IC Design

Homework # 4

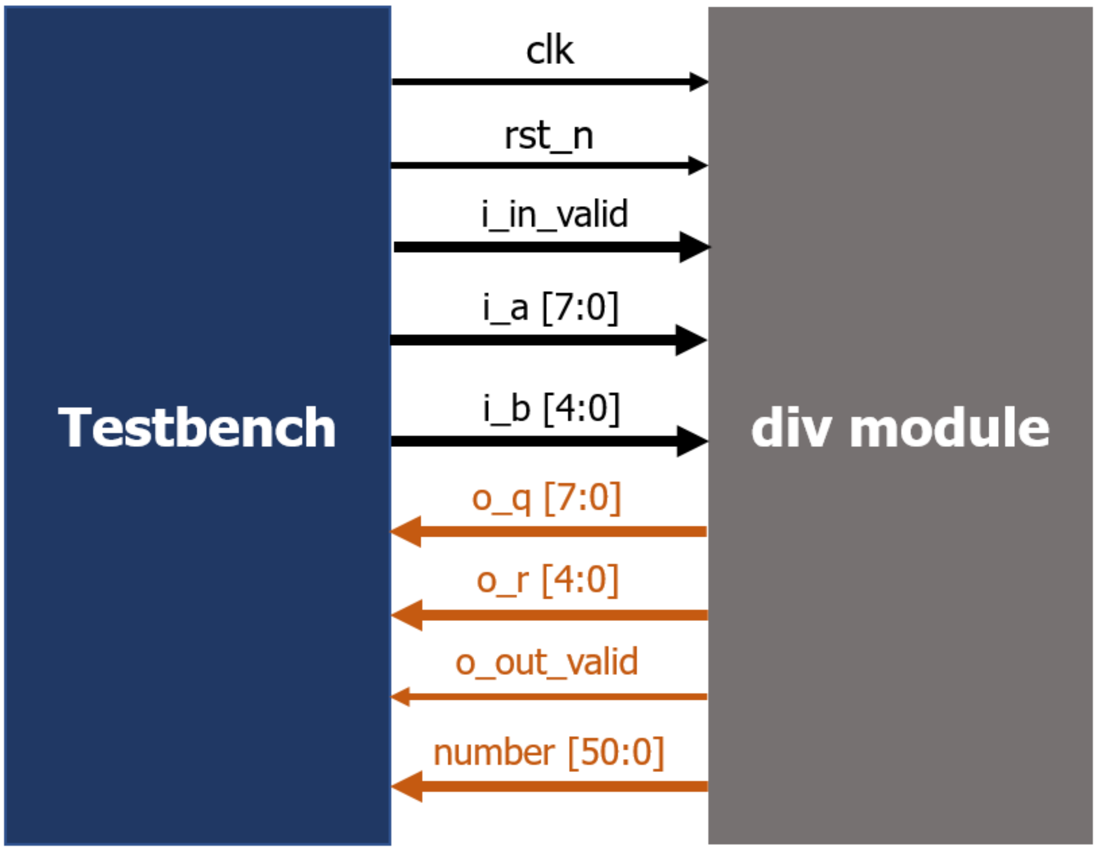
(Due 2021/12/30, 9:00, Verilog code and report upload to CEIBA, no hardcopy)

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

Problem Specification

Design a divider circuit with reset that computes the **quotient Q and reaminder R of two numbers A and B**. There are three input signals for the circuit, i.e., i_a with 8 bits, i_b with 5 bits and a 1-bit i_in_valid. The circuit contains three output signals, i.e., 8-bit o_q for the quotient, 5-bit o_r for the remainder and a 1-bit o_out_valid. Note that all the input and the output signals are **unsigned integers.** The divisor B is guaranteed to be **non-zero**. The relation between the input and the output signals is

