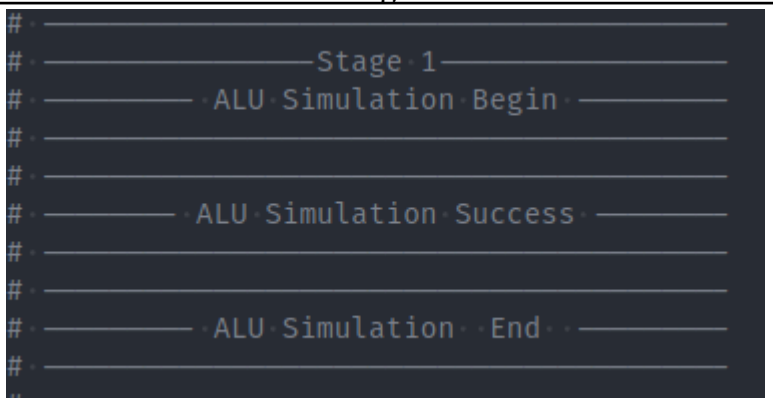
2024 Digital IC Design Homework 1

NAME	楊晴雯
Student ID	P76114511

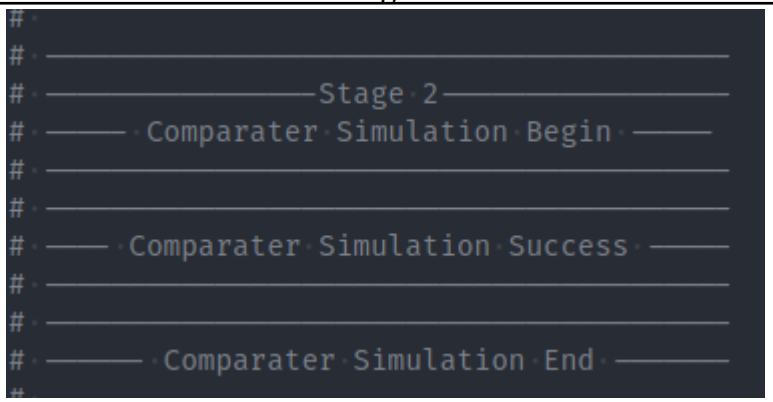
Functional Simulation Result

Stage 1 Pass	Stage 2 Pass	Stage 3 Pass

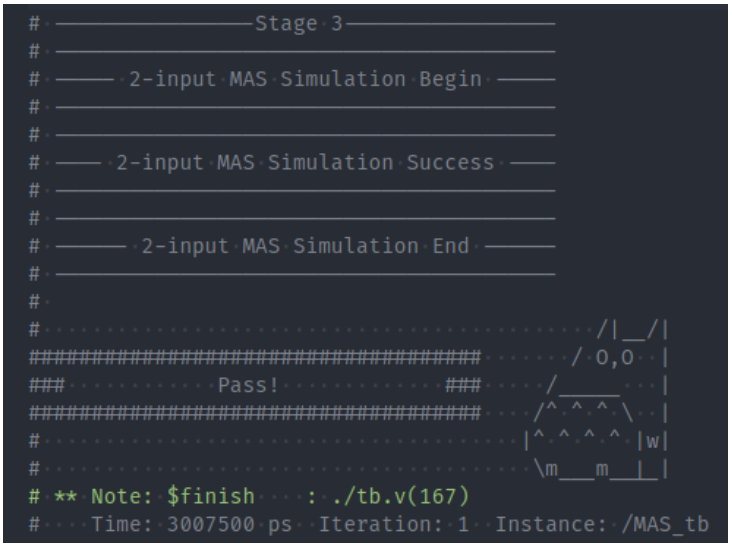
Stage 1



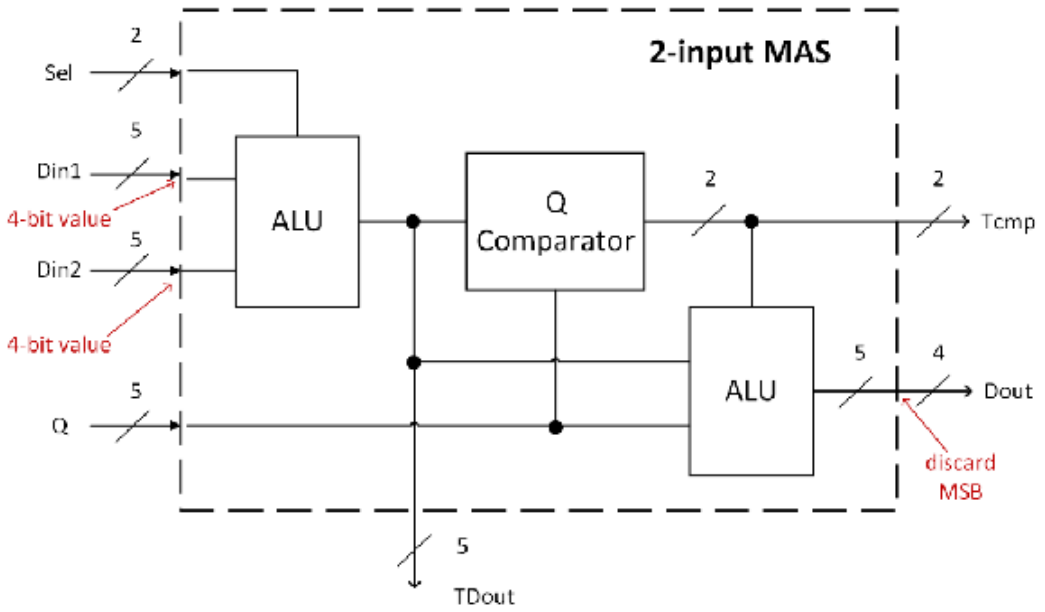
Stage 2



Stage 3



Description of your design



The Modular Adder Subtractor is essentially a ALU (supporting addition and subtract), but will limit the output in a certain range ***[0, +Q)***. Given **the control signal Sel, two numbers Din1 and Din2, the quotient Q**, the LHS ALU performs the normal arithmetic operations based on input signal Sel on **Din1 and Din2**, and then the Q Comparator receives the result as **TDout**; it determines whether TDout >= Q, and whether it >= 0, and then write the 2-bit result **Tcmp (determine if the result is within the range)**. Finally, this result is used as the signal for the RHS ALU to determine whether (1) to trim the value down or to add it up into [0, +Q) or (2) do nothing because the value itself is within the range.

- I write only 1 ALU and instantialize it twice.
- Since the RHS ALU needs 3 states (add, sub, noop) for the control signal, the Sel signal in the ALU module is designed to have 2-bit, despite that in LHS it needs to express 2 states only.

The design itself is simple to describe provided that the inputs are all controlled (eg. Din1, Din2 are all positive numbers of 4-bit…).

- I found that the testbench has small bugs; I think in the 2nd initial block, it should be
  `$display("ERROR: %d + %d should be %d ,not %d\n",Din1,Din2,ALU_GOLD,TDout)` instead of
  `$display("ERROR: %d + %d should be %d ,not %d\n",Din1,Din2,GOLD,ALU_out)` (the 2 signals are misnamed).