

## Lab 5

學號: 108032053

姓名: 陳凱揚

### 1. 實作過程

在此題中，我使用的 clock 訊號頻率為 2000Hz，並使用 count\_sec 和 count\_clk 記秒，達成 LED 以 0.5Hz 閃爍、在 RELEASE 時計 5 秒、在 CHANGE 時每秒扣錢等目的。此外，我有另建一個 module BCD 將 money、price 等數字轉成 BCD 表示，如下圖 1、圖 2 所示，由於此題的數字最大只可能到達 54，只要簡單判斷十位數字為何，即可得知個位數，並輸出此兩位數的 BCD。

在 module lab5 中，我使用了 9 個 flip-flop，功能分別如下：

- (1) state：記錄目前狀態，有 6 個狀態，state diagram 如下圖 3 所示，其中 Up、Dp、Lp、Cp、Rp 分別為經過 debounce、onepulse 處理的 BTN 訊號。
- (2) person：記錄目前選中的顧客，有 3 種客人，分別為 C(child)、S(student)、A(adult)，在 IDLE 或 TYPE 時可以以按鍵選擇。
- (3) amount：記錄目前票的張數，範圍為 1~3 張，在 AMOUNT 時可以以按鍵增減。
- (4) money：記錄目前投入的金額，在 PAYMENT 時根據按鍵增減，並且在進入 RELEASE 時，減掉 price；在 CHANGE 時，當 count\_clk 為 2000 (代表經過了 1 秒)，若  $money \geq 5$ ，減 5 元，否則減 1 元。
- (5) price：在 IDLE、TYPE 時，作為目前選中的票價；在 PAYMENT 時作為應付票價。
- (6) count\_sec：在 RELEASE 時，當 count\_clk 為 2000，加 1，到達 5 時即進入下一個 state。
- (7) count\_clk：在 IDLE、RELEASE、CHANGE 時記數經過的 clock 數量，每 2000 一循環，也就是每一秒一循環。
- (8) nums：以 BCD 的方式表示目前顯示的 4 個數字，根據不同 state，輸出不同結果。
- (9) LED：控制 LED 燈的亮暗，在 IDLE 和 RELEASE 時，當 count\_clk 為 2000，即由亮轉暗或由暗轉亮。

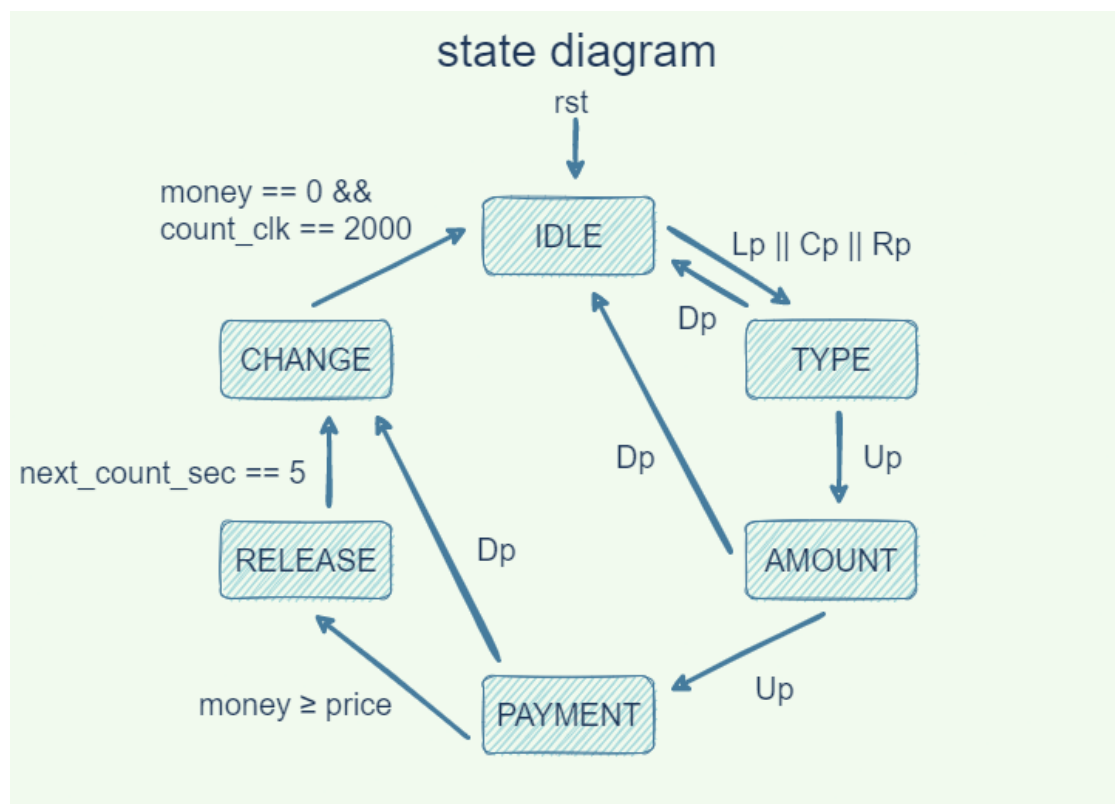
下圖 3、圖 4 分別為 state diagram 及 block diagram。

```
// next_money_BCD, next_price_BCD
BCD b1(.binary(next_money), .BCD(next_money_BCD));
BCD b2(.binary(next_price), .BCD(next_price_BCD));
```

▲ 圖 1

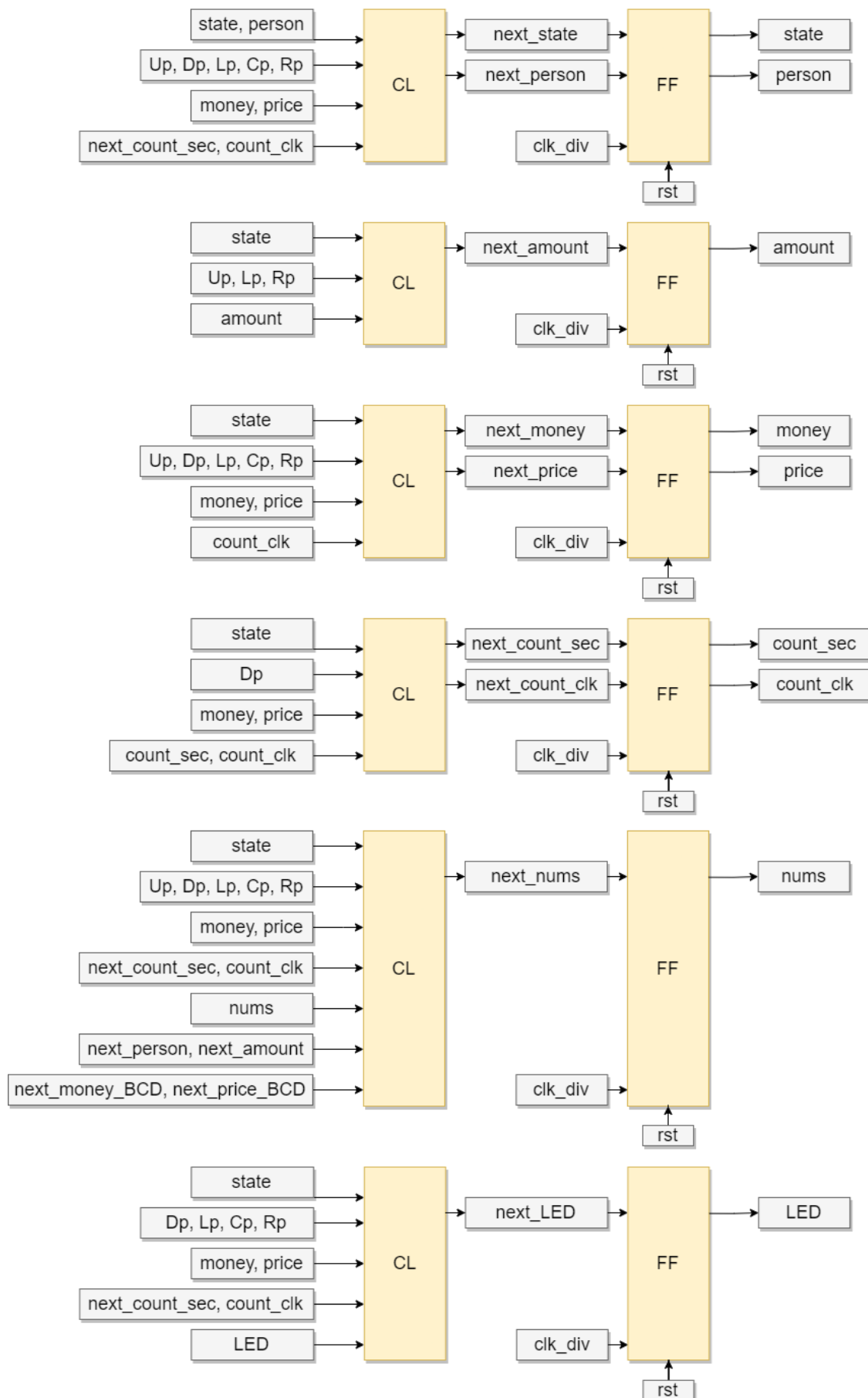
```
module BCD(
    input [7:0] binary,
    output reg [7:0] BCD);
    wire [7:0] b1, b2, b3, b4, b5;
    assign b1 = binary-8'd10;
    assign b2 = binary-8'd20;
    assign b3 = binary-8'd30;
    assign b4 = binary-8'd40;
    assign b5 = binary-8'd50;
    // BCD
    always @* begin
        BCD = binary;
        if(binary < 8'd10) BCD = binary;
        else if(binary < 8'd20) BCD = {4'd1, b1[3:0]};
        else if(binary < 8'd30) BCD = {4'd2, b2[3:0]};
        else if(binary < 8'd40) BCD = {4'd3, b3[3:0]};
        else if(binary < 8'd50) BCD = {4'd4, b4[3:0]};
        else if(binary < 8'd60) BCD = {4'd5, b5[3:0]};
    end
endmodule
```

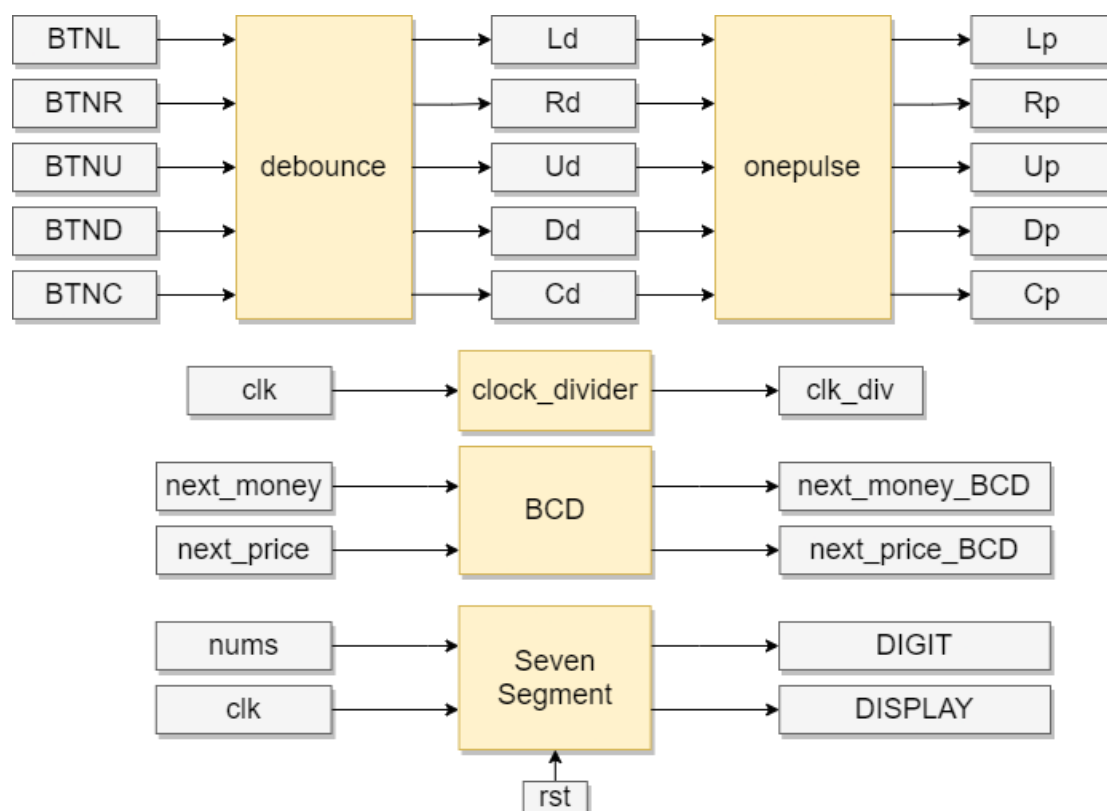
▲ 圖 2



▲ 圖 3

# block diagram





▲ 圖 4

## 2. 學到的東西與遇到的困難

這次的 lab5 像是 lab3 和 lab4 的結合，讓我更熟悉了對於 LED、Switch 的控制，還有七段顯示器，在 lab4 時，我使用了 4 個變數儲存了七段顯示器上的 4 個數字，並直接以 BCD 來判斷是否需要進位或到達最大值等，這使得 code 很複雜又混亂。而這次我選擇先以一般的數字做計算，再建一個 module 將數字轉換成以 BCD 表示，使整個程式碼變得比較簡潔又易讀，我覺得這樣會是比較好的處理方法，雖然目前的 BCD module 還很陽春，但將來希望能慢慢加強這個轉換的 module。

此外，這次我採用了老師的建議，只使用 1 個 clock 訊號，其他不同的頻率都以 count\_clk 來達成，我覺得幫助蠻大的，減少了很多 debug 的時間，架構出來後就不容易產生奇怪的錯誤。

## 3. 想對老師或助教說的話

請問為什麼宋朝都沒有外送？

...

因為他們很多契丹人。