

# Digital Logic Design Experiments Experiment Report

WIREFRAME FORMATION

電機一甲 110310138 劉千榮



**TAIPEI  
TECH**  
Since 1912

**國立臺北科技大學**  
National Taipei University of Technology

# 數位邏輯設計實習報告

## 第 10 週：正反器

### 組員

110310138 劉千榮

110310147 王瑞鴻

# 壹、基本題

用lpm\_ff實作T F/F，並用波形模擬及/或燒錄至DE-10作驗證。

資料輸入端: T

脈波輸入端: Clk

清除控制端: Clr (高態動作)

輸出端 : Q

# 實驗原理

使用Quartus prime lite編輯邏輯閘電路，燒入DE-10 stander板子，則可確認電路功能。只要做出基本T型正反器即可。在完成實驗後，驗證波型輸出，再將其燒錄製至開發板，執行功能驗證。



# 設計程序

先確認電路功能，畫出真值表並且轉換成布林代數，則可以使用邏輯閘組合成電路。

# 模擬驗證

先確認電路功能，畫出真值表並且轉換成布林代數，則可以使用邏輯閘組合成電路。

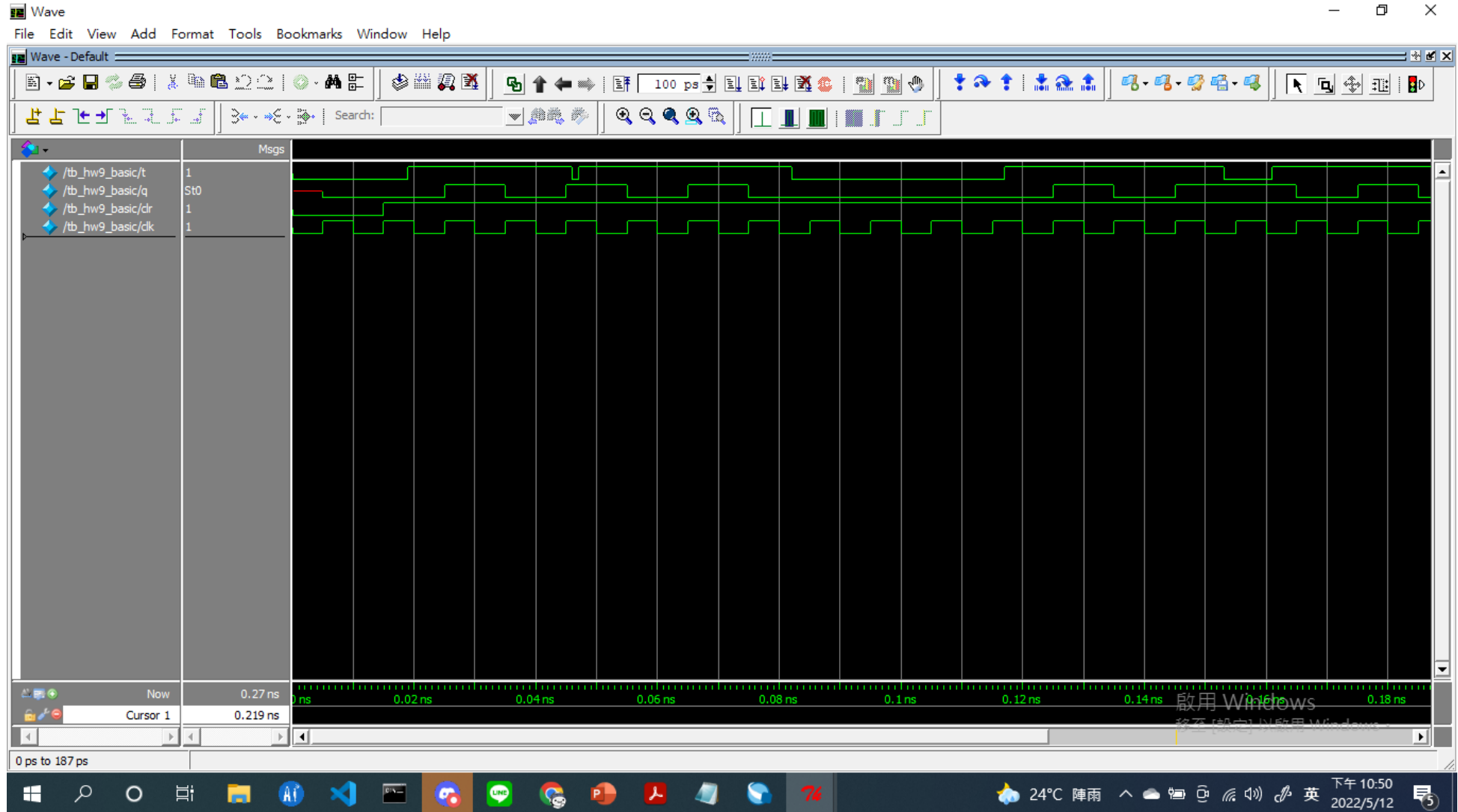
# 設計程序 (Verilog HDL)

```
1  module hw9_basic(t, clk, clr, q);
2
3      // Input Port(s)
4      input clk;
5      input clr;
6      input t;
7
8      // Output Port(s)
9      output reg q;
10
11     always@(posedge clk)
12     begin
13         if(!clr)
14             q <= 0;
15         else
16             if(t)
17                 q <= ~q;
18             else
19                 q <= q;
20     end
21 endmodule
```

# TestBench (Verilog HDL)

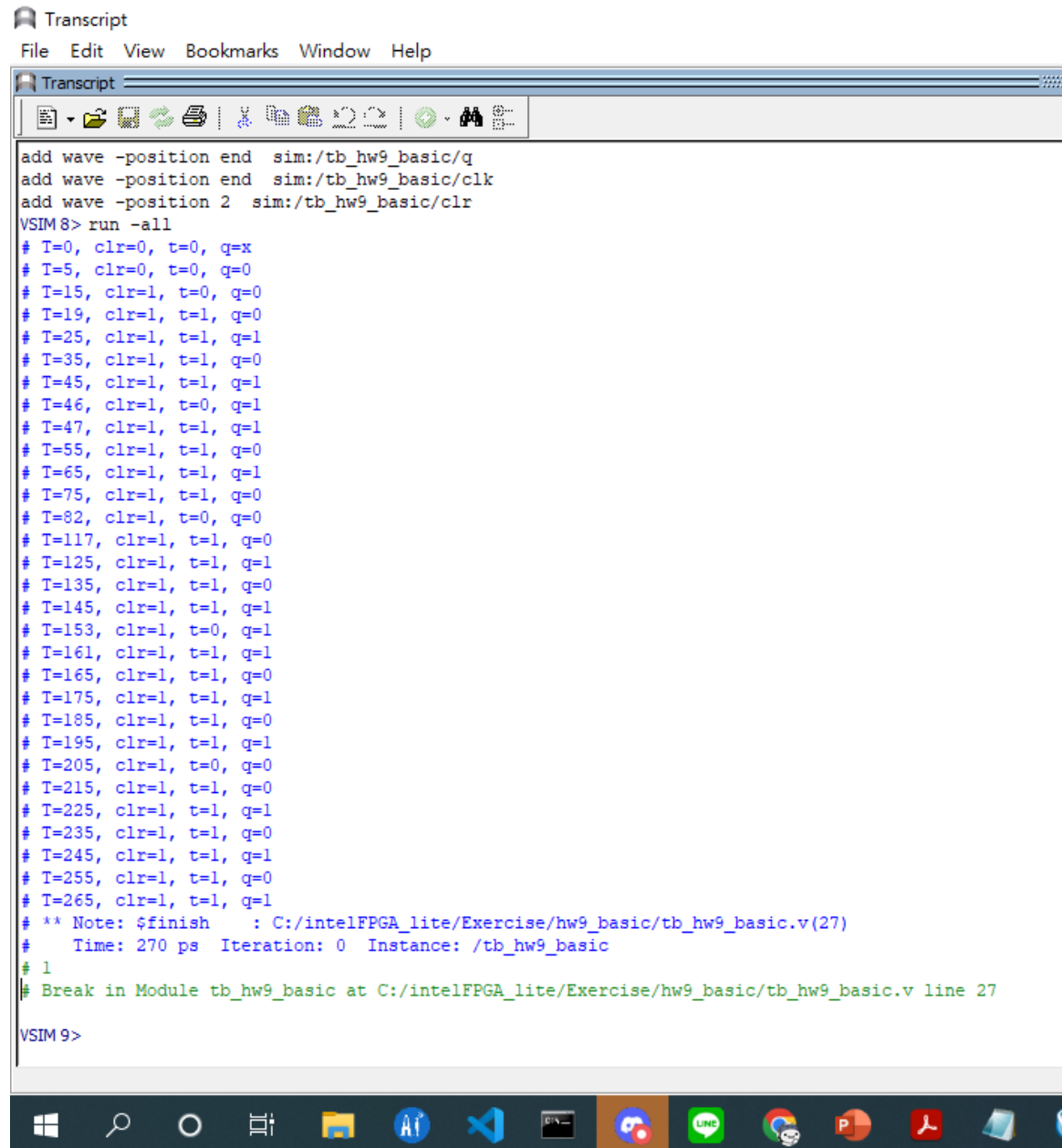
```
1  module tb_hw9_basic;
2
3      wire q;
4      reg clk;
5      reg clr;
6      reg t;
7
8      integer i;
9      reg [4:0] dly;
10
11     hw9_basic tff(t, clk, clr, q);
12
13     always #5 clk = ~clk;
14
15     initial begin
16         {clr, clk, t} <= 0;
17
18         $monitor("T=%0t, clr=%0b, t=%0d, q=%0d", $time, clr, t, q);
19         repeat(2)@(posedge clk);
20         clr <= 1;
21
22         for(i = 0; i < 20 ; i = i + 1) begin
23             dly = $random;
24             #(dly) t <= $random;
25         end
26
27         #20 $finish;
28     end
29 endmodule
```

