

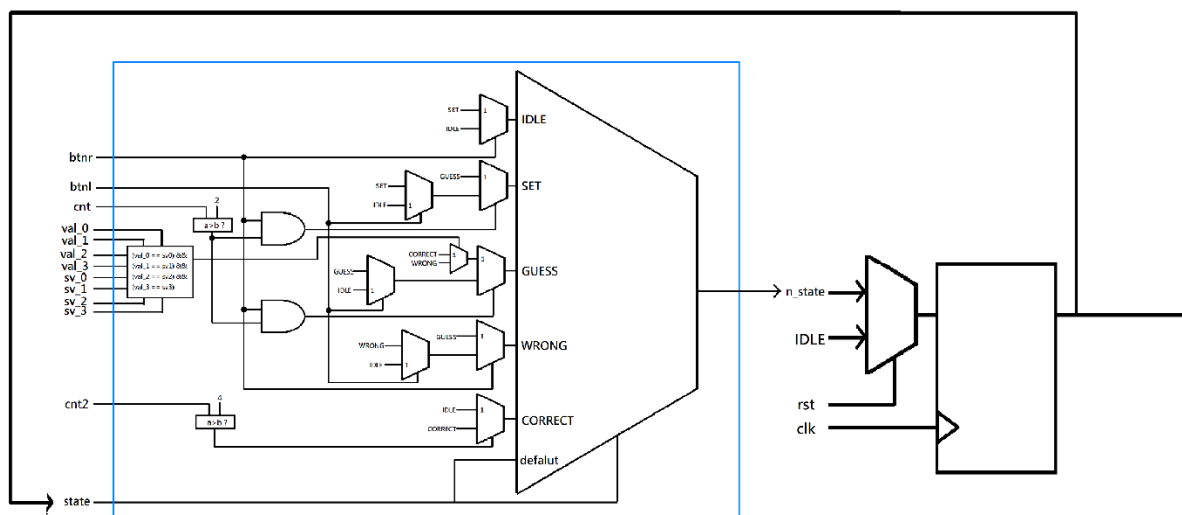
## Lab 5

學號:109062110

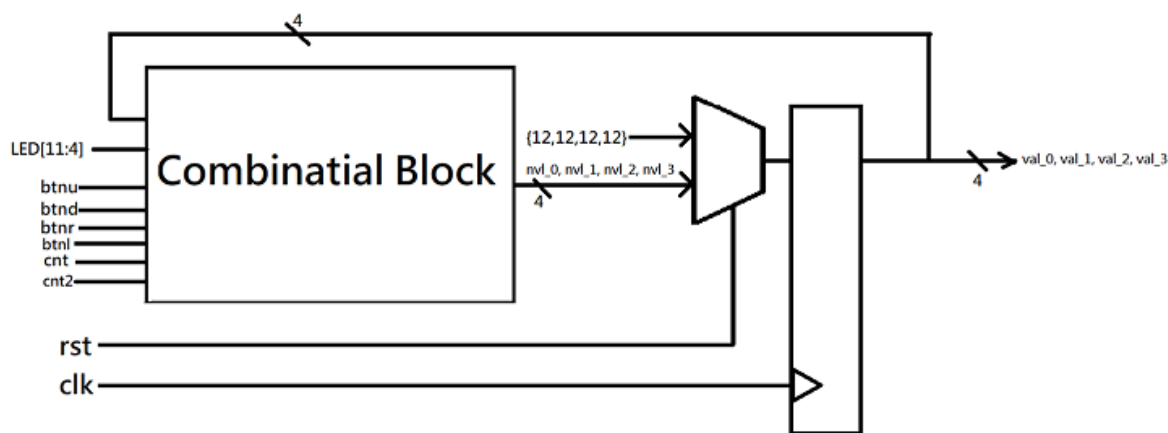
姓名:祝語辰

### A. Lab Implementation

#### 1. Block diagram



1 FSM 控制 state 的變化



2 處理 val\_0~3，這個值是要用來送去顯示在 7segment 中

#### 2. Partial code screenshot with the explanation

1	<pre>//counter for btn always @(posedge clk) begin     if(rst) cnt &lt;= 0;     else cnt &lt;= n_cnt; end  always @(*)begin     case(state)         SET:begin             if(btrn &amp;&amp; cnt &lt; 3) n_cnt = cnt + 1;             else if(btrn &amp;&amp; cnt &gt; 2) n_cnt = 0;             else if(btrnl) n_cnt = 0;             else n_cnt = cnt;         end         GUESS:begin             if(btrn &amp;&amp; cnt &lt; 3) n_cnt = cnt + 1;             else if(btrn &amp;&amp; cnt &gt; 2) n_cnt = 0;             else if(btrnl) n_cnt = 0;             else n_cnt = cnt;         end         default:n_cnt = 0;     endcase end</pre>	<pre>//counter for 1 sec always @(posedge sec)begin     cnt2 &lt;= n_cnt2; end  always @(*)begin     case(state)         CORRECT:begin             n_cnt2 = cnt2 + 1;         end         default:begin             n_cnt2 = 0;         end     endcase end</pre>
---	--	---

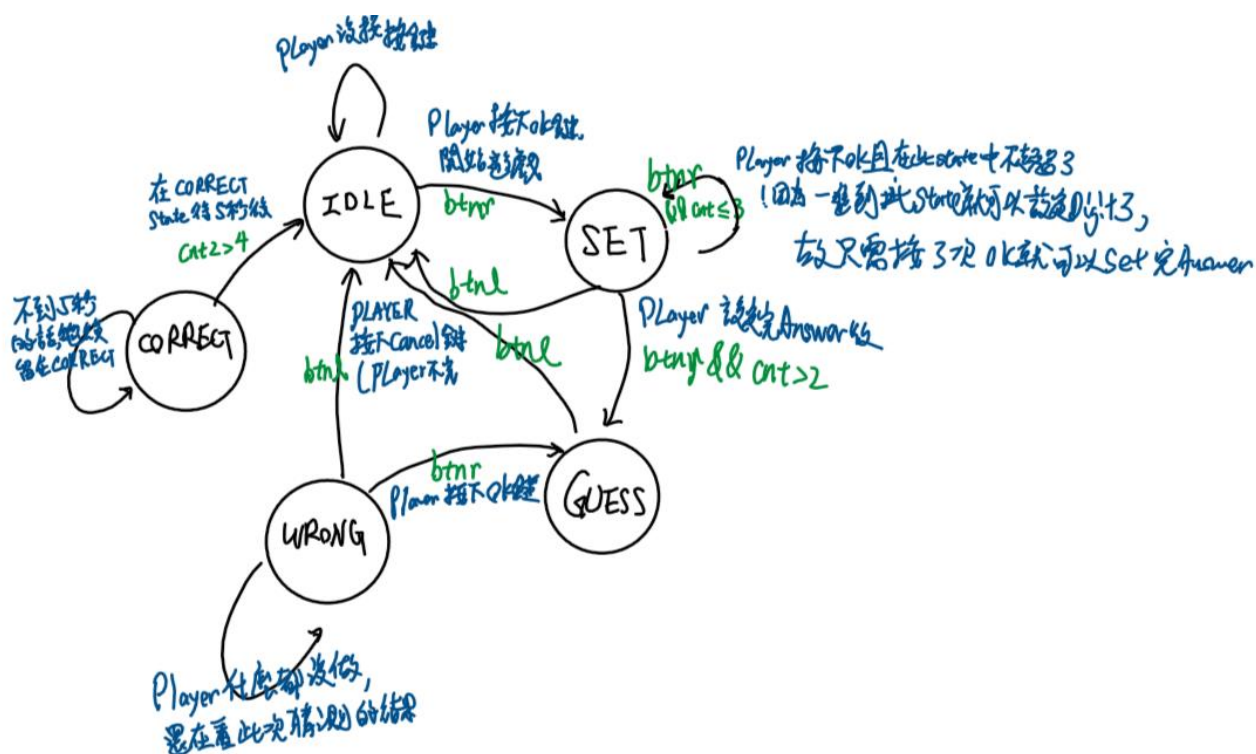
左側的 code 是兩個 counter。

Cnt(靠左)是紀錄特定按鍵在一個 state 中按了幾次，用來做 state 的轉換。

Cnt2(靠右)是在 Correct state 中，每一秒加一，用來實現在 correct state 中 5 秒後進入到 IDLE 中。

2	<pre>//led always@(posedge clk)begin     if(rst) led &lt;= {4'b1111, 12'b00};     else led &lt;= n_led; end  always @(*) begin     case(state)         IDLE:begin             if(btrn) n_led= {4'b0, 1'b1, 11'b00};             else n_led = {4'b1111, 12'b00};         end         SET:begin             if( btrn &amp;&amp; cnt &lt; 3 ) n_led = led &gt;&gt; 1;             else if( btrn &amp;&amp; cnt &gt; 2 ) n_led = {0'b0, 1'b1, 7'b00};             else if(btrn) n_led = {4'b1111, 12'b00};             else n_led = led;         end         GUESS:begin             if( btrn &amp;&amp; cnt &lt; 3 ) n_led = led &gt;&gt; 1;             else if( btrn &amp;&amp; cnt &gt; 2 ) n_led = ( (val_0 == sv0) &amp;&amp; (val_1 == sv1) &amp;&amp; (val_2 == sv2) &amp;&amp; (val_3 == sv3) )?{16'b0, 4'b1111};             else if(btrn) n_led = {4'b1111, 12'b00};             else n_led = led;         end         WRONG:begin             if(btrn) n_led = {0'b0, 1'b1, 7'b00};             else if(btrn) n_led = {4'b1111, 12'b00};             else n_led = led;         end         CORRECT:begin             if(cnt &gt; 4) n_led = {4'b1111, 12'b00};             else n_led = ~led;         end         default:begin             n_led = led;         end     endcase end</pre>	左側的 code 是用來控制 led 的顯示。
3	<pre>//save the answer always@(posedge clk)begin     if(rst)begin         sv0 &lt;= 0;         sv1 &lt;= 0;         sv2 &lt;= 0;         sv3 &lt;= 0;     end else begin         sv0 &lt;= nsv_0;         sv1 &lt;= nsv_1;         sv2 &lt;= nsv_2;         sv3 &lt;= nsv_3;     end end  GUESS:begin     if(btrn &amp;&amp; cnt &gt; 2 ) n_state = ( (val_0 == sv0) &amp;&amp; (val_1 == sv1) &amp;&amp; (val_2 == sv2) &amp;&amp; (val_3 == sv3) )?CORRECT:WRONG;     else if(btrn) n_state = IDLE;     else n_state = GUESS; end  always@(*)begin     case(state)         SET:begin             