105061254 林士平 邏輯設計實驗報告 pre_Lab5

1.

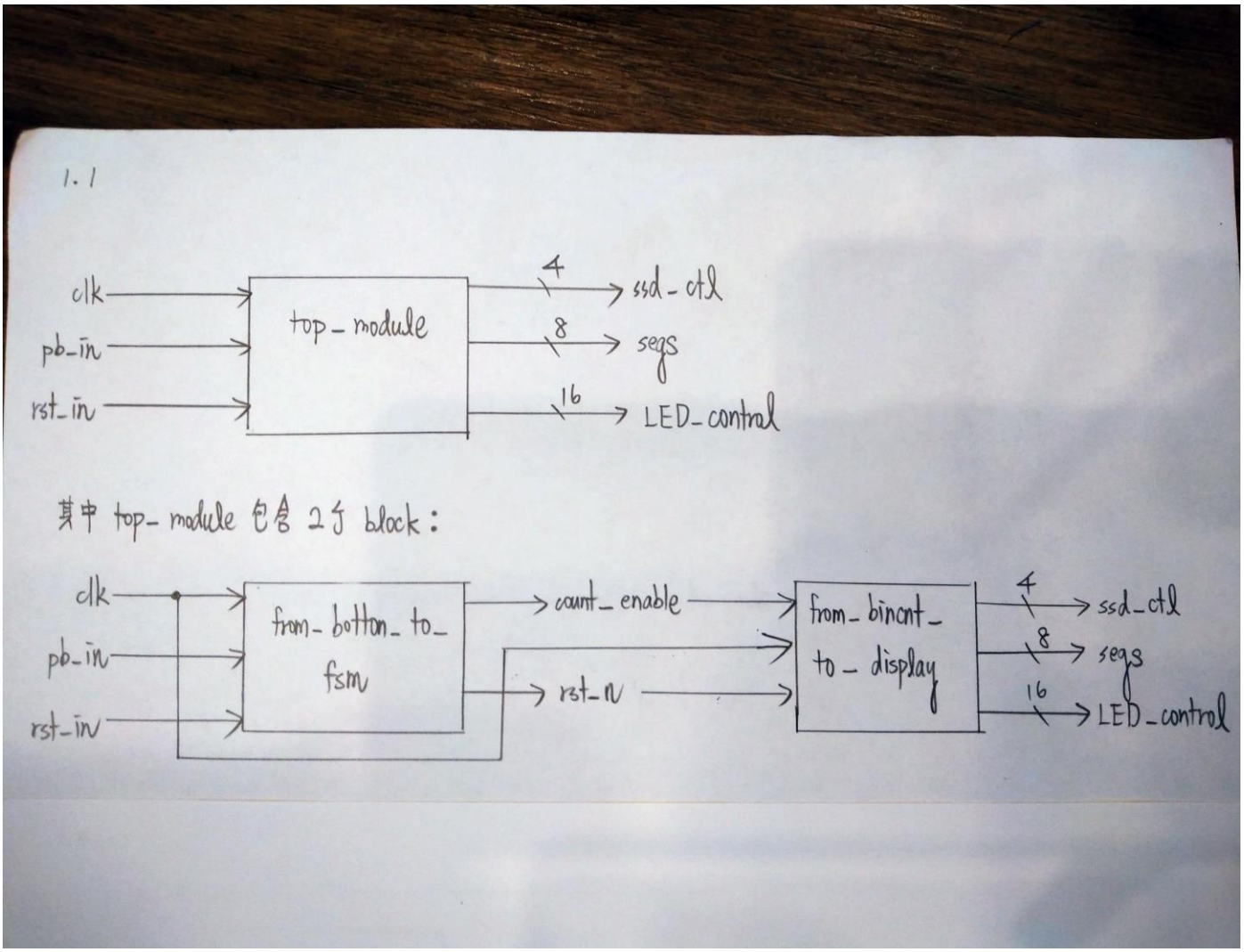
(1) Design specification :

A. Inputs and outputs(表一)：

Inputs	clk, rst_in, pb_in
Outputs	ssd_ctl[3:0], segs[7:0], LED_control[15:0]

