UNITED STATES AIR FORCE ACADEMY

ECE 281

Lesson 3 Notes

Objectives:

- Understand the basic logic gates and their functions

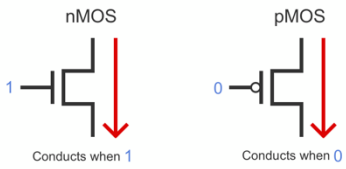
Lesson Notes:

Review: In CS 210 we introduced a few key logic terms:
- **Not** - Essentially the inverse of the input
- **Or** - If any input is true, the output is true
- **And** - The output is true only if all of the inputs are true
- **XOR** - Exclusive Or - The output is true exclusively if some but not all of the inputs are true

We will now extend these concepts and lay the foundation for all digital logic.

Logical Gates:

At the simplest level, logical gates are made of transistors. Transistors can be thought of as a simple switch that either passes a logical true or false. In this class we won't focus much at this level, but you will see this a lot more in later classes in the ECE department

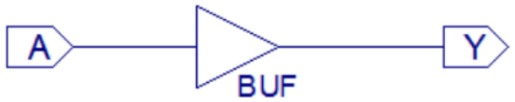
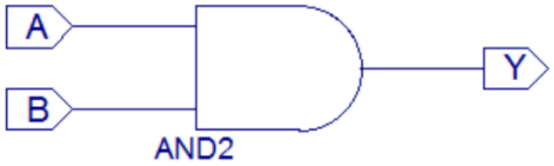


Digital Abstraction

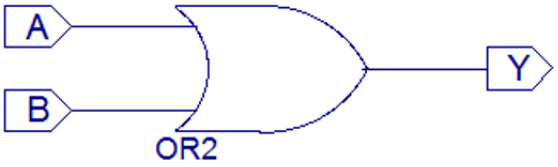
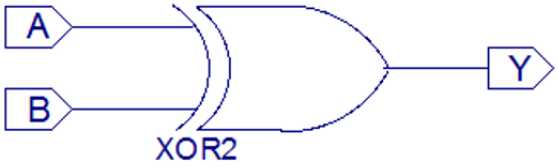
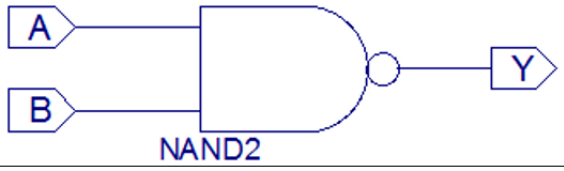
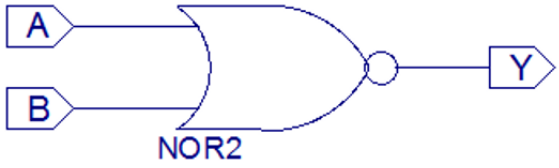
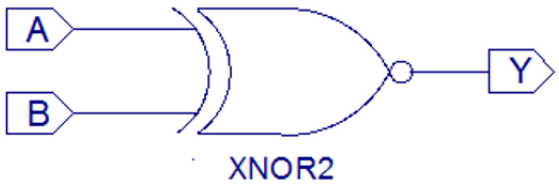
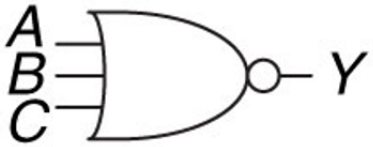
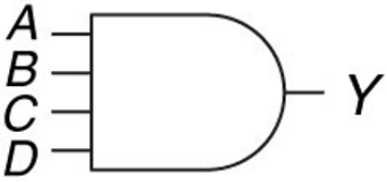
From our reading - "A ***digital circuit*** has voltages that are treated as either high or low, and is typically built as a connection of switches."

However, depending on the components being used, the voltage level for high and low may vary. It will be very important to eventually understand the voltage ranges that correspond with input and output logic high and low levels.

Logic Gates

Type and Equation	Schematic symbol	Truth table	VHDL
Buffer (BUF)		A | Y 0 1	y <= a;
Inverter (INV or NOT)		A | Y 0 1	y <= not a;
AND		A B | Y 0 0 0 1 1 0 1 1	y <= a and b;

Logic Gates Continued

OR		A B Y 0 0 0 1 1 0 1 1	y <= a or b;
XOR		A B Y 0 0 0 1 1 0 1 1	y <= a xor b;
NAND		A B Y 0 0 0 1 1 0 1 1	y <= a nand b; y <= not (a and b);
NOR		A B Y 0 0 0 1 1 0 1 1	y <= a nor b; y <= not (a or b);
XNOR		A B Y 0 0 0 1 1 0 1 1	y <= a xnor b; y <= not (a xor b);
Three Input NOR		ABC Y 000 Else	y <= not (a or b or c);
Four Input AND		ABCD Y 1111 Else	y <= not (a and b and c and d);