

EASTERN MEDITERRANEAN UNIVERSITY
COMPUTER ENGINEERING DEPARTMENT
CMPE224 DIGITAL LOGIC SYSTEMS
EXPERIMENTAL WOK #3
REGISTERS IN VeriLog HDL

OBJECTIVES:

This laboratory work aims to introduce a practical work on the design of Registers from architectural and behavioral descriptions. The architectural description covers both the schematic and the software implementation of circuits designed through the conventional design procedure. The behavioral descriptions cover the implementation using state transition diagrams.

Important Note: *For each of the following experimental tasks (in each phase), open a new project to avoid compilation errors due to multiple use of components within the same project's files.*

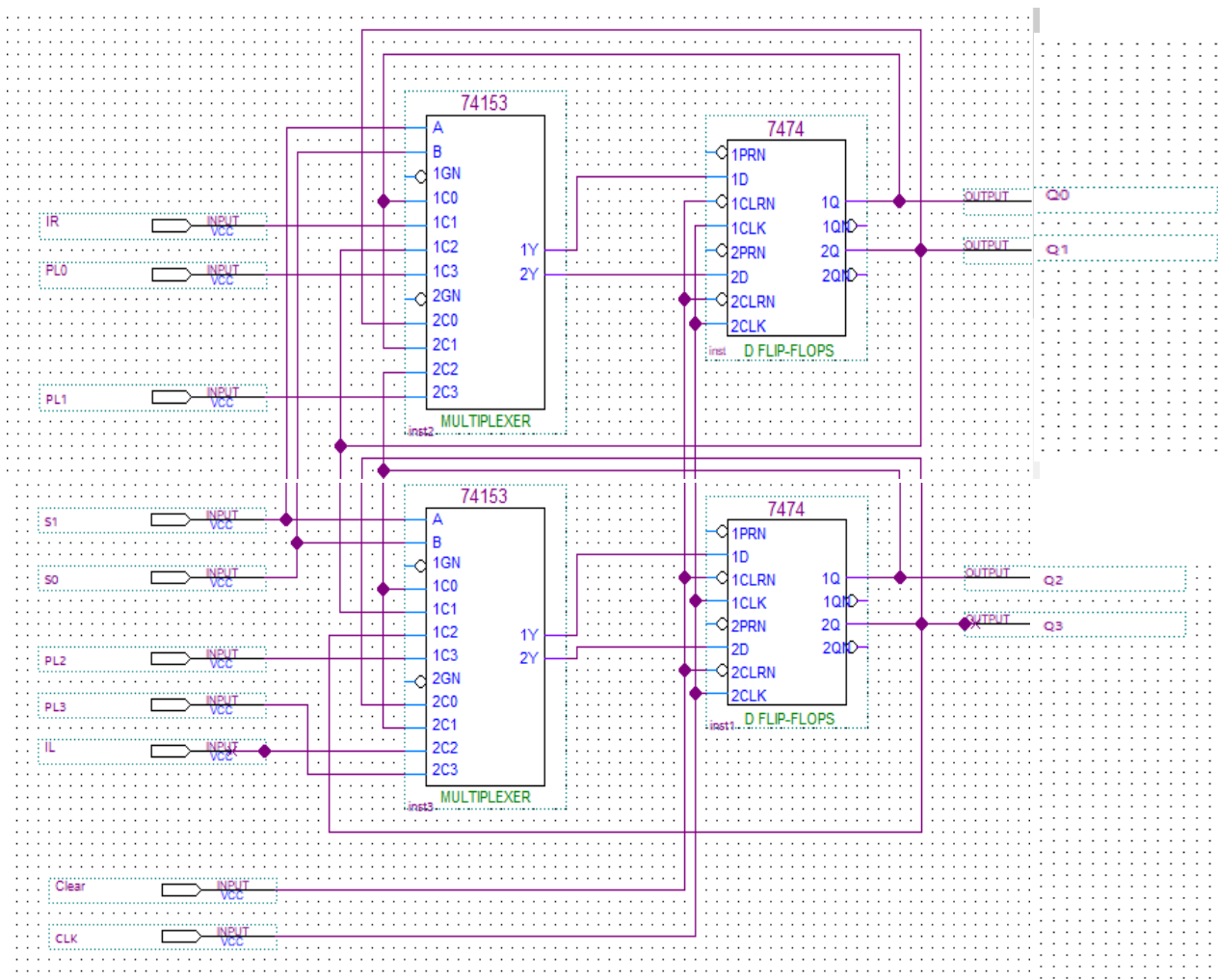
Phase 1: Schematic-Entry

Assume that we want to design the following multi-function register that is controlled by two control inputs S1 and S0 as follows:

Mode Control		Register Operation
<i>S1</i>	<i>S0</i>	
0	0	<i>No change</i>
0	1	<i>Shift left</i>
1	0	<i>Shift right</i>
1	1	<i>Parallel Load</i>

The schematic circuit corresponding to this multi-function register is given below:

1.1. Draw the circuit of this multifunction register, compile and simulate it in VeriLog HDL environment and verify its mode of operations. Adjust and apply appropriate waveforms to observe different function modes easily.