# 輸出模擬 (ModelSim)





# 輸出模擬 (ModelSim)



The screenshot shows the ModelSim Transcript window. The title bar reads 'Transcript'. The menu bar includes 'File', 'Edit', 'View', 'Bookmarks', 'Window', and 'Help'. The toolbar contains icons for file operations and simulation control. The main text area displays the following content:

```
add wave -position end  sim:/tb_hw9_basic/q
add wave -position end  sim:/tb_hw9_basic/clk
add wave -position 2    sim:/tb_hw9_basic/clr
VSIM 8> run -all
# T=0, clr=0, t=0, q=x
# T=5, clr=0, t=0, q=0
# T=15, clr=1, t=0, q=0
# T=19, clr=1, t=1, q=0
# T=25, clr=1, t=1, q=1
# T=35, clr=1, t=1, q=0
# T=45, clr=1, t=1, q=1
# T=46, clr=1, t=0, q=1
# T=47, clr=1, t=1, q=1
# T=55, clr=1, t=1, q=0
# T=65, clr=1, t=1, q=1
# T=75, clr=1, t=1, q=0
# T=82, clr=1, t=0, q=0
# T=117, clr=1, t=1, q=0
# T=125, clr=1, t=1, q=1
# T=135, clr=1, t=1, q=0
# T=145, clr=1, t=1, q=1
# T=153, clr=1, t=0, q=1
# T=161, clr=1, t=1, q=1
# T=165, clr=1, t=1, q=0
# T=175, clr=1, t=1, q=1
# T=185, clr=1, t=1, q=0
# T=195, clr=1, t=1, q=1
# T=205, clr=1, t=0, q=0
# T=215, clr=1, t=1, q=0
# T=225, clr=1, t=1, q=1
# T=235, clr=1, t=1, q=0
# T=245, clr=1, t=1, q=1
# T=255, clr=1, t=1, q=0
# T=265, clr=1, t=1, q=1
# ** Note: $finish      : C:/intelFPGA_lite/Exercise/hw9_basic/tb_hw9_basic.v(27)
#   Time: 270 ps  Iteration: 0  Instance: /tb_hw9_basic
# 1
# Break in Module tb_hw9_basic at C:/intelFPGA_lite/Exercise/hw9_basic/tb_hw9_basic.v line 27
VSIM 9>
```

The Windows taskbar is visible at the bottom of the window, showing icons for various applications including File Explorer, Visual Studio Code, and the Windows Start menu.

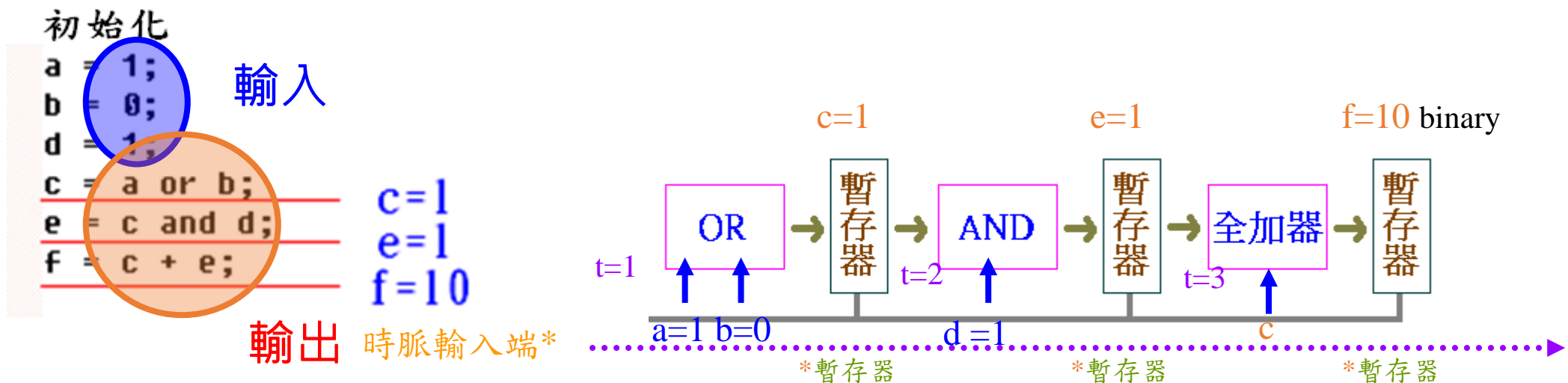
# 問題與心得

**110310138 劉千榮**

終於進入循序電路的部分了，一開始的題目可都是基礎中的基礎，所以在這情況下，學會寫testbench是比會做題目的還要重要。在基本題中，就不用UVM檔測試，而是再寫一份testbench後，直接送進ModelSim測，雖然好像沒有差多少，卻是重要的技能之一。

# 壹、進階題

如下圖所示，這是一個C語言的序向程式，紅線和藍字部分則是單步執行的結果，請設計出下列的程式，並可以單步執行



# 實驗原理

使用Quartus prime lite編輯邏輯閘電路，燒入DE-10 stander板子，則可確認電路功能。在完成實驗後，驗證波型輸出，再將其燒錄製至開發板，執行功能驗證。



