|  |  |
| --- | --- |
| **Lab 1** | |
| 學號: 109021115 | 姓名: 吳嘉濬 |

1. **Lab Implementation**

以下是module lab1\_2的block diagram：

一張含有 文字, 圖表, 字型, 行 的圖片

自動產生的描述

我在module lab1\_2底下instantiate兩個reference為lab1\_1的instance，利用lab1\_1設計的功能各自算出相對應的output d，即result\_0和result\_1，之後再利用input request來判斷最終lab1\_2要output的result是result\_0還是result\_1。

一張含有 文字, 字型, 筆跡, 螢幕擷取畫面 的圖片

自動產生的描述

以上是module lab1\_1在判斷不同op值的情況下做出相對應動作的式子。

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

以上是instantiate兩個instance並各自分別得到result\_0和result\_1之後，判斷request的值得到正確的output result。(注意當request==2’b11時，op\_0有更高的priority)

1. **Questions and Discussions**

A. In the testbench lab1\_1\_t.v, please explain why we place #DELAY between input assignment and output verification. Hint: Gate delay.

因為輸入完input之後，期間需要經過許多電路邏輯閘的運算，才能得到想要的output，在這過程當中會有相對應的gate delay，因此我們必須加上#DELAY等待一些時間，確保output的結果出來之後，再去驗證output是否正確。

B. If we want to let the 2’b00 operation of op\_0 and op\_1 have the highest priority, 2’b01 have the 2nd highest priority, and so on. When op\_0 and op\_1 has same operation, op\_0 still has higher priority. How would you modify the code?

只有當request==2’b11時，這裡的改動才會有所影響。我選擇直接把request分成4種情況：

1.當request==2’b00，result=4’b0000；

2.當request==2’b01，result=result\_0； //從op\_0做運算得到的result

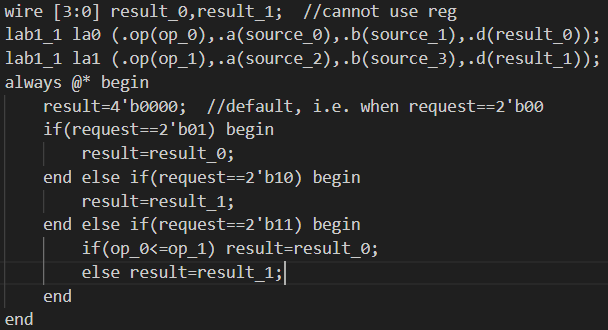
3.當request==2’b10，result=result\_1； //從op\_1做運算得到的result

4.當request==2’b11，if(op\_0<=op\_1) result=result\_0 //從op\_0做運算得到的result

else result=result\_1 //從op\_1做運算得到的result

其中要注意的地方是：當op\_0==op\_1時，op\_0有higher priority

實作的code如下：



1. **Problem Encountered**

最一開始在設計lab1\_2.v時，寫出了以下的code：

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

結果出現了以下error message：



才意識到這個寫法是有問題的，因為always block內是會被重複call的，但是在instantiate完某個instance後，該線路就固定住了，所以同一個instance是不能一直被重複instantiate的。至此，我了解到當我們要instantiate一個instance時，一定不能寫在always block內。

1. **Suggestions**

有機會的話希望老師可以在課堂上多提供畫FSM和dataflow的訣竅，總感覺這些比打Verilog還要複雜。

笑話：

一個阿姨去醫院打針。

護理師：「先深呼吸。」

阿姨：「我是小姐。」

廢到笑…😊