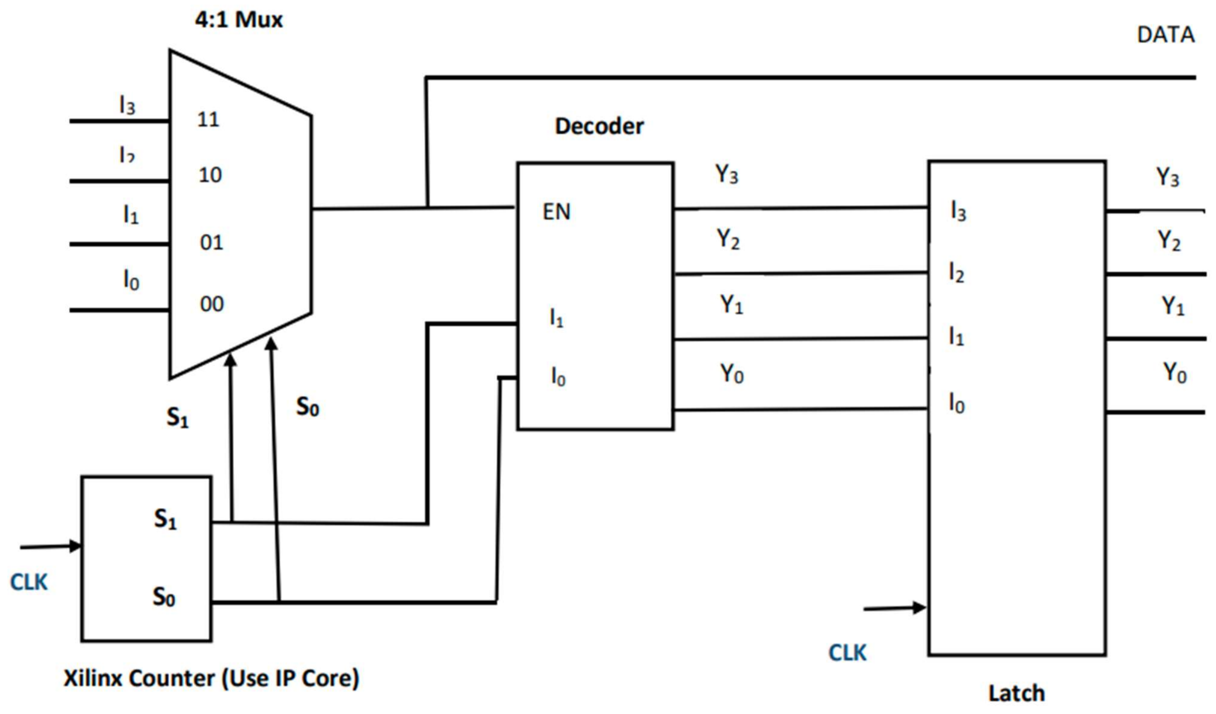
MINI PROJECT

Submitted by,
AHIRAJ K
LW-FPGA-Batch9

OBJECTIVE



- To code the above design in verilog HDL and implement the same on the FPGA Kit allotted to you
- Use Virtual input and Output IP Core for giving input I3, I2, I1, I0
- CLK from the Kit.
- Y3 Y2 Y1 Y0 need to be displayed on Chipscope IP Core.
  1. 4-1 Mux use dataflow
  2. Decoder use behaviour modeling
  3. Latch use behaviour modeling
  4. Counter- Use Xilinx IP Core. 5. Connect everything as miniproject.v and implement on the Kit.

Verilog HDL Codes

The project was developed using a **modular design approach** in Verilog HDL. Each functional block was coded in a **separate Verilog source file**, allowing better clarity, debugging, and reusability. The following modules were created:

- **mux4to1.v** – 4:1 Multiplexer using **dataflow modeling**
- **decoder2to4.v** – 2-to-4 Decoder using **behavioral modeling**
- **latch4.v** – 4-bit Latch using **behavioral modeling**
- **miniproject.v** – Top-level module that **instantiates and integrates all the above modules along with a Xilinx IP Core** (2-bit counter)

The final design was **simulated using a dedicated testbench file tb_miniproject.v**, which applies test inputs and verifies the correctness of outputs using waveform analysis.

Each module and the top-level integration are shown below:

mux4to1.v

(4:1 MUX using Dataflow Modeling)

```
module mux4to1 (
    input  [3:0] D,
    input  [1:0] S,
    output       Y
);
    assign Y = (S == 2'b00) ? D[0] :
               (S == 2'b01) ? D[1] :
               (S == 2'b10) ? D[2] :
                              D[3];
endmodule
```

decoder2to4.v

(2-to-4 Decoder using Behavioral Modeling)

```
module decoder2to4 (
    input        EN,
    input  [1:0] A,
    output reg [3:0] Y
);
    always @(*) begin
        if (!EN)
            Y = 4'b0000;
        else begin
            case (A)
                2'b00: Y = 4'b0001;
                2'b01: Y = 4'b0010;
                2'b10: Y = 4'b0100;
                2'b11: Y = 4'b1000;
            endcase
        end
    end
endmodule
```

latch4.v

(4-bit Latch using Behavioral Modeling)

```
module latch4 (
    input        CLK,
    input  [3:0] D,
    output reg [3:0] Q
);
    always @(CLK or D) begin
        if (CLK)
            Q = D;
    end
endmodule
```

