



CprE 281: Digital Logic

Instructor: Alexander Stoytchev

<http://www.ece.iastate.edu/~alexs/classes/>

Counters & Solved Problems

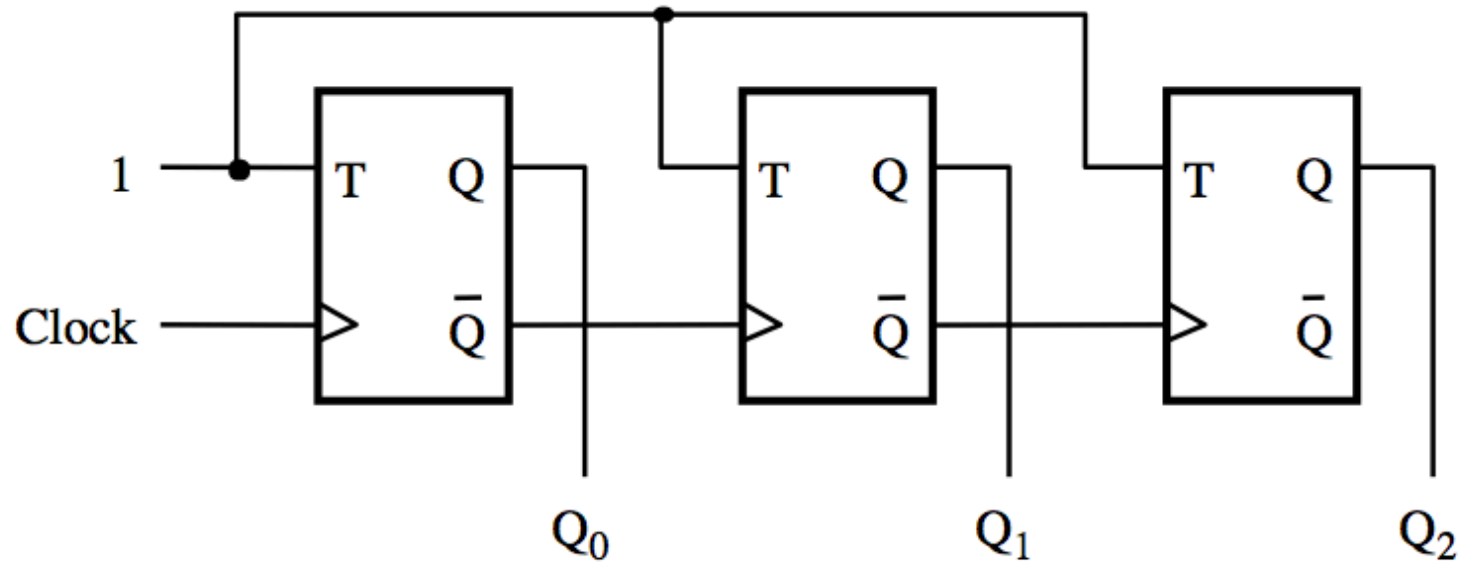
CprE 281: Digital Logic
Iowa State University, Ames, IA
Copyright © 2013

Administrative Stuff

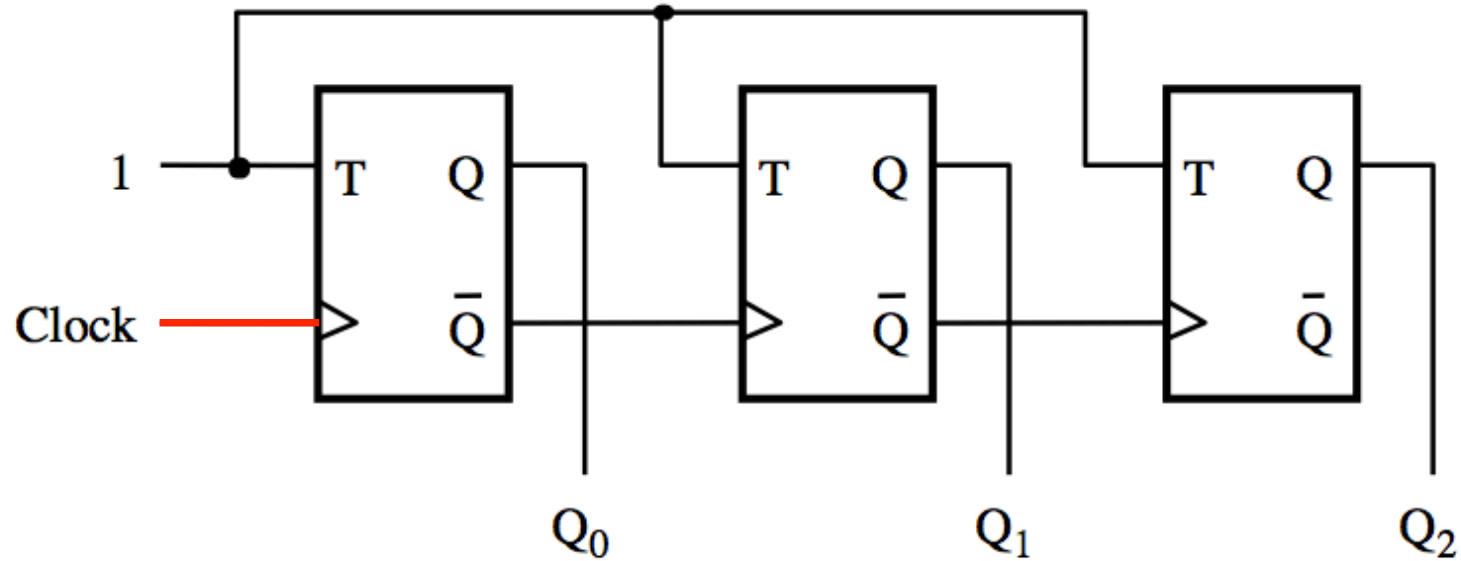
- **Homework 9 is out**
- **It is due on Monday Oct 9, 2015**

Quick Review: Counters

A three-bit up-counter

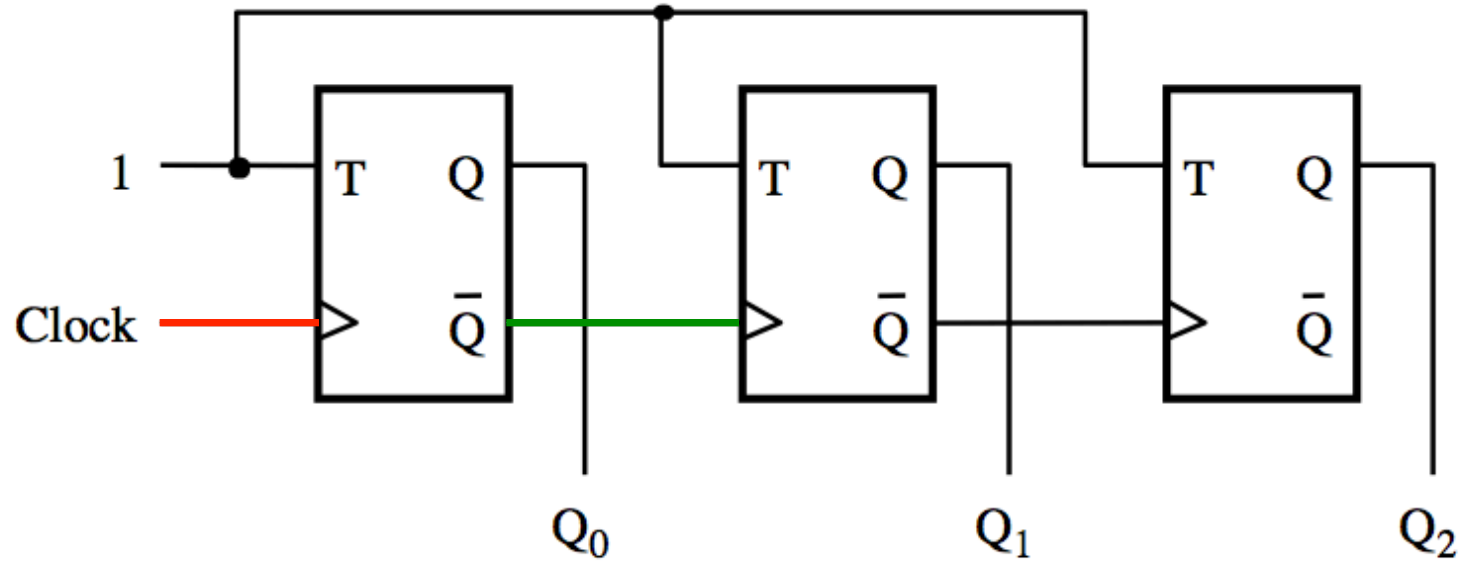


A three-bit up-counter



The first flip-flop changes
on the positive edge of the clock

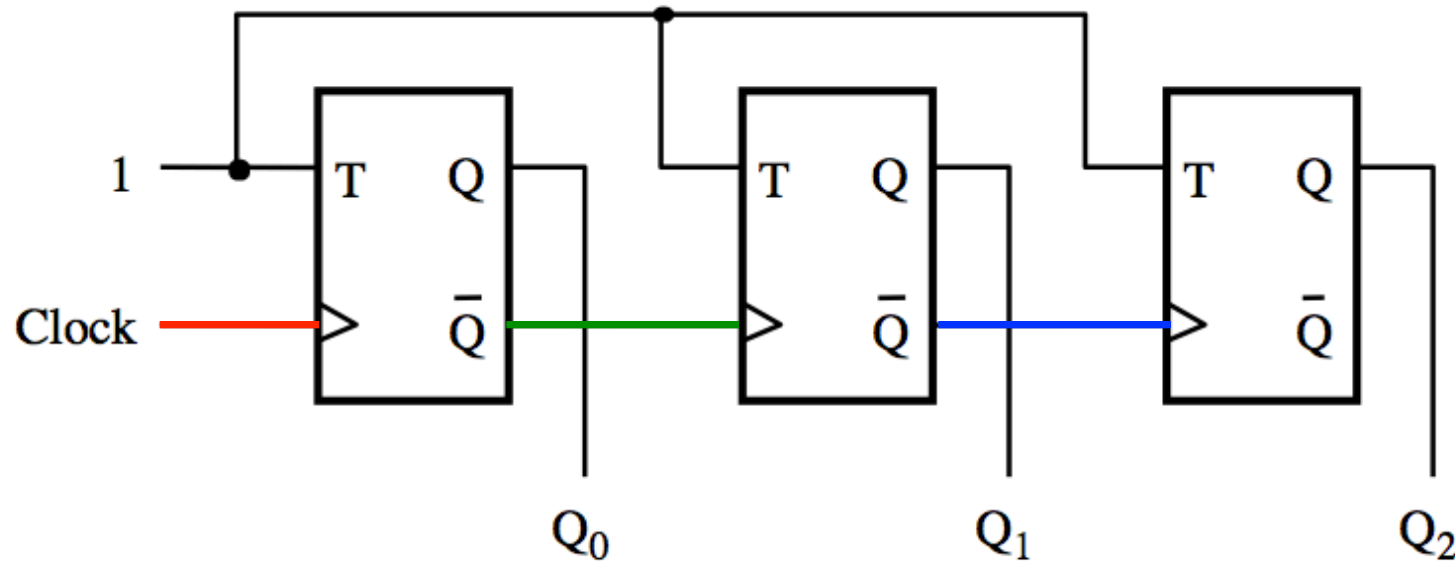
A three-bit up-counter



The first flip-flop changes
on the positive edge of the clock

The second flip-flop changes
on the positive edge of \bar{Q}_0

A three-bit up-counter

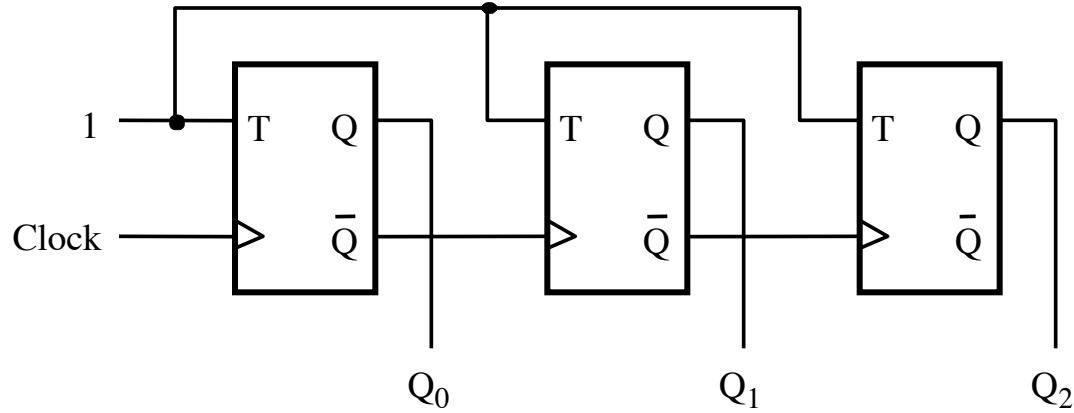


The first flip-flop changes on the positive edge of the clock

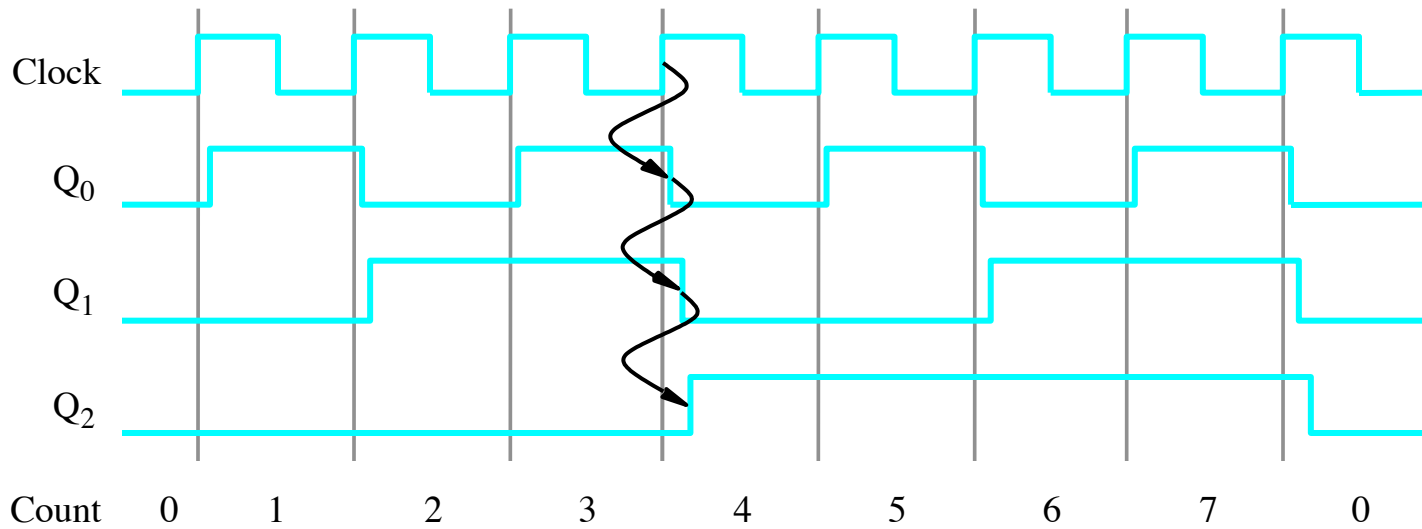
The second flip-flop changes on the positive edge of \bar{Q}_0

