National Tsing Hua University
Department of Electrical Engineering
EE429200 IC Design Laboratory, Fall 2021

Lab 09-2: ECO-Integer Division (2%)
Assigned on Nov 11, 2021
Due day on Nov 18, 2021

Objective

In this lab, you will learn:

1. How to do ECO to correct the bugs in synthesized netlist.

Demo checklist:

- ☐ (Mission 1) The modified *inverse_table.v*. [Only two lines shall be changed compared to the original one, and it should pass the simulation using *test_my_div.f* but fail using *test_my_div_buggy.f*]
- ☐ (Mission 2) Synthesized netlist *my_div_syn.v*, its area and timing report, and the log file of LEC for verifying the equivalence between the design in *test_my_div.f* and this netlist.
- ☐ (Mission 3) The corrected RTL *inverse_table_corrected.v*.
- ☐ (Mission 4) The ECO-modified netlist *my_div_syn_corrected.v* and the log of LEC for verifying the equivalence between the design in *test_my_div_corrected.f* and this netlist.

Environment Setup

Copy lab file packages from ee4292. Decompress the package and enter it. You can check the file list in Appendix.

```
$ cp ~ ee4292/iclab2021/lab09.zip .
$ unzip lab09.zip
$ cd lab09_part2/
```

Note: please run simulation in the ***sim*** directory to maintain a fine data management.

Description

You are a digital designer for a world-wide famous IC design house: NTHUTek. One day, when you walked in the company, you were told that one of your colleagues was fired because his design in a flagship project completely failed. Your boss asked you to take charge of this design. So, try your best to fix the problem (and avoid getting fired)! Good luck!

The function of this design, *my_div*, is non-negative integer division as shown in Fig.

1. For 16-bit input *dividend* and 5-bit input *divisor*, this block generates a 16-bit integer quotient by converting the division into multiplication by

$$\frac{dividend}{divisor} = dividend \times \frac{2^{div\_shift}}{divisor} \times \frac{1}{2^{div\_shift}} = (dividend \times div\_inverse) \gg div\_shift.$$

The module *inverse_table* is responsible for the table look-up for *div_inverse* and *div_shift*, and the *mul_and_shift* for the multiplication and shifting.

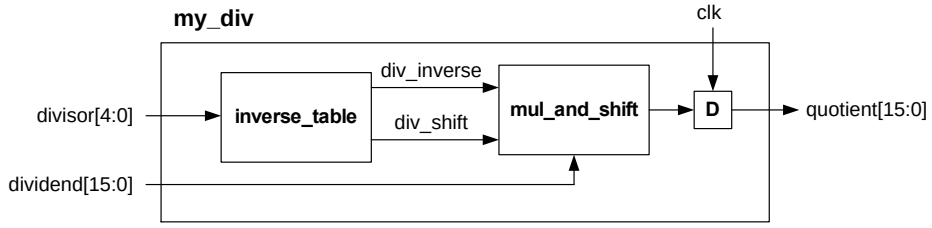

Fig. 1. Block diagram of *my_div*

Action Items

Mission 1：

After reading the RTL codes of your lazy and careless ex-colleague, you found that he didn't prepare a solid testbench to test this module and this is why the design completely fails. So, you wrote a testbench Verilog code *test_my_div.v* and found there were two bugs in *inverse_table.v*. Please fix them by modifying the RTL code *inverse_table.v* directly.

Mission 2：

You need to handover your netlist, *my_div_syn.v*, to the backend team. Please run logic synthesis using the synthesis script prepared by the backend team, and also run LEC to verify the logic equivalence. Then you can enjoy your long vacation!

Mission 3：

Today, three months later since Mission 2, it suddenly occurred to you that your testbench DOES NOT cover all possible inputs. The 16-bit *dividend* should range from 0 to 65535, instead of only 0 to 32767 in *test_my_div.v*. You then quickly modified a new testbench (*test_my_div_corrected.v*), ran the simulation, and prayed. Now, the infamous story of "Pentium FDIV bug" starts replaying in your mind because you got hidden bugs in YOUR *inverse_table.v*. Fix the bugs and save the code to *inverse_table_corrected.v*.

Mission 4：

Unfortunately, the backend team doesn't allow you to resynthesize your RTL code. Instead, they asked you to do ECO using only three kinds of available spare cells: NAND, NOR, and INV. You have no choice, so please do ECO and use LEC to verify the logic equivalence between the correct RTL codes and the modified netlist *my_div_syn_corrected.v*.

**Only three standard cells are allowed：

Inverter：INVX0_HVT (A, Y); //Y=~A

NAND：NAND2X0_HVT (A, B, Y); //Y=~(A1&A2)

NOR：NOR2X0_HVT (A, B, Y); //Y=~(A1|A2)

Appendix

Directory	**Description**
source/	Your ex-colleague's RTL codes: *my_div.v*, *inverse_table.v*, and *mul_and_shift.v*. Put your modified *inverse_table.v* and *inverse_table_corrected.v* here for running simulation.
sim/	Testbench and file lists (you can't modify them) - *test_my_div.f* for Mission 1; *test_my_div_buggy.f* for Mission 3 (buggy RTL); *test_my_div_corrected.f* for Mission 3 (corrected RTL).
syn/	Synthesis environment prepared by the backend team. (you can't modify them) Run synthesis using the batch file *./run_dc.bat*.