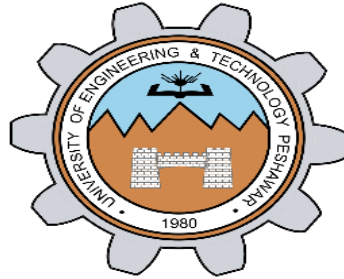


**Lab report no 7**



**Fall 2022**

**CSE-308L Digital Systems Design Lab**

**Submitted By**

<b>Names</b>	<b>Registration No</b>
<b>Muhammad Ali</b>	<b>19pwcse1801</b>

**Section: A**

**Date:10,6,22**

**Submitted To: MAM. Madiha Sher**

Department of Computer Systems Engineering  
University of Engineering and Technology, Peshawar

## Level to pulse converter: -

```
`timescale 1ns / 1ps
```

```
module D_FF (q, d, clock, reset);
```

```
    output reg q;
```

```
    input d, clock, reset;
```

```
    always @(negedge clock or posedge reset)
```

```
    begin
```

```
        if (reset)
```

```
            q = 1'b0;
```

```
        else
```

```
            q = d;
```

```
    end
```

```
endmodule
```

```
module synchronizer(
```

```
    input clk, rst, in,
```

```
    output out
```

```
);
```

```
    wire w1;
```

```
    D_FF DF01 (w1, in, clk, rst);
```

```
    D_FF DF02 (out, w1, clk, rst);
```

```
endmodule
```

```

module clk_divider(clk, clk1, rst);
    integer cnt;
    input clk, rst;
    output reg clk1;
    always @(posedge clk)
        begin
            if(rst)
                begin
                    cnt = 0;
                    clk1 = 0;
                end
            else
                begin
                    cnt = cnt + 1;
                    if(cnt == 100000000)
                        begin
                            clk1 = ~clk1;
                            cnt = 0;
                        end
                end
            end
        end
endmodule

```

```

module L2P(
    input clk, rst, in,

```

```

output reg out
);

synchronizer sync (clk1, rst, in, synin);

clk_divider clk_Div (clk, clk1, rst);

// state = in
// next_state = out

reg s0 = 0;
reg s1 = 1;
// reg [1:0] s2 = 2'b10;
// reg [1:0] s3 = 2'b11;

reg state;
reg next_state;

always @(posedge clk1)
    if(rst)
        begin
            state = s0;
        end
    else
        state = next_state;

always @(*) begin

```

```
case (state)
s0:
begin
    if(synin == 0)
    begin
        out = 0;
        next_state = s0;
    end
    else if(synin== 1)
    begin
        out = 1;
        next_state = s1;
    end
end
s1:
begin
    if(synin== 0)
    begin
        out = 0;
        next_state = s0;
    end
    else if(synin == 1)
    begin
        out = 0;
        next_state = s1;
    end
end
end
```

endcase

end

endmodule

**Output: -**

