數位邏輯設計 Verilog 實作

113820032 曾靖諺
電資一

1. Verilog Example 3.1 (課本 P.139)

Figure 3.35 is a logic diagram for a simple circuit in which the output of an OR gate is one of two inputs to an AND gate. The Boolean equation for the output of the circuit can be written directly from the diagram: E = (A + B)C.

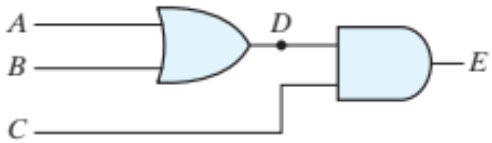


FIGURE 3.35
A logic diagram (schematic) for the Boolean equations $D = A + B$ $E = CD$

實作結果：

or_and.v

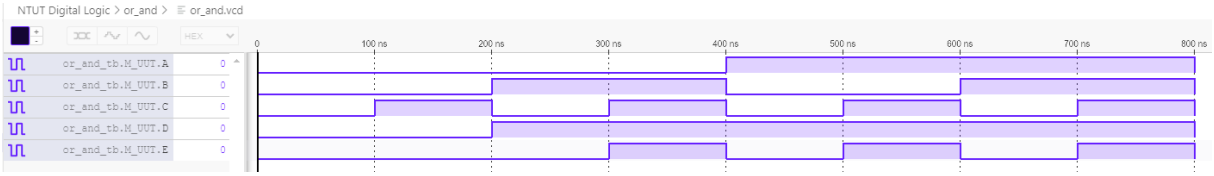
```
module or_and (
    input A, B, C,
    output E
);
    wire D;
    assign D = A | B;
    assign E = C & D;
endmodule
```

or_and_tb.v

```
`timescale 1ns/1ps
`include "or_and.v"

module or_and_tb;
    reg A, B, C;
    wire E;
    or_and M_UUT (A, B, C, E);
    initial begin
        $dumpfile("or_and.vcd");
        $dumpvars(0, or_and_tb);
        A = 0; B = 0; C = 0;
```

```
        #100 A = 0; B = 0; C = 1;
        #100 A = 0; B = 1; C = 0;
        #100 A = 0; B = 1; C = 1;
        #100 A = 1; B = 0; C = 0;
        #100 A = 1; B = 0; C = 1;
        #100 A = 1; B = 1; C = 0;
        #100 A = 1; B = 1; C = 1;
        #100 $finish;
    end
endmodule
```

Simulation waveforms



==========

2. Verilog Example 3.2 (課本 P.141)

This example develops a Verilog model of a circuit having inputs A, B, C, D and outputs E, F, with functionality specified by the following Boolean expressions:

E = A + BC + B′D

F = B′C + BC′D′

實作結果：

Circuit_Boolean_CA.v

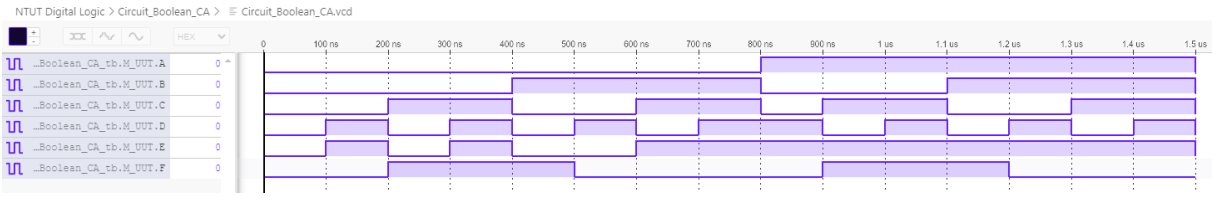
```
module Circuit_Boolean_CA (
    input A, B, C, D,
    output E, F
);

    assign E = A | (B && C) | ((!B) && D);
    assign F = ((!B) && C) | (B & (!C) && (!D));
endmodule
```

