|  |  |
| --- | --- |
| **Lab 4** | |
| 學號: 109021115 | 姓名: 吳嘉濬 |

1. **Lab Implementation**

以下是lab4\_1的FSM和block diagram：

一張含有 文字, 筆跡, 字型, 行 的圖片

自動產生的描述

一張含有 文字, 筆跡, 圖畫, 兒童藝術 的圖片

自動產生的描述

當rst==1’b1時，進入INITIAL state。按下start button時，進入PREPARE state。這裡要特別留意，因為press button時，從低電位到穩定的高電位過程之中會有一段不穩定的訊號，我們可以用debounce來把它過濾掉；另外，因為我們一般人在按一次按鈕的時長對高頻率的電路而言其實經過了很多clock cycle，所以要用one-pulse方法來讓訊號只有被trigger剛好一個clock cycle，以下為debounce和one-pulse的電路：

一張含有 筆跡, 圖表, 寫生, 字型 的圖片

自動產生的描述

Debounce過濾掉沒有連續4個cycle都接收到高電位的訊號；one-pulse讓原本的訊號和delay過一個clock cycle的訊號的相反取交集，如此，可以讓連續的高電位訊號經處理後只剩下第一個cycle的高電位訊號還在。

接著我們利用除頻器，做出頻率為100Hz的clock，實作方法如下：

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

利用counter，每一個clk的posegde會使num加一，一直從0加到499999，也就是經過500000個clk頻率的clock cycle，就讓tick從0變成1或從1變成0，之後在經過500000個clk頻率的clock cycle，再tick讓轉變一次。tick轉變兩次就是clk\_div一次的clock period，總共經過1000000個clk頻率的clock cycle，相當於clk\_div的頻率是clk的一百萬分之一，而clk的頻率是100Mhz，所以clk\_div頻率是100Hz。

如此，我們可以設計出一個每0.01秒加一的counter，進而實作出”進入PREPARE state之後，經過3秒後進入COUNTING state”以及”每經過一秒讓led燈亮暗亮暗交替”等等的功能。



關於7-seg的實作原理，我以2500Hz的頻率去持續切換每個cycle要亮的7-seg led燈，在這樣的頻率之下，人類肉眼感受不到led燈的交替閃爍感，實作的kernel code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 一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

值得注意的是，因為實際電路設計的關係，7-seg是negative enable，即bit 0才會使對應的led燈亮，不是一般直覺的bit 1。

以下是lab4\_2的FSM和block diagram：

一張含有 文字, 行, 字型, 圖表 的圖片

自動產生的描述一張含有 文字, 圖表, 筆跡, 方案 的圖片

自動產生的描述

基本上實作的原理和lab4\_1差不多，我針對lab4\_1沒有出現的內容作說明。

1. 如何實作在COUNTINTG state3秒鐘之後讓7-seg的顯示從counting的數值轉變成”-“。

一張含有 文字, 螢幕擷取畫面, 軟體, 字型 的圖片

自動產生的描述 一張含有 文字, 字型, 螢幕擷取畫面 的圖片

自動產生的描述

我的作法是額外設計一個signal並命名為show。在利用counter判定進入COUNTING state3秒鐘之後，讓show從1’b1變成1’b0，如此，配合”xx?xx:xx;“的語法可以選擇想要在7-seg顯示的內容。

1. 如何存取使用者設定的欲猜測秒數。



我的作法很簡單，就是額外設計6個4-bit大小register去儲存和更新欲猜測秒數。

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

設計可以調整百位數、十位數和個位數欲猜測秒數的機制。

一張含有 文字, 字型, 螢幕擷取畫面, 設計 的圖片

自動產生的描述

並在按下start button進入COUNTING state前，讓欲猜測秒數存進register ans。

利用lab4\_1的實作原理加上以上兩點重要設計觀念，能夠順利成功完成lab4\_2。

1. **Questions and Discussions**

A. Why do we need the debounce and one-pulse modules? What is the relation

between the FSM’s clock rate and the one-pulse module’s clock rate? Explain your

reason. And what will happen when we implement with a wrong clock rate? (Please

draw waveforms to explain)

