

邏輯系統實驗

Lab 8 Sequential (3)

2022/03/02

第 5 組	
組員姓名	學號
林珮玉	E24084096
廖本恩	E24102179
蘇冠誠	E24084143

實作題(一): Microwave Oven

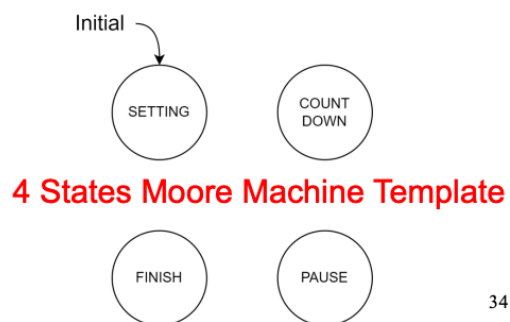
1. 簡述題目：

Create a 10-bit register to store Count Down Time. Only one button work each time. If more than one button been activated, the button with highest priority will be chosen.

Try touse following 4 States Moore Machine Template to create your own Finite State Machine. (By adding new transition to the template)

6 buttons on the control panel

Button Name	Priority
10 Minutes	1
1 Minute	2
10 Seconds	3
1 Second	4
Start	5
Pause	6



34

I/O Interface of Module Microwave in Microwave.v

Name	I/O	Width	Description
clk	I	1	System clock signal. This system is synchronized with the positive edge of the clock.
rst	I	1	Active-high asynchronous reset signal.
tenMinutes	I	1	High when 10 Minutes Button is pressed.
minute	I	1	High when 1 Minute Button is pressed.
tenSeconds	I	1	High when 10 Seconds Button is pressed.
second	I	1	High when 1 Second Button is pressed.
pause	I	1	High when Pause Button is pressed.
start	I	1	High when Start Button is pressed.
countDown	O	10	Show the Count Down Time.
finish	O	1	Show microwave has finished.

2. 實現方式：

```
Microwave.v
module Microwave(clk, rst, tenMinutes, minute, tenSeconds, second,
pause, start, countDown, finish);
    input clk;
    input rst;
    input tenMinutes;
```

```

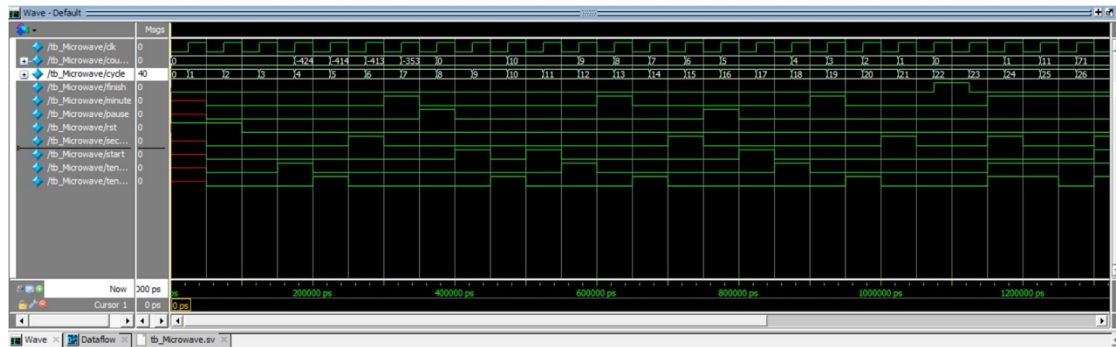
input minute;
input tenSeconds;
input second;
input pause;
input start;
output reg [9:0] countDown;
output reg finish;
reg state;

always @(rst)
begin
    countDown = 0;
    finish = 0;
    state = 0;
end

always @(posedge clk)
begin
    finish = 0;
    if(countDown === 0) state = 0;
    if(state)
    begin
        countDown = countDown - 1;
        if(countDown === 0) finish = 1;
    end
    if(pause)
    begin
        if(!state) countDown = 0;
        else state = 0;
    end
    else if(start) state = 1;
    else if(!state && second) countDown = countDown + 1;
    else if(!state && tenSeconds) countDown = countDown + 10;
    else if(!state && minute) countDown = countDown + 60;
    else if(!state && tenMinutes) countDown = countDown + 600;
end
endmodule

```

3. 結果呈現



實作題(二): Microwave Oven with Door

1. 簡述題目：

Try to reuse Finite State Machine you created in Implementation(1), and add some constraints to transition.

- Setting : The Start Button is valid only when the Door is closed.
- Count Down : Microwave food and count down can only proceed with the Door being closed. If the Door is opened, the microwave oven will be forced to pause.
- Pause : The Start Button is valid only when the Door is closed.