Circuit_Boolean_CA_tb.v

```
`timescale 1ns/1ps
`include "Circuit_Boolean_CA.v"
```

```
module Circuit_Boolean_CA_tb;
    reg A, B, C, D;
    wire E, F;

    Circuit_Boolean_CA M_UUT (A, B, C, D, E, F);

    initial begin
        $dumpfile("Circuit_Boolean_CA.vcd");
        $dumpvars(0, Circuit_Boolean_CA_tb);
        A = 0; B = 0; C = 0; D = 0;
        #100 A = 0; B = 0; C = 0; D = 1;
        #100 A = 0; B = 0; C = 1; D = 0;
        #100 A = 0; B = 0; C = 1; D = 1;
        #100 A = 0; B = 1; C = 0; D = 0;
        #100 A = 0; B = 1; C = 0; D = 1;
        #100 A = 0; B = 1; C = 1; D = 0;
        #100 A = 0; B = 1; C = 1; D = 1;
        #100 A = 1; B = 0; C = 0; D = 1;
        #100 A = 1; B = 0; C = 1; D = 0;
        #100 A = 1; B = 0; C = 1; D = 1;
        #100 A = 1; B = 1; C = 0; D = 0;
        #100 A = 1; B = 1; C = 0; D = 1;
        #100 A = 1; B = 1; C = 1; D = 0;
        #100 A = 1; B = 1; C = 1; D = 1;
        #100 $finish;
    end
endmodule
```

Simulation waveforms



==========

3. Verilog Example 3.3 (Structural Modeling with Primitives) (課本 P.145)

D = AB+C’ E = C’

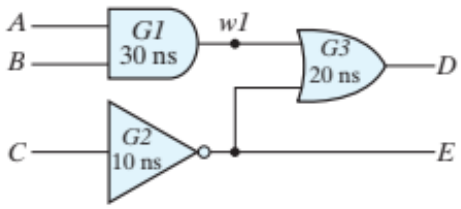


FIGURE 3.37
Schematic for *and_or_prop_delay*

實作結果：

and_or_prop_delay.v

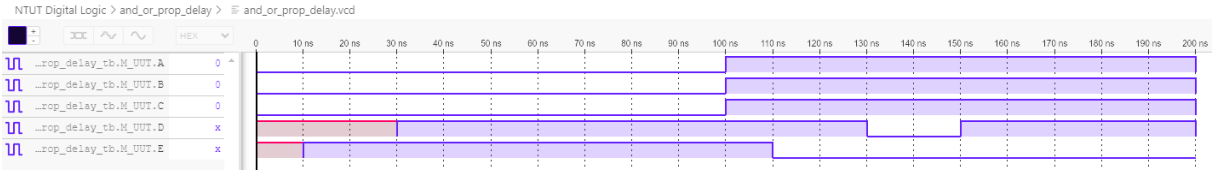
```
module and_or_prop_delay (
    input A, B, C,
    output D, E
);
    wire w1;
    and #30 G1(w1, A, B);
    not #10 G2(E, C);
    or #20 G3(D, w1, E);
endmodule
```

and_or_prop_delay_tb.v

```
`timescale 1ns/1ps
`include "and_or_prop_delay.v"

module and_or_prop_delay_tb;
    reg A, B, C;
    wire D, E;

    and_or_prop_delay M_UUT (A, B, C, D, E);

    initial begin
        $dumpfile("and_or_prop_delay.vcd");
        $dumpvars(0, and_or_prop_delay_tb);
        A = 1'b0; B = 1'b0; C = 1'b0;
        #100 A = 1'b1; B = 1'b1; C = 1'b1;
        #100 $finish;
    end
```

```
endmodule
```

Simulation waveforms



實作波型

vs

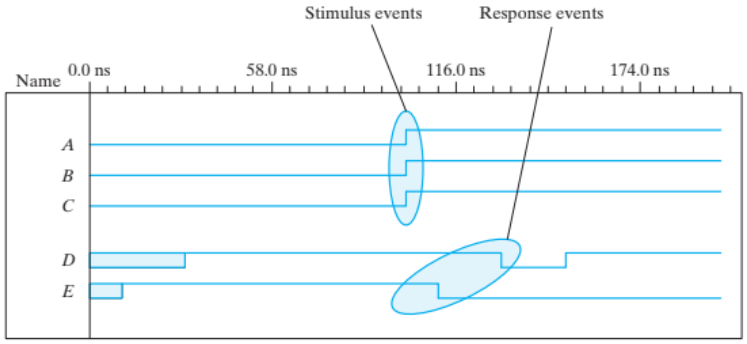


FIGURE 3.38
Simulation waveforms of *and_or_prop_delay*

課本波型

==========

4. Verilog Example 3.4 (User-Defned Primitive) (課本 P.155)

UDP_02467: f (A, B, C) = ∑(0, 2, 4, 6, 7)

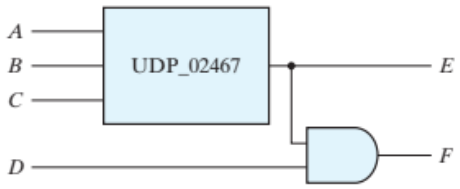


FIGURE 3.40
Schematic for *Circuit with_UDP_02467*

實作結果：

Circuit_with_UDP_02467.v

```
primitive UDP_02467 (D, A, B, C);
    output D;
    input A, B, C;
    //Truth table for D = f (A, B, C) = ∑(0, 2, 4, 6, 7);
    table
```

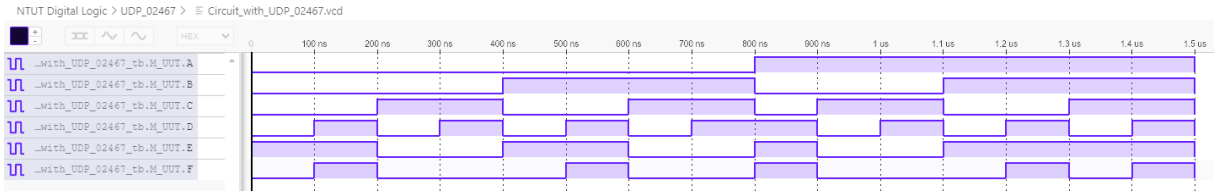
```
        //A B C : D
        0 0 0 : 1;
        0 0 1 : 0;
        0 1 0 : 1;
        0 1 1 : 0;
        1 0 0 : 1;
        1 0 1 : 0;
        1 1 0 : 1;
        1 1 1 : 1;
    endtable
endprimitive

module Circuit_with_UDP_02467 (
    output E, F,
    input A, B, C, D
);
    UDP_02467 (E, A, B, C);  //把結果給 e(因為 UDP_02467 說第一個是 output)
    and (F, E, D);
endmodule
```

Circuit_with_UDP_02467_tb.v

```
`timescale 1ns/1ps
`include "Circuit_with_UDP_02467.v"

module Circuit_with_UDP_02467_tb;
    reg A, B, C, D;
    wire E, F;

    Circuit_with_UDP_02467 M_UUT (E, F, A, B, C, D);

    initial begin
        $dumpfile("Circuit_with_UDP_02467.vcd");
        $dumpvars(0, Circuit_with_UDP_02467_tb);
        A = 0; B = 0; C = 0; D = 0;
        #100 A = 0; B = 0; C = 0; D = 1;
        #100 A = 0; B = 0; C = 1; D = 0;
        #100 A = 0; B = 0; C = 1; D = 1;
        #100 A = 0; B = 1; C = 0; D = 0;
```

```
        #100 A = 0; B = 1; C = 0; D = 1;
        #100 A = 0; B = 1; C = 1; D = 0;
        #100 A = 0; B = 1; C = 1; D = 1;
        #100 A = 1; B = 0; C = 0; D = 1;
        #100 A = 1; B = 0; C = 1; D = 0;
        #100 A = 1; B = 0; C = 1; D = 1;
        #100 A = 1; B = 1; C = 0; D = 0;
        #100 A = 1; B = 1; C = 0; D = 1;
        #100 A = 1; B = 1; C = 1; D = 0;
        #100 A = 1; B = 1; C = 1; D = 1;
        #100 $finish;
    end
endmodule
```

Simulation waveforms



==========

5. 習題 3.36(a)　(課本 P.160)

實作結果：

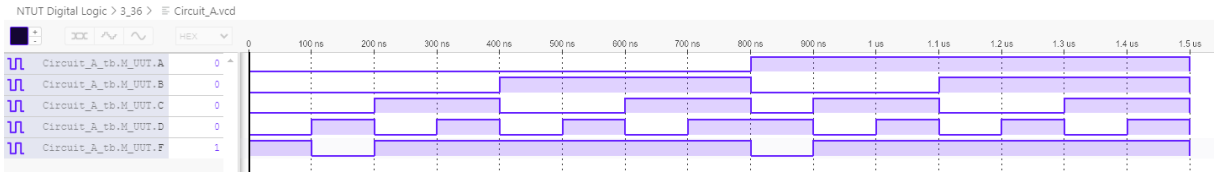
Circuit_A.v

```
module Circuit_A (A, B, C, D, F);
    input A, B, C, D;
    output F;
    wire w, x, y, z, a, d;

    or (x, B, C, d);
    and (y, a ,C);
    and (w, z ,B);
    and (z, y, A);
    or (F, x, w);
    not (a, A);
    not (d, D);
endmodule
```

Circuit_A_tb.v

```
`timescale 1ns/1ps
`include "Circuit_A.v"

module Circuit_A_tb;
    reg A, B, C, D;
    wire F;

    Circuit_A M_UUT (A, B, C, D, F);

    initial begin
        $dumpfile("Circuit_A.vcd");
        $dumpvars(0, Circuit_A_tb);
        A = 0; B = 0; C = 0;
        #100 A = 0; B = 0; C = 1;
        #100 A = 0; B = 1; C = 0;
        #100 A = 0; B = 1; C = 1;
        #100 A = 1; B = 0; C = 0;
        #100 A = 1; B = 0; C = 1;
        #100 A = 1; B = 1; C = 0;
        #100 A = 1; B = 1; C = 1;
        #100 $finish;
    end
endmodule
```

Simulation waveforms



==========

6. 習題 3.36(b) (課本 P.160)

實作結果：

Circuit_B.v

```
module Circuit_B (F1, F2, F3, A0, A1, B0, B1);
    output F1, F2, F3;
    input A0, A1, B0, B1;
```

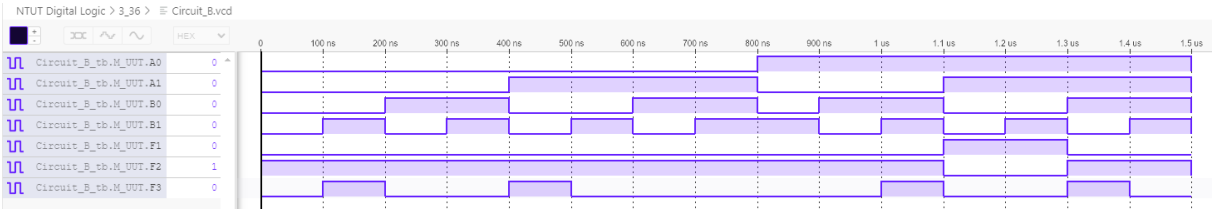
```
    nor (F1, F2, F3);
    or (F2, w1, w2, w3);
    and (F3, w4, w5);
    and (w1, w6, B1);
    or (w2, w6, w7, B0);
    and (w3, w7, B0, B1);
    not (w6, A1);
    not (w7, A0);
    xor (w4, A1, B1);
    xnor (w5, A0, B0);
endmodule
```

Circuit_B_tb.v

```
`timescale 1ns/1ps
`include "Circuit_B.v"

module Circuit_B_tb;
    reg A0, A1, B0, B1;
    wire F1, F2, F3;

    Circuit_B M_UUT (F1, F2, F3, A0, A1, B0, B1);

    initial begin
        $dumpfile("Circuit_B.vcd");
        $dumpvars(0, Circuit_B_tb);
        A0 = 0; A1 = 0; B0 = 0; B1 = 0;
        #100 A0 = 0; A1 = 0; B0 = 0; B1 = 1;
        #100 A0 = 0; A1 = 0; B0 = 1; B1 = 0;
        #100 A0 = 0; A1 = 0; B0 = 1; B1 = 1;
        #100 A0 = 0; A1 = 1; B0 = 0; B1 = 0;
        #100 A0 = 0; A1 = 1; B0 = 0; B1 = 1;
        #100 A0 = 0; A1 = 1; B0 = 1; B1 = 0;
        #100 A0 = 0; A1 = 1; B0 = 1; B1 = 1;
        #100 A0 = 1; A1 = 0; B0 = 0; B1 = 1;
        #100 A0 = 1; A1 = 0; B0 = 1; B1 = 0;
        #100 A0 = 1; A1 = 0; B0 = 1; B1 = 1;
        #100 A0 = 1; A1 = 1; B0 = 0; B1 = 0;
        #100 A0 = 1; A1 = 1; B0 = 0; B1 = 1;
```

```
        #100 A0 = 1; A1 = 1; B0 = 1; B1 = 0;
        #100 A0 = 1; A1 = 1; B0 = 1; B1 = 1;
        #100 $finish;
    end
endmodule
```