if( btrn &amp;&amp; cnt &gt; 2 ) begin                 nsv_0 = val_0;                 nsv_1 = val_1;                 nsv_2 = val_2;                 nsv_3 = val_3;             end             else begin                 nsv_0 = sv0;                 nsv_1 = sv1;                 nsv_2 = sv2;                 nsv_3 = sv3;             end         end         default: begin             nsv_0 = sv0;             nsv_1 = sv1;             nsv_2 = sv2;             nsv_3 = sv3;         end     endcase end</pre>	這邊是用來在儲存設定的 answer。以及該值用來轉換 state 的部分。

3. Event-based finite state machine (FSM) with the explanation:  
請看圖內文字說明。



## B. Questions and Discussions

- A. Please explain how your design identifies the correct answer set in the SET\_ANSWER state when the machine is in the other state.

在離開 SET\_ANSWER 前，會先將四個 Digit 的 Value 紀錄在內部儲存的 reg 中，以利在離開 GUESS 前做比較。簡單來說，就是在處理 state 的 always block 用 if 來決定 next\_state 是什麼。If 的條件就是將四個 digit 與對應的答案比，一樣的話就回傳 true，並用 and 做連結。

```
GUESS:begin
  if(btnr && cnt > 2 ) n_state = ( (val_0 == sv0) && (val_1 == sv1) && (val_2 == sv2) && (val_3 == sv3) )?CORRECT:WRONG;
  else if(btnl) n_state = IDLE;
  else n_state = GUESS;
end
```

- B. Please explain how your design calculates the number of A and B.

因為題目中有假設使用者在設定答案，以及猜答案的時候都不會出現重複的值，也就是說，不需要去考慮這些情形，因此在判斷幾 A 幾 b 時，我的作法如下，在離開 SET\_ANSWER 時，會記錄設定的答案，在離開 GUESS 時，會做比較，紀錄對應位置值相同的數量給 A，而 B 的部分，以 Digit3 為例，他會檢查 Digit3~0，只要有就紀錄。最後在顯示，因此假如設定的答案是 0000，則在進入 CORRECT 時，顯示的是 4A4B，雖然有此 bug，但因為題目的需求，所以受影響。

- C. Please explain how to control LEDs to blink at a frequency of 1 second in the CORRECT state.

下圖是控制 LED 顯示的 always block，看到在 cnt2 > 4 之前，每進到這個 block 就會將 led 由亮變暗，暗變亮。而 cnt2 則是在另外一個 sequential block 控制，其更新的 clock 則是 1sec，如此以達到 LED change every second in CORRECT state

```
end
CORRECT:begin
  if(cnt2 > 4) n_led = {4'b1111, 12'b0};
  else n_led = ~tled;
end
default:begin
```

## D. Problem Encountered

因為這一次的作業做起來可以明顯感受到相較於前幾次(也比期中考都比較簡單，所以碰到的問題，比較小，多是在前幾次 lab 有處理過的小細節，只是不小心忽略了。在加上這此我終於依照老師推薦的 coding style 去寫了，所以這次沒有碰到什麼稀奇古怪的問題。

## E. Suggestions

無