Task 4

Verilog Code for FIR Filter:

```
module fir_filter (
    input wire clk,
    input wire rst,
    input wire [7:0] x_in,  // 8-bit input sample
    output reg [15:0] y_out // 16-bit output sample
);
    parameter h0 = 8'd1;
    parameter h1 = 8'd2;
    parameter h2 = 8'd3;
    parameter h3 = 8'd4;
    reg [7:0] x [0:3];

    always @(posedge clk or posedge rst) begin
        if (rst) begin
            x[0] <= 0;
            x[1] <= 0;
```

```
      x[2] <= 0;
      x[3] <= 0;
      y_out <= 0;
    end else begin
      x[3] <= x[2];
      x[2] <= x[1];
      x[1] <= x[0];
      x[0] <= x_in;          y_out <= (h0 * x[0]) + (h1 * x[1]) + (h2 * x[2]) + (h3 * x[3]);
    end
  end
endmodule
```

Testbench for FIR Filter:

```
module tb_fir_filter;
    reg clk, rst;
    reg [7:0] x_in;
    wire [15:0] y_out;

    // Instantiate FIR Filter
    fir_filter uut (
        .clk(clk),
        .rst(rst),
        .x_in(x_in),
        .y_out(y_out)
    );
    always #5 clk = ~clk;
 initial begin;
        clk = 0;
        rst = 1;
        x_in = 0;
        #10 rst = 0;
        #10 x_in = 8'd10;  // Input Sample 1
        #10 x_in = 8'd20;  // Input Sample 2
        #10 x_in = 8'd30;  // Input Sample 3
        #10 x_in = 8'd40;  // Input Sample 4
        #10 x_in = 8'd50;  // Input Sample 5
        #10 x_in = 8'd60;  // Input Sample 6

        #50 $finish;
    end
    initial begin
        $monitor("Time=%0t | x_in=%d | y_out=%d", $time, x_in, y_out);
    End;
endmodule
```