VLSI SYSTEMS AND ARCHITECTURE

2021-22

Lab-1

Implementation of Majority Circuit using Gate Level Modeling

By Dr. Sanjay Vidhyadharan

Hardware Description Language

Hardware Description Language

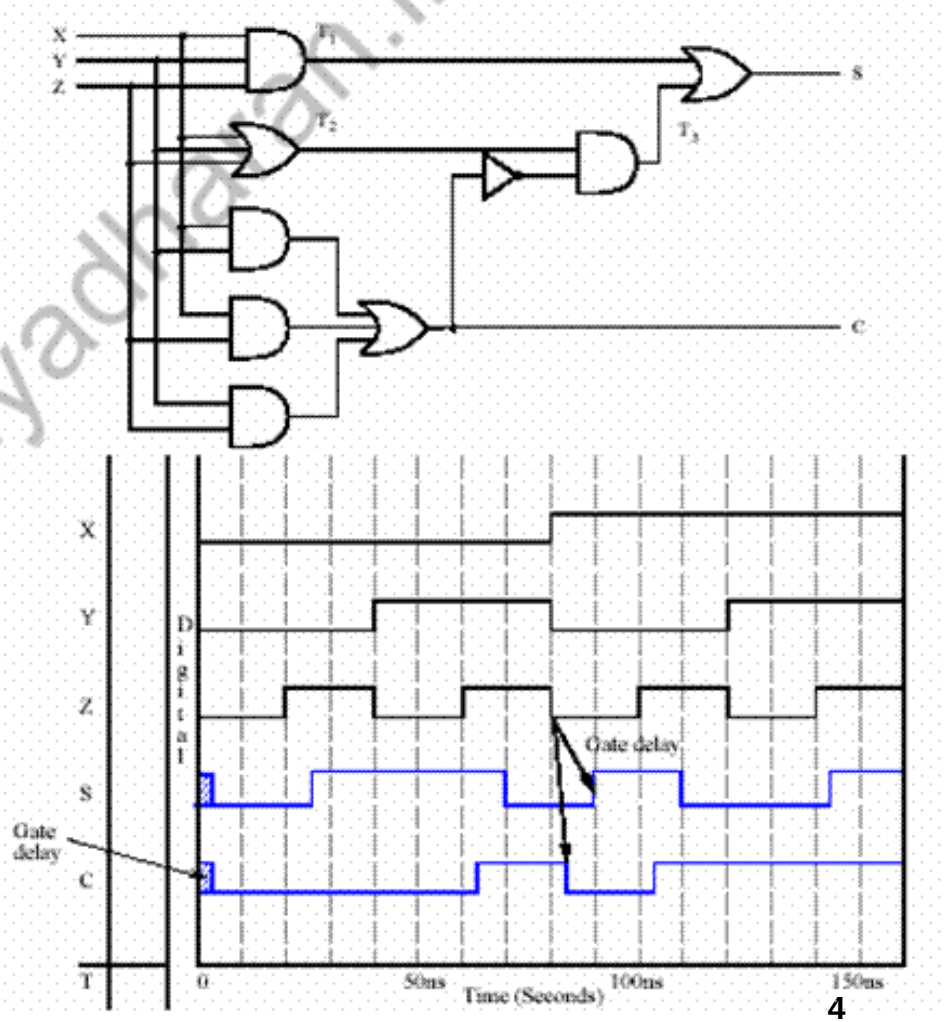
Logic Simulation

- A simulator interprets the HDL description and produces a readable output, such as a timing diagram, that predicts how the hardware will behave before its is actually fabricated.
- Simulation allows the detection of functional errors in a design without having to physically create the circuit.
- The stimulus that tests the functionality of the design is called a test bench.
- To simulate a digital system
  - Design is first described in HDL
  - Verified by simulating the design and checking it with a test bench which is also written in HDL.

Hardware Description Language

Logic Simulation

- Logic simulation is a fast, accurate method of analyzing a circuit to see its waveforms



Hardware Description Language

- There are two standard HDL's that are supported by IEEE.
  - **VHDL** (*Very-High-Speed Integrated Circuits Hardware Description Language*) - Sometimes referred to as VHSIC HDL, this was developed from an initiative by US. Dept. of Defense.
  - **Verilog HDL** – developed by Cadence Data systems and later transferred to a consortium called *Open Verilog International* (OVI).

