100 Days of RTL

DAY 3

Gate Level Modelling

Achutha Aswin Naick

Introduction

Gate-level modelling in Verilog represents digital circuits using predefined logic gate primitives such as AND, OR, NOT, NAND, NOR, XOR, and XNOR. This modelling technique closely resembles actual hardware implementation at the logic gate level. In this project, basic logic gates were implemented using gate-level modelling in Verilog. A corresponding testbench was developed to simulate the circuit and verify its correctness.

Implementation Details

The logic gate module, logic_gates, takes two input signals (a and b) and produces outputs corresponding to the seven basic logic operations. Unlike structural and behavioural modelling, gate-level modeling employs built-in gate primitives such as ‘and’, ‘or’, ‘not’, ‘nand’, ‘nor’, ‘xor’, and ‘xnor’ to define logic functions. These primitives directly represent logic components without the need for assign statements or procedural blocks.

The testbench module, logic_gate_tb, is used to verify the functionality of the logic_gates module. The testbench instantiates the design under test (DUT) and applies different input combinations with time delays (#10). The outputs are declared as wire since they are driven by the DUT.

Code Explanation

The main module and testbench include:

- **Input and Output Declaration:** Inputs (a and b) are declared as input types, while outputs (and_gate, or_gate, etc.) are declared as output wire since they are connected to predefined gate primitives.
- **Gate Primitives**: Standard Verilog gate primitives (and, or, not, nand, nor, xor, xnor) are used to create logic functions.

- **Instantiation in Testbench:** The DUT is instantiated with named mapping to connect signals.
- **Simulation Execution**: The initial block runs only once and assigns different values to the inputs over time to simulate different conditions.

Simulation and Results

The testbench applies four input combinations (00, 01, 10, 11) at 10-time unit intervals. The expected outputs for each logic operation are observed using waveform simulation. The results confirm that the gate-level model correctly implements the logic gates.

Conclusion

This project successfully demonstrates the implementation of logic gates using gate-level modelling in Verilog. The use of gate primitives highlights how digital circuits are constructed at the fundamental logic level.

Future Enhancements

Future enhancements could include additional gates and more complex combinational logic circuits.