↑表一：Inputs and outputs of 1.1

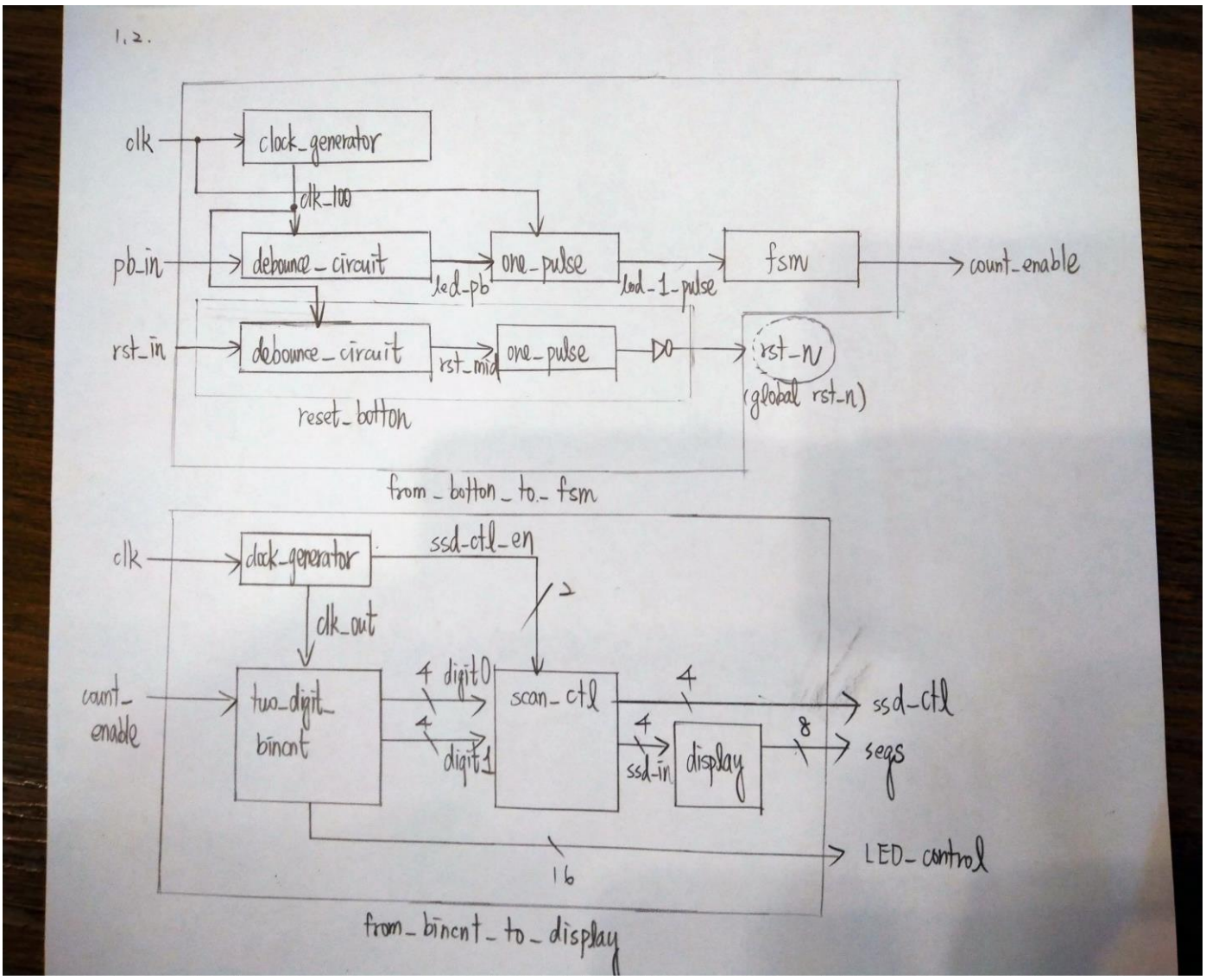
B. Block diagram(function table)(圖一)：



↑圖一：The block diagram of 1.1

(2) Design implementation :

