

第 7 講 行為描述---if and case

1. 參考文獻:

[1]Online Verilog Simulator: edaplayground 教學:

<https://www.youtube.com/watch?v=f9uwtAax4v0&t=249s>

[2]金門大學陳鍾誠老師電子書: <http://ccckmit.wikidot.com/ve:main>

[3]林灶生，verilog 晶片設計第 6 章，全華書局

2. 基本觀念[3]:

2.1 if 敘述

```
if(<條件表示 1>
  begin
    <敘述方塊 1>;
  end
else if(<條件表示式 2>)
  begin
    <敘述方塊 2>;
  end
else if(<條件表示式 n>)
  begin
    <敘述方塊 n>;
  end
else
  begin
    <敘述方塊 n+1>;
  end
end
```

或

```
if(<條件表示式 1>
  begin
    if(<條件表示式 2>)
      begin
        <敘述區塊 1>;
      end
    else
      begin
        <敘述區塊 2>;
      end
    end
  else
    begin
      if(<條件表示式 3>)
        begin
          <敘述區塊 3>;
        end
      else
        begin
          <敘述區塊 3>;
        end
      end
    end
  end
```

2.2 case 敘述

```

case(狀況表示式)
  狀況 1:
    begin
      <敘述區 1>;
    end
  狀況 2:
    begin
      <敘述區 2>;
    end
  .....
  default : begin
    <敘述區 n>;
  end
endcase

```

2.2.1 casez 敘述

casez 敘述與 case 之語法結構之執行過程完全一樣。唯一不同的地方在於其狀況表示若出現(Z)及(?)時，其狀況判定時可當成 don't care，亦即出現(z)或(?)之位元不做比較。

2.2.2 casex 敘述

casex 敘述與 casez 與 case 之語法結構之執行過程完全一樣。唯一不同的地方在於其狀況表示若出現(Z)、(?)及(x)時，其狀況判定時可當成 don't care 而不做比較。

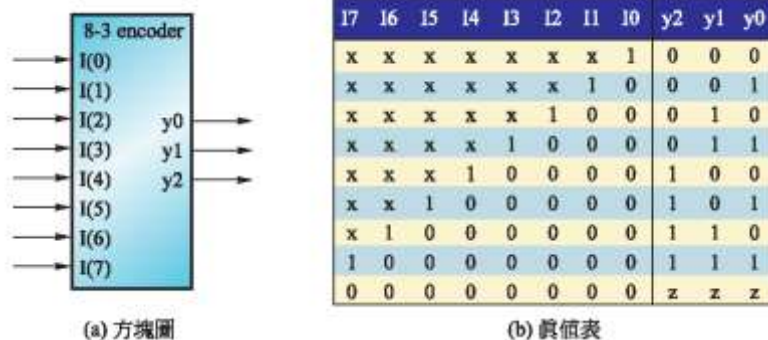
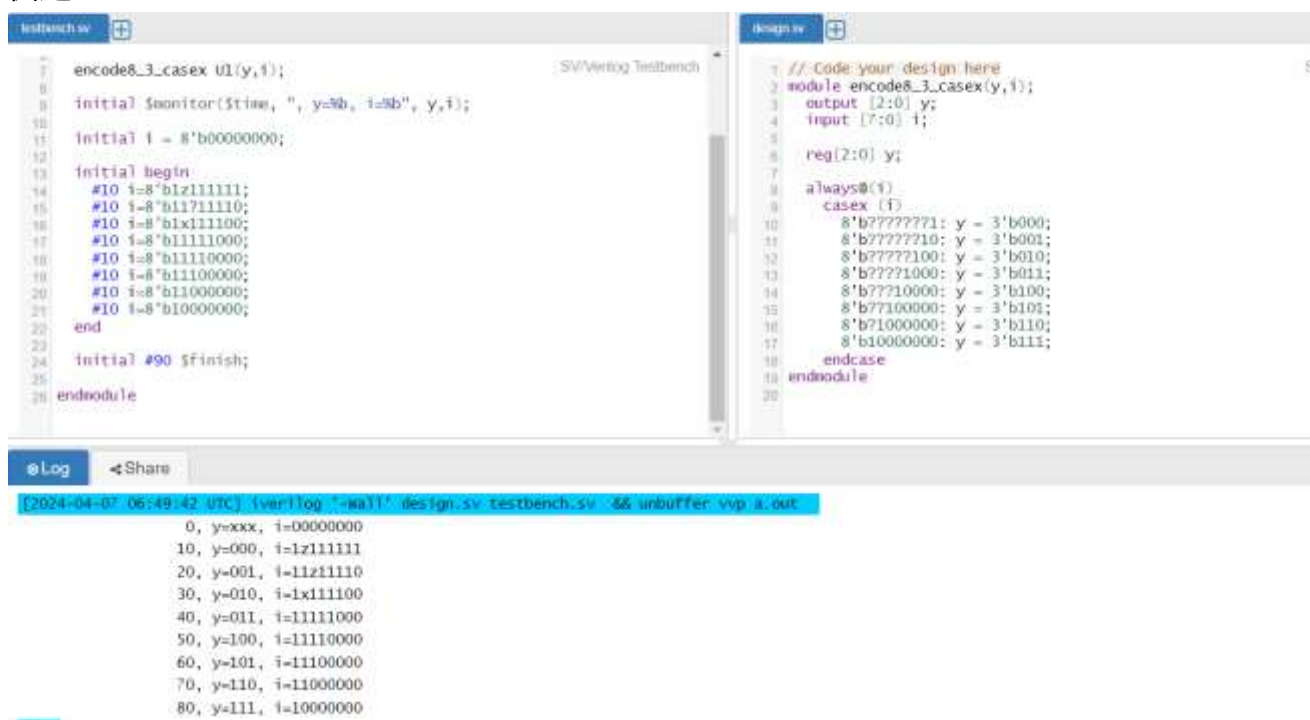


