Internship Task-2: RAM Design

Objective

To develop a simple synchronous RAM module with read and write operations using Verilog.

Verilog Code: Synchronous RAM

```
module synchronous_ram (
    input clk,
    input we,
    input [3:0] addr,
    input [7:0] din,
    output reg [7:0] dout
);

    reg [7:0] mem [15:0];

    always @(posedge clk) begin
        if (we)
            mem[addr] <= din;
        else
            dout <= mem[addr];
    end

endmodule
```

Verilog Testbench

```
module tb_synchronous_ram;
    reg clk;
    reg we;
    reg [3:0] addr;
    reg [7:0] din;
    wire [7:0] dout;

    synchronous_ram uut (
        .clk(clk),
        .we(we),
        .addr(addr),
        .din(din),
        .dout(dout)
    );

    always #5 clk = ~clk;

    initial begin
        $display("Starting RAM Test...");
        $monitor("Time=%0d we=%b addr=%h din=%h dout=%h", $time, we, addr, din, dout);
```

```
        clk = 0;
        we = 0;
        addr = 0;
        din = 0;

        #10 we = 1; addr = 4'h1; din = 8'hAA;
        #10 addr = 4'h2; din = 8'h55;
        #10 we = 0;
        #10 addr = 4'h1;
        #10 addr = 4'h2;
        #10 $finish;
    end
endmodule
```

Simulation Output Explanation

```
Simulation Output Explanation:
- Time=10: Writing 0xAA to address 0x1
- Time=20: Writing 0x55 to address 0x2
- Time=30: Reading from address 0x1 -> Output: 0xAA
- Time=40: Reading from address 0x2 -> Output: 0x55

This confirms the RAM module's read and write functionality.
```