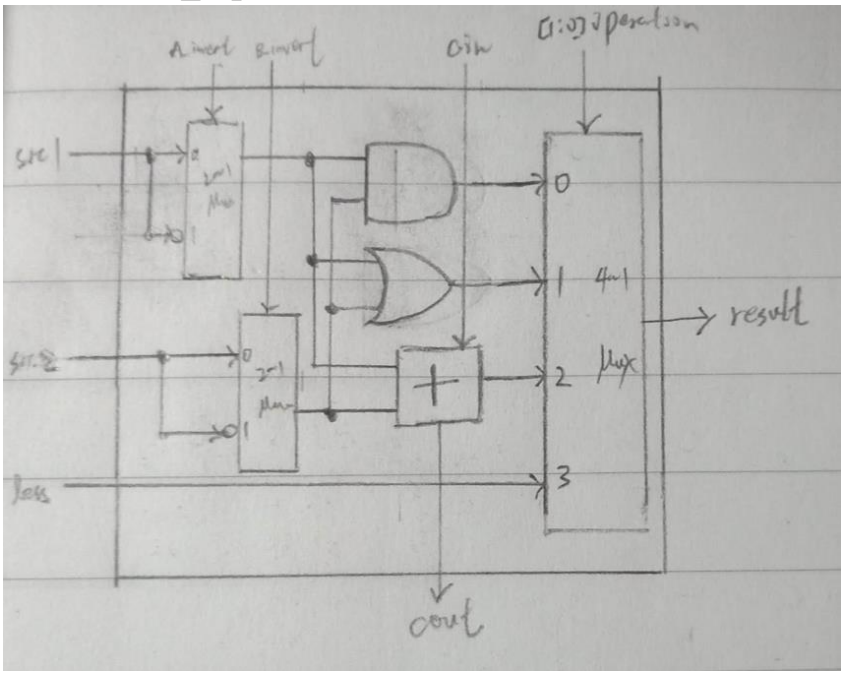
Computer Organization Lab 1: 32-bit ALU

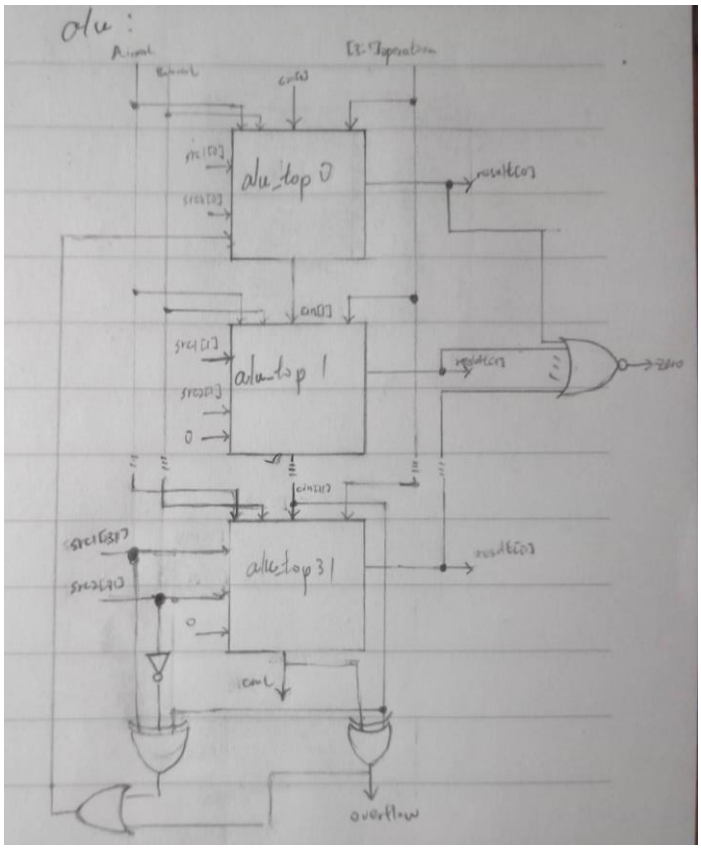
Student ID: Name:

(1) Architecture diagram

module alu_top:



module alu:



(2) Detailed description of the implementation

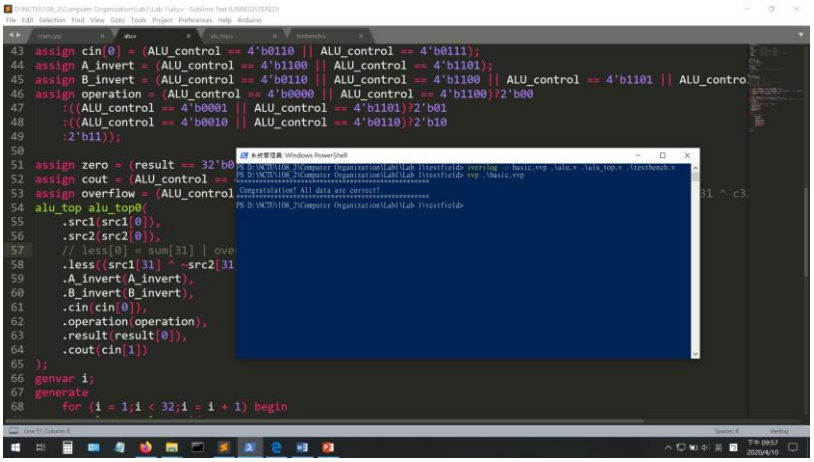
In this alu, we need to implement following operations.

operation	ALU_control	symbol	modify
AND	0000	a&b	
OR	0001	a|b	
ADD	0010	a+b	s = a^b^c_i c_{i+1}= (a&b)|(c&(a^b))
SUB	0110	a-b	a+b'+1
NOR	1100	(a|b)'	a'&b'
NAND	1101	(a&b)'	a'|b'
SLT	0111	a<b?	a-b = a+b'+1<0?

For the inputs 32-bits ALU (module alu),
set cin[0] into 1 if operation is SUB or SLT,
set A_invert into 1 if operation is NOR or NAND,
set B_invert into 1 if operation is SUB or SLT or NOR or NAND,
set operation in 1-bit ALU (alu_top) into 0 if operation is AND or NOR,
set operation in 1-bit ALU (alu_top) into 1 if operation is OR or NAND,
set operation in 1-bit ALU (alu_top) into 2 if operation is ADD or SUB,
set operation in 1-bit ALU (alu_top) into 3 if operation is SLT.
For the outputs 32-bits ALU (module alu),
set zero into 1 if result is 0,
set cout into cin[32] if operation is ADD or SUB,
set overflow into (cin[31] xor cin[32]) if operation is ADD or SUB.
And each 1-bit ALU alu_top[i] where 0 <= i <= 31,
input src1[i] to port src1,
input src2[i] to port src2,
input A_invert to port A_invert,
input B_invert to port B_invert,
input cin[i] to port cin, port result will output result[i],
port cout will output cin[i + 1].
The port less in 1-bit ALU alu_top[i] where 1 <= i <= 31 are all input with 0.
For i=0, input ((src1[31] xor complement of src2[31] xor cin[31]) or overflow) to port less.

SUB(src1[31],src2[31]) = (src1[31]^!src2[32]^cin[31])
less[0] = SUB(src1[31],src2[31]) | overflow
Below is Truth table:

overflow / sub[31]	0	1
0	0	1
1	1	1



Successfully pass all data.

(3) Commands for executing your source codes

iverilog -o basic.vvp testbench.v alu.v alu_top.v

(4) Problems encountered and solutions

In original alu.v, the output result, zero, cout, overflow have reg type. When I do the compile, it comes up a reg result error. It seems that reg type can't be assign directly. To solve this problem, I delete the reg type of these outputs. It works successfully and shows the result is true.

(5) Lesson learnt

How to descript an 32-bits ALU by Verilog.