The third flip-flop changes on the positive edge of \bar{Q}_1

A three-bit up-counter

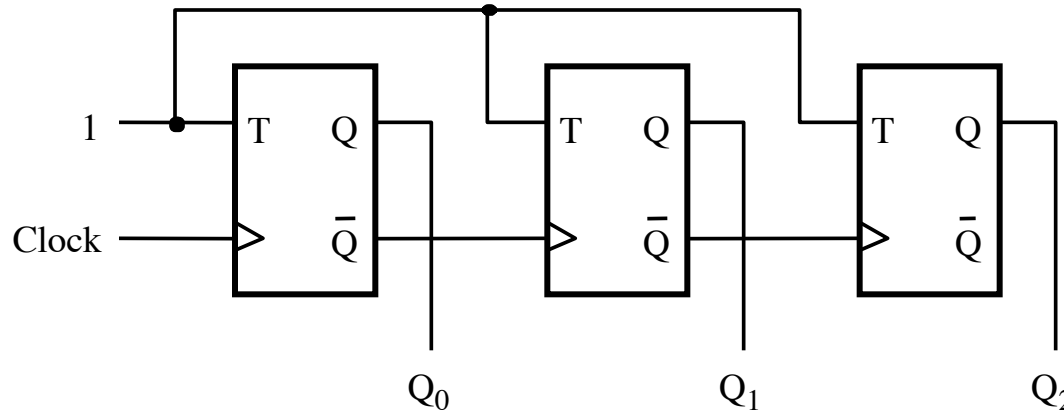


(a) Circuit



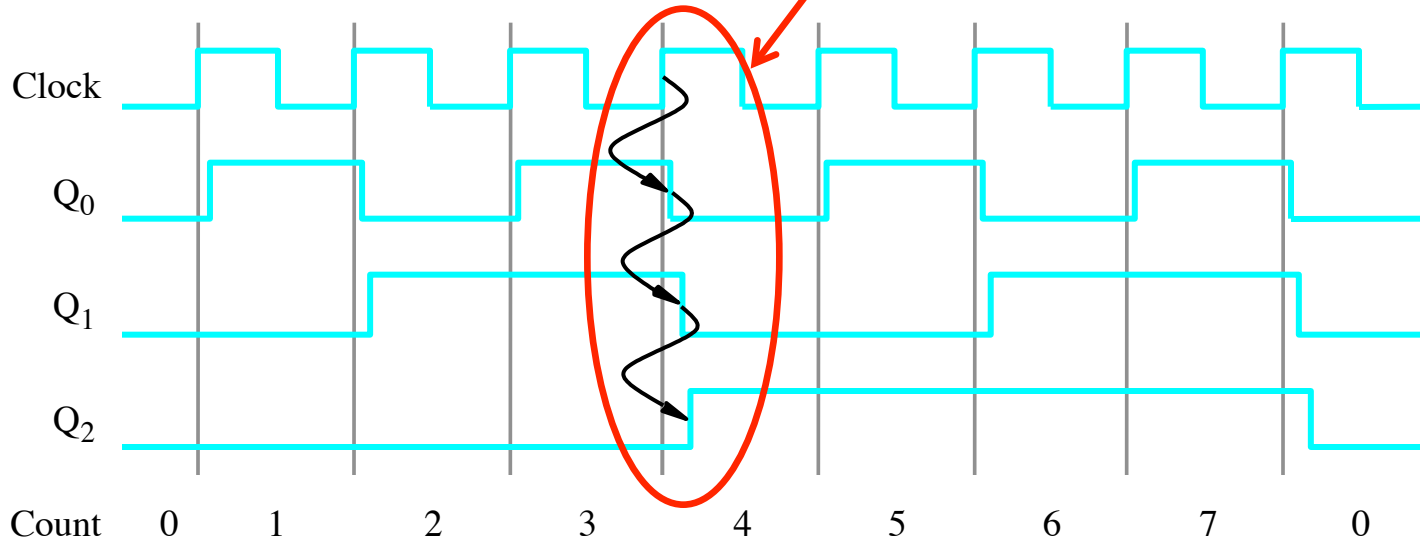
(b) Timing diagram

A three-bit up-counter



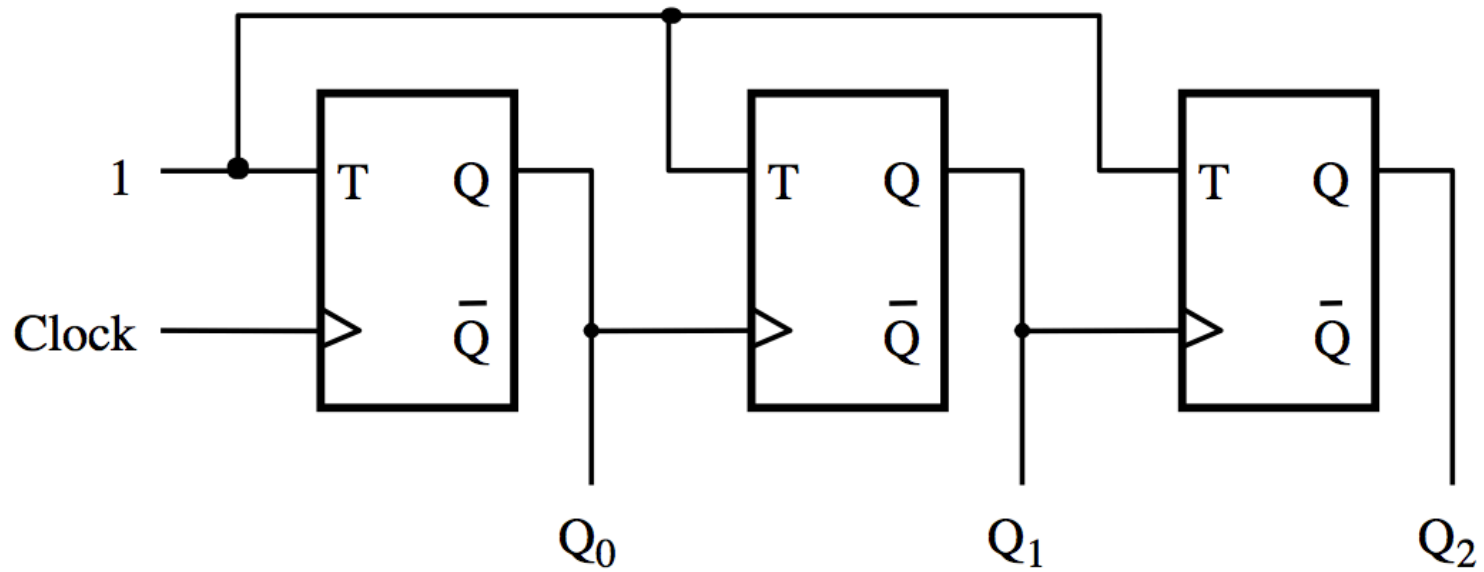
(a) Circuit

The propagation delays get longer

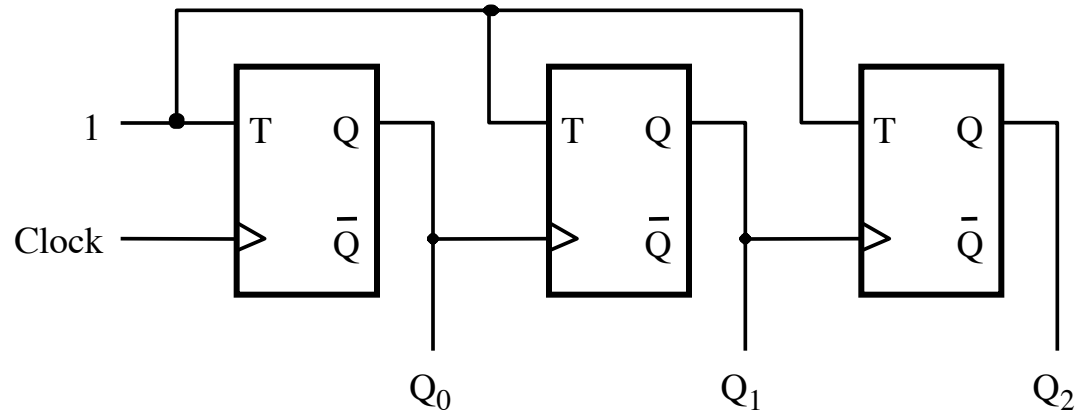


(b) Timing diagram

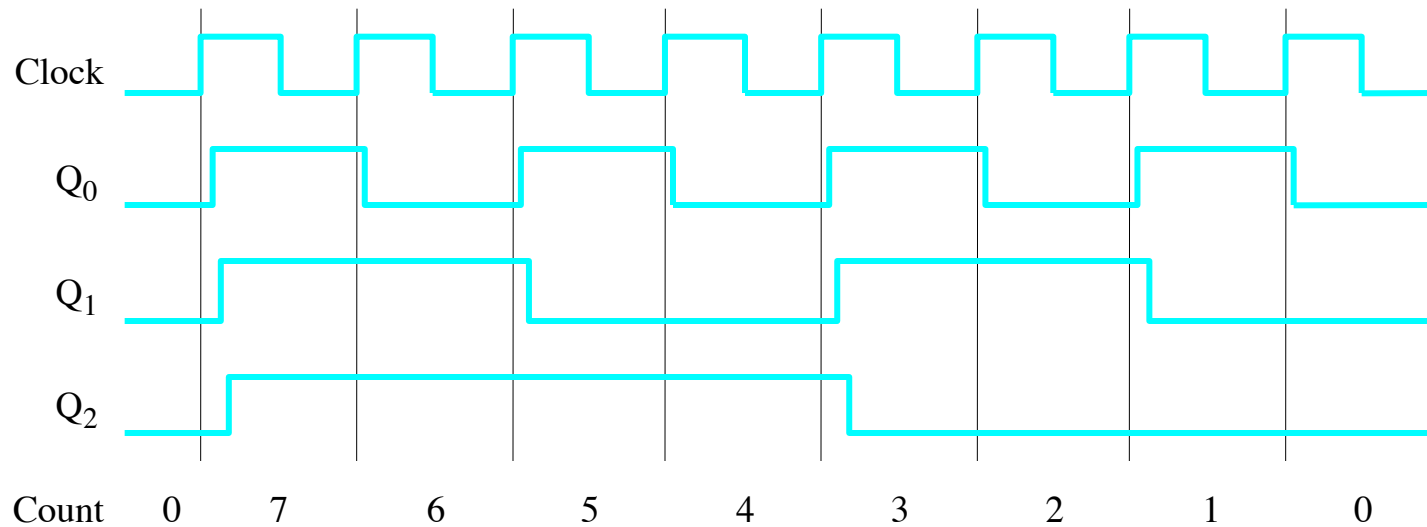
A three-bit down-counter



A three-bit down-counter



(a) Circuit

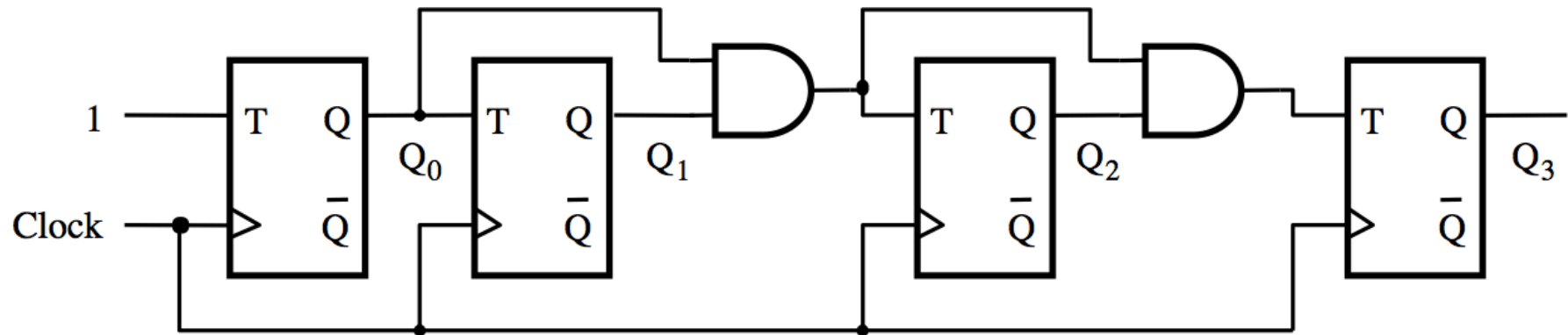


(b) Timing diagram

[Figure 5.20 from the textbook]

