

硬體描述語言設計與模擬

Homework #4

Title : Finite_State_Machine

班級：電子二 B

學號：B10713138

姓名：蔡尚哲

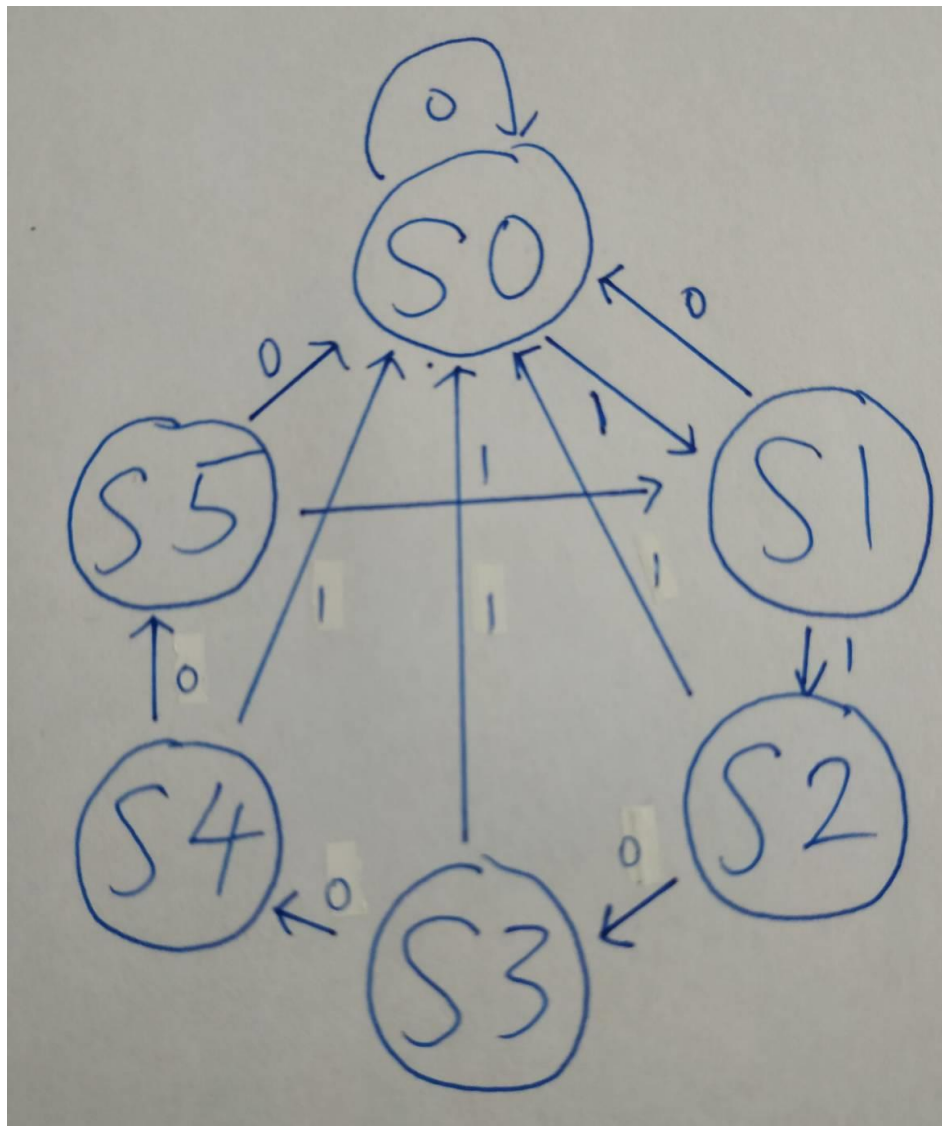
教師：許明華老師

繳交日期：2019/11/10

電路原理

