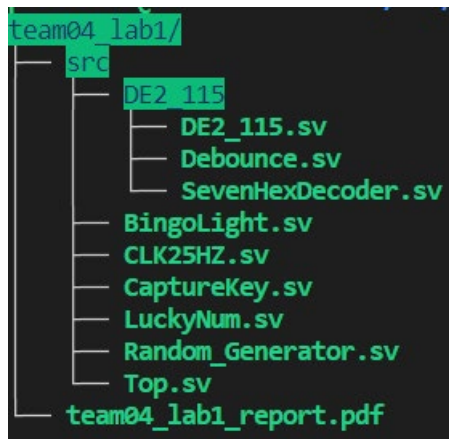
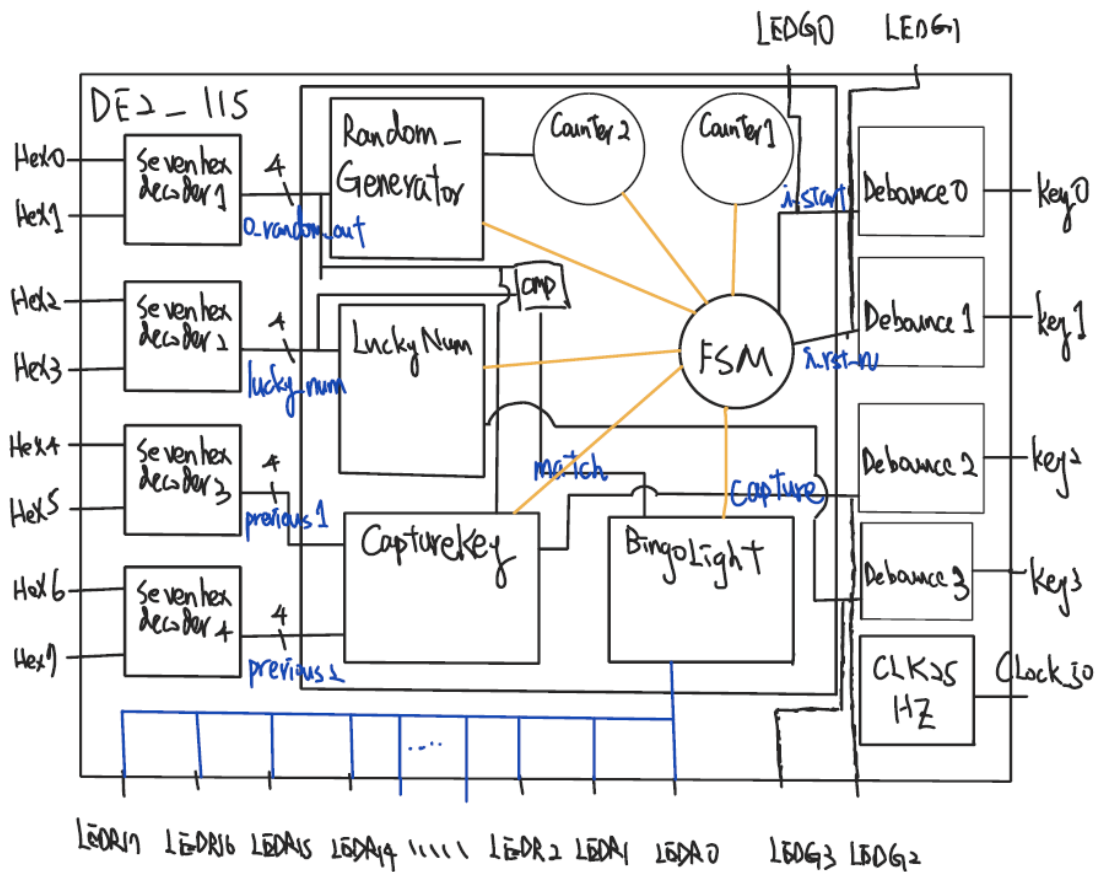


Team04_lab1_report

1. File structure



2. System architecture



大致的 Datapath 架構如上圖所示，FSM 會產生控制訊號去控制每個 submodule。每個 module 的詳細作用，會在 3. Hardware scheduling 詳述。

