CSIE DIC Lab 2023

Lab03b. Calculate the Area of a Trapezoid

Description

In this lab, students are required to design three key components: an adder, an unsigned multiplier, and a shifter. These components will be used in conjunction to calculate the trapezoidal area, and the formula is shown in Figure 1.

$$Area = \frac{(a+b) \times c}{2}$$

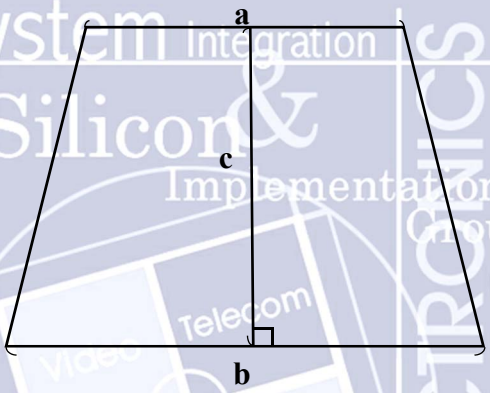


Figure 1: Calculation for trapezoid area.

To better understand the functionality of the circuit being designed in this lab, refer to Figure 2, which outlines the example computational process. First, the circuit takes three input data points labeled **a**, **b**, and **c**, corresponding to the trapezoid's top line, baseline, and height. Next, the circuit calculates the sum of **a** and **b** using an adder. The output of the adder is then multiplied by **c** using an unsigned multiplier. Finally, the resulting product (**mulout**) is shifted to the right by one bit using a shifter. The final output of the circuit is the area of the trapezoid.

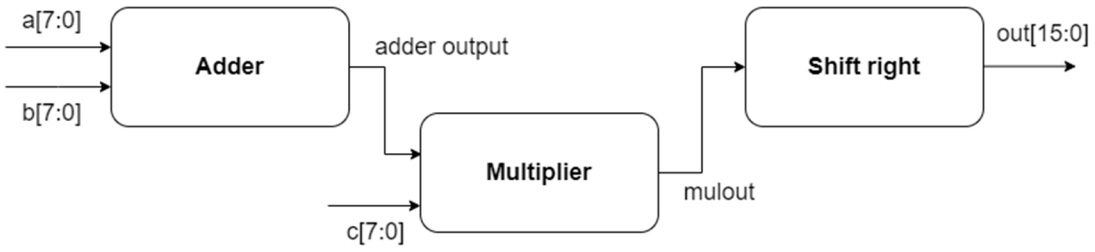


Figure 2: Circuit block diagram

Specification

1. Input: ***a[7:0], b[7:0], c[7:0]***
2. Output: ***out[15:0]***.
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

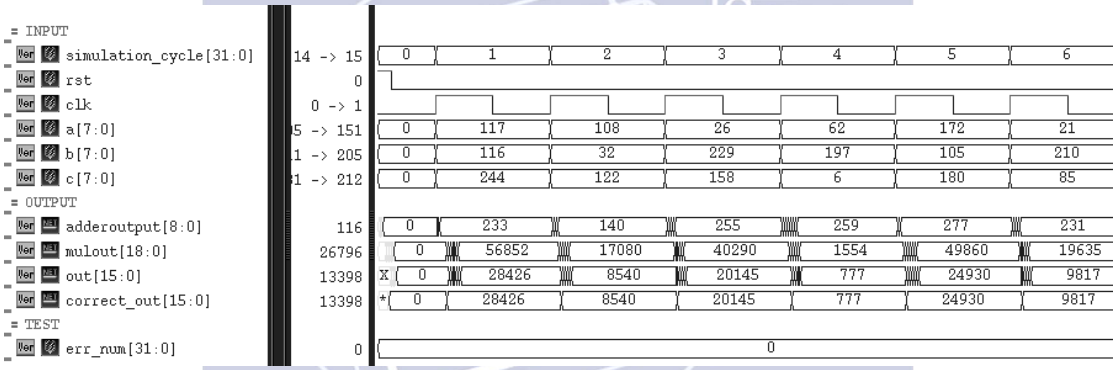
1. Extract LAB data from TA's directory:
   % tar xzvf ~dicta/lab03b.tar.gz
2. The extracted LAB directory (**lab03b**) contains:
   a. **TEST.v** : Design test bench
   b. **lab03b_beh.v** : Behavioral-level reference design
   c. **lab03b.v** : Empty gate-level design
3. Change directory to **lab03b**
   % 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 "**umc18_neg.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 example simulation waveform associated with the lab exercise displays ***a[7:0]***, ***b[7:0]***, and ***c[7:0]*** in "unsigned" decimal format. Additionally, the waveform displays ***correct_out [15:0]*** and ***out [15:0]*** in "unsigned" decimal format, allowing students to analyze the output of their Gate-level HDL design more effectively. In TEST.v, the results of the Gate-level HDL design will be compared with the behavior-level reference design to verify its functionality.