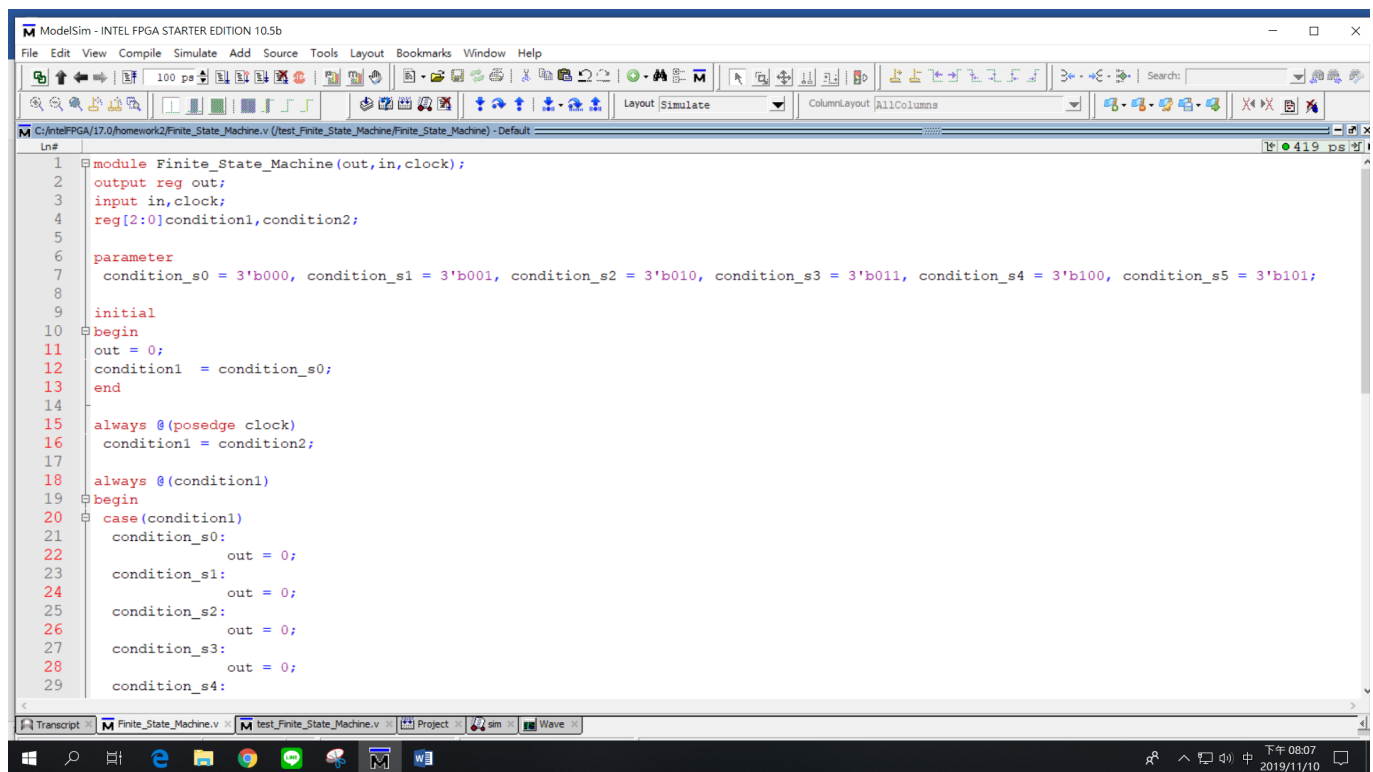
利用有限狀態機作設計，當輸入出現 11000 的連續數字，輸出為 1，其餘狀態輸出皆為 0。

電路架構



S0 是最初始的階段，S 後面的數字代表已經配對出幾個數字，中間只要有一個數字不符合，則跳回 S0，當到 S5 時即輸出 1，S5 時如輸入出現 1 就跳到 S1 為下一輪的判斷。