3. Hardware scheduling

- Module DE2_115

- a. SevenHexDecoder

我們有使用 4 個 sevenhexdecoder，分別顯示 random number、lucky number、previous1 和 previous2。當 key0 按下去會，top module 會產生一個 random number，並顯示在七段顯示器的 hex0、hex1。當 key3 按下去，top module 會重置一個 Lucky number，並顯示在七段顯示器的 hex2、hex3。當 key2 按下去則可以擷取任意時間產生的 random number 並顯示在七段顯示器的 hex4、hex5(previous1)以及 hex6、hex7(previous2)。詳細的控制訊號會在 top module 解釋。

- b. Debounce

Debounce module 主要是用來穩定 key 鍵按下產生的訊號，我們有使用 4 個 key，所以有 4 個 debounce module。值得注意的是，我們有用 debounce 產生的訊號去讓 LEDG 發光，因此我們可以透過發光的 LEDG 訊號，得知我們按了哪個 key 鍵。

- c. CLK25HZ

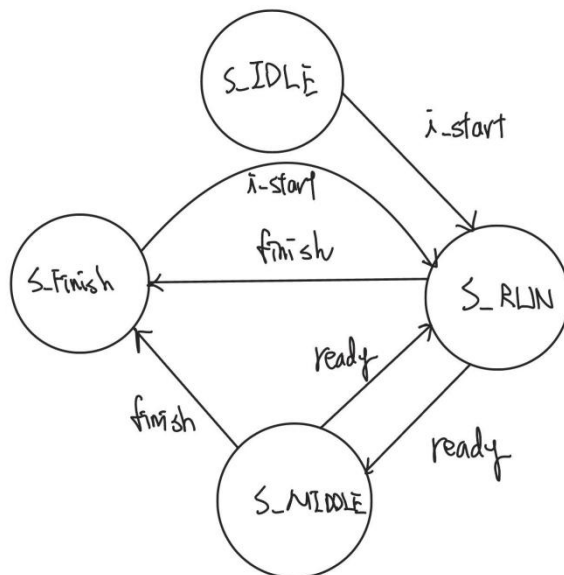
為了讓人眼可以看清楚七段顯示器數字的變化，我們把內建

50MHZ 的 CLK 轉成 25HZ。我們用了一個 counter 來計算原本內建 CLOCK_50 的週期，每當 CLOCK_50 過了 1M 個 cycle，也就是 counter 數到 999999，就會使新的 Clk_25HZ negate，使新的 CLK 週期變成原本 CLK 的 2M 倍。

