第 4 講 閘層(Gate Level)描述

1. **參考文獻**:

   [1]Online Verilog Simulator: edaplayground 教學:
   https://www.youtube.com/watch?v=f9uwtAax4v0&t=249s

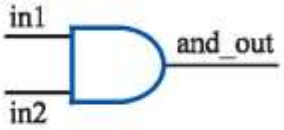
   [2]金門大學陳鍾誠老師電子書**:** http://ccckmit.wikidot.com/ve:main

   [3]林灶生，verilog 晶片設計第 4 章，全華書局

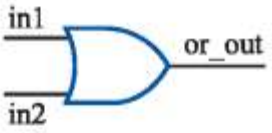
2. **基本觀念[3]**

   - and、or、nand、nor、xor、xnor、buf 及 not 閘

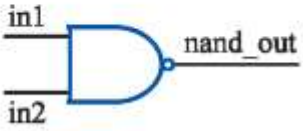
and_out	in2			
	0	1	x	z
0	0	0	0	0
1	0	1	x	x
in1 x	0	x	x	x
z	0	x	x	x



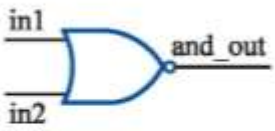
or_out	in2			
	0	1	x	z
0	0	0	0	0
1	1	1	1	1
in1 x	x	1	x	x
z	x	1	x	x



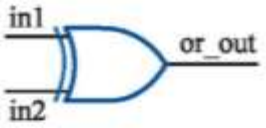
nand_out	in2			
	0	1	x	z
0	1	1	1	1
1	1	0	x	x
in1 x	1	x	x	x
z	1	x	x	x



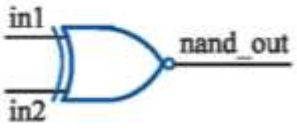
nor_out	in2			
	0	1	x	z
0	1	0	x	x
1	0	0	0	0
in1 x	x	0	x	x
z	x	0	x	x



xor_out	in2			
	0	1	x	z
0	0	1	x	x
1	1	0	x	x
in1 x	x	x	x	x
z	x	x	x	x



xnor_out	in2			
	0	1	x	z
0	1	0	x	x
1	0	1	x	x
in1 x	x	x	x	x
z	x	x	x	x

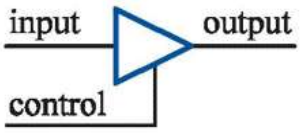


```
// 產生基本邏輯閘元件
and (out, in1, in2) ;          //產生二輸入，一輸出之 and 閘，且無 instance 名稱
and and3(out, a1, a2, a3);     //產生三個輸入，一輸出之 and 元件 and3
or   or2(out, in1, in2);       //以 or 閘產生二個輸入，一個輸出之 or2 元件
nor nor3(out, in1, in2, in3);  //以 nor 閘產生三個輸入一個輸出之 nor3 元件
xor xor3(out, in1, in2, in3);  //以 xor 產生三個輸入，一個輸出之 xor3 元件
xnor(out,in1,in2);             //產生二個輸入一個輸出之 xnor 元件，且無 instance 名稱
```

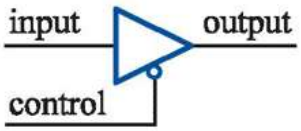
```
//利用 buf 及 not 產生元件例證
buf(out1, out2, out3, in);        //產生三輸出，一輸入之 buf 元件
not(out, in);                     //產生一輸出，一輸入之 not 元件
                                  //上述二元件均無 instance 名稱
not n2(out, out2, in);            //利用 not 閘產生雙輸出，一輸入之元件 n2
```

