AI on Chip 2024

LAB Ⅳ Processing Elements

REPORT

Student name: \_\_胡家豪\_\_\_\_\_\_\_\_\_

Student ID: \_\_N26122246\_\_\_\_\_\_\_\_

目錄

[一. Implementation Result 2](#_Toc166444781)

[1. TB.0 2](#_Toc166444782)

[2. TB.1 2](#_Toc166444783)

[3. TB.2 2](#_Toc166444784)

[二. Implementation Cycle Result 2](#_Toc166444785)

[三. PE architecture and Design ideas 2](#_Toc166444786)

[1. PE architecture 2](#_Toc166444787)

[2. Spad storage space and functionality 2](#_Toc166444788)

[3. FSM state and functionality 2](#_Toc166444789)

[4. Waveform Result 2](#_Toc166444790)

[四. Demonstrate Scenario B dataflow 3](#_Toc166444791)

[五. Compare scenario A,B,C from different aspects 3](#_Toc166444792)

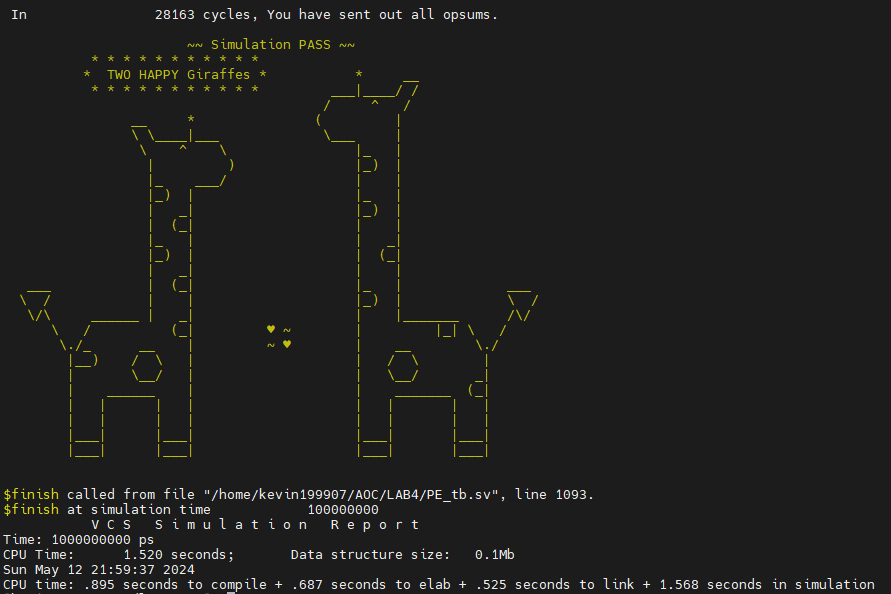
[六. Share your thoughts 3](#_Toc166444793)

1. Implementation Result
2. TB.0

一張含有 文字, 螢幕擷取畫面, 圖表, 字型 的圖片

自動產生的描述

1. TB.1



1. TB.2

一張含有 文字, 螢幕擷取畫面, 字型, 圖表 的圖片

自動產生的描述

1. Implementation Cycle Result

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Testbench | Tb0 | Tb1 | Tb2 | Tb3 |
| Clock Cycle | 922 | 28163 | 8109 | - |

1. PE architecture and Design ideas
2. PE architecture一張含有 文字, 圖表, 螢幕擷取畫面, Rectangle 的圖片

   自動產生的描述
3. Spad storage space and functionality

一張含有 文字, 螢幕擷取畫面, 圖表, Rectangle 的圖片

自動產生的描述

1. FSM state and functionality

一張含有 文字, 圖表, 字型, 線條藝術 的圖片

自動產生的描述

IDLE：閒置狀態

SET\_INFO：吃到「set\_info」後進入，此stage在儲存Layer的狀態。

STORE\_INITIAL：存完layer後直接進入，在此狀態會將要儲存的ifmap以及filter讀入，之後每次要拿新的ifmap就會進入這個狀態重新拿取ifmap與filter，並且會有counter計算要讀取幾個ifmap與filter，都讀取後將STORE\_INITIAL\_done拉起

CACULATE：在這個stage會將ifmap與filter進行相乘相加，利用counter計算要相乘相加多少筆資料，完成後將CACULATE\_filter\_done拉起。

WAIT\_ACCUMULATE：用於緩衝

ACCUMULATE：這邊將讀入的ipsum與算出來的結果相加乘opsum，並把結果輸出。並且分為兩個情況。

情況1：該次計算還沒有算完這個ifmap，那就等待ipsum Handshake後將ifmap右移一格。

情況2：該次計算已經算完這個ifmap，那就等待ipsum Handshake後重新讀入filter與ifmap。

SHIFT\_IFMAP：這個狀態是一個ifmap還沒算完，所以進行右移，移完後就直接回到CACULATE進行下一次的運算。

1. Waveform Result
2. Demonstrate Scenario B dataflow

Demonstrate Scenario B dataflow by completing 2 **3row x 3col x 2ch ofmaps** by accumulating **2-channel of 2 3x3 ilters(kernel)**

**and 2 5x5 ifmaps** under **parameters n=1, p=2(kernel), q=1(channel).**

N：一個PE一次執行幾個ifmap

p：PE執行一次有多少kernal存在PE

q：一次算多少個channel

Scenario B： ifmap reuse

根據上面的要求，我們可以將要進行的運算畫成如下所示，下面藍色為ifmap、綠色為kernal、黃色為opsum前三次運算會得到的部分。

一張含有 Rectangle, 正方形, 行 的圖片

自動產生的描述

假設我們用PE1,1當作示例，ifmap會直接存入spad之後在完成這個row之前都不再另外讀取其他的data，而filter的kernal channel 1 與 channel2的一個元素則會由左至右不斷讀取進PE內進行運算，一次讀取兩個，在重複使用ifmap的情況下直接將兩個kernal完成運算，並且將輸出與其他的PE結果相加，完成一個opsum，如下所示：

一張含有 Rectangle, 正方形, 圖表, 螢幕擷取畫面 的圖片

自動產生的描述 一張含有 Rectangle, 正方形, 圖表, 螢幕擷取畫面 的圖片

自動產生的描述

一張含有 Rectangle, 正方形, 圖表, 螢幕擷取畫面 的圖片

自動產生的描述

如此往復就可以算完一個output feature map。

之後進行第二個ifmap與第二組kernal，完成第二個ofmap

1. Compare scenario A,B,C from different aspects
2. Share your thoughts