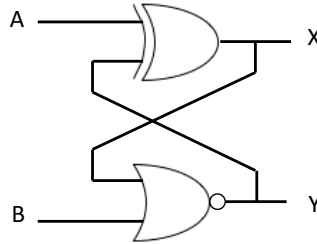


## 1 SR Latches (25 points)

We revise a two-NOR SR latch by replacing a NOR gate with an XOR gate, i.e.  $f(x, y) = x'y + xy'$  as follows. Complete the state table.



- 25pts for correct answers; -2pts for wrong final state  $X_{t+3}Y_{t+3}$  in each row
- 0pt for completely wrong or no attempt

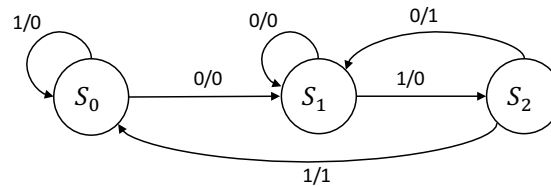
| $id$ | $A$ | $B$ | $X_t$ | $Y_t$ | $X_{t+1}$ | $Y_{t+1}$ | $X_{t+2}$ | $Y_{t+2}$ | $X_{t+3}$ | $Y_{t+3}$ |
|------|-----|-----|-------|-------|-----------|-----------|-----------|-----------|-----------|-----------|
| 0    | 0   | 0   | 0     | 0     | 0         | 1         | 1         | 1         | 1         | 0         |
| 1    | 0   | 0   | 0     | 1     | 1         | 1         | 1         | 0         | 0         | 0         |
| 2    | 0   | 0   | 1     | 0     | 0         | 0         | 0         | 1         | 1         | 1         |
| 3    | 0   | 0   | 1     | 1     | 1         | 0         | 0         | 0         | 0         | 1         |
| 4    | 0   | 1   | 0     | 0     | 0         | 0         | 0         | 0         | 0         | 0         |
| 5    | 0   | 1   | 0     | 1     | 1         | 0         | 0         | 0         | 0         | 0         |
| 6    | 0   | 1   | 1     | 0     | 0         | 0         | 0         | 0         | 0         | 0         |
| 7    | 0   | 1   | 1     | 1     | 1         | 0         | 0         | 0         | 0         | 0         |
| 8    | 1   | 0   | 0     | 0     | 1         | 1         | 0         | 0         | 1         | 1         |
| 9    | 1   | 0   | 0     | 1     | 0         | 1         | 0         | 1         | 0         | 1         |
| 10   | 1   | 0   | 1     | 0     | 1         | 0         | 1         | 0         | 1         | 0         |
| 11   | 1   | 0   | 1     | 1     | 0         | 0         | 1         | 1         | 0         | 0         |
| 12   | 1   | 1   | 0     | 0     | 1         | 0         | 1         | 0         | 1         | 0         |
| 13   | 1   | 1   | 0     | 1     | 0         | 0         | 1         | 0         | 1         | 0         |
| 14   | 1   | 1   | 1     | 0     | 1         | 0         | 1         | 0         | 1         | 0         |
| 15   | 1   | 1   | 1     | 1     | 0         | 0         | 1         | 0         | 1         | 0         |

## 2 Finite State Machine (30 points)

A pattern recognizer inputs a string of binary bits  $x(t)$  and outputs  $y(t) = 1$  whenever  $x(t-2, t) = 011$  or  $010$ ;  $y(t) = 0$  otherwise.

- For 2.1, 10pts for correct state diagram; -2 pts for each missing input, output or state transition; -8pts for using 4 or more states
- For 2.2, 10pts for correct state table; -2 pts for each error in the table; 0pt for no attempt.
- For 2.3, 10pts for correct expressions; -2 pts for each incorrect expression. If all the expressions are incorrect, -6pts if the 3 corresponding K-maps are given; -8pts for no K-map but the excitation table is given; 0pt for no attempt.

