NYCU-EE IC LAB – Spring2023

Lab01 Exercise

Design: Chinese Course

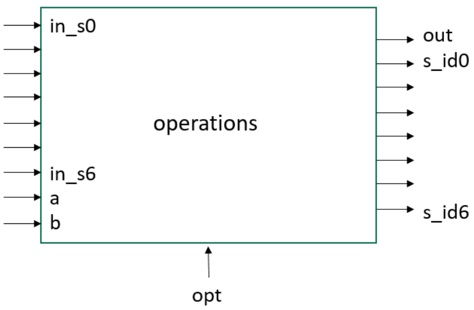
Data Preparation

1. Extract files from the TA directory:
   % tar xvf ~iclabta01/Lab01.tar
2. The extracted LAB directory contains:
   a. Practice/ : example code
   b. Exercise/ : your design

Design Description and Examples

At the end of the Chinese course at National Yang Ming Chiao Tung University, the teacher CC decided to entrust you with adjusting the scores of the seven students taking the course, after the score adjustment, the teacher CC would like to know how many students passed/failed the course and the ranking of the scores.

You will receive a sequence with **4-bit** 7 scores {**in_s0, in_s1, in_s2, in_s3, in_s4, in_s5, in_s6**}, a **3-bit** signal **opt**, a **2-bit** signal **a** and a **3-bit** signal **b**. Then you should do some operations in the following order to receive the number of passing students and ranking of the scores:



First, please do the 5 possible operations in the following order:

1. Signed/Unsigned	If **opt[0] is 1**, the 7 numbers will be regarded as **2's complement signed** values, which means that there MSB is signed bit. For example, in_n0=4'b1010, then its value is -6. in_n0=4'b0010, then its value is 2 If **opt[0] is 0**, the 7 numbers will be regarded as **unsigned** values. For example, in_n0=4'b1010, then its value is 10 in_n0=4'b0010, then its value is 2
2. Sort	If **opt[1] is 1**, sort the scores from the **largest to the smallest** . If **opt[1] is 0**, sort the sequence from the **smallest to the largest**. And output the corresponding student id (s_id0~s_id6) after sorting.

For example, if original scores are {2, -1, 3, 5, 5, 4, -3}

Corresponding student id : {0, 1, 2, 3, 4, 5, 6}

When opt[1] is 1

Order of scores becomes {5, 5, 4, 3, 2, -1, -3}.

Corresponding outputs **s_id0~s_id6**: {3, 4, 5 ,2 ,0 ,1 ,6}

When opt[1] is 0

Order of scores becomes {-3, -1, 2, 3, 4, 5, 5}.

Corresponding outputs **s_id0~s_id6**: {6, 1, 0, 2, 5, 3, 4}

original score	**2**	**-1**	**3**	**5**	**5**	**4**	**-3**
original student id s_id	0	1	2	3	4	5	6
Score after Sorting & opt[1]=1	5	5	4	3	2	-1	-3
Corresponding student id s_id	3	4	5	2	0	1	6
Score after Sorting & opt[1]=0	-3	-1	2	3	4	5	5
Corresponding student id s_id	6	1	0	2	5	3	4

Note. If encountering the same score, always output the student id in **ascending order**.

3. Calculate

Calculate the passing score.

1. Calculate average $\mu=\frac{\sum_{i=0}^{6} in\_Si}{7}$ (round down)

2. Calculate passing score=μ - a

For example, if original scores are {2, -1, 3, 5, 5, 4, -3}, a=1

Average $\boldsymbol{\mu} = \frac{[2+(-1)+3+5+5+4+(-3)]}{7} = 2$(round down)

Passing score $= \mu - a = 1$

Note. a is an input signal.

4. Linear-transformation	If the student's score is **negative,** adjust the score to $\frac{score}{a+1} + b$. Otherwise, adjust the score to $(a+1) * score + b$ e.g. original scores: {2, -1, 3, 5, 5, 4, -3} and parameter a = 1, b = 3 1st number of new scores: 2 * 2 + 3 =7 2nd number of new scores: (-1) / 2 (round down) + 3 = 0 + 3 = 3 3rd number of new scores: 3 * 2 + 3 = 9 4th number of new scores: 5 * 2 + 3 = 13 5th number of new scores: 5 * 2 + 3 = 13 6th number of new scores: 4 * 2 + 3 = 11 7th number of new scores: -3 / 2 (round down) + 3 = -1 + 3 = 2 After linear transformation, new scores: {7, 3, 9, 13, 13, 11, 2} Note. You should round-down first if $\frac{score}{a+1}$ is not integer.
5. Count	If **opt[2] is 1**, count the number of students who failed. If **opt[2] is 0**, count the number of students who passed. If one's score is smaller than the passing score, this student failed. Otherwise, this student passed. e.g. original scores: {2, -1, 3, 5, 5, 4, -3} and parameter a = 1, b = 3 After linear transformation, new scores: {7, 3, 9, 13, 13, 11, 2} and Passing score=1. Thus, if opt[2] = 1, output **out** equals to **0**. If opt[2] = 0, output **out** equals to **7**.

