NCTU-EE IC LAB – Fall 2018

Lab01 Exercise

Design: Sort & Divider

Data Preparation

1. Extract files from TA's directory:
   % tar xvf ~train_ta/Lab01.tar

Design Description and Examples

"Win or Go Home!". At the final stage of NCTU Millionaire, you are asked to answer a question based on a series of simple mathematic operations. The only challenge is the remaining time, 20 ns. If you answer it in time correctly, you will win a prize of a million dollar. "Ready… Start…".

First, you will receive a sequence with 4 numbers {**in_n0, in_n1, in_n2, in_n3**} and a 1-bit **mode** signal. Then the possible operations are given in the following order:

- **Sort**:
  Sort the sequence from the largest to the smallest.
  For example, {2, 1, 3, 5} becomes {5, 3, 2, 1}.

After sorting, you will get a sequence **n0, n1, n2, n3**. Finally, the output answer can be obtained by one of the following modes (**remember all the numbers are unsigned**):

- mode0 : n0 divide n3 (you need to design divider by yourself)
- mode1 : (n0 - n1) + (n2 - n3)

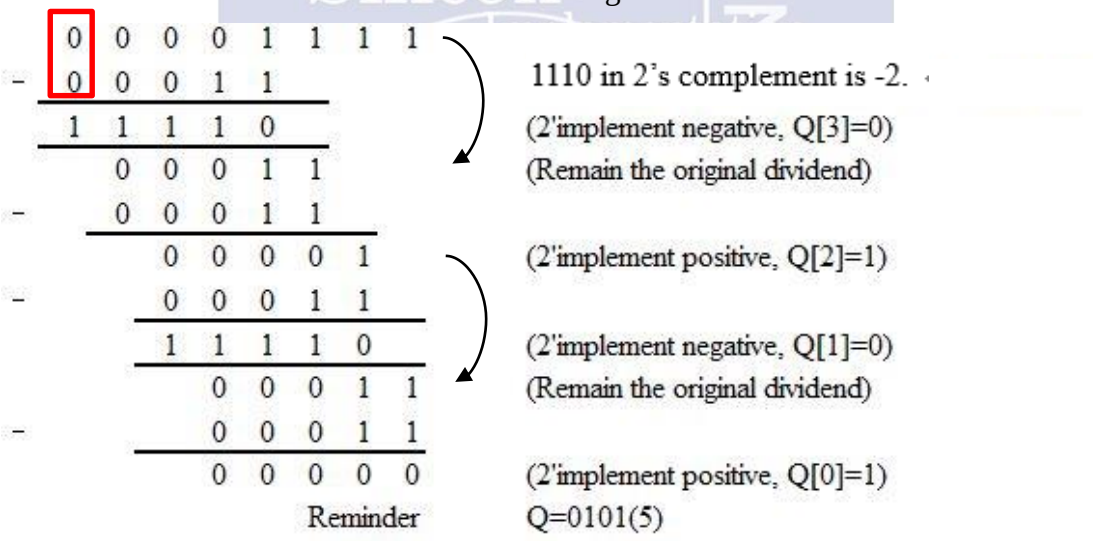
You have to design a divider by yourself in this lab. There is the divider design flow at the next page.

Divider Design Flow:

Step	Method
1	Subtract the signed-extension divisor, n3 from the most significant bit (MSB) of the dividend, n0.
2	Check the sign for the result of step 1. (1) If the result of step 1 is positive: a. Set the corresponding bit of the quotient, Q to 1. b. The result of step 1 is a new result. (2) If the result from step 2 is negative: a. Set the corresponding bit of the quotient, Q to 0. b. Remain the original dividend as a new result.
3	Repeat steps 1 and 2 until all bits of the quotient are determined.

Example: n0 = 1111(15), n3 = 0011(3)

Dividend and divisor should be extended signed bit



The summary of the description and specifications are as followings:

Input Signal	Bit Width	Description
in_n0	4	The first number of code. Ranged from **1~15**.
in_n1	4	The second number of code. Ranged from **1~15**.
in_n2	4	The third number of code. Ranged from **1~15**.
in_n3	4	The forth number of code. Ranged from **1~15**.
mode	1	Operator for different modes. The operation is encoded as above.

Output Signal	Bit Width	Description
out_n	4	The answer. Ranged from 0~15

Examples:

1. **Initial numbers {13, 2, 6, 11} with mode = 1'b0:**

   $\{13, 2, 6, 11\} \ -(sort)-> \ \{13, 11, 6, 2\} \ -(mode0)-> \ 6$

2. **Initial numbers {5, 14, 14, 3} with mode = 1'b0:**

   $\{5, 14, 14, 3\} \ -(sort)-> \ \ \{14, 14, 5, 3\} \ -(mode0)-> \ 4$

3. **Initial numbers {12, 3, 6, 14} with mode = 1'b1:**

   $\{12, 3, 6, 14\} \ -(sort)-> \ \ \{14, 12, 6, 3\} \ -(mode1)-> \ 5$

4. **Initial numbers {8, 8, 15, 4} with mode = 1'b1:**

   $\{8, 8, 15, 4\} \ -(sort)-> \ \ \{15, 8, 8, 4\} \ -(mode1)-> \ 11$

Inputs

1. The input signals **in_n0, in_n1, in_n2,** and **in_n3** are 4-bit inputs ranged from 1 to 15.
2. The input signal mode is a 1-bit inputs indicated which equation to use to get the final result.

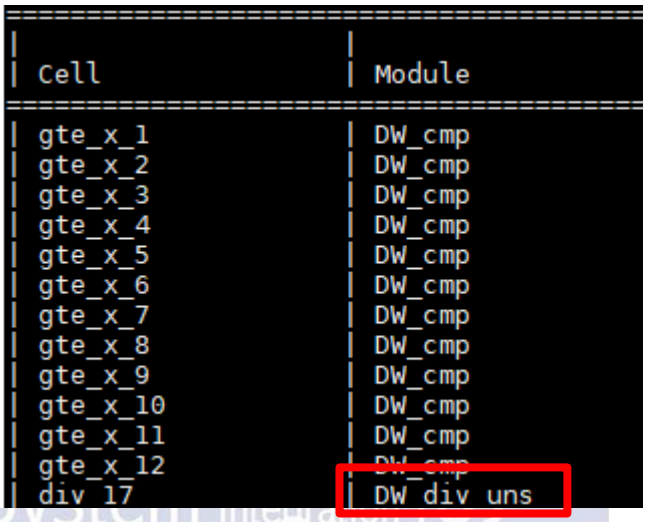
Outputs

The output signal **out_n** is a signed number ranged from 0 to 15. This represents the correct password.

Specifications

1. Top module name : SD (File name: SD.v)
2. Input pins : in_n0, in_n1, in_n2, in_n3, opt
3. Output pins : out_n0
4. After synthesis, check the "SD.area" and "SD.timing" in the folder "Report". The area report is valid only when the slack in the end of "SD.timing" is non-negative.
5. The synthesis result **cannot** contain any **latch**.
   Note: You can check if there is a latch by searching the keyword "**Latch**" in 02_SYN/**syn.log**

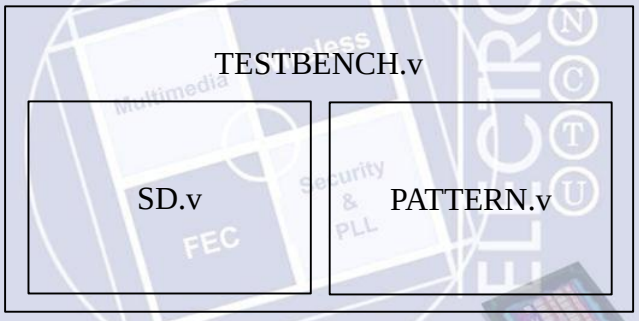
In this lab, you should design your own divider. You **cannot** use **"/"** operator here. TAs will check the "SD.resource" in the folder "Report". It **cannot** contain any divider IP from Designware as shown in the below figure.



Cell	Module
gte_x_1	DW_cmp
gte_x_2	DW_cmp
gte_x_3	DW_cmp
gte_x_4	DW_cmp
gte_x_5	DW_cmp
gte_x_6	DW_cmp
gte_x_7	DW_cmp
gte_x_8	DW_cmp
gte_x_9	DW_cmp
gte_x_10	DW_cmp
gte_x_11	DW_cmp
gte_x_12	DW_cmp
div_17	DW_div_uns

6. You also cannot use the for function in your design.
   TAs will search the keyword for in your SD.v file.

Block Diagram



Grading Policy

The performance is determined by the area of your design. The less cost your design has, the higher grade you get.

Function Validity: 75%

Performance: 25% (area: 25%)

Note

1. Template folders and reference commands:

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

   01_RTL:

   "**./01_run"**

   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/SD.timing)

   03_GATE_SIM/:

   ./01_run

Example Waveform

Input and output signal: