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 11 週 : 順序邏輯—狀態機

組員

110310138 劉千榮

110310147 王瑞鴻

TAIPEI TECH Since 1912 國立臺北科技大學 National Taipei University of Technology

壹、基本題

請以以上狀態機之範例，將以下之真值表中Tin於S1之值顛倒改為0,1設計出你的電路，並模擬出其輸出Qout波形結果。

上次狀態	控制線			輸出狀態	輸出線
	clk	reset	Tin		Qout
X	X	1	X	S0	0
S0	↑	0	1	S1	1
S0	↑	0	0	S1	1
S1	↑	0	0	S0	0
S1	↑	0	1	S1	1

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。依照Verilog製作有限狀態機的步驟，分部分的完成。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) – 1/4

```
1   module hw10(tin, clk, rst_n, q);
2
3       localparam s0   = 1'b0;
4       localparam s1   = 1'b1;
5
6       // Input Port(s)
7       input tin;
8       input clk;
9       input rst_n;
10
11      // Output Port(s)
12      output reg q;
13
14      reg cur_state, next_state;
```

模組所需用到腳位以及狀態的宣告

設計程序 (Verilog HDL) – 2/4

三段式狀態機 第一段

```
16    always @(posedge clk or posedge rst_n) begin
17        if(rst_n)
18            cur_state <= s0;
19        else
20            cur_state <= next_state;
21    end
```

第一段always描述現態(cur_state)

設計程序 (Verilog HDL) – 3/4

三段式狀態機 第二段

```
23    always @(cur_state or tin) begin
24
25       case(cur_state)
26
27          s0: begin
28             next_state = s1;
29          end
30
31          s1: begin
32             if(tin) next_state = s1;
33             else next_state = s0;
34          end
35       endcase
36    end
```

第二段always以組合邏輯描述次態(next_state)

設計程序 (Verilog HDL) – 4/4

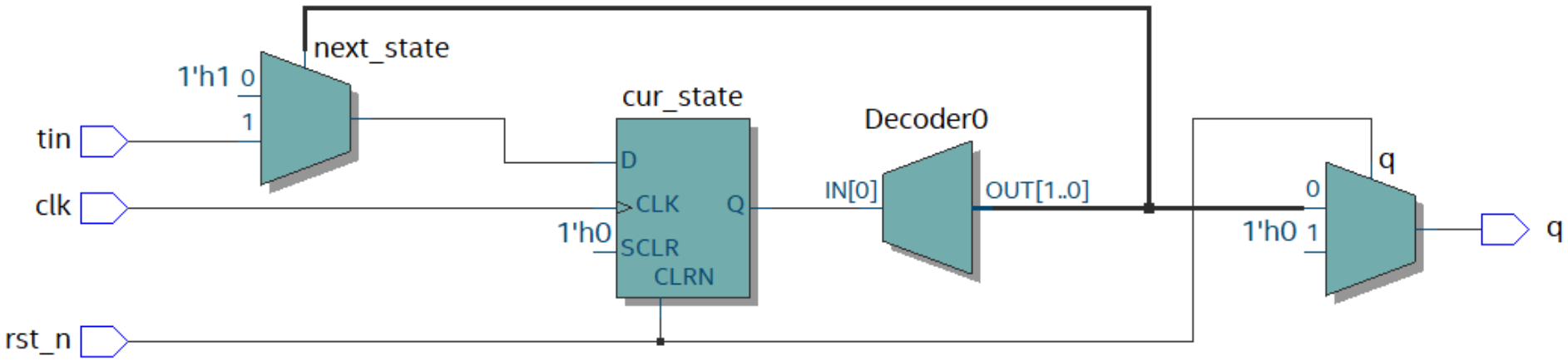
三段式狀態機 第三段

```
38      always@(cur_state)begin
39
40          if(rst_n)
41              q <= 1'b0;
42          else begin
43              case(cur_state)
44                  s0:     q <= 1'b0;
45                  s1:     q <= 1'b1;
46              endcase
47          end
48      end
49  endmodule
```

第三段always以描述各時序狀態下的輸出

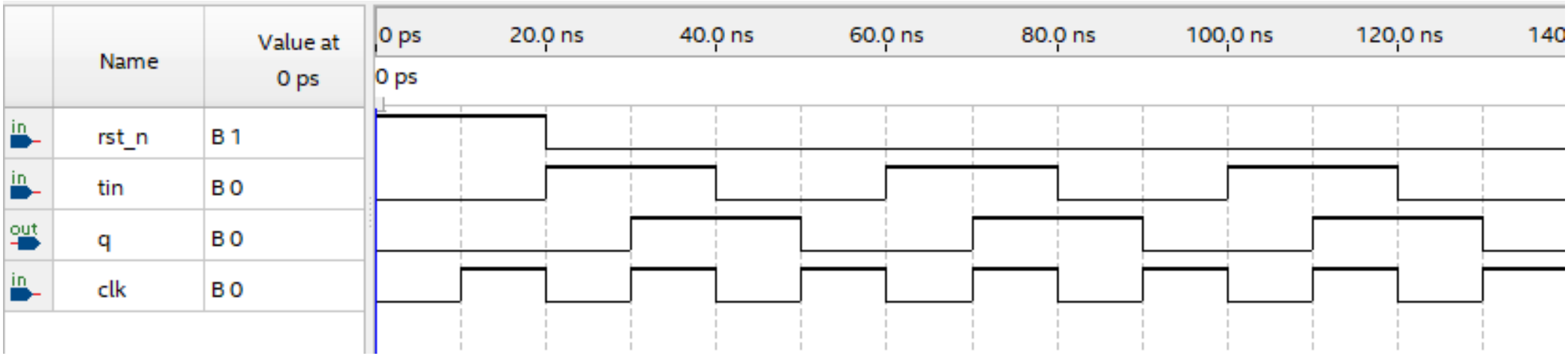
TAIPEI TECH

RTL Module



從以上的模型中，不難看出這是個Mealy Machine。
因為下一次的輸出明顯受到當前狀態及輸出的影響。

輸出模擬 (VWF)



從0~20ns時，因為Reset信號為1，故輸出狀態被拉至S0且輸出為0。當Reset信號為0後，即按照題目所需在每個clk的正緣改變信號，且遵照題目之指示產生相對應的輸出。