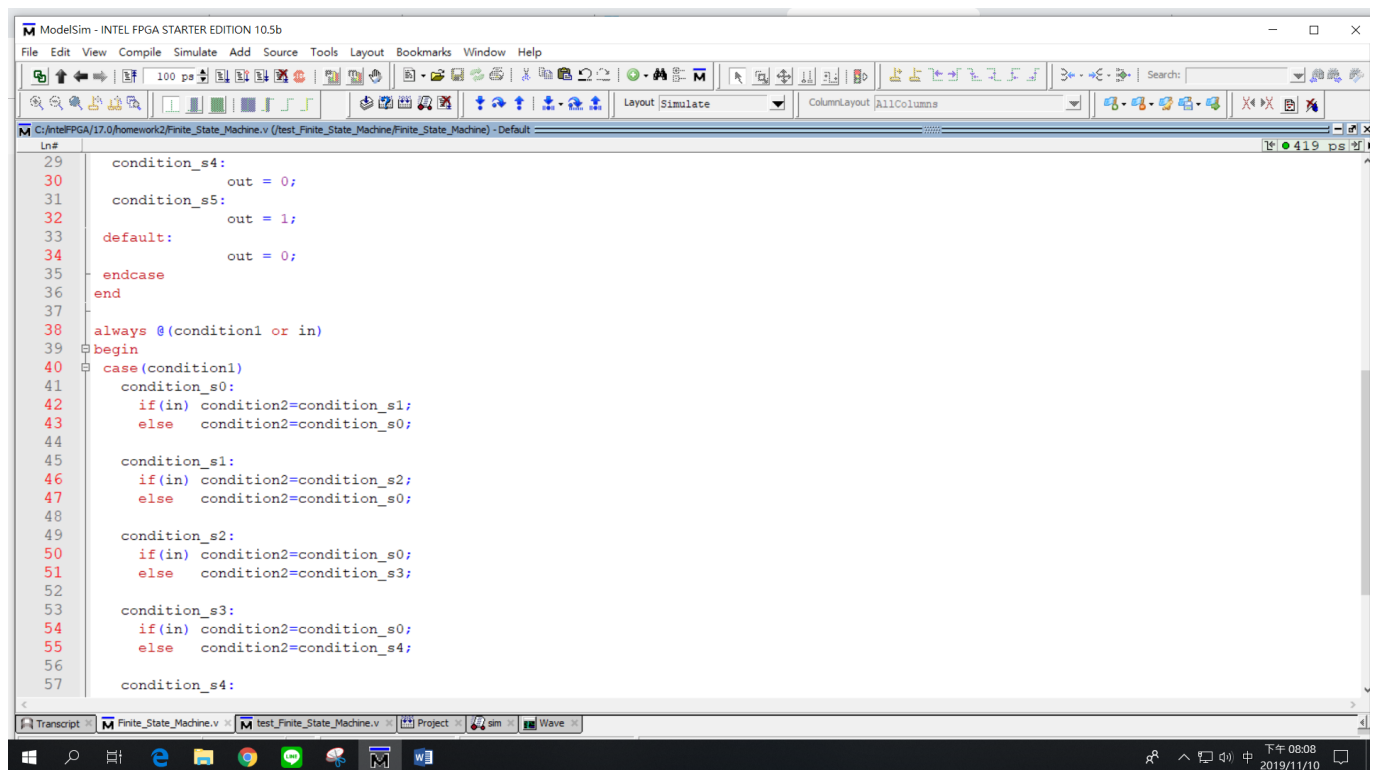
程式擷取畫面



The screenshot shows the ModelSim - INTEL FPGA STARTER EDITION 10.5b interface. The code editor displays the first 29 lines of a Verilog module named `Finite_State_Machine`. The code defines an output `out`, inputs `in` and `clock`, and a 2-bit register `condition1`. It also defines five 3-bit parameters: `condition_s0`, `condition_s1`, `condition_s2`, `condition_s3`, and `condition_s4`. The `initial` block sets `out` to 0 and `condition1` to `condition_s0`. The `always @(posedge clock)` block updates `condition1` to `condition2`. The `always @(condition1)` block contains a `case` statement for `condition1` with five branches: `condition_s0`, `condition_s1`, `condition_s2`, `condition_s3`, and `condition_s4`, each setting `out` to 0.

