Design of Digital Circuits (252-0028-00L), Spring 2019
Optional HW 2: Combinational Logic

Instructor: Prof. Onur Mutlu
TAs: Mohammed Alser, Can Firtina, Hasan Hassan, Juan Gomez Luna, Lois Orosa, Giray Yaglikci

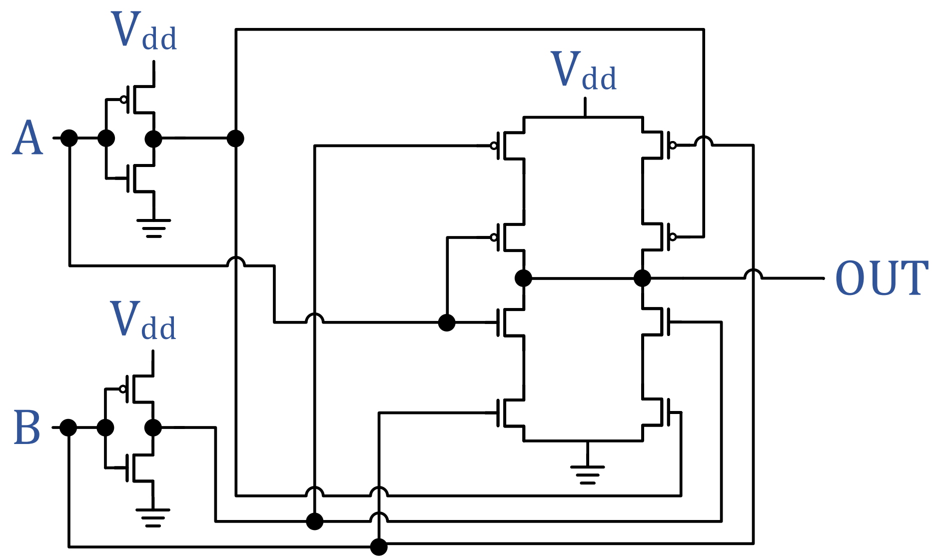
Released: Monday, Mar 11, 2019

1 Transistor-Level Circuit Design

In Lecture 4, we learned how to implement digital circuits using the CMOS technology (i.e., p-type and n-type MOS transistors). In this assignment, we ask you to schematically design circuits using CMOS transistors for the following logic gates:

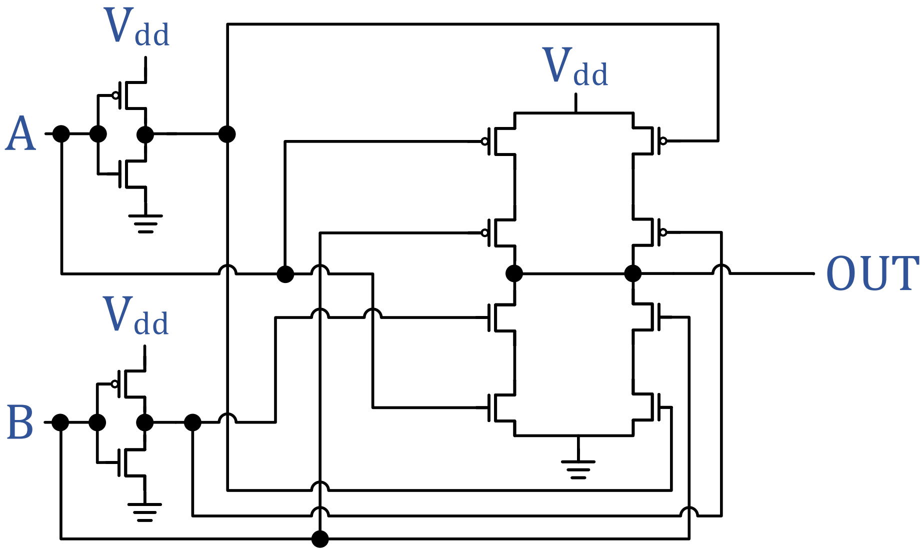
- Exclusive OR Gate (XOR)

Solution.



- Exclusive NOT OR Gate (XNOR)

Solution.