TAIPEI TECH

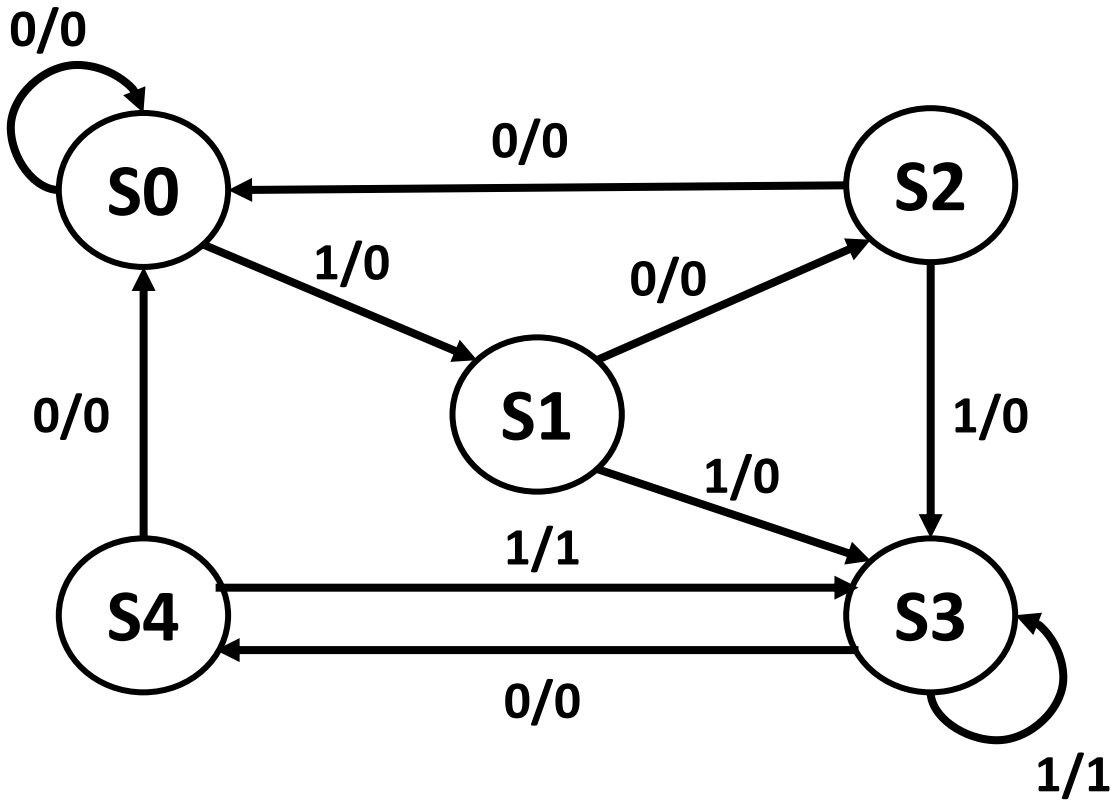
問題與心得

110310138 劉千榮

終於來到循序邏輯的部分，期待很久了。Verilog是可以直接描述電路邏輯及功能，使編譯器產生電路，所以就不太需要考慮使用什麼正反器組成，而是使用當今最常使用的三段或兩段式狀態機的寫法來實作本題。該題的測試信號就直接拉VWF來展示。

壹、進階題

用D型正反器實作所示之狀態圖，報告中要顯示設計過程。



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。依照Verilog製作有限狀態機的步驟，分部分的完成。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) – 1/6

```
1   module hw10_extra(clk, rst, In_Data, Out_Data);
2
3       // Input Port(s)
4       input clk, rst;
5       input In_Data;
6
7       // Output Port(s)
8       output reg Out_Data;
9
10      reg [3:0] cur_state, next_state;
11
12      // Parameter Declaration(s)
13      localparam s0 = 3'd0;
14      localparam s1 = 3'd1;
15      localparam s2 = 3'd2;
16      localparam s3 = 3'd3;
17      localparam s4 = 3'd4;
```

模組所需用到腳位以及狀態的宣告

設計程序 (Verilog HDL) – 2/6

三段式狀態機 第一段

```
19      // 設定rst為低態觸發
20      always @(posedge clk or negedge rst) begin
21
22          // rst輸入為0
23          if(!rst)
24              // 將當前狀態設定為s0
25              cur_state <= s0;
26          else
27              // 將當前狀態設定為讀入的次態
28              cur_state <= next_state;
29      end
```

第一段always描述現態(cur_state)

TAIPEI TECH

設計程序 (Verilog HDL) – 3/6

三段式狀態機 第二段

```
31    always @(*) begin
32        // 讀取當前狀態
33        case(cur_state)
34
35            // 當前狀態為s0
36            s0: begin
37                if(In_Data) next_state = s1;
38                else next_state = s0;
39            end
40
41            // 當前狀態為s1
42            s1: begin
43                if(In_Data) next_state = s3;
44                else next_state = s2;
45            end
46
47            // 當前狀態為s2
48            s2: begin
49                if(In_Data) next_state = s3;
50                else next_state = s0;
51            end
```

設計程序 (Verilog HDL) – 4/6

```
53          // 當前狀態為s3
54          s3: begin
55              if(In_Data) next_state = s3;
56              else next_state = s4;
57          end
58
59          // 當前狀態為s4
60          s4: begin
61              if(In_Data) next_state = s3;
62              else next_state = s0;
63          end
64      endcase
65  end
```

第二段always以組合邏輯描述次態 (next_state)

設計程序 (Verilog HDL) – 5/6

三段式狀態機 第三段

```
67    always@(cur_state) begin
68
69        // 若reset為0
70        if(!rst)
71            Out_Data <= 1'b0;
72        else begin
73            case(cur_state)
74
75                // 當前狀態為s0
76                s0: begin
77                    Out_Data <= 1'b0;
78                end
79
80                // 當前狀態為s1
81                s1: begin
82                    Out_Data <= 1'b0;
83                end
```