miniproject.v

(Top-Level Integration of All Modules + IP Counter)

```
module miniproject (
    input        CLK,
    input  [3:0] I,
    output [3:0] Y
);

    wire [1:0] S;
    wire [3:0] dec_out;
    wire DATA;

    // Counter IP core instantiation
    c_counter_binary_0 my_counter (
        .CLK   (CLK),
        .Q     (S)
    );

    // MUX
    mux4to1 mux_inst (
        .D (I),
        .S (S),
        .Y (DATA)
    );

    // Decoder
    decoder2to4 dec_inst (
        .EN (DATA),
        .A  (S),
        .Y  (dec_out)
    );

    // Latch
    latch4 latch_inst (
        .CLK (CLK),
        .D   (dec_out),
        .Q   (Y)
    );

endmodule
```

tb_miniproject.v

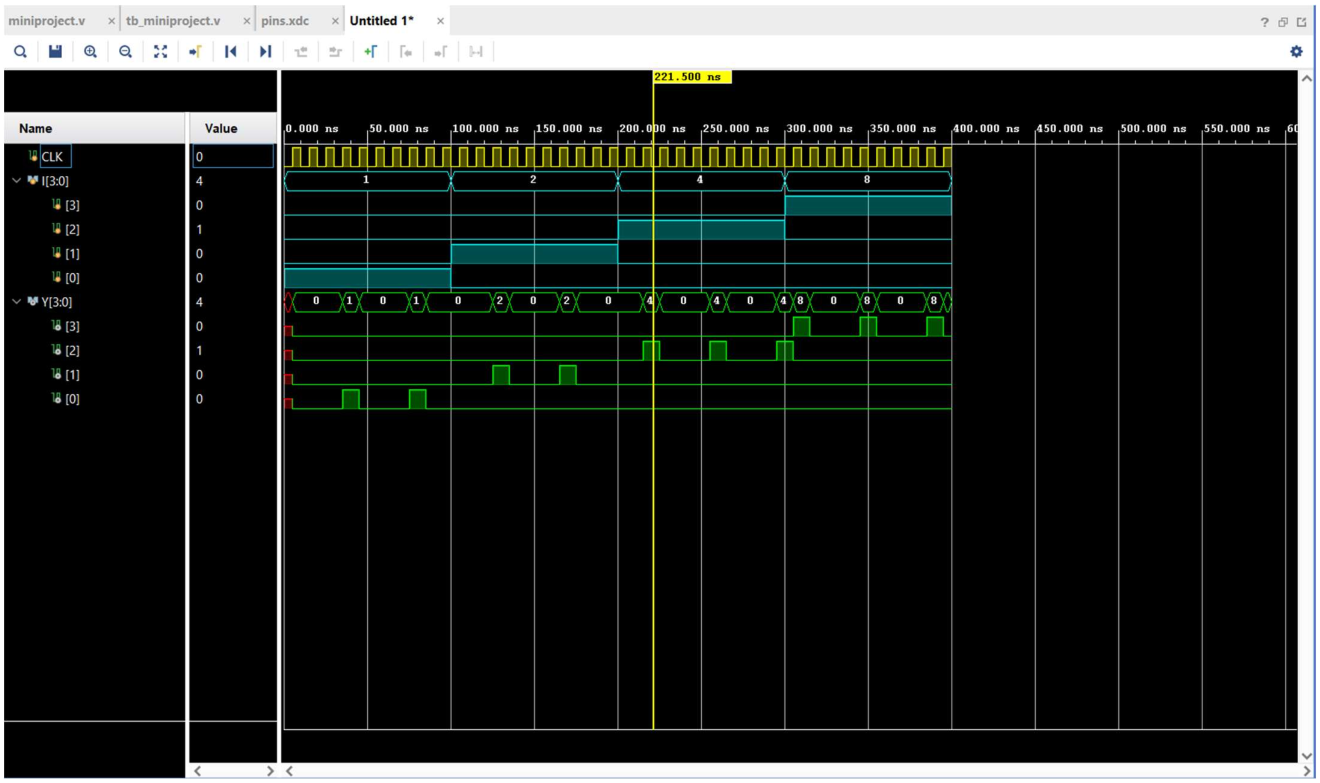
(Simulation Testbench for Verification)

```
`timescale 1ns / 1ps
module tb_miniproject;

    reg CLK = 0;
    reg [3:0] I;
    wire [3:0] Y;

    // Instantiate the DUT (miniproject)
    miniproject dut (
        .CLK(CLK),
        .I(I),
        .Y(Y)
    );

    // Generate 100 MHz clock (10ns period)
    always #5 CLK = ~CLK;

    initial begin
        // Apply test inputs
        I = 4'b0001; #100;
        I = 4'b0010; #100;
        I = 4'b0100; #100;
        I = 4'b1000; #100;
        $stop;
    end
endmodule
```

RESULTS

Simulation Output

The design was simulated using the Vivado simulator with the custom testbench **tb_miniproject.v**.
The waveform shows how Y[3:0] is activated one by one based on the values of I[3:0], passed through the MUX and Decoder, and latched on each clock cycle.



Name	Value
CLK	0
I[3:0]	4
[3]	0
[2]	1
[1]	0
[0]	0
Y[3:0]	4
[3]	0
[2]	1
[1]	0
[0]	0

Bitstream Generation

The bitstream was successfully generated after synthesis and implementation.The **miniproject.bit** file is attached with the project folder submitted along with this report.