Verilog

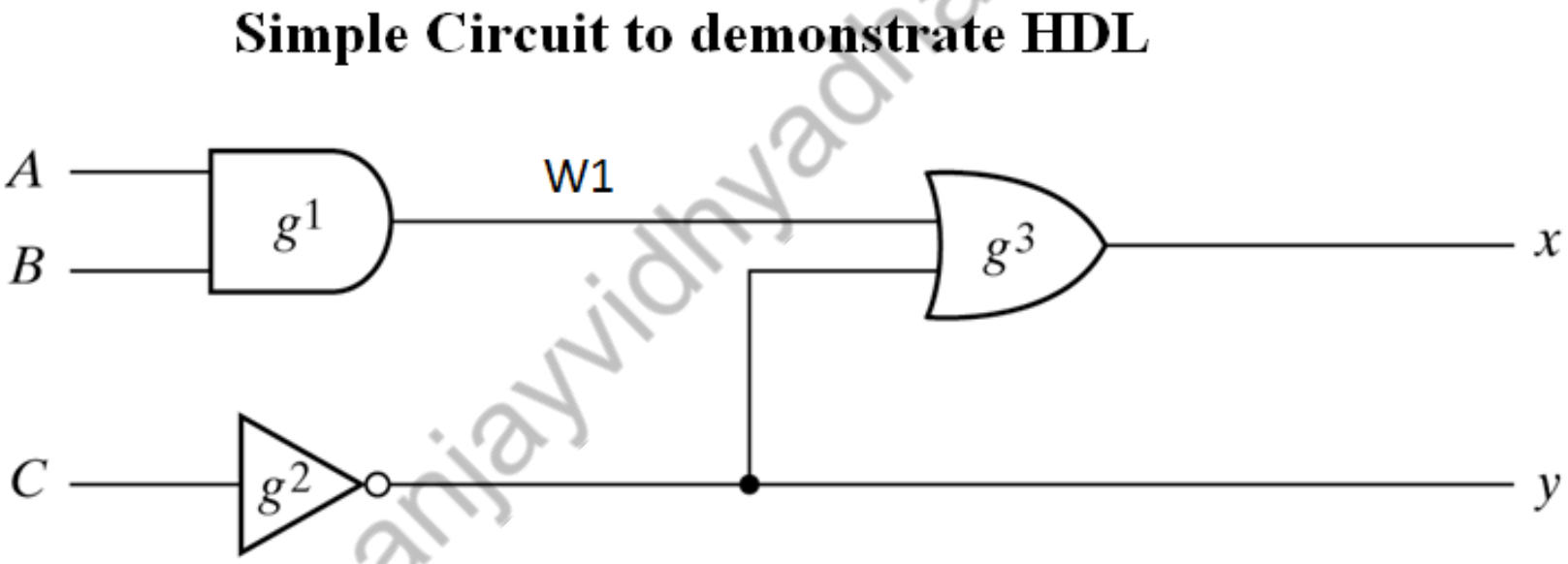
- Verilog HDL has a syntax that describes precisely the legal constructs that can be used in the language.
- It uses about 100 keywords pre-defined, lowercase, identifiers that define the language constructs.
- Example of keywords: *module*, *endmodule*, *input*, *output* *wire*, *and*, *or*, *not* , etc.,
- Any text between two slashes (//) and the end of line is interpreted as a comment.
- Blank spaces are ignored and names are case sensitive.

Link for Xilinx Download

https://www.xilinx.com/member/forms/download/xef.html?filename=Xilinx_ISE_14.7_Win10_14.7_VM_0213_1.zip

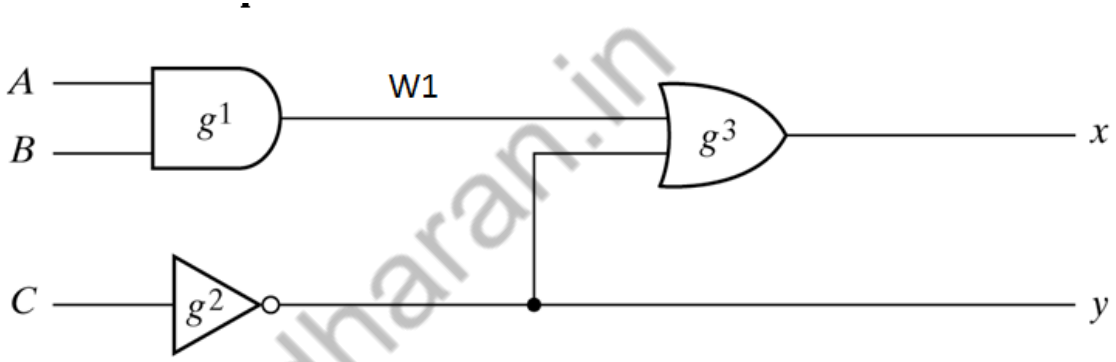
Simple Circuit for Demonstration

Gate Level Modelling



Simple Circuit for Demonstration

Gate Level Modelling



Verilog Code

```
module   Simple_circuit (input A, input B, input C, output x, output y);
wire      w1;

and       g1 (w1,A,B); // and gate instance
not       g2 (y,C);
or        g3 (x,w1,y);

endmodule
```

Problem Definition

Problem 1: Implement the Majority Circuit

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

$$F = AB + BC + CA + ABC$$

$$F = AB + BC + CA$$

Thank you