DIGITAL LOGIC

Chapter 5:
Flip-Flops, Registers, and Counters II

Ru Han
Fall

1

OUTLINE

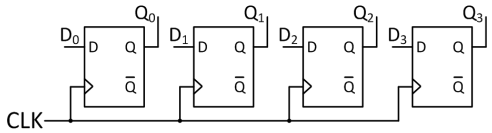
Flip-Flops, Registers, and Counters

- Logic circuits that can store information
- Flip-flops, which store a single bit
- Registers, which store multiple bits
- Shift registers, which shift the contents of the register
- Counters of various types
- Verilog constructs used to implement storage elements

2

5.8 REGISTERS

- The basic register is an n-bit memory device constructed using n flip-flops, all sharing a common clock.
- Example: $n = 4$

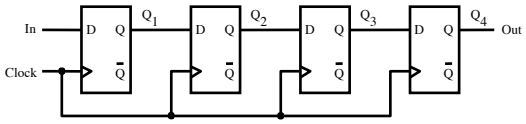


- Data are loaded in parallel.
- All n bits may be loaded in 1 clock cycle.

3

SHIFT REGISTER

- A shift register is made from D flip-flops where each Q output feeds the next D input.
- Used in various computing and signal processing applications.

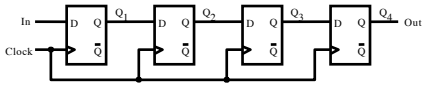


(a) Circuit

- Data are loaded serially.
- Require n clock cycles to replace all n bits.

4

SHIFT REGISTER



(a) Circuit

	In	Q_1	Q_2	Q_3	Q_4 = Out
t_0	1	0	0	0	0
t_1	0	1	0	0	0
t_2	1	0	1	0	0
t_3	1	1	0	1	0
t_4	1	1	1	0	1
t_5	0	1	1	1	0
t_6	0	0	1	1	1
t_7	0	0	0	1	1

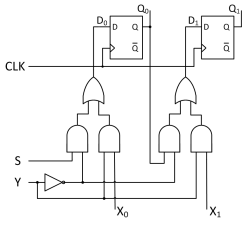
(b) A sample sequence

Figure 5.17. A simple shift register.

5

PARALLEL-ACCESS SHIFT REGISTER

- We can construct a register that can be loaded in serial or parallel fashion by connecting a logic network to the D input of each flip-flop.



Inputs & Outputs

- Inputs:
  - **CLK**: common clock
  - **S**: Serial data input
  - **X_0, X_1, …, X_{n-1}**: Parallel data inputs
  - **Y**: Shift/Load
- Outputs:
  - **Q0, Q1, … Qn-1. Data outputs**

6

PARALLEL-ACCESS SHIFT REGISTER

Parallel Access Shift Register

- Boolean Equations

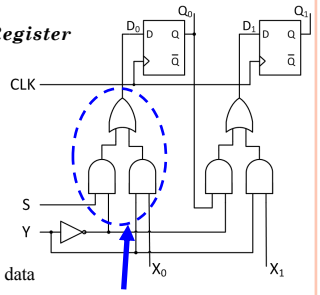
$D_0 = \overline{Y} \cdot S + Y \cdot X_0$

$D_1 = \overline{Y} \cdot Q_0 + Y \cdot X_1$

…

$D_{n-1} = \overline{Y} \cdot Q_{n-2} + Y \cdot X_{n-1}$

- Diagram ($n = 2$)



MUX logic

Logic:

If Y = 0, each D input should get data serially.

$D_0 = S$, $D_1 = Q_0$, $D_2 = Q_1$, …, $D_{n-1} = Q_{n-2}$

If Y = 1, load D's in parallel

$D_0 = X_0$, $D_1 = X_1$, …, $D_{n-1} = X_{n-1}$.

