CSIE DIC Lab 2024

Lab03B. Compare areas of right-angled triangles

Description

In this lab, students must create three essential elements: an unsigned multiplier, a shifter, and a comparator. These elements are crucial for determining the area of right-angled triangles and comparing the magnitudes of these areas, as depicted by the formula shown in Figure 1.

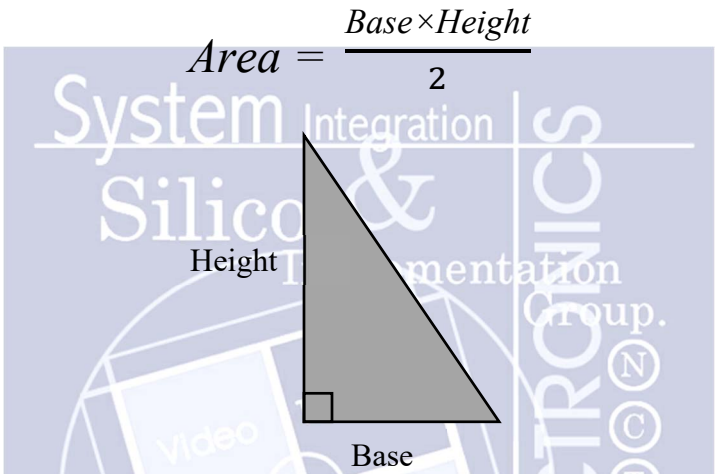


Figure 1: Calculation for right-angled triangles area.

Upon determining the areas of two right-angled triangles, a comparative analysis is conducted to identify which triangle is larger. Following this, the circuit outputs the larger triangle's area along with its hypotenuse's length, as illustrated in Figure 2.

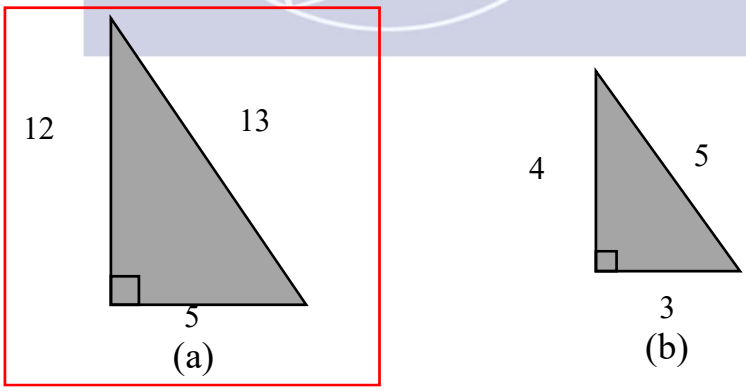


Figure 2: Compare the areas of right-angled triangles.

Please refer to Figure 3 for an understanding of the circuit computation process. Initially, the circuit receives six inputs: ***a***, ***b***, ***c***, ***d***, ***e***, and ***f***. Here, ***a***, ***b***, and ***c*** represent the base, height, and hypotenuse of the first triangle, respectively, while ***d***, ***e***, and ***f*** correspond to those of another triangle. Subsequently, an unsigned multiplier is

employed to calculate the products of the bases and heights for each triangle. Following this, a shifter is used to right-shift the output of the unsigned multiplier by one bit. Finally, a comparator determines which right-angled triangle has the larger area. The output ultimately displays the area (**out[16:0]**) of the triangle with the greater area and its corresponding hypotenuse length (**max[8:0]**).

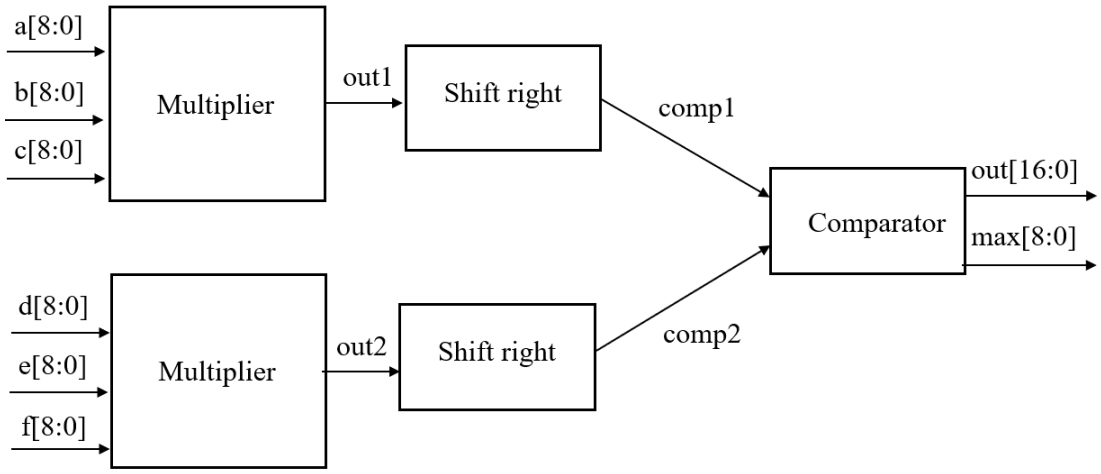


Figure 3: Circuit block diagram

Specification

1. Input: ***a[8:0], b[8:0], c[8:0], d[8:0], e[8:0], f[8:0]***.
2. Output: ***out[16:0] , max[8:0]***.
3. It is important to note that the entire design for this lab **MUST** be implemented using Verilog HDL structural-level descriptions. The use of Verilog HDL behavior expressions is strictly prohibited. Additionally, the design **MUST** be implemented **using only standard cells**.

