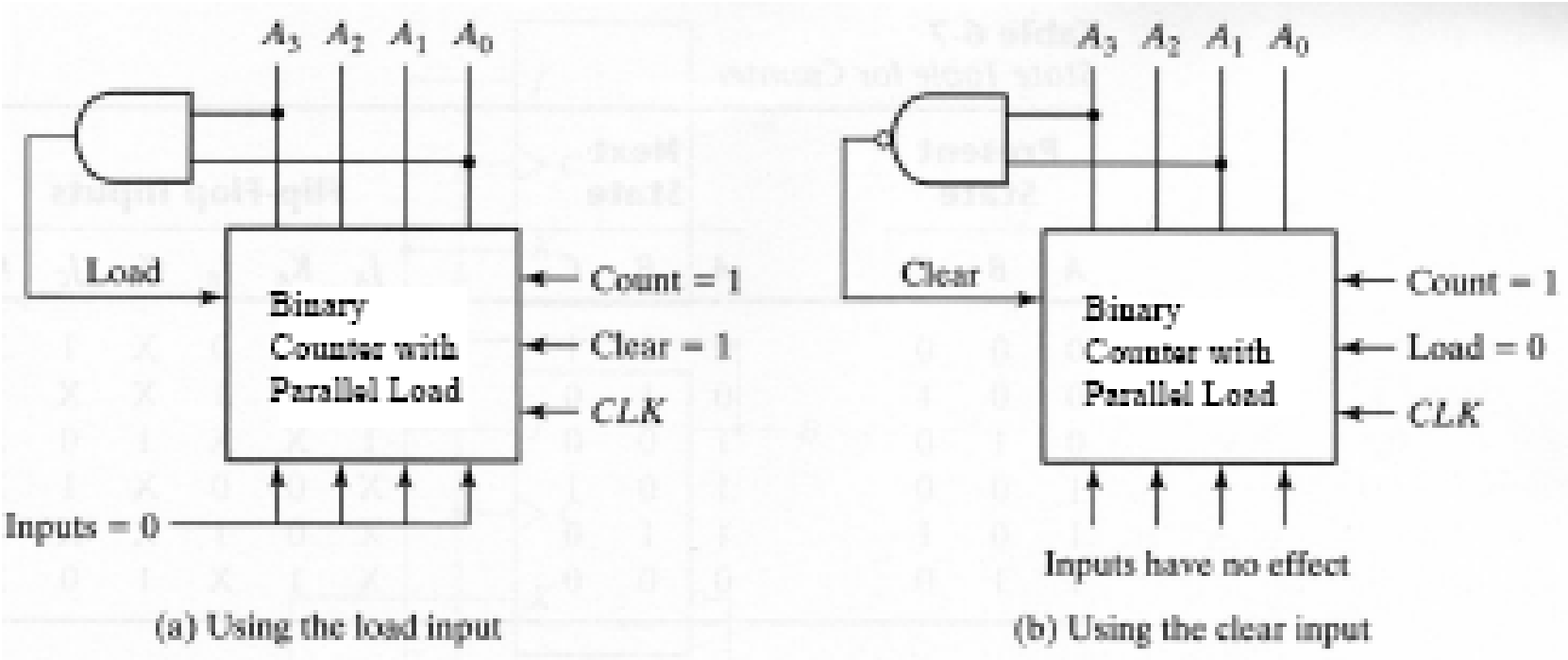
Clear	CLK	Load	Count	Function
0	X	X	X	Clear to 0
1	↑	1	X	Load inputs
1	↑	0	1	Count next binary state
1	↑	0	0	No change

Implementing BCD Counter Using a Binary Counter with Parallel Load

- **Two Ways:**

(a) Using the load input

(b) Using the clear input



Counter with Unused States

- **Example:** **Design a counter that has repeated binary sequence of 000, 001, 010, 100, 101, and 110.**

State Table for Counter

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

Flip-Flop Excitation Tables

Q(t)	Q(t + 1)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

(a) J K

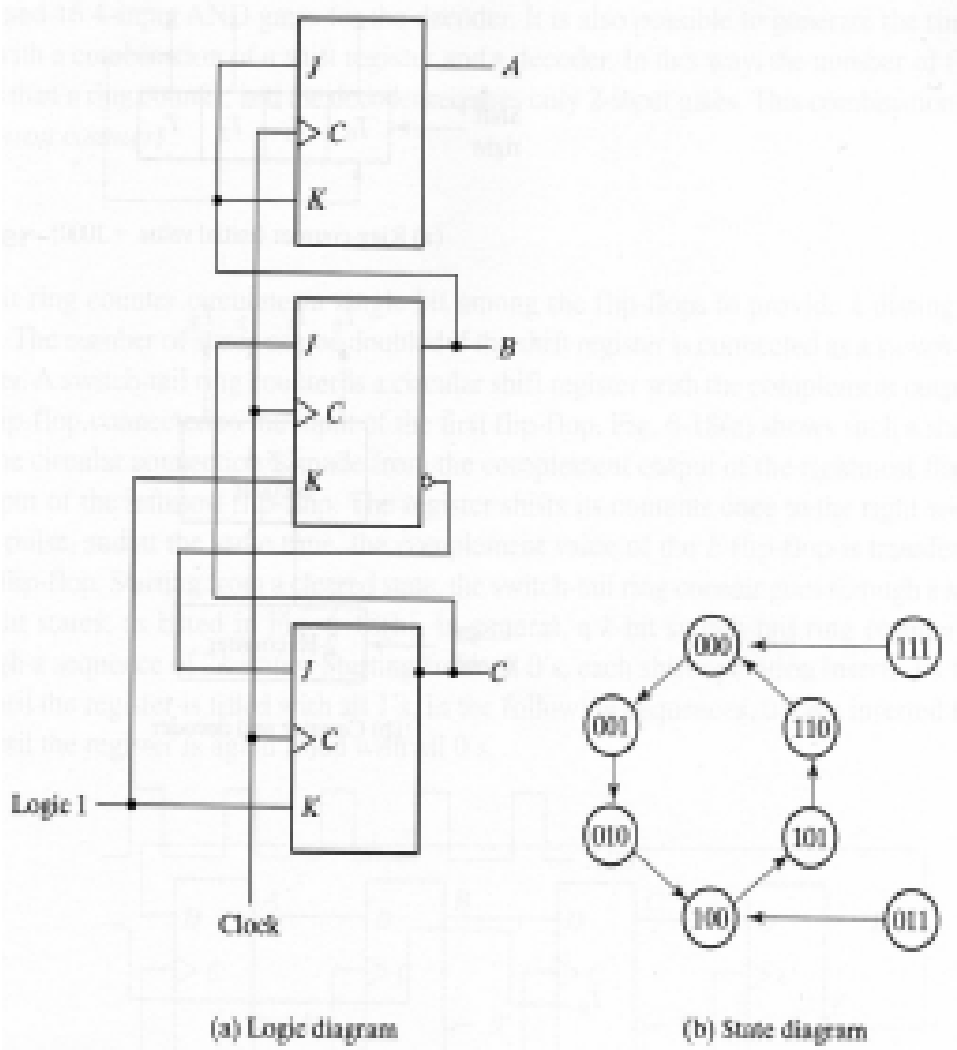
$$J_A = B \quad K_A = B$$

$$J_B = C \quad K_B = 1$$

$$J_C = B' \quad K_C = 1$$

- **The flip-flop input equations can be simplified using minterms 3 and 7 as don't-care conditions.**

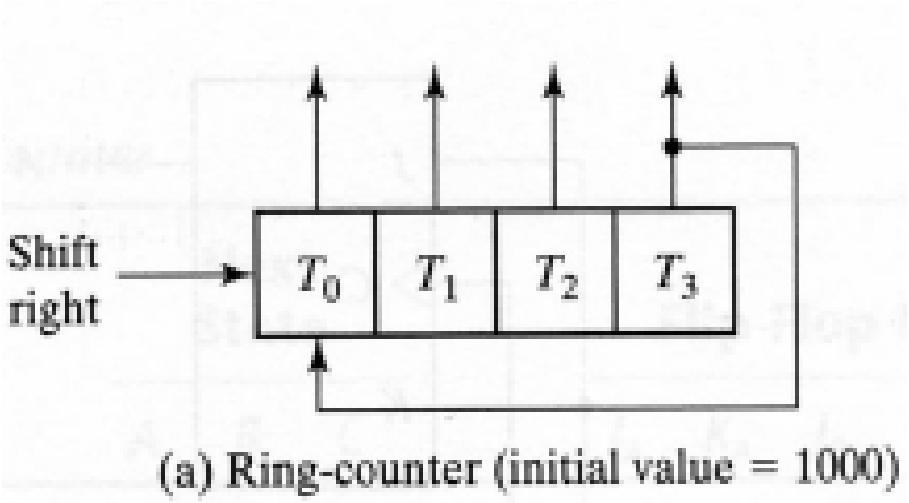
Counter with Unused States



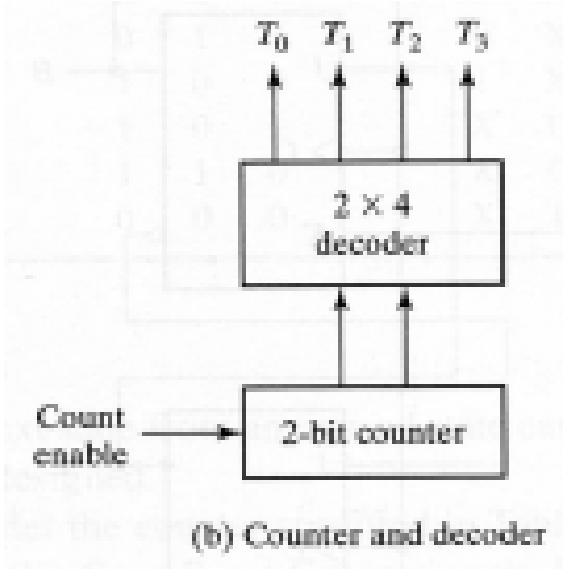
- There are two unused states: 011 and 111.
- A circuit may go to one of the unused states because of an outside interference.
- This counter is self-correcting.

Ring Counter

- **A 4-bit ring counter generates the sequence 1000, 0100, 0010, 0001, and back to 1000.**
- **Two ways to implement a ring counter:**
  - **(a) Using shift register.**
  - **(b) Using a binary counter and decoder.**



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



(b) Counter and decoder