PARALLEL-ACCESS SHIFT REGISTER

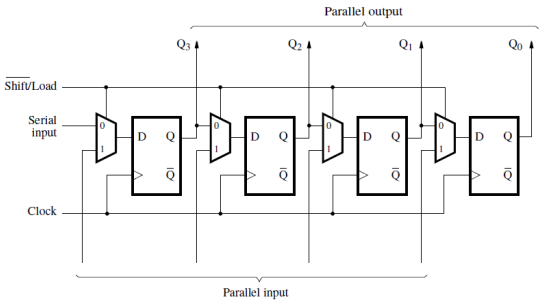


Figure 5.18. Parallel-access shift register.

5.9 COUNTERS

- A counter is a device that produces as output a specified sequence of numbers.
- Counters can be implemented using the adder/subtractor circuits and the registers.
- We will show how the counter circuits can be designed using T and D flip-flops.

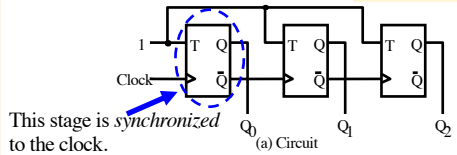
ASYNCHRONOUS COUNTERS

- The simplest counter circuits can be built using T flip-flops because the toggle feature is naturally suited for the implementation of the counting operation.

ASYNCHRONOUS COUNTERS

Asynchronous counter
Ripple counter

- **Up-Counter with T Flip-Flops**



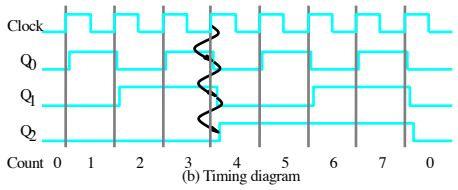
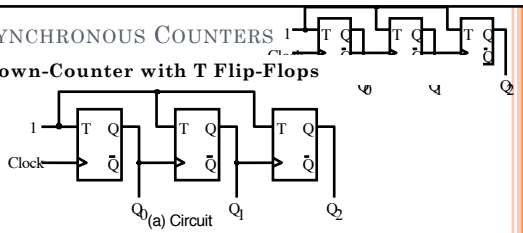


Figure 5.19. A three-bit up-counter.

ASYNCHRONOUS COUNTERS

- **Down-Counter with T Flip-Flops**



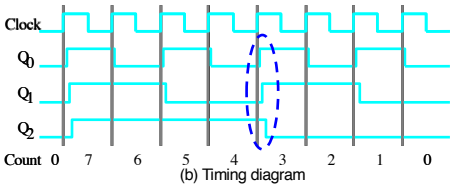


Figure 5.20. A three-bit down-counter.

SYNCHRONOUS COUNTERS

- **Synchronous Counter with T Flip-Flops**

Clock cycle	Q_2 Q_1 Q_0	
0	0 0 0	
1	0 0 1	
2	0 1 0	Q_1 changes
3	0 1 1	
4	1 0 0	Q_2 changes
5	1 0 1	
6	1 1 0	
7	1 1 1	
8	0 0 0	

$T_0 = 1$

$T_1 = Q_0$

$T_2 = Q_0Q_1$

$T_3 = Q_0Q_1Q_2$

. . . . . .

$T_n = Q_0Q_1 \cdots Q_{n-1}$

Table 5.1. Derivation of the synchronous up-counter.

Synchronous Counters

- **Synchronous Counter with T Flip-Flops**

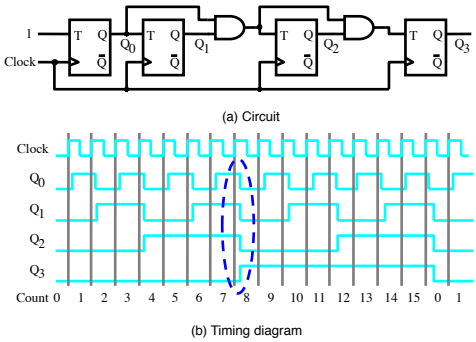


Figure 5.21. A four-bit synchronous up-counter.

