4-11 組合電路的硬體描述語言

- 閘階層模型

  關鍵字 and 、nand 、or 、nor 、xor 、xnor 、not 、buf

表 4-9　預先定義原始閘的真值表

and	0	1	x	z
0	0	0	0	0
1	0	1	x	x
x	0	x	x	x
z	0	x	x	x

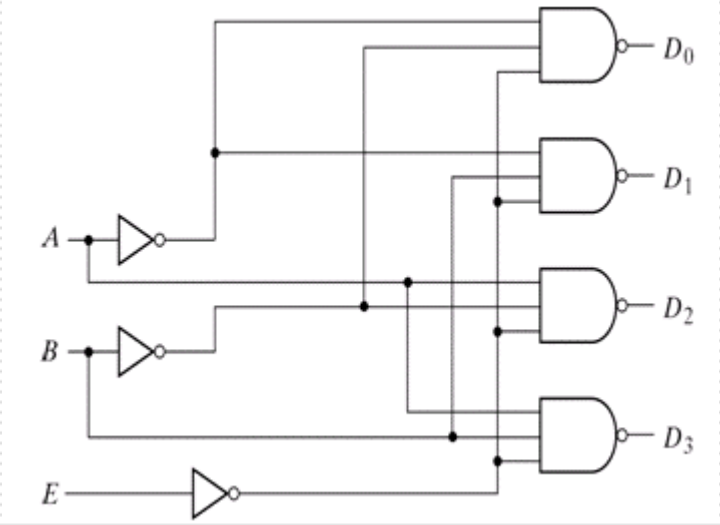
or	0	1	x	z
0	0	1	x	x
1	1	1	1	1
x	x	1	x	x
z	x	1	x	x

xor	0	1	x	z
0	0	1	x	x
1	1	0	x	x
x	x	x	x	x
z	x	x	x	x

not	輸入	輸出
	0	1
	1	0
	x	x
	z	x

HDL 範例 4-1
(2對4線解碼器的閘階層描述)

```
//Gate-level description of a 2-to-4-line
   decoder
//Figure 4-19
module decoder_gl (A,B,E,D);
   input A,B,E;
   output [0:3]D;
   wire Anot,Bnot,Enot;
   not
      n1 (Anot,A),
      n2 (Bnot,B),
```



HDL 範例 4-1
(2對4線解碼器的閘階層描述)

```
    n3 (Enot,E);
  nand
    n4 (D[0],Anot,Bnot,Enot),
    n5 (D[1],Anot,B,Enot),
    n6 (D[2],A,Bnot,Enot),
    n7 (D[3],A,B,Enot);
endmodule
```

HDL 範例 4-2
(4位元加法器之底部向上層次化描述)

```
//Gate-level hierarchical description of 4-bit
   adder
// Description of half adder (see Fig 4-5b)
module  halfadder (S,C,x,y);
   input  x,y;
   output  S,C;
//Instantiate primitive gates
   xor  (S,x,y);
   and (C,x,y);
```

HDL 範例 4-2
(4位元加法器之底部向上層次化描述)

```
endmodule
//Description of full adder (see Fig 4-8)
module  fulladder (S,C,x,y,z);
   input  x,y,z;
   output  S,C;
   wire  S1,D1,D2;
//Outputs of first XOR and two AND gates
//Instantiate the halfadder
    halfadder  HA1 (S1,D1,x,y),
```

HDL 範例 4-2
(4位元加法器之底部向上層次化描述)

```
                HA2 (S,D2,S1,z);
    or  g1(C,D2,D1);
endmodule
//Description of 4-bit adder (see Fig
   4-9)
module  _4bit_adder (S,C4,A,B,C0);
   input  [3:0] A,B;
   input  C0;
```

HDL 範例 4-2
(4位元加法器之底部向上層次化描述)

```
    output [3:0] S;
    output C4;
    wire C1,C2,C3;   //Intermediate carries
  //Instantiate the fulladder
    fulladder FA0 (S[0],C1,A[0],B[0],C0),
                  FA1 (S[1],C2,A[1],B[1],C1),
                  FA2 (S[2],C3,A[2],B[2],C2),
                  FA3 (S[3],C4,A[3],B[3],C3);
endmodule
```

三態閘

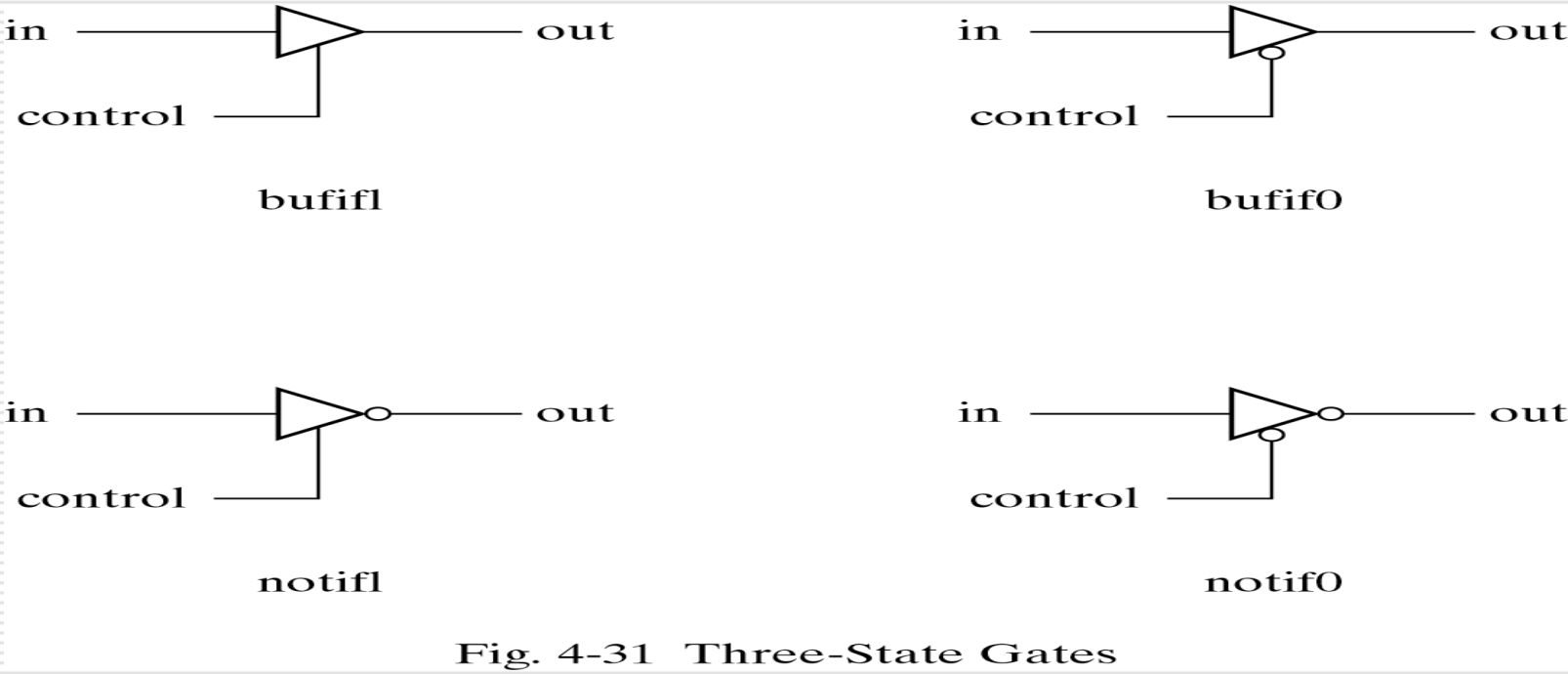


Fig. 4-31 Three-State Gates

```
bufif1 (OUT, A, control);
notif0 (Y, B, enable);
```

具有三態緩衝器之2對1線多工器

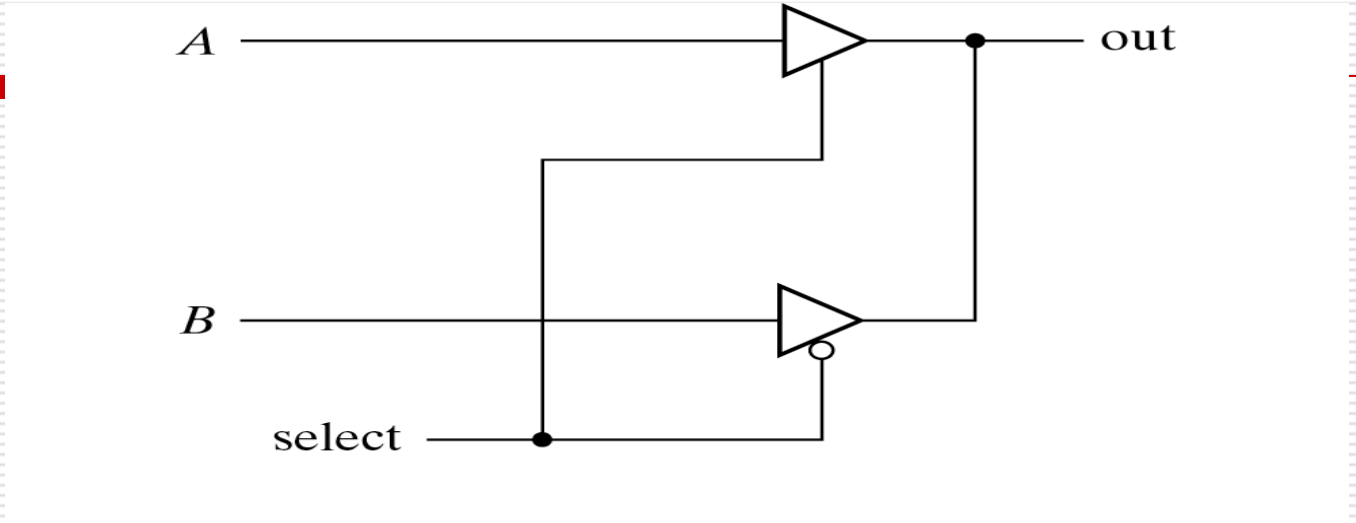


Fig. 4-32 2-to-1-Line Multiplexer with Three-State Buffers

```
module muxtri (A, B, select, OUT);
    input A, B, select;
    output OUT;
    tri OUT;
    bufif1 (OUT, A, select);
    bufif0 (OUT, B, select);
endmodule
```

資料流程模型

- Verilog HDL運算子

表 4-10　Verilog HDL 運算子

符號	運算
+	binary addition；二進位加法
−	binary subtraction；二進位減法
&	bit-wise AND；位元的及運算
|	bit-wise OR；位元的或運算
^	bit-wise XOR；位元的互斥或運算
~	bit-wise NOT；位元的反相運算
==	equality；全等
>	greater than；大於
<	less than；小於
{ }	concatenation；連結
?:	conditional；條件式

HDL 範例 4-3
(2對4線解碼器的資料流程描述)

```
//Dataflow description of a 2-to-4-line
   decoder
//See Fig.4-19
module  decoder_df (A,B,E,D);
  input  A,B,E;
  output  [0:3] D;
  assign  D[0] = ~(~A & ~B & ~E),
          D[1] = ~(~A & B & ~E),
          D[2] = ~(A & ~B & ~E),
          D[3] = ~(A & B & ~E);
endmodule
```

HDL 範例 4-4
(4位元加法器之資料流程描述)

```
//Dataflow description of 4-bit adder
module  binary_adder (A,B,Cin,SUM,Cout);
   input  [3:0] A,B;
   input  Cin;
   output  [3:0] SUM;
   output  Cout;
   assign  {Cout,SUM} = A + B + Cin;
endmodule
```

HDL 範例 4-5 (大小比較器之資料流程描述)

```
//Dataflow description of a 4-bit comparator.
module  magcomp (A,B,ALTB,AGTB,AEQB);
  input  [3:0] A,B;
  output  ALTB,AGTB,AEQB;
  assign  ALTB = (A < B),
          AGTB = (A > B),
          AEQB = (A = = B);
endmodule
```

HDL 範例 4-6
(2對1線多工器使用條件式運算子描述)

```
//Dataflow description of 2-to-1-line
    multiplexer
module  mux2x1_df (A,B,select,OUT);
   input  A,B,select;
   output  OUT;
   assign  OUT = select ? A : B;
endmodule
```

行為模型

- HDL 範例 4-7
  (2對1線多工器使用條件式運算子描述)

```
//Behavioral description of 2-to-1-line multiplexer
module  mux2x1_bh(A,B,select,OUT);
  input  A,B,select;
  output  OUT;
  reg  OUT;
  always @ (select  or  A  or  B)
      if  (select == 1)  OUT = A;
      else  OUT = B;
endmodule
```

HDL 範例 4-8
(4對1線多工器之行為描述)

```
//Behavioral description of 4-to-1- line
    multiplexer
//Describes the function table of  Fig. 4-25(b).
module  mux4x1_bh (i0,i1,i2,i3,select,y);
    input  i0,i1,i2,i3;
    input  [1:0] select;
    output  y;
    reg  y;
```

HDL 範例 4-8
(4對1線多工器之行為描述)

```
always @ (i0 or i1 or i2 or i3 or select)
        case (select)
            2'b00: y = i0;
            2'b01: y = i1;
            2'b10: y = i2;
            2'b11: y = i3;
        endcase
endmodule
```

測試平台

- 一個模擬模組是具有下列形式的HDL程式

  module 測試名稱。

  宣告特有的reg和wire識別字。

  在測試下例示設計模組。

  利用initial及always敘述產生模擬。

  顯示輸出響應。

  endmodule。

系統功能

- $ display-- 顯 示 具 有 end-of-line return之變數或字串的一次值。
- $write—與 $display相同，但是沒有到下一行。
- $monitor—當在模擬期間值改變時及顯示變數。
- $time--顯示模擬時間。
- $finish--結束模擬。

模擬與設計的交互模組

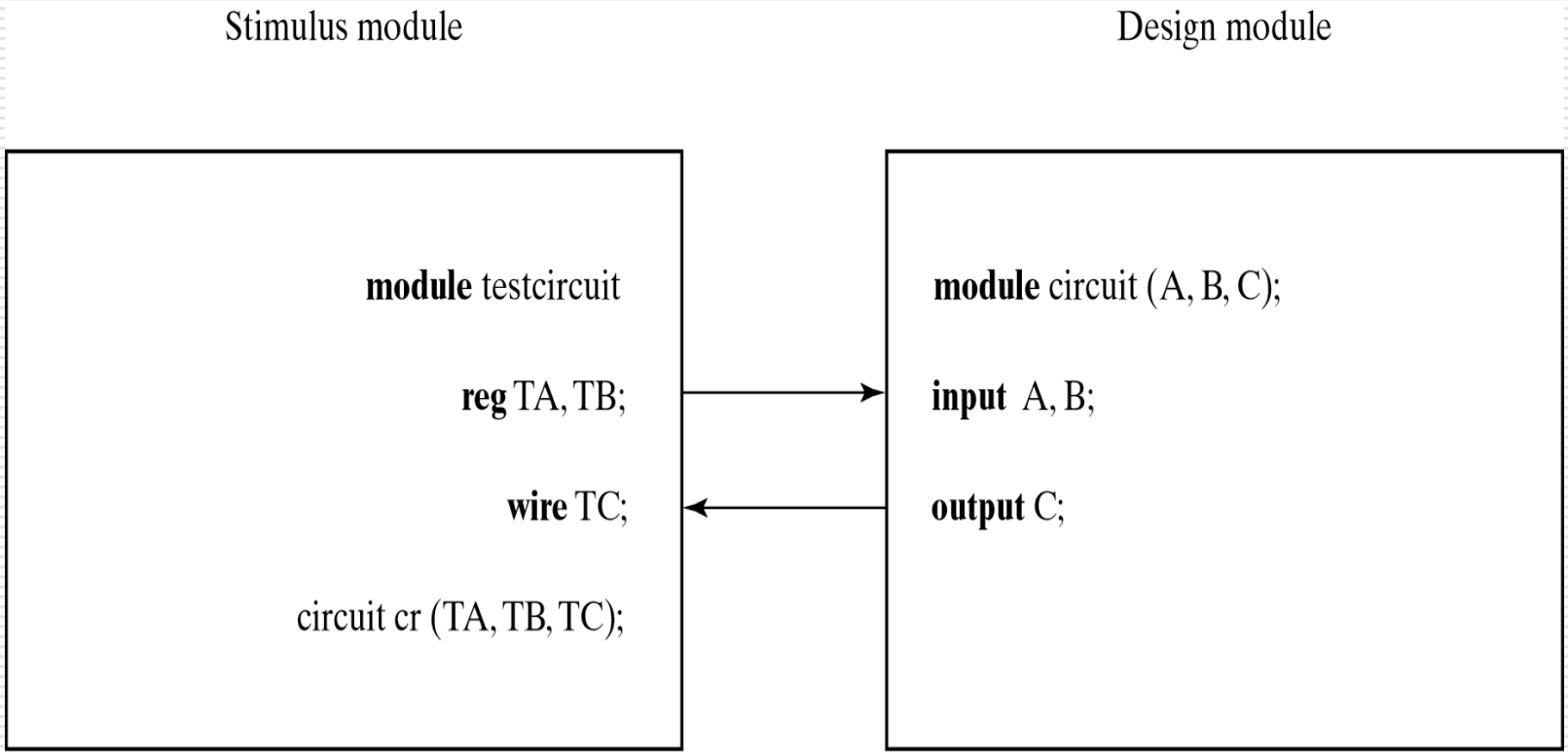


Fig. 4-33 Stimulus and Design Modules Interaction

HDL 範例 4-9
(測試範例4-6所描述的2對1多工器)

```
//Stimulus for mux2x1_df.
module  testmux;
    reg  TA,TB,TS;  //inputs for mux
    wire  Y;        //output from mux
    mux2x1_df mx (TA,TB,TS,Y);
 //   instantiate mux
     initial
```

HDL 範例 4-9
(測試範例4-6所描述的2對1多工器)

```
begin
        TS = 1; TA = 0; TB = 1;
   #10 TA = 1; TB = 0;
   #10 TS = 0;
   #10 TA = 0; TB = 1;
end
```

HDL 範例 4-9
(測試範例4-6所描述的2對1多工器)

```
    initial
        $monitor("select = %b  A = %b
    B = %b  OUT = %b  time = %0d",
    TS, TA, TB, Y, $time);
endmodule
```

HDL 範例 4-9
(測試範例4-6所描述的2對1多工器)

```
