DIGITAL DESIGN AND COMPUTER ORGANIZATION

Synchronous Counters

Team DDCO

Department of Computer Science and Engineering

Synchronous Counter(T1-section 6.4)

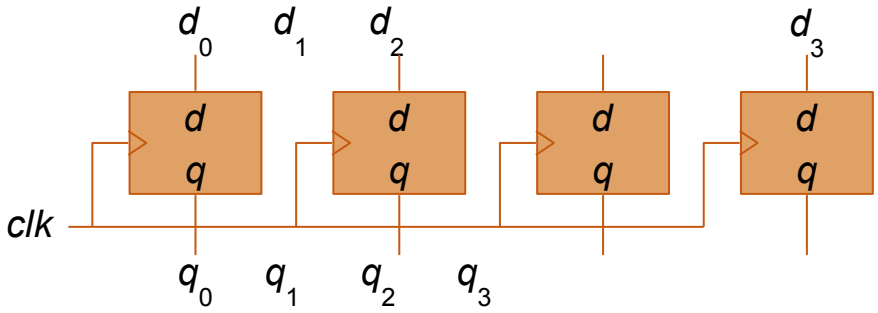
Synchronous counters are different from ripple counters in that clock pulses are applied to the inputs of all flip-flops**. A common clock triggers all flip-flops simultaneously, rather than one at a time in succession as in a ripple counter.**

The decision whether a flip-flop is to be complemented is determined from the values of the data inputs, such as T or J and K at the time of the clock edge. If T = 0 or J = K = 0, the flip-flop does not change state. If T = 1 or J = K = 1, the flip-flop complements

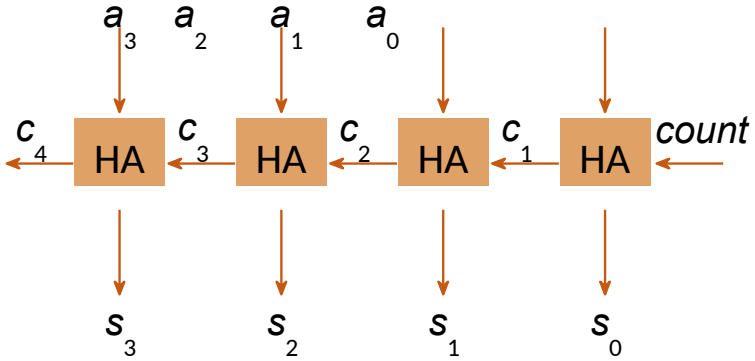
Counters -Recap

Incrementing Counters- Example- count number of people entering a mall

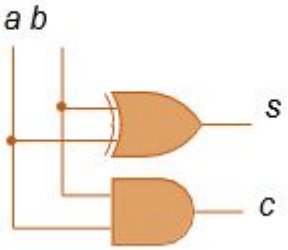
- How to store *n*-bits? *n*-bit register