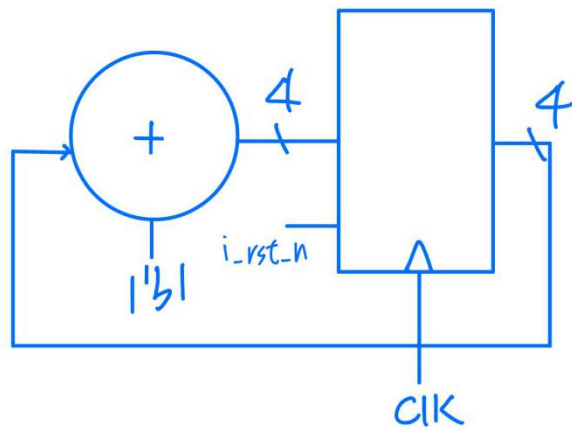
d. Top

Top module 主要由 FSM 和 4 個 submodule 和 兩個 counter 組成。

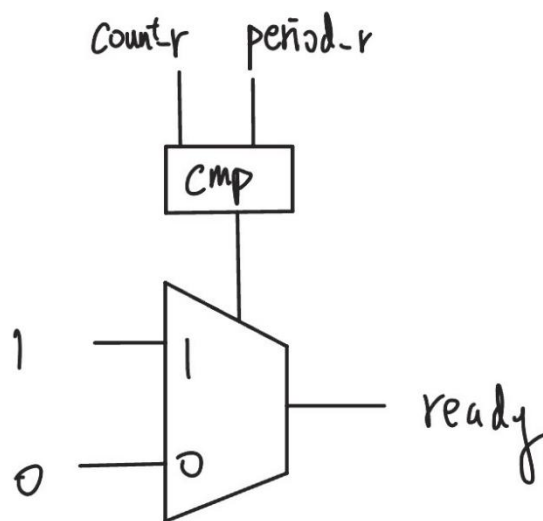
1. FSM



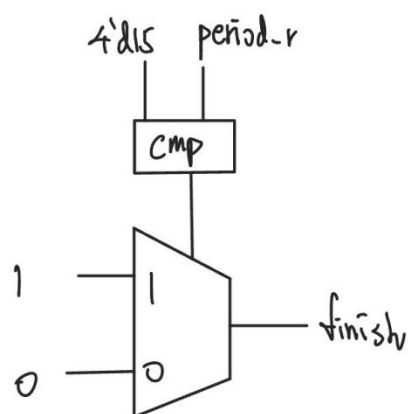
我們有兩個 counter 與 FSM 相關，分別是 count 和 period，period 是顯示目前 random number 會持續的時間，count 則是用來計 cycle，當 count 數到等於 period 時，ready 就會變 1，使得 FSM 跳下一個 state，此時 period 也會加 1，使得下一個 random number 顯示的時間長一個 cycle；當 period 加到 15 時，finish 就會變 1，使得 FSM 跳到 S_FINISH 的 state。



(上圖藍色的圖是我們 counter 主要的結構)



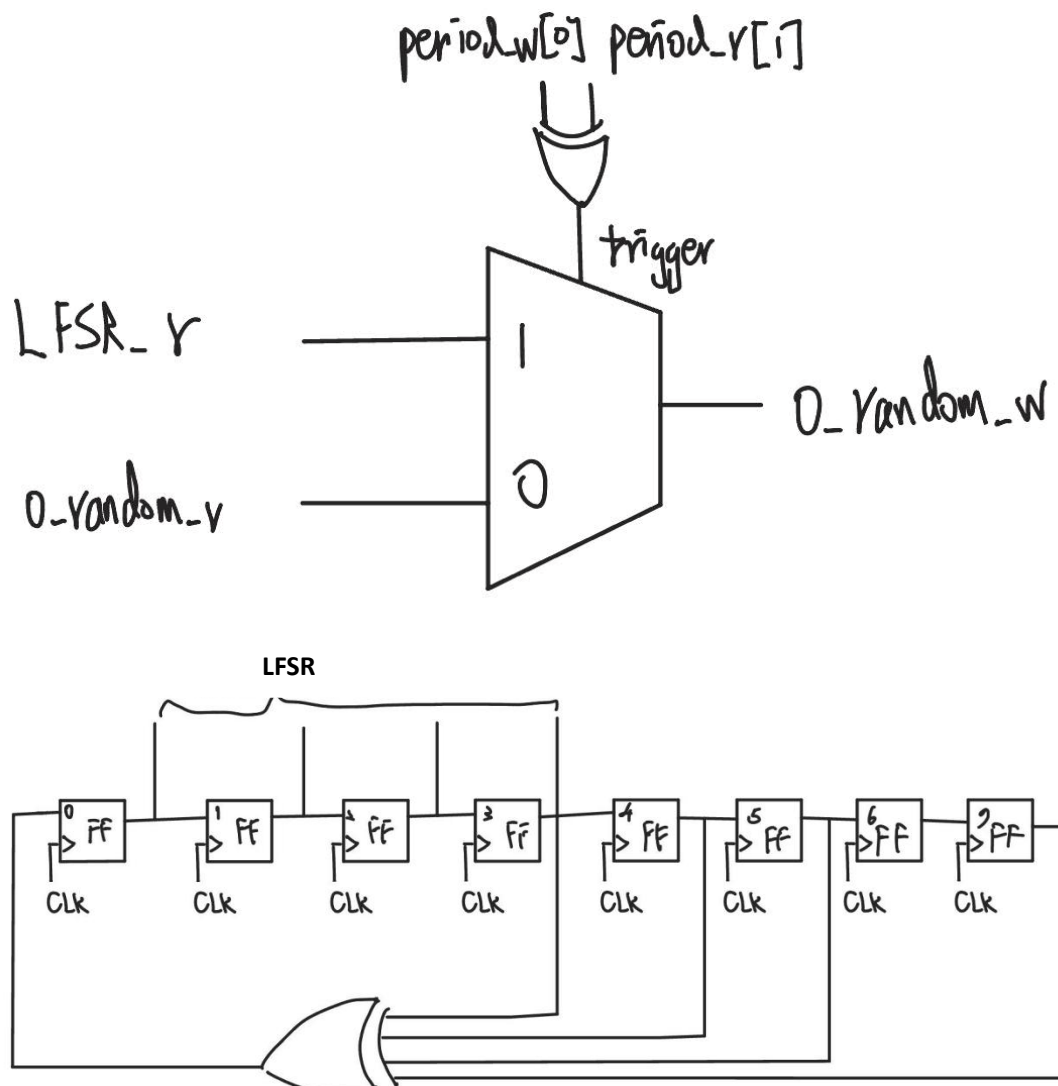
(ready 訊號的產生)



