
Content

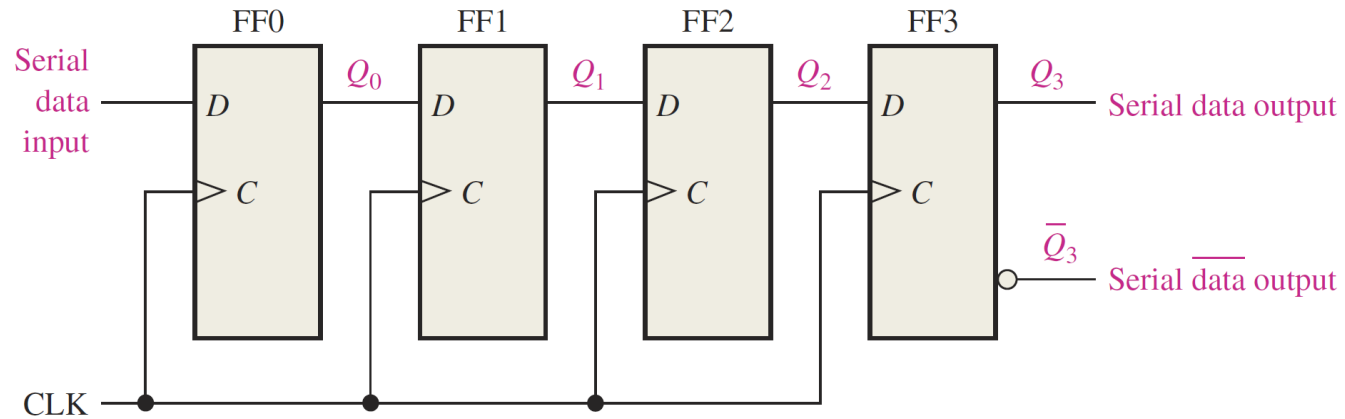
Sequential Logic

Analyze the Sequential Logic

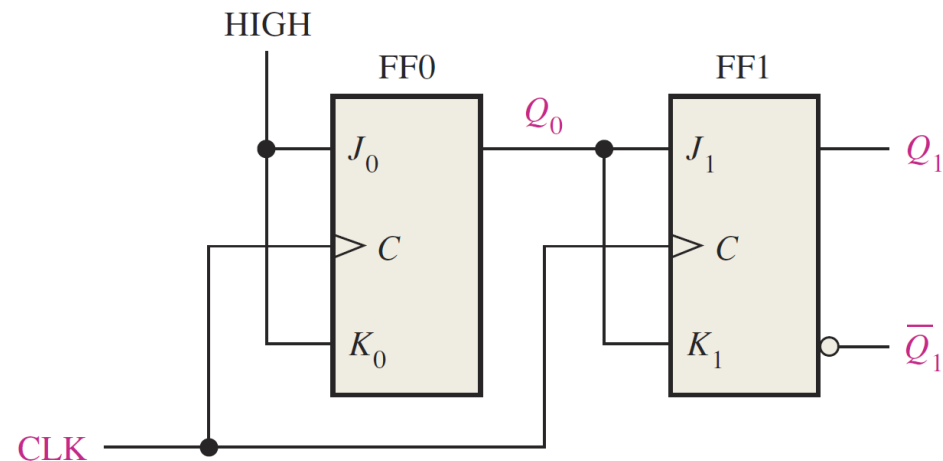
Sequential Logic

- The output depends not only on the present input but also on the history of the input