1. Design the state machine with less than 4 states. Draw the state diagram.



2. Design the state machine with 2 D flip-flops. Write the state table.

| PS \ Input   | $x(t) = 0$ | $x(t) = 1$ |
|--------------|------------|------------|
| $S_0$ : Init | $S_1, 0$   | $S_0, 0$   |
| $S_1$ : 0    | $S_1, 0$   | $S_2, 0$   |
| $S_2$ : 01   | $S_1, 1$   | $S_0, 1$   |

3. Design the system with two D flip-flops. Describe  $y(t)$ ,  $D_1(t)$ ,  $D_0(t)$  as functions of variables  $(x, Q_1(t), Q_0(t))$  in minimal sum-of-products expressions. No need to draw the logic diagram.

Assign binary bits for each state,

$$S_0 = 00$$

$$S_1 = 01$$

$$S_2 = 10$$

The excitation table is derived from the state table. Since we are using D Flip-Flops to design the state machine, the state expressions for the 2 D Flip-Flops are

$$D_1(t) = Q_1(t+1)$$

$$D_0(t) = Q_0(t+1)$$

| $id$ | $Q_1(t)$ | $Q_0(t)$ | $x(t)$ | $Q_1(t+1)$ | $Q_0(t+1)$ | $y(t)$ |
|------|----------|----------|--------|------------|------------|--------|
| 0    | 0        | 0        | 0      | 0          | 1          | 0      |
| 1    | 0        | 0        | 1      | 0          | 0          | 0      |
| 2    | 0        | 1        | 0      | 0          | 1          | 0      |
| 3    | 0        | 1        | 1      | 1          | 0          | 0      |
| 4    | 1        | 0        | 0      | 0          | 1          | 1      |
| 5    | 1        | 0        | 1      | 0          | 0          | 1      |
| 6    | 1        | 1        | 0      | X          | X          | X      |
| 7    | 1        | 1        | 1      | X          | X          | X      |

K-maps are used to simplify the expression. With the following K-map for  $D_1(t)$ , we have  $D_1(t) = Q_0(t)X(t)$ .

| $X(t) \setminus Q_1(t)Q_0(t)$ | 00  | 01  | 11  | 10  |
|-------------------------------|-----|-----|-----|-----|
| 0                             | 0 0 | 0 2 | X 6 | 0 4 |
| 1                             | 0 1 | 1 3 | X 7 | 0 5 |

For  $D_0(t)$ , we have  $D_0(t) = X'(t)$ .

| $X(t) \setminus Q_1(t)Q_0(t)$ | 00  | 01  | 11  | 10  |
|-------------------------------|-----|-----|-----|-----|
| 0                             | 1 0 | 1 2 | X 6 | 1 4 |
| 1                             | 0 1 | 0 3 | X 7 | 0 5 |

For  $y(t)$ , we have  $y(t) = Q_1(t)$ .

| $X(t) \setminus Q_1(t)Q_0(t)$ | 00  | 01  | 11  | 10  |
|-------------------------------|-----|-----|-----|-----|
| 0                             | 0 0 | 0 2 | X 6 | 1 4 |
| 1                             | 0 1 | 0 3 | X 7 | 1 5 |

### 3 Mealy and Moore Machine Conversion (20 points)

Given a state table as below, each entry of the table is the next state and output, i.e.  $Q_1(t+1) Q_0(t+1), y$ .

| $id$ | $Q_1(t) Q_0(t)$ | $x(t) = 0$ | $x(t) = 1$ |
|------|-----------------|------------|------------|
| 0    | 0 0             | 00,0       | 01,0       |
| 1    | 0 1             | 00,0       | 10,0       |
| 2    | 1 0             | 01,1       | 11,0       |
| 3    | 1 1             | 00,1       | 10,0       |

- For 3.1, 5pts for correct outputs; -1pt for each error in  $y$
- For 3.2, 10pts for correct state table with Moore machine (including 3pts for each extra state/output, 2pts for each transition/output from existing states to new states); -2.5pts for missing output column but correct outputs in 3.3; -5pts if there is no state table but a state diagram
- For 3.3, 5 pts for correct outputs; -1pt for each error in  $y$ ; -2.5pts for incorrect answer but consistent to state table in 3.2

1. Fill the following table as the machine starts at state  $Q_1(t=1) Q_0(t=1) = 00$  and process through the given input  $x$  from CLK (Clock) sequence =1 to 8.