```
1 module Finite_State_Machine(out,in,clock);
2   output reg out;
3   input in,clock;
4   reg[2:0]condition1,condition2;
5
6   parameter
7     condition_s0 = 3'b000, condition_s1 = 3'b001, condition_s2 = 3'b010, condition_s3 = 3'b011, condition_s4 = 3'b100, condition_s5 = 3'b101;
8
9   initial
10    begin
11      out = 0;
12      condition1 = condition_s0;
13    end
14
15    always @(posedge clock)
16      condition1 = condition2;
17
18    always @(condition1)
19    begin
20      case(condition1)
21        condition_s0:
22          out = 0;
23        condition_s1:
24          out = 0;
25        condition_s2:
26          out = 0;
27        condition_s3:
28          out = 0;
29        condition_s4:
```

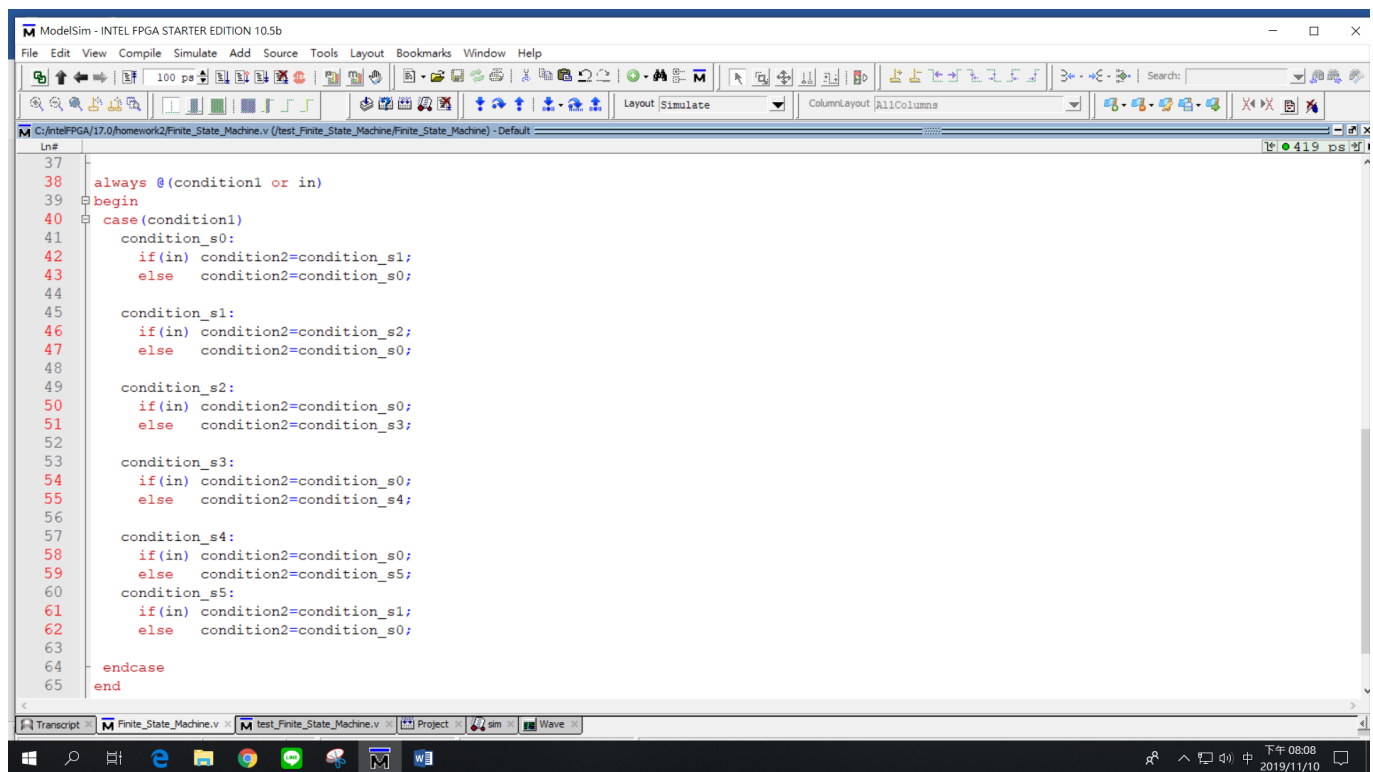
主程式(上)



The screenshot shows the continuation of the Verilog code from the previous image, starting at line 29. It completes the `case` statement for `condition1` with branches for `condition_s4` and `condition_s5`, both setting `out` to 0. The `default` branch also sets `out` to 0. The `endcase` and `end` statements close the `always @(condition1)` block. The `always @(condition1 or in)` block contains a `case` statement for `condition1` with four branches: `condition_s0`, `condition_s1`, `condition_s2`, and `condition_s3`. Each branch contains an `if` statement that updates `condition2` based on the value of `in`. For example, in the `condition_s0` branch, `condition2` is updated to `condition_s1` if `in` is 1, and remains `condition_s0` otherwise.

```
29    condition_s4:
30      out = 0;
31    condition_s5:
32      out = 1;
33    default:
34      out = 0;
35  endcase
36 end
37
38 always @(condition1 or in)
39 begin
40   case(condition1)
41     condition_s0:
42       if(in) condition2=condition_s1;
43       else condition2=condition_s0;
44     condition_s1:
45       if(in) condition2=condition_s2;
46       else condition2=condition_s0;
47     condition_s2:
48       if(in) condition2=condition_s0;
49       else condition2=condition_s3;
50     condition_s3:
51       if(in) condition2=condition_s0;
52       else condition2=condition_s3;
53     condition_s3:
54       if(in) condition2=condition_s0;
55       else condition2=condition_s4;
56   condition_s4:
57
```

主程式(中)

