York University
Dept. of Electrical Engineering and Computer Science
EECS2021
Lab Test 2

Lab Section 1 *Nov. 30, 2015*

Read before you start:

- This test has 2 questions
- You are asked to design circuits to do a specific task, no simulation is required (or even allowed in your module) so no initial block.
- Submit your code as **submit 2021 lab2_M xyz.v** where the xyz is as indicated in the problem statement.

Problem 1

In the course, you were given the implementation of a 32-bit adder given one full adder Verilog implementation. The figure below shows the full adder implementation.

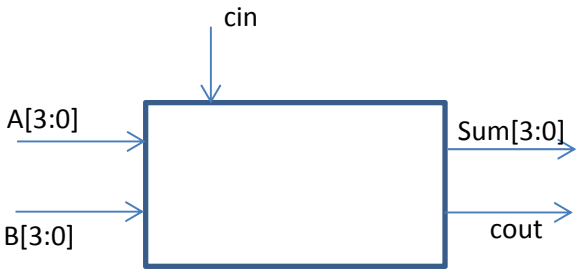
```
module yAdder1(z, cout, a, b, cin);
output z, cout;
input a, b, cin;

xor left_xor(tmp, a, b);
xor right_xor(z, cin, tmp);
and left_and(outL, a, b);
and right_and(outR, tmp, cin);
or my_or(cout, outR, outL);
endmodule
```

Part 1

Implement a 4-bit adder. The module name should be

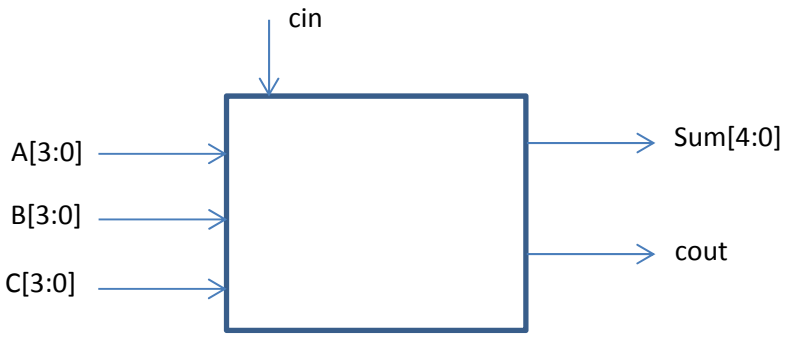
```
module adder4(Sum,cout,A,B,cin);
input [3:0] a,b;
input cin;
output [3:0] Sum;
output cout;
```



Submit as adder4.v

Part 2

Implement a 3-input 4-bit adder, the circuit input is 3 4-bit numbers A,B,C and a carry_in cin, the output is a 5-bit number and a carry_out cout




```
module adder3_4(Sum,cout,A,B,C,cin);
input [3:0] A,B,C;
input cin;
output [4:0] Sum;
output cout;
```

submit as adder3_4.v

Problem 2

Parity is used to check transmission errors during data transmission. In even parity we add a single bit to the data to be transmitted such that the number of 1's in the transmitted word is even. The transmitted word is the data + the parity bit.

For example, consider 4-bit data.

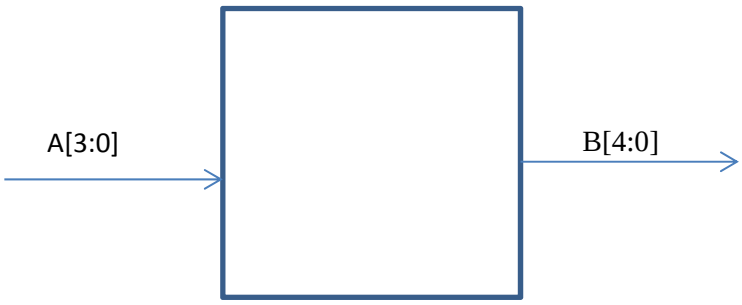
If the data to be transmitted is 0111, there are three 1's, that is an odd number (3) then we add another 1 as the least significant bit to make it 4 1's (even) and we transmit 01111, where the added one (the underlined bit) to the right is the parity it.

If the data to be transmitted is 0101, there are two 1's, that is an even number (2) then we add another 0 as the least significant bit so the total number of 1's remain even (2) and we transmit 01010, where the added zero (the underlined bit) to the right is the parity it.

Design an even-parity generator circuit. It accepts 4-bit input and generates 5-bit output

```
module parityGen(A,b);
input [3:0] A;
output [4:0] B;

submit as parityGen.v
```



For example,

Input	Output
0000	0000**0**
0100	0100**1**
1010	1010**0**
1101	1101**1**