EECS 16B Designing Information Devices and Systems II
Spring 2018 J. Roychowdhury and M. Maharbiz

Discussion 1A

1 Digit Bases

$(N)_p$ is used to indicate that the number N is expressed in base p. For example, $(N)_2$ means that we are working in base 2 and $(N)_{10}$ means N is expressed in base 10, or decimal digits. For example,

- $(245)_{10} = 2 \times 10^2 + 4 \times 10^1 + 5 \times 10^0$
- $(11010)_2 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$

2 Boolean Algebra

In Boolean algebra, *true* statements are denoted 1 and *false* statements are denoted 0. A Boolean function acts on a set of these Boolean values and outputs a set of Boolean values (usually just one). The most common Boolean operators used are **NOT**, **AND**, **OR**, and **XOR**.

2.1 NOT

NOT is a Boolean function that takes in one Boolean value and outputs its negation. Let x be a Boolean variable. **NOT**(x) is denoted as $\bar{x}$. The truth table of **NOT** is:

x	$\bar{x}$
0	1
1	0

2.2 AND

AND is a Boolean function that takes in two Boolean values and outputs 1 if both the values are true. Let x and y be two Boolean variables. **AND**(x,y) is denoted as $x \cdot y$. The truth table of **AND** is:

x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

2.3 OR

OR is a Boolean function that takes in two Boolean values and outputs 1 if at least one of the values is true. Let x and y be two Boolean variables. **OR**(x,y) is denoted as $x+y$. The truth table of **OR** is:

x	y	$x+y$
0	0	0
0	1	1
1	0	1
1	1	1

2.4 XOR

XOR is a Boolean function that takes in two Boolean values and outputs 1 if exactly one of the values is true. Let x and y be two Boolean variables. $\mathbf{XOR}(x,y)$ is denoted as $x \oplus y$. The truth table of **XOR** is:

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

3 De Morgan's Law

De Morgan's Law is stated as follows. Let x and y be two Boolean variables. Then,

$$\overline{x \cdot y} = \overline{x} + \overline{y}$$
$$\overline{x + y} = \overline{x} \cdot \overline{y}$$

4 Boolean Formulae and Natural Numbers

We can represent natural numbers with a Boolean function that evaluates to 1 if and only if its inputs match the binary representation of that numbers. For example, say we want to find a binary function f that returns true if the input is $(11)_{10}$. The first thing to note is that $(11)_{10} = (1011)_2$. Then, define $f(b_3, b_2, b_1, b_0)$, where b_3, b_2, b_1, b_0 are Boolean variables such that b_i represents the ith digit in binary representation, as follows.

$$f(b_3, b_2, b_1, b_0) = b_3 \cdot \overline{b_2} \cdot b_1 \cdot b_0$$

Note that f is true if the input is $(11)_{10}$ in base 2.

5 Transistor Introduction

Transistors (as presented in this course) are 3 terminal, voltage controlled switches. This means that, when a transistor is "on," it connects the Source (S) and Drain (D) terminals via a low resistance path (short circuit). When a transistor is "off," the Source and Drain terminals are disconnected (open circuit).

Two common types of transistors are NMOS and PMOS transistors. Their states (shorted or open) are determined by the voltage difference across the Gate (G) and Source (S) terminals, compared to a "threshold

voltage." Transistors are extremely useful in digital logic design since we can implement Boolean logic operators using switches.

Given that V_{tn} and V_{tp} are the threshold voltages for the NMOS and PMOS transistors, respectively, the figures below depict the closing conditions for these "switches." It's important to note that V_{tn} is generally positive and that V_{tp} is negative.

Figure 1: NMOS Transistor

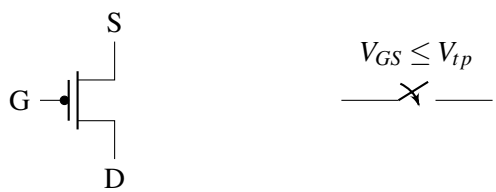

Figure 2: PMOS Transistor

Transistors can be strung together to perform boolean algebra. For example, the following circuit is called an "inverter" and represents a NOT gate.

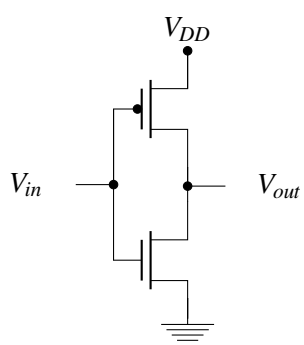

Figure 3: CMOS Inverter

When the input is high ($V_{in} \geq V_{tn}$, $V_{in} \geq V_{DD} + V_{tp}$), then the NMOS transistor is on, the PMOS transistor is off, and $V_{out} = 0$. When the input is low ($V_{in} \leq V_{tn}$, $V_{in} \leq V_{DD} + V_{tp}$), the NMOS transistor is off, the PMOS transistor is on, and $V_{out} = V_{DD}$. When working with digital circuits like the one above, we usually only consider the values of $V_{in} = 0, V_{DD}$. This yields the following truth table:

V_{in}	V_{out}	NMOS	PMOS
V_{DD}	0	on	off
0	V_{DD}	off	on

If you think of V_{DD} being a logical 1 and 0 V being a logical 0, we have just created the most elementary logical operation using transistor circuits!