圖 1 8 對 3 優先編碼器

例題 1:



```
testbench.sv
7 encode8_3_casex u1(y,i);
8 initial $monitor($time, " y=%b, i=%b", y,i);
9
10 initial i = 8'b00000000;
11
12 initial begin
13     #10 i=8'b1z111111;
14     #10 i=8'b11711111;
15     #10 i=8'b1x111100;
16     #10 i=8'b11111000;
17     #10 i=8'b11110000;
18     #10 i=8'b11100000;
19     #10 i=8'b11000000;
20     #10 i=8'b10000000;
21     #10 i=8'b10000000;
22 end
23
24 initial #90 $finish;
25
26 endmodule

design.sv
1 // Code your design here
2 module encode8_3_casex(y,i);
3     output [2:0] y;
4     input [7:0] i;
5
6     reg[2:0] y;
7
8     always@(i)
9         casex (i)
10            8'b??????1: y = 3'b000;
11            8'b?????10: y = 3'b001;
12            8'b?????100: y = 3'b010;
13            8'b?????1000: y = 3'b011;
14            8'b????10000: y = 3'b100;
15            8'b??100000: y = 3'b101;
16            8'b?1000000: y = 3'b110;
17            8'b10000000: y = 3'b111;
18        endcase
19 endmodule

[2024-04-07 06:49:42 UTC] [verilog -wall' design.sv testbench.sv -&& unbuffer svp a.out
0, y=xxx, i=00000000
10, y=000, i=1z111111
20, y=001, i=11z11110
30, y=010, i=1x111100
40, y=011, i=11111000
50, y=100, i=11110000
60, y=101, i=11100000
70, y=110, i=11000000
80, y=111, i=10000000
```

- **design.sv** 程式:

```
module encode8_3_casex(y,i);
    output [2:0] y;
    input [7:0] i;

    reg[2:0] y;

    always@(i)
        casex (i)
            8'b??????1: y = 3'b000;
            8'b?????10: y = 3'b001;
            8'b?????100: y = 3'b010;
            8'b?????1000: y = 3'b011;
            8'b????10000: y = 3'b100;
            8'b??100000: y = 3'b101;
            8'b?1000000: y = 3'b110;
            8'b10000000: y = 3'b111;
        endcase
endmodule
```

- **testbench.sv** 程式:

```
module testbench();
  reg [7:0] i; //input
  wire [2:0] y; //output

  encode8_3_casex U1(y,i);

  initial $monitor($time, ", y=%b, i=%b", y,i);

  initial i = 8'b00000000;

  initial begin
    #10 i=8'b1z1111111;
    #10 i=8'b11?111110;
    #10 i=8'b1x1111100;
    #10 i=8'b111111000;
    #10 i=8'b111110000;
    #10 i=8'b111100000;
    #10 i=8'b110000000;
    #10 i=8'b100000000;
  end

  initial #90 $finish;

endmodule
```

- **執行後結果:**

```
[2024-04-07 10:39:11 UTC] iverilog '-wall' design.sv testbench.sv &&
```

```
unbuffer vvp a.out
```

```
0, y=0000, sel=000, a=0000, b=1010, c=1111
10, y=0000, sel=001, a=0000, b=1010, c=1111
20, y=1010, sel=010, a=0000, b=1010, c=1111
30, y=0000, sel=011, a=0000, b=1010, c=1111
40, y=1111, sel=100, a=0000, b=1010, c=1111
50, y=0000, sel=101, a=0000, b=1010, c=1111
60, y=1010, sel=110, a=0000, b=1010, c=1111
70, y=0000, sel=111, a=0000, b=1010, c=1111
80, y=0000, sel=000, a=0000, b=1010, c=1111
```

Done

3. 作業 7-1: 3 對 1 優先多工器

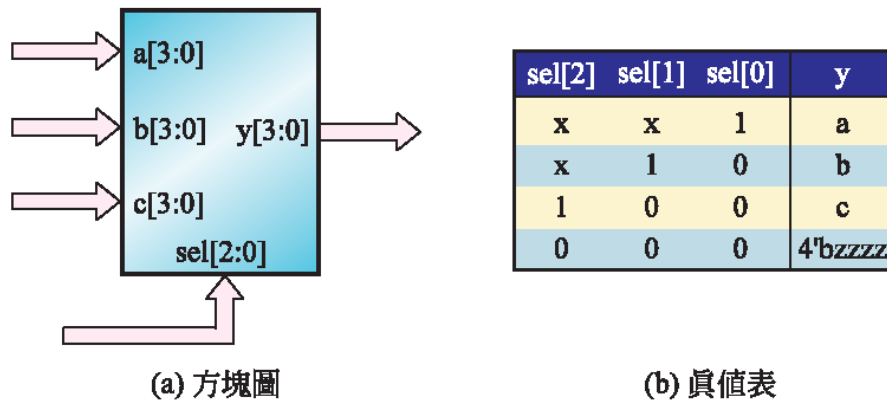


圖 2 3 對 1 優先多工器

● **design.sv** 程式:

```
module mul3_1_casez(y,sel,a,b,c);
    output [3:0] y;
    input [2:0] sel;
    input [3:0] a,b,c;

    reg [3:0] y;

    always@(sel or a or b or c)
        begin
            casez (sel)
                3'b100: y=c;
                3'bz10: y=b;
                3'bzzz: y=a;
                default: y=4'bzzzz;
            endcase
        end
endmodule
```

(1) 請編寫 testbench 程式，並執行該程式，將結果貼到作業報告中。