CS 223 - Digital Design

Section 4

Melike Demirci 21702346

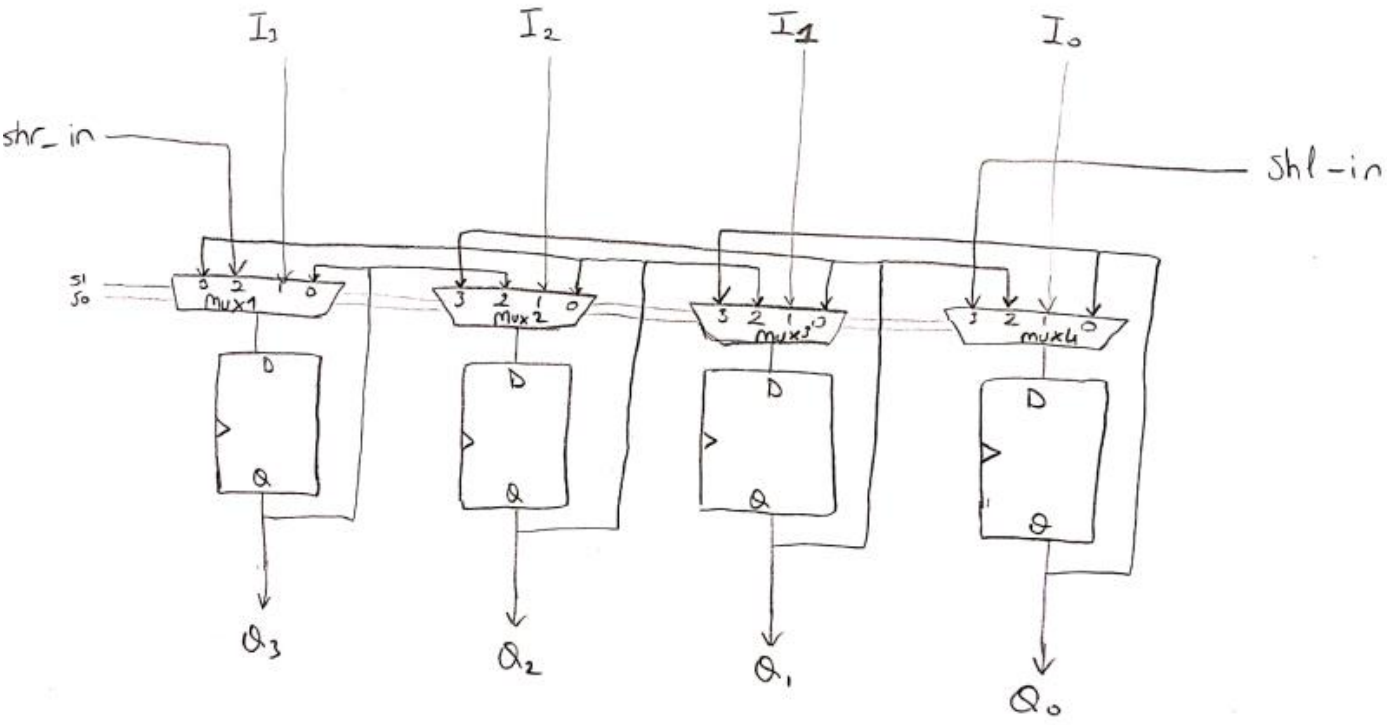
Lab04

22/04/2020

(b) Synchronously resettable D flip-flop

```
module dFlipFlop( input logic clk, reset,
                  input logic [3:0] D,
                  output logic [3:0] Q
   );
always_ff @(posedge clk)
   begin
      if(reset)
      Q <=4'b0;
      else
      Q<= D;
   end
endmodule
```

(c) Draw a circuit schematic (block diagram) for the internal design of the multifunction register by using 4:1 multiplexers and synchronously resettable D flip-flops.



(d) *Structural* module for the Multifunction Register. For the simulation this module is used but for the FPGA there is another module which includes clock divider.

```
module MultifunctionReg(input logic clk, reset, shr_in, shl_in,
                        input logic [1:0] select,
                        input logic [3:0] in,
                        output logic [3:0] q);
   logic [3:0] y;
   mux4to1 mux1(q[2], shr_in , in[3], q[3], select[1], select[0],y[3]);
   mux4to1 mux2(q[1], q[3], in[2], q[2],select[1], select[0],y[2]);
   mux4to1 mux3(q[0], q[2], in[1], q[1],select[1], select[0],y[1]);
   mux4to1 mux4(shl_in, q[1], in[0], q[0],select[1], select[0],y[0]);
   dFlipFlop dFF(clk, reset, y ,q);
endmodule
```

Module for the FPGA(normal MultifunctionReg and Clock Divider used together)

```
module clockDivider(input logic clk,
                            output logic slowerClk);
 logic [26:0] clk_counter;
always @(posedge clk)
begin
   if(clk_counter >= 99999999)
      clk_counter <= 0;
   else
      clk_counter <= clk_counter + 1;
end
assign slowerClk =(clk_counter == 99999999)? 1:0;
```

```
module MultifunctionRegister_forFpga(input logic clk, reset, shr_in, shl_in,
                        input logic [1:0] select,
                        input logic [3:0] in,
                        output logic [3:0] q
   );
   logic slowerClk;
   clockDivider cD(clk, slowerClk);
   MultifunctionReg  mReg (slowerClk, reset, shr_in, shl_in,select,in,q);
endmodule
```

Testbench

```
module MultiReg_tb();

logic clk, reset, shr_in, shl_in;

logic [1:0] select;

logic [3:0] in;

logic [3:0] q;

MultifunctionReg mReg(clk, reset, shr_in, shl_in,select,in,q); // instantiate dut

always

   begin

      clk = 1; #10; clk = 0; #10;

   end

initial begin

   in = 4'b1010;

   shr_in = 1;

   shl_in = 1;

   reset = 1 ; #100;

   select[1] = 0; select[0] = 0; #100;

   select[1] = 0; select[0] = 1; #100;

   select[1] = 1; select[0] = 0; #100;

   select[1] = 1; select[0] = 1; #100;

   reset = 0 ; #100;

   select[1] = 0; select[0] = 0; #100;

   select[1] = 0; select[0] = 1; #100;

   select[1] = 1; select[0] = 0; #100;

   select[1] = 1; select[0] = 1; #100;

end

endmodule
```

Simulation showed that the module MultifunctionReg works correctly.