- How to increment an *n*-bit number? *n*-bit incrementer

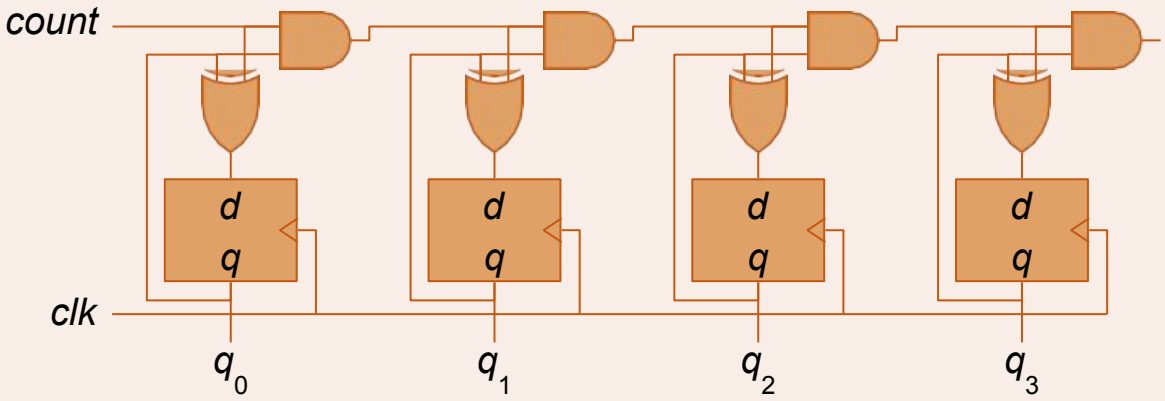


- Half adder logic circuit:



Incrementing Counters

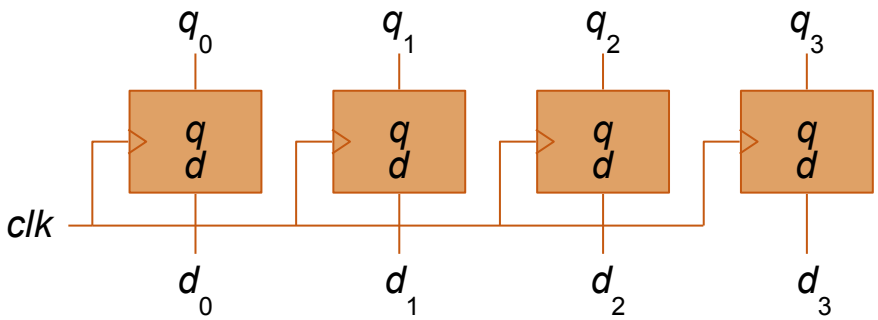
4-Bit Incrementing Counter



Counters Recap –Decrementing counters

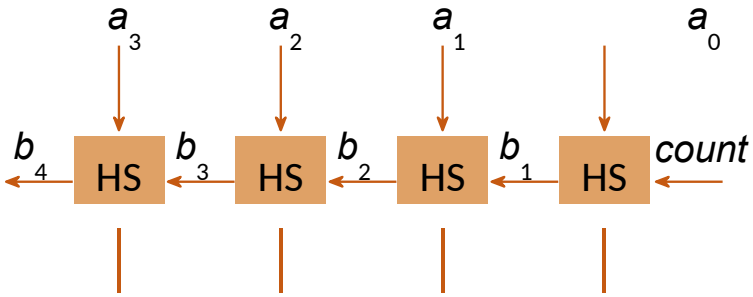
- An *n*-bit counter that counts back from $2^n - 1$ to 0 (and back to $2^n - 1$)
  - Ex: a 2-bit counter, counts 11, 10, 01, 00, 11, 10, ….

- How to store *n*-bits? *n*-bit register

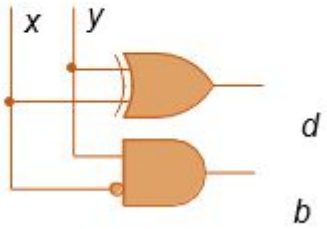


Parking lot counter: Starts at 20 and decrements by 1 for each car entering, displaying remaining slots until it reaches 0 → "Parking Full."

- How to decrement an n-bit number? n-bit decrementer

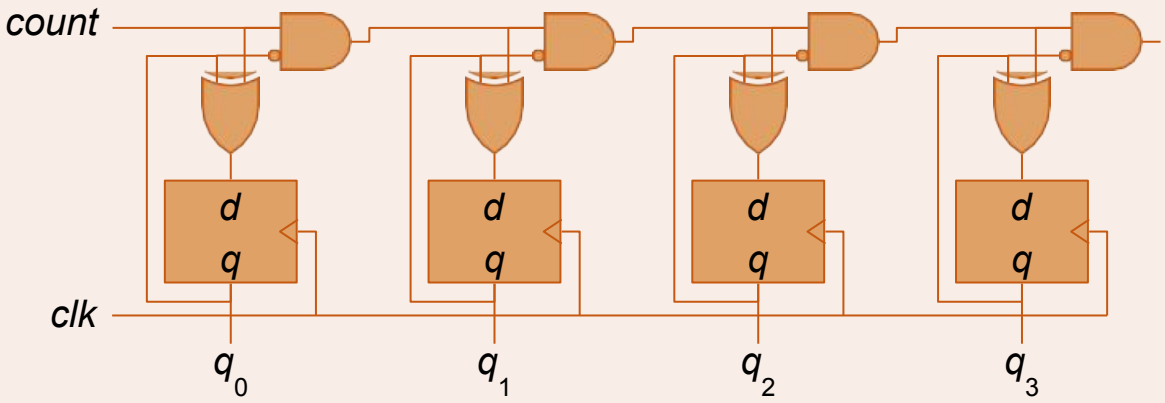


- Half Subtractor Logic Circuit:



4-bit Decrementing Counters

4-Bit Decrementing Counter



Synchronous Counters

Binary Counters

In a synchronous binary counter, **the flip-flop in the least significant position is complemented with every pulse. A flip-flop in any other position is complemented when all the bits in the lower significant positions are equal to 1** . For example, if the present state of a four-bit counter is A3A2A1A0 = 0011, the next count is 0100. A0 is always complemented. A1 is complemented because the present state of A0 = 1. A2 is complemented because the present state of A1A0 = 11. However, A3 is not complemented, because the present state of A2A1A0 = 011, which does not give an all-1's condition.

The synchronous counter can be triggered with either the positive or the negative clock edge.