- I/O Interface of Module Microwave in Microwave.v

Name	I/O	Width	Description
clk	I	1	System clock signal. This system is synchronized with the positive edge of the clock.
rst	I	1	Active-high asynchronous reset signal.
tenMinutes	I	1	High when 10 Minutes Button is pressed.
minute	I	1	High when 1 Minute Button is pressed.
tenSeconds	I	1	High when 10 Seconds Button is pressed.
second	I	1	High when 1 Second Button is pressed.
pause	I	1	High when Pause Button is pressed.
start	I	1	High when Start Button is pressed.
door	I	1	High when Door is closed.
countDown	O	10	Show the Count Down Time.
finish	O	1	Show microwave has finished.

2. 實現方式：

MicrowaveDoor.v

```

module MicrowaveDoor(clk, rst, tenMinutes, minute, tenSeconds, second,
pause, start, door, countDown, finish);

    input clk;
    input rst;
    input tenMinutes;
    input minute;
    input tenSeconds;
    input second;
    input pause;
    input start;
    input door;
    output reg [9:0] countDown;
    output reg finish;
    reg state, door_state;

    always @(rst)
    begin
        countDown = 0;
        finish = 0;
        state = 0;
    end

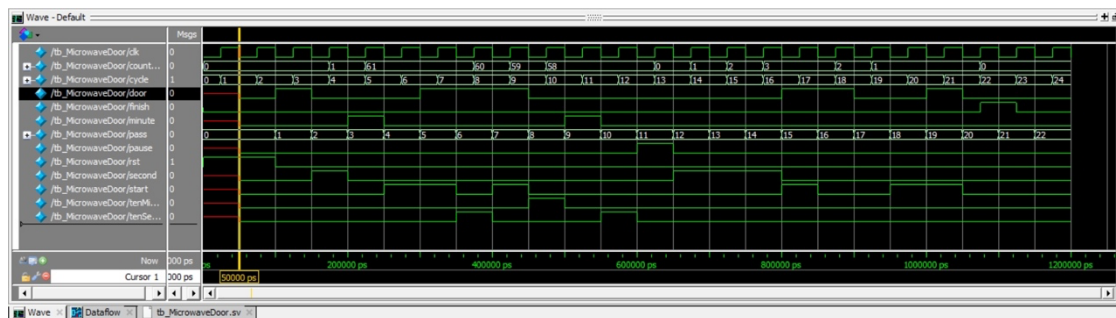
```

```

always @(posedge clk)
begin
    finish = 0;
    if(countDown === 0) state = 0;
    if(state && door_state)
    begin
        countDown = countDown - 1;
        if(countDown === 0) finish = 1;
    end
    if(pause)
    begin
        if(!state || !door_state) countDown = 0;
        else state = 0;
    end
    else if(start) state = 1;
    else if(!state && second) countDown = countDown + 1;
    else if(!state && tenSeconds) countDown = countDown + 10;
    else if(!state && minute) countDown = countDown + 60;
    else if(!state && tenMinutes) countDown = countDown + 600;
    door_state = door;
end
endmodule

```

3. 結果分析：



心得

組員一 林珮玉 E24084096

這次實驗要寫的程式比較複雜，加上距離修邏輯系統有兩年了，就算有事先預習還是不太清楚狀態機的背後的原理與作用。尤其是 Moore Machine 和 Mealy Machine，雖然概念簡單，分辨狀態機種類的方式就是看是不是只跟 output 有關，可是當要實際應用到程式時，我不知從何開始。很謝謝我的組員，當天要考線性代數期中考還是有用心討論，雖然實驗當天沒有做完，但透過晚上努力趕工，也順利完成本週的實驗 Demo。

組員二 廖本恩 E24102179

最近的線代課剛好講到馬可夫鏈跟有限狀態機的概念，老師上課也有提到在電路的應用，沒想到這麼快就有實作的機會。這次的程式碼我們一開始就寫成助教講義的 bad example，把所有的判斷擠在一個式子裡，讓我學習到了 verilog sequential circuits 有效率且正確的寫法是什麼，也更明瞭它作為硬體描述語言與其他程式語言的不同之處。

組員三 蘇冠誠 E24084143

這次實驗算是不簡單的類型，考慮的地方從一開始的加時間和倒數時間逐漸增加，最初甚至還把優先度寫反，之後開始考慮把狀態存下來，再後來因為倒數時不能加時間以及暫停鍵在狀態不同的情況下表現不同所以決定把狀態分開考慮，到後來因為結束訊號只持續一週期讓我重新考慮觸發它的條件等等，可以說是相當複雜。第二題反而在第一題確定後很快就以增加一個寄存器的方式完成了。