# 設計程序

先確認電路功能，畫出真值表並且轉換成布林代數，則可以使用邏輯閘組合成電路。

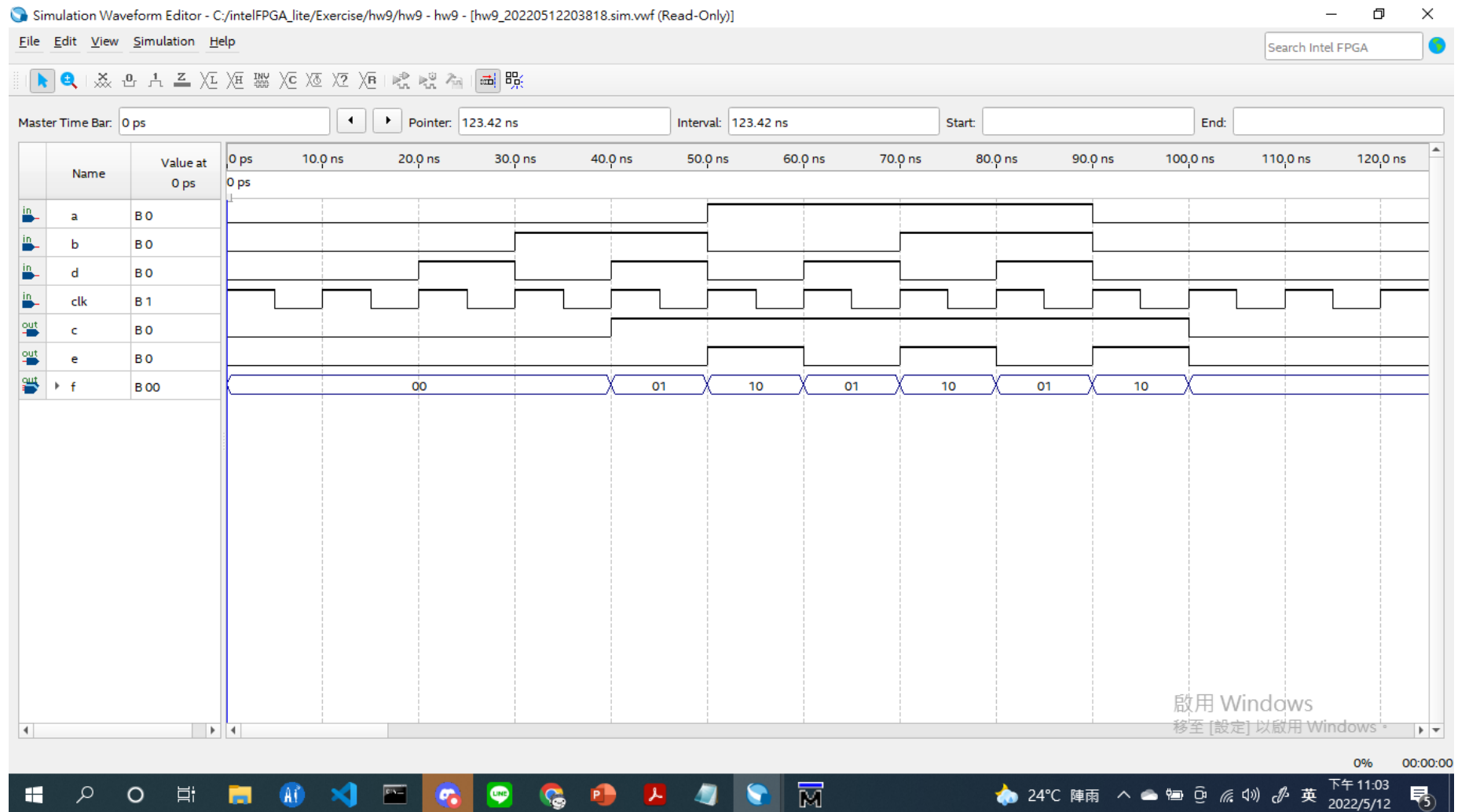
# 模擬驗證

先確認電路功能，畫出真值表並且轉換成布林代數，則可以使用邏輯閘組合成電路。

# 設計程序 (Verilog HDL)

```
1  module hw9(a, b, c, d, e, f[1:0], clk);
2
3      input clk;
4      input a, b, d;
5      output reg c, e;
6      output reg [1:0] f;
7
8      always@(posedge clk)
9      begin
10         c = a | b;
11         e = c & d;
12         f = c + e;
13     end
14
15 endmodule
```

# 模擬 Simulation



# 問題與心得

**110310138 劉千榮**

Verilog其實和SystemVerilog一樣已經非常接近C語言，所以我們只要考慮需要逐步執行的部分，再丟入Verilog中的always宣告中，就可以實現題目所述的逐行讀取。在Verilog程式中，先丟暫存器會使程式產生不必要的延遲，所以不會這樣設計。



WIREFRAME FORMATION

工業推手一世紀 · 企業搖籃一百年

*100 Years of Excellence · Cultivating Entrepreneurs of Tomorrow*



國立臺北科技大學  
National Taipei University of Technology