Discussion 6
Digital Circuit

Yutong Wang

<wangyt32023@>

3/21/2025

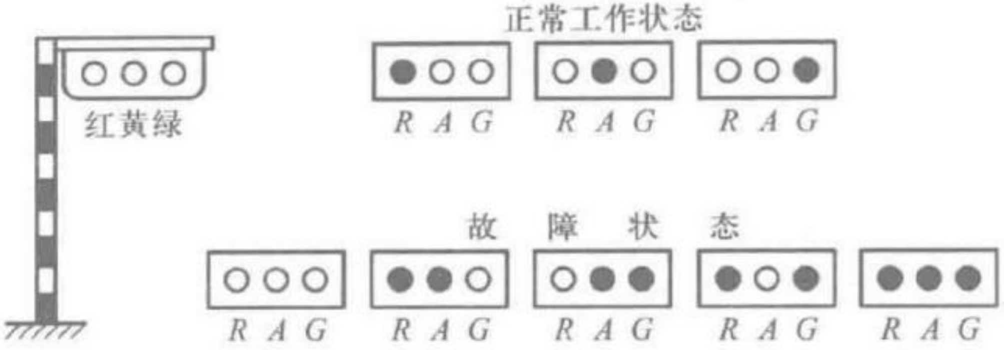
Logic

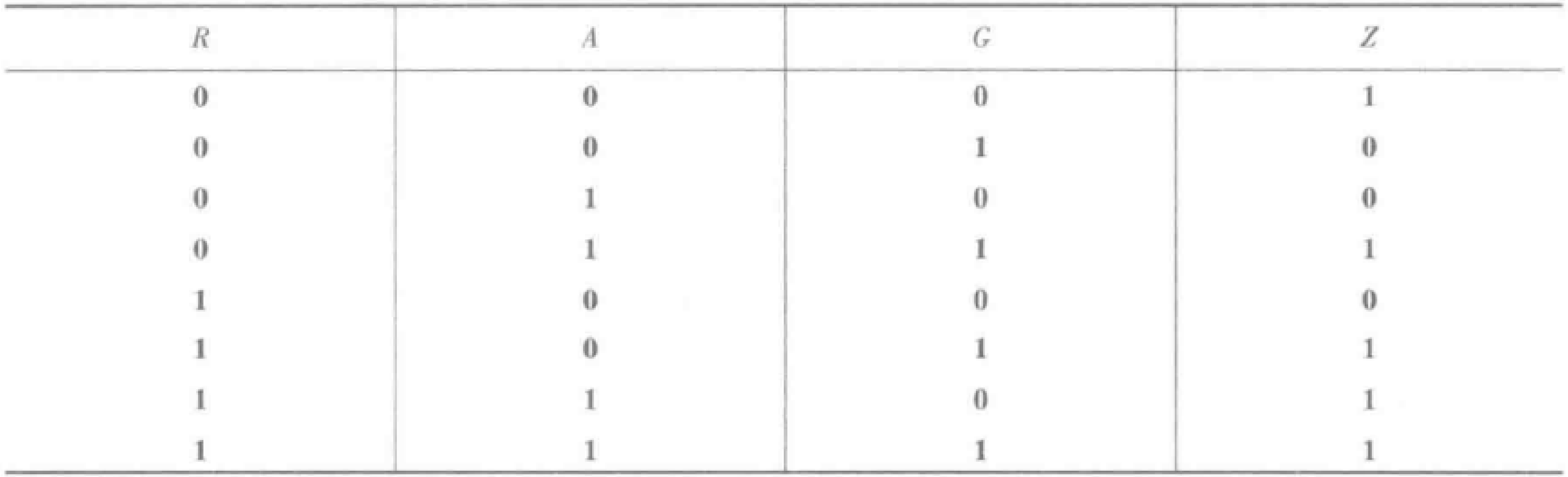
Laws of Boolean Algebra

AND form	OR form	
$X\overline{X} = 0$	$X+\overline{X} = 1$	Complementarity
$X0 = 0$	$X+1 = 1$	Laws of 0’s and 1’s
$X1 = X$	$X+0 = X$	Identities
$XX = X$	$X+X = X$	Idempotent Laws
$XY = YX$	$X+Y = Y+X$	Commutativity
$(XY)Z = X(YZ)$	$(X+Y)+Z = X+(Y+Z)$	Associativity
$X(Y+Z) = XY+XZ$	$X+YZ = (X+Y)(X+Z)$	Distribution
$XY+X = X$	$(X+Y)X = X$	Absorption
$\overline{XY} = \overline{X}+\overline{Y}$	$\overline{X+Y} = \overline{X}\,\overline{Y}$	DeMorgan’s Law

Example

Logic abstraction





R	A	G	Z
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Example

- Get the Boolean expression

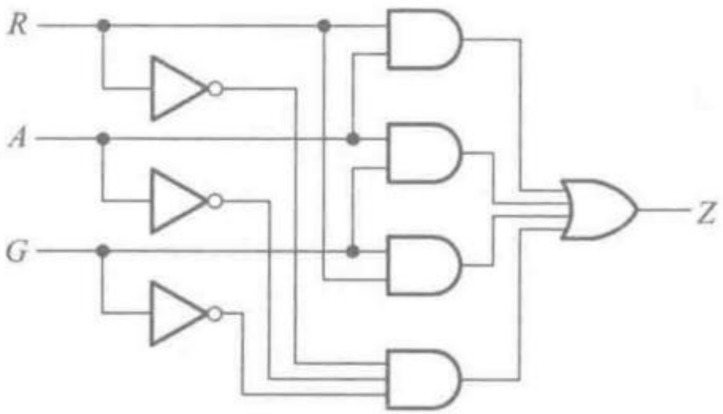
$$Z = R'A'G' + R'AG + RA'G + RAG' + RAG$$

- Simplify the Boolean expression

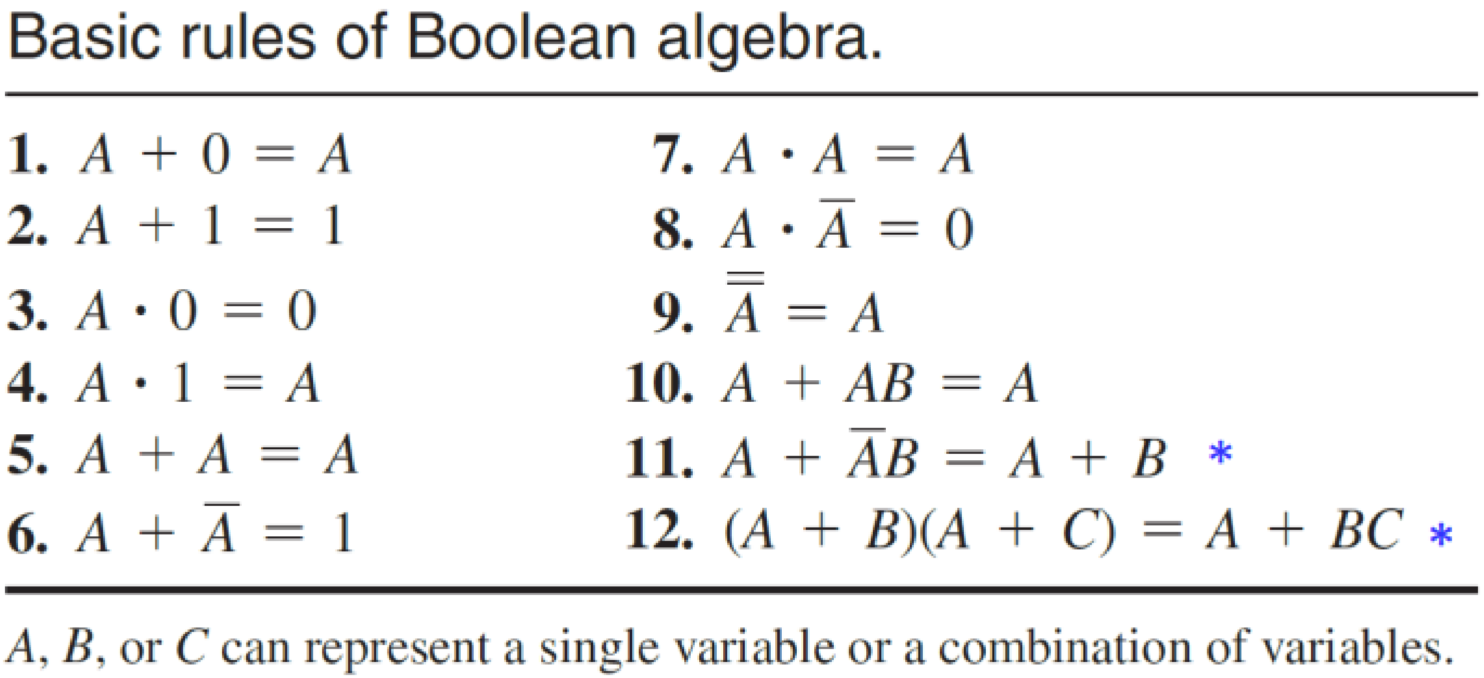
$$Z = R'A'G' + RA + RG + AG$$

How about the Karnauph map?

- Draw the circuit



Rules of Boolean Algebra



Basic rules of Boolean algebra.

1. $A + 0 = A$	**7.** $A \cdot A = A$
2. $A + 1 = 1$	**8.** $A \cdot \overline{A} = 0$
3. $A \cdot 0 = 0$	**9.** $\overline{\overline{A}} = A$
4. $A \cdot 1 = A$	**10.** $A + AB = A$
5. $A + A = A$	**11.** $A + \overline{A}B = A + B$ *
6. $A + \overline{A} = 1$	**12.** $(A + B)(A + C) = A + BC$ *

A, B, or C can represent a single variable or a combination of variables.