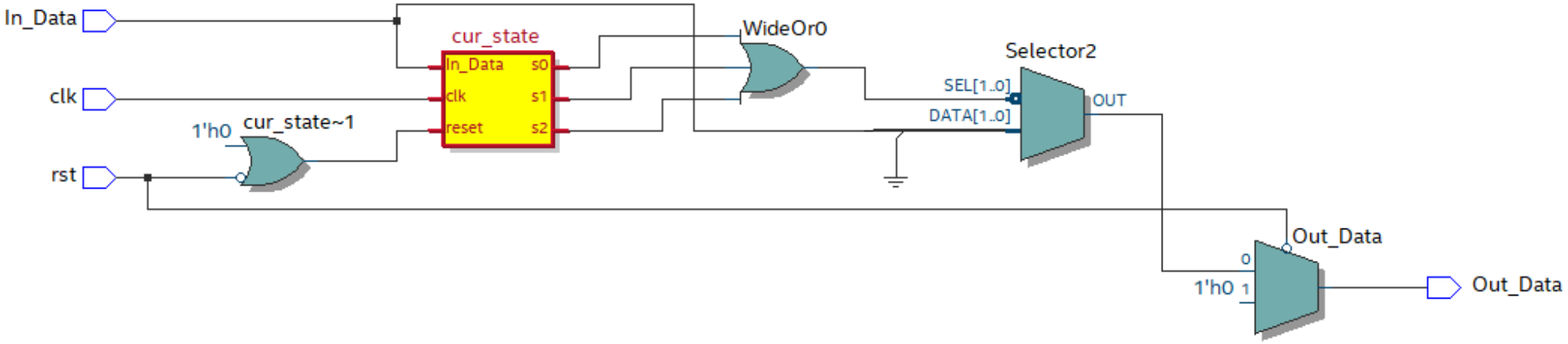
設計程序 (Verilog HDL) – 6/6

```
85              // 當前狀態為s2
86              s2: begin
87                 Out_Data <= 1'b0;
88              end
89
90              // 當前狀態為s3
91              s3: begin
92                 if(In_Data) Out_Data <= 1'b1;
93                 else Out_Data <= 1'b0;
94              end
95
96              // 當前狀態為s4
97              s4: begin
98                 if(In_Data) Out_Data <= 1'b1;
99                 else Out_Data <= 1'b0;
100             end
101
102         endcase
103     end
104 end
```

