

Day 2 Topic: Half Adder and Full Adder in Verilog Using Cadence (Xcelium)

1 Half Adder

Performs addition of 2 bits.

- Inputs: A, B
- Outputs: Sum, Carry
- Equations:
 - $\text{Sum} = A \oplus B$
 - $\text{Carry} = A \cdot B$

◆ 1.1) Design Code

```
ha.v

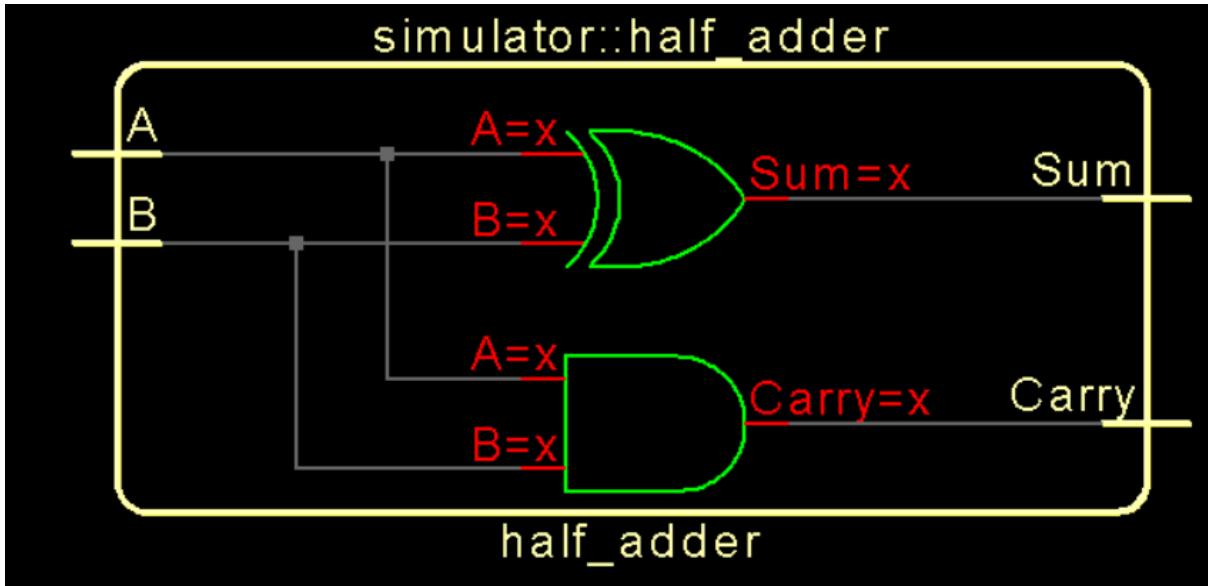
// Half Adder Design
module half_adder (
    input A, B,           // Two 1-bit inputs
    output Sum, Carry    // Outputs: Sum and Carry
);
    assign Sum = A ^ B;   // XOR operation gives Sum
    assign Carry = A & B; // AND operation gives Carry
endmodule
```

◆ 1.2) Test Bench Code

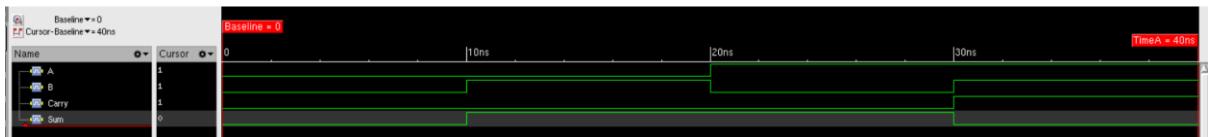
```
ha.v

// Testbench for Half Adder
module tb_half_adder;
    reg A, B;           // Testbench inputs
    wire Sum, Carry;   // Outputs to observe
    // Instantiate Design Under Test (DUT)
    half_adder HA (A, B, Sum, Carry);
    initial begin
        $display("A B | Sum Carry");
        // Apply all input combinations
        A = 0; B = 0; #10;
        $display("%b %b | %b %b", A, B, Sum, Carry);
        A = 0; B = 1; #10;
        $display("%b %b | %b %b", A, B, Sum, Carry);
        A = 1; B = 0; #10;
        $display("%b %b | %b %b", A, B, Sum, Carry);
        A = 1; B = 1; #10;
        $display("%b %b | %b %b", A, B, Sum, Carry);
        $finish; // End simulation
    end
endmodule
```

◆ 1.3) Schematic



◆ 1.4) Wave Forms



2 Full Adder

Performs addition of 3 bits (`A, B, Cin`).

