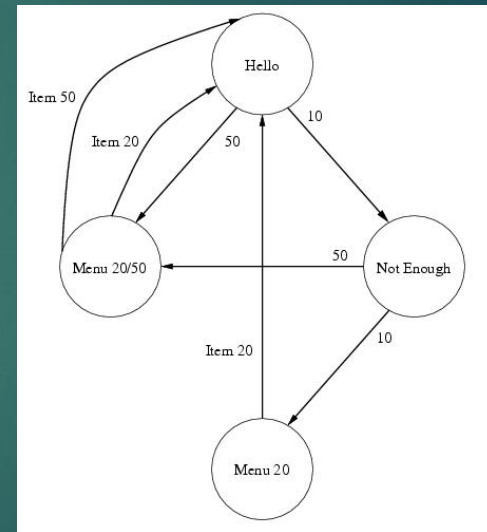
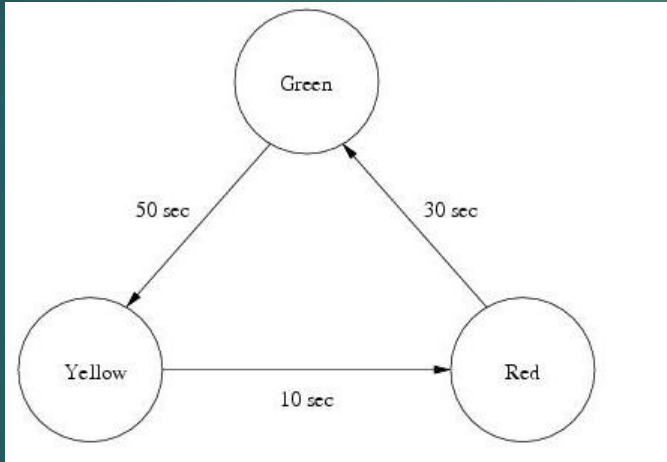


數位邏輯設計

20210521

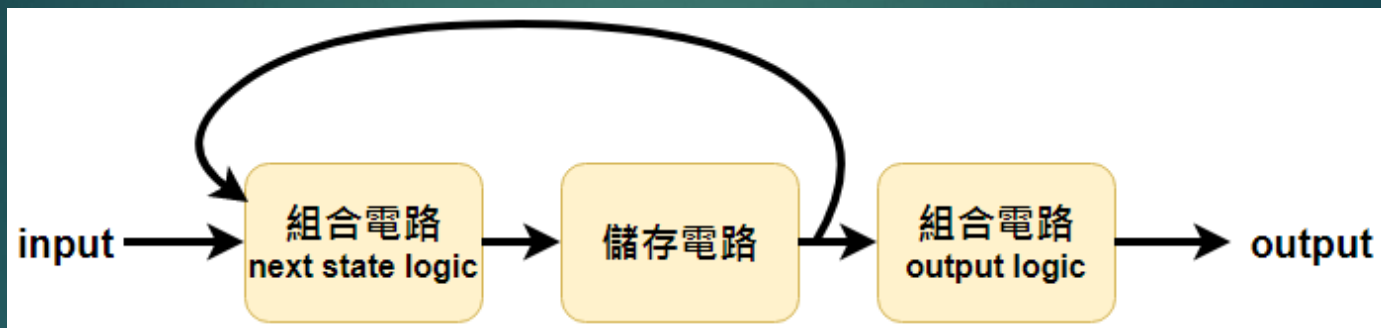
有限狀態機 Finite State Machine

- ▶ 有限狀態機是由一組狀態、一個起始狀態、輸入、將輸入與現在狀態轉換為下一個狀態的轉換函數所組成。
- ▶ Exp.紅綠燈、自動販賣機

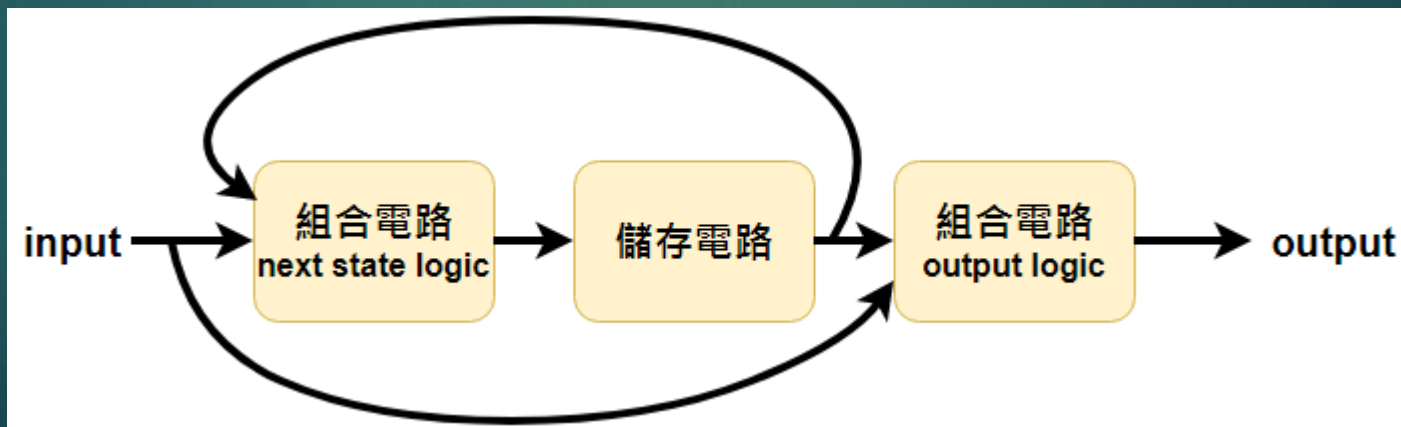


有限狀態機 Finite State Machine

- ▶ 主要有兩類
- ▶ Moore machine:輸出邏輯只和當下的狀態相關，和輸入並無關係。



- ▶ Mealy machine:在輸出時，除了不同的狀態切換到不同的輸出之外，且這個輸出也同時被輸入影響著。





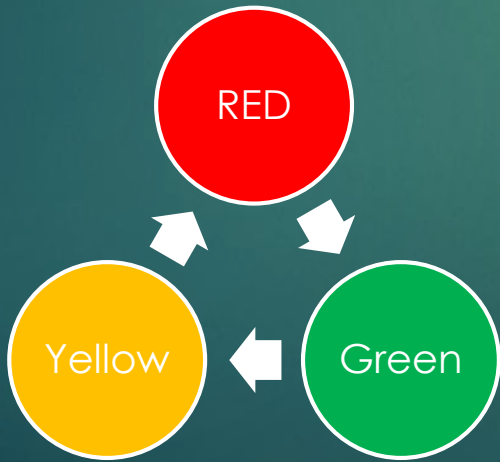
Homework2-1:紅綠燈說明

- ▶ 綠燈為600ms後，黃燈維持200ms秒，換紅燈維持600ms，做循環
- ▶ 所用的語法

case(expression)

Alternative 1: statement 1;
Alternative 2: statement 2;
Alternative 3: statement 3;
default:default_state;

endcase



```
always @ (posedge clk) begin
case(state)
0:begin if (cnt==10) //設定狀態維持的時間
begin cnt<=0; state<=1;end //計數器歸零
//進入下一狀態
else
begin cnt<=cnt+1; state<=state; end
end
1: begin end
2:begin end
default: begin end
endcase
end
```

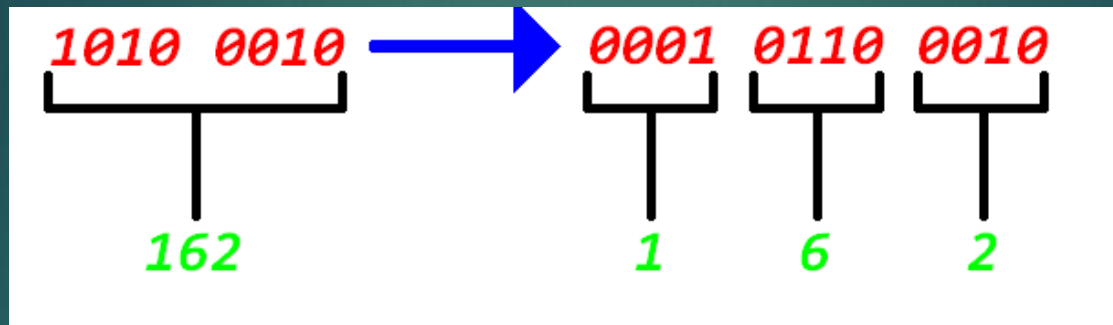
State 的轉換
(Next State logic)

```
always @ (posedge clk) begin
case(state)
0: begin
red<=1; green<=0; yellow<=0;//設定此狀態下要做的事
end
1:
2:
default:
endcase
end
```

在該State下要做的事
(Output logic)

Homework 2-2 :

Binary to Decimal code



- 162 = 8' b10100010; 轉成三個變數 :
 - 百位數=4' b0001;
 - 十位數=4' b0110;
 - 個位數=4' b0010。
- (程式一開始就要宣告這三個暫存器來做運算)

Algorithm:

- ▶ If any column (100's, 10's, 1's, etc.) is 5 or greater, add 3 to that column.
- ▶ Shift all #'s to the left 1 position.
- ▶ If 8 shifts have been performed, it's done! Evaluate each column for the BCD values.
- ▶ Go to step 1.

100's	10's	1's	Binary	Operation
			1010 0010	← 162
		1	010 0010	<< #1
		10	10 0010	<< #2
		101	0 0010	<< #3
		1000		add 3
	1	0000	0010	<< #4
	10	0000	010	<< #5
	100	0000	10	<< #6
	1000	0001	0	<< #7
	1011			add 3
1	0110	0010		<< #8

↑ 1 ↑ 6 ↑ 2

作業說明

▶ HW 2-1

1. 截圖模擬結果並說明

▶ HW 2-2 引入時脈為 50MHz 的clock

1. 模擬輸入binary，改變頻率為50Hz (20ms一個新的binary)
2. 不可使用while loop以及 for loop。
3. 僅用條件判斷if else if else與case進行撰寫
4. 得到解答後，輸出值不變直到下次改變
5. 附上主程式的.v與測試的.v (須有註解)
6. 截圖模擬結果並說明

▶ 繳交期限 6/4 9:59 a.m.

- ▶ 寄到信箱 gain514@g.ncu.edu.tw
- ▶ 郵件主旨 數位邏輯設計_你的名字_HW2