2024 Digital IC Design

Homework 1: Modular Adder-Subtractor

1. Introduction:

The Modular Adder-Subtractor (MAS) is a combinational circuit that can perform addition and subtraction on the finite field (Galois field). In fact, when results are computed, these values should be bounded within the selected modular number, such as "14" and "-11" would be reduced to "1" and "2" with modular number "13". From a mathematical viewpoint, it respectively performs subtraction for exceeded value and addition for negative value. Different modular numbers may contribute to varying special reduction methods, however, in this homework, you are required to design a general solution to tackle this algorithm.

2. Specification:

2.1. The 2-input MAS

The logic diagram of the 2-input MAS in this homework is shown in Fig. 1, Fig. 2, and Fig. 3 with its I/O specification listed in Table I and Table II.

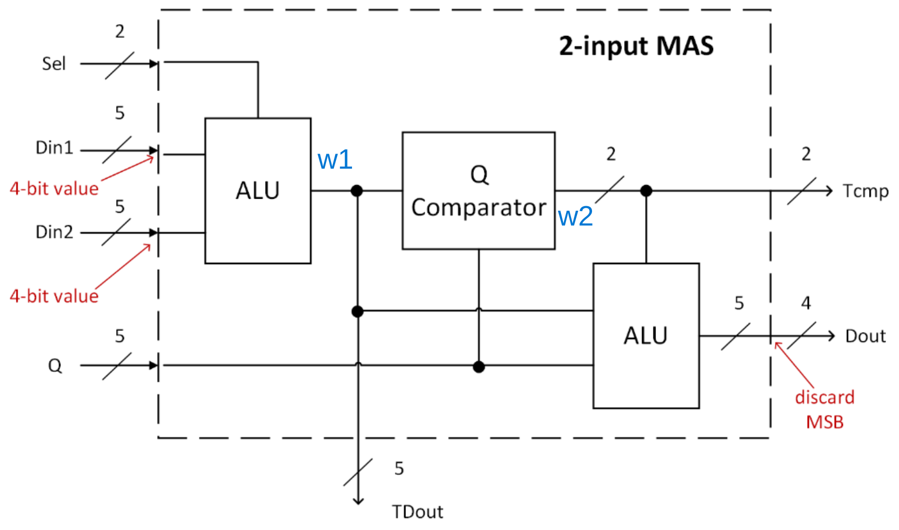


Fig. 1 The logic diagram of the 2-input MAS

Table I I/O specification of the 2-input MAS

Signal	I/O	width	Description
Din1	I	5	Input operand 1 (4-bit positive value)
Din2	I	5	Input operand 2 (4-bit positive value)
Sel	I	2	Operation select signal, when select is 00, ALU operates addition, while 11 operates subtraction. select 10, 01 stands for no operation.
Q	I	5	Input positive modular number (4-bit positive value)
TDout	O	5	Intermediate result (signed)
Tcmp	O	2	Decision result
Dout	O	4	Output result