The summary of the description and specifications are as followings:

Input Signal	Bit Width	Description
in_s0	4	The score of the first student. If opt[0] is 0, in_s0 will be regarded as **unsigned** integer and ranged from **0~15.** If opt[0] is 1, in_s0 will be regarded as **2's complement signed** integer and ranged from **-8~7**.
in_s1	4	The score of the second student. If opt[0] is 0, in_s1 will be regarded as **unsigned** integer

		and ranged from **0~15.** If opt[0] is 1, in_s1 will be regarded as **2's complement signed** integer and ranged from **-8~7**.
in_s2	4	The score of the third student. If opt[0] is 0, in_s2 will be regarded as **unsigned** integer and ranged from **0~15.** If opt[0] is 1, in_s2 will be regarded as **2's complement signed** integer and ranged from **-8~7**.
in_s3	4	The score of the fourth student. If opt[0] is 0, in_s3 will be regarded as **unsigned** integer and ranged from **0~15.** If opt[0] is 1, in_s3 will be regarded as **2's complement signed** integer and ranged from **-8~7**.
in_s4	4	The score of the fifth student. If opt[0] is 0, in_s4 will be regarded as **unsigned** integer and ranged from **0~15.** If opt[0] is 1, in_s4 will be regarded as **2's complement signed** integer and ranged from **-8~7**.
in_s5	4	The score of the sixth student. If opt[0] is 0, in_s5 will be regarded as **unsigned** integer and ranged from **0~15.** If opt[0] is 1, in_s5 will be regarded as **2's complement signed** integer and ranged from **-8~7**.
in_s6	4	The score of the seventh student. If opt[0] is 0, in_s6 will be regarded as **unsigned** integer and ranged from **0~15.** If opt[0] is 1, in_s6 will be regarded as **2's complement signed** integer and ranged from **-8~7**.
opt	3	Operator for different mode. The operation will be encode as following: opt[0]: 1: Signed. 0: Unsigned opt[1]: 1: Sort from L->S. 0: Sort from S->L. opt[2]: 1: fail 0: pass
a	2	Parameter of linear transformation.

		Ranged from 0~3.
b	3	Parameter of linear transformation. Ranged from 0~7.

Output Signal	**Bit Width**	**Description**
out	3	If opt[2] is 0 , output the number of people who passed. If opt[2] is 1 , output the number of people who failed. Ranged from **0~7.**
s_id0	3	The student id corresponds to the first score after sorting. Ranged from 0~7.
s_id1	3	The student id corresponds to the second score after sorting. Ranged from 0~7.
s_id2	3	The student id corresponds to the third score after sorting. Ranged from 0~7.
s_id3	3	The student id corresponds to the fourth score after sorting. Ranged from 0~7.
s_id4	3	The student id corresponds to the fifth score after sorting. Ranged from 0~7.
s_id5	3	The student id corresponds to the sixth score after sorting. Ranged from 0~7.
s_id6	3	The student id corresponds to the seventh score after sorting. Ranged from 0~7.

Inputs& Outputs

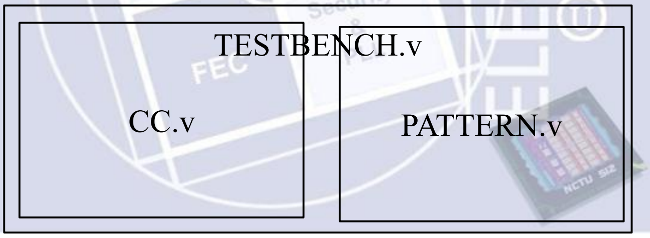
1. The input signals : **in_s0**, **in_s1**, **in_s2**, **in_s3**, **in_s4**, **in_s5** and **in_s6** are 4-bit inputs .
2. The input signal **opt** is a 3-bit input indicated whether to do the operations and which equation to use to get the final result and the input signal **a** and **b** are the parameters of linear transformation.

3. The output **out** is an unsigned number ranged from **0~7**. This represents the number of students passed/failed.
4. The output signals : **s_id0**, **s_id1**, **s_id2**, **s_id3**, **s_id4**, **s_id5**, **s_id6** are 3-bit outputs which represent the student ids.

Specifications

1. Top module name : CC (File name: CC.v)
2. Input pins : in_s0, in_s1, in_s2, in_s3, in_s4, in_s5, in_s6, opt, a, b
3. Output pins : out , s_id0, s_id1, s_id2, s_id3, s_id4, s_id5, s_id6

Block Diagram



Grading Policy

The performance is determined by the area of your design. The less area your design has, the higher grade you get. Try to reach better performance by thinking your architecture before coding.

Function Validity: 70%

Performance: area 30%

Note

1. Submit your design (CC.v) at **Lab01/Exercise/09_SUBMIT**
2. Submit your design through **Lab01/Exercise/09_SUBMIT/01_submit**
   a. 1st_demo deadline: 2023/02/27(Mon.) 12:00:00
   b. 2nd_demo deadline: 2023/03/01(Wed.) 12:00:00
3. If your file violates the naming rule, you will lose 5 points.
4. Don't use any wire/reg/submodule/parameter name called *error*, *congratulations*, *latch* or *fail* otherwise you will fail the lab. Note: * means any char in front of or behind the word. e.g: error_note is forbidden.

Be careful about all details!

Template folders and reference commands:

In RTL simulation, the name of template folder and reference commands is:

1. 01_RTL (RTL simulation):
   ./01_run
2. 02_SYN/ (Synthesis):
   ./01_run_dc
   (Check **latch** by searching the keyword "**Latch**" in 02_SYN/**syn.log**)
   (Check the design's timing in /Report/ CC.timing)
   (Check the design's area in /Report/ CC.area)
3. 03_GATE/ (Gate-level simulation):
   ./01_run
4. 09_SUBMIT/ (submit your files):
   ./01_submit
   ./02_check

You can key in **./09_clean_up** to clear all log files and dump files in each folder

Example Waveform

Input and output signal: