

2024/XX/XX

實驗六

姓名：黃文祺 學號：01057013

班級：資工 3A

E-mail：OOOOOOOOO

注意

1. 繳交時一律轉 PDF 檔
2. 繳交期限為
隔週三上午九點
3. 一人繳交一份
4. 檔名：學號_HW?.pdf
檔名請按照作業檔名格式進行填寫
未依照格式不予批改

一、RS-232 Transmitter(一)

■ 實驗說明：

1. 寫出 RS-232 Transmitter，並顯示其模擬圖。

2. 輸入:

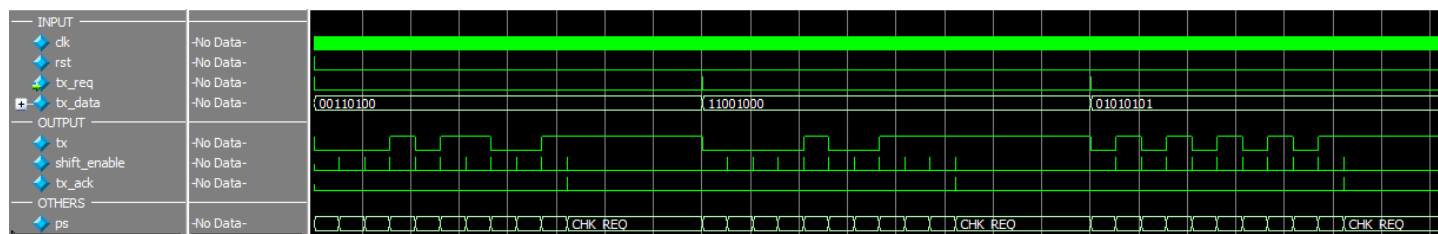
- clk : 50MHz
- rst : 系統重製訊號。
- tx_req : 發送請求。
- tx_data [7:0] : 要發送的資料。

3. 輸出:

- tx : 傳送訊號。
- shift_enable : 表示完整傳送完一個 bit。
- tx_ack: 表示完整傳送完一筆資料。

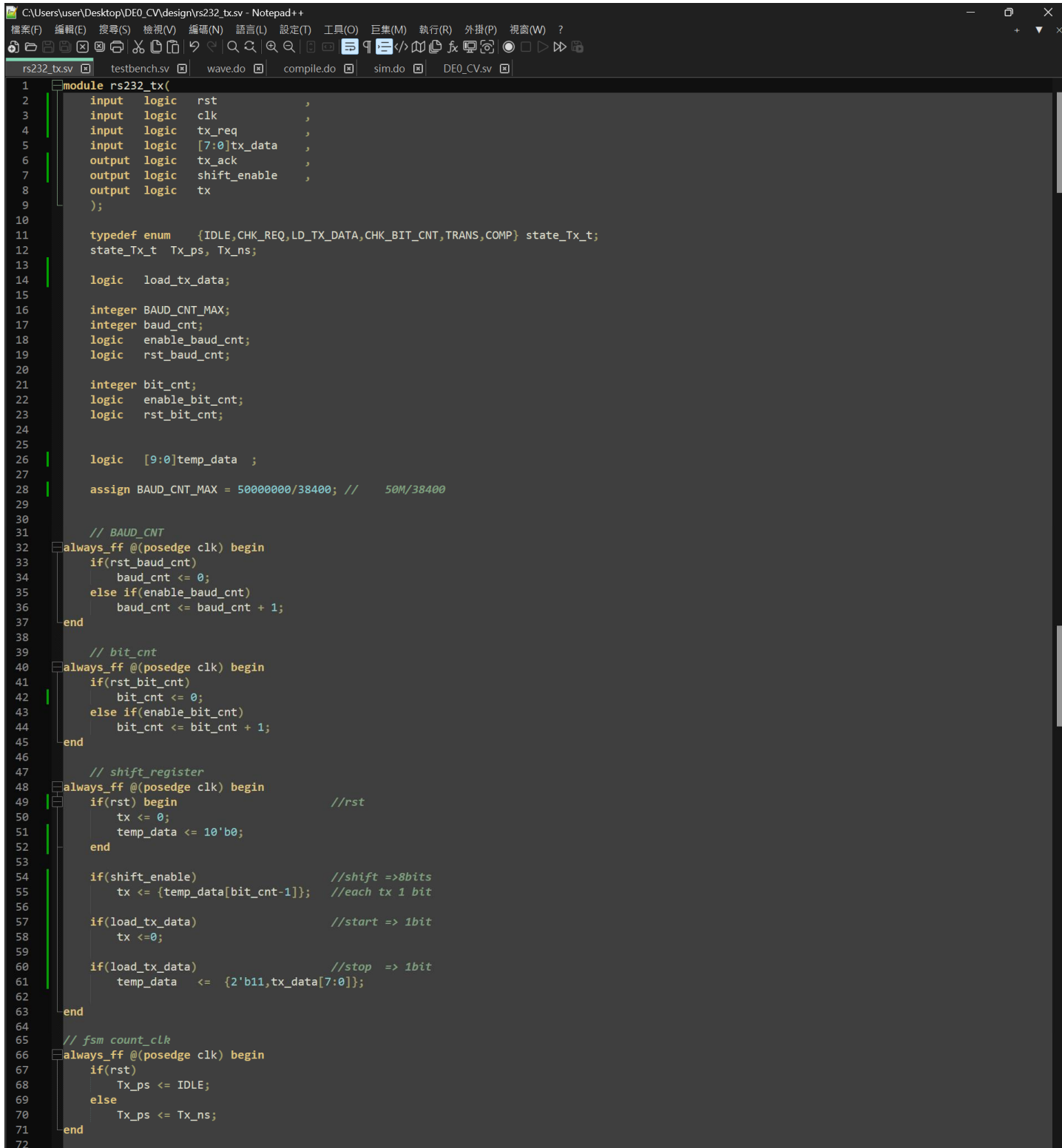
4. 測資:

- 8'b00110100
- 8'b11001000
- 8'b01010101



■ 系統架構程式碼與程式碼說明

截圖請善用 win+shift+S



```
C:\Users\user\Desktop\DE0_CV\design\rs232_tx.sv - Notepad++
檔案(F) 編輯(E) 搜尋(S) 檢視(V) 編碼(N) 語言(L) 設定(T) 工具(O) 巨集(M) 執行(R) 外掛(P) 視窗(W) ?
rs232_tx.sv testbench.sv wave.do compile.do sim.do DE0_CV.sv

1 module rs232_tx(
2     input  logic  rst           ,
3     input  logic  clk          ,
4     input  logic  tx_req       ,
5     input  logic  [7:0]tx_data ,
6     output logic  tx_ack       ,
7     output logic  shift_enable ,
8     output logic  tx
9 );
10
11     typedef enum    {IDLE,CHK_REQ,LD_TX_DATA,CHK_BIT_CNT,TRANS,COMP} state_Tx_t;
12     state_Tx_t  Tx_ps, Tx_ns;
13
14     logic  load_tx_data;
15
16     integer BAUD_CNT_MAX;
17     integer baud_cnt;
18     logic  enable_baud_cnt;
19     logic  rst_baud_cnt;
20
21     integer bit_cnt;
22     logic  enable_bit_cnt;
23     logic  rst_bit_cnt;
24
25
26     logic  [9:0]temp_data ;
27
28     assign BAUD_CNT_MAX = 50000000/38400; // 50M/38400
29
30
31     // BAUD_CNT
32     always_ff @(posedge clk) begin
33         if(rst_baud_cnt)
34             baud_cnt <= 0;
35         else if(enable_baud_cnt)
36             baud_cnt <= baud_cnt + 1;
37     end
38
39     // bit_cnt
40     always_ff @(posedge clk) begin
41         if(rst_bit_cnt)
42             bit_cnt <= 0;
43         else if(enable_bit_cnt)
44             bit_cnt <= bit_cnt + 1;
45     end
46
47     // shift_register
48     always_ff @(posedge clk) begin
49         if(rst) begin //rst
50             tx <= 0;
51             temp_data <= 10'b0;
52         end
53
54         if(shift_enable) //shift => 8bits
55             tx <= {temp_data[bit_cnt-1]}; //each tx 1 bit
56
57         if(load_tx_data) //start => 1bit
58             tx <= 0;
59
60         if(load_tx_data) //stop => 1bit
61             temp_data <= {2'b11,tx_data[7:0]};
62
63     end
64
65     // fsm count_clk
66     always_ff @(posedge clk) begin
67         if(rst)
68             Tx_ps <= IDLE;
69         else
70             Tx_ps <= Tx_ns;
71     end
72 end
```

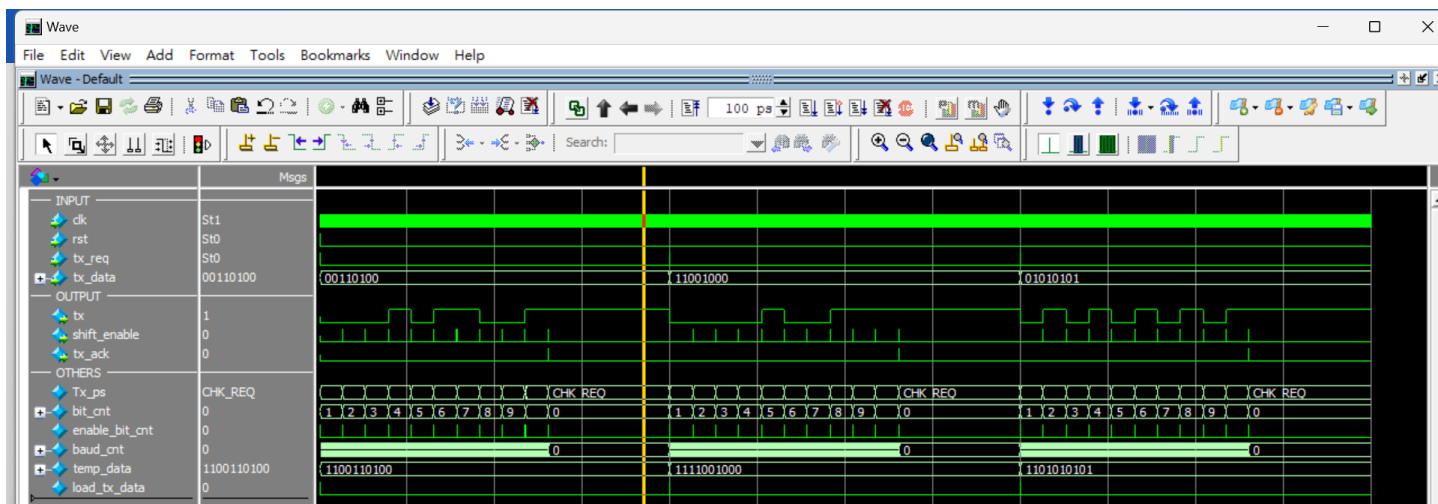
```

73 always_comb begin
74     rst_baud_cnt = 0;
75     enable_baud_cnt = 0;
76
77     shift_enable = 0;
78     tx_ack = 0;
79     load_tx_data = 0;
80
81     enable_bit_cnt = 0;
82     rst_bit_cnt = 0;
83
84     Tx_ns = Tx_ps;
85
86 case(Tx_ps)
87
88     IDLE: begin
89         rst_bit_cnt=1;
90         rst_baud_cnt=1;
91         Tx_ns = CHK_REQ;
92     end
93
94     CHK_REQ: begin
95
96         if(tx_req)
97             Tx_ns = LD_TX_DATA;
98         else
99             Tx_ns = CHK_REQ;
100     end
101
102     LD_TX_DATA: begin
103         load_tx_data=1;
104         Tx_ns=CHK_BIT_CNT;
105     end
106
107     CHK_BIT_CNT: begin
108         enable_bit_cnt=1;
109         if(bit_cnt>=10) begin
110             rst_bit_cnt=1;
111             tx_ack=1;
112             Tx_ns=COMP;
113         end
114         else
115             Tx_ns=TRANS;
116         end
117     end
118
119     TRANS: begin
120         enable_baud_cnt=1;
121         if(baud_cnt>=BAUD_CNT_MAX) begin
122             shift_enable=1;
123             rst_baud_cnt=1;
124             Tx_ns=CHK_BIT_CNT;
125         end
126         else
127             Tx_ns=TRANS;
128         end
129     end
130
131     COMP: begin
132         if(tx_req==0)
133             Tx_ns=CHK_REQ;
134         else
135             Tx_ns=COMP;
136         end
137     end
138 endcase
139 end
140 endmodule