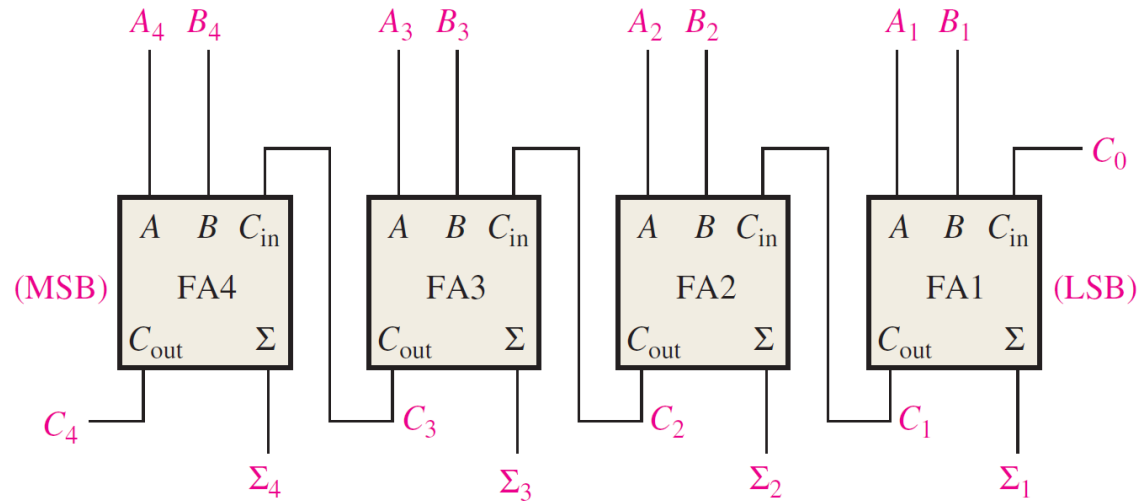
Shift register



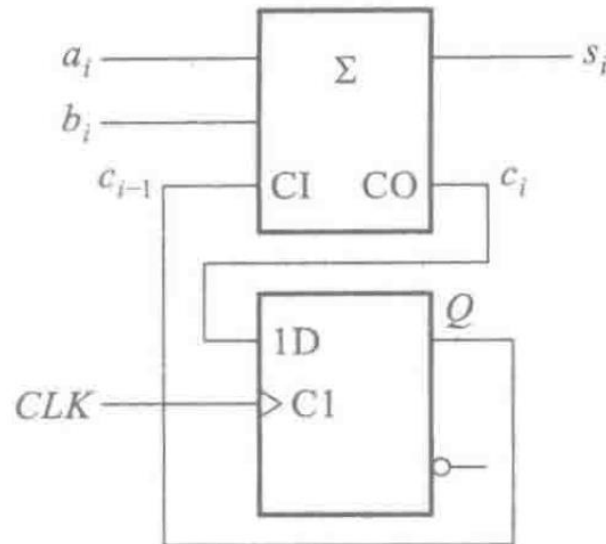
Counter



Ripple Carry Adder

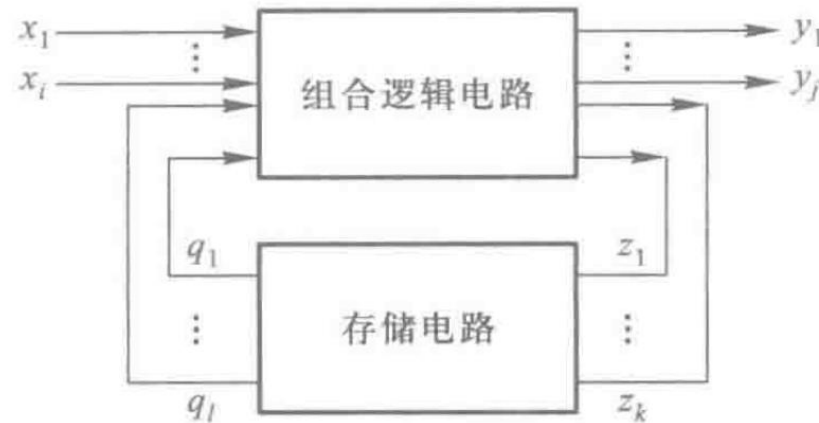


- Use sequential logic to construct the ripple carry adder



Sequential Logic

- The sequential logic consists of the storage unit and combinational logic



$$\begin{cases} y_1 = f_1(x_1, x_2, \dots, x_i, q_1, q_2, \dots, q_l) \\ y_2 = f_2(x_1, x_2, \dots, x_i, q_1, q_2, \dots, q_l) \\ \vdots \\ y_j = f_j(x_1, x_2, \dots, x_i, q_1, q_2, \dots, q_l) \end{cases}$$

$$Y = F[X, Q]$$

输出方程

$$\begin{cases} z_1 = g_1(x_1, x_2, \dots, x_i, q_1, q_2, \dots, q_l) \\ z_2 = g_2(x_1, x_2, \dots, x_i, q_1, q_2, \dots, q_l) \\ \vdots \\ z_k = g_k(x_1, x_2, \dots, x_i, q_1, q_2, \dots, q_l) \end{cases}$$

$$Z = G[X, Q]$$

驱动方程

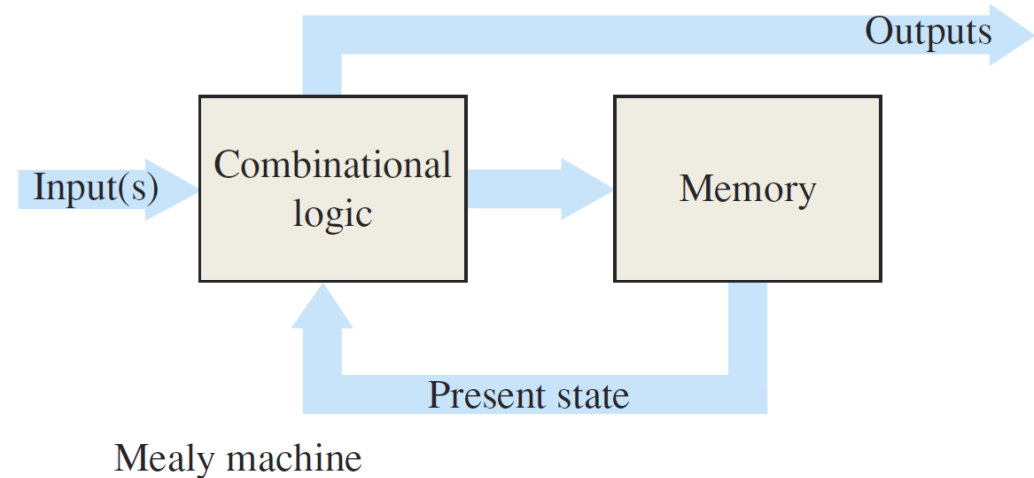
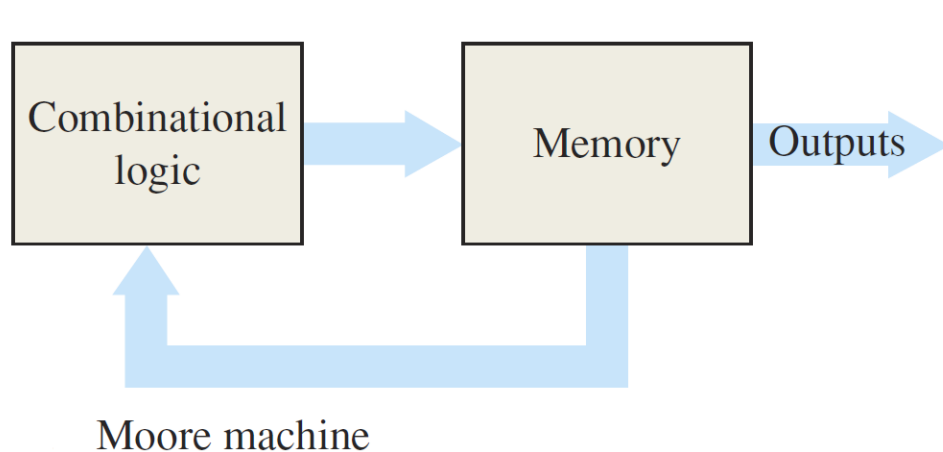
$$\begin{cases} q_1^* = h_1(z_1, z_2, \dots, z_k, q_1, q_2, \dots, q_l) \\ q_2^* = h_2(z_1, z_2, \dots, z_k, q_1, q_2, \dots, q_l) \\ \vdots \\ q_l^* = h_l(z_1, z_2, \dots, z_k, q_1, q_2, \dots, q_l) \end{cases}$$

$$Q^* = H[Z, Q]$$

状态方程

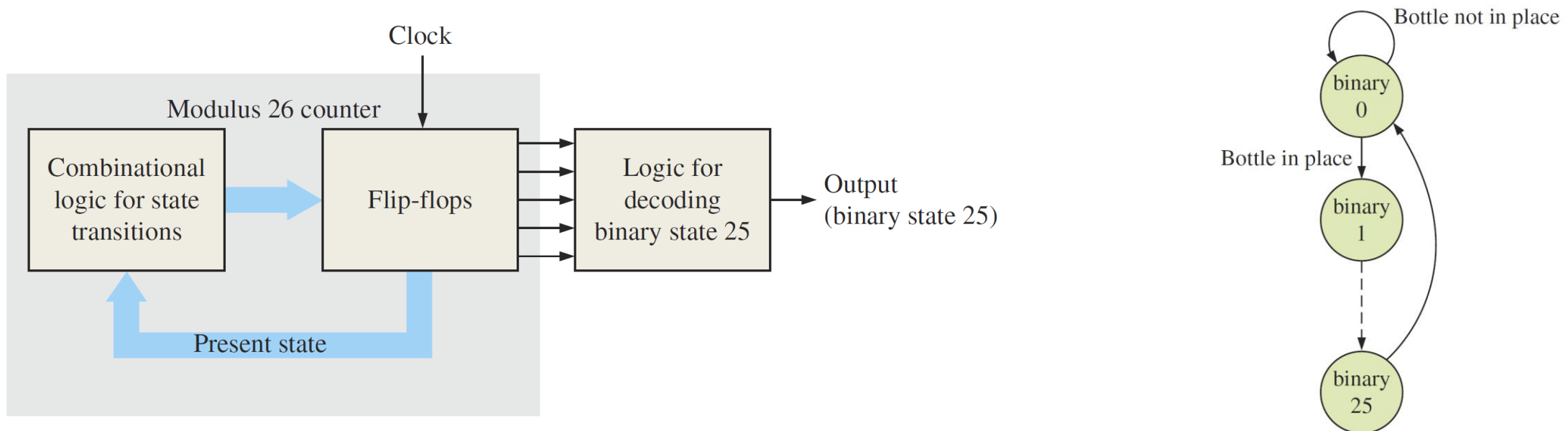
Sequential Logic

- The sequential logic can be classified as synchronous and asynchronous
- It can also be classified as Moore and Mealy types. In Moore type, the outputs depend only on the present state of the memory and the input does not affect the outputs. In Mealy type, the outputs depend on both the internal present state and the inputs.