Proof:

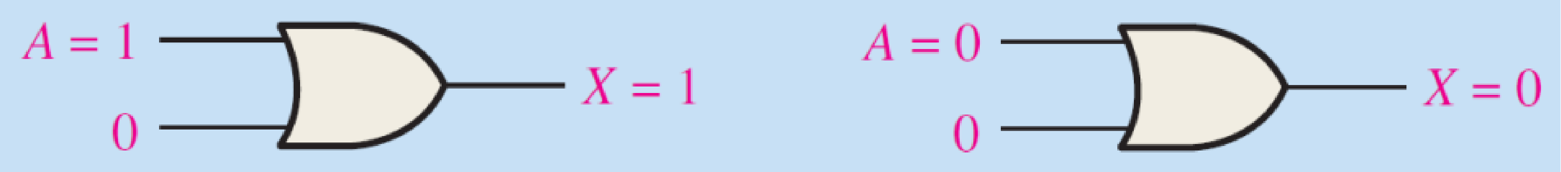
AB+A’C+BC = AB + A’C

Proof:

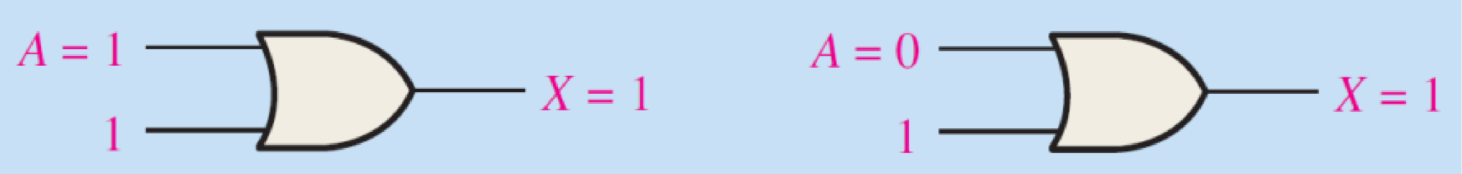
AB+A’C+BCD = AB + A’C

Rules of Boolean Algebra

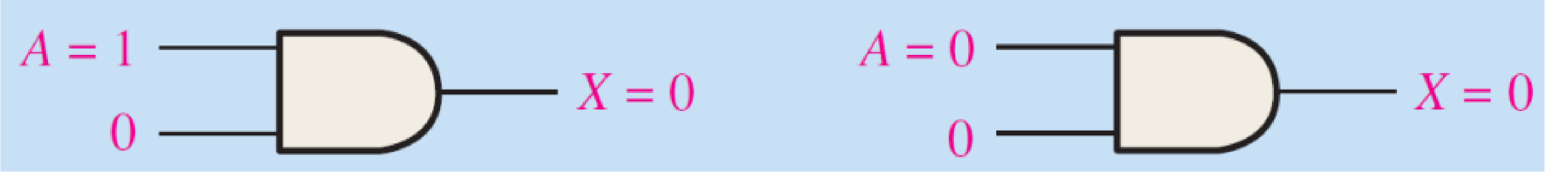
- Rule 1: A + 0 = A, A variable ORed with 0 is always equal to the variable



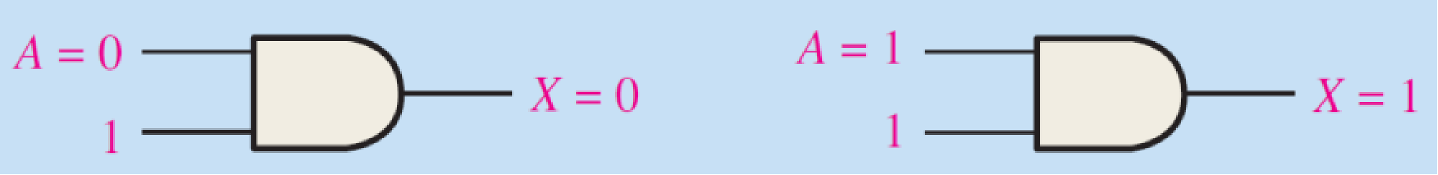
- Rule 2: A + 1 = 1, A variable ORed with 1 is always equal to 1.



- Rule 3: A · 0 = 0, A variable ANDed with 0 is always equal to 0.

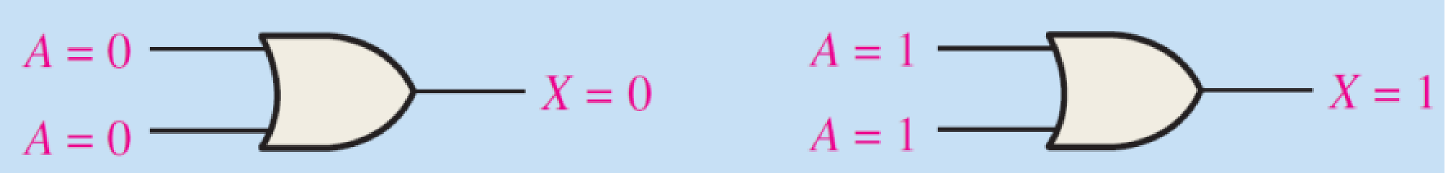


- Rule 4: A · 1 = A, A variable ANDed with 1 is always equal to the variable.

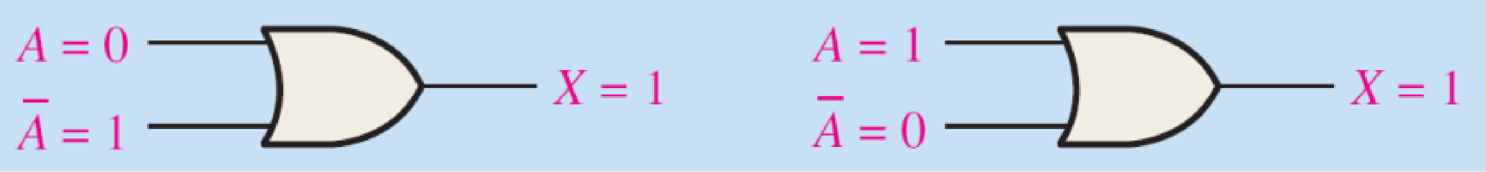


Rules of Boolean Algebra

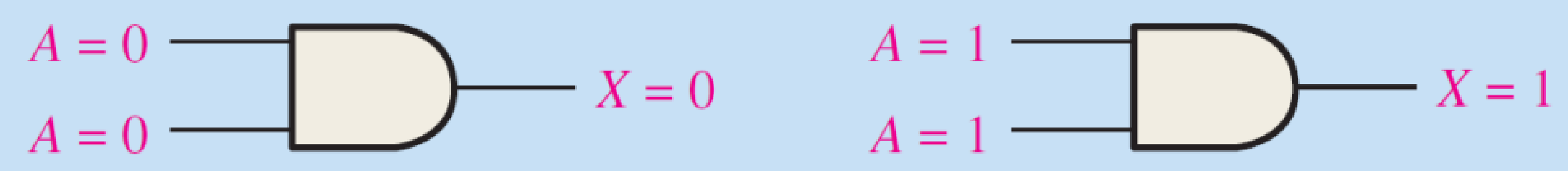
- Rule 5: A + A = A, A variable ORed with itself is always equal to the variable.



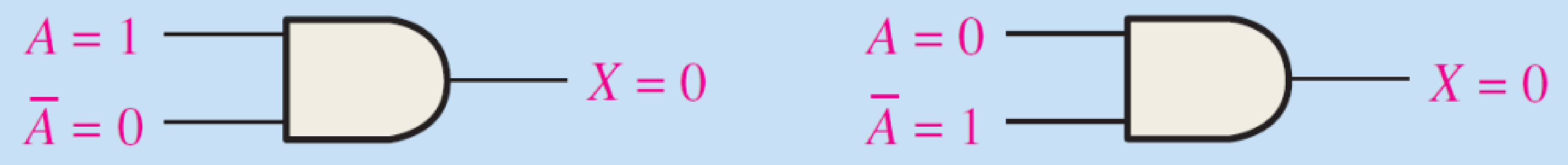
- Rule 6: A + Abar = 1, A variable ORed with its complement is always equal to 1.



- Rule 7: A · A = A, A variable ANDed with itself is always equal to the variable.

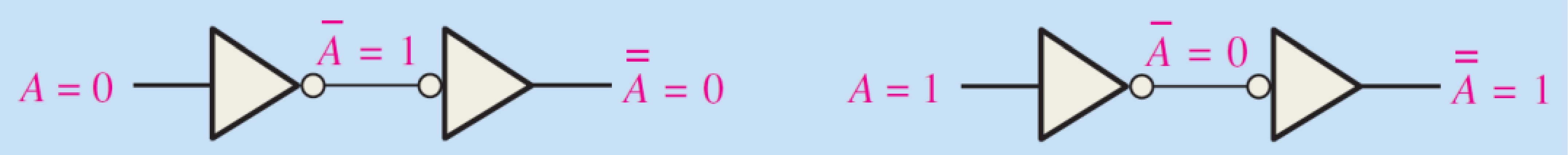


- Rule 8: A · Abar = 0, A variable ANDed with its complement is always equal to 0.



Rules of Boolean Algebra

- Rule 9: The double complement of a variable is always equal to the variable.



