NYCU-EE IC LAB – spring 2023

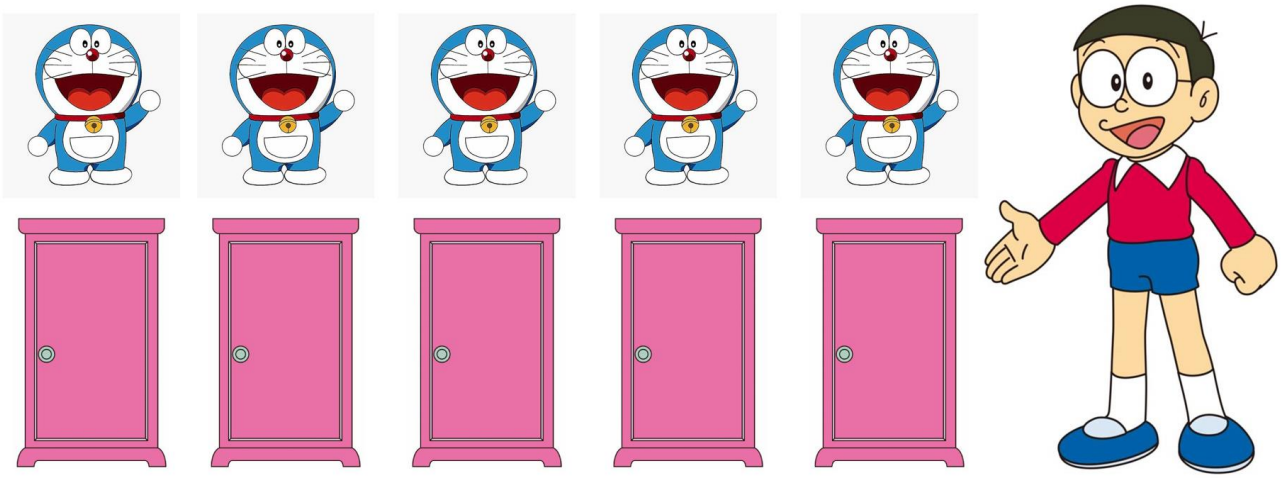
Lab07 Exercise

Design: Clock Domain Crossing (CDC)

Data Preparation

1. Extract the test data from TA's directory:
   % tar xvf ~iclabta01/Lab07.tar
2. The extracted LAB directory contains:
   a. PRACTICE/
   b. EXERCISE/

Design Description



Nobi is a Doraemon tester who is responsible for testing the Doraemon sent from the factory. There are a total of five Anywhere Doors in the factory, and as soon as Nobi chooses a Doraemon behind one of the doors, the factory will produce another one and fill it behind that door.

In this lab, you will need to choose the Doraemon that best matches (with the preference score) Nobi's preferences for this test from the five doors, and output the corresponding door number and Doraemon ID.

You will receive information which includes:

1. **doraemon_id[4:0]**, ranged from 0-31.
2. Doraemon info. : **size[7:0]**, **iq_score[7:0]**, **eq_score[7:0]**, ranged from 50-200.
3. Nobi's preference weight:
   size_weight[2:0], **iq_weight[2:0]**, **eq_weight[2:0]**, ranged from 0-7.

Preference score is defined as "**size*size_weight+ iq_score*iq_weight+ eq_score*eq_weight**".

As soon as a Doraemon is selected for testing, a new set of preference weights (**size_weight, iq_weight, eq_weight**) will be given.

In the case of equal scores, Nobi will choose the Doraemon with a smaller door_number.

You should output **out[7:0]** defined as 8bit**{door_number, doraemon_id}**.

Due to the difference in production and testing time, the input and output will be in different clock domains.

TA will provide an Asynchronous FIFO memory (8bits, 16words). You should learn how to use this FIFO to communicate between different clock domain signals. If you make a good design, you can complete all functions without any stall.

Examples

- **Test 1**

  First five doremon_id : [19 , 0 , 28, 15, 3]
  size, iq_score , eq_score of doraemon at door0: [115, 141, 185]
  size, iq_score , eq_score of doraemon at door1: [169, 113, 63]
  size, iq_score , eq_score of doraemon at door2: [90, 78, 145]
  size, iq_score , eq_score of doraemon at door3: [113, 147, 189]
  size, iq_score , eq_score of doraemon at door4: [196, 113, 53]
  size_weight, iq_weight, eq_weight of test1: [3, 6, 4]
  preference score of doraemon at door0 : 115*3+141*6+185*4 = 1931
  preference score of doraemon at door1 : 169*3+113*6+62*4 = 1437
  preference score of doraemon at door2 : 1318
  preference score of doraemon at door3 : 1977
  preference score of doraemon at door4 : 1478
  After comparison, it was found that Doraemon at Door 3 has the highest score, and is selected in this test.
  Thus , **out** ={door_number, doraemon_id} = {3'd3, 15} = 8'd111.

- **Test 2**

  The selected door will receive a new Doraemon.
  New input:
  doreamon_id = 2 at door3.
  size, iq_score , eq_score = [123, 141, 183]
  size_weight, iq_weight, eq_weight of test2: [1, 7, 4]

Therefore, now doremon_id at first five: [19, 0, 28, 2, 3]
Preference score = [1842, 1212, 1216, 1842, 1199]
Doraemon at Door 0 and Door 3 has the same highest score, we choose smaller door 0 in this test.
Thus , **out** ={door_number, doraemon_id} = {3'd0, 19} = 8'd19.
…

Inputs and Outputs

• Inputs

I/O	Signal name	Bit Width	Description
Input	**clk1**	1	Clock
Input	**clk2**	1	Clock
Input	**rst_n**	1	Asynchronous active low reset
Input	**in_valid**	1	High when input signals are valid and the **ready** signal is high. Be synchronized in **clk1** domain.
Input	**doraemon_id**	5	**doraemon_id** is valid when **in_valid** is high. Be synchronized in **clk1** domain.
Input	**size**	8	**size** is valid when **in_valid** is high. Be synchronized in **clk1** domain.
Input	**iq_score**	8	**iq_score** is valid when **in_valid** is high. Be synchronized in **clk1** domain.
Input	**eq_score**	8	**eq_score** is valid when **in_valid** is high. Be synchronized in **clk1** domain.
Input	**size_weight**	3	**size_weight** is valid when **in_valid** is high, except for the first four **in_valid** high cycles. Be synchronized in **clk1** domain.
Input	**iq_weight**	3	**iq_weight** is valid when **in_valid** is high, except for the first four **in_valid** high cycles. Be synchronized in **clk1** domain.
Input	**eq_weight**	3	**eq_weight** is valid when **in_valid** is high, except for the first four **in_valid** high cycles. Be synchronized in **clk1** domain.

• Outputs

I/O	Signal name	Bit Width	Description
output	**ready**	1	Should be set to low after reset**.** Should be set to high when you are ready to get the input signals. Be synchronized in **clk1** domain.
output	**out_valid**	1	Should be set to low after reset**.** Should be set to high when your **out** is ready. Be synchronized in **clk2** domain.
output	**out**	8	Should be set to low after reset**.** Output the doraemon's { door, doraemon_id} who has the best preference score (largest size_weight*size + iq_weight*iq_score + eq_weight*eq_score) for that test. Be synchronized in **clk2** domain.

Specifications

1. Top module name: CDC (design file name: CDC.v)
2. Input pins: **clk1**, **clk2**, **rst_n, in_valid, doraemon_id[4:0], size[7:0], iq_score[7:0], eq_score[7:0] , size_weight[2:0], iq_weight[2:0], eq_weight[2:0]** .
   Output pins: **ready**, **out_valid, out[7:0]**.
3. All operations are **unsigned**.
4. Use **asynchronous** reset active low architecture.
5. All your output registers should be set to zero after reset.
6. All your output registers should be set to zero after the last output is complete.
7. The **clk1** and **clk2** signal will be released after 1 **clk1** cycle after the **rst_n** pull high.
8. You can select the clk1, clk2 period by yourself. TA will check the functionality from fast to slow and slow to fast. (case1: clk1=15.3 ns, clk2=18.5 ns ; case2: clk1=18.5ns, clk2=15.3 ns) And calculate the performance by clk1, clk2 you select.
9. The maximum value of clk1 and clk2 that you have selected is 20ns.
10. TA's PATTERN will totally give 6000 inputs. And the total latency, which is the time from first in_valid to last out_valid, can not over 100000 clk2 cycles.
11. You should use the AFIFO module to communicate with different clock domain signals TA will check this instance. But the number of AFIFO module is not limited.
12. After synthesis, check the "CDC.area" and "CDC.timing" in the folder "Report". The area report is valid only when the slack in the end of "CDC.timing" is **non-negative** and the result should be **MET**.
13. Regardless of whether signal **ready** is high or not, the first four inputs will be given after 3-5 clk1 cycles after the **rst_n** pull high.
    Starting from the 5th input, the input will come only when the **ready** is high. But it may not come as soon as **ready** rises.(The delay from the rise of **ready** to the rise of **invalid** is at most 150 clk1 cycles per times.)
14. The synthesis result **cannot** contain any **LATCH.**
15. The synthesis result **cannot** contain any **error.**
16. **out_valid** is set to high only when **out** is valid.
17. The output signal **out** should be correct only when **out_valid** is high.
18. The output signal **out** should be reset when **out_valid** is low.
19. The reset signal (rst_n) would be given only once at the beginning of the simulation. All output signals must be **reset to low** after the reset signal is asserted.
20. Input data are synchronous to clk1, output data are synchronous to clk2.
21. Your design **can't use DesignWare IP** in this lab.
22. First 4 inputs can't output **out**.
23. The **out_valid** can overlap with **in_valid**.
24. **Changing top module is prohibited.**
25. After DC synthesis, you can check CDC.area and CDC.timing. The area report is valid when the