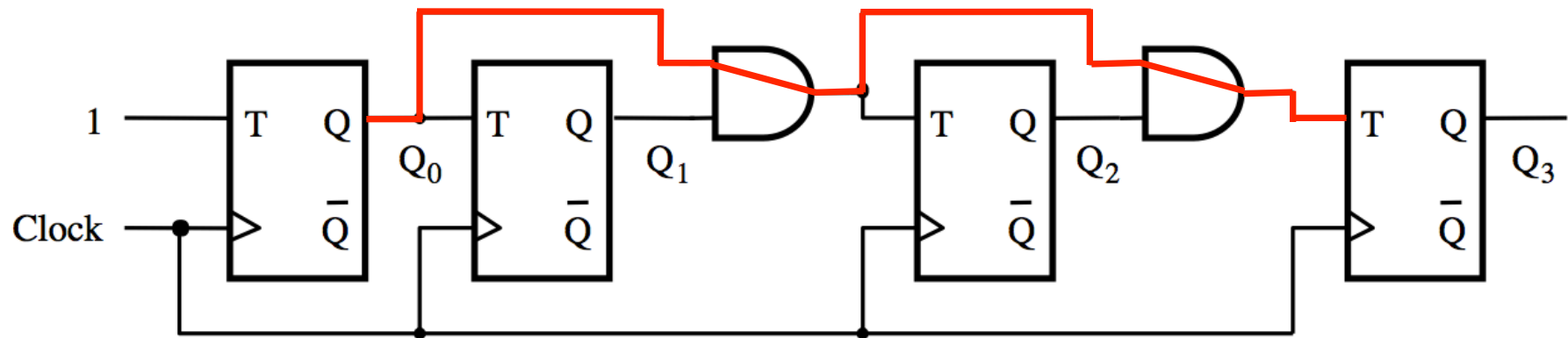
Synchronous Counters

A four-bit synchronous up-counter



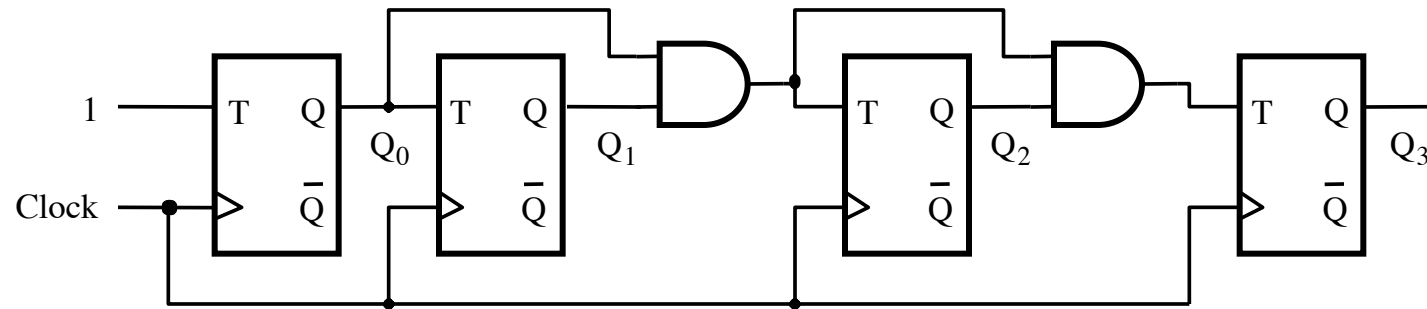
[Figure 5.21 from the textbook]

A four-bit synchronous up-counter

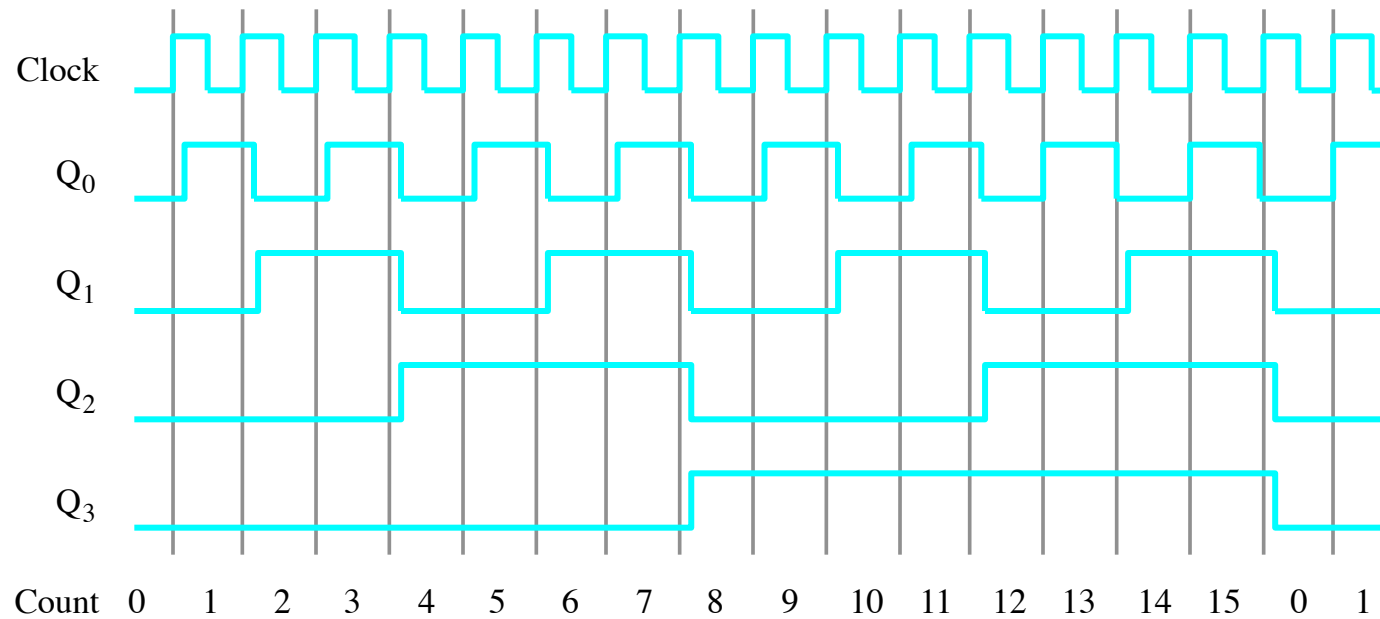


The propagation delay through all AND gates combined must not exceed the clock period minus the setup time for the flip-flops

A four-bit synchronous up-counter



(a) Circuit



(b) Timing diagram

[Figure 5.21 from the textbook]

Derivation of the synchronous up-counter

Clock cycle	Q ₂	Q ₁	Q ₀
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0

Derivation of the synchronous up-counter

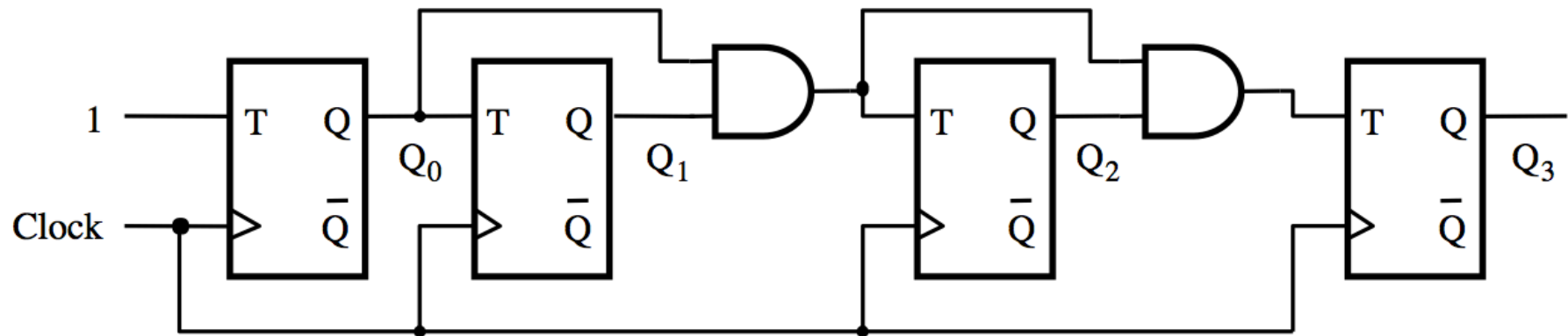
Clock cycle	Q ₂	Q ₁	Q ₀
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0

$$T_0 = 1$$

$$T_1 = Q_0$$

$$T_2 = Q_0 Q_1$$

A four-bit synchronous up-counter



$$T_0 = 1$$

$$T_1 = Q_0$$

$$T_2 = Q_0 Q_1$$

In general we have

$$T_0 = 1$$

$$T_1 = Q_0$$

$$T_2 = Q_0 Q_1$$

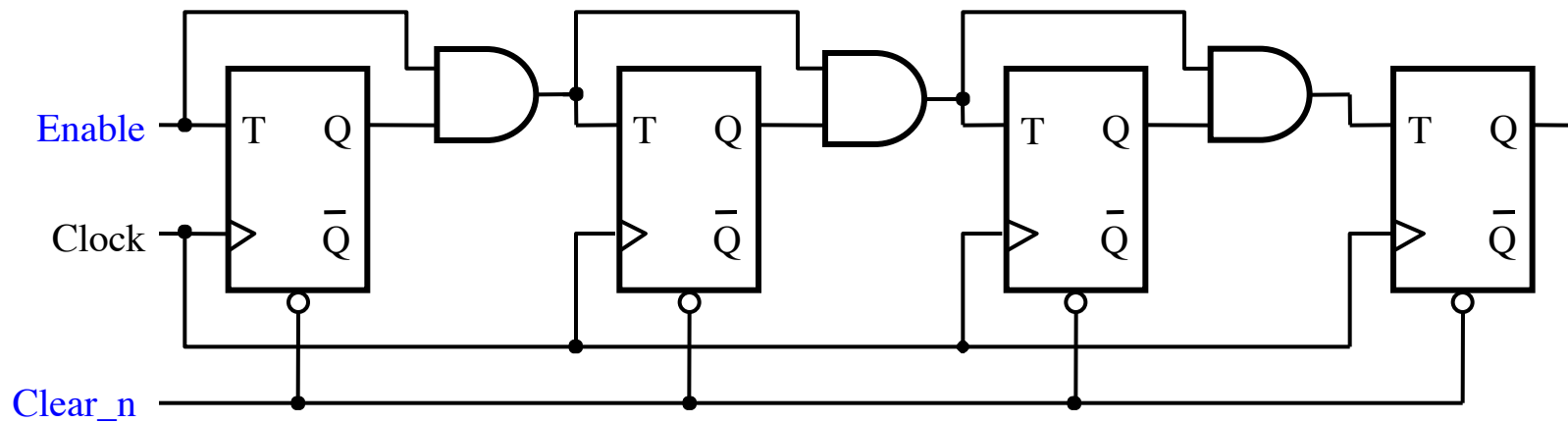
$$T_3 = Q_0 Q_1 Q_2$$

...

$$T_n = Q_0 Q_1 Q_2 \cdots Q_{n-1}$$

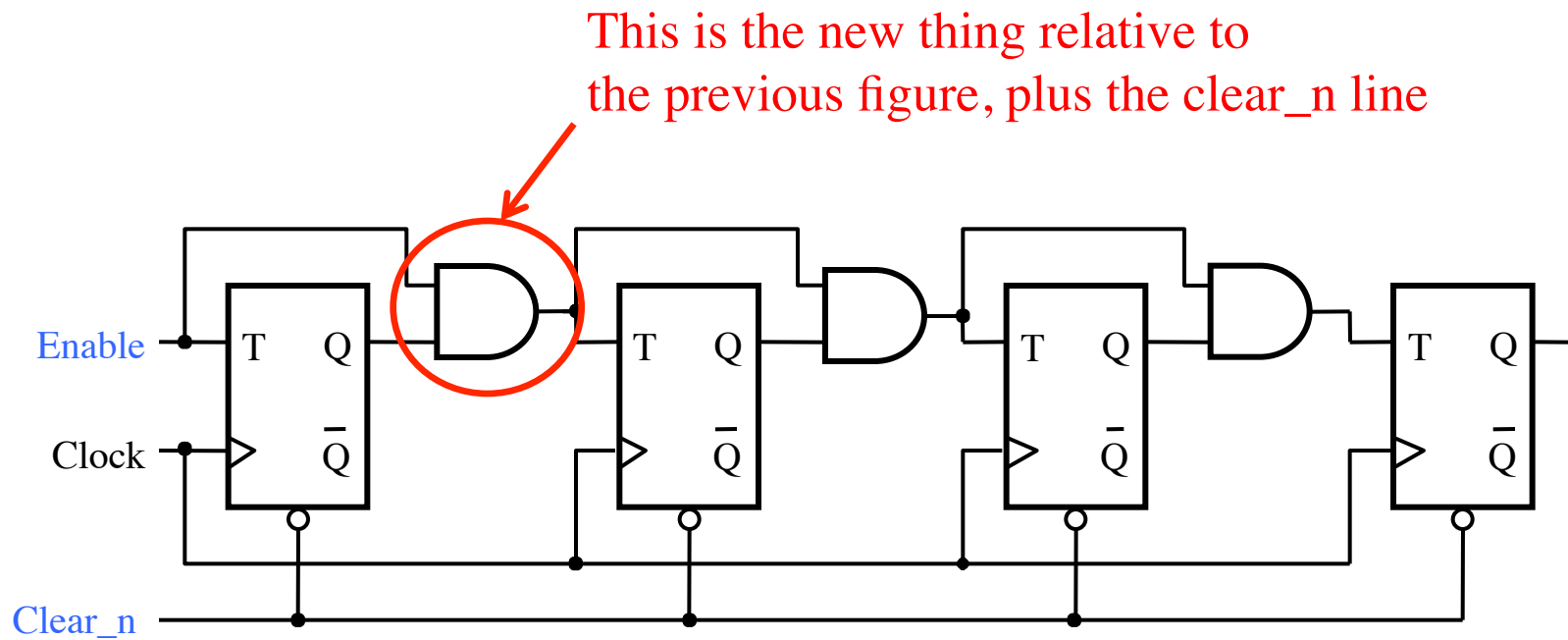
Adding Enable and Clear Capability

Inclusion of Enable and Clear capability

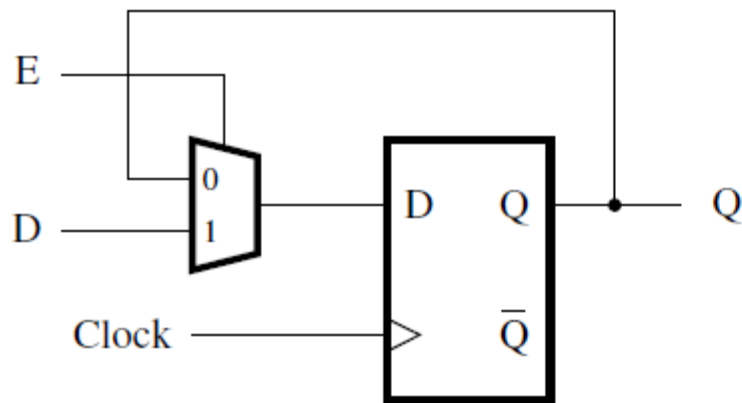


[Figure 5.22 from the textbook]

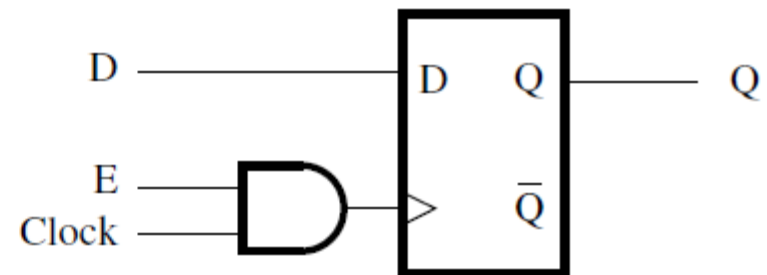
Inclusion of Enable and Clear capability