Synchronous Counters

- **Enable and Clear Capability**

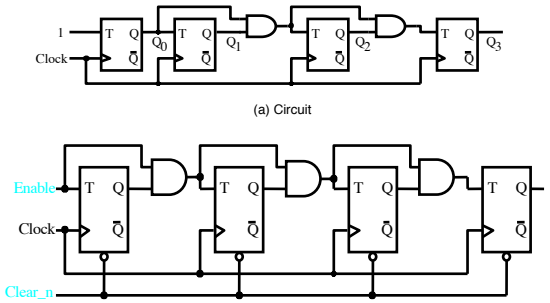
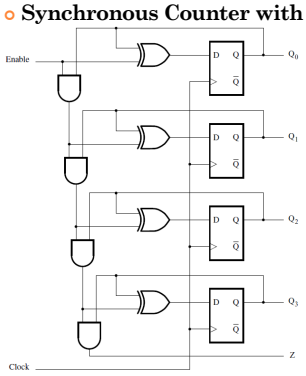


Figure 5.22. Inclusion of Enable and Clear capability.

Synchronous Counters

- **Synchronous Counter with D Flip-Flops**



If *Enable* = 1, the *D* inputs of the flip-flops are defined by the expressions

$$D_0 = Q_0 \oplus 1 = \overline{Q}_0$$
$$D_1 = Q_1 \oplus Q_0$$
$$D_2 = Q_2 \oplus Q_1Q_0$$
$$D3 = Q_3 \oplus Q_2Q_1Q_0$$

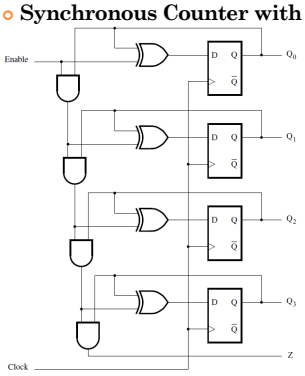
For a larger counter the *i*th stage is defined by

$$D_i = Q_i \oplus Q_{i-1}Q_{i-2}\cdots Q_1Q_0$$

Figure 5.23. A four-bit counter with D flip-flops.

Synchronous Counters

- **Synchronous Counter with D Flip-Flops**



If *Enable* = 1, the *D* inputs of the flip-flops are defined by the expressions

$$D_0 = Q_0 \oplus 1 = \overline{Q}_0$$
$$D_1 = Q_1 \oplus Q_0$$
$$D_2 = Q_2 \oplus Q_1Q_0$$
$$D3 = Q_3 \oplus Q_2Q_1Q_0$$

$$D_0 = Q_0 \oplus Enable$$
$$D_1 = Q_1 \oplus Q_0 \cdot Enable$$
$$D_2 = Q_2 \oplus Q_1 \cdot Q_0 \cdot Enable$$
$$D_3 = Q_3 \oplus Q_2 \cdot Q_1 \cdot Q_0 \cdot Enable$$

Figure 5.23. A four-bit counter with D flip-flops.

Synchronous Counters

- **Synchronous Counter with D Flip-Flops**

A T flip-flop can be formed from a D flip-flop by providing the extra gating that gives

$$D = Q\overline{T} + \overline{Q}T$$
$$= Q \oplus T$$

Thus in each stage in Figure 5.23, the D flip-flop and the associated XOR gate implement the functionality of a T flip-flop.

Counters with Parallel Load

Sometimes it is desirable to start with a different count.

To allow this mode of operation, a counter circuit must have some inputs through which the initial count can be loaded.

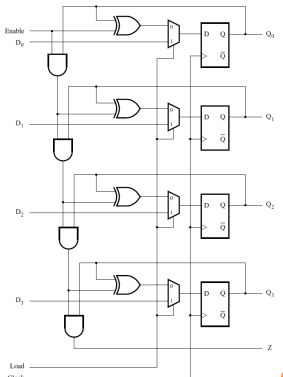
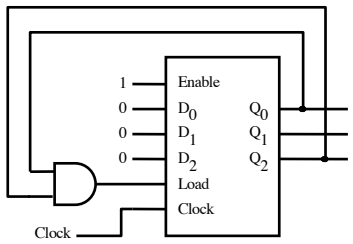


Figure 5.24. A counter with parallel-load capability.