A. Logic diagram(function table)(圖二)：



↑圖二：logic diagram of 1.1

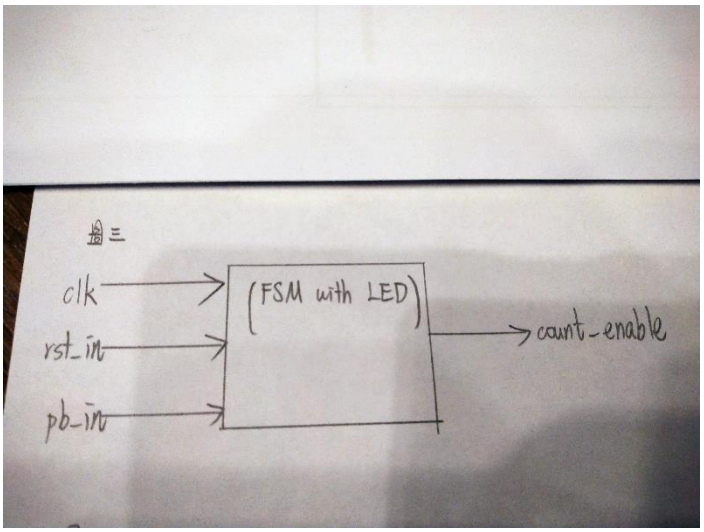
(3)

a. Design specification :

(a) Inputs and outputs(表二)：

Inputs	clk, rst_in, pb_in
Outputs	count_enable
↑表二：Inputs and outputs of 1.3	

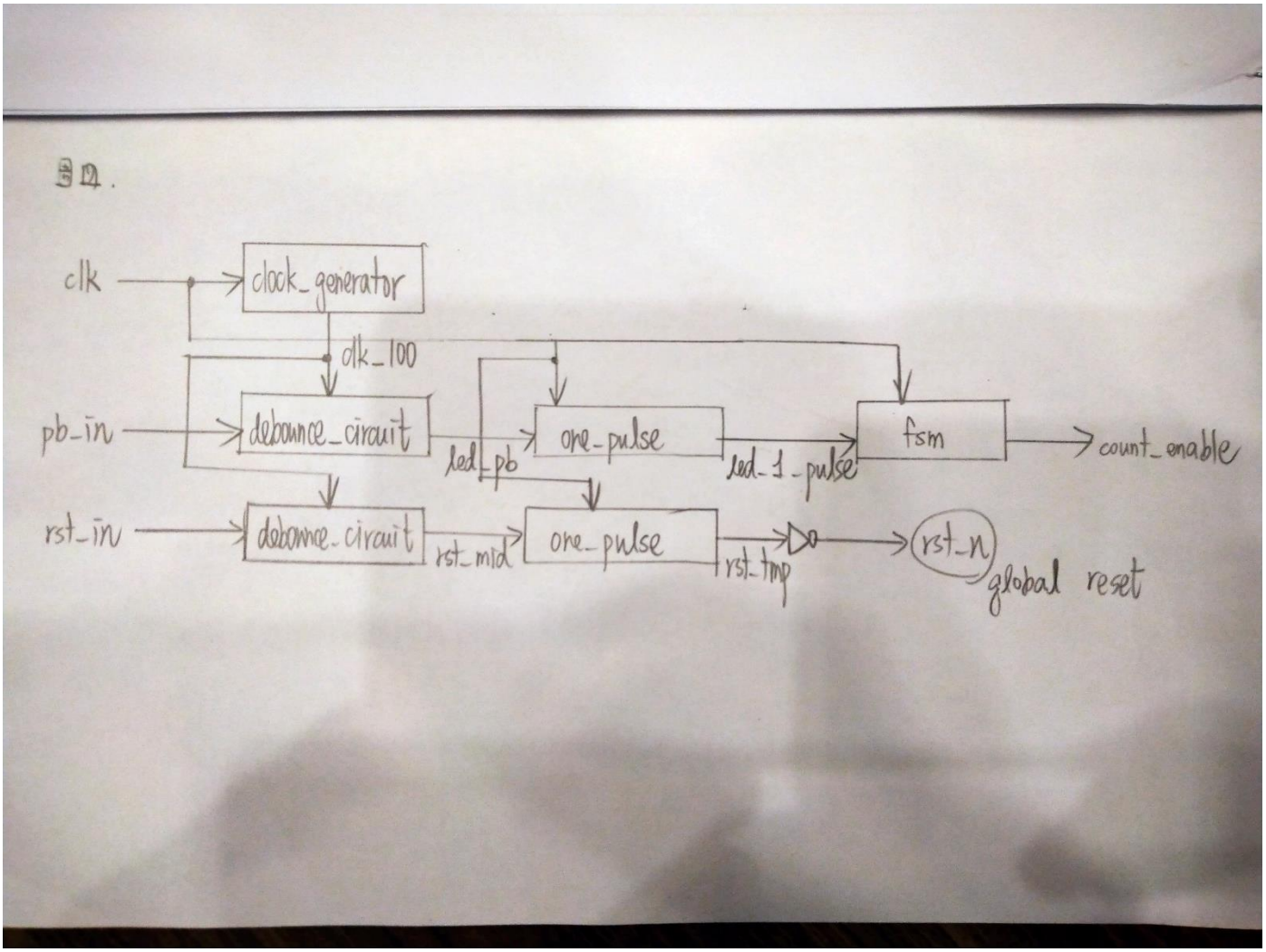
(b) Block diagram(圖三)：



↑圖三：The block diagram of 1.3

b. Design implementation :

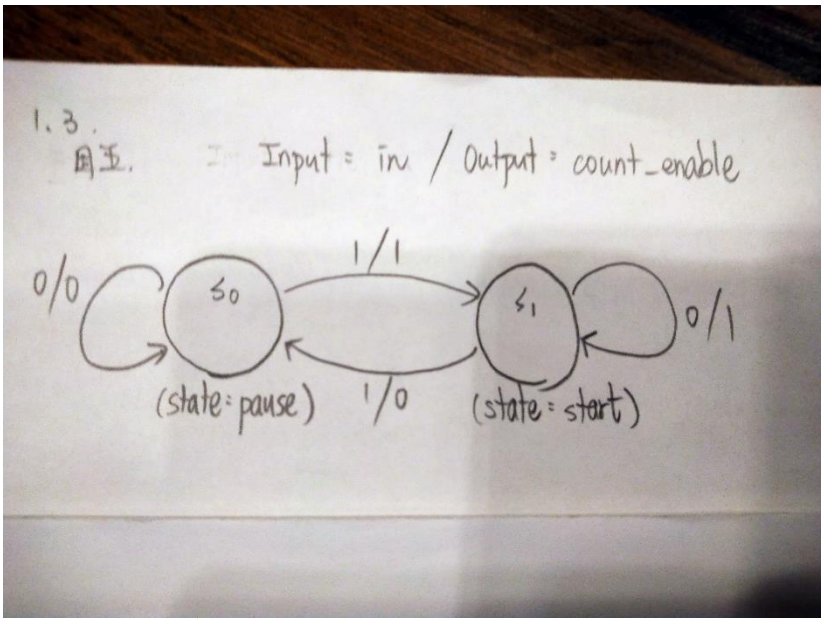
(a) Logic diagram (圖四)：



↑圖四：logic diagram of 1.3

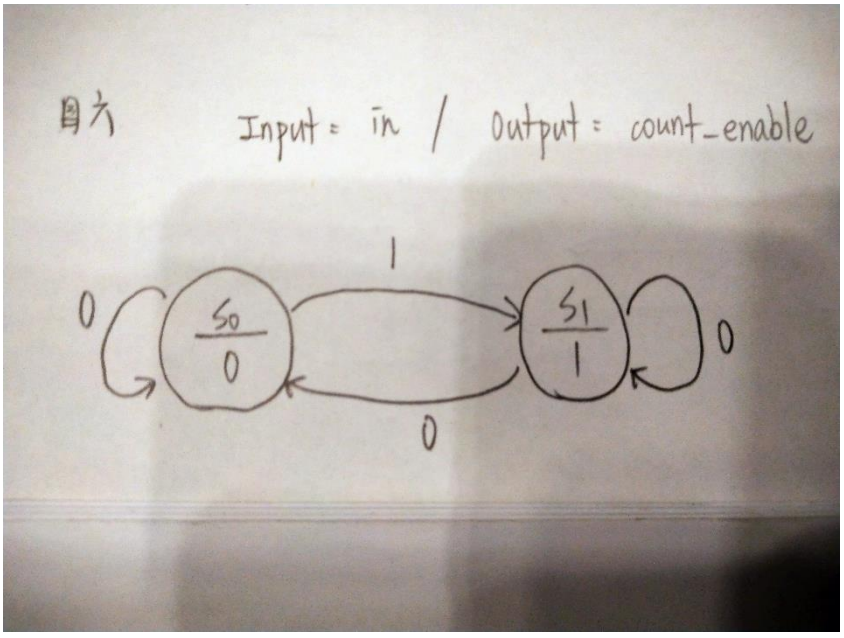
(c) state graph of fsm：

在此 finite state machine 可以用兩種做法，一種是 Mealy machine， output 會同時受到 input 和 present state 的影響，state graph 如圖五所示：