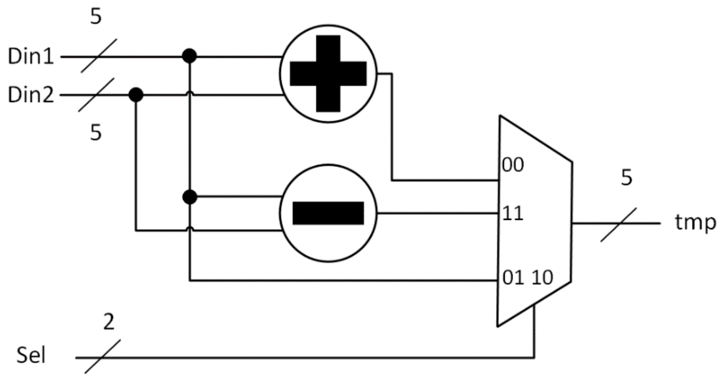


Fig. 2 ALU basic implementation

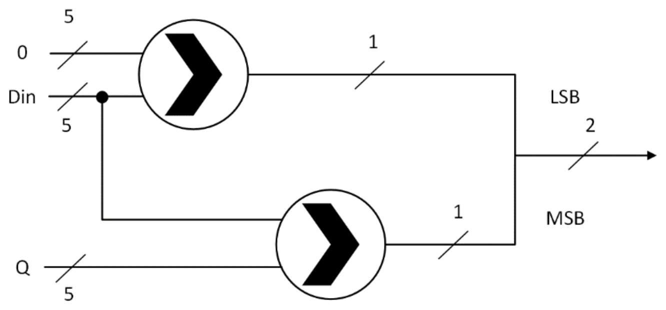


Fig. 3 Q comparator implementation

Table II The selection case of ALU

Select	operation
00	Din1 + Din2
11	Din1 – Din2
10, 01	Din1

This architecture includes 2 ALU and 1 comparator, with the first ALU output connected to the comparator and tailed with the other ALU. From an algorithm viewpoint, this general solution first computes the intermediate result of two operands. The value might be a negative number, less than or larger than one times the modular number, which could be decided by the Q comparator. After this, the other ALU is used to tune the intermediate value to the result. For your convenience, we summarized this algorithm in the following steps.

Step 1: Input values from testbench would be restricted to positive numbers of the finite field and compute the intermediate result of the two operands with the corresponding operation.

Step 2: Use the Q comparator to determine which operation the next stage would execute.

Step 3: Output the result.

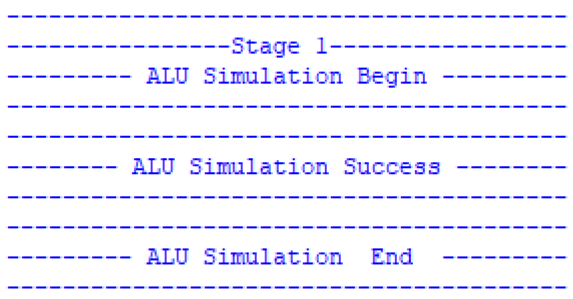
According to Fig. 3, one might find that the decision which Q comparator makes can support the next operation by letting the zero-comparison result be LSB and the Q-comparison result be MSB. (Note: In this way, the basic comparator unit should be set to "no less than".)

$$(Din1 \geq 0, and\ Din1 \geq Q)$$

3. Scoring:

3.1. ALU **[30%]**

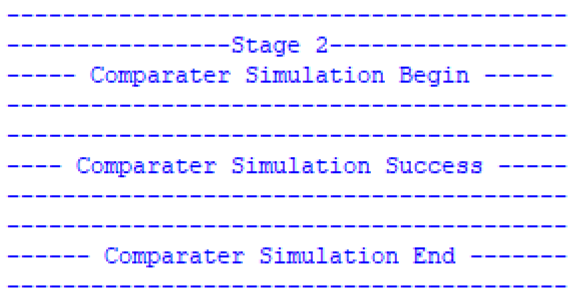
We will check the intermediate result from port “TDout”, the result should be generated correctly, and you will get the following message in ModelSim simulation.



```
----------------------------------------
----------------Stage 1-----------------
--------- ALU Simulation Begin ---------
----------------------------------------
----------------------------------------
-------- ALU Simulation Success --------
----------------------------------------
----------------------------------------
--------- ALU Simulation  End  ---------
----------------------------------------
```

