Binary Coded Decimal (BCD) Adder

Presented by – Aditya Ankitkumar Mistry
Enrolment – 240280763001
ME - VLSI Design

Agenda

Introduction

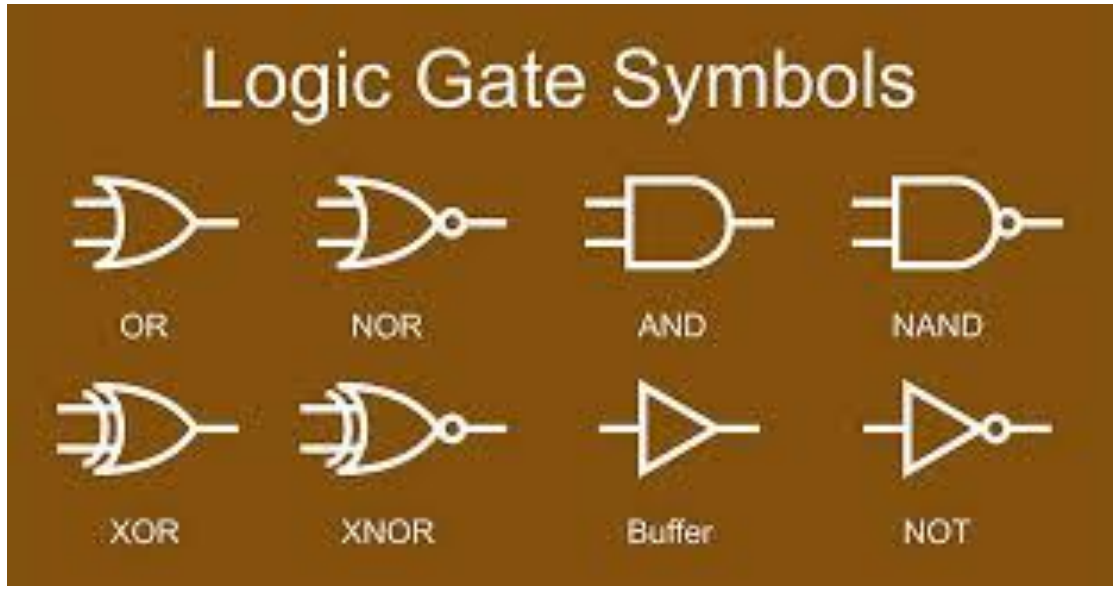
Verilog Code

Testbench

Simulation Output

Layout

Advantages and Disadvantages



What is BCD?

- **BCD** stands for binary coded decimal. It is used to perform the addition of BCD numbers. A BCD digit can have any of ten possible four-bit representations. Suppose, we have two 4-bit numbers A and B. The value of A and B can vary from 0 (0000 in binary) to 9 (1001 in binary) because we are considering decimal numbers.

- The output will vary from 0 to 18 if we are not considering the carry from the previous sum. But if we are considering the carry, then the maximum value of output will be 19 (i.e. 9+9+1 = 19). When we are simply adding A and B, then we get the binary sum. Here, to get the output in BCD form, we will use BCD Adder.

