Name: Dhruvi Patel
Enroll No: 92200133029
Sub: DDV

Lab Exercise 5

1. Write the pure behavioral code for 16 bit adder as a part of ALU and update the following flag bits.

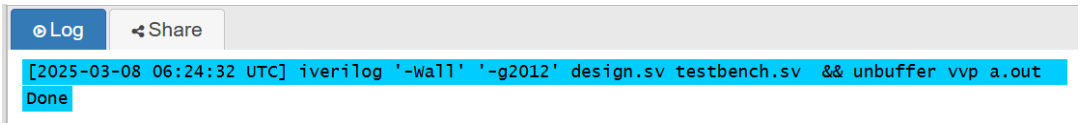
Carry flag,Zero flag,Sign flag,Parity flag and Overflow flag.

Code:

```
// 16-bit Pure Behavioral Adder with Flag Updates (Part of ALU)
module alu_adder_16bit (
    input [15:0] A, B,
    input Cin,
    output reg [15:0] Sum,
    output reg Cout, Zero, Sign, Parity, Overflow
);
    reg [16:0] temp_sum;
    integer i;

    always @(*) begin
        // Perform addition
        temp_sum = A + B + Cin;
        Sum = temp_sum[15:0];

        // Update flags
        Cout = temp_sum[16]; // Carry Flag
        Zero = (Sum == 16'b0) ? 1'b1 : 1'b0; // Zero Flag
        Sign = Sum[15]; // Sign Flag
        Overflow = (A[15] == B[15]) && (Sum[15] != A[15]); // Overflow Flag

        // Compute Parity Flag (Even Parity)
        Parity = 1'b0;
        for (i = 0; i < 16; i = i + 1) begin
            Parity = Parity ^ Sum[i];
        end
        Parity = ~Parity; // Even parity: 1 if even number of 1s, 0 if odd
    end
```

endmodule

Output:



2. Write the test bench for above and verify the verilog code.

Testbench:

```
// Testbench for 16-bit ALU Adder
module tb_alu_adder_16bit();
    reg [15:0] A, B;
    reg Cin;
    wire [15:0] Sum;
    wire Cout, Zero, Sign, Parity, Overflow;

    // Instantiate the 16-bit ALU Adder
    alu_adder_16bit uut (A, B, Cin, Sum, Cout, Zero, Sign, Parity, Overflow);

    initial begin
        $monitor("Time=%0t A=%h B=%h Cin=%b | Sum=%h Cout=%b Zero=%b
Sign=%b Parity=%b Overflow=%b",
                 $time, A, B, Cin, Sum, Cout, Zero, Sign, Parity, Overflow);

        // Test cases
        A = 16'h0000; B = 16'h0000; Cin = 1'b0; #10;
        A = 16'hFFFF; B = 16'h0001; Cin = 1'b0; #10;
        A = 16'h7FFF; B = 16'h0001; Cin = 1'b0; #10;
        A = 16'h8000; B = 16'h8000; Cin = 1'b0; #10;
        A = 16'hAAAA; B = 16'h5555; Cin = 1'b1; #10;
        A = 16'hFFFF; B = 16'hFFFF; Cin = 1'b1; #10;

        $stop;
    end
endmodule
```

Output:

Log Share

```
[2025-03-08 06:26:43 UTC] iverilog '-Wall' '-g2012' design.sv testbench.sv  && u
Time=0 A=0000 B=0000 Cin=0 | Sum=0000 Cout=0 Zero=1 Sign=0 Parity=1 Overflow=0
Time=10 A=ffff B=0001 Cin=0 | Sum=0000 Cout=1 Zero=1 Sign=0 Parity=1 Overflow=0
Time=20 A=7fff B=0001 Cin=0 | Sum=8000 Cout=0 Zero=0 Sign=1 Parity=0 Overflow=1
Time=30 A=8000 B=8000 Cin=0 | Sum=0000 Cout=1 Zero=1 Sign=0 Parity=1 Overflow=1
Time=40 A=aaaa B=5555 Cin=1 | Sum=0000 Cout=1 Zero=1 Sign=0 Parity=1 Overflow=0
Time=50 A=ffff B=ffff Cin=1 | Sum=ffff Cout=1 Zero=0 Sign=1 Parity=1 Overflow=0
testbench.sv:23: $stop called at 60 (1s)
** VVP Stop(0) **
** Flushing output streams.
** Current simulation time is 60 ticks.
```