- bufif1、bufif0、notif1、notif0:

bufif1		Control			
		0	1	x	z
in	0	z	1	H	H
	1	z	0	L	L
	x	z	x	x	x
	z	z	x	x	x

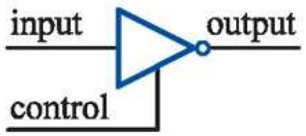


notif0		Control			
		0	1	x	z
in	0	1	z	H	H
	1	0	z	L	L
	x	x	z	x	x
	z	x	z	x	x

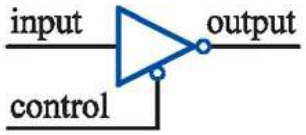



```
//以 bufif1,bufif0,notif1 及 notif0 產生元件例證
bufif1(outb, inb, ctrlb);   //產生一個 bufif1 之緩衝器,輸入爲 inb,控制訊號爲 ctrlb,
                            //.及輸出爲 outb
```

notif1		Control			
		0	1	x	z
in	0	z	1	H	H
	1	z	0	L	L
	x	z	x	x	x
	z	z	x	x	x

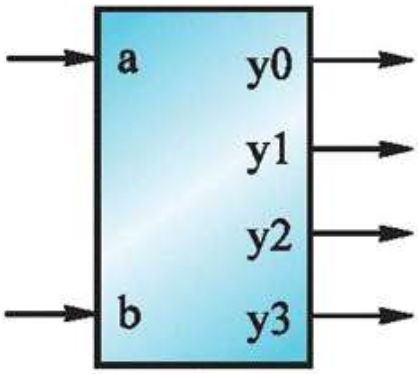


notif0		Control			
		0	1	x	z
in	0	1	z	H	H
	1	0	z	L	L
	x	x	z	x	x
	z	x	z	x	x




```
notif0 n0(outn, inn, ctrln);//利用 notif0 產生 n0 之緩衝器輸入爲 inn，控制訊號爲 ctrln
                            //及輸出爲 outn
```

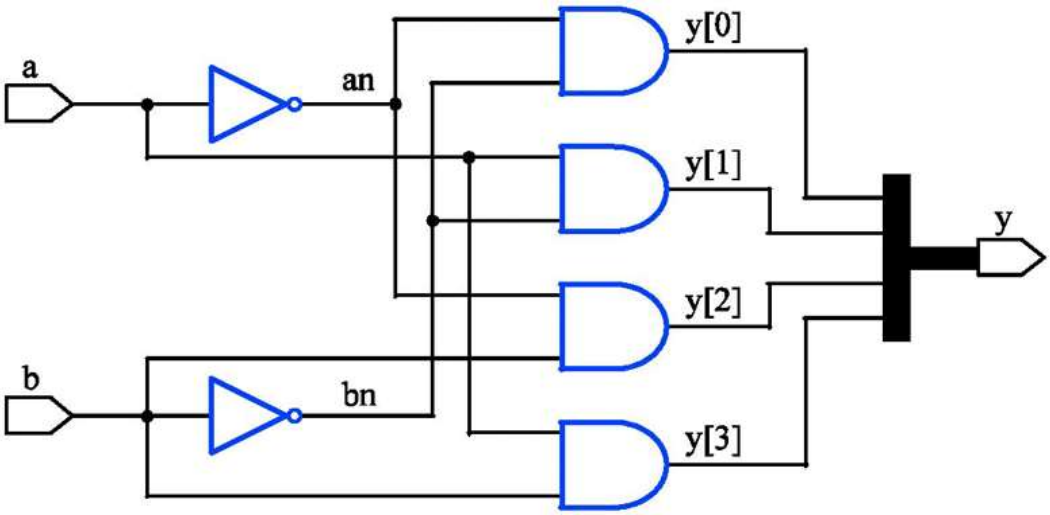
例題 1: 解碼器[3]



(a) 方塊圖

a	b	y3	y2	y1	y0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

(b) 眞值表



testbench.sv

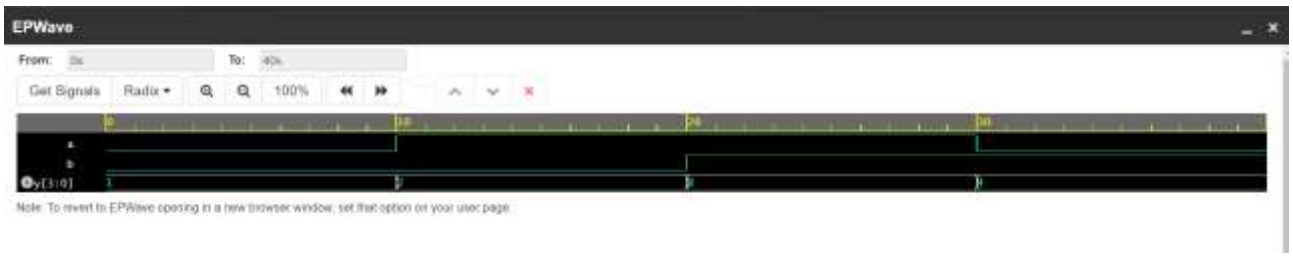
```
// Code your testbench here
// or browse Examples

module testbench;
  wire [3:0] y;
  reg a;
  reg b;

  deco2_4g UUT( .y(y), .a(a), .b(b));

  initial begin
    $dumpfile("dump.vcd");
    $dumpvars(1);

    a=1'b0; b=1'b0; //time = 0s

    #10  a=1'b1; //time = 10s

    #10  a=1'b1; b=1'b1;//time = 20s

    #10  a=1'b0; b=1'b1;//time = 30s

  end

  initial #40 $finish;

  initial begin
    $monitor($time, "    a=%b, b = %b, y= %4b", a,b,y);
```