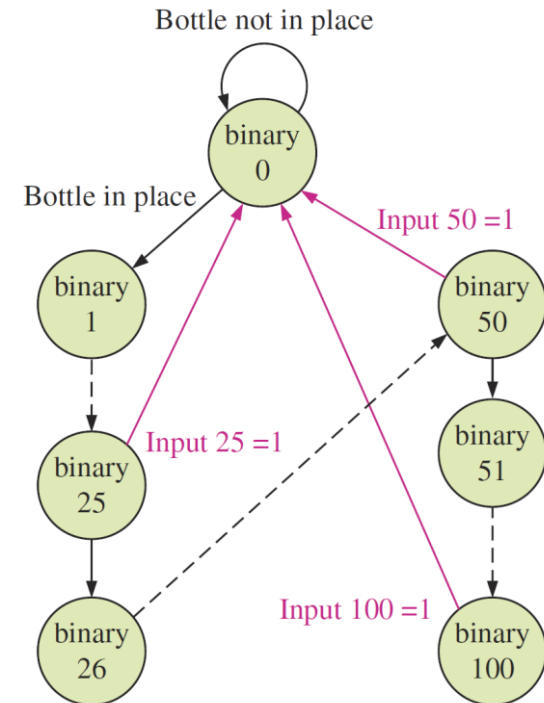
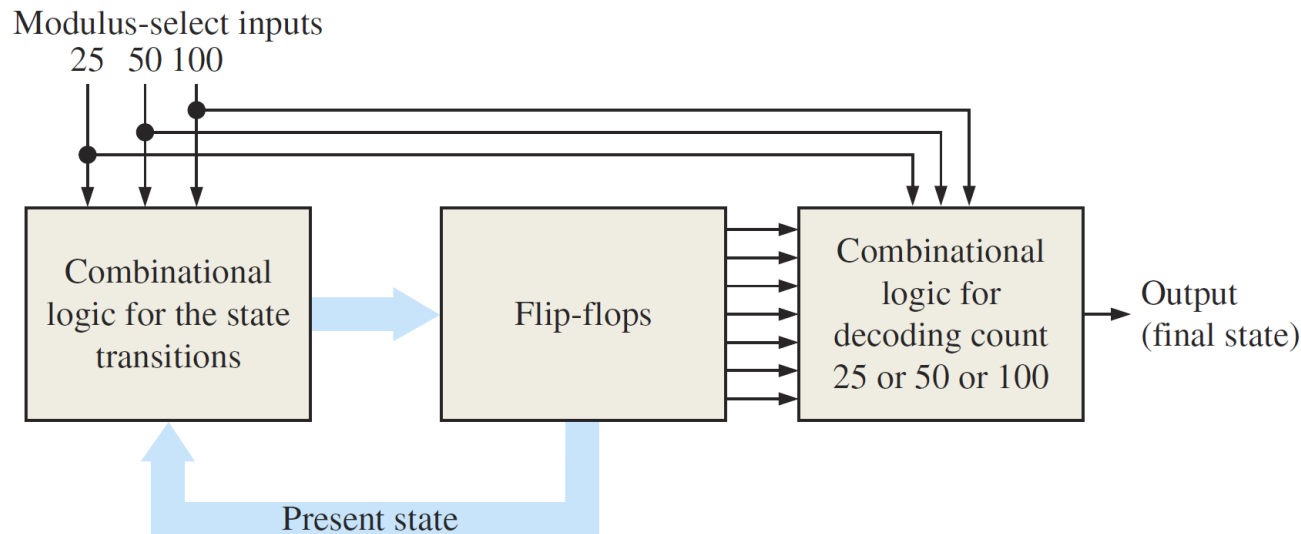
Moore Machine Example

- Once a bottle is in place, the first tablet is inserted at binary state 1, the second at binary state 2, and the twenty-fifth tablet when the binary state is 25. Count 25 is decoded and used to stop the flow of tablets and the clock. The counter stays in the 0 state until the next bottle is in position



Mealy Moore Machine Example

- Assume a tablet-bottling system uses three bottles sizes: a 25, 50 and 100 tablet bottle
- The combinational logic sets the modulus of the counter depending on the modulus-select inputs. The output of depends on both the present state and the modulus-select inputs, making this a Mealy machine.