Providing an enable input for a D flip-flop



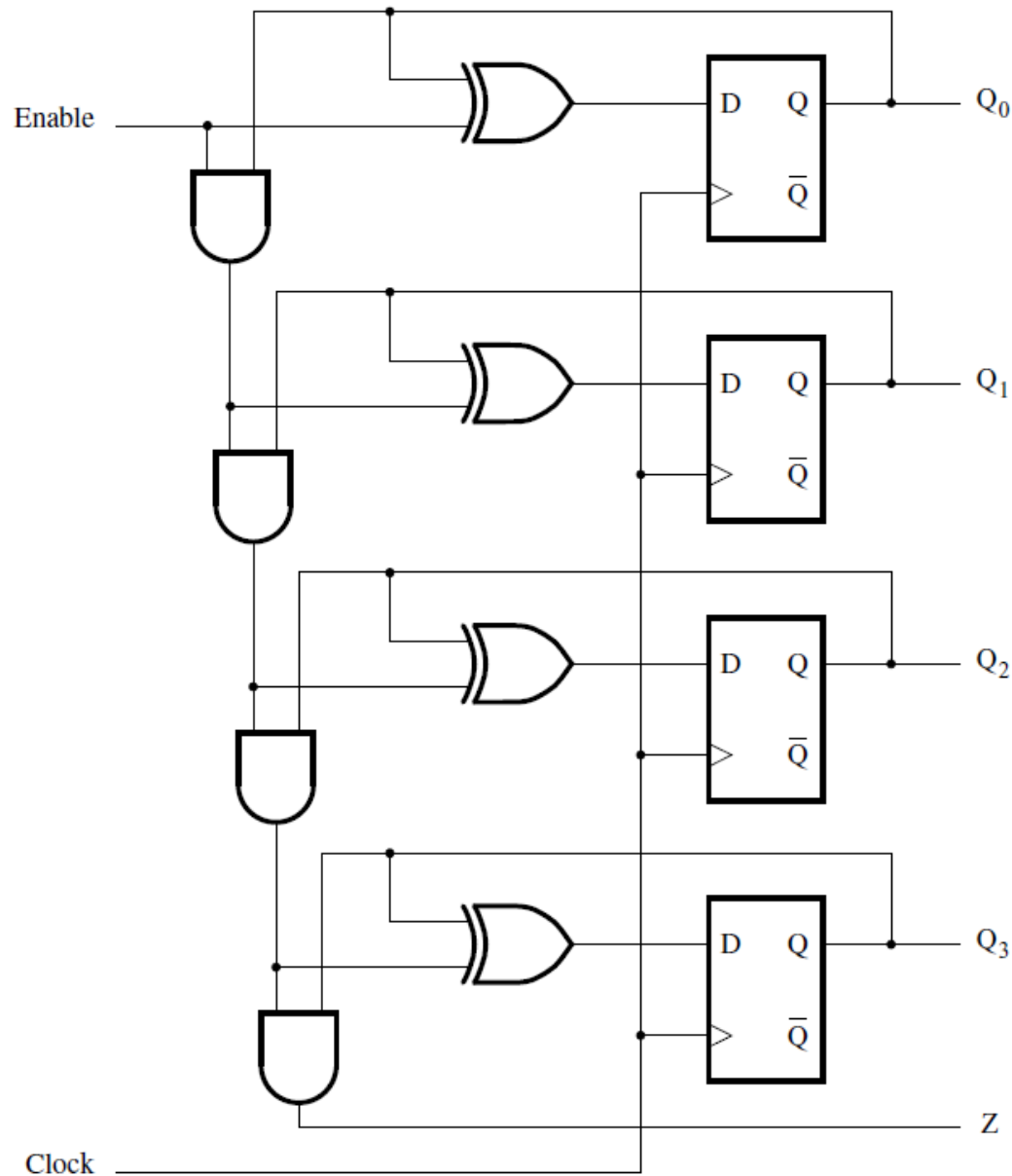
(a) Using a multiplexer



(b) Clock gating

Synchronous Counter with D Flip-Flops

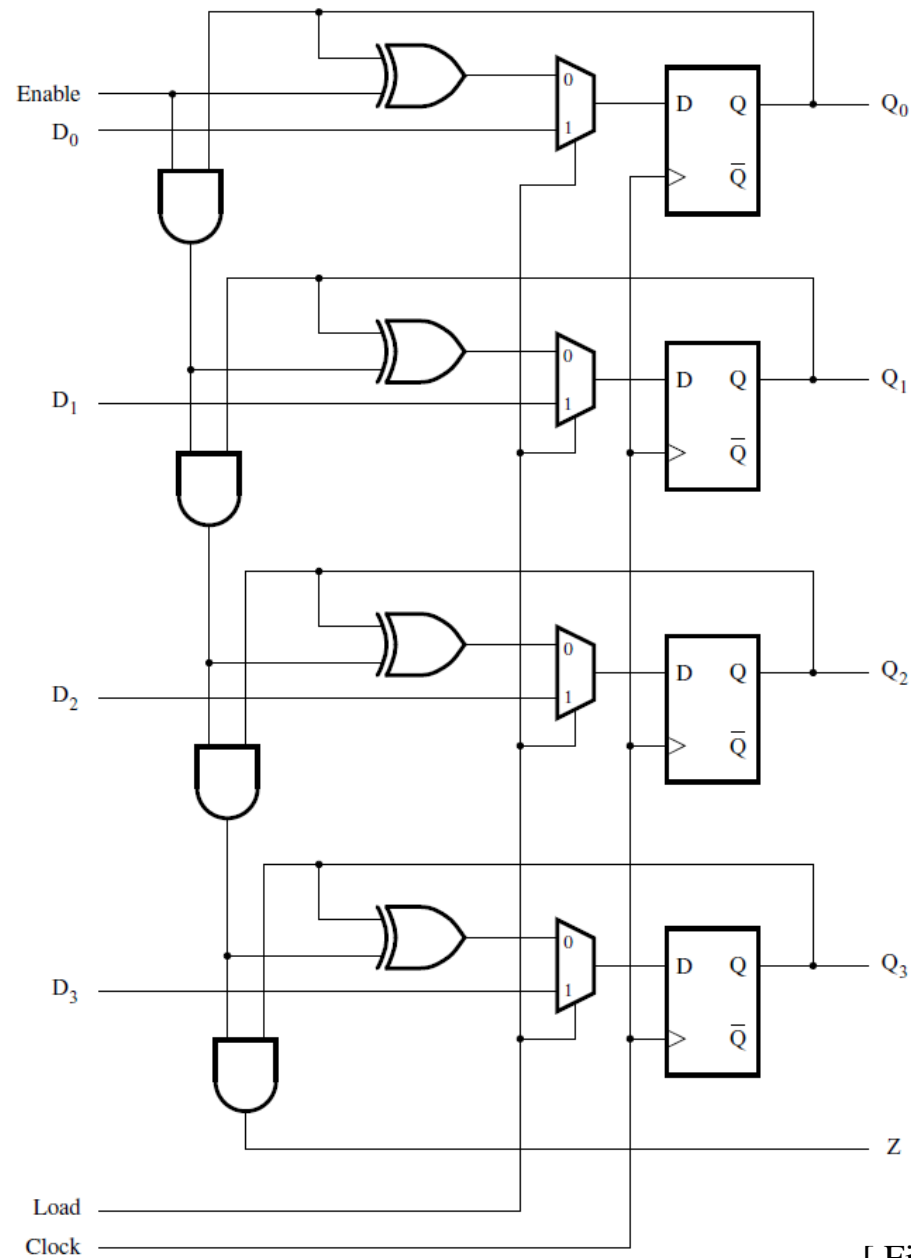
A four-bit counter with D flip-flops



[Figure 5.23 from the textbook]

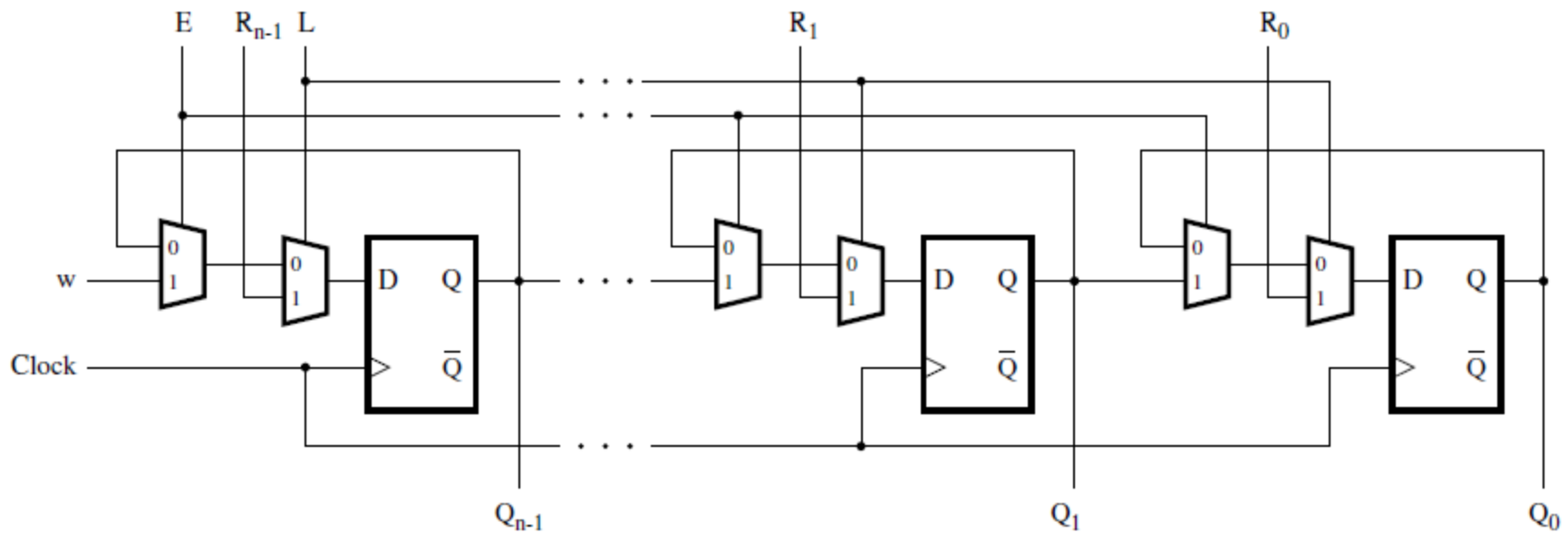
Counters with Parallel Load

A counter with parallel-load capability



[Figure 5.24 from the textbook]

A shift register with parallel load and enable control inputs



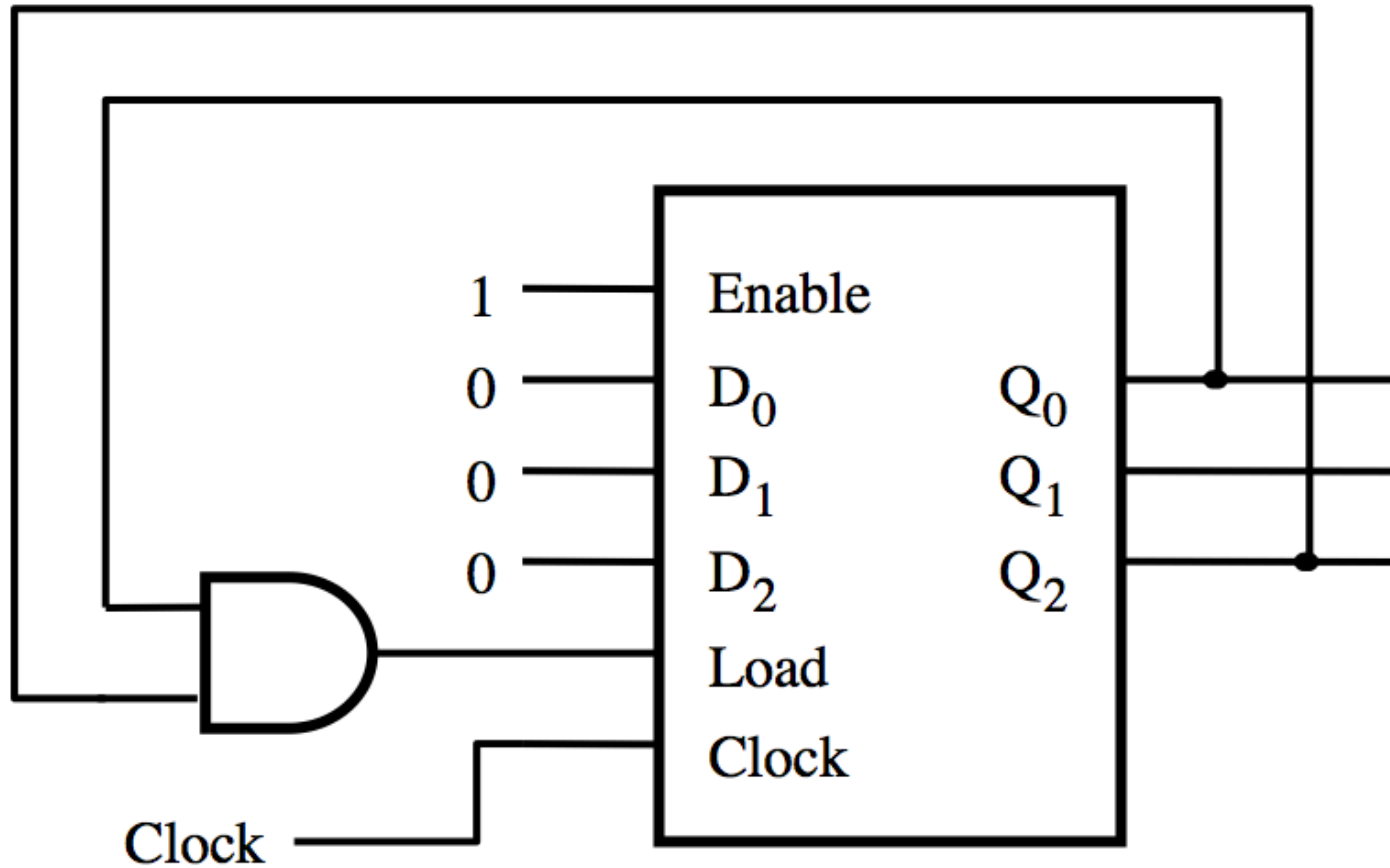
[Figure 5.59 from the textbook]

Reset Synchronization

Motivation

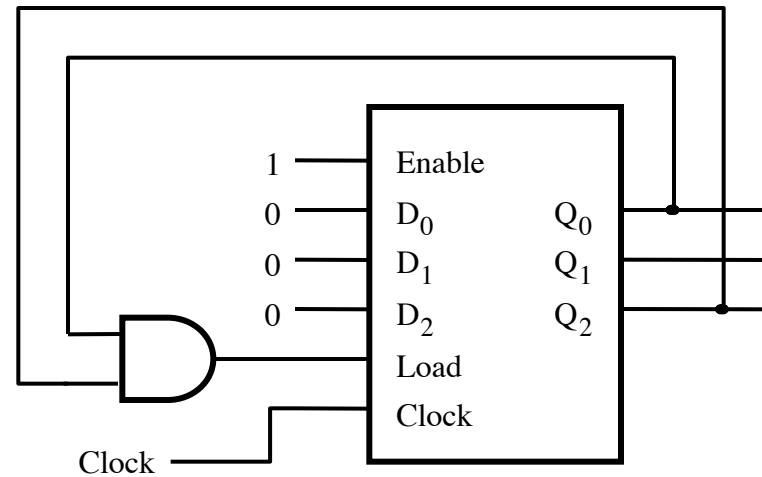
- An n-bit counter counts from 0, 1, ..., 2^n-1
- For example a 3-bit counter counts up as follow
 - 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, ...
- What if we want it to count like this
 - 0, 1, 2, 3, 4, 5, 0, 1, 2, 3, 4, 5, 0, 1, ...
- In other words, what is the cycle is not a power of 2?

What does this circuit do?

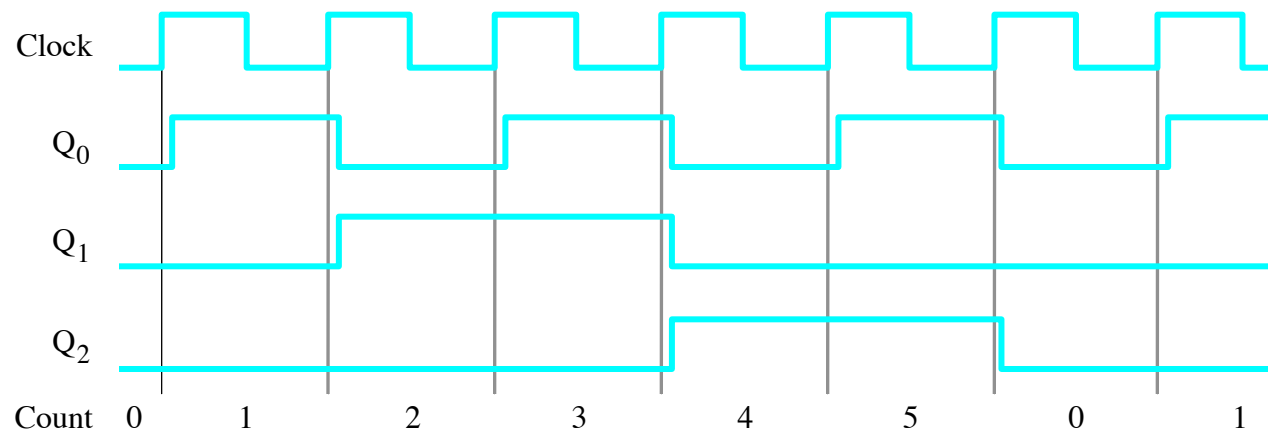


[Figure 5.25a from the textbook]

A modulo-6 counter with synchronous reset



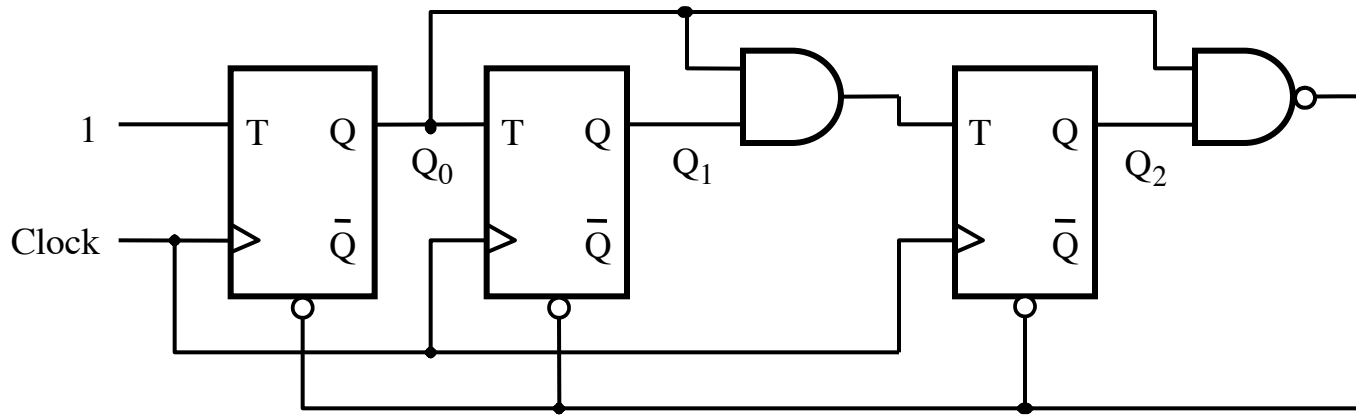
(a) Circuit



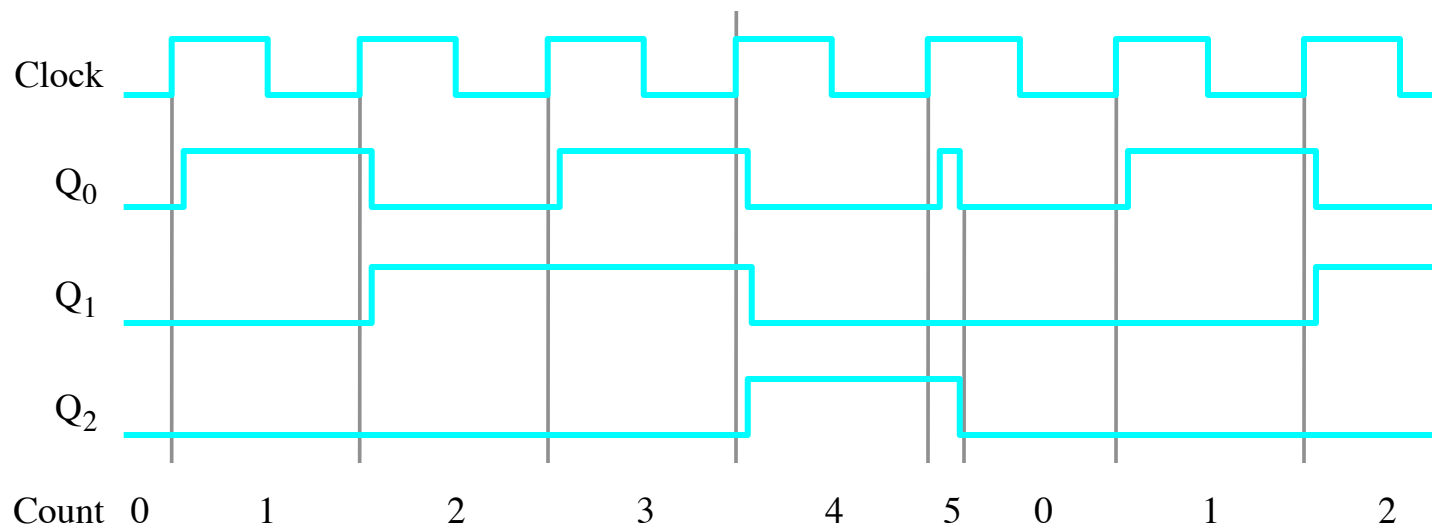
(b) Timing diagram

[Figure 5.25 from the textbook]

A modulo-6 counter with asynchronous reset



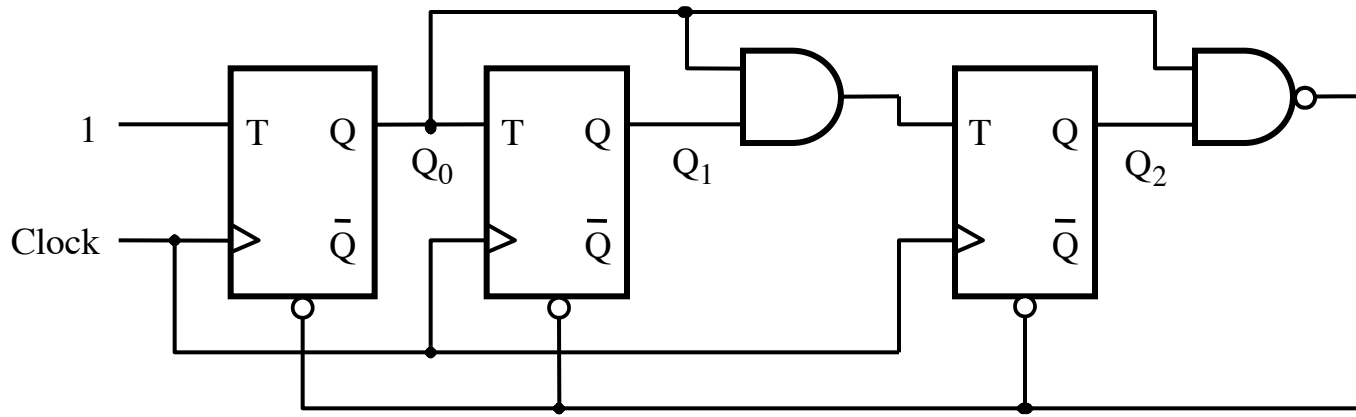
(a) Circuit



(b) Timing diagram

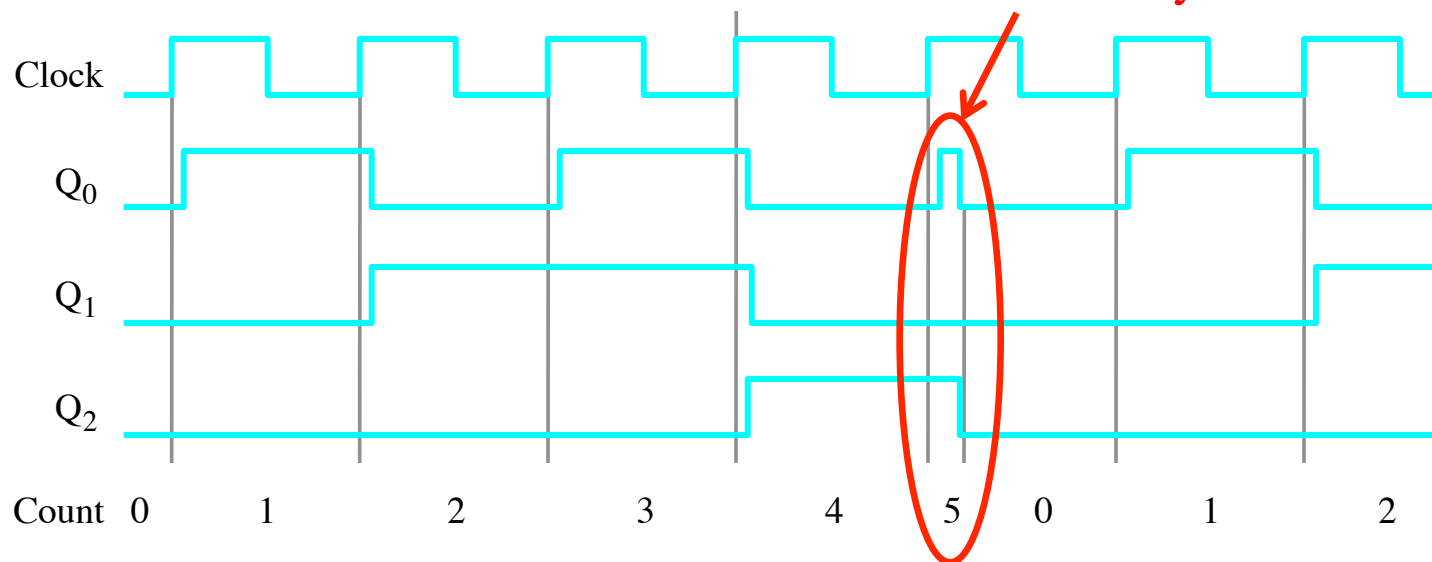
[Figure 5.26 from the textbook]

A modulo-6 counter with asynchronous reset



(a) Circuit

The number 5 is displayed for a very short amount of time



(b) Timing diagram

[Figure 5.26 from the textbook]

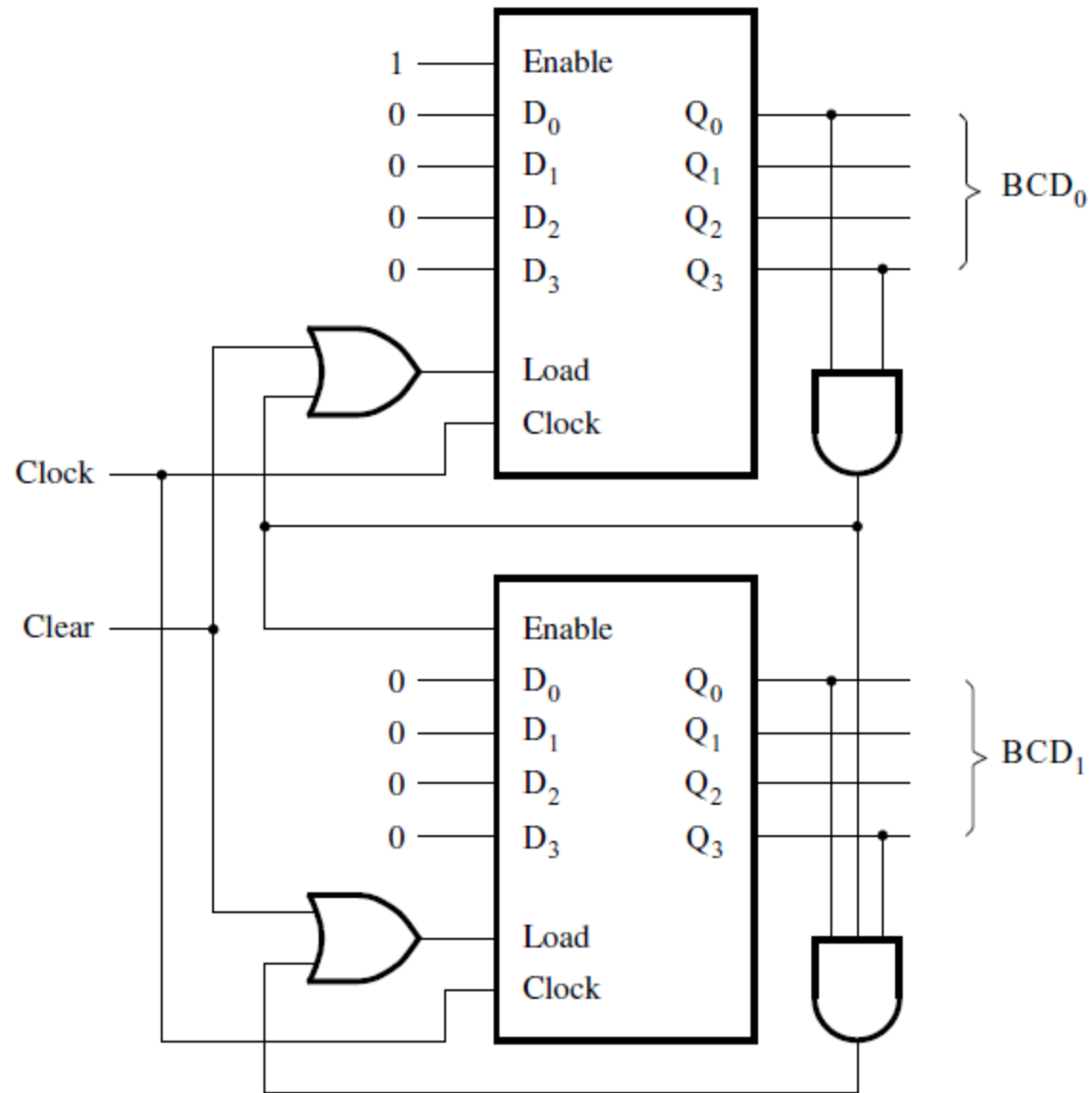
End of Quick Review

Other Types of Counters (Section 5.11)

A two-digit BCD counter

- **2: Parallel-load four-bit counter**
 - **Figure 5.24**
- **Each counts in binary**
 - **0-9**
- **Resets generated on 9**
 - **Reset by loading 0' s**
- **Second digit enabled by a 9 on first counter**

A two-digit BCD counter

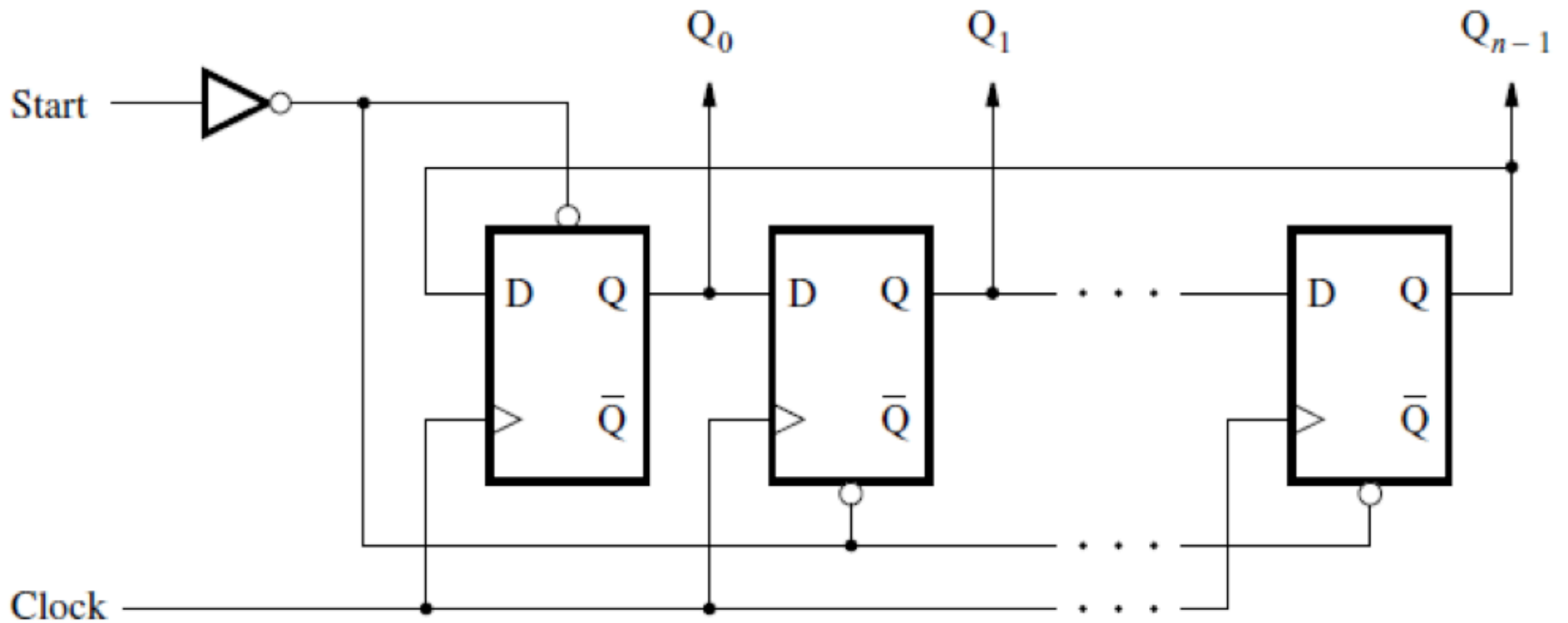


[Figure 5.27 from the textbook]

N-bit ring counter

- 1000, 0100, 0010, 0001, 1000.....
- Reset
 - Set start to 1
 - Sets output to 1000

N-bit ring counter

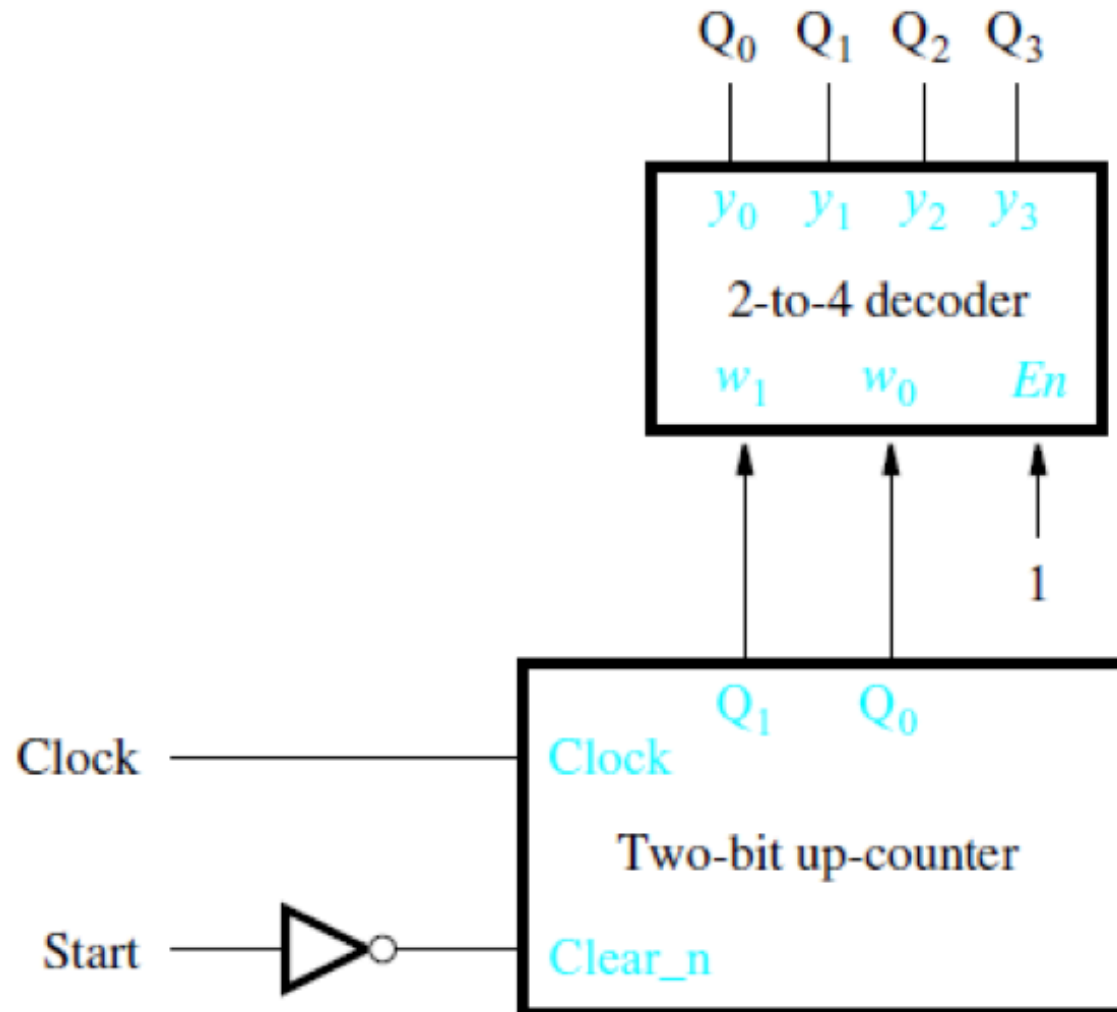


[Figure 5.28a from the textbook]

4-bit ring counter

- **Use a 2-bit counter**
 - 00, 01, 10, 11, 00.....
- **2-4 Decoder**
 - 1000, 0100, 0010, 0001, 1000.....

4-bit ring counter



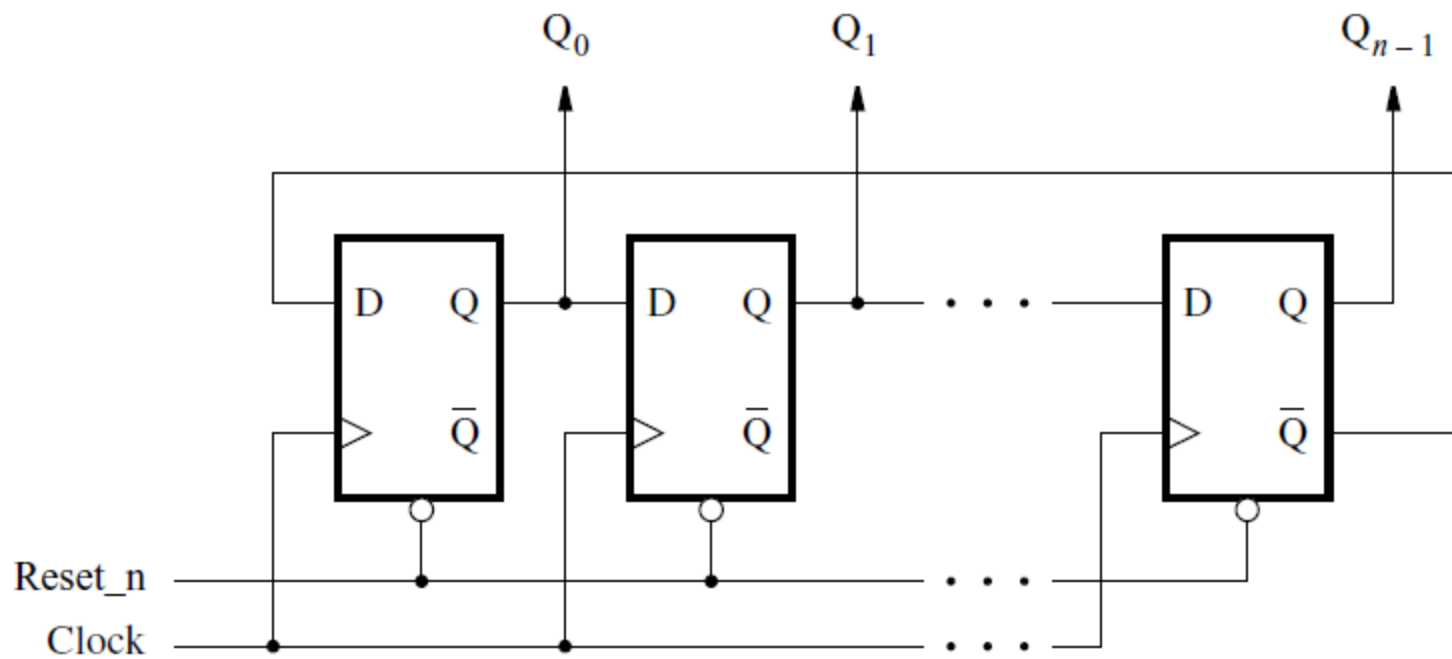
[Figure 5.28b from the textbook]

Johnson Counter

- **1-bit changes at a time**
- **0000, 1000, 1100, 1110, 1111, 0111, 0011, 0001, 0000**
- **Begin with a reset of all flip-flops**

- **An n-bit Johnson counter has a counting sequence of length $2n$**

Johnson counter



[Figure 5.29 from the textbook]

Reaction Timer Circuit (Section 5.14)

Problem Statement

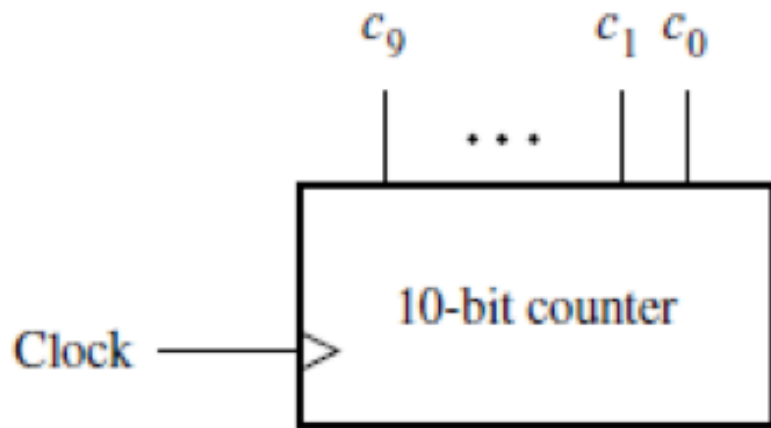
- **Want to design a reaction timer**
- **Circuit turns on light (LED)**
- **Person then presses switch**
- **Measures time from LED on until the switch is pressed**

Clock Divider

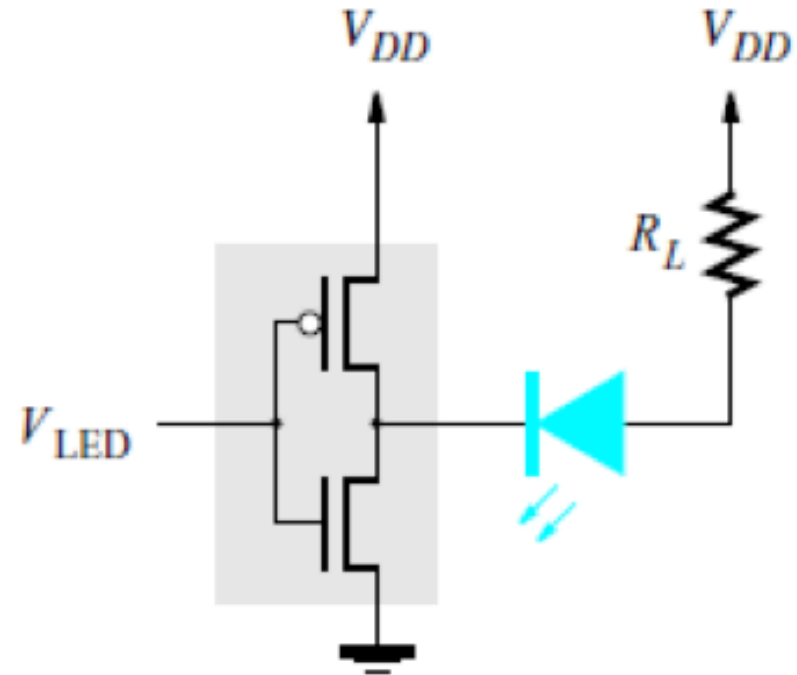
- **Input: 102.4kHz**
- **Output: 100Hz**
- **10-bit Counter to divide**

- **Output Frequency = $102.4k / 2^{10} = 100Hz$**

A reaction-timer circuit



(a) Clock divider

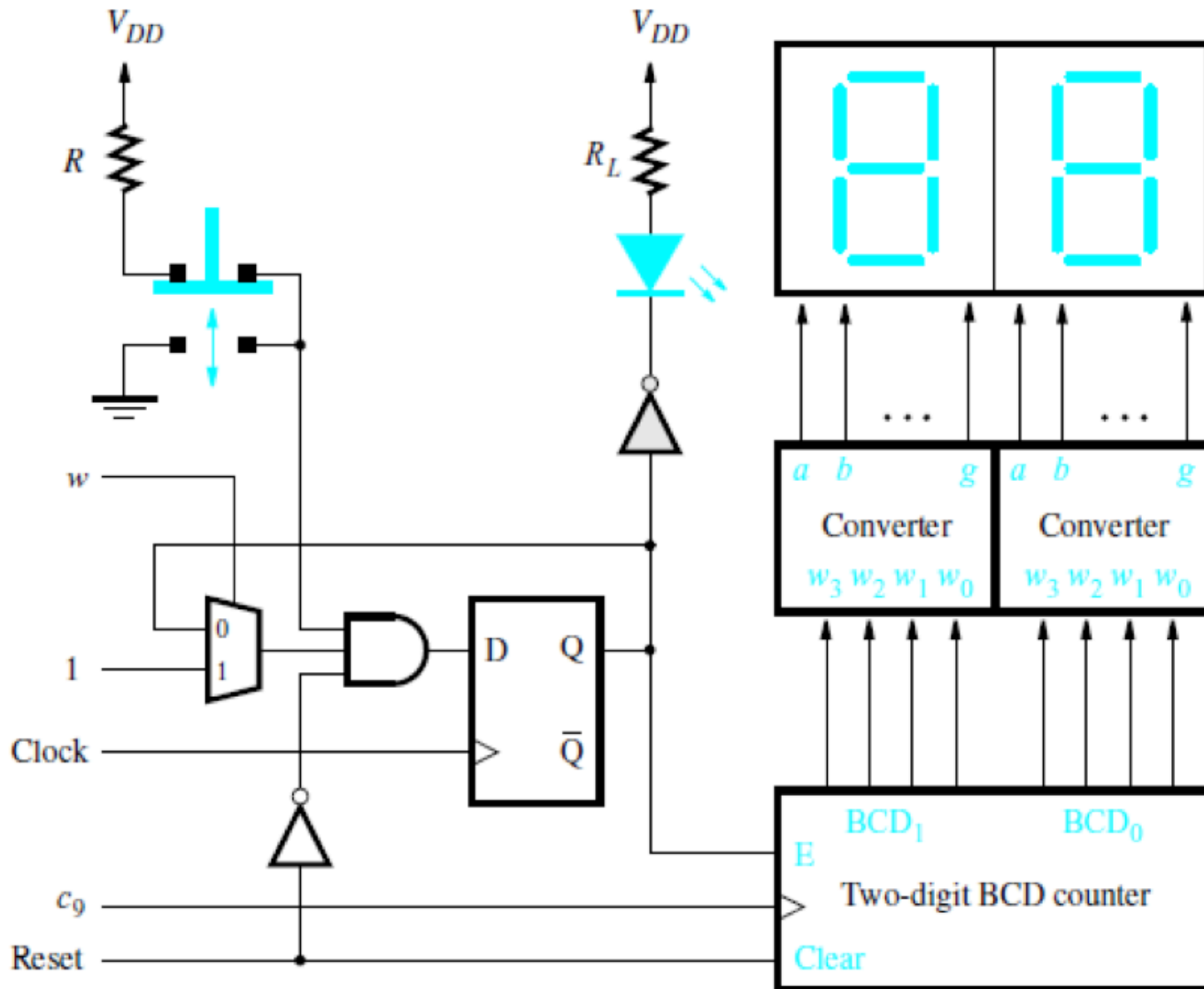


(b) LED circuit

Functionality of circuit

- **Push switch**
 - Nominally 1
- **DFF to keep track of the state**
- **Two-digit BCD counter**
 - Output goes to converters to a 7-segment display
- **Start-up**
 - **Assert the Reset signal**
 - Clears counter
 - Clears flip-flop
 - **Assert $w=1$ for one cycle**
 - **Once switch is hit**
 - Clears flip-flop
 - Stops counting

Push-button switch, LED, and 7-segment displays



[Figure 5.61c from the textbook]

Timing Analysis of Flip-Flop Circuits (Section 5.15)

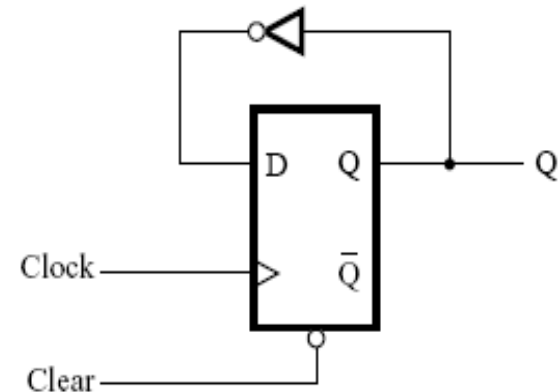
Timing Review

- **tsu: setup time**
- **th: hold time**
- **tcQ: propogation delay**

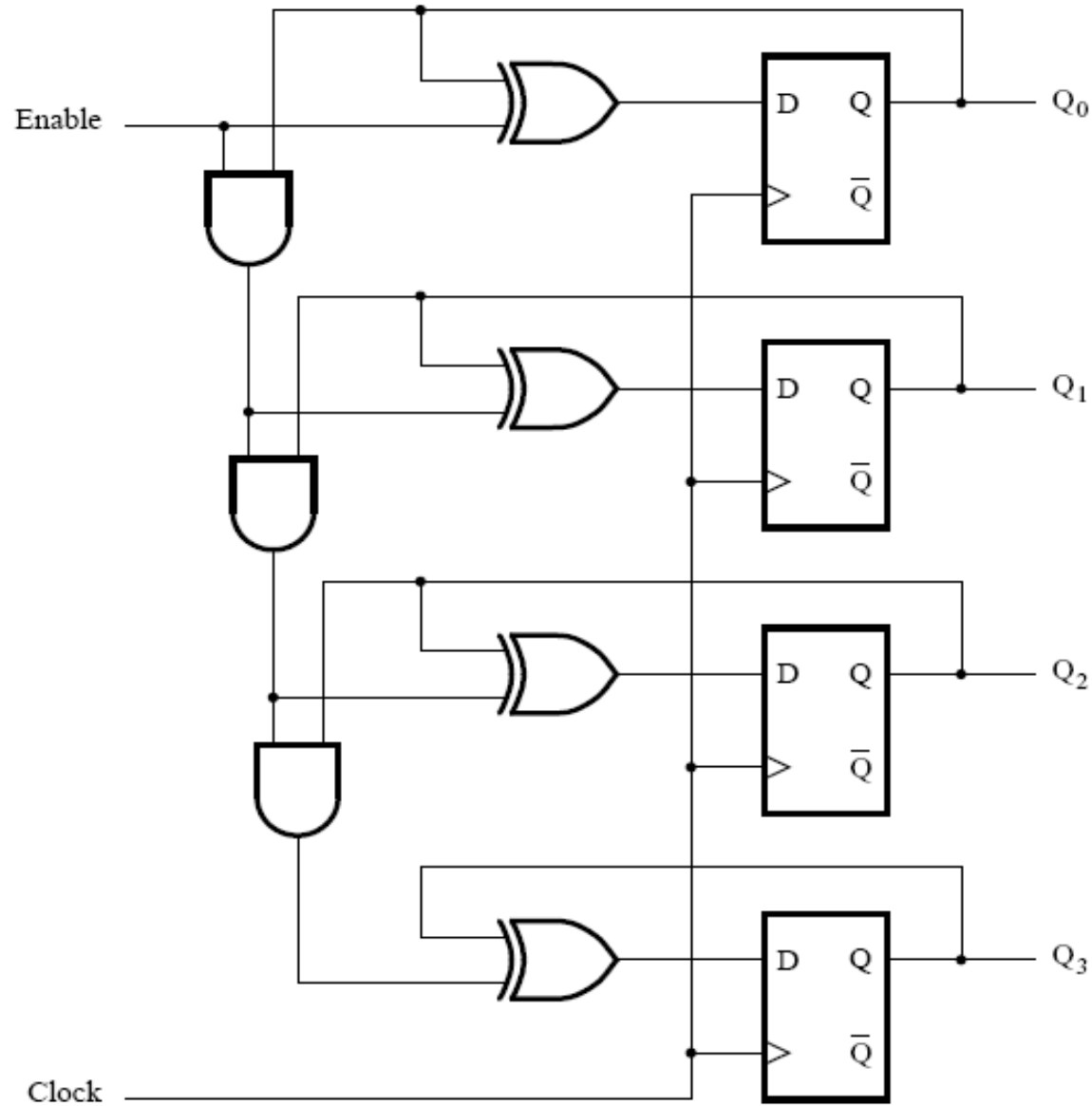
Timing Example

- **tsu: 0.6ns**
- **th: 0.4ns**
- **tcQ: 0.8ns to 1.0ns**
 - Which value to use?
- **Logic gate delay: $1+0.1k$**
 - k is equal to the number of inputs
- **$T_{min} = t_{su} + t_{cQ} + t_{not} = 0.6 + 1.0 + 1.1 = 2.7ns$**
- **$F_{max} = 1/T_{min} = 370.37MHz$**

- **Check for hold violations**
 - Fastest Q can change = $t_{cQ} + t_{not} = 0.8 + 1.1 = 1.9ns$
 - $1.9ns > 0.4ns$ therefore no hold violations



Timing Example: 4-bit counter



[Figure 5.67 from the textbook]

Timing Example: 4-bit counter

- Look for longest path
 - Q0 to Q3
- Propagation delay of Q0
- 3 AND propagation delays
- 1 XOR propagation delay
- Setup delay for Q3

- $T_{min} = 1.0 + 3(1.2) + 1.2 + 0.6 = 6.4ns$
- $F_{max} = 1/6.4ns = 156.25MHz$

- Check for hold violations
 - Fastest Q can change = $t_{cQ} + t_{XOR} = 0.8 + 1.2 = 2ns$
 - $2.0ns > 0.4ns$ therefore no hold violations

Timing Example: Clock Skew

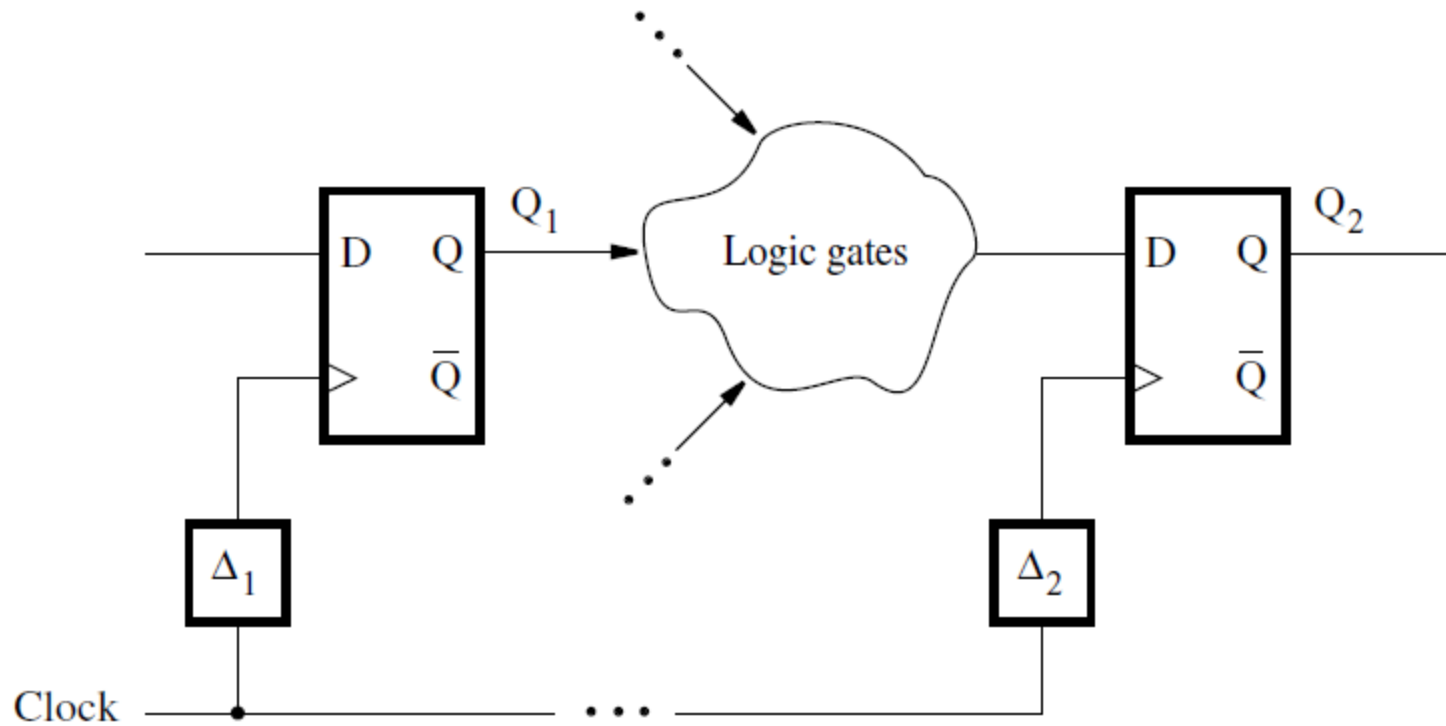


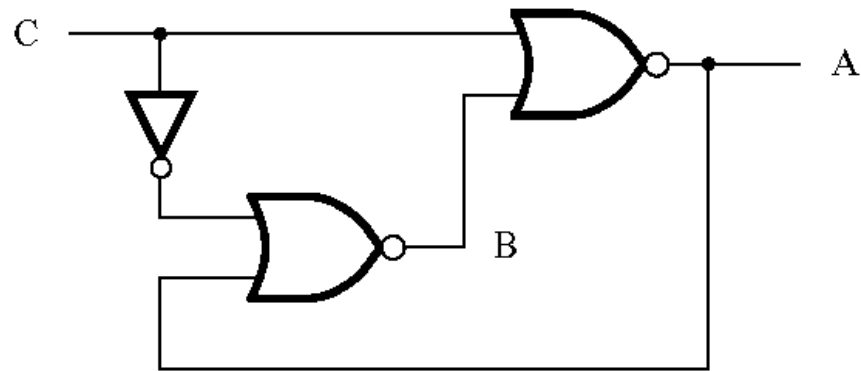
Figure 5.68. A general example of clock skew.