3.2. Q comparator **[30%]**

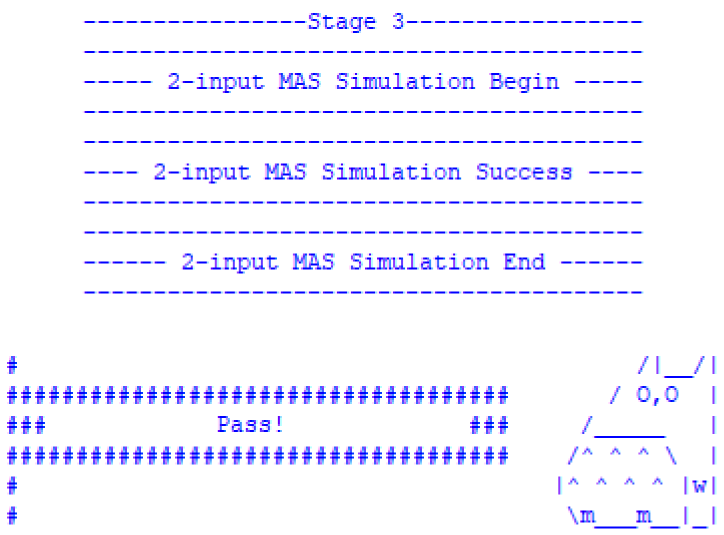
We will check the Q comparator result from port “Tcmp”, the result should be generated correctly, and you will get the following message in ModelSim simulation.



```
----------------------------------------
----------------Stage 2-----------------
----- Comparater Simulation Begin -----
----------------------------------------
----------------------------------------
---- Comparater Simulation Success -----
----------------------------------------
----------------------------------------
------ Comparater Simulation End -------
----------------------------------------
```

3.3. 2-input Modular Adder-Subtractor **[40%]**

We will check the result with different Q. Each test vectors follow the principle that Din1 and Din2 are restricted to a positive number bounded by Q. The result should be generated correctly, and you will get the following message in ModelSim simulation.



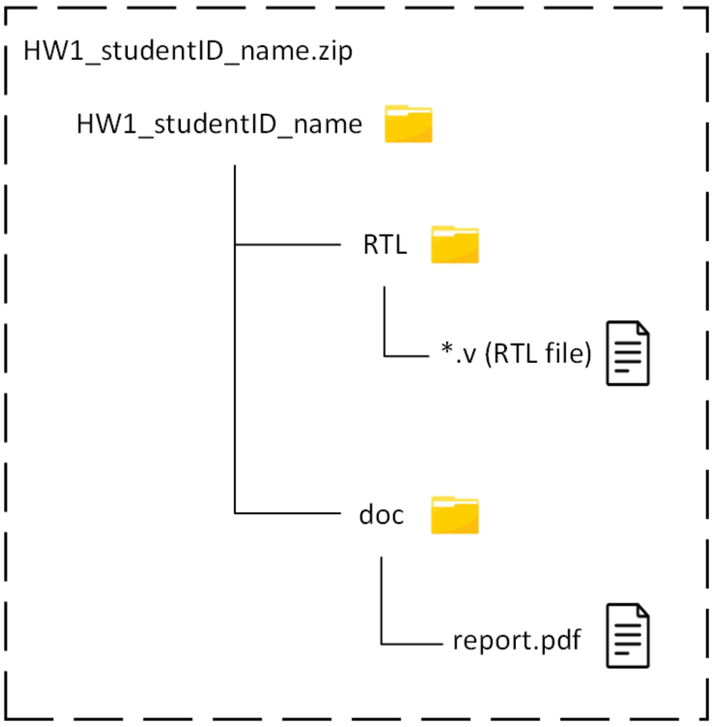
```
----------------Stage 3-----------------
----------------------------------------
----- 2-input MAS Simulation Begin -----
----------------------------------------
----------------------------------------
---- 2-input MAS Simulation Success ----
----------------------------------------
----------------------------------------
------ 2-input MAS Simulation End ------
----------------------------------------

#                                              /|__/|
####################################          / O,O  |
###            Pass!               ###       /_____  |
####################################        /^ ^ ^ \ |
#                                          |^ ^ ^ ^ |w|
#                                           \m___m__|_|
```

4. Submission

4.1. Submitted files

Please submit your files, followed by the illustration below. For example, HW1_P78121506_廖國佑.zip. (Note: if your operating system is MACOS, please remove the directory "__MACOSX" when compressing your files.



Report file

Please follow the specifications of the report. You are asked to describe how the circuit is designed as detailed as possible.

4.2. Note

Please submit your .zip file to folder HW1 in moodle.

Deadline: **2024/3/25 23:55**

If you have any problem, please contact TA by email

p78121506@gs.ncku.edu.tw

sr840112210@gmail.com