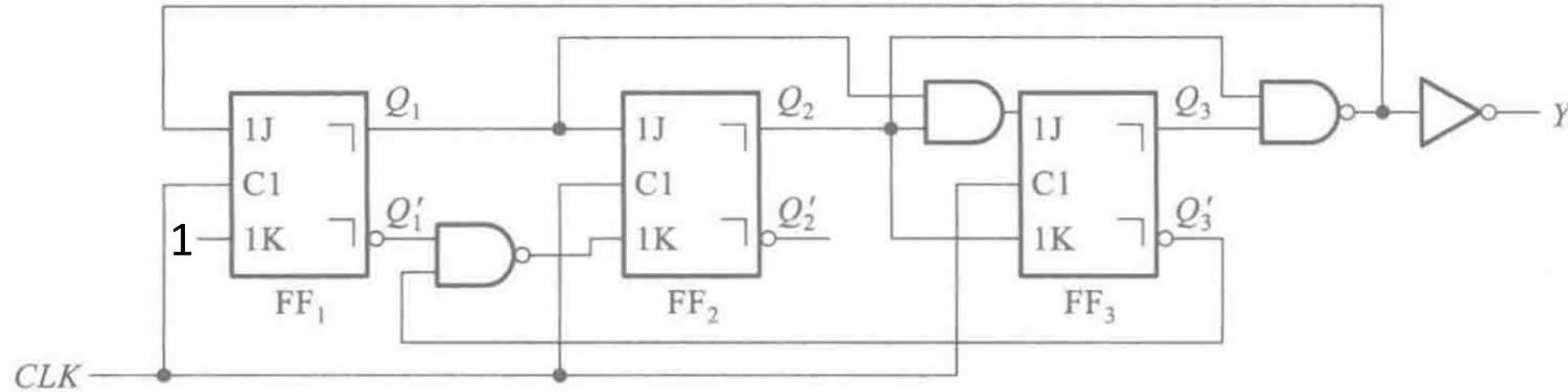


Content

Sequential Logic

Analyze the Sequential Logic

Sequential Logic example



驱动方程



$$\begin{cases} J_1 = (Q_2 \cdot Q_3)' & K_1 = 1 \\ J_2 = Q_1 & K_2 = (Q_1' \cdot Q_3)' \\ J_3 = Q_1 \cdot Q_2 & K_3 = Q_2 \end{cases}$$

状态方程

$$Q^* = J Q' + K' Q$$

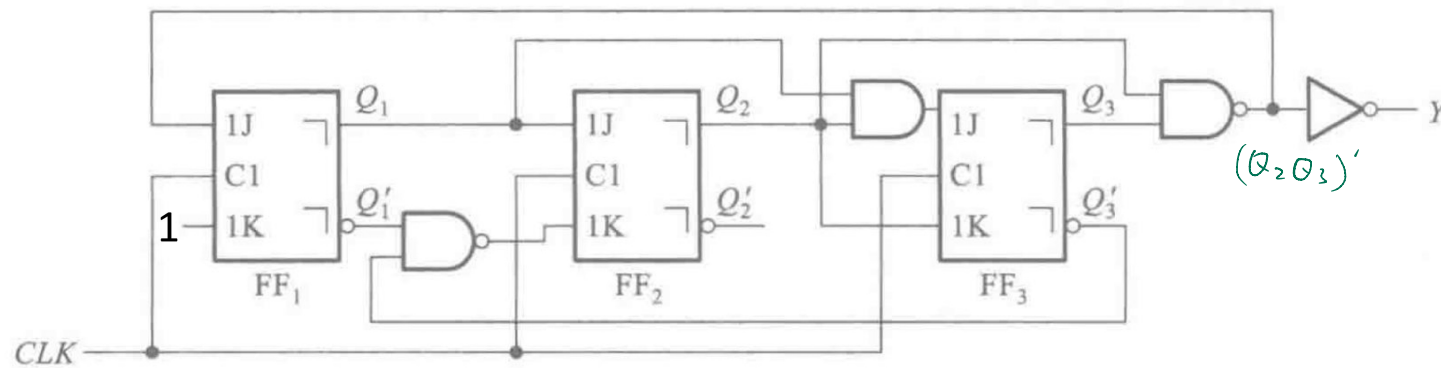


$$\begin{cases} Q_1^* = (Q_2 \cdot Q_3)' Q_1' \\ Q_2^* = Q_1 \cdot Q_2' + Q_1' \cdot Q_3' \cdot Q_2 \\ Q_3^* = Q_1 \cdot Q_2 \cdot Q_3' + Q_2' \cdot Q_3 \end{cases}$$

