Computer Aided Digital System Design
(99-00-1)
Dr. Hajar Falahati

Homework 4: Technology Mapping, Synthesis **Deadline:** 1399/11/02 23:59			
Problem	**Definition**	**Credit**	**Your Mark**
P1	**SPLD**	**30**	
P2	**FPGA Synthesis**	**40**	
P3	**Synthesis Rules**	**30**	
Total		**100**	

Required File:
Upload a zip file titled as "CAD-HW4- *Student numberi- …- Student numberj"*.

Contact Information:
Ask your questions via the course website or send an email to:
hosseinaminiii75@gmail.com
hosseini99.zahra@gmail.com
hfalahati@ipm.ir

1- SPLD: Simple Programmable Logic Device [30 points]

Design a circuit which gets a non-negative 4-bit input and return 1 when the input is divisible by 3.

i. Implement the circuit by a ROM with 4 input lines and two output lines. [10 points]

ii. Implement the circuit by a PLA consisting of 4 input lines and two output lines. [10 points]

iii. Implement the circuit by a PAL consisting of 4 input lines, four 2-input OR. The OR output and its complement are sent back to the logic. [10 points]

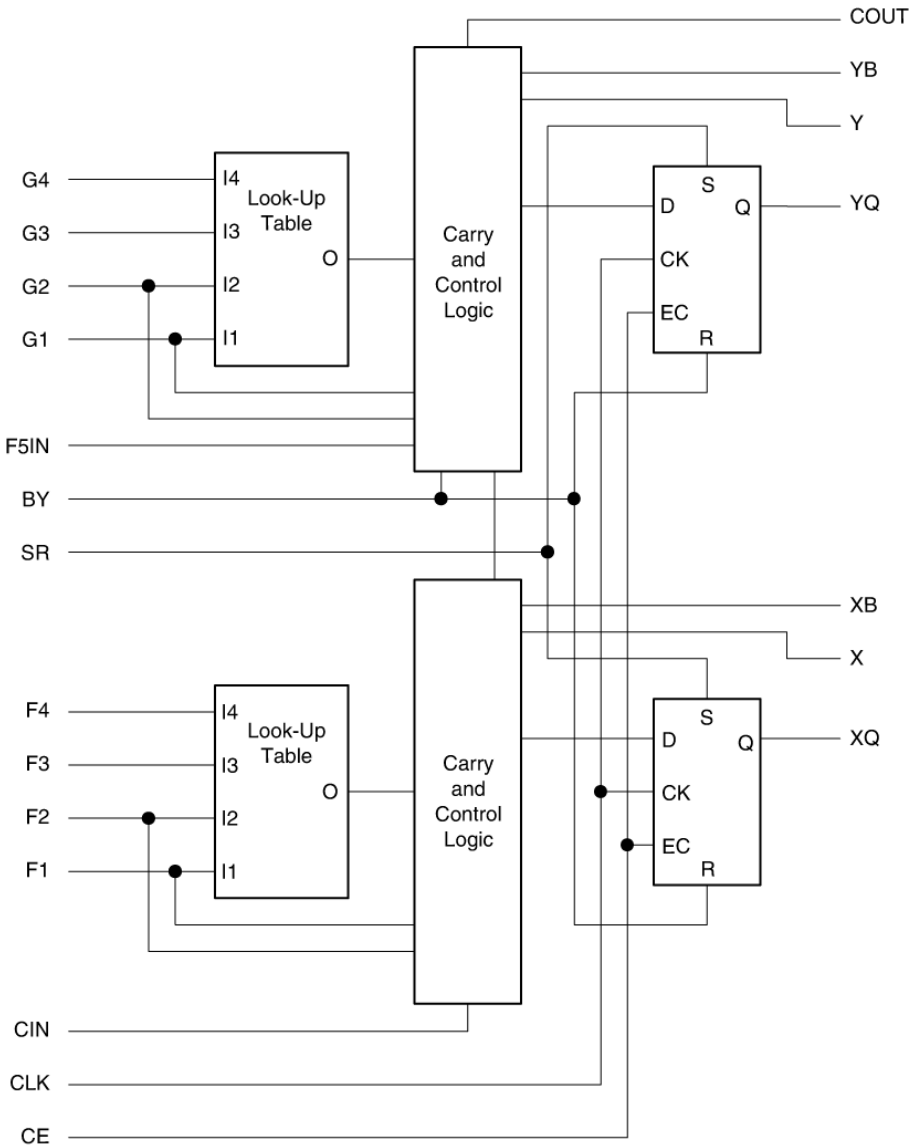
2- FPGA Synthesis [40 points]

i. Implement a protein detector which detect the LDLD protein using the FPGA whose basic block is shown in the following.

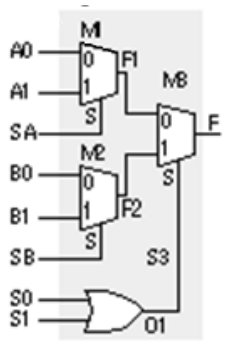
Hint 1: Draw the FSM at first.

Hint 2: You have already drawn the FSM!

.



ii. Implement the function $f = B + (A + \bar{C}) . (\bar{A} + \bar{B} + \bar{C})$ on the following mux-based FPGA.



3- Synthesis Rules [30 points]

Suppose you have all the logic resources. Synthesis the following codes and elaborate if there are any mistakes. Determine the type of errors

i. Code 1: (For example, i++ is wrong, replace it by i=i+1. Suppose 1< n <4)

```
always @ (A or B)
begin
      for (i = 0; i< n; i++)
        begin
          M[i] = A[i] * B[i];
          PS[i] = PS[i] + M[i];
        end
```

ii. Code 2

```
Module concatination2 (input clk, input dataIn1, input dataIn2, output [1:0] dataO);

    always @ (posedge clk, data_in1)
          if (data_in1)
              dataO = {dataIn1, dataIn2};
        else if (data_in2)
            dataO = {dataIn2, dataIn1};

endmodule
```