NCTU-EE IC LAB – Spring2022

Lab07 Exercise

Design: Clock Domain Crossing (CDC)

◆ Data Preparation

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

◆ Design Description

In this lab, you are asked to find out an account who has the best performance in every 5 accounts. The input data will be given continuously when you are ready. You will get the information including account number (in_account[7:0]), area (in_A[7:0]), latency (in_T[7:0]). Your task is to calculate and output the accounts who get best performance (smallest in_A*in_T) in every 5 accounts.

For example:

in_account	0	4	3	7	11	25	9	10
in_A	20	9	11	35	71	3	22	53
in_T	1	7	4	20	3	8	3	4
Performance	20	63	44	700	213	24	66	212

Output : 0 Output : 25 Output : 25

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.

◆ 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	**in_account**	8	**in_account** is valid when **in_valid** is high. Be synchronized in **clk1** domain.
Input	**in_A**	8	**in_A** is valid when **in_valid** is high. Be synchronized in **clk1** domain.
Input	**in_T**	8	**in_T** is valid when **in_valid** is high. 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.
output	**out_valid**	1	Should be set to low after reset. Should be set to high when your **out_account** is ready. Be synchronized in **clk2** domain.
output	**out_account**	8	Should be set to low after reset. Output the account who has the best performance (smallest in_A*in_T). Be synchronized in **clk2** domain.

◆ Specifications

1. Top module name：CDC (File name: CDC.v)
2. Input pins ：**clk1**, **clk2**, **rst_n**, **in_valid**, **in_account[8:0]**, **in_A[8:0]**, **in_T[8:0]**
   Output pins：**ready**, **out_valid, out_account [8:0]**
3. Use **asynchronous** reset active low architecture.
4. All your output registers should be set zero after reset.
5. You can select the clk1, clk2 period by yourself. TA will check the functionality from fast to slow and slow to fast. (case1: clk1=10 ns, clk2=15 ns ; case2: clk1=16 ns, clk2=15 ns) And calculate the performance by fast to slow.
6. TA's PATTERN will totally give 4000 inputs. And the total latency, which is the time from first in_valid to last out_valid, can not over 100000 clk2 cycles.
7. 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.
8. 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**.
9. The input will come only when the **ready** is high. But it may come at any time.(at most wait 150 clk1 cycles per times, totally you have to reserve 500 clk1 cycles for TA's PATTERN)
10. The synthesis result **cannot** contain any **LATCH except for clocking gating latch.**
11. The synthesis result **cannot** contain any error**.**
12. The output loading is set to 0.05.
13. Input delay and output delay are 0.5*Clock Period.
14. You can't have timing violation in gate-level simulation
15. Your design **can't use DesignWare IP** in this lab.
16. First 4 inputs don't have to output the best account.
17. The **out_valid** can overlap with **in_valid**.
18. You need to write your own .sdc file. (setting the false path). ↗ find first FF
19. You need to modify the pt.tcl file. (setting set_annotated_check). You can find the first FF name of synchronizer in CDC_SYN.sdf file.
20. The gate level simulation cannot include any timing violations without the *notimingcheck* command

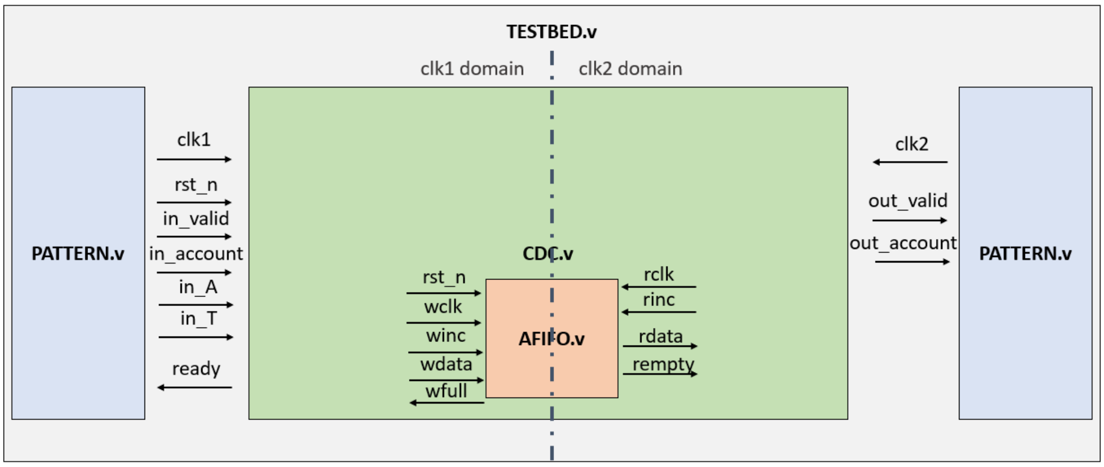
21. Don't use any wire/reg/submodule/parameter name called *error*, *congratulation*, *latch* or *fail* otherwise you will fail the lab. Note: * means any char in front of or behind the word. e.g: error_note is forbidden.

22. Don' t write Chinese comments or other language comments in the file you turned in.

23. 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.

24. 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.

◆ Block diagram



◆ Grading Policy

- **Functionality correctness (70%)：**
- **Performance (30%)：**
  - clk1 * Total Latency$^{1.3}$ * Area：**30%**

    Total Latency means the time from first in_valid to last out_valid.