Skew Timing Example: 4-bit counter

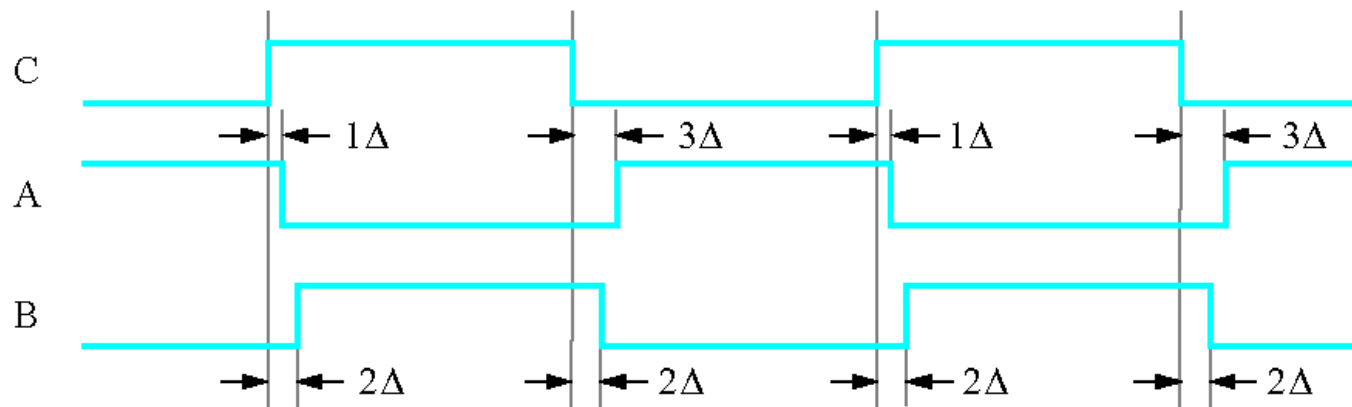
- **Q3 now has a clock slew delay: 1.5ns**
 - $T = 1.0 + 3(1.2) + 1.2 + 0.6 - 1.5 = 4.9\text{ns}$
- **Now might not be the longest path**
- **Check Q0 to Q2**
 - $T = 1.0 + 2(1.2) + 1.2 + 0.6 = 5.2\text{ns}$
- **$F_{\text{max}} = 1/5.2\text{ns} = 192.31\text{MHz}$**

Examples of Solved Problems (Section 5.17)

Example 5.18



(a) Circuit



(b) Timing diagram

Figure 5.70. Circuit for Example 5.18.

Example 5.19

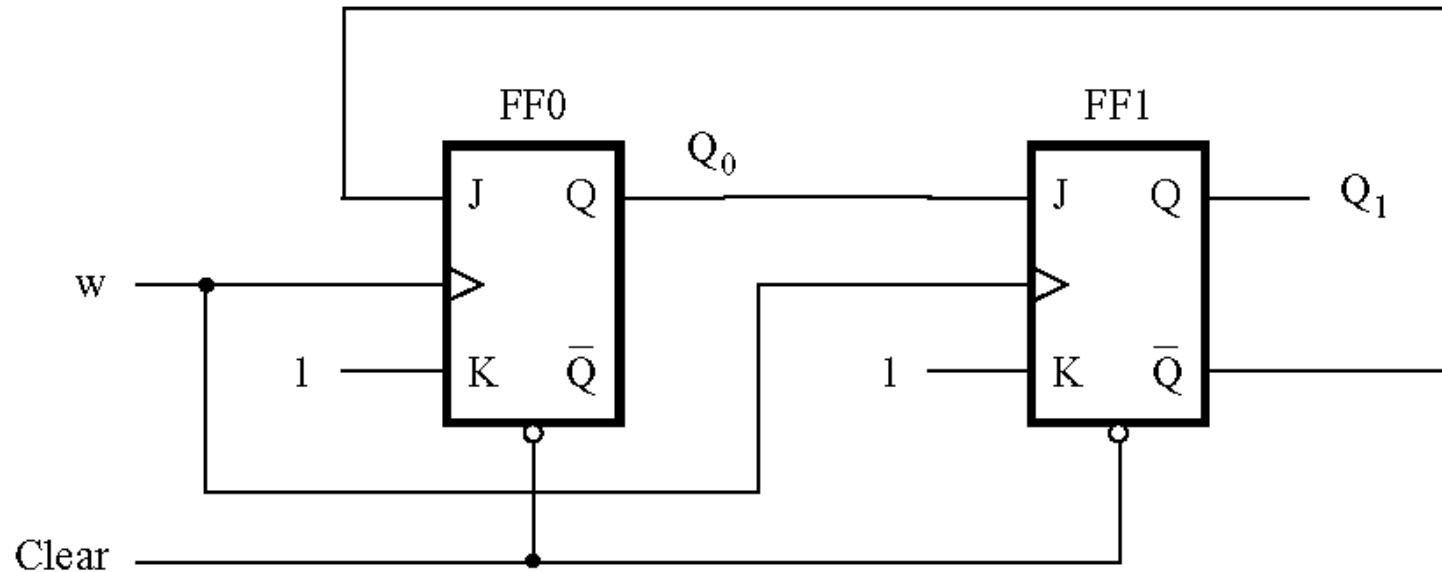


Figure 5.71. Circuit for Example 5.19.

J	K	Q(t+1)	
0	0	Q(t)	No Change
0	1	0	Reset
1	0	1	Set
1	1	Q'(t)	Complement

Time interval	FF0			FF1		
	J_0	K_0	Q_0	J_1	K_1	Q_1
Clear	1	1	0	0	1	0
t_1	1	1	1	1	1	0
t_2	0	1	0	0	1	1
t_3	1	1	0	0	1	0
t_4	1	1	1	1	1	0

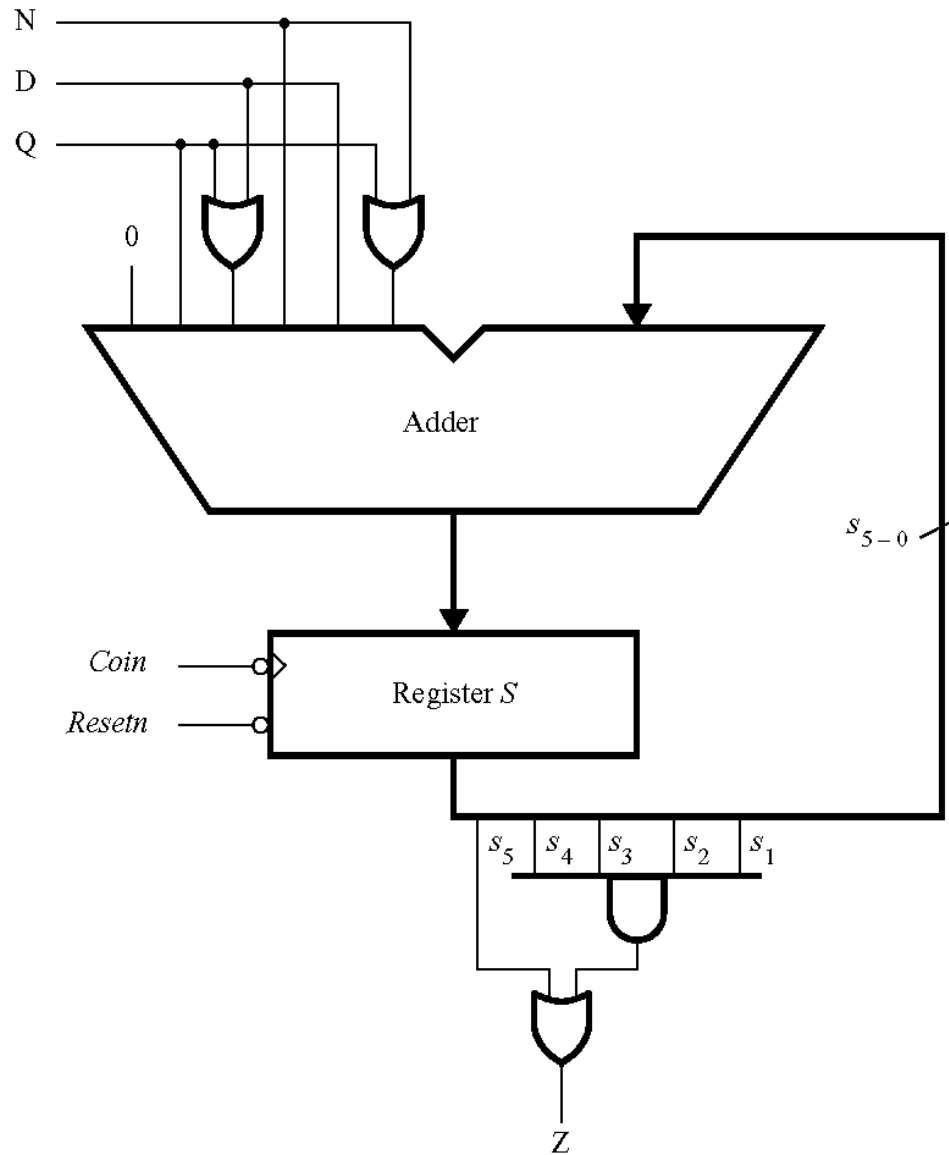
Figure 5.72. Summary of the behavior of the circuit in Figure 5.71.

Example 5.20

Vending machine example

- **Inputs N, D, Q, Coin, Resetn**
 - N, D, Q: nickel, dime, quarter
 - Coin: pulsed when a coin is entered
 - Used to store values into register
 - Resetn: resets the register value to zero
- **Add up new coin with old value**
 - Store new sum into old value register
- **See if total is above thirty cents**
 - If so output Z goes high

Circuit for Example 5.20



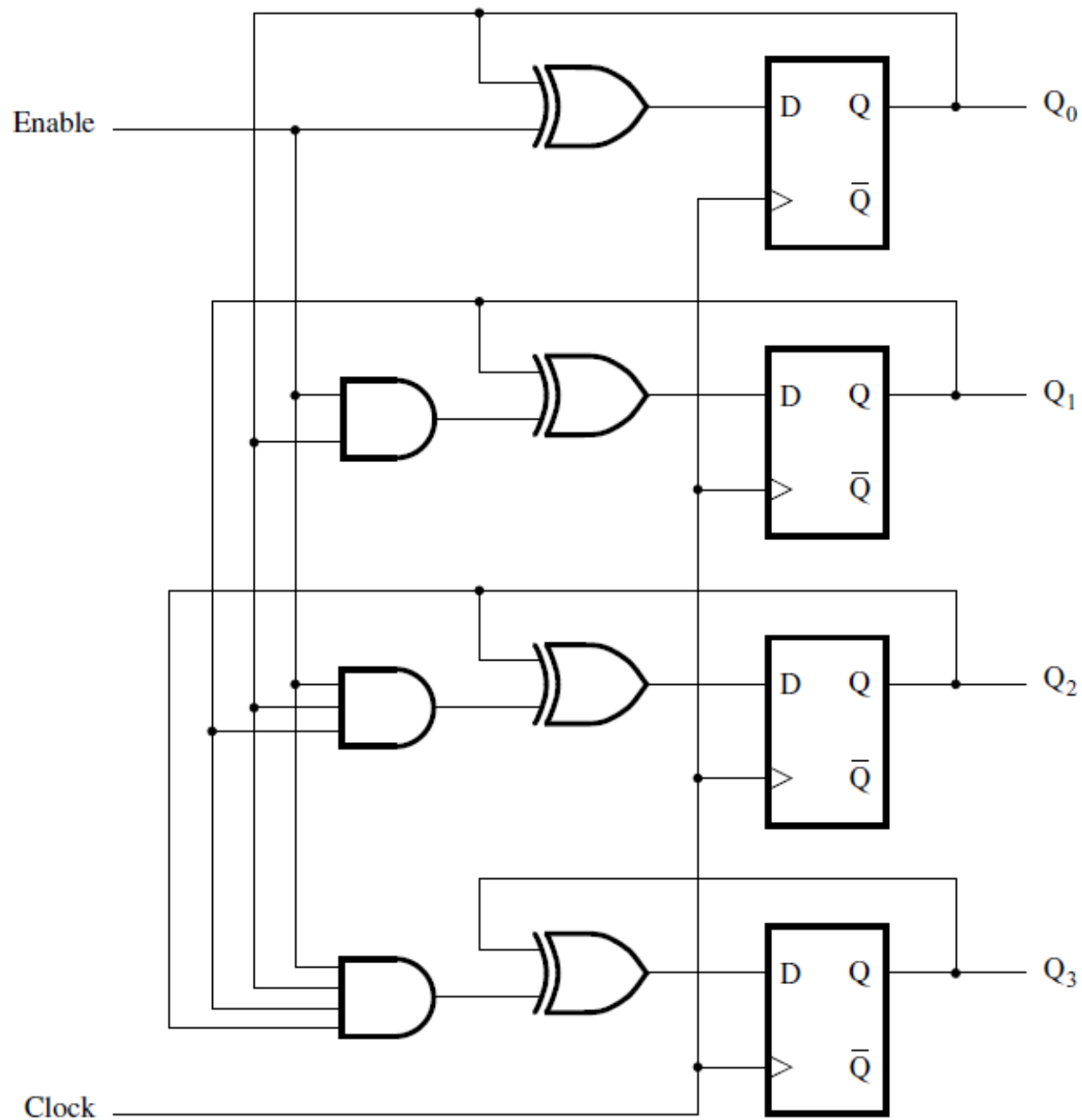
[Figure 5.73 from the textbook]

Example 5.22

Faster 4-bit Counter

- **Want to increase the speed of the 4-bit counter**
- **Use similar method as used in 4-bit adder**
- **Remove series AND gates**

A faster 4-bit counter



[Figure 5.75 from the textbook]

Faster 4-bit Counter

- Longest path: Q0 to Q3
- $T_{min} = t_{cQ0} + t_{AND} + t_{XOR} + t_{su}$
- $= 1.0 + 1.4 + 1.2 + 0.6 = 4.2ns$
- $F_{max} = 1/4.2ns = 238.1MHz > 156.25MHz$

Questions?

THE END