| CLK   | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  |
|-------|----|----|----|----|----|----|----|----|
| $x$   | 1  | 0  | 1  | 1  | 0  | 1  | 1  | 1  |
| state | 00 | 01 | 00 | 01 | 10 | 01 | 10 | 11 |
| $y$   | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  |

2. Convert the above machine into a Moore machine. Write the state table. Notice that extra variables are needed to store all the states. The original states are kept by adding one more variable  $Q_2(t) = 0$ .

| $id$ | $Q_2(t) Q_1(t) Q_0(t)$ | $x(t) = 0$ | $x(t) = 1$ | $y(t)$ |
|------|------------------------|------------|------------|--------|
| 0    | 0 0 0                  | 000        | 001        | 0      |
| 1    | 0 0 1                  | 000        | 010        | 0      |
| 2    | 0 1 0                  | 100        | 011        | 0      |
| 3    | 0 1 1                  | 101        | 010        | 0      |
| 4    | 1 0 0                  | 000        | 010        | 1      |
| 5    | 1 0 1                  | 000        | 001        | 1      |

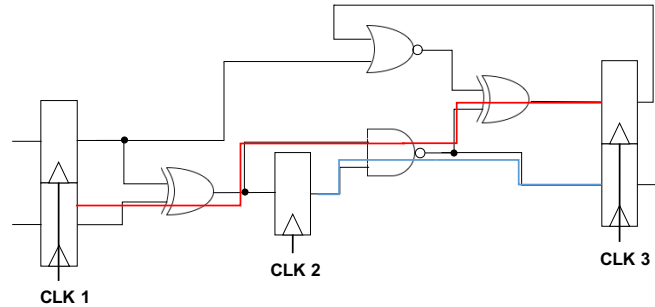
3. Use the Moore machine to fill the table starting at state  $Q_1(t=1) Q_0(t=1) = 00$  and process through the given input  $x$  from CLK (Clock) sequence 1 to 8.

| CLK   | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|
| $x$   | 1   | 0   | 1   | 1   | 0   | 1   | 1   | 1   |
| state | 000 | 001 | 000 | 001 | 010 | 100 | 010 | 011 |
| $y$   | 0   | 0   | 0   | 0   | 0   | 1   | 0   | 0   |

## 4 Timing and Retiming (25 points)

For the following circuit, the timing characteristics of the components are summarized below.

- Flip-flop: clock-to-Q maximum delay (propagation delay)  $t_{pcq} = 30\text{ps}$ , clock-to-Q minimum delay (contamination delay)  $t_{ccq} = 20\text{ps}$ , setup time  $t_{setup} = 20\text{ps}$ , hold time  $t_{hold} = 30\text{ps}$
- Logic gate (each AND, OR, XOR): propagation delay  $t_{pd}(\text{gate}) = 25\text{ps}$ , contamination delay  $t_{cd}(\text{gate}) = 15\text{ps}$ .



- **For 4.1, 15pts for correct answer; -5pts for incorrect answer due to wrong  $t_{pd(max)}$**
- **For 4.2, 10pts for correct answer; -5pts for incorrect answer due to wrong  $t_{cd(min)}$**
- **0pt for completely incorrect answers or no attempt**

1. Derive the minimal clock period  $T$ , suppose that there is no clock skew.

The longest path of combinational logic between 2 flip-flops passes 3 logic gates.

$$t_{pd(max)} = 3 \times t_{pd(gate)} = 75\text{ps}$$

The clock period should satisfy the setup time constraint

$$T_c \geq t_{pcq} + t_{pd(max)} + t_{setup} = 30 + 75 + 20 = 125\text{ps}$$

So the minimal clock period is 125ps.

2. How much clock skew can the circuit tolerate before it experiences a hold time violation?

The minimum contamination delay of the combinational logic passes one logic gate.

$$t_{cd(min)} = 1 \times t_{cd(gate)} = 15\text{ps}$$

Consider clock skew, the hold time constraint gives

$$t_{ccq} + t_{cd(min)} = 20 + 15 = 35\text{ps} \geq t_{hold} + t_{skew}$$

So  $t_{skew} \leq 35 - 30 = 5\text{ps}$ , the circuit can tolerate a  $t_{skew} = 5\text{ps}$  before it experiences a hold time violation.