$$A = Q * B + R.$$



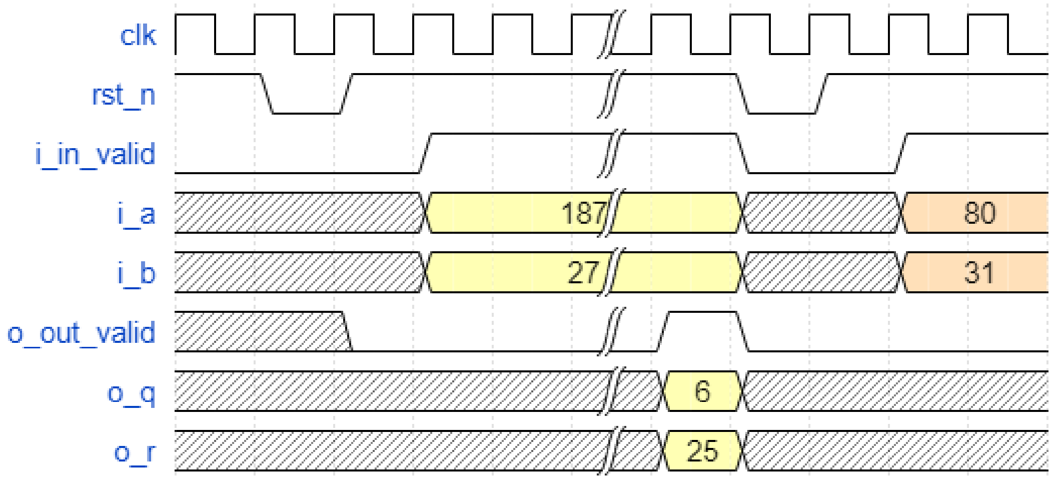
Pull up "o_out_valid" and output your answer to "o_q[7:0]" and "o_r[4:0]" once your circuit finish the calculation, the testbench will then check your answers. Note that the output signals: "o_q [7:0]" , "o_r[4:0]" and "o_out_valid" must be registered, i.e., they are outputs of DFFs **(use module FD2 (positive edge) in lib.v)**.

Timing Diagram

Since a design can be either recursive or pipelined, the testbench provides two input strategies, i.e., recursive and pipeline.

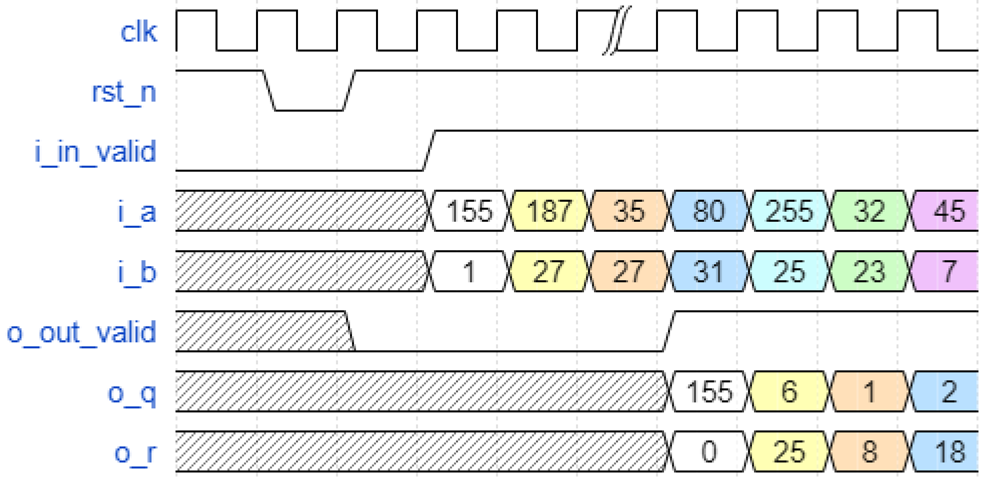
1. Recursive

In the recursive input strategy, **the circuit would be reset before every new input**.



2. Pipeline

In the pipeline input strategy, the circuit would only be reset once. After reset, **new values of i_a and i_b would be input at every cycle**. Besides, **o_out_valid should maintain high once it was pulled up**, and the order of input a and b and their corresponding quotient and remainder should not be changed (**First in, first out**.).