输出方程



$$Y = Q_2 \cdot Q_3$$



驱动方程

$$J_1 = (Q_2 Q_3)' \quad k_1 = 1$$

$$J_2 = Q_1 \quad k_2 = (Q_1' Q_3')'$$

$$J_3 = Q_1 Q_2 \quad k_3 = Q_2$$

J-k FF $Q^* = JQ' + k'Q$

状态方程

$$Q_1^* = (Q_2 Q_3)' Q_1'$$

$$Q_2^* = J_2 Q_1' + k_2' Q_2 = Q_1 Q_2' + Q_1' Q_3' Q_2$$









$$Q_3^* = Q_1 Q_2 Q_3' + Q_2' Q_3$$

输出方程 $Y = Q_2 Q_3$

$$Q_1^* = (Q_2 Q_3)' Q_1'$$

$$Q_2^* = J_2 Q_1' + K_2' Q_2 = Q_1 Q_2' + Q_1' Q_3' Q_2$$

$$Q_3^* = Q_1 Q_2 Q_3' + Q_2' Q_3$$

clk	Q_3	Q_2	Q_1	Q_3^*	Q_2^*	Q_1^*	Y
	0	0	0	0	0	1	0
	0	0	1	0	1	0	0
	0	1	0	0	1	1	0
	0	1	1	1	0	0	0
	1	0	0	1	0	1	0
	1	0	1	1	1	0	0
	1	1	0	0	0	0	1
	1	1	1	0	0	0	1

状态转换表

- Assume the initial state $Q_3Q_2Q_1 = 000$, one can get the next state $Q_3Q_2Q_1 = 001$ and $Y = 0$
- Repeat this procedure, one can get all the future states
- However, one noticed that $Q_3Q_2Q_1 = 111$ is missing. If we start from this state, the next state is $Q_3Q_2Q_1 = 000$ and $Y = 0$

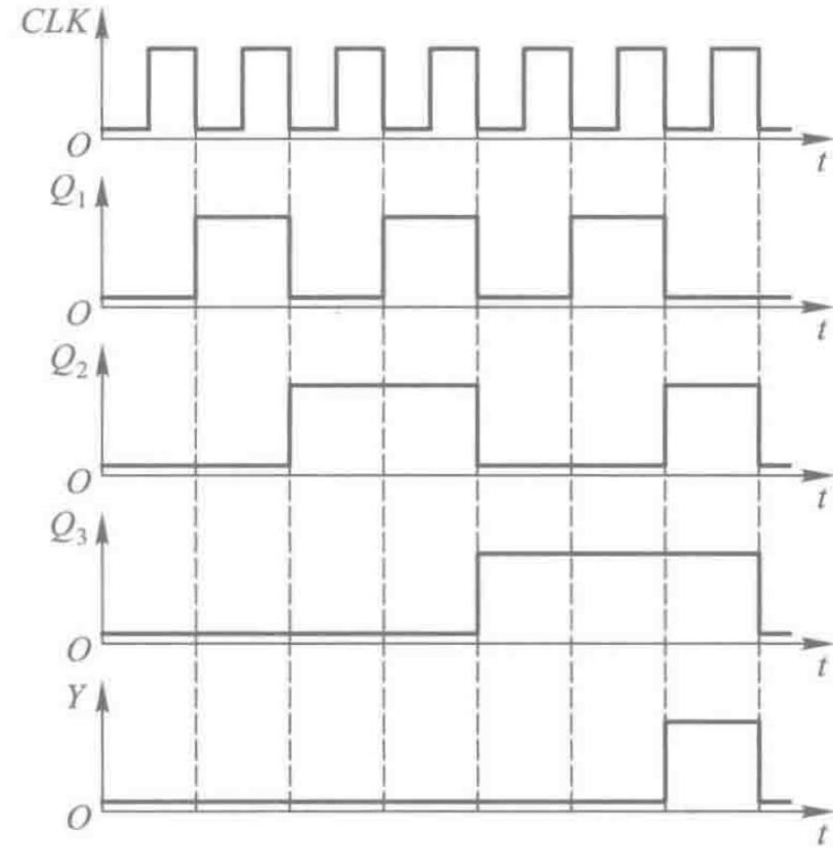
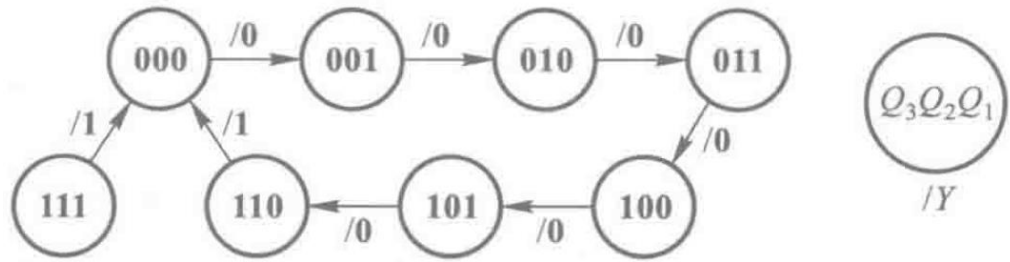
Q_3	Q_2	Q_1	Q_3^*	Q_2^*	Q_1^*	Y
0	0	0	0	0	1	0
0	0	1	0	1	0	0
0	1	0	0	1	1	0
0	1	1	1	0	0	0
1	0	0	1	0	1	0
1	0	1	1	1	0	0
1	1	0	0	0	0	1
1	1	1	0	0	0	1