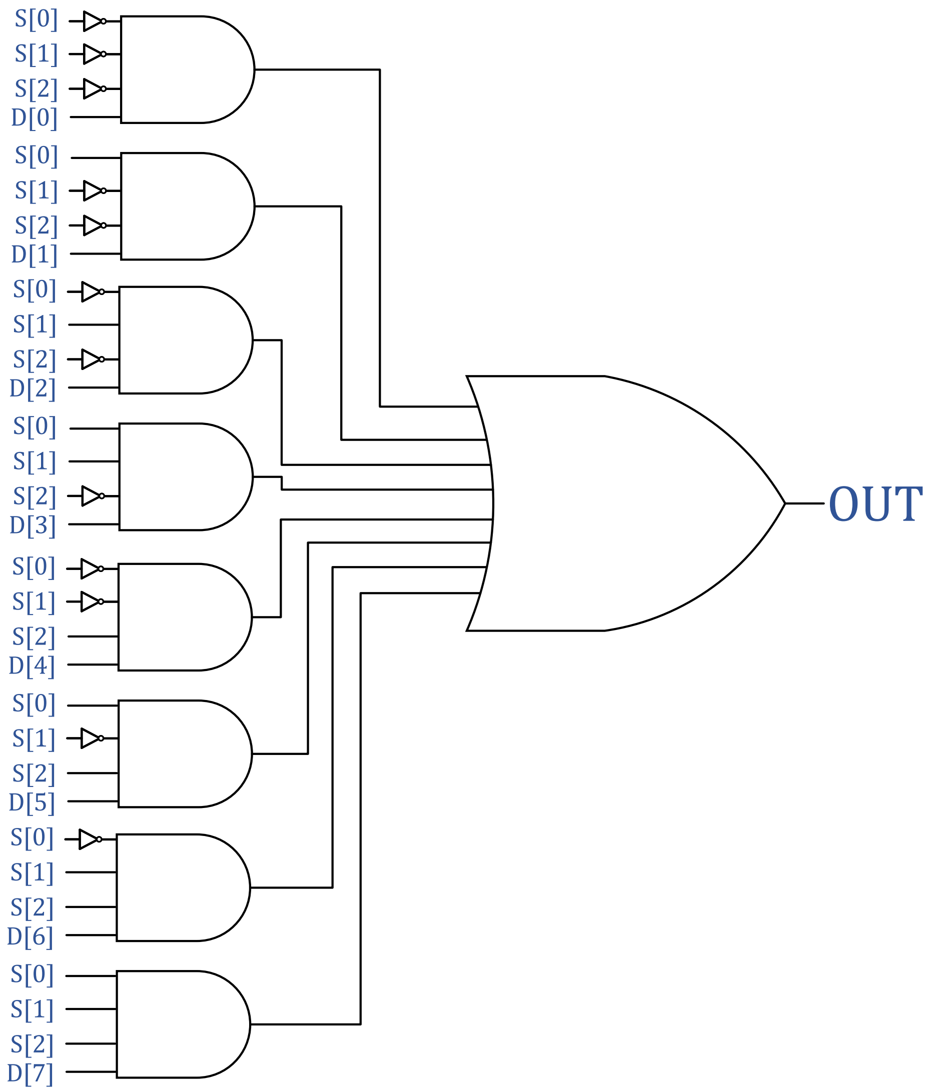


2 Multiplexer (MUX)

Draw the following schematics for an 8-input (8:1) MUX.

