

Lab14

實驗主題: Registers & Counters

實驗日期:2022/12/05

學號姓名:B103040009 尹信淳

實驗內容: Design and Verify the following circuits using Verilog HDL

1. Four-bit universal shift register
2. BCD synchronous counter

Exercise1: Four-bit universal shift register

Design and verify the four-bit universal shift register using Verilog HDL structural (gate-level) modeling

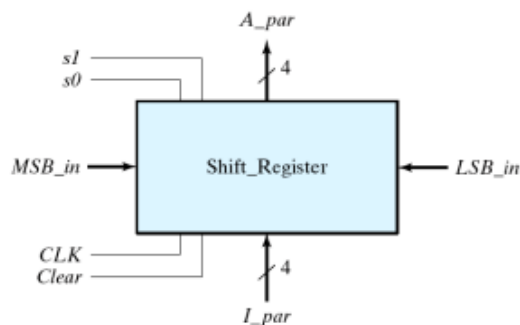


Fig. 6.7

(a)

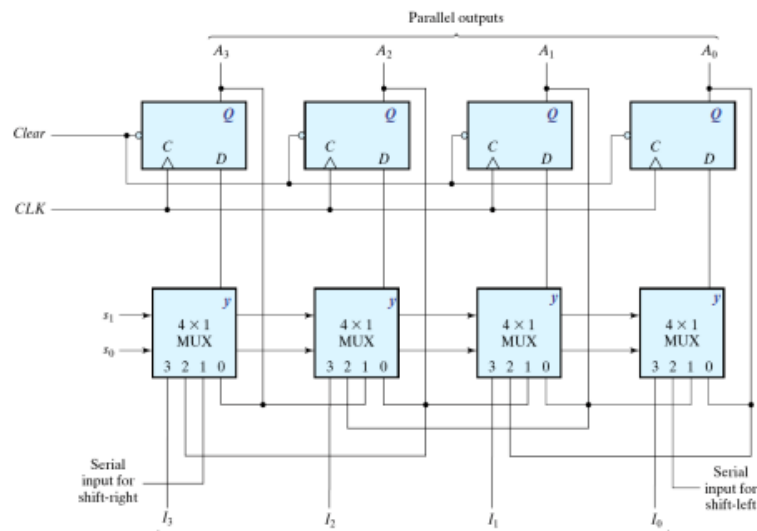


Fig. 6.7 Four-bit universal shift register
(continued)

Parallel inputs
(b)

Exercise1 實驗過程描述及推導:

這題我把整個 shift register 分成四塊組合而成，每塊都是由一個 D Flip-flop 跟一個 4x1 multiplexer 組成。D Flip-flop 跟 4x1 multiplexer 的 code 以前練習過，所以只是複習。照著老師給的圖就能照著接成其中一塊。再把這四小塊接起來就會是一個完整的 universal shift register。

在實作時遇到的困難是這四個小部分的 input/output 有點多，容易接錯，所以很考驗細心程度。

Testbench:

我用的是老師提供的 tb 來稍作修改。

老師給的 tb:

```

module t_Shift_Register_4_beh ();
    reg      s1, s0,      // Select inputs
            MSB_in, LSB_in, // Serial inputs
            clk, reset_b;  // Clock and Clear_b

    reg [3:0] I_par;      // Parallel input
    wire [3:0] A_par;     // Register output

    Shift_Register_4_beh M0 (A_par, I_par, s1, s0, MSB_in, LSB_in, clk, reset_b);

    initial #200 $finish;
    initial begin clk = 0; forever #5 clk = ~clk; end

    initial fork
        // test reset action load
        #3 reset_b = 1;
        #4 reset_b = 0;
        #9 reset_b = 1;

        // test parallel load
        #10 I_par = 4'hA;
        #10 {s1, s0} = 2'b11;

        // test shift right
        #30 MSB_in = 1'b0;
        #30 {s1, s0} = 2'b01;

        // test shift left
        #80 LSB_in = 1'b1;
        #80 {s1, s0} = 2'b10;

        // test circulation of data
        #130 {s1, s0} = 2'b11;
        #140 {s1, s0} = 2'b00;

        // test reset on the fly
        #150 reset_b = 1'b0;
        #160 reset_b = 1'b1;
        #160 {s1, s0} = 2'b11;

    join
endmodule

```

Exercise 2: BCD synchronous counter

Design and verify the BCD synchronous counter using T flip-flops with asynchronous reset and Verilog HDL structural (gate-level) modeling