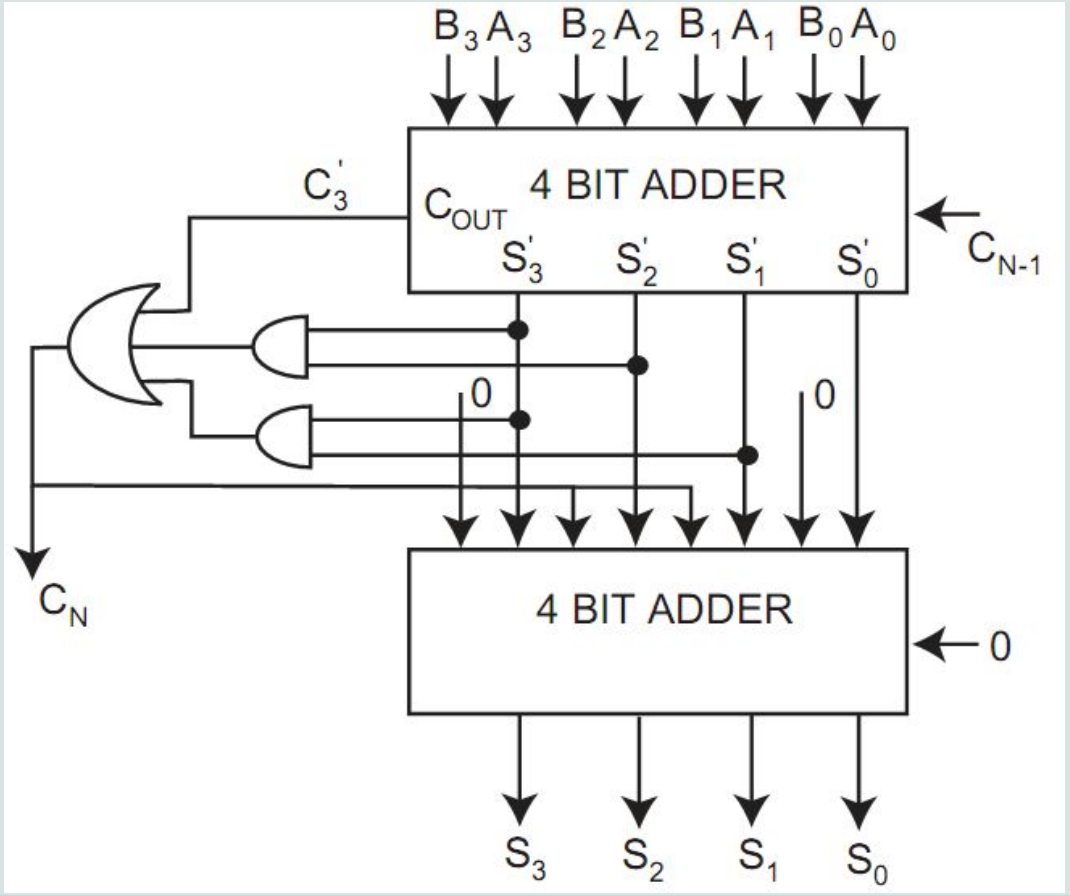
What is BCD Adder?

- A **BCD Adder** adds two **Binary-Coded Decimal (BCD)** numbers.
- Each BCD digit is represented using **4-bit binary** (0000 to 1001 for 0-9).
- It consists of **four full-adders** in cascade.
- If the sum exceeds **9 (1001_2)**, an extra **0110_2 (6 in decimal)** is added for correction.
- Ensures the output remains a valid BCD number.
- Used in **calculators, digital clocks, and financial applications**

Steps to Design a BCD Adder?

- **Find Number of Digits :** Find out how many Extended digits of BCD the adder should support.
  A BCD digit requires 4 bits.
- **Adder Structure :** The full adder connected in series could be selected as the general adder structure.
  Any extra full-adder shall be incremented by one BCD digit addition plus a carry from the previous stage.
- **Full-Adder Circuit Implementation :** Implement a full-adder circuit capable of adding two 4-bit BCD digits with a carry-in. The full adder shall output a sum bit and a carry-out bit
- **Interconnect the Full-Adders :** Now, full-adders are connected in series with each other; at this point, carry-out from each stage will be given to carry-in of the next higher order stage.
- **Provide BCD Correction :** The logic is implemented detecting whether the sum of two BCD digits is greater than 9. In the case of such, 0110 is added to the sum, and carry propagates to the next higher order stage.
- **Test the BCD Adder :** The BCD adder functionality needs to be checked with the application of different BCD numbers to its input for the correctness of addition and correction handling.

Architecture of BCD Adder



Components Include-

- Two 4-Bit Adders
- Correction Logic Detection using Logic Gates
- Inputs ($A_3A_2A_1A_0$ and $B_3B_2B_1B_0$) and Cin.
- Outputs ($S_3S_2S_1S_0$) and Cout.