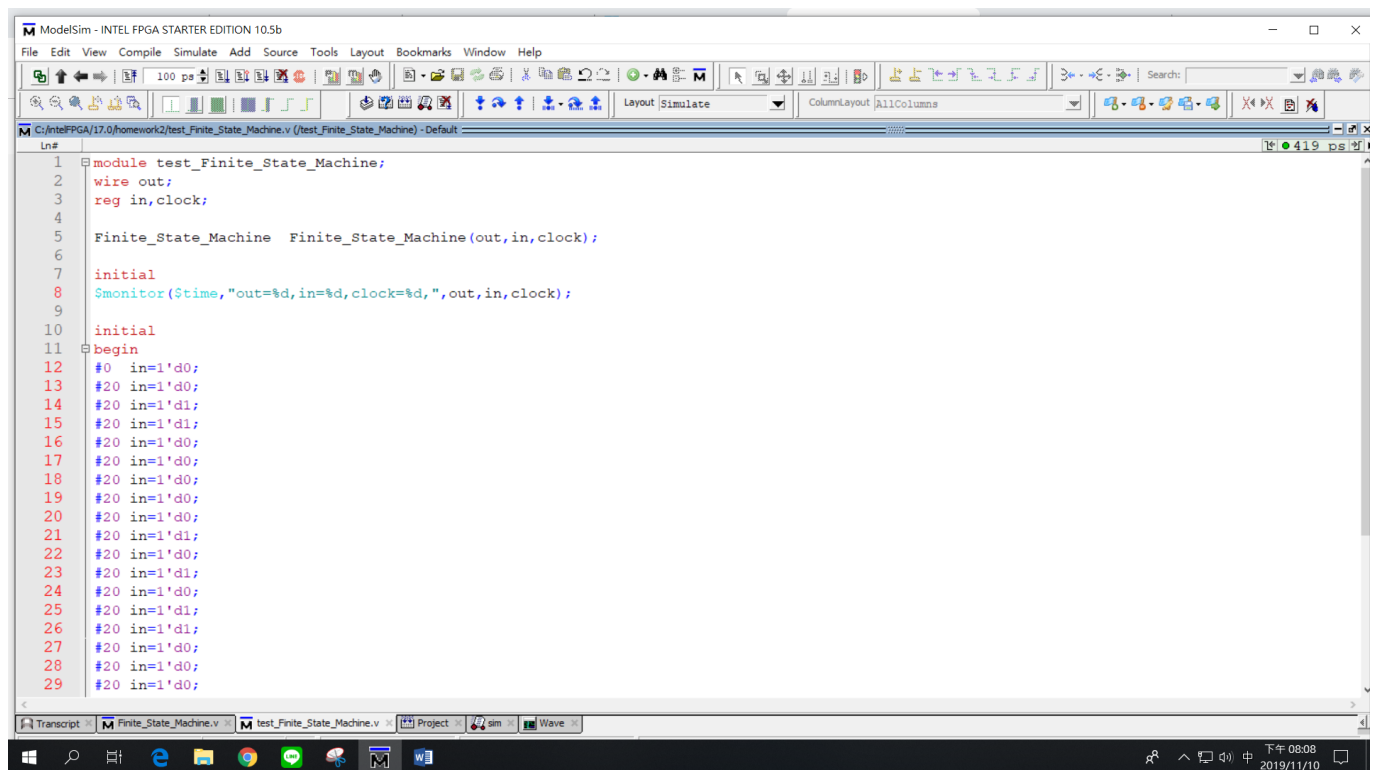


The screenshot shows the ModelSim - INTEL FPGA STARTER EDITION 10.5b interface. The main window displays the Verilog code for a Finite State Machine (FSM). The code is as follows:

```
37  
38 always @(condition1 or in)  
39 begin  
40 case(condition1)  
41 condition_s0:  
42 if(in) condition2=condition_s1;  
43 else condition2=condition_s0;  
44  
45 condition_s1:  
46 if(in) condition2=condition_s2;  
47 else condition2=condition_s0;  
48  
49 condition_s2:  
50 if(in) condition2=condition_s0;  
51 else condition2=condition_s3;  
52  
53 condition_s3:  
54 if(in) condition2=condition_s0;  
55 else condition2=condition_s4;  
56  
57 condition_s4:  
58 if(in) condition2=condition_s0;  
59 else condition2=condition_s5;  
60  
61 condition_s5:  
62 if(in) condition2=condition_s1;  
63 else condition2=condition_s0;  
64  
65 endcase  
66 end
```

The bottom status bar shows the time as 下午 08:08 on 2019/11/10.

主程式(下)