Signals Description

Signal name	I/O	Width	Simple description
clk	Input	1	Clock signal.
rst_n	Input	1	Active low asynchronous reset.
i_in_valid	Input	1	Indicate that the input is valid
i_a	Input	8	Dividend
i_b	Input	5	Divisor
o_q	Output	8	Quotient
o_r	Output	5	Remainder
o_out_valid	Output	1	Indicate that the calculation was finished.
number	Output	51	The number of transistors.

Design Rules

Those who do not design according to the following rules will not be graded.

- **LUT-based designs are not allowed.**
- There should be a **reset signal** for the register.
- You are free to add pipeline registers.
- You can loosen your simulation timing first, (i.e., **`define CYCLE XXXX** in the tb.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 "<".
- Using "assign" to concatenate signals or specify constant values is allowed.
- Design your homework in the given "div.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 a new file for them**, just put those modules in "div.v."

Grading Policy

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

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

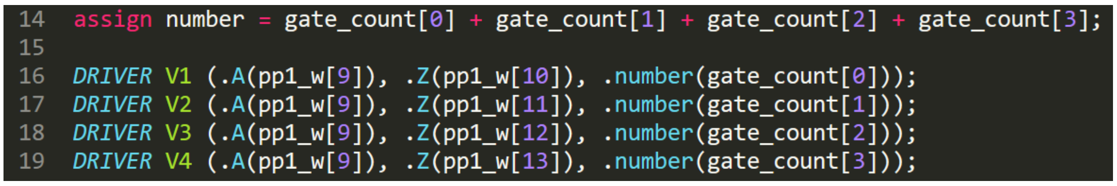
(a) Correctness Score (40%)

At this stage, we will only evaluate whether the function of the divider module is correct. Time and area are not considered. We provide a testbench 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 div module and connect it to number [50:0]**.
Only in this section, you are allowed to use "+" to help with calculations.



```
assign number = gate_count[0] + gate_count[1] + gate_count[2] + gate_count[3];

DRIVER V1 (.A(pp1_w[9]), .Z(pp1_w[10]), .number(gate_count[0]));
DRIVER V2 (.A(pp1_w[9]), .Z(pp1_w[11]), .number(gate_count[1]));
DRIVER V3 (.A(pp1_w[9]), .Z(pp1_w[12]), .number(gate_count[2]));
DRIVER V4 (.A(pp1_w[9]), .Z(pp1_w[13]), .number(gate_count[3]));
```

We will rank all students who pass ***(a)*** and have **no connection errors on number [50:0] port**. There will be a ranking according to **A*T**, where A represents the **number of transistors** and T represents the **total execution time**. Your performance score will be graded by your ranking according to 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
Incorrect number [50:0] connection	0
Correctness failed	0
Plagiarism	0

2. Report (30%)

(a) Simulation (0%)

Specify your **minimum cycle time** and **which strategy you used**. If you do not provide this information, **You will not get any score** for part 1(70%)". **This minimum cycle time would be verified by TAs.** Also, please put the screenshot of the summary provided by the testbench in the report.

(b) Circuit diagram (10%)

You are encouraged to use software to draw the 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.

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

(c) Discussion (20%)

Discuss your design.

- (3%) Introduce your design.
- (2%) How do you cut your pipelined or recursive design?
- (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_2021.pdf:** This document.
- **HW4_tutorial_2021.pdf:** Tutorial in class.
- **div.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.
- **tb.v:**

  The testbench for your design.
- **pattern/Ina.dat & Inb.dat :**

  Input patterns for the testbench. Please keep the hierarchy when simulation.
- **pattern/Golq.dat & Golr:**