(finish 訊號的產生)

2. RandomGenerator

我們使用 LFSR 的演算法來計算 random number，我們取第 3、4、5、7 bit 來做 XOR 當作 feedback；而當接收 trigger 為 1 時，就會將 random number 換成目前 LFSR 的值；另外，我們有用一個 8-bit counter(上述藍色結構)的值當作按下 i_start 時的 seed。



3. LuckyNum

當我們按下 `KEY[3]` 時，會根據一個上述藍色結構的 4-bit

counter 來產生一個 lucky number。

4. CaptureKey

我們用兩個 flip-flop 去存取 previous1 和 previous2，每當按下 KEY[2]時，previous1 就會存取當前 random number 的值，而 previous2 則會存取 previous1 的值。

5. BingoLight

接收來自 Top module 的控制訊號 match，match 訊號是由一個 comparator(在上述之 block diagram)，比較 lucky number 和上次停下來 random number 是否相同，如果相同則 match 會變成 1，然後 light 訊號就會不斷 negate，使得 LEDR 不斷閃爍。

4. Fitter Summary

Fitter Summary	
Fitter Status	Successful - Mon Mar 13 17:44:23 2023
Quartus II 64-Bit Version	15.0.0 Build 145 04/22/2015 SJ Full Version
Revision Name	DE2_115
Top-level Entity Name	DE2_115
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	156 / 114,480 (< 1 %)
Total combinational functions	147 / 114,480 (< 1 %)
Dedicated logic registers	89 / 114,480 (< 1 %)
Total registers	89
Total pins	480 / 529 (91 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	0 / 4 (0 %)

5. Time analyer

Table of Contents				Unconstrained Paths			
>	Estimated Delay Added				Property	Setup	Hold
	Messages			1	Illegal Clocks	0	0
	Suppressed Messages			2	Unconstrained Clocks	1	1
	Flow Messages			3	Unconstrained Input Ports	4	4
	Flow Suppressed Message			4	Unconstrained Input Port Paths	98	98
>	Assembler			5	Unconstrained Output Ports	65	65
✓	TimeQuest Timing Analyz			6	Unconstrained Output Port Paths	181	181
	Summary						
	Parallel Compilation						
	SDC File List						
	Clocks						
>	Slow 1200mV 85C Moc						
>	Slow 1200mV 0C Mode						
>	Fast 1200mV 0C Model						
	Multicorner Timing Ana						
	Multicorner Datasheet F						
>	Advanced I/O Timing						
>	Clock Transfers						
	Report TCCS						
	Report RSKM						
	Unconstrained Paths						
	Messages						

6. 心得

- 1.一開始在設定 Quartus 時，遇到不少問題，包括筆電無法搜尋到 usb_blaster，以及無法 import assignment。
2. 儘管之前在積體電路設計和計算機結構都有寫過 verilog，但這次的作業要寫的 code 數相較於之前多很多，讓我開始真正熟悉 verilog 的使用，除此之外也想了一些額外的功能讓亂數產生器更多元。希望經過這學期的課程讓我喜歡上打 verilog。
3. 之前我們沒有使用過 fpga，透過這次的 lab，我們也對 fpga 的腳位更熟悉。