NYCU-EE IC LAB - Spring 2023

Lab04 Practice

Design: Vector Inner Product

Data Preparation

1. Extract test data from TA's directory:

   % tar xvf ~iclabta01/Lab04.tar

2. The extracted LAB directory contains:

   a. **00_TESTBED**

   b. **01_RTL**

   c. **02_SYN**

   d. **03_GATE**

Design Description

In this practice, we will send **two vector** with **2 elements** to your design. You need to calculate the inner product of this two vector. The arithmetic representation of the number is IEEE-754 floating point format. Thus, you need to use **related IPs provided by Designware** to finish the task.

- IEEE 754 Floating-Point Format

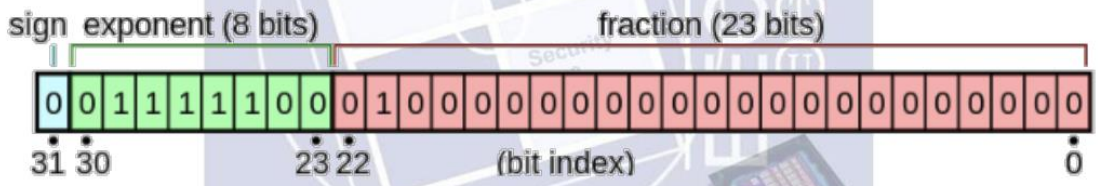


Fig. The IEEE 754 single-precision floating point format

The actual value of the number can be calculated: value = $(-1)^{\text{sign}} \times 2^{exponent-127} \times (1 + \sum_{i=1}^{23} f_{23-i} 2^{-i})$ For further information, you can refer https://en.wikipedia.org/wiki/Single-precision_floating-point_format

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 input is valid.
vector_1	32	The elements of first vector. The arithmetic representation follows the IEEE-754 floating number format.
vector_2	32	The elements of second vector. The arithmetic representation follows the IEEE-754 floating number format.

The following are the definitions of output signals

Output Signals	Bit Width	Definition
out_valid	1	High when out is valid.
out	32	The inner product result. The arithmetic representation follows the IEEE-754 floating number format.

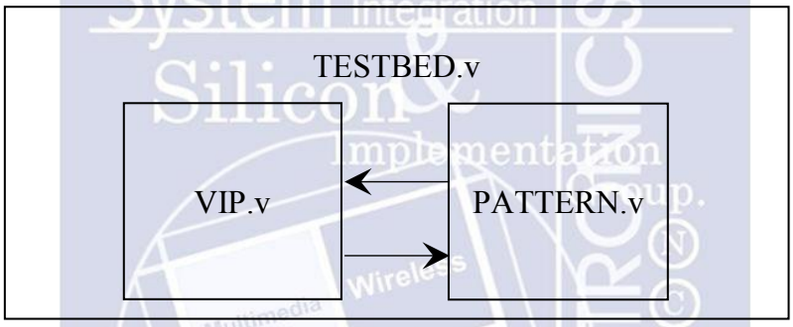
1. The input signal **vector_1** is delivered in **raster scan order** for **2 cycles** continuously. When **in_valid** is low, input is tied to unknown state.
2. The input signal **vector_2** is delivered in **raster scan order** for **2 cycles** continuously. When **in_valid** is low, input is tied to unknown state.
3. All input signals are synchronized at negative edge of the clock.
4. The output signal **out** must be delivered for **1 cycle**, and **out_valid** should be high simultaneously.
5. The next round of the data will come in **2 negative edge of clock** after your **out_valid** is pulled down.

Specifications

1. Top module name: VIP (design file name: VIP.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** should be reset after your **out_valid** is pulled down.

5. The **out_valid** is limited to high only **one** cycle when you want to output the result.
6. 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**.
7. The clock period is **10 ns.**
8. The input delay is set to **0.5*(clock period)**.
9. The output delay is set to **0.5*(clock period)**, and the output loading is set to **0.05**.
10. The synthesis result of data type **cannot** include any **latches**.
11. The gate level simulation cannot include any timing violations without the *notimingcheck* command.
12. After synthesis, you can check VIP.area and VIP.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/VIP**.timing**)

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

Sample Waveform

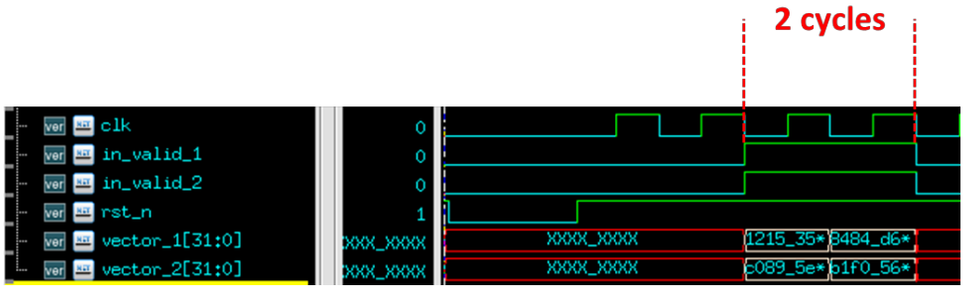


Fig 1. Input waveform

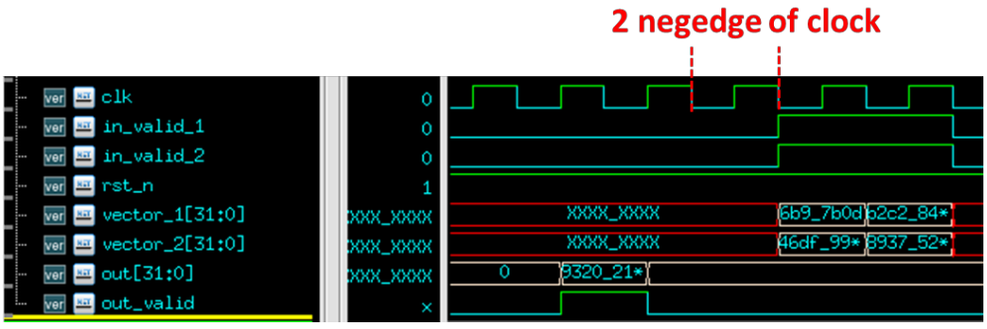
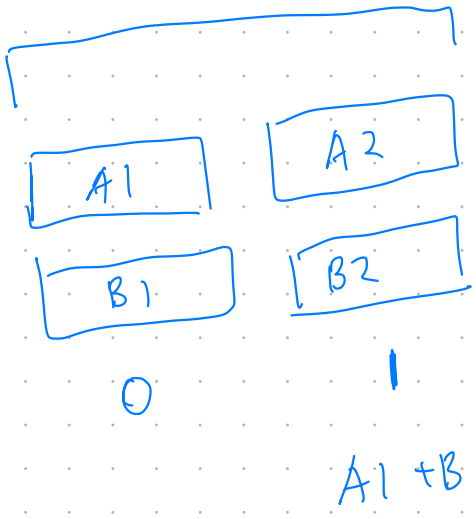


Fig 2. Output waveform



2

buf

3