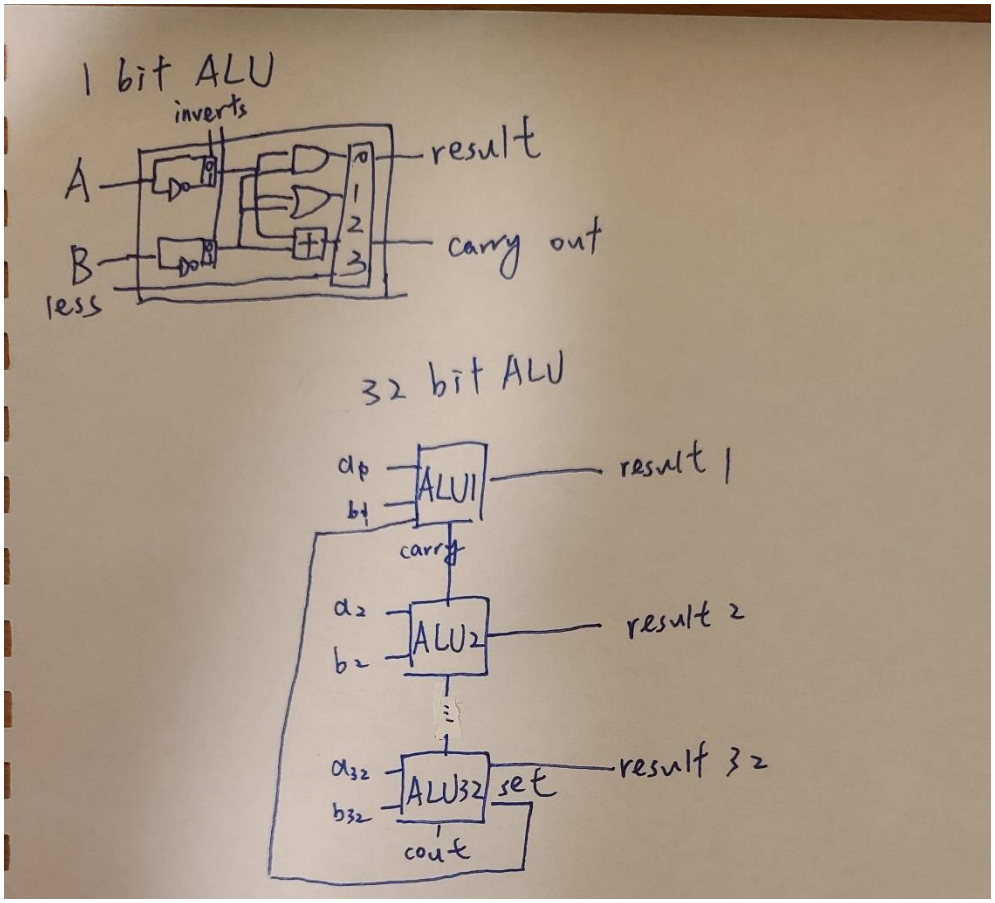
Computer Organization

Architecture diagrams:

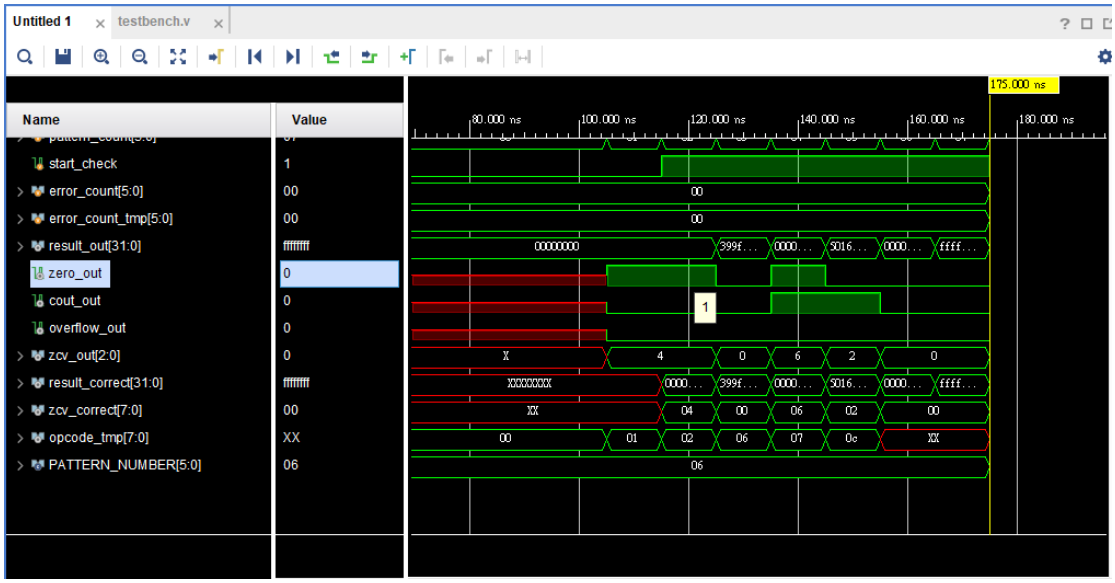


Hardware module analysis:

Build 1-bit ALU first and use generate block to call the module for 32 times and use them to build a 32-bit ALU. Using generate block is faster and more concise when the same module requires multiple calls.

Experiment result:

The result waveform is the same as the correct answer despite that I’ ve defined the result before negedge rst as 32’ b0.



Problems you met and solutions:

I’ve no idea that why my computer is stuck at running simulation on Vivado, and still do not come up with a solution by now, so I use remote control to connect to the PC at home as I have an appointment outside during the holiday. Also, I didn’t do always block when the input changes in alu_top.v in the beginning, so every parameter that has been placed into the module will not get the right result and waveform. I spend a few hours to find the bug and fix the code.

Summary:

I’ve learned generate block and some of the principles of Verilog syntax, such as only wire is allowed to put into a module as output while only reg can be placed in always block to be assigned a value.