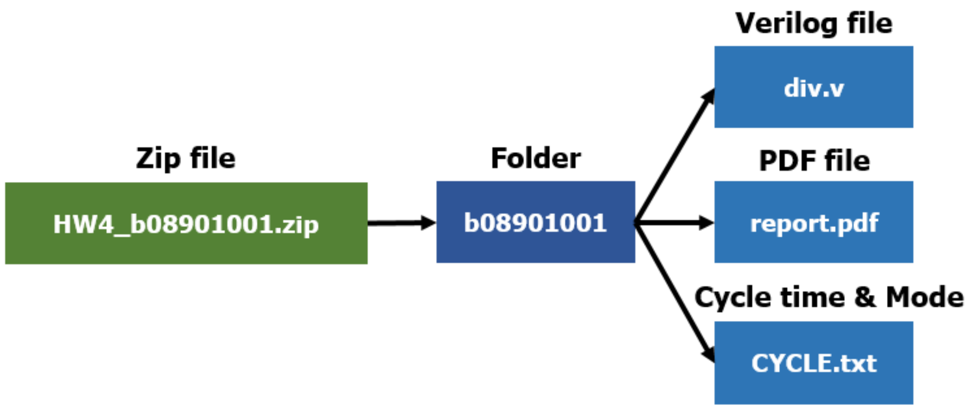
  Patterns of the correct answers for the testbench. Please keep the hierarchy when simulation.

Submission

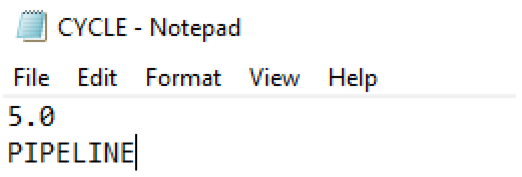
All students who do not submit files according to the rules **will get a 20% penalty.**

- You should upload a **zip file** to **CEIBA**, the file name is “HW4_Student ID”, e.g., HW4_b08901001.zip

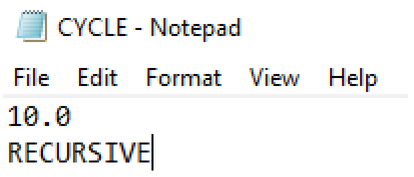
- Your file must conform to the following structure.



- Specify your cycle time and which input mode you prefer in CYCLE.txt, all letters are in uppercase, e.g.,



```
5.0
PIPELINE
```

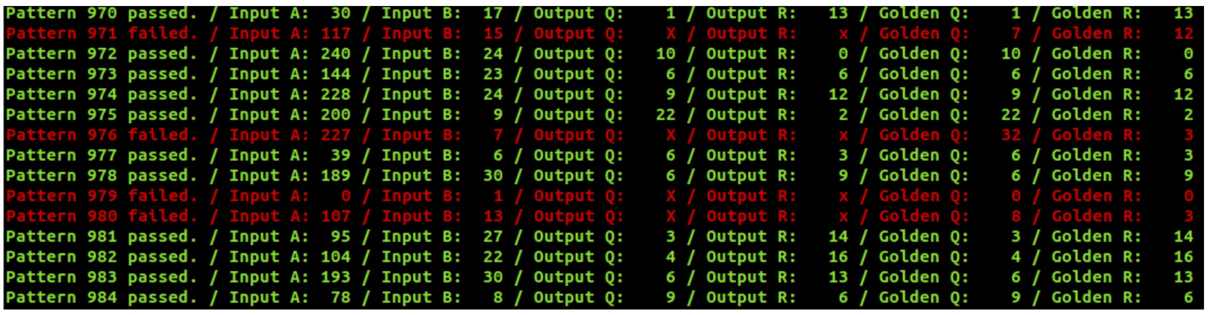


```
10.0
RECURSIVE
```

Testbench

1. Description

- The output waveform will be dumped to file "div.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 data. (**`define PATTERN   1000**)
- You can enable the debug function, which will display the data sent and received.