使用者在按按鈕時，訊號從低電位到高電位的過渡過程當中，會產生許多不穩定、忽高忽低的訊號，此時如果沒有做過濾的話，除了可能會讓訊號的高電位被多次讀取之外，每一次得到的訊號也都會不一，所以debounce module幫助我們在按按鈕時能過濾掉前期不穩定的訊號。使用者在按按鈕時，對使用者而言雖然可能只是一瞬間的事情，但對高頻率的電路而言，其實已經產生超過一個clock周期的連續高電位訊號，而為了避免持續許多週期的高電位訊號被重複讀取(被判定成按許多次按鈕)，我們使用one-pulse module以確保每次使用者在按一次的按鈕時，只產生剛好一個clock cycle的高電位訊號。

FSM’s clock rate需要和one-pulse module’s clock rate一致，如此一來，按按鈕所產生的one clock cycle訊號才會被FSM讀取到且剛好作用一次。如果FSM’s clock rate大於one-pulse module’s clock rate，因為按按鈕產生一個clock cycle的訊號是depend on one-pulse module’s clock rate，所以如果FSM’s clock rate是one-pulse module’s clock rate的n倍，這樣FSM會認為他連續接收到n個clock cycle的高電位訊號，進而作用n次。waveform示意圖如下：

一張含有 筆跡, 文字, 字型, 行 的圖片

自動產生的描述

而如果FSM’s clock rate小於one-pulse module’s clock rate，因為FSM是在每個clock cycle的posedge時更新狀態，所以有可能one-pulse module產生一個clock cycle的高電位訊號時，FSM沒有經歷到它自己clock的posedge，導致按按鈕的one-pulse訊號沒有被FSM判斷到。waveform示意圖如下：

一張含有 文字, 筆跡, 字型, 行 的圖片

自動產生的描述

B. Please propose two different design methods to manipulate two or more clocks in

one module.

第一個方法，也就是我在實作這次lab時使用的方法，如下圖所示：

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述 一張含有 字型, 螢幕擷取畫面, 文字, 行 的圖片

自動產生的描述

我的做法是設計一個頻率為原本頻率除以2n的除頻器module，且讓n為parameter，方便我們在instantiate兩個不同頻率的clock時，可以利用#(xxxx)的語法去設定不同頻率的除頻器。另外，我讓n預設為500000，產生頻率為100Hz的clock(週期為0.01秒)。

第二個方法是instantiate完一個clock之後，利用always block設計一個counter，當它從0 count到m-1時，clk\_div從0變成1或從1變成0，如此一來就能設計頻率為第一個clock除以2m的除頻器。

1. **Problem Encountered**

起初在設計除頻器時，我把FSM的clock頻率設為原本未除頻過的clk，結果發現bug百出，像是按按鈕有時有反應有時沒反應，7-seg出現閃爍、每個digit亮度不一或是digit的led顯示殘影跑到隔壁digit的問題。

之後看spec時發現B.A的問題討論，於是往這個方向去思考，覺得可能就是因為FSM和digit切換的頻率出錯才導致出現這些問題。最後，我把FSM的clock頻率改成和one-pulse module的clock頻率一樣，並且調整digit切換的頻率之後順利地把問題都解決了。

以下為相對應的code segment：

一張含有 文字, 螢幕擷取畫面, 字型 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

其中clk\_div1頻率為2500Hz；clk\_div2頻率為100Hz。

1. **Suggestions**

這次的lab還是和以往一樣很time consuming，但不同的是老師這次有在課堂中講解lab，幫助我們快速了解lab要我們實作什麼，進而增加我們自己看spec時的效率，希望以後也能繼續在課堂中講解lab。