實驗五 **7-Seg LED**

1. 實驗目的

* 了解MAX7219使用原理
* 設計7-Seg LED程式

1. 實驗原理

請參考上課lab5\_note講義。

1. 實驗步驟
   1. **Lab5.1: Max7219**與**7-Seg LED**練習**—without code B decode mode**

完成以下程式碼，並利用GPIO控制Max7219並在7-Seg LED上顯的第一位依序顯示0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, b, C, d, E, F (時間間隔1秒)。

|  |
| --- |
| .syntax unified  .cpu cortex-m4  .thumb  .data  arr: .byte 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0 //TODO: put 0 to F 7-Seg LED pattern here  .text  .global main  main:  BL GPIO\_init  BL max7219\_init  loop:  BL Display0toF  B loop  GPIO\_init:  //TODO: Initialize three GPIO pins as output for max7219 DIN, CS and CLK  BX LR  Display0toF:  //TODO: Display 0 to F at first digit on 7-SEG LED. Display one per second.  BX LR  MAX7219Send:  //input parameter: r0 is ADDRESS , r1 is DATA  //TODO: Use this function to send a message to max7219  BX LR  max7219\_init:  //TODO: Initialize max7219 registers  BX LR  Delay:  //TODO: Write a delay 1sec function  BX LR |

* 1. **Lab5.2: Max7219**與**7-Seg LED**練習**—use code B decode mode**

利用GPIO控制Max7219並在7-Seg LED上顯示自己的學號，例如學號為1234567則顯示下圖：



|  |
| --- |
| .syntax unified  .cpu cortex-m4  .thumb  .data  student\_id: .byte 1, 2, 3,4, 5, 6, 7 //TODO: put your student id here  .text  .global main  main:  BL GPIO\_init  BL max7219\_init  //TODO: display your student id on 7-Seg LED  Program\_end:  B Program\_end    GPIO\_init:  //TODO: Initialize three GPIO pins as output for max7219 DIN, CS and CLK  BX LR  MAX7219Send:  //input parameter: r0 is ADDRESS , r1 is DATA  //TODO: Use this function to send a message to max7219  BX LR  max7219\_init:  //TODO: Initial max7219 registers.  BX LR |

* 1. **Lab5.3 Max7219**與**7-SEG LED**練習**—**顯示**Fibonacci**數

請設計一組語程式偵測實驗板上的User button，當User button按N次時7-Seg LED上會顯示fib(N)的值。User button長按1秒則將數值歸零。

fib(0) = 0、fib(1) = 1、fib(2) = 1 、…

若fib(N) 100000000則顯示-1。

1. 實驗結果

3-1

如何實作**delay 1 sec**

雖然老師上課已教到如何用timer實作delay，但本次實驗仍採用busy waiting的方式。由於實驗板子的cpu頻率為4MHz，要延遲一秒，便使其跑1000000次一個約佔4個clk cycle的loop。

**clk**的輸入問題

七段顯示器的init、display、sent message皆是參考老師講義的code。

在此之前，我以為七段顯示器的clk和FPGA上的lcd一樣，需要將cpu的clk透過接口給顯示器。看了範例後才恍然大悟，原來七段顯示器中所謂的’clk'，只是單純0101的變化，不需同步、頻率更是無所謂。如:範例中的clk，在0到1之間的cycle數與1到0的cycle數其實沒有相等。

3-2

如何讓數字持續顯示？

根據七段顯示器的特性，只要沒有覆寫新的程式碼進去，便會一直保留最後一次輸入的狀態，與FPGA的LCD不同。

**3-3**

如何顯示結果**?**

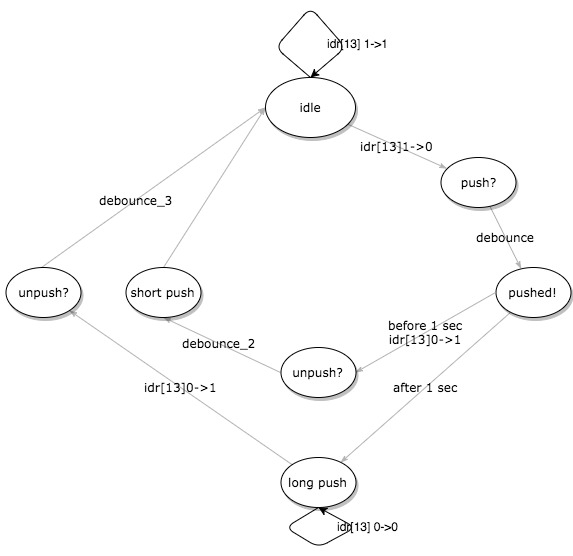
計算的結果是存在一個32bit的register，但是輸出時卻要拆成十位、百位、千位顯示。我的做法是將結果一次次的除以10，直到結果為0。

arms 除以無號數指令：udiv r0,r1,r2  
arms 除以有號數指令：sdiv r0,r1,r2

判斷長壓與短壓

我的做法是利用FSM的架構（用組語寫看不太出來），來判斷長壓短壓  
示意圖如下(我盡力了)：

**5.**實驗心得

這次的實驗要利用user button 和七段顯示器，因為lab4已經有接過GPIO的經驗了，這次實驗才進行得較快。老師講義裡的範例code幫了很多忙，諸如:如何給七段顯示器clk、如何將din切成16個bit...等。雖然如此，第三題還是讓我苦惱了許久，所幸最後有在時限內完成，實驗器材也一切平安。