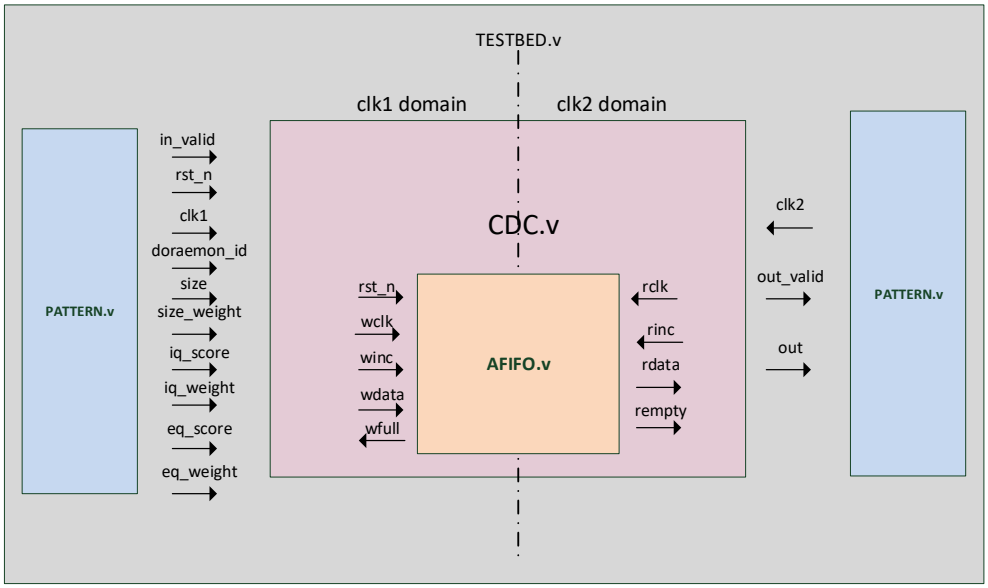
slack at the end of the timing report should be **non-negative (MET)**.

26. After PT synthesis, the timing should be **non-negative (MET)** (in pt.log).
27. The DC synthesis result **cannot** include any **latches** or **errors** (in syn.log).
28. The input delay is set to **0.5*(clock1 period)**.
29. The output delay is set to **0.5*(clock2 period)**, and the **output loading is set to 20**.
30. **You need to write your own .sdc file. (setting the false path).**
31. **You need to modify the pt.tcl file. (setting set_annotated_check). You can find the first FF name of the synchronizer in the CDC_SYN.sdf file.**
32. The gate level simulation cannot include any timing violations without the *notimingcheck* command.
33. Don't use any wire/reg/submodule/parameter name called *error*, *congratulation*, *latch* or *fail* otherwise you will fail the lab. Note: * means any char in front of or behind the word. e.g: error_note is forbidden.
34. Verilog commands //synopsys dc_script_begin, //synopsys dc_script_end //synopsys translate_off, //synopsys translate_on are only allowed during the usage of including and setting designware IPs; other design compiler optimizations are forbidden.
35. Using the above commands are allowed, however any error messages during synthesis and simulation, regardless of the result will lead to failure in this lab.
36. Synthesis time should not exceed 2 hours.

Grading Policy

- **Functionality (70%)**
- **Performance (30%)：clk1 * Total Latency * Area^2**
- Total Latency means the time from first in_valid to last out_valid.

Block diagram



Note

1. Submit your design (CDC.v) in Lab07/EXERCISE/09_SUBMIT
   a. 1st_demo deadline: **2023/04/17(Mon.) 12:00:00**
   b. 2nd_demo deadline: **2023/14/19(Wed.) 12:00:00**
2. **Please upload the following files under 09_ SUBMIT:**
   - CDC.v, CDC.sdc and pt.tcl
   - In this lab, you can adjust your clock cycle time. Consequently, make sure to key in your clock cycle time after the command like the figure below.

