NCTU-EE IC LAB - SPRING2020

Lab03 Practice

Design: Vector Reverse Ordering

Data Preparation

1. Extract test data from TA's directory:

   % tar xvf ~iclabta01/Lab03.tar

2. The extracted LAB directory contains:

   a. **00_TESTBED**

   b. **01_RTL**

   c. **02_SYN**

   d. **03_GATE**

Design Description

Six numbers as a vector are given from the input signal **in** for continuous **6 cycles**, and you job is to reverse the order of the vector you get. When you finish the calculation, you should pull up the output signal **out_valid** and output your result to signal **out** for exactly **6 cycles**.

In this practice, we will provide the design for you, however, the design contains some bugs. You need to write a pattern to test the design, find out the bugs and correct them.

Inputs and Outputs

The following are the definitions of input signals

Input Signals	Bit Width	Definition
clk	1	Clock.
rst_n	1	**Asynchronous active-low reset.**
in_valid	1	High when in is valid.
in	3	6 numbers given in 6 successive cycles as a vector.

The following are the definitions of output signals

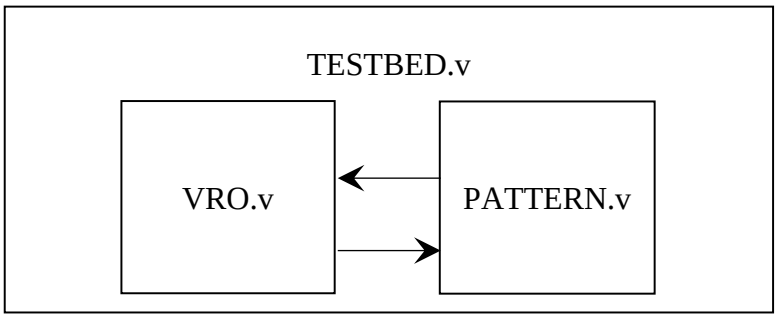
Output Signals	Bit Width	Definition
out_valid	1	High when out is valid.
out	3	The reverse order of input vector.

1. The input signal **in** is delivered for **6 cycles** when signal **in_valid** is high. When signal **in_valid** is low, input signal **in** is tied to unknown state.
2. All input signals are synchronized at negative edge of the clock.
3. The output signal **out** must be delivered for **6 cycles**, and **out_valid** should be high simultaneously.
4. The next round of the inputs will come in **4 negative edge of clock** after your **out_valid** is pulled down.

Specifications

1. Top module name: VRO (design file name: VRO.v)
2. **It is asynchronous reset and active-low architecture. If you use synchronous reset (considering reset after clock starting) in your design, you may fail to reset signals.**
3. **The reset signal (rst_n) would be given only once at the beginning of simulation. All output signals should be reset after the reset signal is asserted.**
4. The **out_valid** is limited to high for **6 cycles** when you want to output the result.
5. The execution latency is limited in **30 cycles**. The latency is the clock cycles between the falling edge of the last **in_valid** and the rising edge of the first **out_valid**.
6. The clock period is **5 ns.**
7. The input delay is set to **0.5*(clock period)**.
8. The output delay is set to **0.5*(clock period)**, and the output loading is set to **0.05**.
9. The synthesis result of data type **cannot** include any **latches**.
10. The gate level simulation cannot include any timing violations without the *notimingcheck* command.
11. After synthesis, you can check VRO.area and VRO.timing. The area report is valid when the slack in the end of timing report should be **non-negative**.

Block diagram



Note

1. **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/VRO**.timing**)

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

Sample Waveform

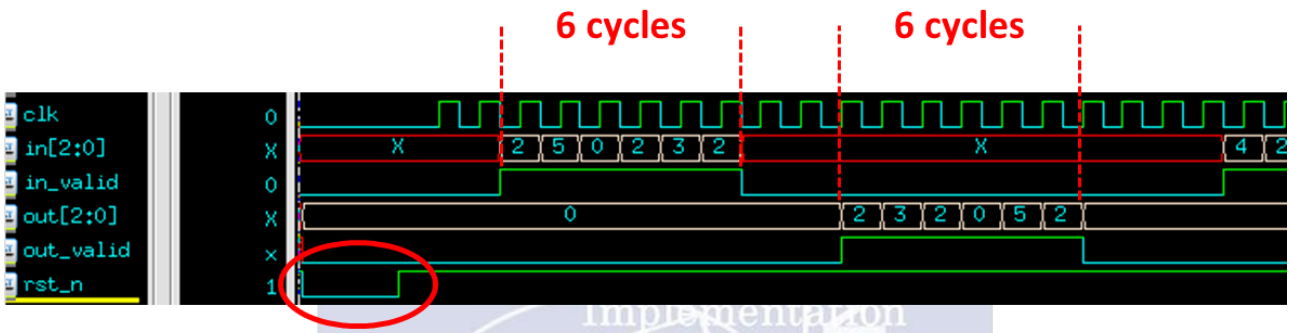


Fig 1. Input and Output waveform