design.sv

```
// Code your design here
module deco2_4g(a, b, y);

  input a, b;
  output [3:0] y;

  wire an, bn;

  not(an, a);
  not(bn, b);

  and(y[0], an, bn);
  and(y[1], a, bn);
  and(y[2], an, b);
  and(y[3], a, b);
endmodule
```

Log | Share

```
[2024-03-10 10:21:55 UTC] iverilog '-Wall' design.sv testbench.sv  && unbuffer vvp a.out
VCD info: dumpfile dump.vcd opened for output.
                   0    a=0, b = 0, y= 0001
                  10    a=1, b = 0, y= 0010
                  20    a=1, b = 1, y= 1000
                  30    a=0, b = 1, y= 0100
```



以下為 design.sv 欄內的模組程式

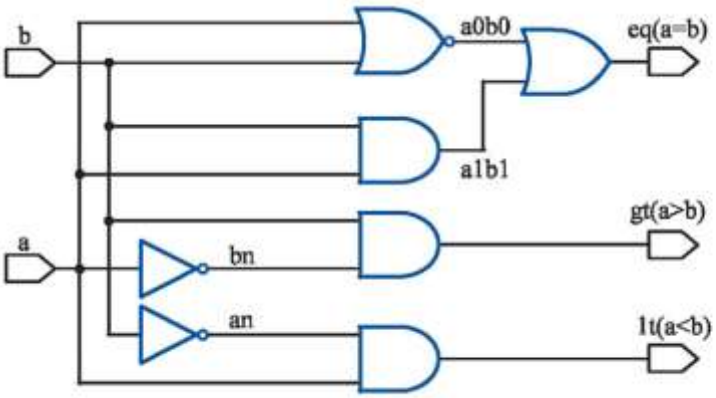
```
module deco2_4g(a, b, y);

   input a, b;
   output [3:0] y;

   wire an, bn;

   not(an, a);
   not(bn, b);

   and(y[0], an, bn);
   and(y[1], a, bn);
   and(y[2], an, b);
   and(y[3], a, b);
endmodule
```

以下為 testbench.sv 欄內的測試程式

```
module testbench;
   wire [3:0] y;
   reg a;
   reg b;

   deco2_4g UUT( .y(y), .a(a), .b(b));

   initial begin
      $dumpfile("dump.vcd");
      $dumpvars(1);

      a=1'b0; b=1'b0; //time = 0s

      #10    a=1'b1; //time = 10s

      #10    a=1'b1; b=1'b1;//time = 20s

      #10    a=1'b0; b=1'b1;//time = 30s

   end
```

```
    initial #40 $finish;

    initial begin
        $monitor($time, "        a=%b, b = %b, y= %4b", a,b,y);
    end

endmodule
```

3. 作業 **4-1: 1-bit comparator (1** 位元比較器**)**



(a) 電路圖

a	b	eq	gt	lt
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

(b) 真值表

```
module comp1 (a, b, eq, gt,lt);
    input a,b;
    output eq,gt,lt;

    wire an, bn, a0b0, a1b1;

    not(an, a);
    not(bn, b);
    and(a1b1, a, b);
    and(a0b0, a, b);

    or(eq, a0b0, a1b1);
    and(gt, a, bn);
    and(lt, an, b);
```

endmodule

(1) 請編寫 testbench 程式、執行結果與圖形貼到作業報告中。