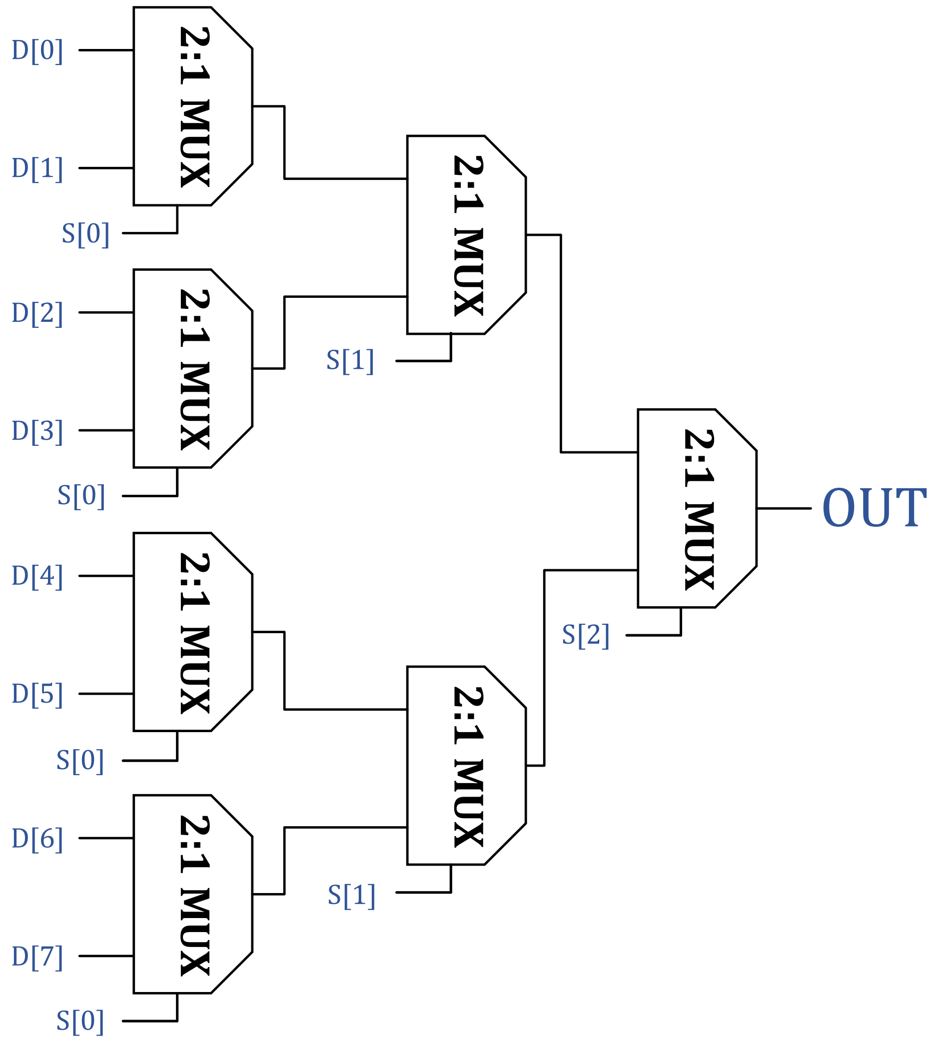
- Gate level: as a combination of basic AND, OR, NOT gates. Use as few gates as possible.

Solution.



- Module level: as a combination of 2-input (2:1) MUXes. Use as few 2-input MUXes as possible.

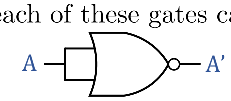
Solution.



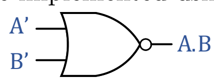
3 Logical Completeness

The set of {AND, OR, and NOT} gates is logically complete. We can build a circuit to carry out the specification of any truth table we wish, without using any other kind of gate. From Lecture 4, you know that the NOR gate by itself is also logically complete. Prove that you can build a circuit to carry out the specification of any truth table, by using only NOR gates.

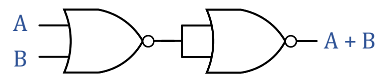
Solution.
To provide that the NOR gate is logically complete, it would be sufficient to show that it is possible to build the logically complete set of {AND, OR, and NOT} gates using only NOR gates. The figures below show how each of these gates can be implemented using NOR gates.



NOT



AND



OR

4 Boolean Logic and Truth Tables

In this question we ask you to derive the boolean equations for two 4-input logic functions, X and Y. Please use the truth table below to answer the following three questions.

Inputs				**Outputs**	
A_3	A_2	A_1	A_0	X	Y
0	0	0	0	1	0
0	0	0	1	1	0
0	0	1	0	1	0
0	0	1	1	1	0
0	1	0	0	1	0
0	1	0	1	1	1
0	1	1	0	1	0
0	1	1	1	0	0
1	0	0	0	1	0
1	0	0	1	1	0
1	0	1	0	1	1
1	0	1	1	1	0
1	1	0	0	1	0
1	1	0	1	1	0
1	1	1	0	0	0
1	1	1	1	0	0

