IC Design

Homework # 4

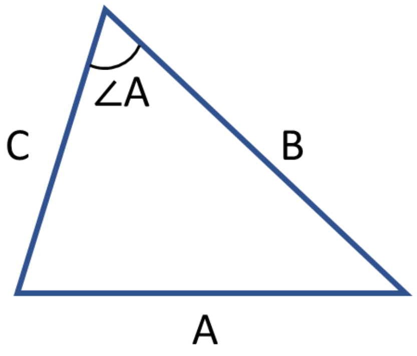
(Due on 2020/01/03, 14:00, Verilog code and report upload to CEIBA, no hardcopy)

✧ Plagiarism is not allowed. 10% penalty for each day of delay.

✧ (Ex: Before 2020/01/04, 13:59, 10% penalty.)

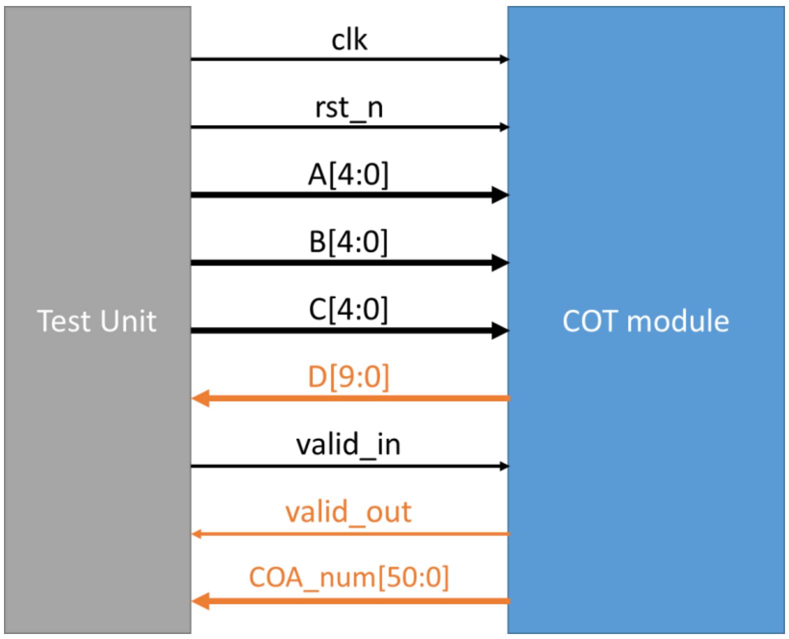
✧ Any further questions, you can leave messages on the board of the class website or send e-mail to the TA.

Specifications

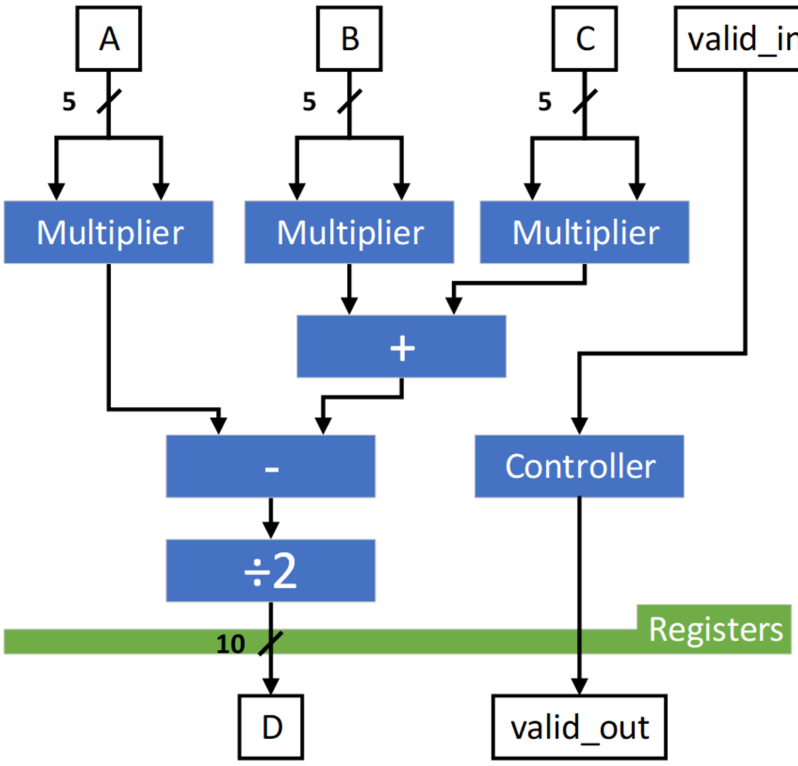


$$D = BC\cos\angle A = \frac{B^2 + C^2 - A^2}{2}$$

Design a calculation circuit with reset that computes the **Cosine of an Angle**. There are three inputs, i.e., A with 5 bits, B with 5 bits, and C with 5 bits. And there is one output D with 10 bits. Note that all the inputs and output are **unsigned integers**. The relation between input and output is