第三段always以描述各時序狀態下的輸出

TAIPEI TECH

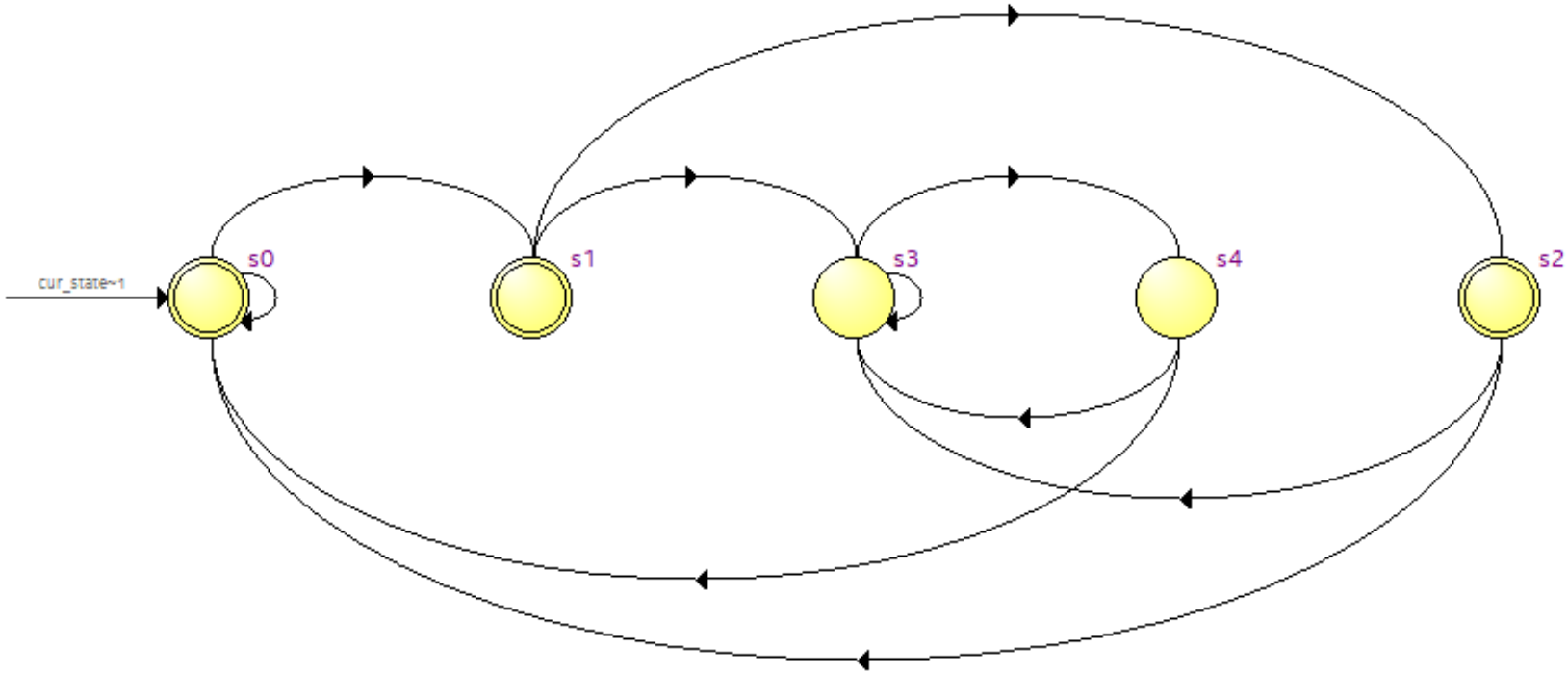
RTL Module



從以上的模型中，不難看出這是個Mealy Machine。
因為下一次的輸出明顯受到當前狀態及輸出的影響。
我們還可以把其中的狀態重新用Quartus畫一次。

TAIPEI TECH

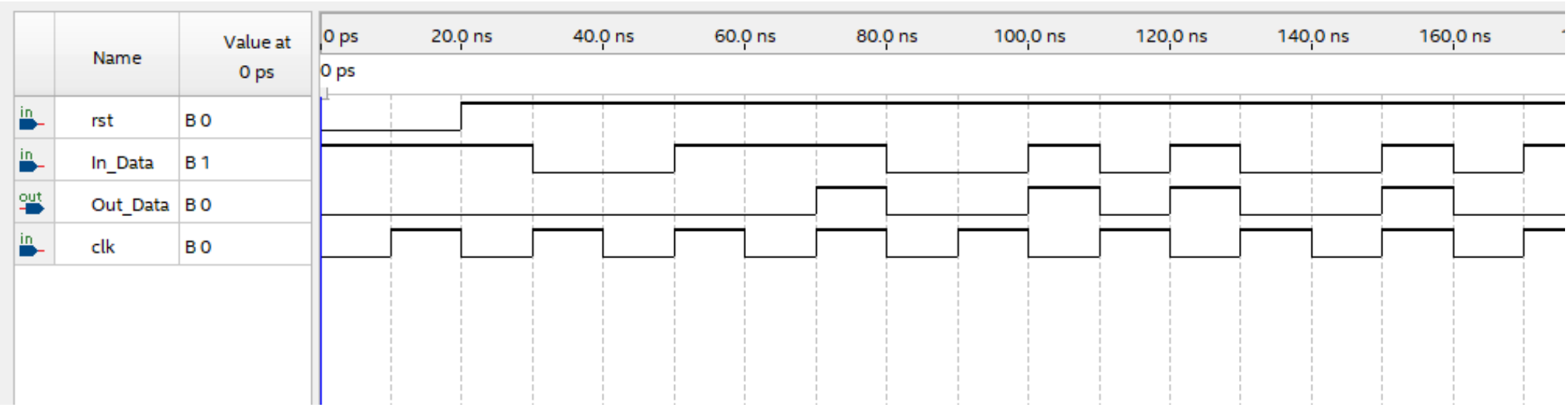
State Machine Viewer



雖然上方沒有如題目標示得那麼清楚，不過詳細的資訊仍然可以從該視窗去查看。

TAIPEI TECH

模擬 Simulation



從0~20ns時，因為Reset信號為0，故輸出狀態被拉至S0且輸出為0。當Reset信號為1後，即按照題目所需在每個clk的正緣改變信號，且遵照題目之指示產生相對應的輸出。

問題與心得

110310138 劉千榮

這題看上去很是複雜，不過如果使用三段式狀態機的寫法，還是能很清楚的在每段always中，完成題目所需的功能。我認為最複雜應該是撰寫testbench，其實我也跟同學一樣第一次接觸Verilog，對循序機的testbench還沒有夠成熟的撰寫經驗，希望日後的測試都可以使用ModelSim來完成。

WIREFRAME FORMATION

工業推手一世紀．企業搖籃一百年

100 Years of Excellence · Cultivating Entrepreneurs of Tomorrow

TAIPEI TECH
Since 1912

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