↑圖五：the state graph of pause/start function(using Mealy machine)

另一種為 Moore machine，output 只會和 present state 有關，state graph 如圖六所示：



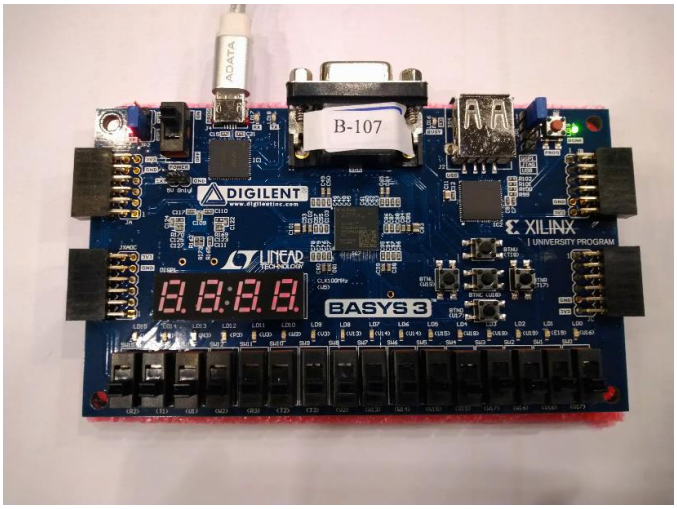
↑圖六：the state graph of pause/start function(using Moore machine)

(d) I/O pin assignment(表三)：

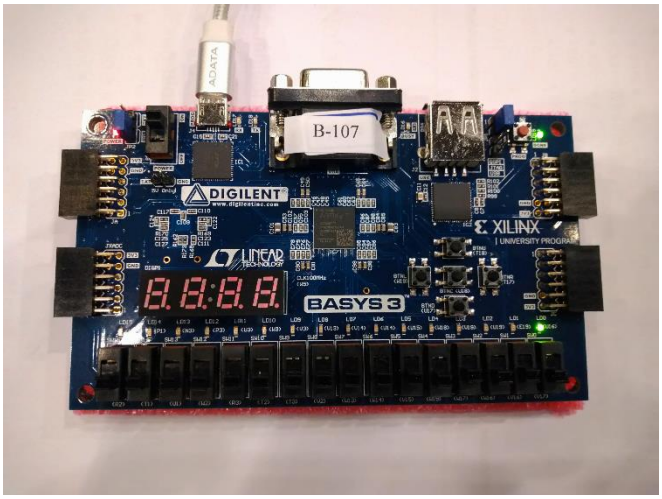
I/O	clk	count_enable	pb_in	rst_in
LOC	W5	U16	T17	U17
↑表三：I/O pin assignment of 1.3				

c. result（圖七）：

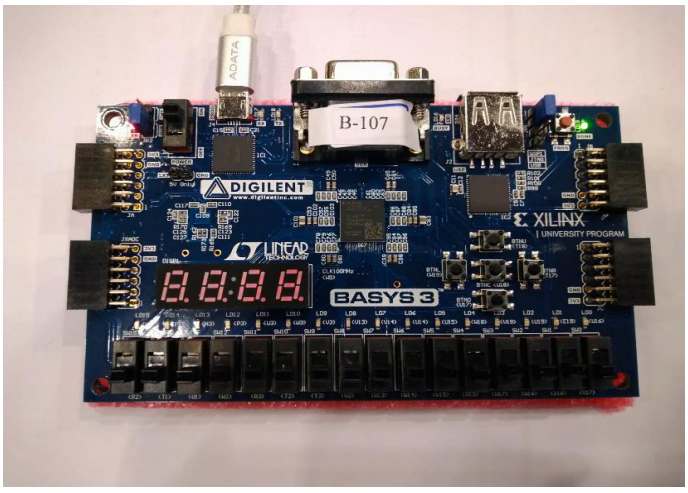
不管用 Mealy machine 或是 Moore machin，最後用 FPGA 實現的結果是相同的（圖七）：



↑圖七之一：用 FPGA 板實現 1.3(一開始 state = S0，LED 不亮)



