Assignment IV

Question 2

CS220 Homework Assignment

March 6, 2022

March 6, 2022

Problem 1

Write a detailed description of eight-bit adder/subtracter to add/subtract two eight-bit two's complement numbers. and it's working with the proper circuit diagram in a PDF file. Then write the Verilog code module to implement an eight-bit adder/subtracter. It will be implemented in two modules. First, module implements a one-bit adder/subtracter with four inputs a, b, cin, and opcode, and two outputs sum and carry. For the addition operation the input opcode will be 0 and 1 for subtraction operation. The top module implements the eight-bit adder/subtracter using the one-bit adder/subtracter module. There will be two inputs for this module the two input numbers and the opcode and produces the sum and whether there is an overflow as the outputs. Now, add a test bench to test the eight-bit adder/subtracter. Your test bench must have fifteen different inputs. Put five-time unit delay between consecutive inputs.

Solution. The verilog code written for this part is organized in two module. The first module performs single bit operation and second one instantiation eight one bit module to produce the desired result. In this document we will firstly explain simple addition operation and the we will suggest some modification to finally produce an adder subtractor circuit.

Addition operation is achieved by the simplistic approach of taking Xor of the inputs.

$$result = a \oplus b \oplus C_{in}$$

Carry for the next operation is produced whenever any two in a, b, C_{in} is 1.

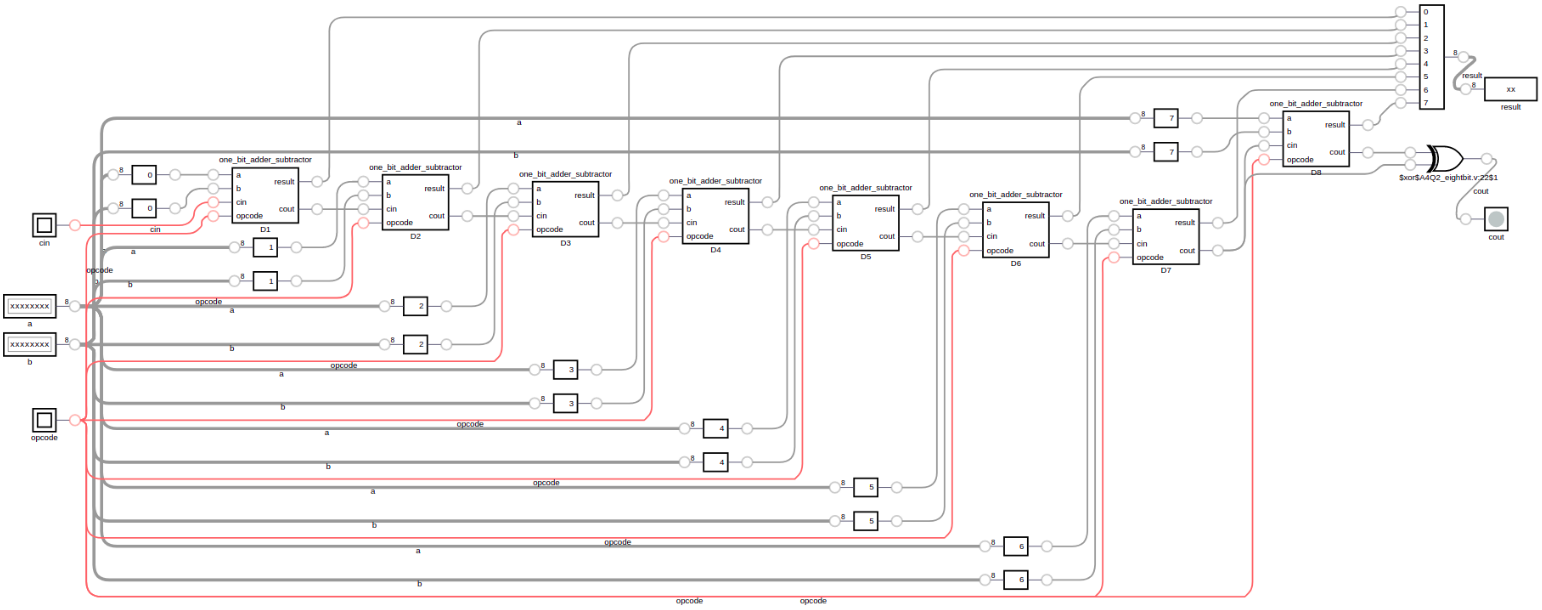
$$C_{out} = (a \&\& b) || (b \&\& C_{in}) || (C_{in} \&\& a)$$

Now subtraction in binary numeral system can be considered as adding 2's complement of the number. So we just need to transform b into its two's complement for performing the subtraction operation. As we know, 2's complement can be fetched by adding 1 to the number after inverting its bits.
In our implementation, this mechanism is achieved by introducing a new variable **opcode**. When opcode is equal to 0 then addition operation is performed whereas if the opcode is 1, subtraction operation is performed.
We take the xor of the bits of b with opcode along with providing it as the initial carry cin.

CASE I : When opcode is 0 On taking xor of bits of b with 0, no change takes place. Since cin is equal to opcode, cin is also 0. Therefore the operation performed is simple addition.

CASE II : When opcode is 1 On taking xor of bits of b with 1, filpping of all the bits take place. Since cin is equal to opcode, cin is 1 thereby adding 1. Therefore the operation performed is subtraction.

Circuit Diagram



one_bit_adder_subtractor
a
b
cin
opcode
result
cout
D1
D2
D3
D4
D5
D6
D7
D8
0
1
2
3
4
5
6
7
8
cin
opcode
xxxxxxxx
a
xxxxxxxx
b
result
xx
result
xorA4Q2_eightbit.v:22$1
cout
cout