100 Days of RTL

DAY 1

Behavioural Modelling

Achutha Aswin Naick

Introduction

Behavioural modelling is a high-level approach in Verilog that describes the functionality of a digital circuit without detailing its structural implementation. In this project, basic logic gates (AND, OR, NOT, NAND, NOR, XOR, and XNOR) were implemented using behavioural 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. The module uses an always block (always@(*)), ensuring that whenever a or b changes, the operations inside the block execute. The outputs are declared as reg because they are assigned inside a procedural block. Boolean operations are performed inside the always block to determine the values of the outputs.

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 reg since they are assigned within an always block.
- **Procedural Block (always@(*)):** Ensures that any change in inputs triggers re-evaluation of outputs.
- **Boolean Operations:** The bitwise operators (&, |, ~, ^) are used to define the logical relationships.
- **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, starting from t=0. The expected outputs for each logic operation are observed using waveform simulation. The results confirm that the behavioural model correctly implements the logic gates.

Conclusion

This project successfully demonstrates the implementation of logic gates using behavioural modelling in Verilog. The use of procedural blocks and testbenches highlights the importance of simulation in verifying digital designs before hardware implementation.

Future Enhancements

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