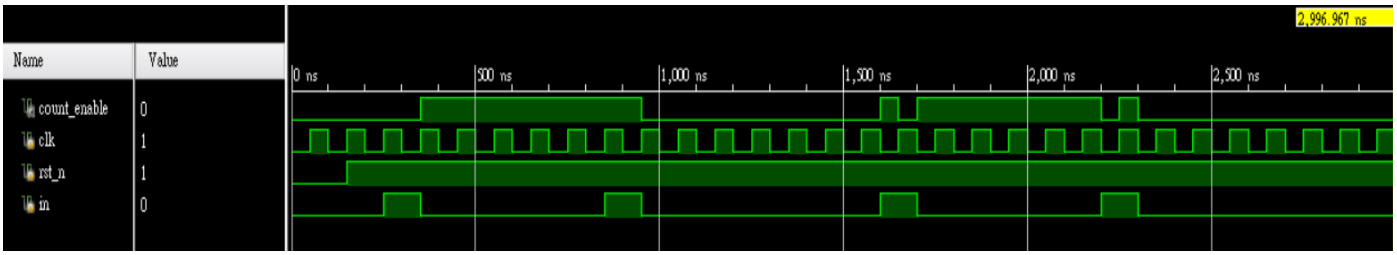
↑圖七之二：用 FPGA 板實現 1.3(按一下 in 按鈕，state = S1，LED 亮)



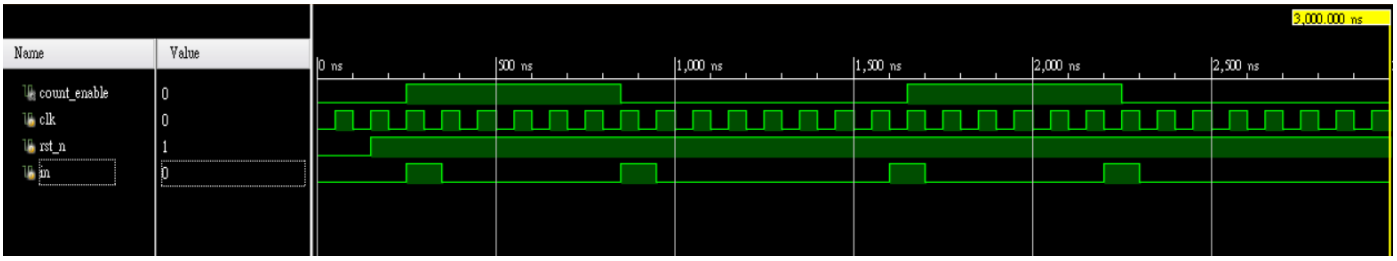
↑圖七之三：用 FPGA 板實現 1.3(再按一下 in 按鈕，state = S0，LED 不亮)

值得一提的是，雖然 Moore machine 和 Mealy machine 用 FPGA 板實現的結果相同，但他們用 verilog 模擬的結果會因為 input 給的時機不同而有不一樣的結果(見 1.4 小題)。

(4) (verilog 模擬只用圖四中 fsm 的部分，沒有 debounce、one_pulse、clock_generator)



↑圖八：verilog 模擬結果(Using Mealy machine)



↑圖九：verilog 模擬結果(Using Moore machine)

比較圖八和圖九可以發現，在 0~1000ns 的部分，不管是 Mealy machine 和 Moore machine 的結果均符合預期：第一次按 in，state(count_enable)從 0 -> 1，並維持在 state = 1；第二次按 in，state(count_enable)從 1 -> 0，並維持在 state = 0。但是在 1000ns~2500ns 的部分，Mealy machine 出現了我們不想要的 state 的跳動，這個現象稱為 glitch，在 Mealy machine 特別容易發生(因為 output 會同時 depend on input 和 state)，而 Moore machine 在此可以避免這個跳動。由此可見，input 給的時機和 clk 之間的關係非常重要，如果沒有在正確時機給入，便會造成我們不想要的結果。

所以，加入 one_pulse module 便是為了要避免 glitch 的現象，讓 input 給的時機可以和 clock 的 positive edge 切齊(如 0~1000ns 的部分)，也因此 1.3 小題(圖四)實際 implement 到 FPGA 板上的結果不管 fsm 用 Mealy machine 還是 Moore machine 均有我們要的結果。