|  |  |
| --- | --- |
| **Lab 2** | |
| 學號: 109021115 | 姓名: 吳嘉濬 |

1. **Lab Implementation**

以下是實作Lab2\_1的kernel 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

利用sequential circuit去儲存或更新out和direction。

一張含有 文字, 字型, 螢幕擷取畫面, 數字 的圖片

自動產生的描述一張含有 文字, 字型, 螢幕擷取畫面, 數字 的圖片

自動產生的描述

考慮特殊情況以及boundary case，並assign正確的值。

如此，Parameterized\_Ping\_Pong\_Counter可以正確地被實作出來。

以下為lab2\_2的block diagram：

一張含有 文字, 圖表, 方案, 圖解 的圖片

自動產生的描述

因為是sequential circuit，所以會有current state成為下一個cycle的input的情況(如圖右上方)。另外我們可以看到pppc的flip和enable是被input mode所決定的。而offset\_cnt代表著output\_tmp的index，offset\_cnt的遞增幫助我們連續存取output\_tmp內的data並作出相對應的操作，如get(store) data、encrypt data和output data。其中encrypt的過程會使用到pppc的counter\_out，成為加密過程的參數。

以下為lab2\_2的FSM：

一張含有 文字, 字型, 行, 圖表 的圖片

自動產生的描述

當rst\_n==1'b0時，下個cycle會到INIT state。在INIT state時，如果in\_valid==1’b1，下個cycle會進入GET\_DATA state，經過8個cycle並store完8組資料後，in\_valid一定會變回1’b0(根據spec的說明)，而此時如果mode==2’b10，則下個cycle進入ENCRYPT\_DATA state(如果mode!=2’b10則繼續等待)。經過8個cycle後，8組data會依序encrypt完成，下個cycle會進入OUTPUT\_DATA state。再經過8個cycle後，8組data會依序output完成，下個cycle回到INIT state。

1. **Questions and Discussions**

1.In this lab, our reset signal is a synchronous reset. What if it is an asynchronous reset?

And how to modify your design to implement an asynchronous reset?

Synchronous reset和asynchronous reset在waveform中可觀察到最明顯的差別在於：當synchronous reset的值變成1’b0，reset不會馬上執行/生效，直到下一個clk的posedge才會有效(effective)；相對的，當asynchronous reset的值變成1’b0，reset則會立即生效(effective)。

左圖是lab2\_2.v當中的某一段code，右圖則是改成asynchronous reset的code：

一張含有 文字, 螢幕擷取畫面, 字型, 設計 的圖片

自動產生的描述 一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

基本上就是在sensitivity list上加個negedge rst\_n就可以達到asynchronous reset的效果，因為當rst\_n被de-assert時產生falling edge，即negative edge，sensitivity list內有signal產生變化，會立刻執行該always block，不用等到下一個posedge clk，因此產生了asynchronous reset的效果。

2. Why do we need both combinational circuits and sequential circuits in our design?

What are the differences between a combinational circuit and a sequential circuit?

Please explain how each of them works in detail.

我們需要combinational circuits(1)去判斷並決定下一個state/下一個offset\_cnt；(2)去process data(接收8組data/做encryption process+使用Hamming codes去建立error correction code)；

我們也需要sequential circuits(1)去update(in needed) the state/ the offset\_cnt；(2)去儲存8組data；(3)output 最終8組 data。

Combinational circuit和sequential circuit的差異在於：combinational circuit沒有feedback path between input and output，所以output depends on current input ONLY、time independent、可以對data做運算，但不具儲存功能；sequential circuit有feedback path between input and output，所以output depends on current data and past data(或者說是current data and current state)、time dependent、具有資料儲存功能。

1. **Problem Encountered**

最一開始在設計接收八組data時，是設計當我在GET\_DATA state時，把in\_data傳入next\_output\_tmp[offset\_cnt]，如下：

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

結果在跑waveform時發現，實際接收到的in\_data不是在GET\_DATA state那八個cycle對應到的in\_data，而是在GET\_DATA state的第二個cycle才開始接收in\_data，一直到ENCRYPT\_DATA state的第一個cycle，簡單來說就是整個晚一個cycle收集in\_data。下方為舉例的圖片：

一張含有 螢幕擷取畫面, 電子產品, 電路 的圖片

自動產生的描述

正確來說應該接收2,4,6,8,10,12,14,16這八組資料，但是如果用我原本錯誤的code會接收4,6,8,10,12,14,16,16這八組資料。

之後才發現一進入GET\_DATA state的當下就要接收in\_data，所以必須要在進入GET\_DATA state的前一個cycle就要先執行next\_output\_tmp[offset\_cnt] = in\_data，才會讀到GET\_DATA state的第一個in\_data，而判斷進入GET\_DATA state的前一個cycle的方式即當in\_valid==1時，如下圖所示：

一張含有 文字, 螢幕擷取畫面, 字型, 軟體 的圖片

自動產生的描述

利用這方法，才能夠得到正確的八組資料。

1. **Suggestions**

這次lab2的難度遠遠高於lab1，是知道lab會越來越難沒錯，也知道第一次lab是在給我們試水溫，但是lab2的難度真的跳好多好多，花費的時間也變得好多好多，希望之後的lab難度不要再一次跳太多了，請用溫水煮青蛙的方式對待我們QQ。