5.1 Resistor Switch Model

In the real world, transistors don't actually behave as perfect switches. Transistors have a small amount of resistance in the on state. This can be represented by a resistor in our transistor switch model.

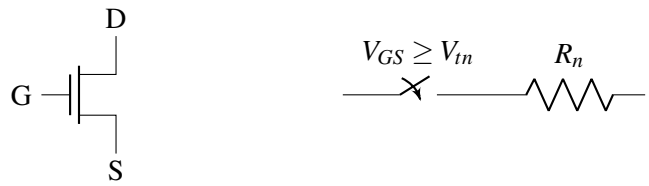

Figure 4: NMOS Transistor Resistor-switch model

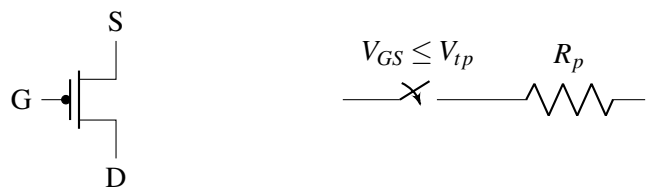

Figure 5: PMOS Transistor Resistor-switch model

1. Transistor Introduction

(a) Assume that the voltage range is from ground to V_{DD}. If the Source of an NMOS is connected to V_{DD}, would the switch ever close? What if it is connected to ground?

Answer:
If the Source is connected to V_{DD} and $V_{tn} \neq 0$, then the NMOS switch will never close. If Source is connected to Ground, then $V_G \geq V_{tn}$ is sufficient to close the switch. This is why in digital logic design, the source of a NMOS is usually connected to Ground, or there is a low resistance path from Source to Ground.

(b) Assume that the voltage range is from ground to V_{DD}. If the Source of a PMOS is connected to V_{DD}, would the switch ever turn on? What if it is connected to ground?

Answer:
If the Source is connected to Ground and $V_{tp} \neq 0$, then the PMOS switch will never close. If the Source is connected to V_{DD}, then $V_G - V_{DD} \leq V_{tp}$ is sufficient to close the switch. This is why in digital logic design, the Source of a PMOS is usually connected to V_{DD}, or there is a low resistance path from the Source to V_{DD}.

2. Digit Bases

(a) Express $(14)_{10}$ with binary encoding (or in base 2).

Answer: $(1110)_2$

(b) Given k binary digits, what is the largest number you can express?

Answer: $(2^k - 1)_{10}$

3. Boolean Formulae And Natural Numbers

(a) Express an integer, $(1)_{10}$, with a Boolean formula f, which includes only one Boolean variable B_0.

Answer: $f = B_0$

(b) Express an integer $(0)_{10}$ with a Boolean formula g, which includes only one Boolean variable B_0.

Answer: $g = \overline{B_0}$

(c) Express an integer $(19)_{10}$, with a Boolean formula h, with 5 Boolean variables, B_0, B_1, B_2, B_3 and B_4.

Answer: $h = B_4 \cdot \overline{B_3} \cdot \overline{B_2} \cdot B_1 \cdot B_0$

(d) Given a natural number $(N)_{10}$, how many Boolean variables do we need? How do we express it in terms of a Boolean formula?

Answer: $\lfloor \log_2 N \rfloor + 1$

4. De Morgan's Laws

(a) Write the truth table for $C = A \cdot B$.

Answer:

A	B	Out
0	0	0
0	1	0
1	0	0
1	1	1

(b) Flip all the Boolean values in the truth table you just made. What is the resulting Boolean expression represented by the flipped truth table? Do you need an **AND** to represent this?

Answer:

$\overline{A}$	$\overline{B}$	Out
1	1	1
1	0	1
0	1	1
0	0	0

You should see that given $\overline{A}$ and $\overline{B}$, the output will be the result of an **OR** operation.
From this, we see that

$$A \cdot B = \overline{\overline{A} + \overline{B}}.$$

This is one of the equality relations shown by De Morgan's Laws.

(c) Now, let's similarly convert $S = A \oplus B$. First, write out $A \oplus B$ using only **NOT**s, **OR**s, and **AND**s.

Answer:
There are many ways to write this. One way is

$$A \oplus B = (A \cdot \overline{B}) + (\overline{A} \cdot B).$$

(d) From the expression you just made, use the rule you learned in part (b) to remove the **AND**s.

Answer:

$$(A \cdot \overline{B}) + (\overline{A} \cdot B) = \overline{\overline{A} + B} + \overline{A + \overline{B}}$$

(e) Was there something special about **OR** vs. **AND**? Using only **NOT**s and **AND**s, rewrite the expression $S = A \oplus B$.

Answer:

$$(A \cdot \overline{B}) + (\overline{A} \cdot B) = \overline{\overline{A \cdot \overline{B}} \cdot \overline{\overline{A} \cdot B}}$$

Contributors:

- Siddharth Iyer.
- Saavan Patel.
- Deborah Soung.