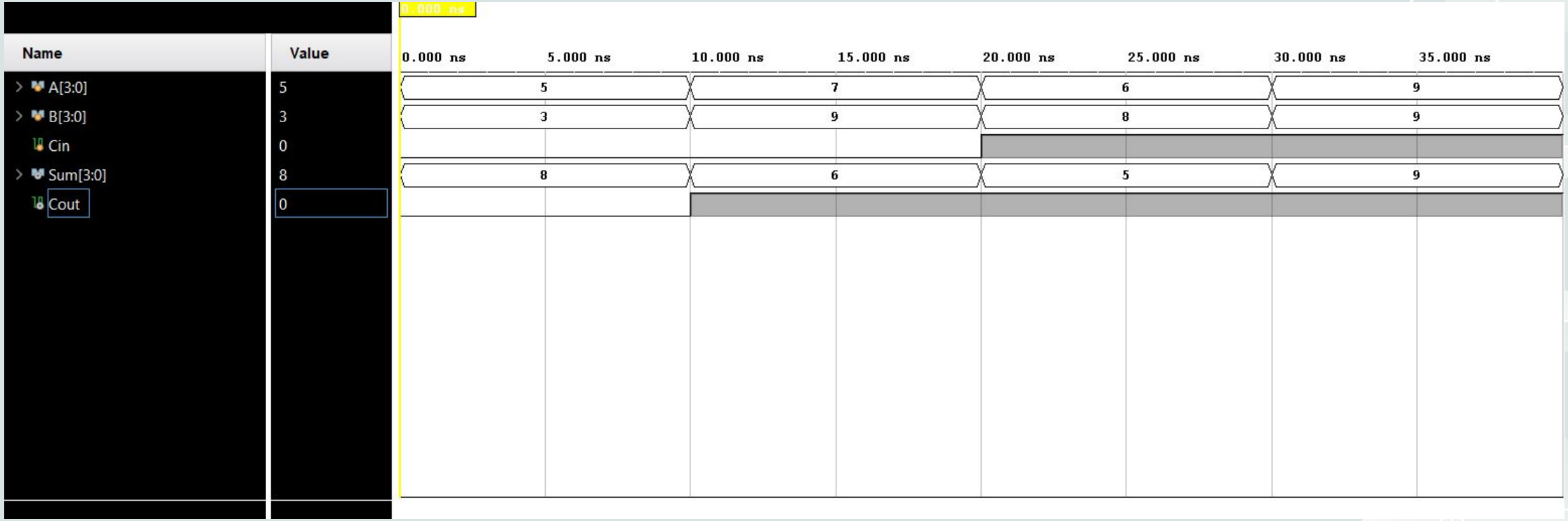
Verilog Code for 4-Bit BCD Adder

```
module BCD_Adder_4bit(
    input [3:0] A,    // First BCD digit
    input [3:0] B,    // Second BCD digit
    input Cin,        // Carry input
    output [3:0] Sum, // BCD Sum
    output Cout       // Carry output
);
    wire [4:0] temp_sum; // 5-bit sum including carry
    wire [4:0] adjusted_sum; // Adjusted sum after BCD correction
    wire correction_needed; // Condition for BCD correction

    assign temp_sum = A + B + Cin;
    assign correction_needed = (temp_sum > 9) ? 1'b1 : 1'b0;
    assign adjusted_sum = correction_needed ? (temp_sum + 6) : temp_sum;

    assign Sum = adjusted_sum[3:0];
    assign Cout = adjusted_sum[4];
endmodule
```