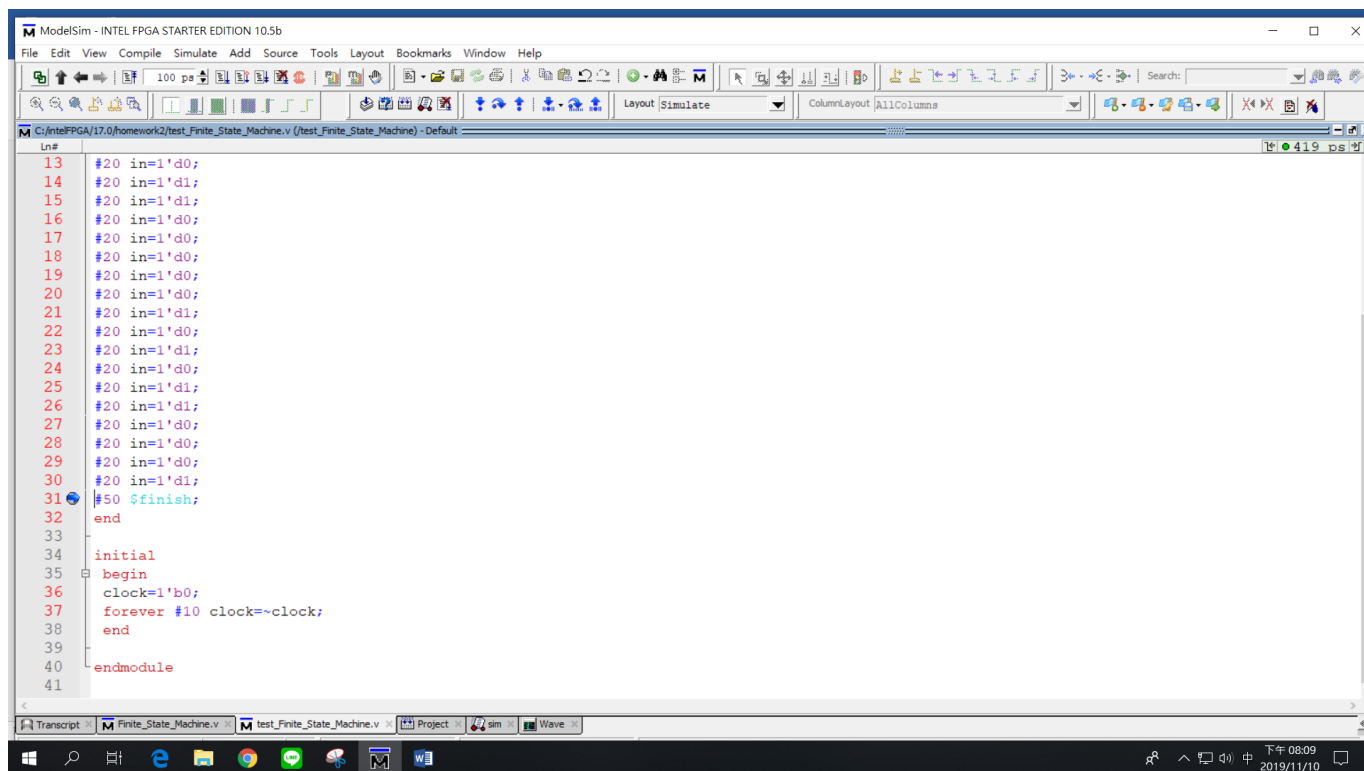


The screenshot shows the ModelSim - INTEL FPGA STARTER EDITION 10.5b interface. The main window displays the Verilog code for a testbench for the Finite State Machine. The code is as follows:

```
1 module test_Finite_State_Machine;  
2 wire out;  
3 reg in,clock;  
4  
5 Finite_State_Machine Finite_State_Machine(out,in,clock);  
6  
7 initial  
8 $monitor($time,"out=%d,in=%d,clock=%d",out,in,clock);  
9  
10 initial  
11 begin  
12 #0 in=1'd0;  
13 #20 in=1'd0;  
14 #20 in=1'd1;  
15 #20 in=1'd1;  
16 #20 in=1'd0;  
17 #20 in=1'd0;  
18 #20 in=1'd0;  
19 #20 in=1'd0;  
20 #20 in=1'd0;  
21 #20 in=1'd1;  
22 #20 in=1'd0;  
23 #20 in=1'd1;  
24 #20 in=1'd0;  
25 #20 in=1'd1;  
26 #20 in=1'd1;  
27 #20 in=1'd0;  
28 #20 in=1'd0;  
29 #20 in=1'd0;
```