- Inputs: `A, B, Cin`
- Outputs: `Sum, Carry`
- Equations:
 - $\text{Sum} = A \oplus B \oplus \text{Cin}$
 - $\text{Carry} = (A \cdot B) + (\text{Cin} \cdot (A \oplus B))$

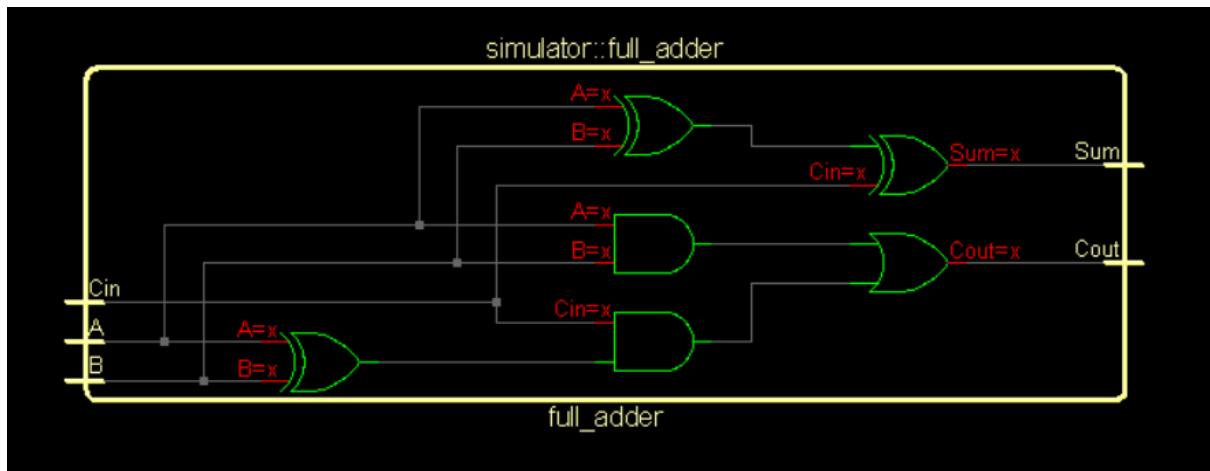
◆ 2.1) Design Code

```
fa.v
// Full Adder Design
module full_adder (
    input A, B, Cin,
    output Sum, Cout
);
    assign Sum = A ^ B ^ Cin;           // XOR for Sum
    assign Cout = (A & B) | (Cin & (A ^ B)); // Carry logic
endmodule
```

◆ 2.2) Test Bench Code

```
fa_tb.v
// Testbench for Full Adder
module tb_full_adder;
    reg A, B, Cin;          // Inputs
    wire Sum, Cout;         // Outputs
    // Instantiate DUT
    full_adder FA (A, B, Cin, Sum, Cout);
    initial begin
        $display("A B Cin | Sum Cout");
        $display("-----");
        // Apply all input combinations
        A=0; B=0; Cin=0; #10;
        $display("%b %b %b | %b %b", A, B, Cin, Sum, Cout);
        A=0; B=1; Cin=0; #10;
        $display("%b %b %b | %b %b", A, B, Cin, Sum, Cout);
        A=1; B=0; Cin=0; #10;
        $display("%b %b %b | %b %b", A, B, Cin, Sum, Cout);
        A=1; B=1; Cin=0; #10;
        $display("%b %b %b | %b %b", A, B, Cin, Sum, Cout);
        A=0; B=0; Cin=1; #10;
        $display("%b %b %b | %b %b", A, B, Cin, Sum, Cout);
        A=0; B=1; Cin=1; #10;
        $display("%b %b %b | %b %b", A, B, Cin, Sum, Cout);
        A=1; B=0; Cin=1; #10;
        $display("%b %b %b | %b %b", A, B, Cin, Sum, Cout);
        A=1; B=1; Cin=1; #10;
        $display("%b %b %b | %b %b", A, B, Cin, Sum, Cout);
        $finish;
    end
endmodule
```

◆ 2.3) Schematic



◆ 2.4) Wave Forms



✓ Day 2 Summary:

- Implemented **Half Adder + Full Adder**.
- Wrote and explained **Verilog + Testbench** step by step.