

# Computer Architecture Homework 4

Spring 2020, March

## 1 Boolean Algebra

Simplify the following expressions step by step (as simple as possible).  
Please write the answer in Sum Of Product (SOP) form.

1.  $(A + B)(A + C)$

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

2.  $(A + \overline{B} + \overline{C})(A + \overline{B}C)$

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

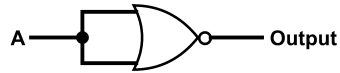
3.  $\overline{\overline{A + B\overline{C}} + D(\overline{E + F})}$

$$\begin{aligned} &= \overline{\overline{(A + B\overline{C})} \overline{(D(\overline{E + F}))}} \\ &= \overline{(A + B\overline{C}) (D(\overline{E + F}))} \\ &= \overline{(A + B\overline{C}) (\overline{D} + \overline{\overline{E + F}})} \\ &= \overline{(A + B\overline{C}) (\overline{D} + E + F)} \\ &= \overline{A\overline{D} + AE + AF + B\overline{C}\overline{D} + B\overline{C}E + B\overline{C}F} \end{aligned}$$

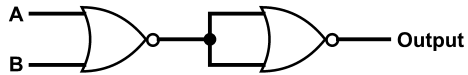
## 2 Logic Gates

### 1. Design Logic Gates

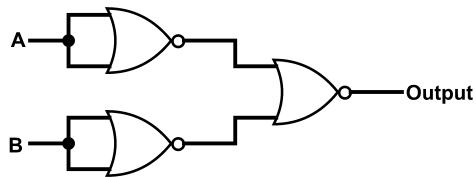
a. Create a NOT gate using only NOR gates.



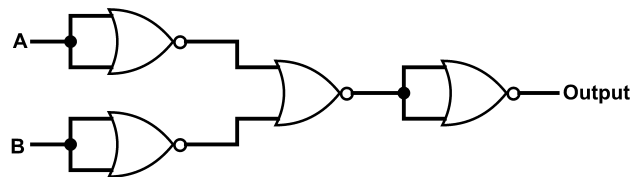
b. Create an OR gate using only NOR gates.



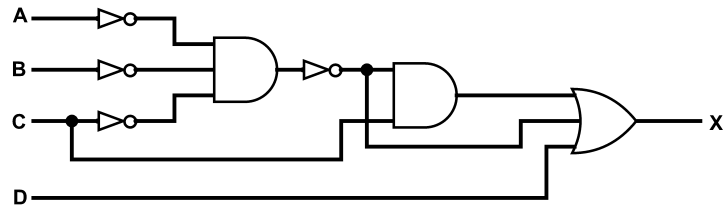
c. Create an AND gate using only NOR gates.



d. Create a NAND gate using only NOR gates.



2. The circuit shown below can be simplified. Please write the origin boolean expression of this circuit and simplify the expression step by step. Then draw the circuit according to the simplified boolean expression using the minimum number of two-input logic gates.

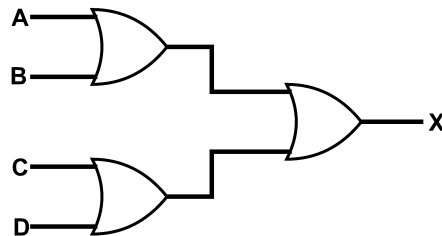


Solution:

The expression for the output of the circuit is

$$\begin{aligned}
 X &= (\overline{\overline{A} \overline{B} \overline{C}})C + \overline{\overline{A} \overline{B} \overline{C}} + D \\
 &= (\overline{\overline{A} + \overline{B} + \overline{C}})C + \overline{\overline{A} + \overline{B} + \overline{C}} + D \\
 &= AC + BC + CC + A + B + C + D \\
 &= AC + BC + C + A + B + \cancel{C} + D \\
 &= C(A + B + 1) + A + B + D \\
 &= A + B + C + D
 \end{aligned}$$

One possible solution for the circuit:

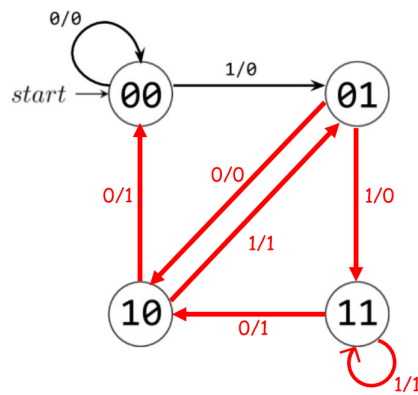


### 3 FSM and SDS

1. Write an FSM that takes in an n-bit binary number (starting with the MSB, ending with the LSB) and performs a **logical right shift by 2** on the input. For example, if our input is 0b01100, then our FSM should output 0b00011.

<b>Input (MSB → LSB)</b>	0	1	1	0	0
<b>Output</b>	0	0	0	1	1

Fill in the following FSM with the correct transitions and outputs. Format state changes as (input/output); we've done two for you. This is the **minimum** number of states; you may not add any more.



State 00 means ‘two most recent inputs are 00’, so an output from this state will always be 0.