     Ex:

     ```
     /09_SUBMIT]$ ./01_submit 15.5 18.3
     ```

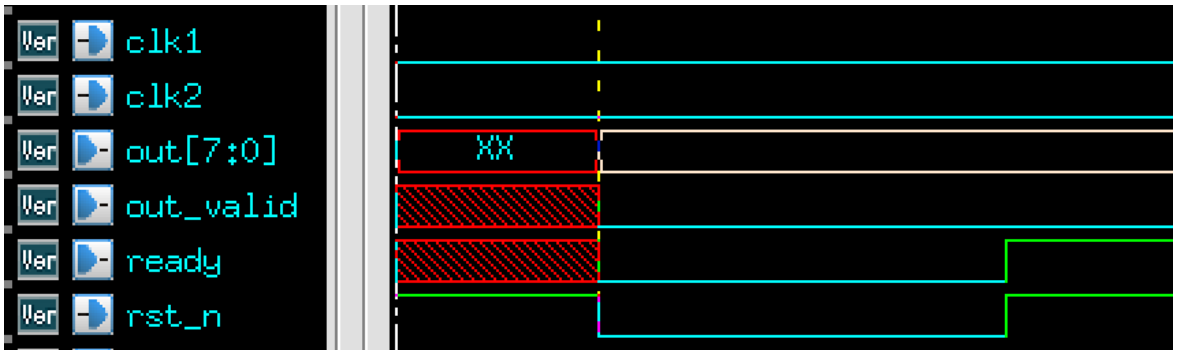
     pt_idabta10.tcl
     15.5_18.3_idabta10.txt
     CDC_idabta10.v
     CDC_idabta10.sdc

   - If your file **violates the naming rule**, you will **lose 5 points**.
3. **Template folders and reference commands:**

   01_RTL/ (RTL simulation) **./01_run**
   02_SYN/ (Synthesis) **./01_run_dc**
   (Check if there is any **latch** in your design in **syn.log)**
   (Check the timing of the design in /Report/ CDC**.timing**)
   ./02_run_pt
   (set_annotated_check for the first FF of synchronizer)
   03_GATE / (Gate-level simulation) **./01_run**
   09_SUBMIT/ (submit files) **./01_submit**
   09_SUBMIT / (check files) **./02_check**
   You can key in ./09_clean_up to clear all log files and dump files in each folder.

Sample Waveform

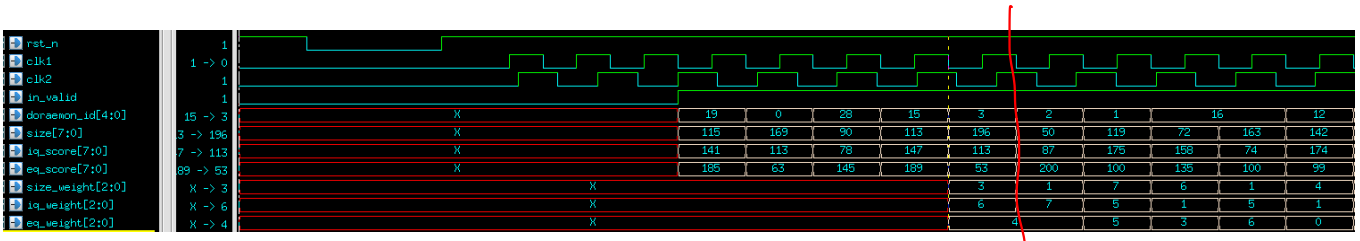
All output signals must be **reset to low** after the reset signal is asserted.



The examples of the first 10 inputs.

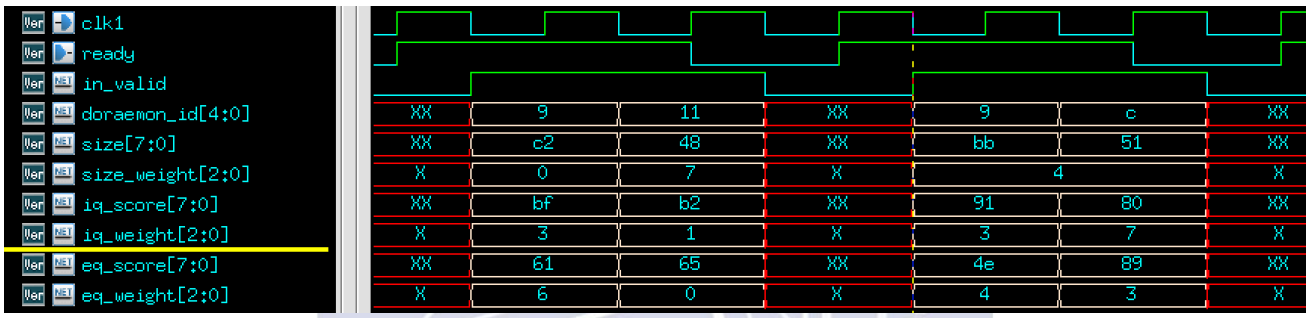
Regardless of whether signal **ready** is high or not, the first four inputs will be given after 3-5 cycles after the **rst_n** pull high.

Starting from the 5th input, **size**, **iq_score**, **eq_score** will be given.



Starting from the 5th input, the input will be based on the output signal **ready**.

When **ready** is low, there will be no input.



Output singal