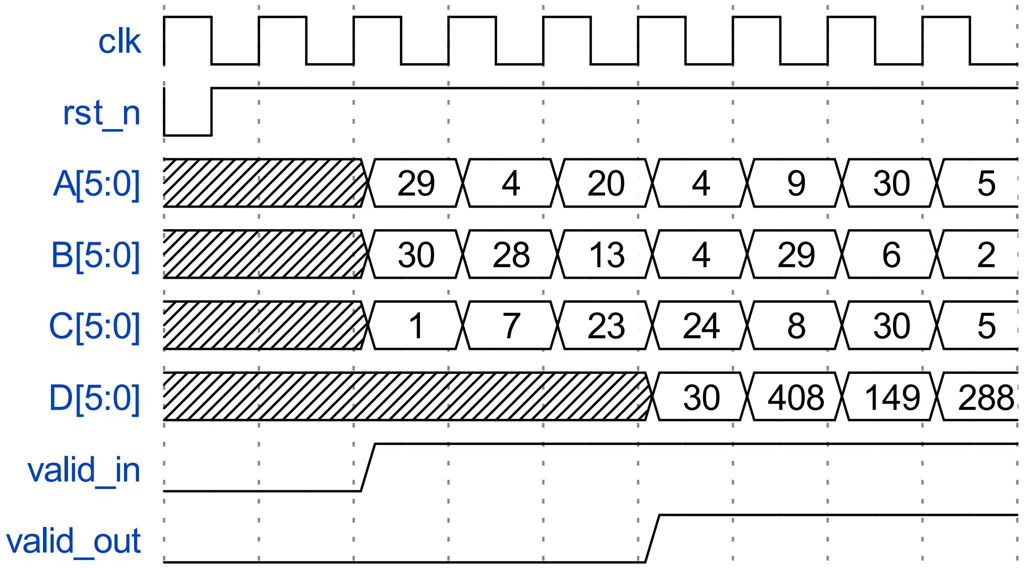


Note that the two output signals: “D[9:0]” and “valid_out” must be registered, i.e., they are outputs of DFFs (**use module FD2 (positive edge)** in lib.v). A possible architecture is as follows (not the best design):



COA Timing Diagram

The following figure is a possible timing diagram of the pipeline architecture. After a certain delay time, the COA module will continue to output results.



Handshake signals

Signal name	**I/O**	**Simple description**
valid_in	Input	When valid_in == 1, Test_Unit will send a set of data (A, B, C) to the COA module.
valid_out	Output	The COA module needs to generate “valid_out” by itself. Once valid_out == 1, Test_Unit treats the data (D) as valid and starts receiving data (D).

Data truncation

Since you need to divide by 2 at the end, you need to truncate the data to 10-bit. Here are some examples.

Signal	Value	
A	29	4
B	30	28
C	1	7
A^2	841	16
B^2	900	784
C^2	1	49
$B^2 + C^2 - A^2$	60	817
D[10:0]	30	408.5
D[9:0]	30	408

Design Rules

- **Those who do not design according to the following rules will not be graded.**
- There should be a **reset signal** for the register.
- You may need to add pipeline registers to speed up design.
- You can loosen your simulation timing first, (i.e., **`define HALF_CYCLE XXXX** in the testbench.v), then shorten the clock period to find your critical path.
- Your design should be based on the **standard cells in the lib.v**. All logic operations in your design **MUST consist of the standard cells** instead of using the operands such as “+”, “-”, “&”, “|”, “>”, and “<”.
- Design your homework in the given “COA.v” file. **You are NOT ALLOWED to change the filename and the header of the top module (i.e. the module name and the I/O ports)**.
- If your design contains more than one module, **don’t create new file for them**, just put those modules in “COA.v.”

Grading Policy

1. Gate-level design using Verilog (70%)

Your score will depend on both the correctness and performance of your design.