Phase 2: Implementing the Architectural Design of Multi-Function Register in Verilog HDL

Enter the following architectural VeriLog code of multifunction register design into Quartus Lite development suite. Compile and simulate your code to verify its correctness.

/* 4-bit Multifunction Register controlled by two control inputs S1 and S0 as follows:

s1 s0=00 No change

01 Shift left

10 Shift right

11 Parallel load

*/

```

module MultiFuncRegister_Arch(Clear,CLK,S,PL,IL,IR,Q);
    input CLK;
    input Clear;
    input IL, IR;           // Serial load from left and right
    input [1:0] S;         // Vector of control inputs S1 and S0
    input [3:0] PL;       // Parallel load
    output [3:0] Q;       // Register outputs Q3, Q2, Q1, Q0
    wire [3:0] W;         // Internal signals among components

    MUX_4_1 m1(W[0],S[1],S[0],PL[0],IL,Q[1],Q[0]);
    MUX_4_1 m2(W[1],S[1],S[0],PL[1],Q[0],Q[2],Q[1]);
    MUX_4_1 m3(W[2],S[1],S[0],PL[2],Q[1],Q[3],Q[2]);
    MUX_4_1 m4(W[3],S[1],S[0],PL[3],Q[2],IR,Q[3]);

    D_FF d1(Q[0],W[0],CLK,Clear);
    D_FF d2(Q[1],W[1],CLK,Clear);
    D_FF d3(Q[2],W[2],CLK,Clear);
    D_FF d4(Q[3],W[3],CLK,Clear);

```

```
endmodule
```

```

module D_FF(Q,D,CLK,CLR);
    input D,CLK,CLR;
    output reg Q;

    always @(posedge CLK)
        if (CLR == 1'b1)
            Q<= 1'b0;
        else
            Q<= D;

```

```
endmodule
```

```

module MUX_4_1(Y,S1,S0,I3,I2,I1,I0);
    input S1,S0,I3,I2,I1,I0;
    output reg Y;

    always @(S1,S0,I3,I2,I1,I0)
    begin
        if (S1==0 & S0==0)
            Y=I0;
        else if (S1==0 & S0==1)
            Y=I1;
        else if (S1==1 & S0==0)
            Y=I2;
        else if (S1==1 & S0==1)
            Y=I3;
    end

```

```

end
endmodule

```

HOMEWORK #3 : (Behavioral Description of a Multi-funtion Register)

Behavioral VeriLog code of the above described multifunction register is given below:

```
/* Behavioral description of a multifunction register in verilog HDL
   s1 s0 =00 Nocahtge
   S1 S0 =01 Shift left
   S1 S0 =10 Shift right
   S1 S0 =11 Parallel load
*/
```

```
module MultiFunctRegister_Behav(Clear,CLK,S,PL,IL,IR,Q);
input Clear, CLK;
input [3:0] PL;
input [1:0] S;
input IL,IR;
output [3:0] Q;

reg [3:0] R;

always @(posedge CLK)
begin
    if (Clear == 1)
        R <= 4'b0000;
    else if (S[1]==0 & S[0]==0) // No change
        R <= R;
    else if (S[1]==0 & S[0]==1) // Shift left
    begin
        R[0] <= IR; R[1] <= R[0];
        R[2] <= R[1]; R[3] <= R[2];
    end
    else if (S[1]==1 & S[0]==0) // Shift right
    begin
        R[3] <= IL; R[2] <= R[3];
        R[1] <= R[2]; R[0] <= R[1];
    end
    else if (S[1]==1 & S[0]==1)
    begin
        R=PL;
    end
end

assign Q = R;

endmodule
```

Modify the above-given behavioral code to design a 4-bit multifunction register that operates as follows:

Enable	S1	S0	Operation Mode
<i>0</i>	<i>x</i>	<i>x</i>	<i>No change</i>
<i>1</i>	<i>0</i>	<i>0</i>	<i>Rotate left</i>
<i>1</i>	<i>0</i>	<i>1</i>	<i>XOR contents with (0101)</i>
<i>1</i>	<i>1</i>	<i>0</i>	<i>Rotate right</i>
<i>1</i>	<i>1</i>	<i>1</i>	<i>Parallel load</i>

Submit your homework at the beginning of the 4-the experimental work.

Prepared by Assoc. Prof. Dr. Adnan ACAN