Lesson 1

Digital Electronics Review Part 1

Mohd Saufy Rohmad
Technical Trainer and Consultant

Gates

- Lets start with the concept of gates/logic gates.
- Think of this blocks as the fundamental digital blocks that you will use to design a digital circuit.
- The basic three gates are known as NOT, AND and OR.
- They can be used to construct other types of gates.
- NOT Gate: NOT gate simply inverts the input.
- The NOT gate takes only one input that is a single bit, either 0 or 1.
- If the input of 0, it gives an output of 1 and vice-versa.
- The symbol of a NOT gate is the following:

Gates

- AND Gate: This gate takes two input bits.
- The output of an AND gate always zero until all the input values are 1.
- We will use something called "truth table" to understand how a gate behaves.
- A truth table shows how a logic circuit's output responds to various combinations of the inputs.
- All permutations of the inputs are listed on the left, and the output of the circuit is listed on the right.
- The desired output can be achieved by a right combination of bits at the input of the logic gates.
- The truth table for the AND gate is the following:

Gates

- You can see from the table that when the input A is 0 and B is 0, the output is 0.
- The output remains zero as long as both A and B are 1.
- The truth table is very helpful to visualise how gates behave.
- OR Gate: The output of OR gate will be 1 (high) if any of the inputs are 1. The truth table of the OR gate is given below:

INPUT		OUTPUT
A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

INPUT		OUTPUT
A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

Gates

- We can create other logic gates like NAND (combination of NOT and AND), NOR, XOR, XNOR by combining the basic three logic gates.
- You can use "Boolean Algebra" and "Karnaugh Maps" to simplify the combination of gates.
- The interested readers can just Google these terms and know about more.
- We are not going to write gate level VHDL code, thats why we will not go into detail about this topic.

Combinational Logic

- We use gates to design a logic block. There can be two types of logics.

  1. Combinational logic

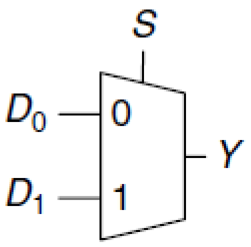
  2. Sequential logic

  Sequential logic has memory elements.

  Combinational logic has no memory elements. We give a few examples of combinational logic in this section.

Multiplexer

- Multiplexers are among the most commonly used combinational circuits.
- They choose an output from among several possible inputs based on the value of a select signal.
- A multiplexer is sometimes affectionately called a mux.
- A simple 2:1 multiplexer figure looks like below.



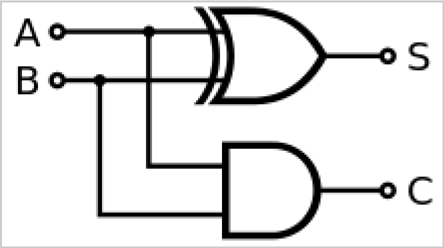
Multiplexer

- The below figure shows the schematic and truth table for a 2:1 multiplexer with two data inputs, D0 and D1, a select input, S, and one output, Y.
- The multiplexer chooses between the two data inputs based on the select: if S .=0, Y .=D0, and if S .=1, Y .=D1.
- S is also called a control signal because it controls what the multiplexer does.

S	D_1	D_0	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Adder

- In electronics the adder is a digital circuit that perform addition of numbers.
- They are important for numerical data processing.
- There are mainly two types of adder are used.
- They are Half adder and Full adder.
- Half Adder: It can add two binary bits.
- The below figure is the simplest construction of half adder.



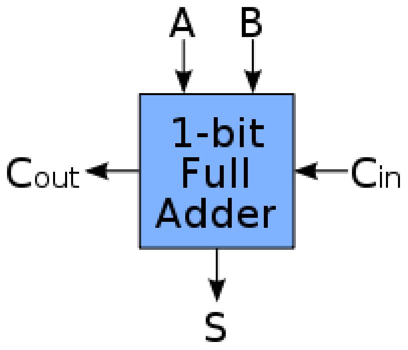
Adder

- The half adder adds two input bits and generates a carry and sum, which are the two outputs of a half adder.
- The input variables of a half adder are called the augend and addend bits.
- The output variables are the sum and carry.
- The truth table for the half adder is:

Inputs		Outputs	
A	*B*	*C*	*S*
0	0	0	0
1	0	0	1
0	1	0	1
1	1	1	0

Adder

- Full Adder: A full adder adds binary numbers and accounts for values carried in as well as out.
- A one-bit full adder adds three one-bit numbers, often written as A, B, and Cin;
- A and B are the operands, and Cin is a bit carried in from the previous less significant stage.



Adder

- The circuit produces a two-bit output, output carry and sum typically represented by the signals Cout and S, where

  Sum =2 x Cout + S

- And the truth table is

Inputs			Outputs	
A	B	C_{in}	C_{out}	S
0	0	0	0	0
1	0	0	0	1
0	1	0	0	1
1	1	0	1	0
0	0	1	0	1
1	0	1	1	0
0	1	1	1	0
1	1	1	1	1