- Rule 10: A + AB = A

$$\begin{aligned} A + AB &= A \cdot 1 + AB = A(1 + B) \\ &= A \cdot 1 \\ &= A \end{aligned}$$

Rules of Boolean Algebra

- Rule 11: A + A’B = A + B

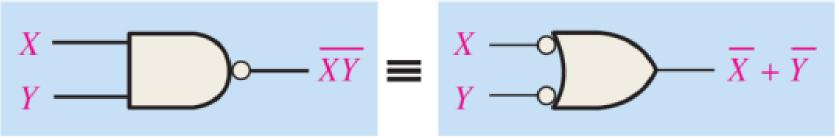
$$
\begin{aligned}
A + \overline{A}B &= (A + AB) + \overline{A}B \\
&= (AA + AB) + \overline{A}B \\
&= AA + AB + A\overline{A} + \overline{A}B \\
&= (A + \overline{A})(A + B) \\
&= 1 \cdot (A + B) \\
&= A + B
\end{aligned}
$$

- Rule 12: (A + B)(A + C) = A + BC

$$
\begin{aligned}
(A + B)(A + C) &= AA + AC + AB + BC \\
&= A + AC + AB + BC \\
&= A(1 + C) + AB + BC \\
&= A \cdot 1 + AB + BC \\
&= A(1 + B) + BC \\
&= A \cdot 1 + BC \\
&= A + BC
\end{aligned}
$$

Rules of Boolean Algebra

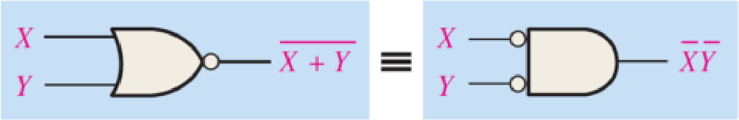
- DeMorgan's first theorem $\overline{XY} = \overline{X} + \overline{Y}$



NAND Negative-OR

Inputs		Output	
X	Y	$\overline{XY}$	$\overline{X} + \overline{Y}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

- DeMorgan's second theorem $\overline{X + Y} = \overline{X}\,\overline{Y}$



NOR Negative-AND

Inputs		Output	
X	Y	$\overline{X + Y}$	$\overline{X}\,\overline{Y}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

Think about it:

- 3 variable DeMorgan's Theorems?

- DeMorgan's theorems provide mathematical equivalency of the NAND and negative-OR gates and the equivalency of the NOR and negative-AND gates

Rules of Boolean Algebra

- **代入定理**

在任何一个包含变量*A*的逻辑等式中，若以另外一个逻辑式代入式中所有*A*的位置，则等式仍然成立

e.g., 3 variable DeMorgan’s Theorems?

- **反演定理**

对于任意一个逻辑式*Y*，若将其中所有的·换成+，+换成·，0换成1，1换成0，原变量换成反变量，反变量换成原变量，则得到的结果就是*Y′*

$Y=A(B+C)+CD$ → $Y'=(A'+B'C')(C'+D')$

- **对偶定理**

若两逻辑式相等，则它们的对偶式也相等

对偶式：对于任何一个逻辑式*Y*，若将其中的·换成+，+换成·，0换成1，1换成0，则得到*Y*的对偶式Y^D。

$Y=A+BC$ → $Y^D=A(B+C)$

$Y=(A+B)(A+C)$ → $Y^D=AB+AC$

Exercise:

$(AB + AC)' + A'B'C$

Midterm 2024

(a) **(Multiple Choice)** Which of the following statement(s) are(is) true about boolean algebra? (   )

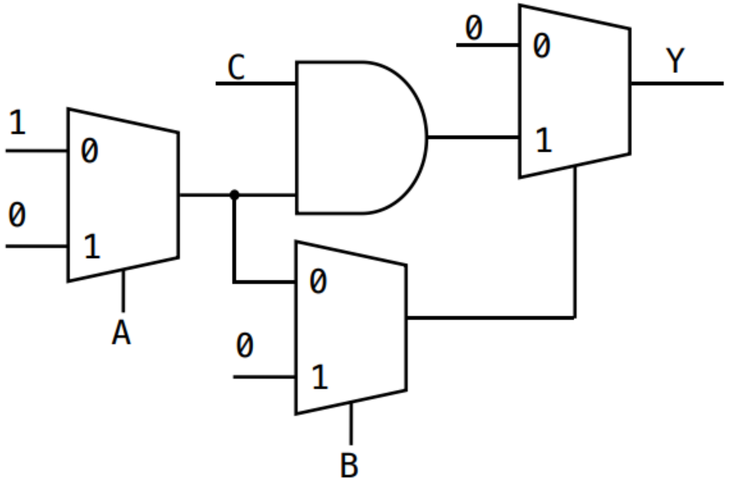
A. $X + YZ = (X + Y)(X + Z)$

B. $(X + \overline{Y})X = X + X\overline{Y}$

C. $XY + X = X$

D. $\overline{XY} = \overline{X} + \overline{Y}$

(b) The following circuit is composed of several basic logic gates and 2-to-1 multiplexers. Please write down the truth table of the circuit below.

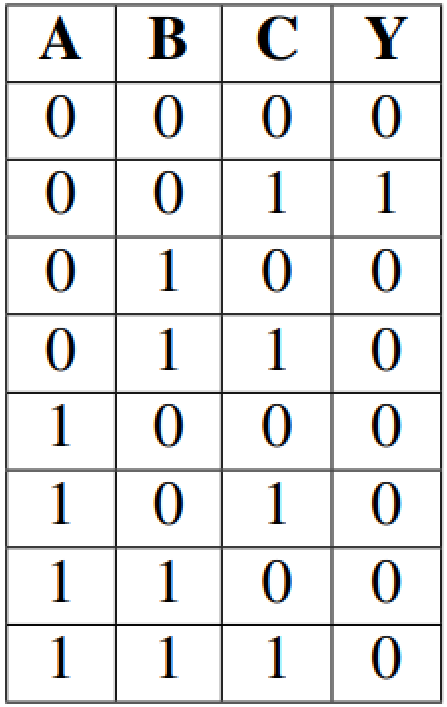


(c) Write down the logic expression that implements the truth table using sum of minterm.

Build a logic circuit that uses only 2-input **AND**, 2-input **OR** and **NOT** gates implementing the same logic above. Use as less logic gates as possible.

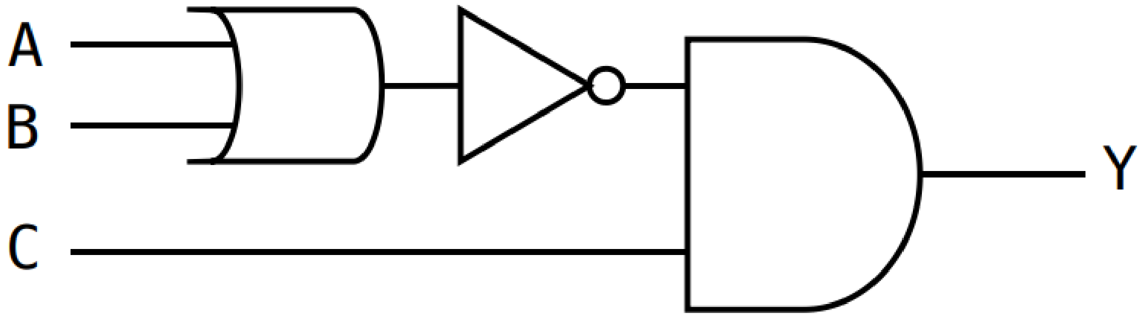
Midterm 2024

Truth Table



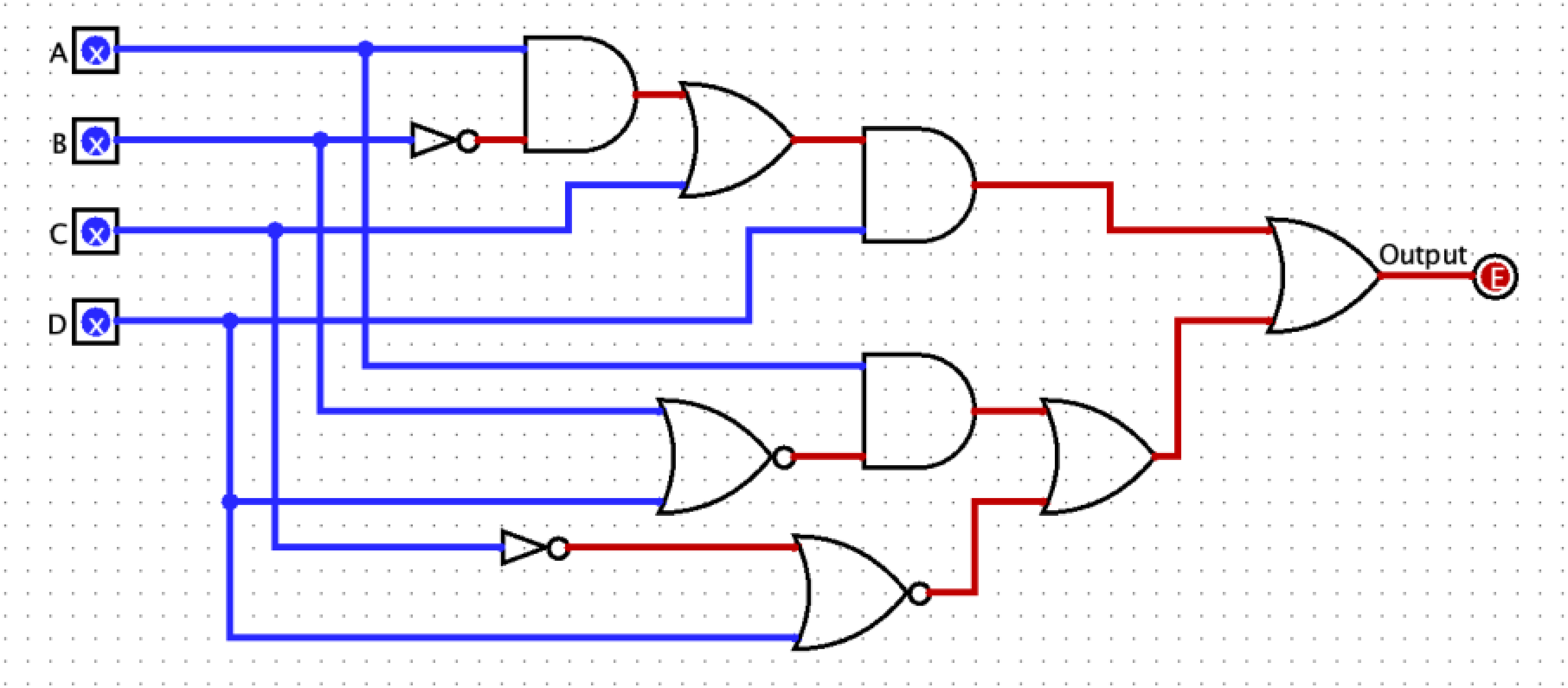
A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

$$Y = \bar{A}\bar{B}C.$$



Midterm 2023

Please write down the truth table of the circuit below and draw its Karnaugh map (A&B as a group; C&D as a group).



Midterm 2023

Truth table:

A	B	C	D	Output
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

Karnaugh map:

		AB			
		00	01	11	10
CD	00	0	0	0	1
	01	0	0	0	1
	11	1	1	1	1
	10	1	1	1	1

$$(A\overline{B}+C)D + A(\overline{B+D}) + \overline{\overline{C}+D} = A\overline{B}D + CD + A\overline{B}\overline{D} + C\overline{D} = A\overline{B} + C$$

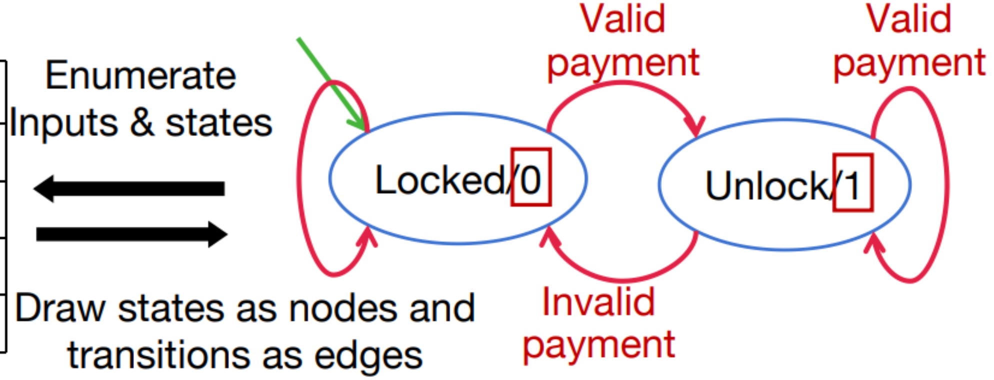
FSM

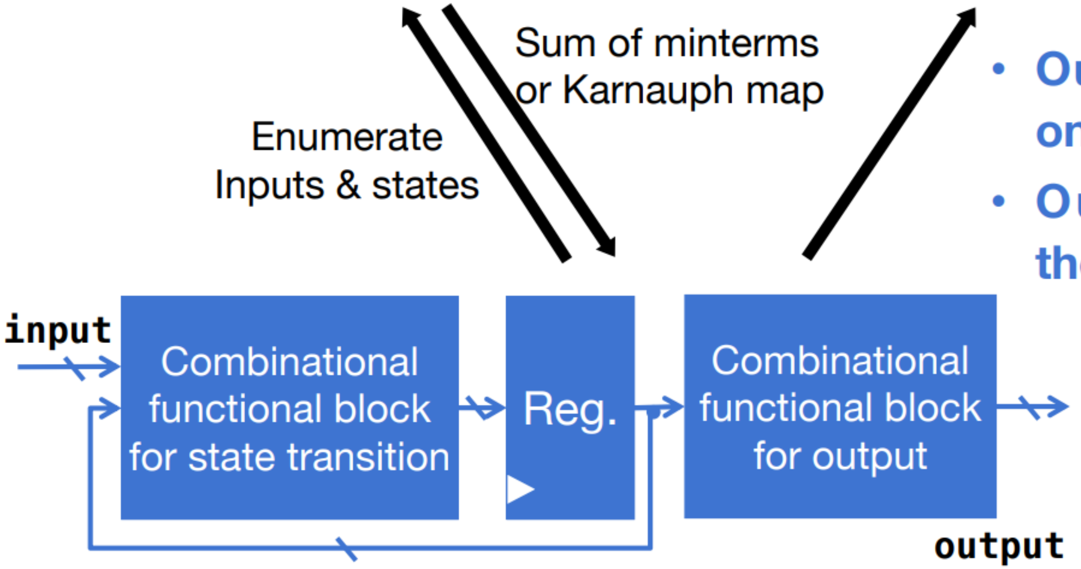
Moore machine

Moore machine vs. Mealy machine

- Moore machine

input	Q_{k-1}	Q_k	output
0	0	0	0
0	1	0	0
1	1	1	1
1	0	1	1





- **Output depends solely on the current state**
- **Output controlled by the clock somewhat**

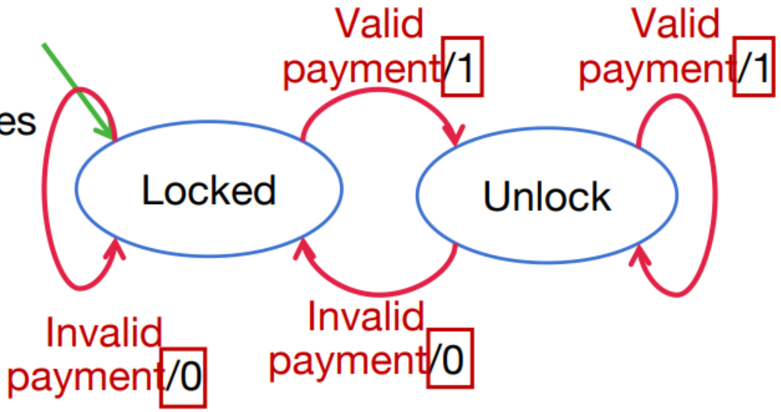
Mealy machine

Moore machine vs. Mealy machine

- Mealy machine

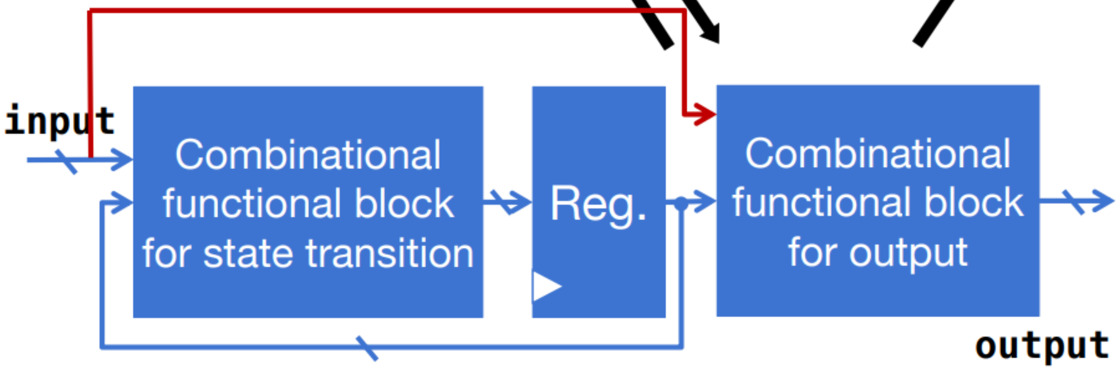
input	Q_{k-1}	Q_k	output
0	0	0	0
0	1	0	0
1	1	1	1
1	0	1	1

Enumerate Inputs & states



Sum of minterms or Karnauph map

Enumerate Inputs & states



input
Combinational functional block for state transition
Reg.
Combinational functional block for output
output

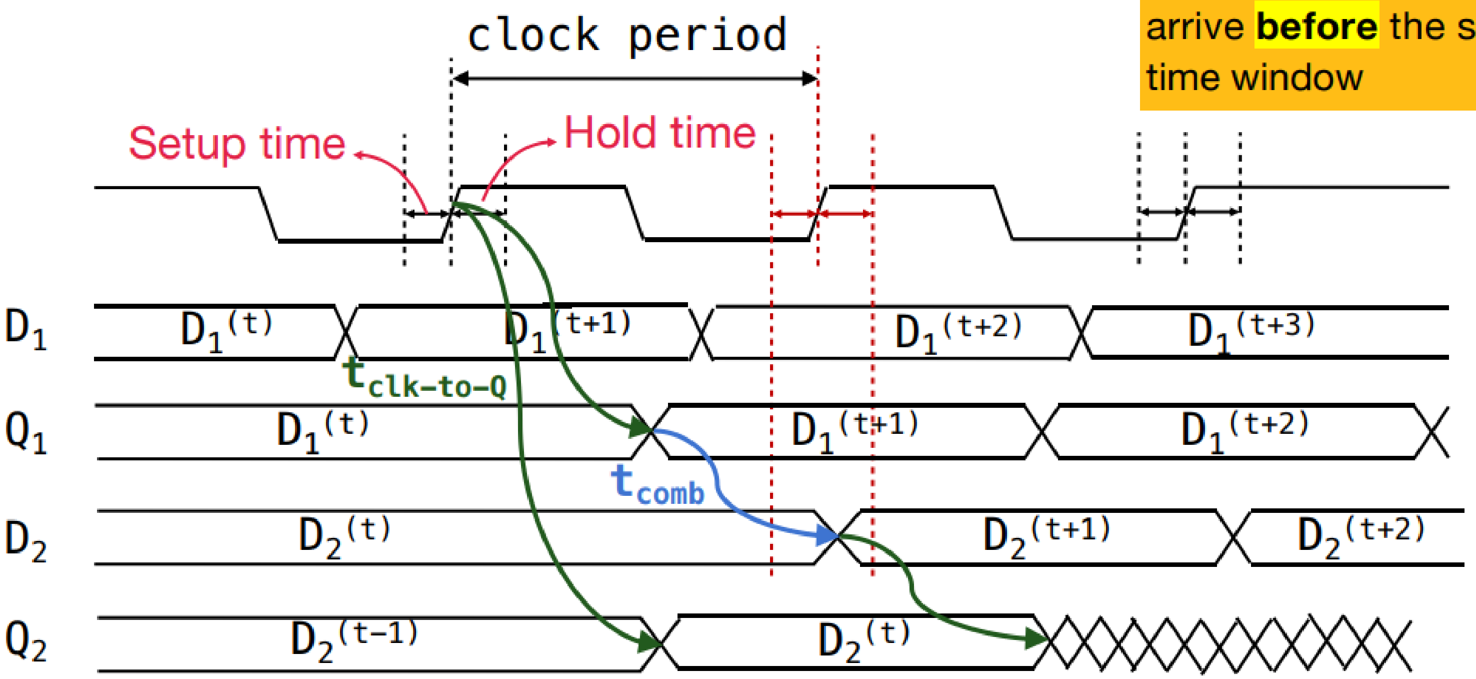
- Output depends on both the current state **and input**
- Output uncontrolled by the clock
- Mealy and Moore machines are interchangable
- Mealy and Moore outputs can co-exist in one synchromous circuit

