24.04.2020

CS 223-4

LAB-4

PRELIMINARY REPORT

ZÜBEYİR BODUR

21702382

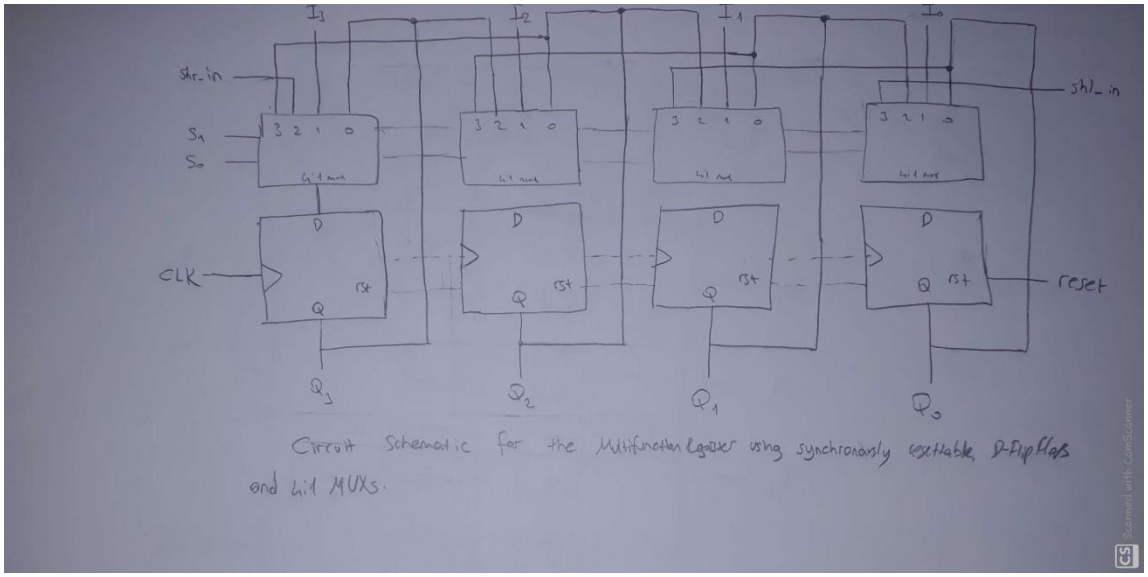
Spring 2020

Zübeyir Bodur

1. Module for Synchronously Resettable D-Flip Flop

```
module d_flip_flop ( input logic clk, rst, d,
                     output logic q
);
    always_ff @(posedge clk) begin
        if (rst) q <= 0;
        else q <= d;
    end
endmodule
```

2. Schematic for Multifunction Register



3. Modules for Multifunction Register & Clock Divider

```
// Module for clock divider to reduce CLK signal from 100 Mhz to 1Hz
module clkDivider(  input logic clk, rst,
                    output logic clk_prime
);
    logic [31:0]divider = 32'b10111110101111000010000000;
    // 10^8 in decimal, can be used to actually see the leds in FPGA
    // logic [31:0]divider = 32'b1; => 2 in decimal, this can be used instead to
see small enough waves in the simulation.

    logic [31:0]count = 32'b0;
```

```
    always_ff @( posedge clk, posedge rst) begin
        if ( rst == 1 )
            count <= 32'b0;
        else if ( count >= divider - 1)
            count <= 32'b0;
        else
            count <= count + 1;
    end

    always_ff @(posedge clk, posedge rst) begin
        if (rst == 1)
            clk_prime <= 32'b0;
        else if (divider != 0 && count == divider - 1)
            clk_prime <= ~clk_prime;
        else
            clk_prime <= clk_prime;
    end
endmodule

// Module for multifunction register
module multifunctionRegister( input logic [1:0]sel, [3:0]in, clk, rst, shr_in, shl_in,
output logic [3:0]q
);
logic clk_prime;
clkDivider divide( clk, rst, clk_prime);

always_ff @(posedge clk_prime, posedge rst)
    if (rst) q <= 4'b0;
    else begin
        case( sel )
            2'b00: q <= q;

            2'b01: q <= in;

            2'b10:
            begin
                q <= q >> 1;
                q[3] <= shr_in;
            end

            2'b11:
```

```
                begin
                    q <= q << 1;
                    q[0] <= shl_in;
                end
            endcase
        end
endmodule
```

4. Testbench for Multifunction Register

```
module testMultiRegister();
    logic [1:0]sel;
    logic [3:0]in;
    logic clk, rst, shr_in, shl_in;
    logic [3:0]q;
    multifunctionRegister test( sel, in, clk, rst, shr_in, shl_in, q);
    initial begin
        sel = 2'b0; in = 4'b0; clk = 0; rst = 1; shr_in = 0; shl_in = 0; #1;
        rst = 0;
        repeat(2) begin
            repeat(2) begin
                repeat(2) begin
                    repeat(2) begin
                        repeat(2) begin
                            repeat(2) begin
                                repeat(2) begin
                                    repeat(2) begin
                                        repeat(2) begin
                                            clk = ~clk; #20;
                                        end
                                        shr_in = ~shr_in; #20;
                                    end
                                    shl_in = ~shl_in; #20;
                                end
                                in[0] = ~in[0]; #20;
                            end
                            in[1] = ~in[1]; #20;
                        end
                        in[2] = ~in[2]; #20;
                    end
                    in[3] = ~in[3]; #20;
                end
                sel[0] = ~sel[0]; #20;
            end
            sel[1] = ~sel[1]; #20;
```

```
            end
        $stop;
        end
endmodule
```