(a) Correctness Score (40%)

At this stage, we will only evaluate whether the function of the COA module is correct. Time and area are not considered. We provide a test bench with 1000 patterns which automatically grades your design. Your score in this part will be

$$40 \times \frac{correct\ number}{1000}$$

(b) Performance Score (30%)

At this stage, you need to add up the number of transistors of all used cells in the COA module and connect it to COA_num[50:0].

Only in this section, you are allowed to use "assign" and "+" to help calculations. (i.e., **assign COA_num = AD3_N + AN4_N;**)

We will rank all students who pass (a) and have **no connection errors on COA_num**. There will be a ranking according to **AT**, where A represents the **number of transistors** and T represents the **execution time**. Your performance score will be according to your ranking as the table below.

Percentage of passing students	Performance Score
If your ranking > 90 %	30
80% ~ 90%	27
70% ~ 80%	24
60% ~ 70%	21
50% ~ 60%	18
40% ~ 50%	15
30% ~ 40%	12
20% ~ 30%	9
10% ~ 20%	6
0% ~ 10%	3
Using operands, not standard cell logic	0
Correctness failed	0
Plagiarism	0

2. Report (30%)

(a) Simulation (0%)

Write down your minimum half-cycle time. If you do not provide this information, "1. Gate-level design using Verilog (70%)" **will not get any score.**

This minimum half-cycle time would be verified by TAs.

(c) Circuit diagram (10%)

You are encouraged to use software to help draw architecture **instead of hand drawing**. Plot it **hierarchically** so that readers can understand your design easily. **All of the above will improve your report score.**

(5%) Plot the gate-level circuit diagram of your design **using a drawing tool instead of by hand**.

(5%) Plot critical path on the diagram above.

(c) Discussion (20%)

Discuss about your design.

- ➢ (3%) Introduce your design.
- ➢ (2%) How do you cut your pipeline?
- ➢ (5%) How do you improve your critical path and the number of transistors?
- ➢ (5%) How do you trade off between area and speed?
- ➢ (5%) Compare with other architectures you have designed (if any).

Notification

Following are the files you will need (available on the class website)

HW4.zip includes

- ■ **HW4_2019.pdf:** This document.
- ■ **HW4_tutorial_2019.pdf:** Tutorial in class.
- ■ **COA.v:**

  Dummy design file. Program the design in this file.

  The header of the top module and the declaration of the I/O ports are predefined in this file and you are not allowed to change them.
- ■ **lib.v:** Standard cells.
- ■ **testbench.v:**

  Test bench for your design.
- ■ **A.dat, B.dat , C.dat:**

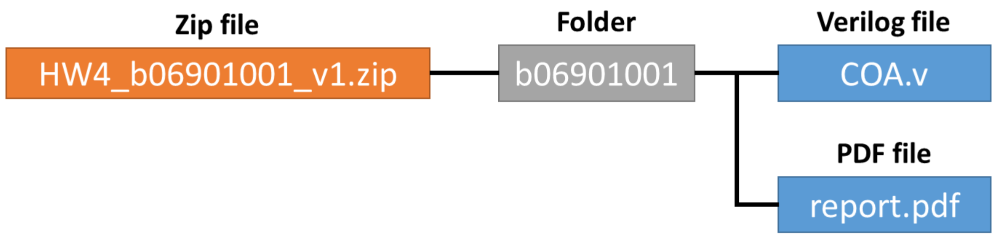
  Input patterns for test bench. Please put these files in the folder that contains testbench.v when doing simulation.
- ■ **answer.dat:**

  Output patterns of correct answers for test bench. Please put the file in the folder that contains testbench.v when doing simulation.

Submit files

All students who do not submit files according to the rules **will get 20% penalty.**

- You should upload a **zip file** to **CEIBA**, the file name is "HW4_Student ID_vx", vx represents the version you submitted. Ex: HW4_b06901001_v1.zip
- Your file must conform to the following structure.



Testbench description

- The output waveform will be dumped to file "COA.fsdb", you can use nWave to examine it.
- You can change the number of test data to debug, but the final score will still test 1000 sets of data. (`**define N 1000**)
- You can enable the debug function, which will display the data sent and received. (`**define debug 0**)

TA email: r07943012@ntu.edu.tw, EE2-329

HW4 Office hours: 2019/12/30 (Mon) 19:00~21:00 @BL212

2019/01/02 (Thus) 19:00~21:00 @BL211

If you have no time at office hours, you can email TA to discuss another time for appointment.