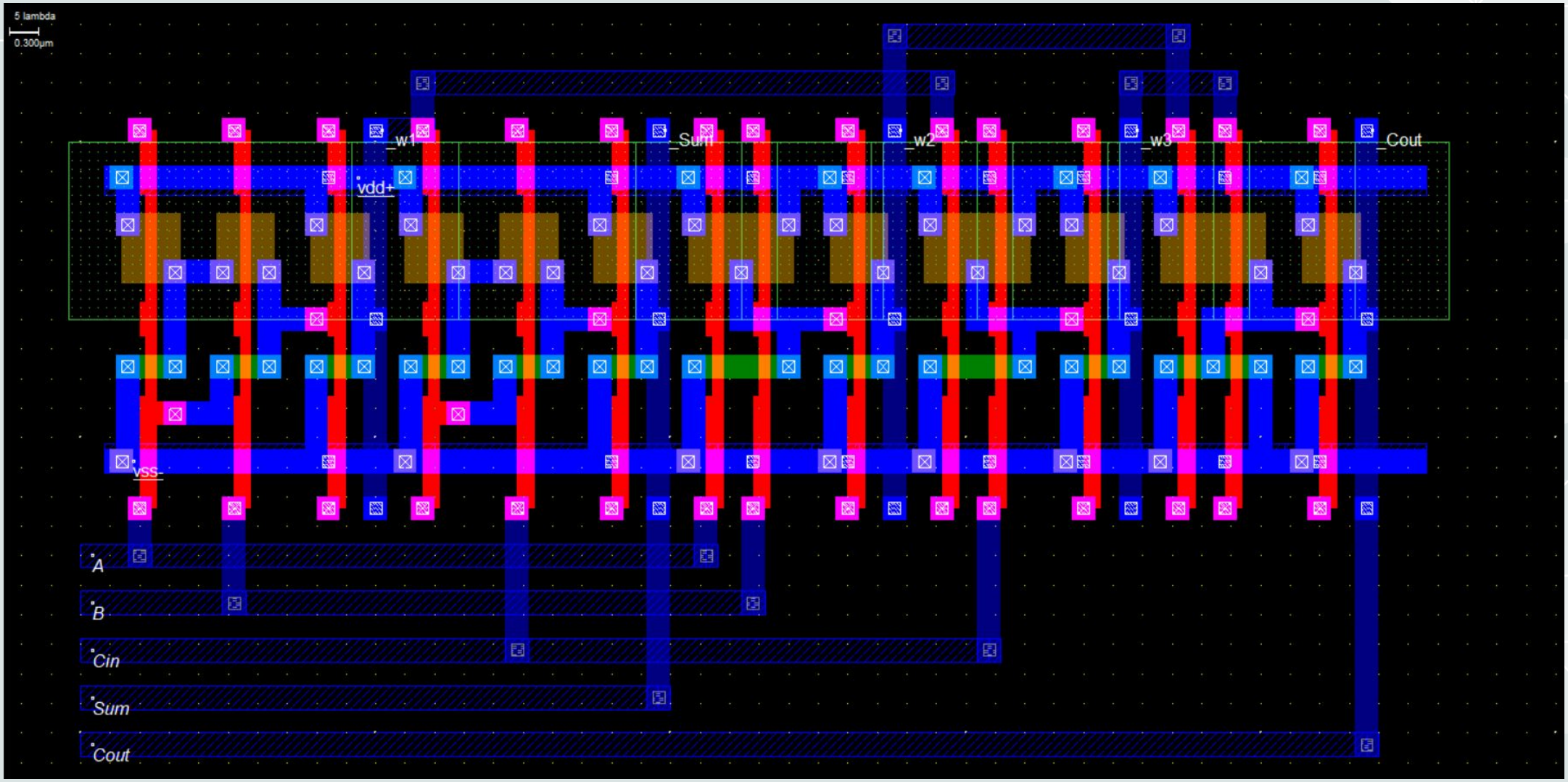
Verilog Testbench Code for 4-Bit BCD Adder

```
module BCD_Adder_4bit_tb;
    reg [3:0] A, B;
    reg Cin;
    wire [3:0] Sum;
    wire Cout;
    BCD_Adder_4bit uut (
        .A(A),
        .B(B),
        .Cin(Cin),
        .Sum(Sum),
        .Cout(Cout)
    );
    initial begin
        $dumpfile("bcd_adder_tb.vcd");
        $dumpvars(0, BCD_Adder_4bit_tb);
        A = 4'd5; B = 4'd3; Cin = 1'b0; #10;
        A = 4'd7; B = 4'd9; Cin = 1'b0; #10;
        A = 4'd6; B = 4'd8; Cin = 1'b1; #10;
        A = 4'd9; B = 4'd9; Cin = 1'b1; #10;
        $finish;
    end
    initial begin
         $monitor("Time = %0t | A = %d, B = %d, Cin = %b | Sum = %d, Cout = %b", $time, A, B, Cin, Sum,
Cout);
    end
endmodule
```

Simulation Results



Layout



Advantages of BCD Adder

- Maintains Decimal Representation: Ensures that the sum remains in valid BCD format, making it easier for decimal-based systems like calculators and financial applications.
- Simplifies Decimal Arithmetic: Directly supports decimal arithmetic without requiring conversions between binary and decimal.
- Easier Human Interpretation: BCD numbers are more readable compared to pure binary, making debugging and display operations simpler.
- Error Detection Capability: Since only 0000 to 1001 are valid BCD representations, invalid states can be easily detected.
- Useful in Financial and Commercial Applications: Ensures accurate decimal calculations where binary floating-point arithmetic may introduce rounding errors.

Disadvantages of BCD Adder

- More Hardware Complexity: Requires additional logic gates and correction circuitry, making it more complex than a standard binary adder.
- Inefficient Use of Storage: Each decimal digit uses 4 bits, but only 10 out of 16 possible values are valid, wasting storage space.
- Slower Computation: The correction step (adding 0110_2) increases delay compared to a simple binary addition.
- Higher Power Consumption: Due to extra logic operations and corrections, it consumes more power than a binary adder.
- Limited Use Cases: Not suitable for general-purpose computing, as most digital systems use pure binary arithmetic.

Thank you

Aditya Ankitkumar Mistry

240280763001