```

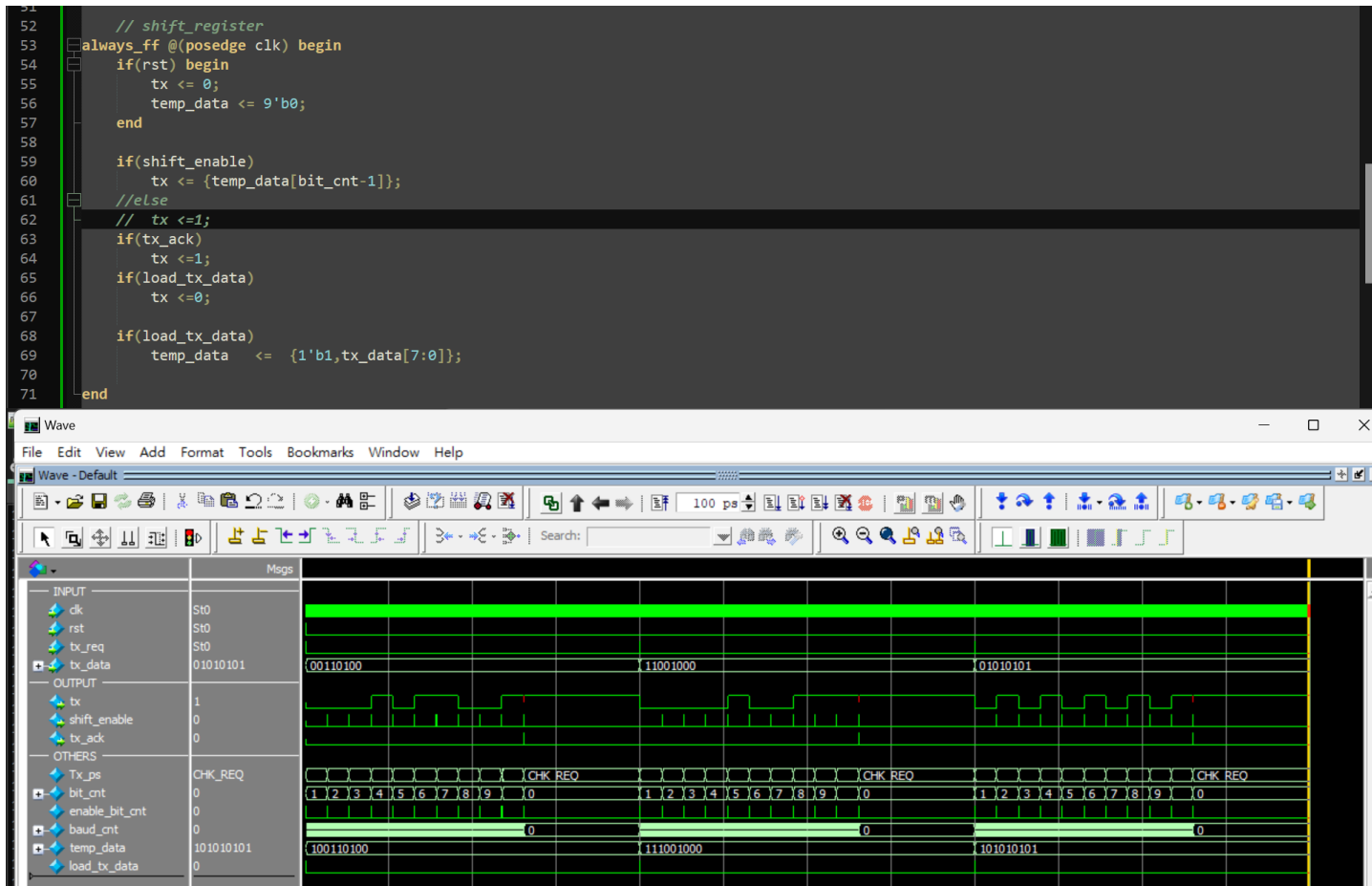
■ 模擬結果與結果說明：

(說明參考心得)



■ 結論與心得：

這是原本的 shift_register，而波型圖的 tx_ack 為 1 時對應的 tx 出現了 X 的紅線



因此我就在想可能是他跟 enable 有重複給值的問題所以後來把 shift_register 改成了下圖

```
47 // shift_register
48 always_ff @(posedge clk) begin
49     if(rst) begin //rst
50         tx <= 0;
51         temp_data <= 10'b0;
52     end
53
54     if(shift_enable) //shift =>8bits
55         tx <= {temp_data[bit_cnt-1]}; //each tx 1 bit
56
57     if(load_tx_data) //start => 1bit
58         tx <=0;
59
60     if(load_tx_data) //stop => 1bit
61         temp_data <= {2'b11,tx_data[7:0]};
62
63 end
```

這樣不只程式碼變得比較好懂，也比較符合題目中電路圖的接線

owob