LEVEL 1

KENNY NISANTH

October 2023

1 Introduction

At this level, I have learned about logic gates. Logic gates are fundamental building blocks in digital electronics that perform logical operations on one or more binary inputs to produce a binary output. These gates are essential in designing and creating complex digital circuits, such as microprocessors and memory units.

logic gates

AND gate: An AND gate is a fundamental digital logic gate that performs a logical AND operation. It has two or more input signals and produces a single output signal only if all of its inputs are high (binary 1). In other words, the output is "1" if and only if all input signals are simultaneously true. The AND gate is a building block for more complex digital circuits and is essential in various applications, such as arithmetic and data processing in computers. Its symbol is typically represented as a triangle with inputs on the left side and an output on the right.**Y=A.B**

OR gate: An OR gate is another fundamental digital logic gate that performs a logical OR operation. It has two or more input signals and produces a single output signal if at least one of its inputs is high (binary 1). In other words, the output is "1" if any of the input signals is true. The OR gate is a key component in digital circuit design, allowing the creation of diverse logical functions. Its symbol is often represented as a curved shape with inputs on the left side and an output on the right. The OR gate is widely used in various applications, including decision-making circuits and Boolean algebra expressions.**Y= A +B**

NOT gate: A NOT gate, also known as an inverter, is a fundamental digital logic gate with a single input and one output. It reverses the input signal, producing an output that is the opposite (complementary) of the input. If the input is high (1), the output is low (0), and vice versa. The NOT gate is represented by a triangle pointing right with a small circle at its tip. It plays a key role in digital circuit design, serving to negate or invert binary signals. ***Y=A'***

NAND gate: A NAND gate is a basic digital logic gate with two or more inputs and one output. It performs the opposite function of an AND gate followed by a NOT gate. In other words, a NAND gate produces a low output (0) only if all of its inputs are high (1), and for all other input combinations, it produces a high output (1). The symbol for a NAND gate is typically an AND gate symbol followed by a small circle at its output. NAND gates are essential building blocks in digital circuitry and are widely used in various applications, contributing to the complexity and versatility of logical operations in electronic systems. $\boldsymbol{Y=A+B}$

NOR gate: A NOR gate is a fundamental digital logic gate with two or more inputs and one output. It performs the opposite function of an OR gate followed by a NOT gate. In simpler terms, a NOR gate produces a high output (1) only if all of its inputs are low (0), and for any other combination of inputs, it produces a low output (0). The symbol for a NOR gate is often represented as an OR gate symbol with a small circle at its output. NOR gates are crucial components in digital circuit design, contributing to the implementation of various logical functions and providing versatility in electronic systems. $\boldsymbol{Y=AB}$

XNOR gate: An XNOR gate, also known as an equivalence gate, is a digital logic gate with two inputs and one output. It produces a high output (1) only if both of its inputs are the same (either both 0 or both 1). In other words, it outputs 1 for equal inputs and 0 for different inputs. The XNOR gate is essentially a combination of an XOR (exclusive OR) gate followed by a NOT gate. $\boldsymbol{Y=AB}$

XOR gate: An XOR gate, or exclusive OR gate, is a digital logic gate that has two inputs and one output. It produces a high output (1) if the number of true inputs is odd. In other words, the output is true when the inputs are different. The XOR gate is commonly used in digital circuit design and binary arithmetic. $\boldsymbol{Y=AB}$

2 Pre - Requisite

- Verilog HDL Language
- • Truth table and expression of logic gates

3 Tools/Requirements

Xiling Vivado To simulate and for observing the behavior of different logic gates we need the Xiling Vivado tool. Vivado is a software for synthesis and analysis of hardware description language designs, superseding Xilinx ISE with additional features for system-on-a-chip development and high-level synthesis.

4 What have I done?

I have written Verilog code for all logic gates in verVerilog HDL language by using modules. basic syntax of module: example, I have written all codes according to the given module syntax in Xilinx Vivado and performed a simulation to observe the functionalities of logic gates according to given input signals. Also, obtained RTL Schematics for each logic gate, and for further use I am making a collection of each logic gate code on Github under project "FPGAspeaks" .

5 For more information

https://github.com/KENNYNISATH/FPGAspeaksGIT HUB