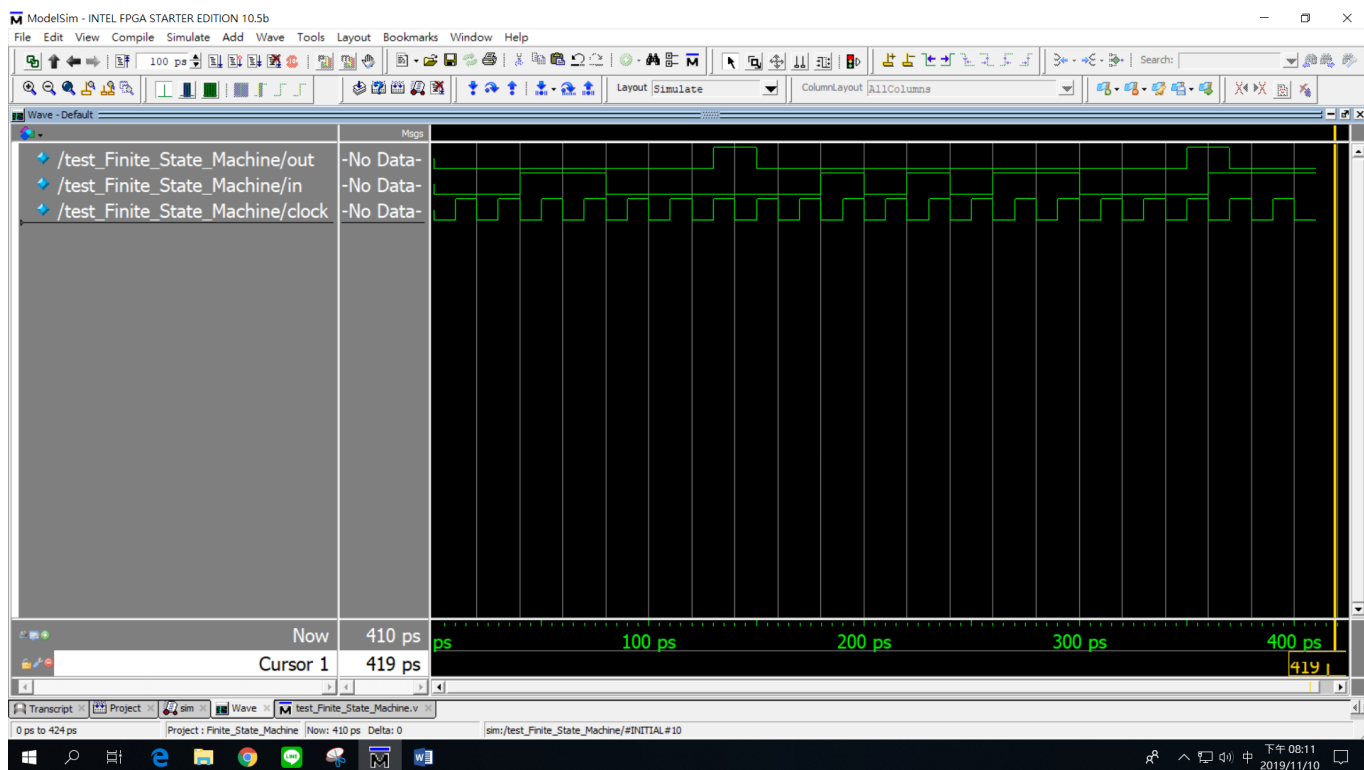
The bottom status bar shows the time as 下午 08:08 on 2019/11/10.

測試檔(上)



測試檔(下)

執行結果



心得(100 字以上)

這一次的作業相對於前幾次稍微容易一些，一部份原因應該是已經有好幾次作業的經驗所以熟能生巧，過程中還是有出現很多非預期的狀況，除錯的時間還是很久，這方面是目前我需要改善的地方，之後或多或少一定還是會需要這個能力，如現在不改善，以後打程式效率會大幅降低。