//Dataflow description of 2-to-1-line
    multiplexer
//from Example 4-6
module  mux2x1_df (A,B,select,OUT);
   input  A,B,select;
   output  OUT;
   assign  OUT = select ? A : B;
endmodule
```

HDL 範例 4-9
(測試範例4-6所描述的2對1多工器)

```
Simulatiom  log:
select = 1  A=0  B=1  OUT=0  time=0
select = 1  A=1  B=0  OUT=1  time=10
select = 0  A=1  B=0  OUT=0  time=20
select = 0  A=0  B=1  OUT=1  time=30
```

HDL 範例 4-10
(全加法器之多階電路的閘階層描述)

```
//Gate-level description of circuit of Fig. 4-2
module analysis (A,B,C,F1,F2);
  input  A,B,C;
  output F1,F2;
  wire   T1,T2,T3,F2not,E1,E2,E3;
  or  g1 (T1,A,B,C);
  and g2 (T2,A,B,C);
```

HDL 範例 4-10
(全加法器之多階電路的閘階層描述)

```
   and  g3 (E1,A,B);
   and  g4 (E2,A,C);
   and  g5 (E3,B,C);
   or   g6 (F2,E1,E2,E3);
   not  g7 (F2not,F2);
   and  g8 (T3,T1,F2not);
   or   g9 (F1,T2,T3);
endmodule
```

HDL 範例 4-10
(全加法器之多階電路的閘階層描述)

```
//Stimulus to analyze the circuit
 module  test_circuit;
    reg  [2:0]D;
    wire  F1,F2;
    analysis fig42(D[2],D[1],D[0],F1,F2);
    initial
       begin
```

HDL 範例 4-10
(全加法器之多階電路的閘階層描述)

```
        D = 3'b000;
        repeat(7)
           #10 D = D + 1'b1;
        end
      initial
         $monitor ("ABC = %b  F1 = %b  F2
    =%b ",
                              D, F1, F2);
endmodule
```

HDL 範例 4-10
(全加法器之多階電路的閘階層描述)

Simulation log:

ABC=000 F1=0 F2=0
ABC=001 F1=1 F2=0
ABC=010 F1=1 F2=0
ABC=011 F1=0 F2=1
ABC=100 F1=1 F2=0
ABC=101 F1=0 F2=1
ABC=110 F1=0 F2=1
ABC=111 F1=1 F2=1