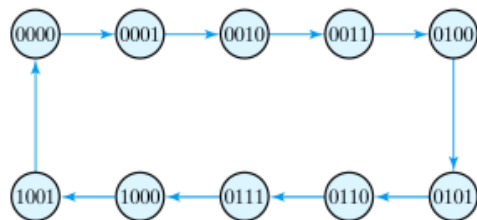


Fig. 6.9 State diagram of a decimal BCD counter

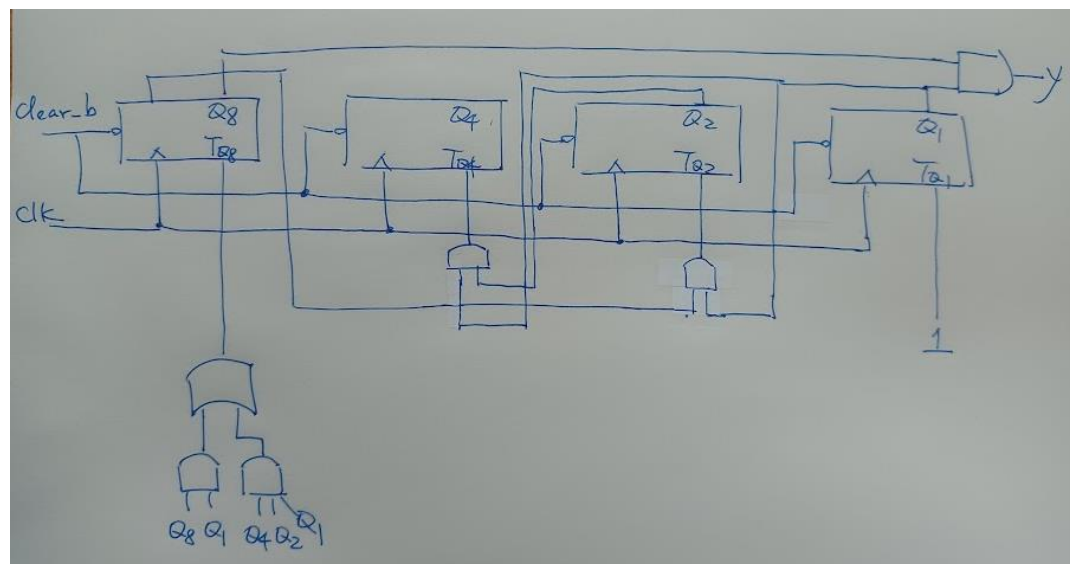
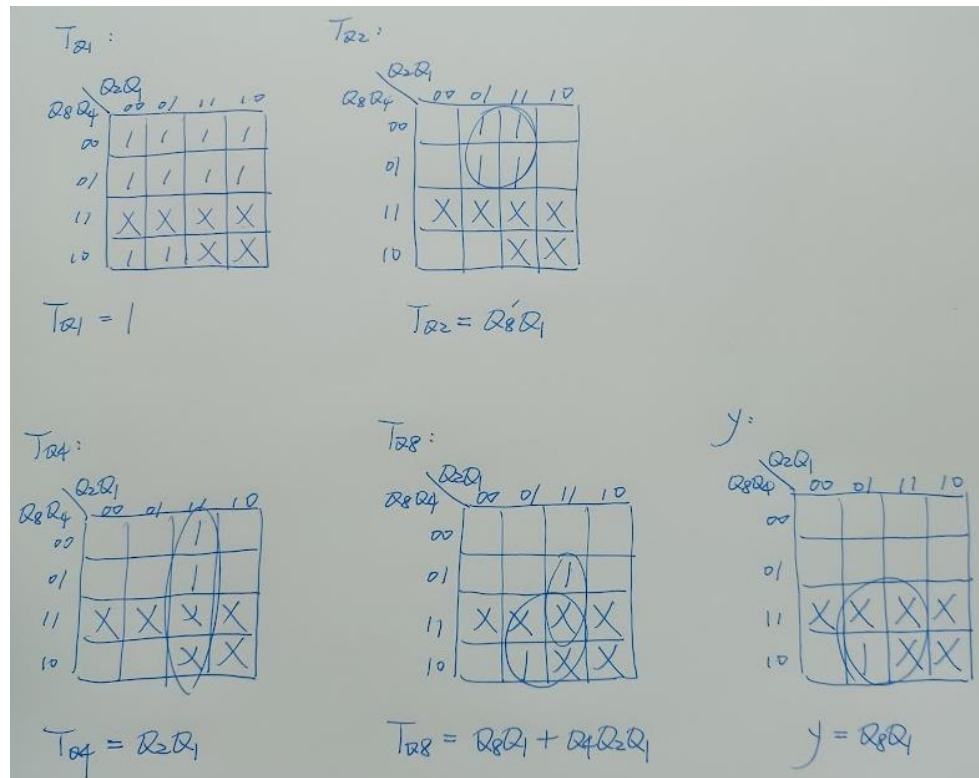
Table 6.5
State Table for BCD Counter

| Present State | | | | Next State | | | | Output | Flip-Flop Inputs | | | |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|--------|------------------|-----------------|-----------------|-----------------|
| Q ₈ | Q ₄ | Q ₂ | Q ₁ | Q ₈ | Q ₄ | Q ₂ | Q ₁ | y | TQ ₈ | TQ ₄ | TQ ₂ | TQ ₁ |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |

| | |
|---------------------------------|--|
| $T_{Q1} = 1$ $T_{Q2} = Q_8'Q_1$ | $T_{Q4} = Q_2Q_1$ $T_{Q8} = Q_8Q_1 + Q_4Q_2Q_1$ $y = Q_8Q_1$ |
|---------------------------------|--|

Exercise2 實驗過程描述與推導:

這題需要先做 design，但其實老師已經把 design 的大部分過程都提供給我們了，以下我把卡諾圖跟最終的電路圖畫出來:



T flip-flop 的寫法練習過，所以只是複習。在電路圖畫出來之後只要電線不要接錯，structural modeling 就不會有太大的問題。

Testbench:

這題的 testbench 一樣是拿老師提供的來修改。

老師提供的 testbench:

```
//Stimulus for testing ripple counter
module testcounter;
    reg Count;
    reg Reset;
    wire A0,A1,A2,A3;
    //Instantiate ripple counter
    Ripple_Counter_4bit M0 (A3, A2, A1, A0, Count, Reset);
    always
    #5 Count = ~Count;
    initial
    begin
        Count = 1'b0;
        Reset = 1'b1;
        #4 Reset = 1'b0;
    end

    initial #170 $finish;

endmodule
```

實驗心得

這次實驗途中電腦突然掛掉，讓我沒截圖到實驗結果電腦就開不起來了，但這次運氣好，掛掉的時候我已經檢查完兩題。我會把 vivado 灌好，下次實驗用自己的筆電來進行，以免下禮拜慘劇又再發生，也謝謝助教讓我有補救的方案。