Lab Instructions

1. Extract LAB data from TA's directory:
   % tar xzvf ~dicta/lab03b.tar.gz
2. The extracted LAB directory (**lab03b**) contains:
   a. **TEST.v** : Design test bench
   b. **lab03b_beh.v** : Behavioral-level reference design
   c. **lab03b.v** : Empty gate-level design
   d. **TSMC_ADFP_stdcells.pdf** : Standard Cell Information
3. Change directory to **lab03b**
   % cd lab03b
4. Before proceeding with the lab exercise, it is recommended that students first run

the Verilog simulation with the encrypted reference design (**lab03b.vp**). This protected Verilog module serves as a valuable reference tool for students to understand better the input and output timing diagram associated with the lab.

% ncverilog -f run.f

5. Once the Verilog simulation with the encrypted reference design has been run, the next step is to open the simulation waveform " **lab03b.fsdb**" using nWave.

   % nWave &

   To open the simulation waveform in nWave, begin by selecting "**File**" from the toolbar, followed by "**Open**" and then "**lab03b.fsdb**." This will load the simulation waveform into the nWave. Next, select "**File**" from the toolbar once again, followed by " **Restore Signal**" and then "**lab03b.rc**." This will load the selected signals from the **lab03b.fsdb** and streamline the debugging process for students. Therefore, we highly recommend using the saved signal file (**lab03b.rc**) to facilitate the debugging process of the lab exercise.

6. Students must write their Gate-level HDL code in **lab03b.v**. This code will be used to calculate the trapezoidal area as outlined in the lab instructions.

   % gedit lab03b.v &
   or % gvim lab03b.v &
   or % joe lab03b.v
   or % vim lab03b.v

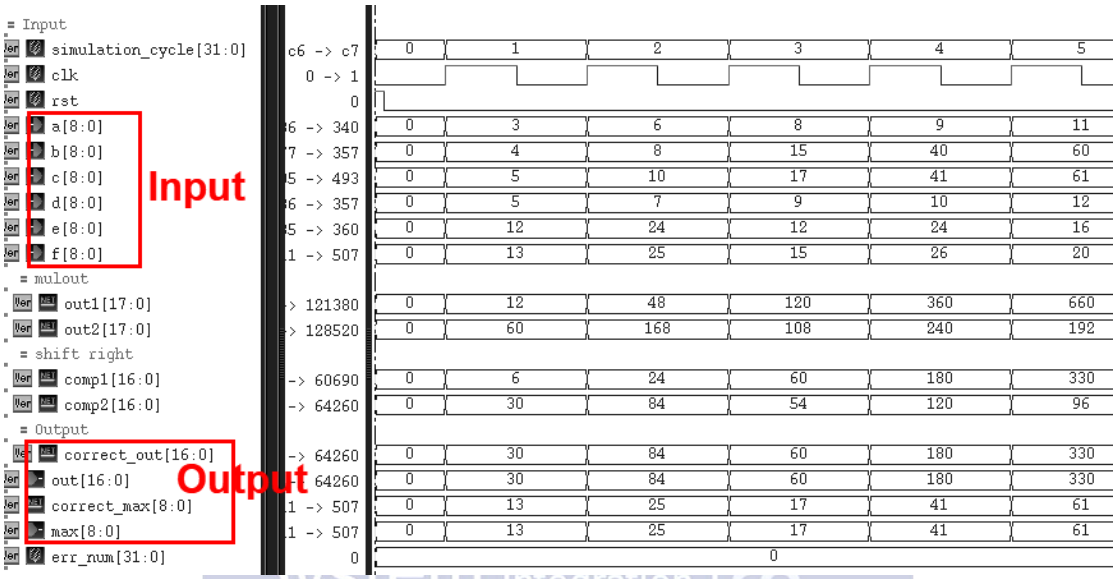
   As part of the lab exercise, students are required to modify **lab03b.v** and add their Gate-level HDL code to the file. Once the Gate-level HDL code has been added to **lab03b.v**, students must run NC-Verilog to re-simulate the design and ensure its proper functionality.

7. When designing and implementing Gate-level HDL code for the lab exercise, students **MUST** only use standard cells from the Verilog model file "**N16ADFP_StdCell.v**."

8. Once the Gate-level HDL design has been completed in **lab03b.v**, the next step is to re-run the Verilog simulation with the new design. Before re-running the simulation, however, it is important to change the include file in TEST.v from **lab03b.vp** to **lab03b.v** to ensure that the new design is properly integrated into the simulation.

   % ncverilog -f run.f

9. Example simulation waveform:



The simulation waveform shows variables ***a[8:0]***, ***b[8:0]***, ***c[8:0]***, ***d[8:0]***, ***e[8:0]***, and **f[8:0]** in an "unsigned" decimal format. This waveform also presents ***correct_out[16:0]***, **correct_max[16:0]**, **out[16:0]**, and **max [8:0]**, also in "unsigned" decimal format. Such a display aids in a more effective analysis of the Gate-level HDL design's output. Within TEST.v, the Gate-level HDL design's outcomes are compared with the behavior-level reference design, serving to confirm its functionality.