Synchronous Counters

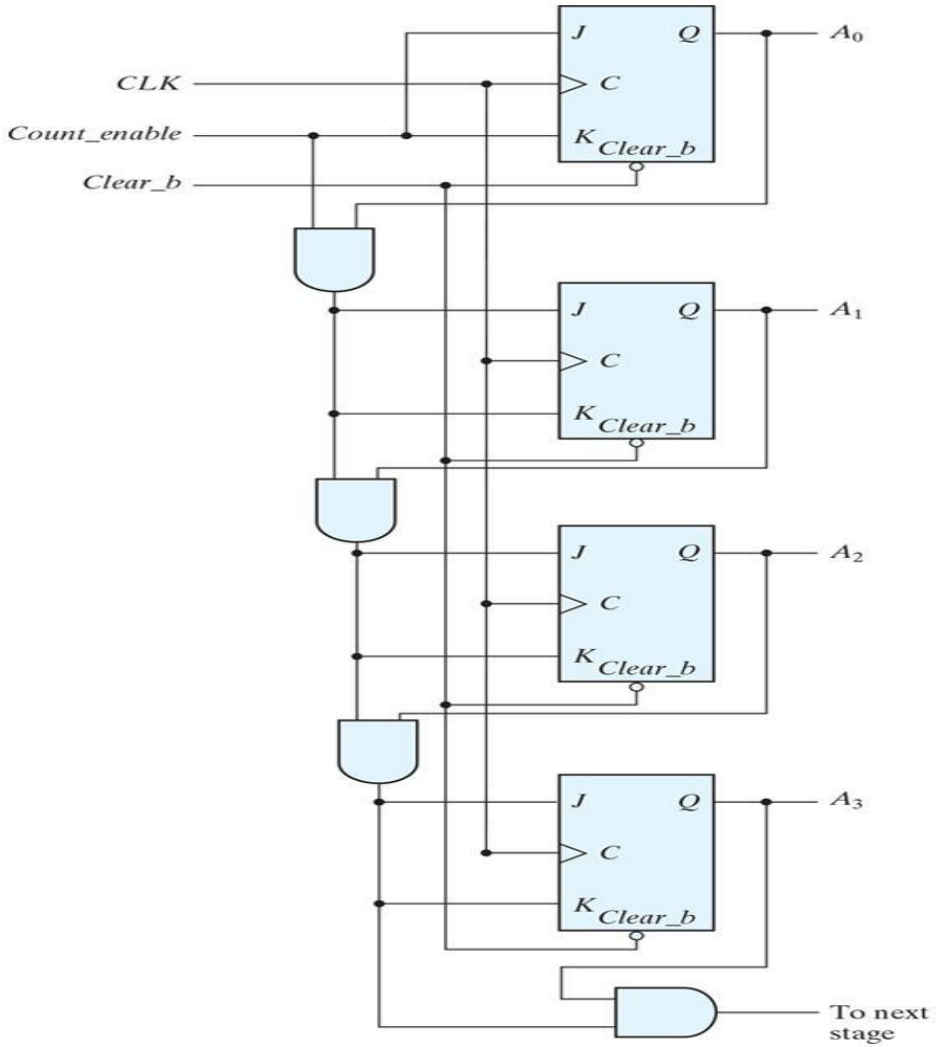


Figure 6.12 Four-bit synchronous binary counter.

Synchronous Counters

The C inputs of all flip-flops are connected to a common clock. The counter is enabled by Count_enable. **If the enable input is 0, all J and K inputs are equal to 0 and the clock does not change the state of the counter.** The first stage, A0, has its J and K equal to 1 if the counter is enabled. The other J and K inputs are equal to 1 if all previous least significant stages are equal to 1 and the count is enabled. The chain of AND gates generates the required logic for the J and K inputs in each stage. **The counter can be extended to any number of stages, with each stage having an additional flip-flop and an AND gate that gives an output of 1 if all previous flip-flop outputs are 1.**

Synchronous Counters

Steps to design the counter using flipflops:

1. Decide the number of flipflop
2. Use excitation of the flipflop
3. State table and ckt excitation table
4. Obtain simplified equation using K map
5. Draw the sequential ckt

https://www.tutorialspoint.com/digital-electronics/desi
-of-synchronous-counter.htm

Synchronous Counters

Design 2 bit up synchronous counter using JK flipflop:
Also called Mod 4 counter

Synchronous Counters

Present (Q1 Q0)	Next ($Q1^+$ $Q0^+$)
00	01
01	10
10	11
11	00

Excitation table

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

Pearson

Synchronous Counters

Present (Q1 Q0)	Next ($Q1^+$ $Q0^+$)	inputs			
		J1	K1	J0	k0
00	01	0	X	1	X
01	10	1	x	x	1
10	11	x	0	1	X
11	00	x	1	x	1

Excitation table

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

Solving using 2 variable K map of present state

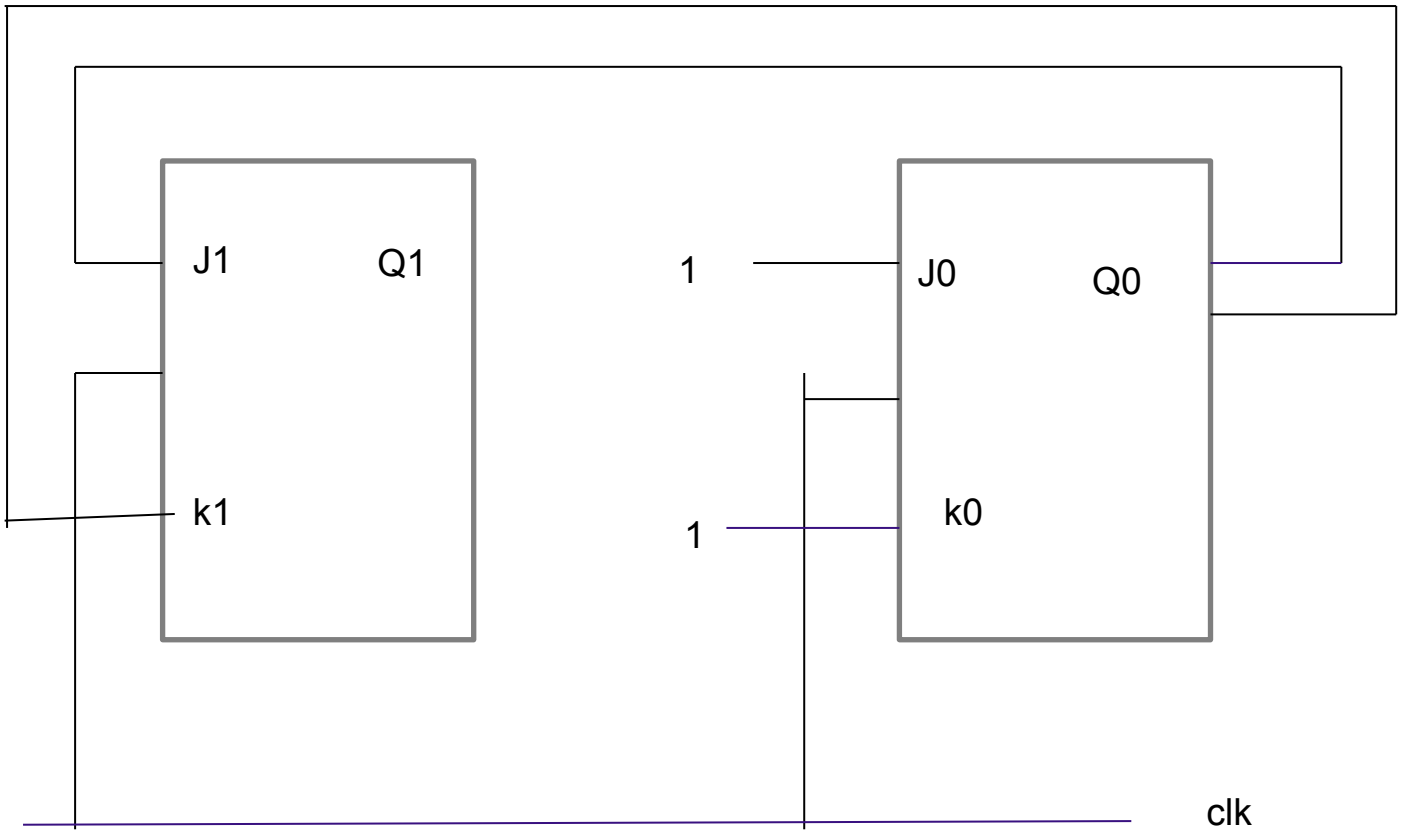
J0 = 1, **K0 = 1**

J1 = Q0, **K1 = Q0**

Pearson

Synchronous Counters

J0 = 1, K0 = 1
J1 = Q0, K1 = Q0



Synchronous Counters

Additional questions/assignment:

1. Design 2 bit synchronous down counter using JK flipflop
2. Design 3 bit synchronous down counter using T flipflop

Synchronous Counters

BCD Counter.

A BCD counter counts in binary-coded decimal from 0000 to 1001 and back to 0000. Because of the return to 0 after a count of 9, a BCD counter does not have a regular pattern, unlike a straight binary count. To derive the circuit of a BCD synchronous counter, it is necessary to go through a sequential circuit design procedure.