Timing

Estimating the max frequency

Max frequency = 1/min clock period

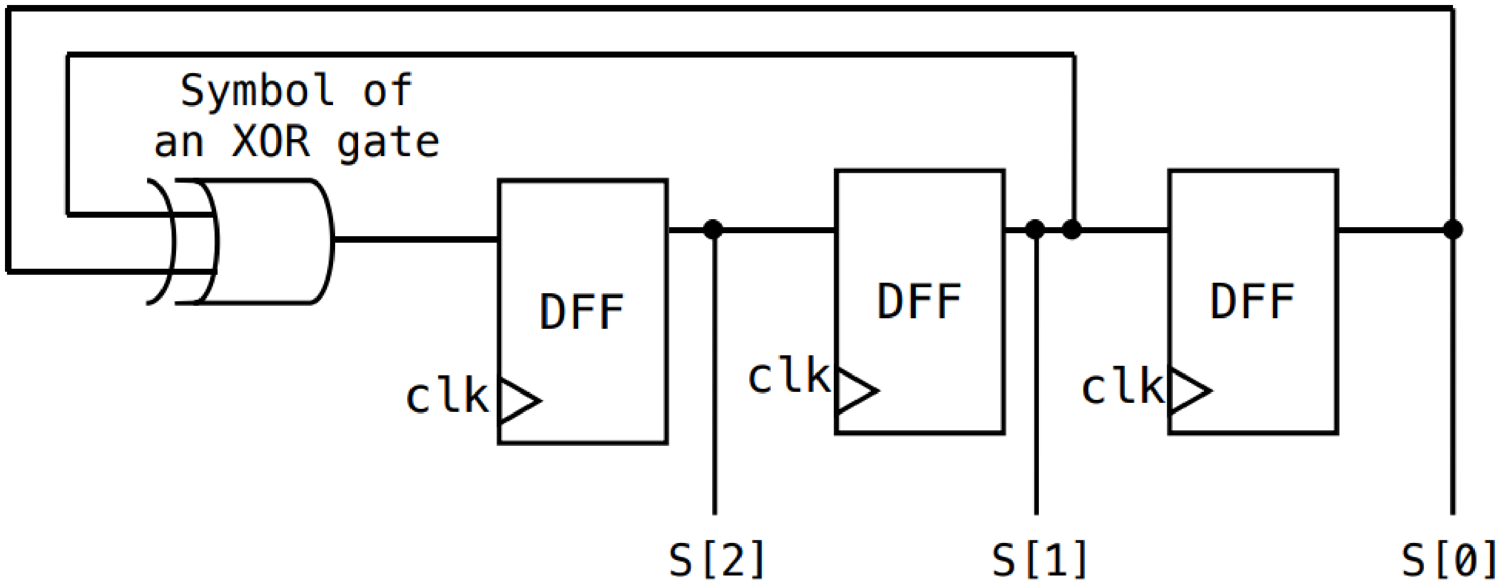
$t_{clk-to-Q} + t_{comb} \le$ **min clock period – setup time**



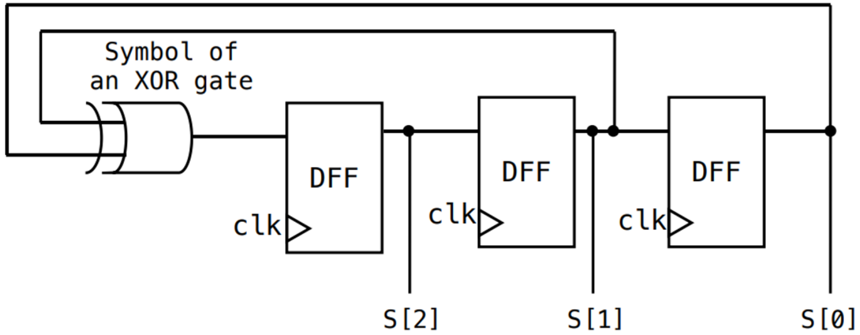
Signal $D_2^{(t+1)}$ should arrive **before** the setup time window

Midterm 2024

Below shows a synchronous circuit called linear feedback shift register (LFSR), consisting of one or more **XOR** gates and several DFFs. It has been widely used for generating pseudorandom numbers. It can be modeled by a finite state machine (FSM) like the other synchronous circuits, however, without any input signals. Given the current state S_{k-1}, fill in the truth table of its next state S_k. S is a 3-bit signal.



Midterm 2024



Truth Table

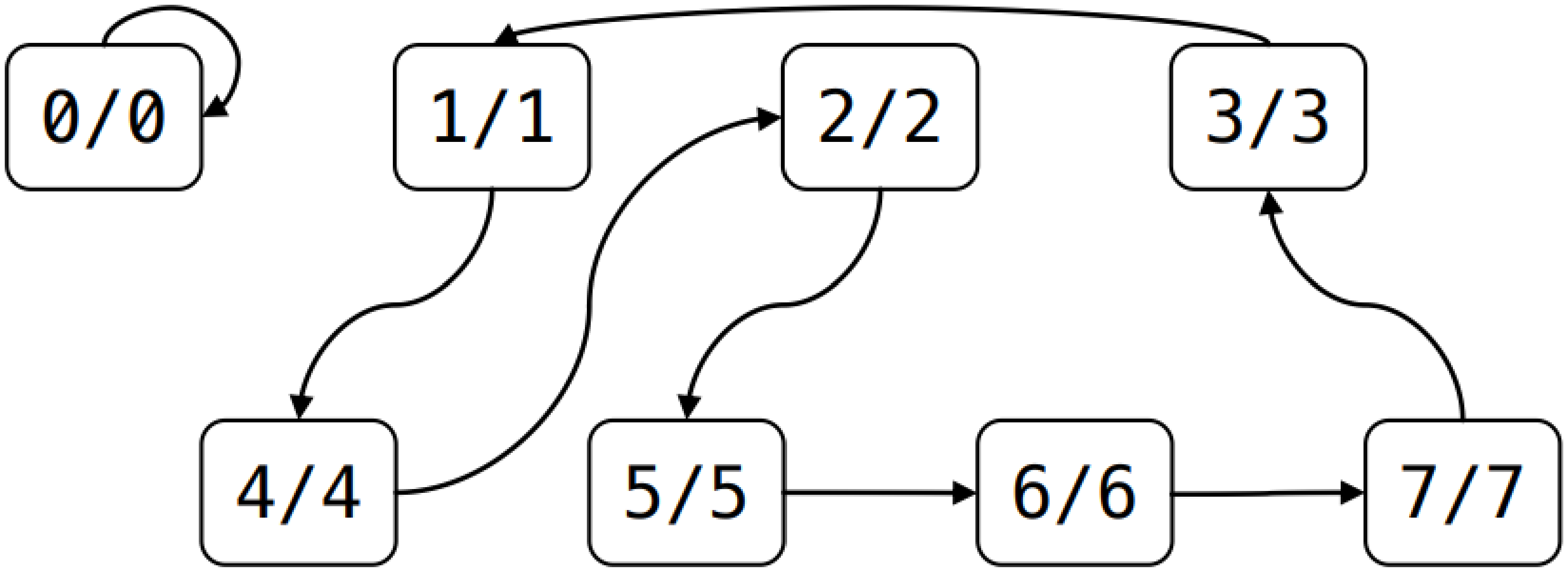
$S[2]_{k-1}$	$S[1]_{k-1}$	$S[0]_{k-1}$	$S[2]_k$	$S[1]_k$	$S[0]_k$
0	0	0	0	0	0
0	0	1	1	0	0
0	1	0	1	0	1
0	1	1	0	0	1
1	0	0	0	1	0
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	1	1

Midterm 2024

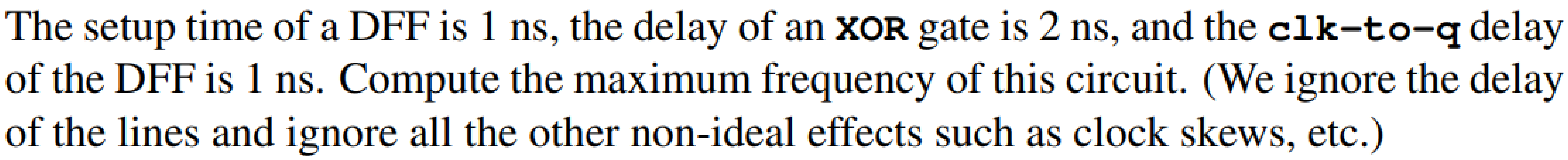
For the above FSM, we use the unsigned number $(\mathrm{S}[2]\mathrm{S}[1]\mathrm{S}[0])_2$ to represent its state and output. Please complete the state transition diagram below. Tips: This FSM has no input, and we do not put the transition condition on the transition edges or lines. Also, we use “0/0” to denote that the FSM is currently at state 0 and its output is 0, respectively.