状态转换表

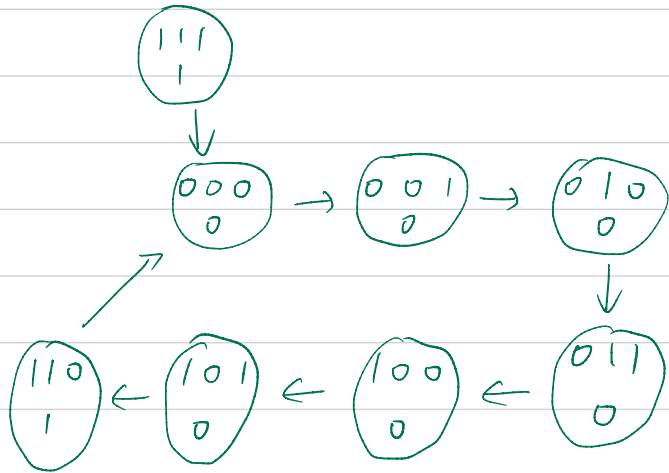
<i>CLK</i> 的顺序	Q_3	Q_2	Q_1	Y
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	0	0	0	0
0	1	1	1	1
1	0	0	0	0

- 这是个七进制计数器， Y 端的输出就是进位脉冲

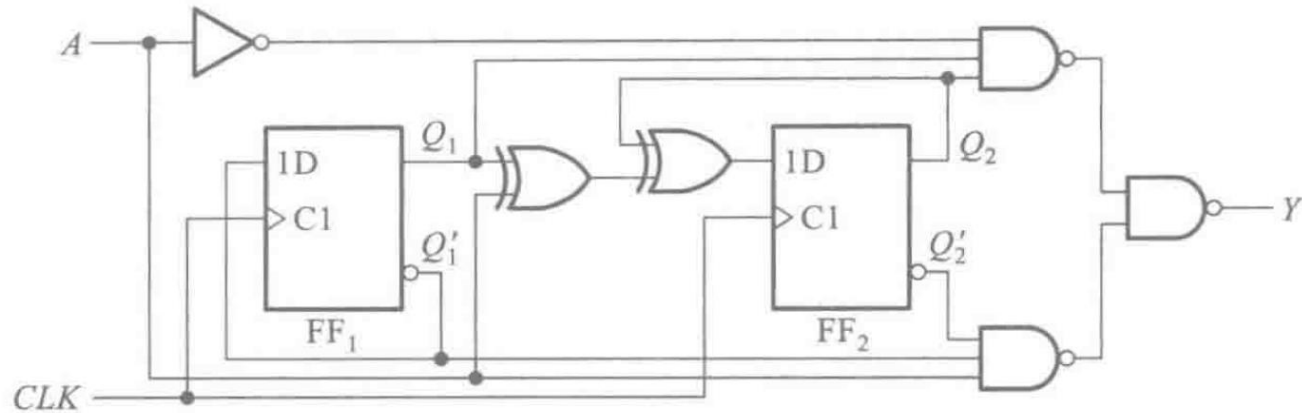
状态转换图&时序图



$Q_3 Q_2 Q_1$
 Y



Sequential Logic example



驱动方程



$$\begin{cases} D_1 = Q'_1 \\ D_2 = A \oplus Q_1 \oplus Q_2 \end{cases}$$

状态方程

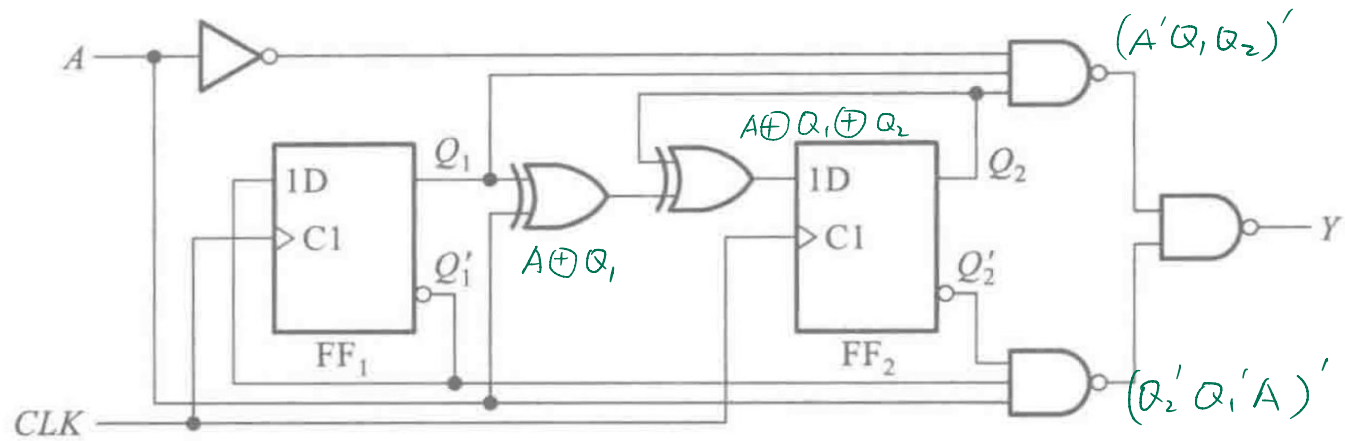


$$\begin{cases} Q_1^* = D_1 = Q'_1 \\ Q_2^* = D_2 = A \oplus Q_1 \oplus Q_2 \end{cases}$$

输出方程



$$\begin{aligned} Y &= ((A'Q_1Q_2)' \cdot (A Q'_1Q'_2)')' \\ &= A'Q_1Q_2 + A Q'_1Q'_2 \end{aligned}$$



驱动方程

$$D_1 = Q_1'$$