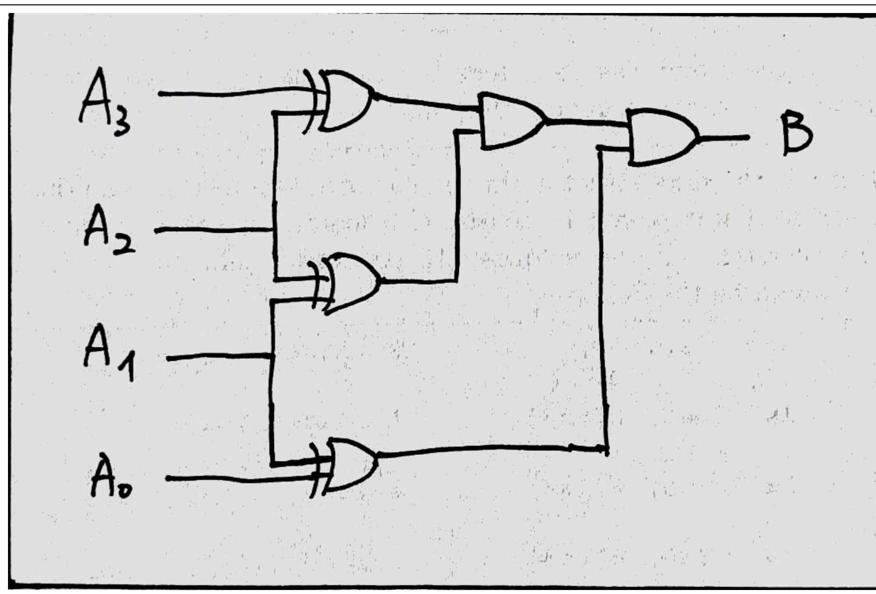
(a) The output X is *one* when the input does **not** contain 3 consecutive 1's in the word A_3, A_2, A_1, A_0. The output X is *zero*, otherwise. **Fill in the truth table above** and use the *product of sums* form to **write the corresponding boolean equation** for X. (*No simplification needed.*)

$$X = (A_3 + \overline{A_2} + \overline{A_1} + \overline{A_0}) \cdot (\overline{A_3} + \overline{A_2} + \overline{A_1} + A_0) \cdot (\overline{A_3} + \overline{A_2} + \overline{A_1} + \overline{A_0})$$

(b) The output Y is *one* when no two adjacent bits in the word A_3, A_2, A_1, A_0 are the same (e.g., if A_2 is 0 then A_3 and A_1 cannot be 0). The output Y is *zero*, otherwise (e.g., 0000). **Fill in the truth table above** and use the *sum of products* form to **write the corresponding boolean equation** for Y. (*No simplification needed.*)

$$Y = \overline{A_3} A_2 \overline{A_1} A_0 + A_3 \overline{A_2} A_1 \overline{A_0}$$

(c) Please represent the circuit of Y using *only* 2-input XOR and AND gates.



5 Boolean Algebra

(a) [5 points] Find the simplest sum-of-products representation of the following Boolean equation. Show your work step-by-step.

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

Answer: $F = A + B + \overline{C}$

Explanation:

$F = B + (A.\overline{A} + A.\overline{B} + A.\overline{C} + \overline{A}.\overline{C} + \overline{B}.\overline{C} + \overline{C}.\overline{C})$
$F = B + 0 + A.\overline{B} + \overline{C}.(A + \overline{A}) + \overline{B}.\overline{C} + \overline{C}$
$F = (B + A.\overline{B}) + \overline{C}.(A + \overline{A}) + (\overline{B}.\overline{C} + \overline{C})$
$F = (B + A) + \overline{C} + \overline{C}.(\overline{B} + 1)$
$F = A + B + \overline{C}$

(b) [5 points] Convert the following Boolean equation so that it only contains NAND operations. Show your work step-by-step.

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

Answer: $F = \overline{(\overline{(\overline{(A.A)}.\overline{(B.C)})}.C)}$

Explanation:

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

(c) [5 points] Using Boolean algebra, simplify the following min-terms: $\sum(3, 5, 7, 11, 13, 15)$

Show your work step-by-step.

Answer: $F = D.(B + C)$

Explanation:

$\{3, 5, 7, 11, 13, 15\} = \{0011, 0101, 0111, 1011, 1101, 1111\}$

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