Midterm 2024

For the above FSM, we use the unsigned number $(S[2]S[1]S[0])_2$ to represent its state and output. Please complete the state transition diagram below. Tips: This FSM has no input, and we do not put the transition condition on the transition edges or lines. Also, we use "0/0" to denote that the FSM is currently at state 0 and its output is 0, respectively.

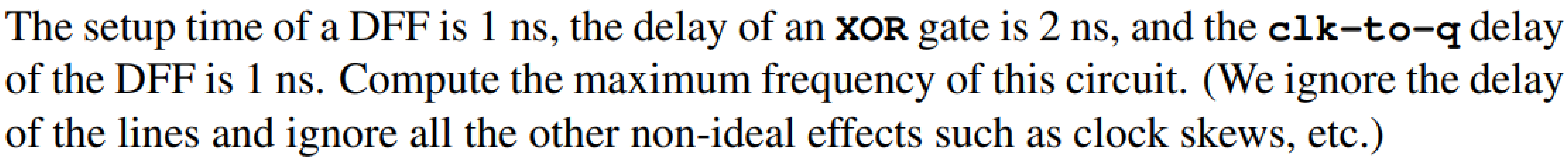


Midterm 2024

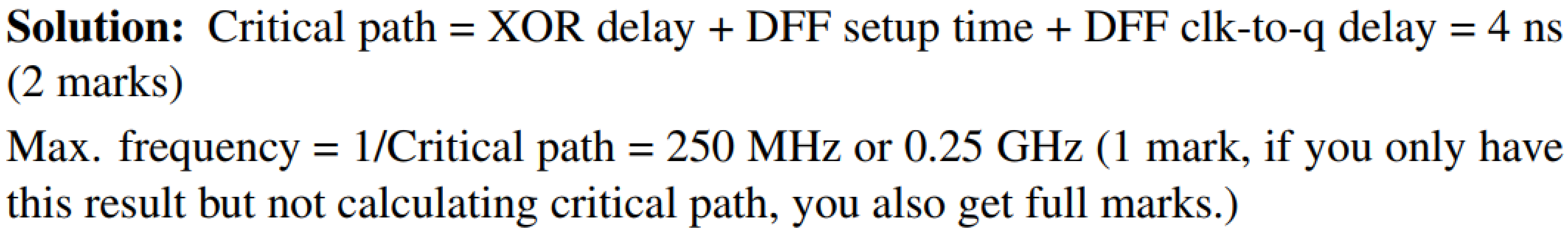


The setup time of a DFF is 1 ns, the delay of an **`XOR`** gate is 2 ns, and the **`clk-to-q`** delay of the DFF is 1 ns. Compute the maximum frequency of this circuit. (We ignore the delay of the lines and ignore all the other non-ideal effects such as clock skews, etc.)

Midterm 2024



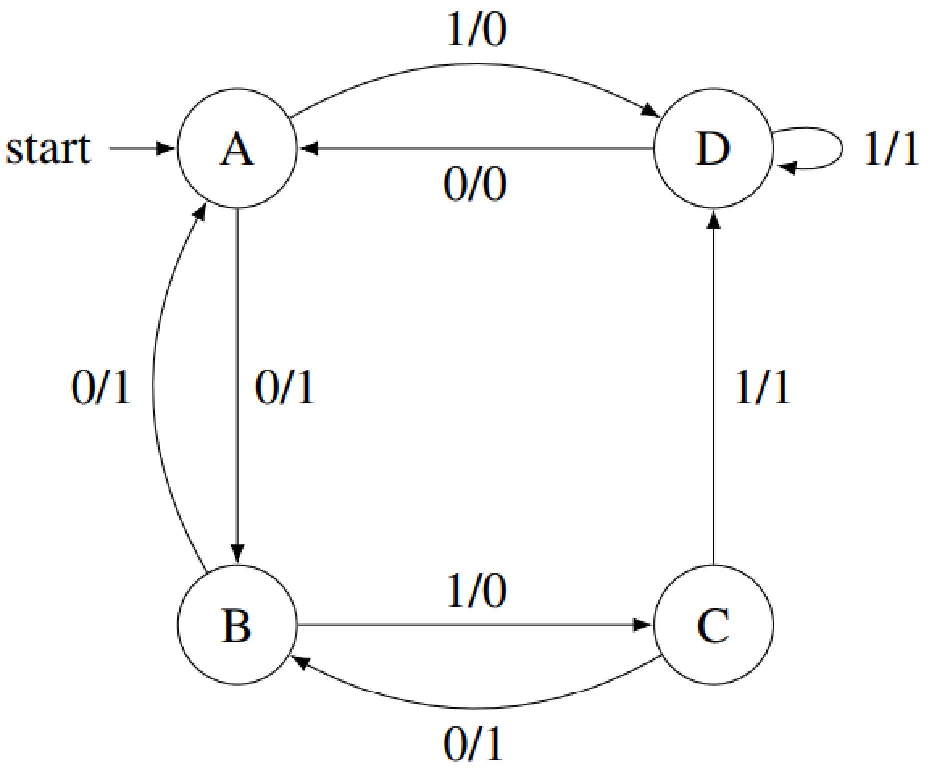
The setup time of a DFF is 1 ns, the delay of an **XOR** gate is 2 ns, and the **clk-to-q** delay of the DFF is 1 ns. Compute the maximum frequency of this circuit. (We ignore the delay of the lines and ignore all the other non-ideal effects such as clock skews, etc.)



Solution: Critical path = XOR delay + DFF setup time + DFF clk-to-q delay = 4 ns (2 marks)

Max. frequency = 1/Critical path = 250 MHz or 0.25 GHz (1 mark, if you only have this result but not calculating critical path, you also get full marks.)

Midterm 2023



Below shows a state transition diagram of a finite state machine (FSM) with 4 states. What is the type of the FSM?

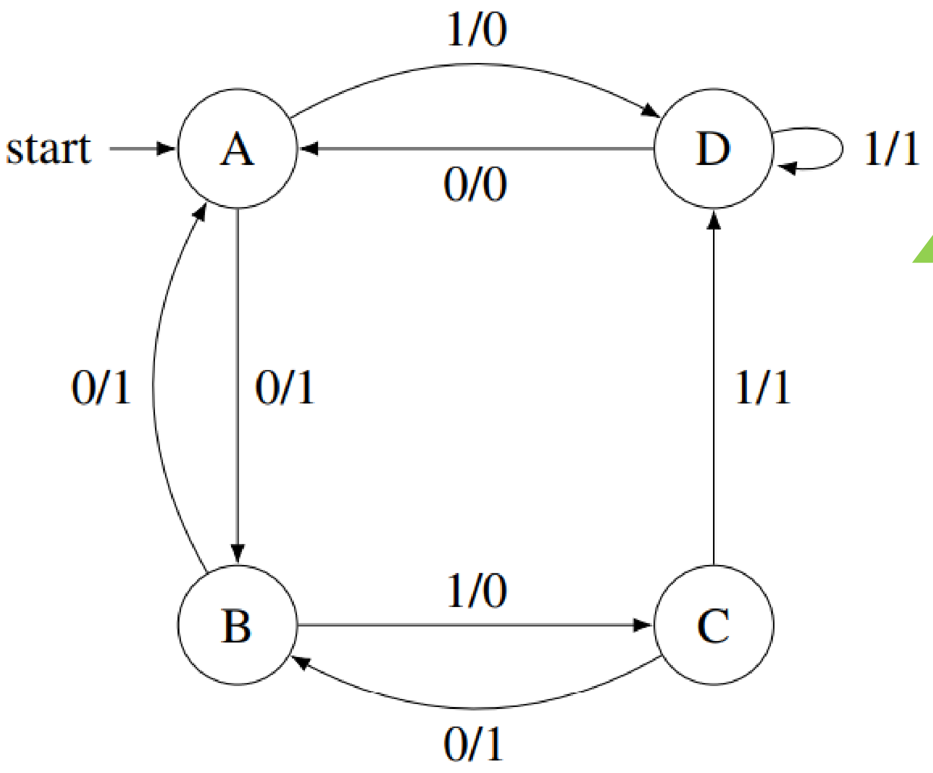
A. A Moore machine.

B. A Mealy machine.

Assume the input bit sequence to the FSM is `10011010`, the output is _

Which state does the FSM arrive at last?_

Midterm 2023



Below shows a state transition diagram of a finite state machine (FSM) with 4 states. What is the type of the FSM?

A. A Moore machine.

[✓ B. A Mealy machine.]

Assume the input bit sequence to the FSM is `10011010`, the output is

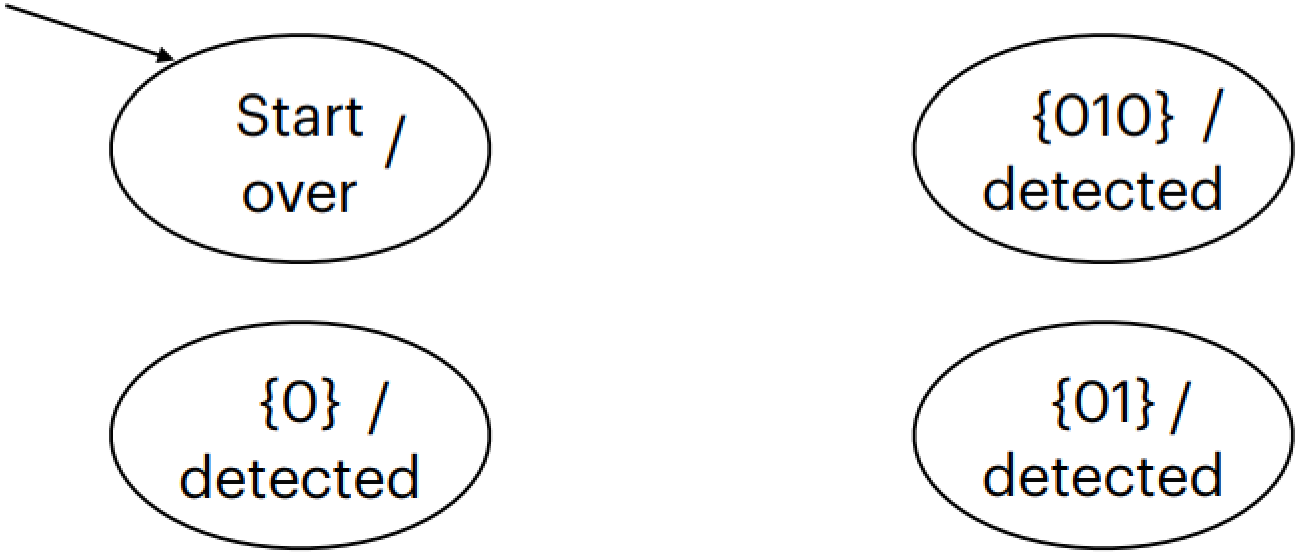
Solution: 00101000.

Which state does the FSM arrive at last?_

Solution: A.

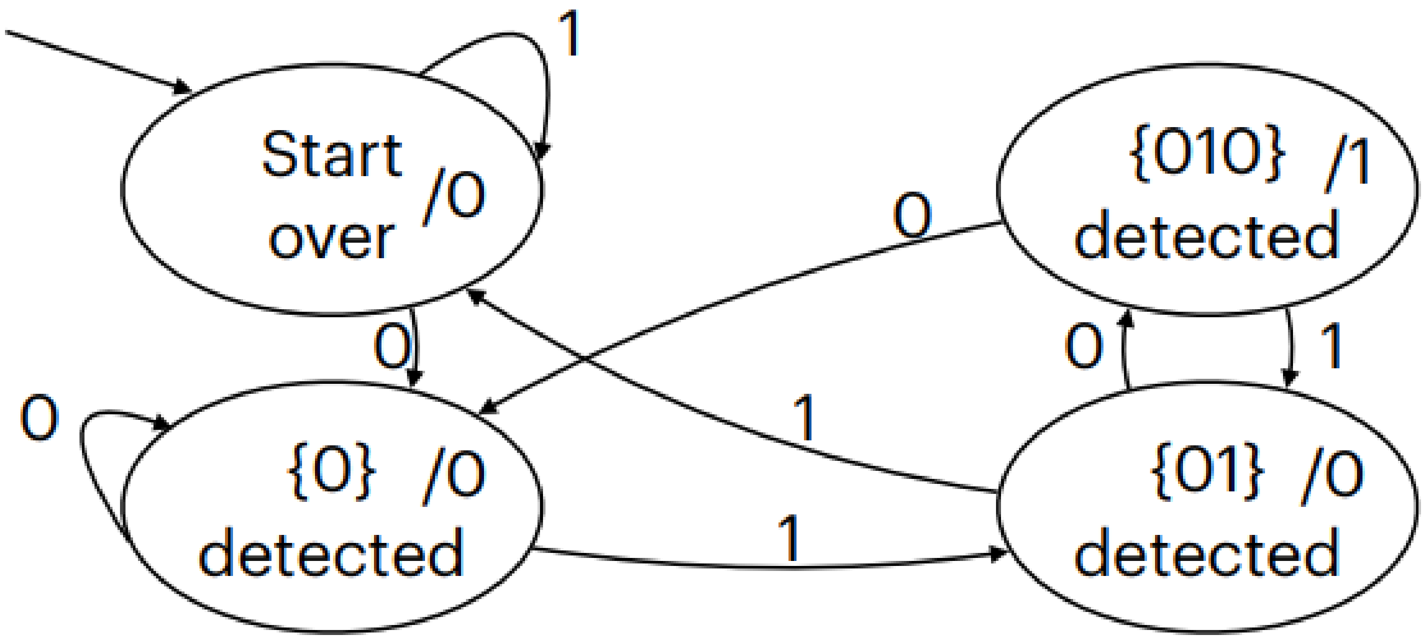
Midterm 2023

- Build a **Moore** FSM model to detect “010” pattern in a bit sequence (use overlapping, i.e., the tail 0 of “010” can be considered as the head 0 for the next detection). The states are given below. Please complete the state transition diagram by adding the transitions, transition conditions and output for each state. [4 points]

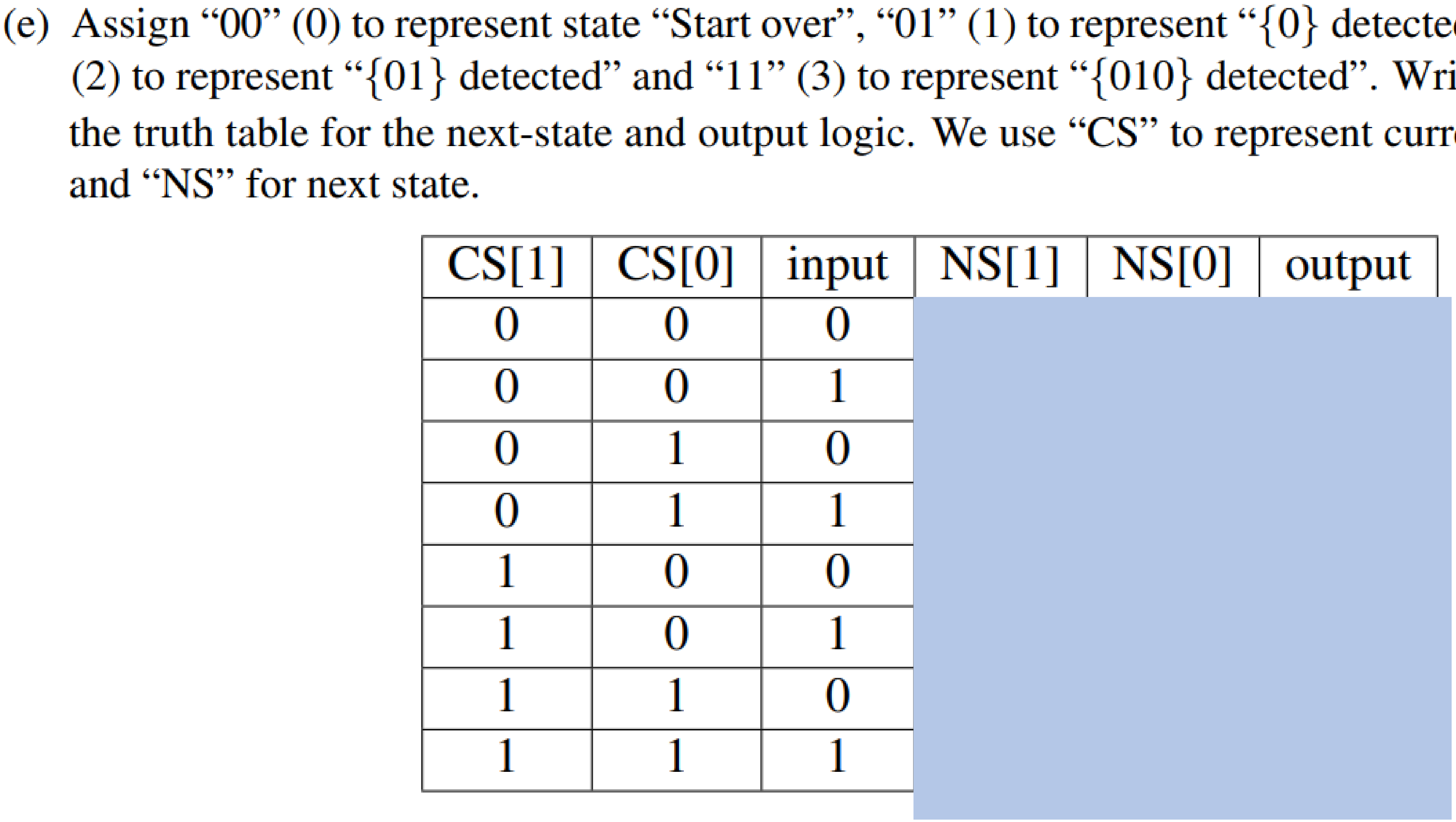


Midterm 2023

- Build a **Moore** FSM model to detect “010” pattern in a bit sequence (use overlapping, i.e., the tail 0 of “010” can be considered as the head 0 for the next detection). The states are given below. Please complete the state transition diagram by adding the transitions, transition conditions and output for each state. [4 points]



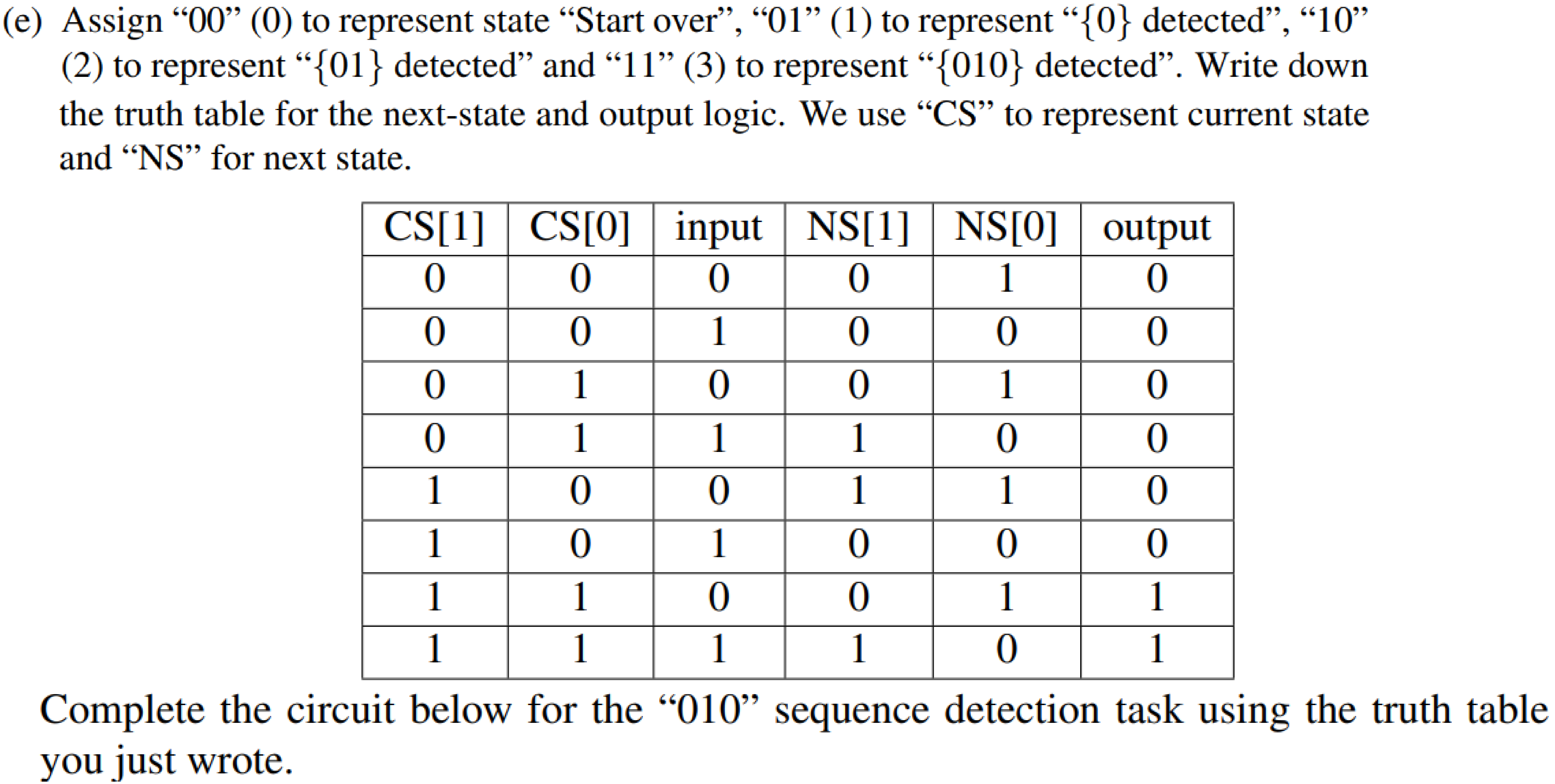
Midterm 2023



(e) Assign “00” (0) to represent state “Start over”, “01” (1) to represent “{0} detected”, “10” (2) to represent “{01} detected” and “11” (3) to represent “{010} detected”. Write down the truth table for the next-state and output logic. We use “CS” to represent current state and “NS” for next state.

CS[1]	CS[0]	input	NS[1]	NS[0]	output
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Midterm 2023



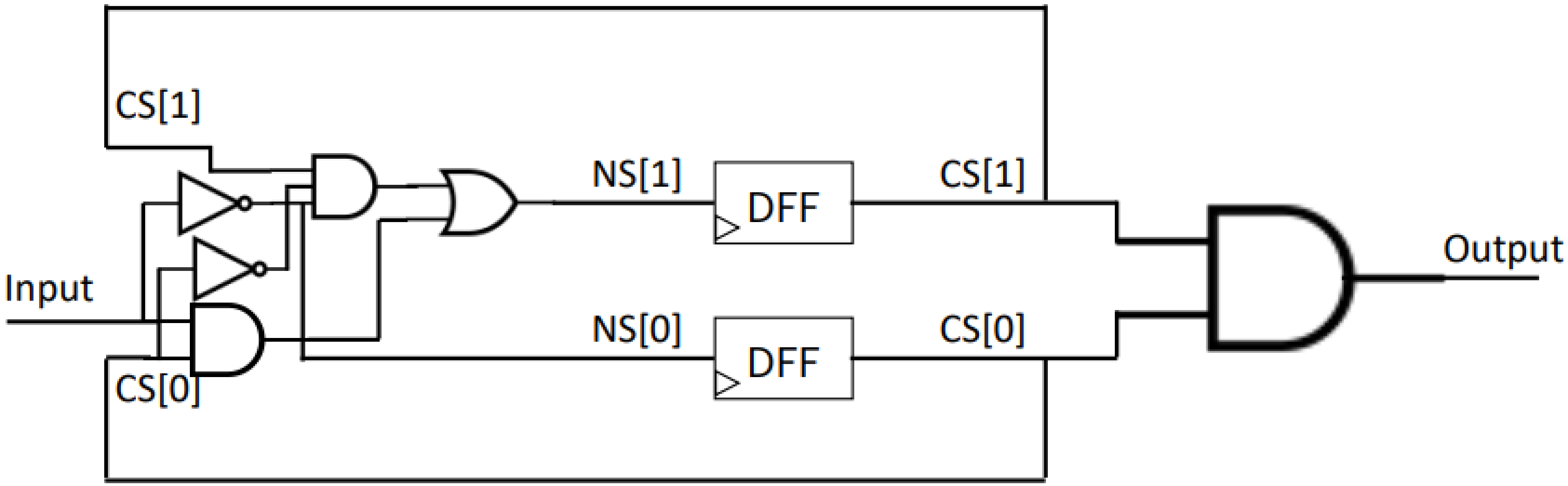
(e) Assign “00” (0) to represent state “Start over”, “01” (1) to represent “{0} detected”, “10” (2) to represent “{01} detected” and “11” (3) to represent “{010} detected”. Write down the truth table for the next-state and output logic. We use “CS” to represent current state and “NS” for next state.

CS[1]	CS[0]	input	NS[1]	NS[0]	output
0	0	0	0	1	0
0	0	1	0	0	0
0	1	0	0	1	0
0	1	1	1	0	0
1	0	0	1	1	0
1	0	1	0	0	0
1	1	0	0	1	1
1	1	1	1	0	1

Complete the circuit below for the “010” sequence detection task using the truth table you just wrote.

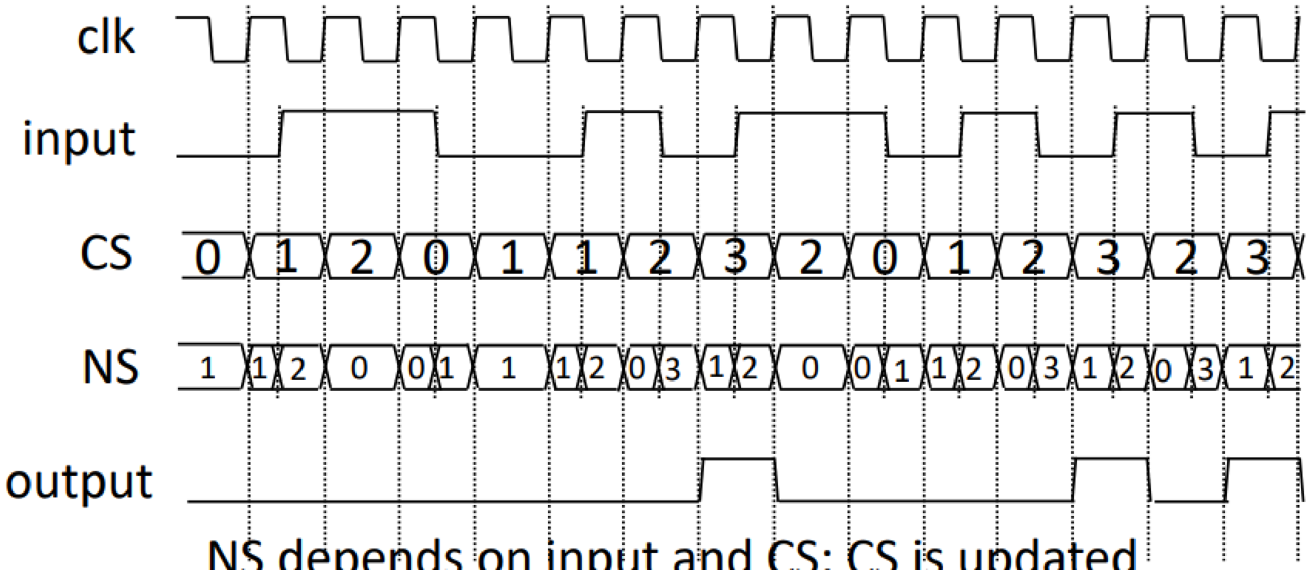
Midterm 2023

Complete the circuit below for the “010” sequence detection task using the truth table you just wrote.



Midterm 2023

(g) Draw the timing diagram given the clock signal and input below. We ignore the non-ideal effects, and integers (use signal grouping) are used to represent the states.



NS depends on input and CS; CS is updated every clock cycle according to NS; Output depends on CS, in Moore machine.

Solution:

Thanks!

Reference

CS110_2023_MidtermExam_I

CS110_2024_MidtermExam_I

EE115