$$D_2 = A \oplus Q_1 \oplus Q_2$$

D FF

$$Q^* = D$$

状态方程

$$Q_1^* = Q_1'$$

$$Q_2^* = A \oplus Q_1 \oplus Q_2$$

输出方程

$$Y = [(A'Q_1Q_2)' (Q_2'Q_1'A)']'$$

$$= A'Q_1Q_2 + AQ_1'Q_2'$$

$Q_2^* Q_1^* / Y$	$Q_2 Q_1$				
A		00	01	10	11
0		01/0	10/0	11/0	00/1
1		11/1	00/0	01/0	10/0

$$Q_1^* = Q_1'$$

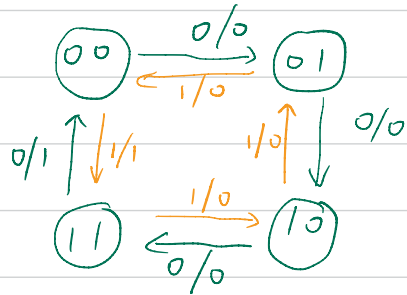
$$Q_2^* = A \oplus Q_1 \oplus Q_2$$

$$Y = [(A'Q_1Q_2)' (Q_2'Q_1'A)']'$$

$$= A'Q_1Q_2 + AQ_1'Q_2'$$

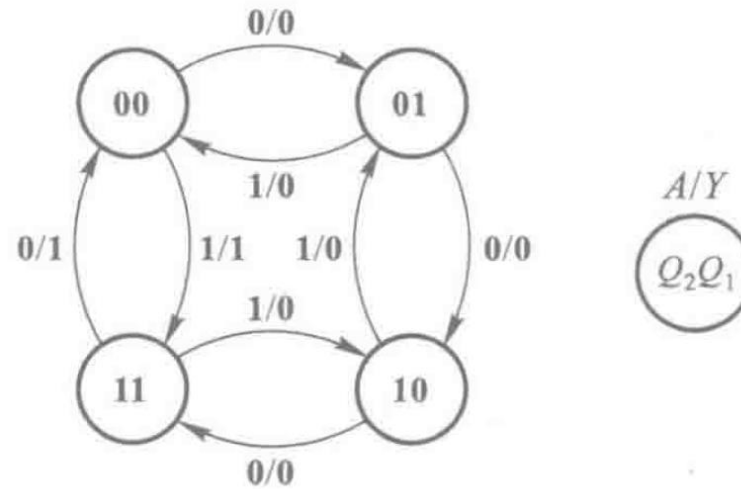
$Q_2 Q_1$

A/Y



Sequential Logic example

$Q_2^* Q_1^* / Y$		$Q_2 Q_1$			
A		00	01	11	10
0		01/0	10/0	00/1	11/0
1		11/1	00/0	10/0	01/0



- 当A=0, 加法计数器
- 当A=1, 减法计数器

Sequential Logic example