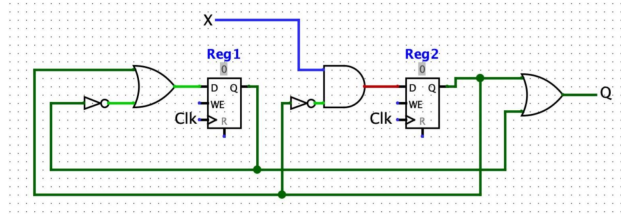
State 01 means ‘two most recent inputs are 01’, so an output from this state will always be 0.

State 10 means ‘two most recent inputs are 10’, so an output from this state will always be 1.

State 11 means ‘two most recent inputs are 11’, so an output from this state will always be 1.

Note that the output from any state is always the second most recent input for that state.

2. Consider the following circuit:



Assume the clock has a frequency of 50 MHz, all gates have a propagation delay of 6 ns, X changes 10ns after the rising edge of Clk, Reg1 and Reg2 have a CLK to Q time of 1 ns.

a. What is the **longest possible setup time** such that there are no setup time violations? Only giving a result will receive no point.

The clock period is  $\frac{1}{50 \cdot 10^6} s = 20 ns$ , which means that if X changes, it changes 10 ns after the clock positive edge.

Reg1 longest possible setup time: the path is the output of Reg1  $\rightarrow$  NOT  $\rightarrow$  OR, with a delay of 1 ns + 6 ns + 6 ns = 13 ns. So 20 - 13 = **7 ns**.

Reg2 longest possible setup time: the path is X changes  $\rightarrow$  AND, with a delay of 10 ns + 6 ns = 16 ns. So 20 - 16 = **4 ns**.

So longest setup time:  $\min(7 ns, 4 ns) = \mathbf{4 ns}$

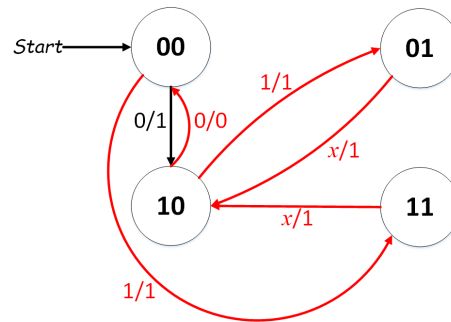
b. What is the **longest possible hold time** such that there are no hold time violations? Only giving a result will receive no point.

Reg1 longest possible hold time: the path is the output of Reg2  $\rightarrow$  OR, with a delay of 1 ns + 6 ns = **7 ns**.

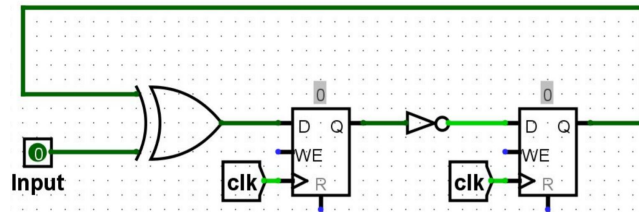
Reg2 longest possible hold time: the path is the output of Reg2  $\rightarrow$  NOT  $\rightarrow$  AND, with a delay of 1 ns + 6 ns + 6 ns = **13 ns**.

So longest hold time:  $\min(7 ns, 13 ns) = \mathbf{7 ns}$

c. Represent the circuit above using an equivalent FSM, where X is the input and Q is the output, with the state labels encoding Reg1Reg2 (e.g. "01" means Reg1=0 and Reg2=1). We did one transition already for you.



3. Assume **Input** comes from a register, and that there are no hold time violations. What's the fastest **frequency** you can run your clock for this circuit so that it executes correctly? Write your answer as a mathematical expression (you can also use  $\min()$ ,  $\max()$ ,  $\text{abs}()$ , and other simple operations if needed) using these variables:  $\mathbf{X}$ =XOR delay,  $\mathbf{N}$ =NOT delay,  $\mathbf{C}=\mathbf{t}_{\text{clk-to-Q}}$ ,  $\mathbf{S}=\mathbf{t}_{\text{setup}}$ ,  $\mathbf{H}=\mathbf{t}_{\text{hold}}$ .



**Solution:**

We want to check the longest path in the circuit.

There are three paths to consider:

- (1) Input  $\rightarrow$  XOR  $\rightarrow$  Register
- (2) Register  $\rightarrow$  XOR  $\rightarrow$  Register
- (3) Register  $\rightarrow$  NOT  $\rightarrow$  Register

Since input comes from a register, so (1) and (2) are equivalent.

The time it will take (2) to execute properly is  $\mathbf{t}_{\text{clk-to-Q}} + \mathbf{XOR\ delay} + \mathbf{t}_{\text{setup}}$ .

The time it will take (3) to execute properly is  $t_{clk-to-Q} + \text{NOT delay} + t_{setup}$ .  
We do not know which number is bigger between  $\mathbf{X}$  and  $\mathbf{N}$ , so we take the maximum.

So the answer is

$$\frac{1}{\max(C + X + S, C + N + S)} = \frac{1}{C + S + \max(X, N)}$$