Simulation waveforms



==========

7. 習題 3.36(c)　(課本 P.161)

實作結果：

Circuit_C.v

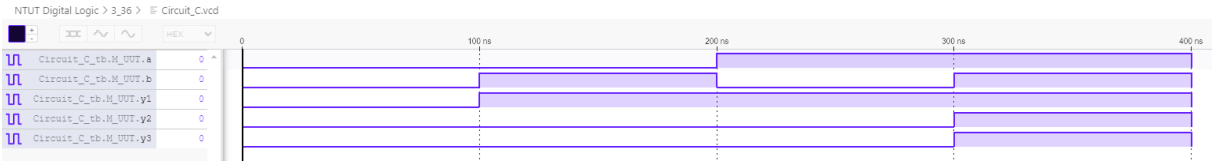
```
module Circuit_C (y1, y2, y3, a, b);
    output y1, y2, y3;
    input a, b;
    assign y1 = a || b;
    and (y2, a, b);
    assign y3 = a && b;
endmodule
```

Circuit_C_tb.v

```
`timescale 1ns/1ps
`include "Circuit_C.v"

module Circuit_C_tb;
    reg a, b;
    wire y1, y2, y3;

    Circuit_C M_UUT (y1, y2, y3, a, b);

    initial begin
        $dumpfile("Circuit_C.vcd");
        $dumpvars(0, Circuit_C_tb);
```

```
        a = 0; b = 0;
        #100 a = 0; b = 1;
        #100 a = 1; b = 0;
        #100 a = 1; b = 1;
        #100 $finish;
    end
endmodule
```

Simulation waveforms



==========

8. 習題 3.38 (課本 P.161)

實作結果：

Circuit_with_UDP_02467.v

```
primitive UDP_02467 (D, A, B, C);
    output D;
    input A, B, C;
    table
        //A B C : D
        0 0 0 : 1;
        0 0 1 : 0;
        0 1 0 : 1;
        0 1 1 : 0;
        1 0 0 : 1;
        1 0 1 : 0;
        1 1 0 : 1;
        1 1 1 : 1;
    endtable
endprimitive

module Circuit_with_UDP_02467 (
    output E, F,
    input A, B, C, D
);
```

```
    UDP_02467 (E, A, B, C);
    and (F, E, D);
endmodule
```

Circuit_with_UDP_02467_tb.v

```
`timescale 1ns/1ps
`include "Circuit_with_UDP_02467.v"

module Circuit_with_UDP_02467_tb;
    reg A, B, C, D;
    wire E, F;

    Circuit_with_UDP_02467 M_UUT (E, F, A, B, C, D);

    initial begin
        $dumpfile("Circuit_with_UDP_02467.vcd");
        $dumpvars(0, Circuit_with_UDP_02467_tb);
        A = 1; B = 1; C = 0; D = 0;
        #100 A = 1; B = 1; C = 1; D = 0;
        #50 A = 1; B = 1; C = 1; D = 1;
        #50 A = 1; B = 0; C = 0; D = 1;
        #100 A = 1; B = 0; C = 1; D = 0;
        #100 A = 0; B = 1; C = 0; D = 0;
        #100 A = 0; B = 1; C = 1; D = 1;
        #100 A = 0; B = 0; C = 0; D = 1;
        #100 A = 0; B = 0; C = 1; D = 0;
        #100 $finish;
    end
endmodule
```

Simulation waveforms