◆ Note

1. **Please upload the following files on e3 platform before 23:59 p.m. on Apr. 24:**
   - CDC_iclab???.v ,clk1_clk2_iclab???.txt , CDC_iclab???.sdc and pt_iclab???.tcl (ex. CDC_iclab099.v & 10.0_15.0_iclab099.txt & CDC_iclab099.sdc & pt_iclab099.tcl).
   - The .txt file contents can be empty, you only need to specify the clock cycle in the file name.
   - You should set_false_path in sdc file and set_annotated_check in pt.tcl file.
   - The 2nd demo deadline is **23:59 p.m. on Apr. 29**.
   - Check whether there is any wire / reg / submodule name called "error", "fail", "pass", "congratulation", if you used, you will fail the lab.
   - If your file violates the naming rule, your will lose 5 point.
2. **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 design in /Report/CDC.**timing**)

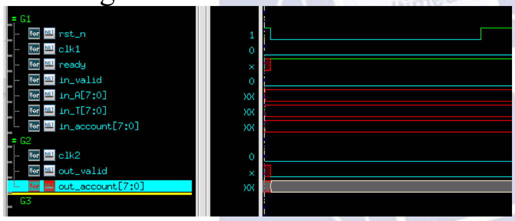
./02_run_pt

(set_annotated_check for the first FF of synchronizer)

03_GATE / (Gate-level simulation) **./01_run**

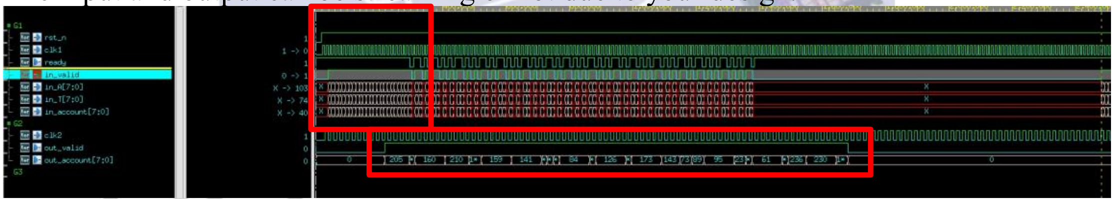
◆ Waveform Example

Reset signals

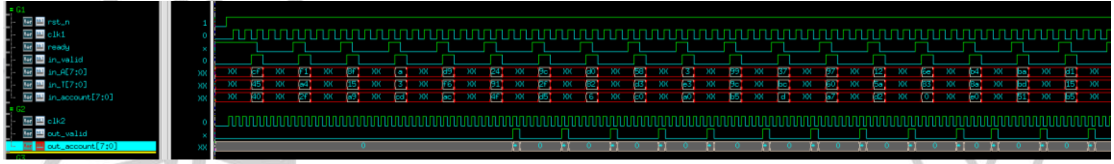


The input and output can be streaming or not due to your design.

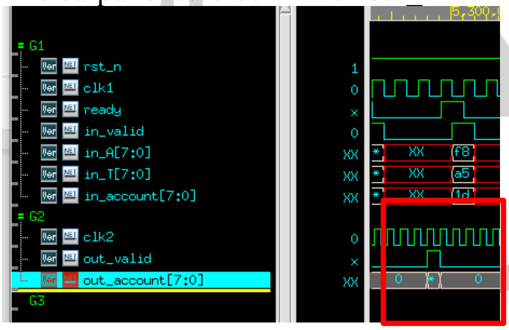
✓



x



The output should be 0 when out_valid = 0.



$\times 10^{10}$

2.2964

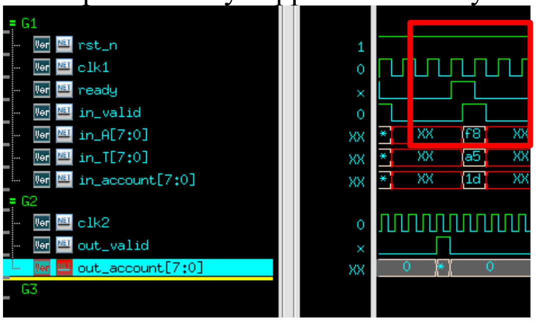
985768 10743

75841 8379 2.5

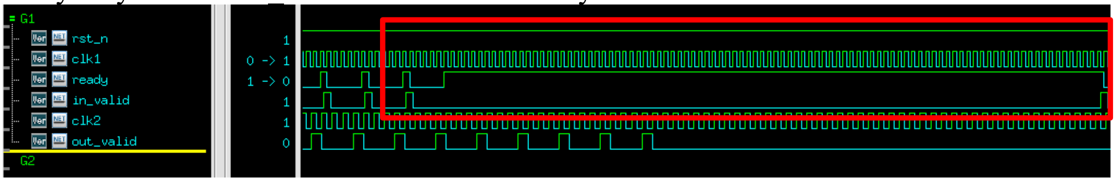
80276 7598 2.5

2.22×10^{10}

The input will only happen when ready is high.



ready may wait the in_vaild at most 150 clk1 cycles.



2.5 5 $95841 \times 8379 \times 2.5^2 = 5.969$

2.5 4.5 $80276 \times 9598 \times 2.5^2 = 5.56$

3 5 $95748 \times 9074 \times 3^2 = 6.889$

3 4.5 $96201 \times 6419 \times 3^2 = 6.108$

2.1 4.5 $80492 \times 8970 \times 2.1^2 = 4.885$

2.1 4.3 $85408 \times 8582 \times 2.1^2 = 4.893$

2.1 4.1 $90594 \times 8194 \times 2.1^2 = 4.887$

$87643 \times 8379 \times 2.1^2 = 4.86$

$69643 \times 4189^{1.3} \times 2.1 = 9.4 \times 10^9$

$45000 \times 4007^{1.3} \times 5.2 = 1.13 \times 10^{10}$

$76334 \times 4083^{1.3} \times 2.1$

$40000 \times 4500^{1.3} \times 4$

8.98×10^{9}

$71000 \times 4100 \times 2$

7.414×10^{9}