Verilog Code: Low-Power FIR Filter:

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

    // Symmetric coefficients: h = [2, 4, 4, 2]
    parameter signed [7:0] coeff [0:3] = '{2, 4, 4, 2};
    reg [7:0] x_reg [0:3];
    integer i;
    integer acc;

    always @(posedge clk or posedge reset) begin
        if (reset) begin
            for (i = 0; i < 4; i = i + 1)
                x_reg[i] <= 8'd0;
            y_out <= 16'd0;
            acc <= 0;
        end
        else if (enable) begin
            // Shift register
            x_reg[3] <= x_reg[2];
```

```
        x_reg[2] <= x_reg[1];
        x_reg[1] <= x_reg[0];
        x_reg[0] <= x_in;

        // FIR filter operation
        acc = coeff[0]*x_reg[0] +
              coeff[1]*x_reg[1] +
              coeff[2]*x_reg[2] +
              coeff[3]*x_reg[3];

        y_out <= acc[15:0];
      end
   end
endmodule
```

🧪 Verilog Testbench:

```
module tb_fir_filter;

   reg clk = 0;
   reg reset = 0;
   reg enable = 1;
   reg [7:0] x_in;
   wire [15:0] y_out;

   fir_filter uut (
```

```
        .clk(clk),
        .reset(reset),
        .enable(enable),
        .x_in(x_in),
        .y_out(y_out)
    );

    // Clock generation
    always #5 clk = ~clk;

    initial begin
        $display("Starting FIR Filter Test...");
        reset = 1; #10;
        reset = 0;

        x_in = 8'd1; #10;
        x_in = 8'd2; #10;
        x_in = 8'd3; #10;
        x_in = 8'd4; #10;
        x_in = 8'd0; #50;

        $finish;
    end

endmodule
```