BCD Counters o decade counter

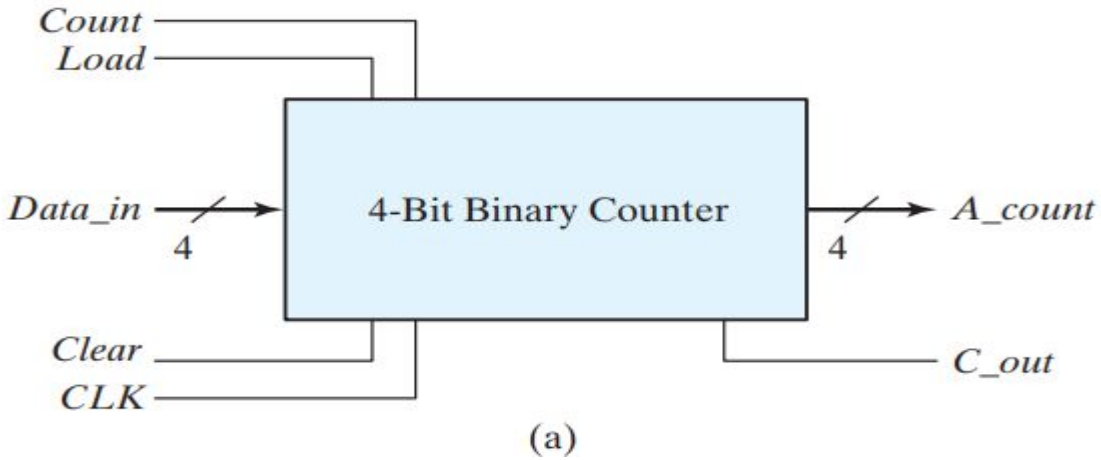
Present State				Next State				Output	Flip-Flop Inputs			
Q_8	Q_4	Q_2	Q_1	Q_8	Q_4	Q_2	Q_1	y	T_{Q8}	T_Q4	T_{Q2}	T_{Q1}
0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	1	0	0	1	0	0	0	0	1	1
0	0	1	0	0	0	1	1	0	0	0	0	1
0	0	1	1	0	1	0	0	0	0	1	1	1
0	1	0	0	0	1	0	1	0	0	0	0	1
0	1	0	1	0	1	1	0	0	0	0	1	1
0	1	1	0	0	1	1	1	0	0	0	0	1
0	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

Table 6.5
State Table for BCD Counter.

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

M. Morris Mano, Michael D. Ciletti, *Digital Design: With an Introduction to the Verilog HDL*, 5th ed., Prentice Hall, 2012, Section 6.4.

Binary Counter with Parallel Load



(a)

Count/Load: A control input that selects whether the counter should count or load data. If Count is active, the counter increments based on the clock. If Load is active, a 4-bit value is loaded into the counter.

Data_in (4 lines): This provides the data to be loaded into the 4-bit counter when the Load signal is active.
A_count (4 lines): The 4-bit output that holds the current count of the counter.

Clear: Resets the counter to zero.

CLK: The clock signal that controls the counting operation.

C_out: A carry-out signal that goes high when the counter overflows (from 15 back to 0).

Binary Counter with Parallel Load

Table 6.6
Function Table for the Counter of Fig. 6.14.

Clear_b	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

Binary Counter with Parallel Load

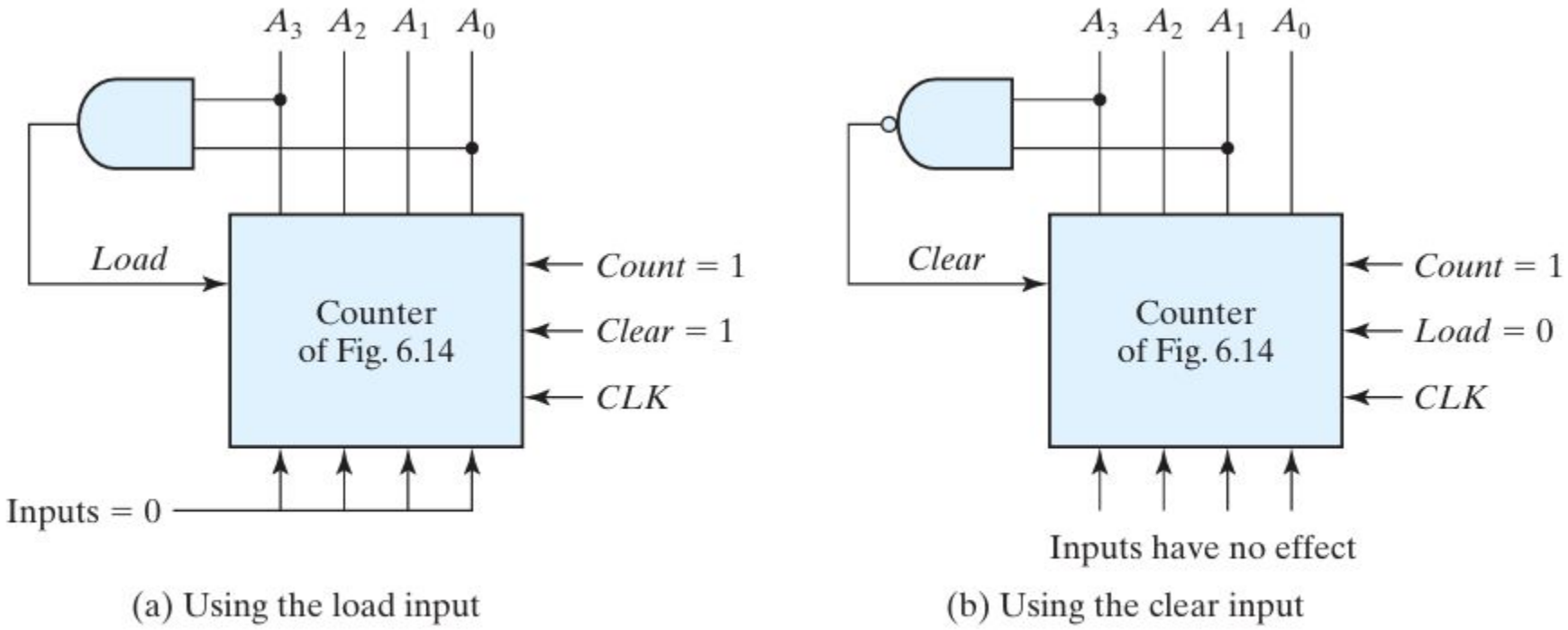


FIGURE 6.15
Two ways to achieve a BCD counter using a counter with parallel load

Synchronous Counters application

Example: Image Processing with Synchronous Counters

- **Scenario**

Suppose we have a grayscale image of 256 × 256 pixels stored in memory.
Each pixel = 1 byte, and all pixels are stored sequentially in RAM.
Pixel(0,0) → Address 0000h
Pixel(0,1) → Address 0001h…

Pixel(255,255) → Address FFFFh

How Synchronous Counter Helps?
A 16-bit synchronous counter acts as the address generator.
On each clock pulse, the counter increments → next memory address is produced instantly.
The memory outputs the pixel value at that address.
Why Synchronous Counter?
All flip-flops update together → no glitching on the address bus.
Essential for high-speed video/image systems (e.g., cameras, medical imaging, graphics cards).

MCQ

A MOD-10 synchronous counter is implemented. What will be the count sequence of the counter?
(A) 0000 to 1111
(B) 0000 to 1001
(C) 0001 to 1110
(D) 0000 to 1010

A synchronous counter is designed to count in the sequence: 000 → 010 → 011 → 101 → 110 → 000. What is the modulus of this counter?
(A) 3
(B) 5
(C) 6
(D) 7

MCQ

A MOD-10 synchronous counter is implemented. What will be the count sequence of the counter?
(A) 0000 to 1111
(B) 0000 to 1001
(C) 0001 to 1110
(D) 0000 to 1010

Answer: (B) 0000 to 1001

A synchronous counter is designed to count in the sequence: 000 → 010 → 011 → 101 → 110 → 000. What is the modulus of this counter?
(A) 3
(B) 5
(C) 6
(D) 7
Answer: (B) 5, The modulus (MOD-N) of a counter = number of unique states it cycles through before repeating.

THANK YOU

Team DDCO

Department of Computer Science and Engineering