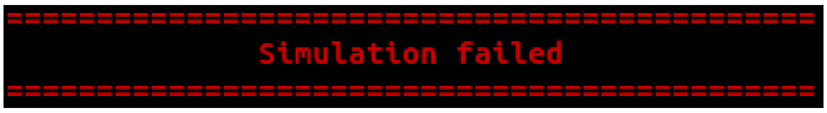


```
Pattern 970 passed. / Input A:  30 / Input B:  17 / Output Q:   1 / Output R:  13 / Golden Q:   1 / Golden R:  13
Pattern 971 failed. / Input A: 117 / Input B:  15 / Output Q:   X / Output R:   x / Golden Q:   7 / Golden R:  12
Pattern 972 passed. / Input A: 240 / Input B:  24 / Output Q:  10 / Output R:   0 / Golden Q:  10 / Golden R:   0
Pattern 973 passed. / Input A: 144 / Input B:  23 / Output Q:   6 / Output R:   6 / Golden Q:   6 / Golden R:   6
Pattern 974 passed. / Input A: 228 / Input B:  24 / Output Q:   9 / Output R:  12 / Golden Q:   9 / Golden R:  12
Pattern 975 passed. / Input A: 200 / Input B:   9 / Output Q:  22 / Output R:   2 / Golden Q:  22 / Golden R:   2
Pattern 976 failed. / Input A: 227 / Input B:   7 / Output Q:   X / Output R:   x / Golden Q:  32 / Golden R:   3
Pattern 977 passed. / Input A:  39 / Input B:   6 / Output Q:   6 / Output R:   3 / Golden Q:   6 / Golden R:   3
Pattern 978 passed. / Input A: 189 / Input B:  30 / Output Q:   6 / Output R:   9 / Golden Q:   6 / Golden R:   9
Pattern 979 failed. / Input A:   0 / Input B:   1 / Output Q:   X / Output R:   x / Golden Q:   0 / Golden R:   0
Pattern 980 failed. / Input A: 107 / Input B:  13 / Output Q:   X / Output R:   x / Golden Q:   8 / Golden R:   3
Pattern 981 passed. / Input A:  95 / Input B:  27 / Output Q:   3 / Output R:  14 / Golden Q:   3 / Golden R:  14
Pattern 982 passed. / Input A: 104 / Input B:  22 / Output Q:   4 / Output R:  16 / Golden Q:   4 / Golden R:  16
Pattern 983 passed. / Input A: 193 / Input B:  30 / Output Q:   6 / Output R:  13 / Golden Q:   6 / Golden R:  13
Pattern 984 passed. / Input A:  78 / Input B:   8 / Output Q:   9 / Output R:   6 / Golden Q:   9 / Golden R:   6
```

- If you pass the simulation, you should see:

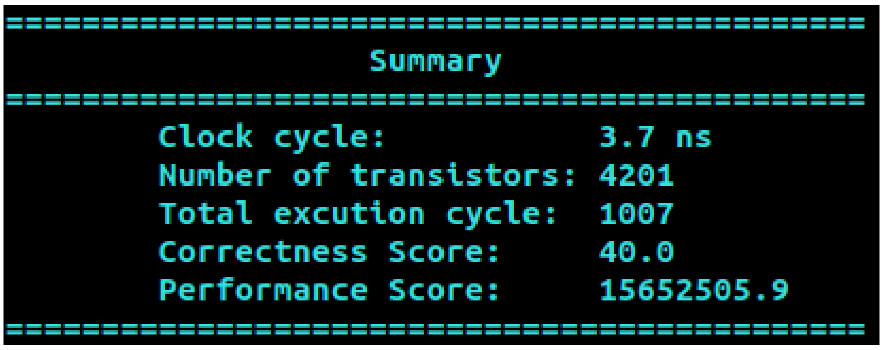
```
============================================
              Simulation passed
============================================
```

Otherwise, you would see:



```
==================================================
                Simulation failed
==================================================
```

You would also see the summary:



```
==================================================
                     Summary
==================================================
        Clock cycle:              3.7 ns
        Number of transistors:  4201
        Total excution cycle:   1007
        Correctness Score:      40.0
        Performance Score:      15652505.9
==================================================
```

Note that the performance score is the A*T value that would be used in ranking.

2. Simulation command

(a) Recursive

> ncverilog tb.v div.v lib.v +access+r

> ncverilog tb.v div.v lib.v +access+r +define+DEBUG

(b) Pipeline

> ncverilog tb.v div.v lib.v +access+r +define+PIPELINE

> ncverilog tb.v div.v lib.v +access+r +define+DEBUG+PIPELINE

TA email: r09943022@ntu.edu.tw, EE2-329

HW4 Office hours: (Mon) 13:00~16:00 @EE2-329

(Wed) 13:00~16:00 @EE2-329

If you are not convinient during office hours, you can email TA to make an alternative appointment.