# Lab 8 9’s Complementer

**Objective**

The objective of this lab is to design a 9’s complementer from a hierarchy of components using Verilog description and simulate using a test bench.

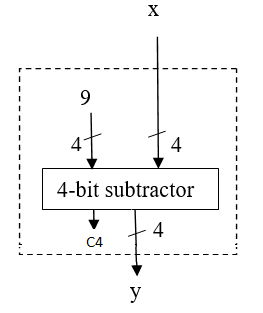
**Lab Procedure**

In this lab assignment, you will continue to practice hierarchical design by designing a 9’s complementer using verilog. You will start with the design from the previous lab (half adder and full adder), then build a 4-bit adder, 4-bit adder/subtractor, and finally a 9’s complementer.

* Half-adders in terms of gates. (Lab 7)
* Full-adders in terms of half-adders (Lab 7)
* 4-bit adder in terms of full-adders
* 4-bit adder/subtractor in terms of 4-bit adder
* 9’s complementer in terms of 4-bit adder/subtractor

Like 2’s complement, 9’s complement is used to subtract a number using addition. 9’s complement of a decimal number is the subtraction of its each digit from 9. The block diagram of a 1-digit 9’s complementer is shown below. The input decimal digit (0 – 9) is represented as a 4-bit binary, and the output generates the 9’s complement in binary by calculating y = 9 - x.

x



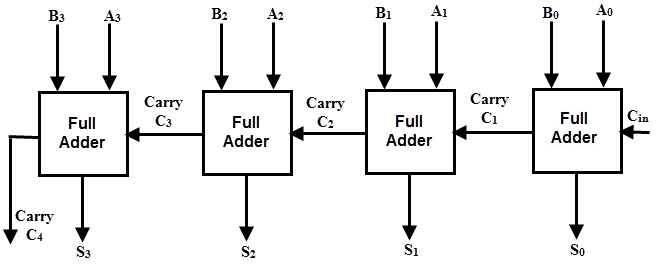
4

9’s complementer

4

y

1. Copy your Lab 7’s verilog code for half adder and full adder (from “your playground” on EDAplayground.com).
2. Build a 4-bit adder: In this part, we will design a 4-bit adder following the topology of the circuit diagram. In the following module for the 4-bit adder, add Verilog statements below the comments so it matches the circuit. The module for 1-bit full adder from Lab 7 is “fulladder (S, C, x, y, cin)”



module four\_bit\_adder (S, C4, A, B, Cin);

input [3:0] A,B;

input Cin;

output [3:0] S;

output C4;

//Declare intermediate carries

wire C1, C2, C3;

//Instantiate the fulladder

fulladder FA0(S[0], C1, A[0], B[0], Cin);

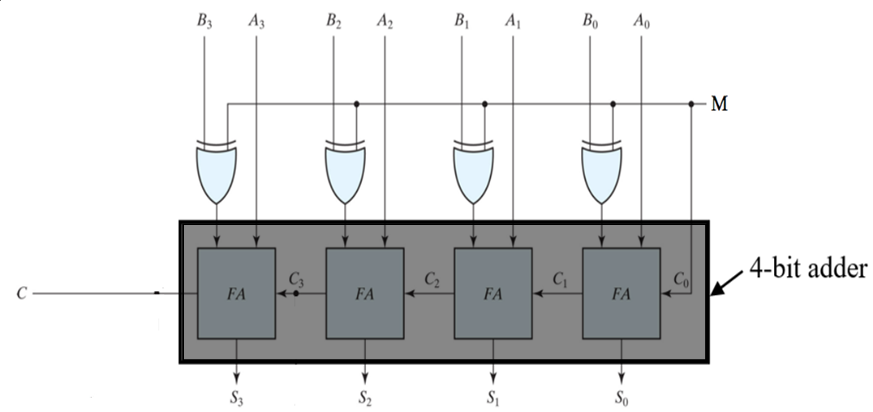
fulladder FA1(S[1], C2, A[1], B[1], C1);

fulladder FA2(S[2], C3, A[2], B[2], C2);

fulladder FA3(S[3], C4, A[3], B[3], C3);

endmodule

1. Build a 4-bit adder/subtractor: Complete the following Verilog module so it matches the circuit below.



module adder\_subtractor(S, C, A, B, M);

input [3:0] A,B;

input M;

output [3:0] S;

output C;

//Declare outputs of XOR gates

wire [3:0]N;

// Instantiate the XOR gates

xor XOR0(N[0],B[0], M);

xor XOR1(N[1],B[1], M);

xor XOR2(N[2],B[2], M);

xor XOR3(N[3],B[3], M);

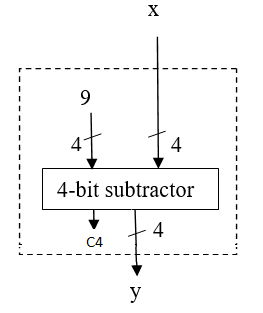
// Declare carry

wire C4;

// Instantiate the 4-bit full adder

four\_bit\_adder FBA(S, C4, A, N, M);

endmodule



1. Write a Verilog module for the 9’s complementer:

module nine\_s\_complementer (x,y);

input [3:0] x;

output [3:0] y;

// Declare wire

wire C4;

// Instantiate the nine\_s\_complementer

adder\_subtractor AS(y, C4, 9, x, 1);

endmodule

1. Write a testbench program to test the 9’s complementer. Fill in the blanks and add all the test cases afterthe comment **“**// Initialize Inputs”

**module test;**

**// input**

**reg[3:0] x;**

**// output**

**wire[3:0] y;**

**// Instantiate the Unit Under Test (UUT)**

**nine\_s\_complementer uut(x,y);**

**initial**

**begin**

**$dumpfile("dump.vcd"); $dumpvars(1, test);**

**// display the inputs and outputs**

**$monitor( "x = %d y = %d", x, y);**

**// Initialize Inputs**

**for(int i = 0; i < 10; i = i + 1) begin**

**{x} = i;**

**#10;**

**end**

**#10 $finish;**

**end**

**endmodule**

1. **Test on EDAplayground.com**

* Log into your EDAplayground.com account.
* Edit your Verilog design code and testbench code in the right and left windows respectively, enter a name such as “Lab 9” for your project in the edit box at the bottom and click the “Save**”** button to save your project.
* On the left panel, under Tools and Simulations, select Icarus Verilog 0.9.7 and check the box of “Open **EPwave** after Run”
* Click “Run” at the top to run the simulation, watch for waveforms and verify your full adder works correctly.

**Submission Instructions:**

Lab work submission

1. Take a screenshot of your wavefroms.
2. Add the following information as comments to the beginning of your code. Make sure to click the “Save” button to save your project, then take a screenshot of your code.

// Author: Name

// Lab 8: put the title here

// Link to your project

1. Copy the link of your design from the address bar of the browser.
2. On the Blackboard, click on Lab 8. Attach the screenshot from the first two steps and paste the link from Step 3 into the Comments area, then click the “Submit” button.

**You do not need to write a lab report for this lab but do the following homework and submit the screenshots and link to the Blackboard under Lab 8 report submission.** It is due one week after the lab is done.

**Homework**: write a Verilog deisgn code and testbench for implementing a 4-bit x 3-bit multiplier using 12 AND gates and two 4-bit adders shown in the diagram. You can use the 4-bit adder module from above. Test at least 10 input cases.

**module multiplier(**C, A, B**); module** test**;**

input [3:0] A; ………

input [2:0] B; ………

output [6:0] C;

………

……… **endmodule**

**endmodule**

sum

sum

Carry-out

Carry-out