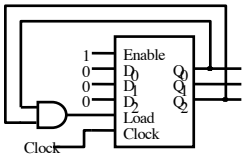
5.10 RESET SYNCHRONIZATION

- We may want to design a modulo-6 counter, for which the counting sequence is 0, 1, 2, 3, 4, 5, 0, 1, and so on.



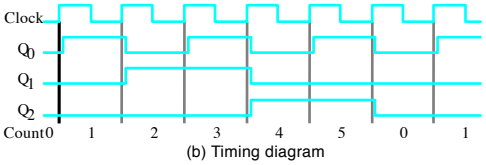
(a) Circuit

RESET SYNCHRONIZATION



(a) Circuit

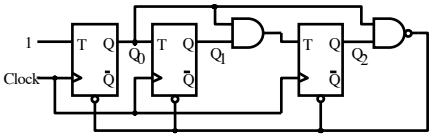
Because the counter is reset on the active edge of the clock, we call that this type of counter as a ***synchronous reset counter***.



(b) Timing diagram

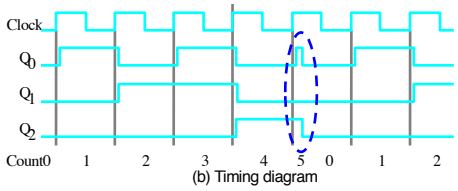
Figure 5.25. A modulo-6 counter with synchronous reset.

RESET SYNCHRONIZATION



(a) Circuit

The approach employed in Figure 5.26*a* is said to use ***asynchronous reset***.



(b) Timing diagram

Figure 5.26. A modulo-6 counter with asynchronous reset.

5.11 OTHER TYPES OF COUNTERS

- **BCD Counter**

It consists of two modulo-10 counters.

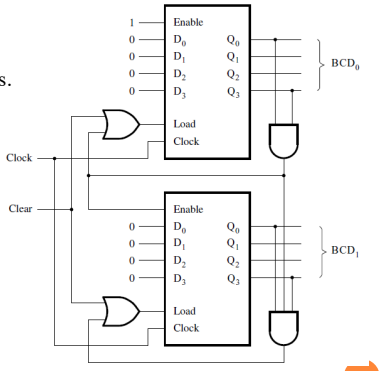
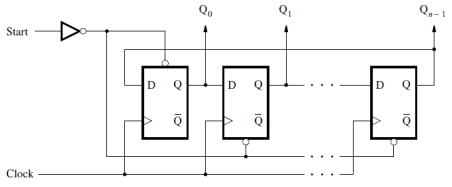


Figure 5.27. A two-digit BCD counter.

OTHER TYPES OF COUNTERS

- **Ring Counter**

It is possible to devise a counterlike circuit in which each flip-flop reaches the state $Q_i = 1$ for exactly one count, while for all other counts $Q_i = 0$. Then Q_i indicates directly an occurrence of the corresponding count.



(a) An *n*-bit ring counter

Fig. 5.28. Ring counter.

OTHER TYPES OF COUNTERS

- **Ring Counter**

Fig. 5.28(*b*) shows how a ring counter can be constructed using a two-bit up-counter and a decoder.

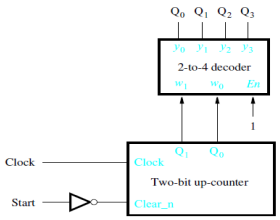


Fig. 5.28 (b). 4-bit Ring counter.

Other Types of Counters

- **Johnson Counter**

An interesting variation of the ring counter is obtained if, instead of the Q output, we take the $\overline{Q}$ output of the last stage and feed it back to the first stage, as shown in Figure 5.29.

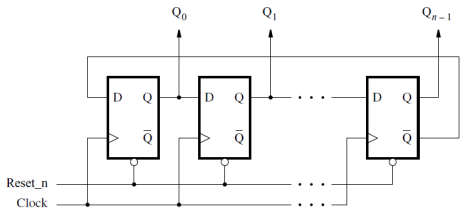


Figure 5.29. Johnson counter.

25

CONCLUSION

- Register
- Shift register
- Counter

- Assignment: 5.4, 5.15, 5.17

26