KARTHICK RAJA B
126004121

Given:
Assume that you have two 8-bit 2's complement numbers, a[7:0] and b[7:0]. These numbers are added to produce s[7:0]. Also compute whether a (signed) overflow has occurred.

Verilog code:

```
module RTL_Day_16(a,b,s,overflow);
input  [7:0] a;
input  [7:0] b;
output [7:0] s;
output overflow;
assign s = a + b;
assign overflow = (~a[7] & ~b[7] & s[7]) | (a[7] & b[7] & ~s[7]);
endmodule
```

Testbench code:

```
module test_bench_tb16();
reg [7:0] a;
reg [7:0] b;
wire [7:0] s;
wire overflow;
RTL_Day_16 f1 (a,b,s,overflow);
initial
begin
a = 8'b00011110;  b = 8'b00101000;  #100;
a = 8'b11100010;  b = 8'b11011000;  #100;
a = 8'b00110010;  b = 8'b01010000;  #100;
a = 8'b11001110;  b = 8'b10110000;  #100;
end
endmodule
```

Functional Simulation: