Lab10

實驗主題: Magnitude Comparator, Decoder & Multiplexer

實驗日期:2022/11/07

學號姓名:B103040009 尹信淳

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

Exercise1:

8-bit Magnitude Comparator

Design and verify the 8-bit magnitude comparator composed of two 4-bit magnitude comparators

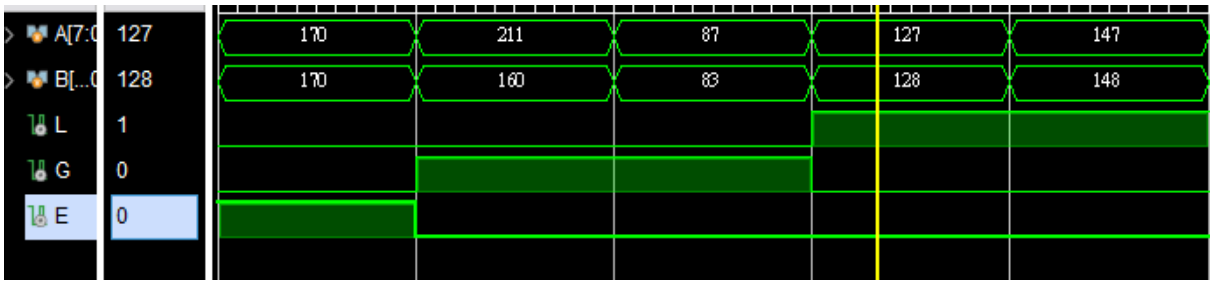
Code:

```
`timescale 1ns / 1ps
module four_bit(A,B,L,G,E);...
module comparator(A,B,L,G,E);
    output L,G,E;
    input[7:0] A,B;
    wire[3:0] A2,A1,B2,B1;
    wire L2,G2,E2,L1,G1,E1;
    wire n1,n2;
    and(A2[3],A[7],1);
    and(A2[2],A[6],1);
    and(A2[1],A[5],1);
    and(A2[0],A[4],1);
    and(A1[3],A[3],1);
    and(A1[2],A[2],1);
    and(A1[1],A[1],1);
    and(A1[0],A[0],1);
    and(B2[3],B[7],1);
    and(B2[2],B[6],1);
    and(B2[1],B[5],1);
    and(B2[0],B[4],1);
    and(B1[3],B[3],1);
    and(B1[2],B[2],1);
    and(B1[1],B[1],1);
    and(B1[0],B[0],1);
    four_bit fb2(A2,B2,L2,G2,E2);
    four_bit fb1(A1,B1,L1,G1,E1);
    and(E,E1,E2);
    and(n1,E2,G1);
    or(G,G2,n1);
    and(n2,E2,L1);
    or(L,L2,n2);
endmodule
```

Testbench:

```
module tb;
reg[7:0] A,B;
wire L,G,E;
comparator UUT(.A(A),.B(B),.L(L),.G(G),.E(E));
initial begin
    A = 8'b10101010;
    B = 8'b10101010;
    #10
    A = 8'b11010011;
    B = 8'b10100000;
    #10
    A = 8'b01010111;
    B = 8'b01010011;
    #10
    A = 8'b01111111;
    B = 8'b10000000;
    #10
    A = 8'b10010011;
    B = 8'b10010100;
    #10 $finish;
    end
endmodule
```

波形圖:



Exercise1 實驗結果與分析: four-bit magnitude comparator 照著 ppt 的圖用 structure level modeling，再把兩個拼接起來就是 8-bit magnitude comparator. 功能正常。

Exercise2:

3-to-8 Decoder

Design and verify a 3-to-8 decoder composed of two 2-to-4 decoders

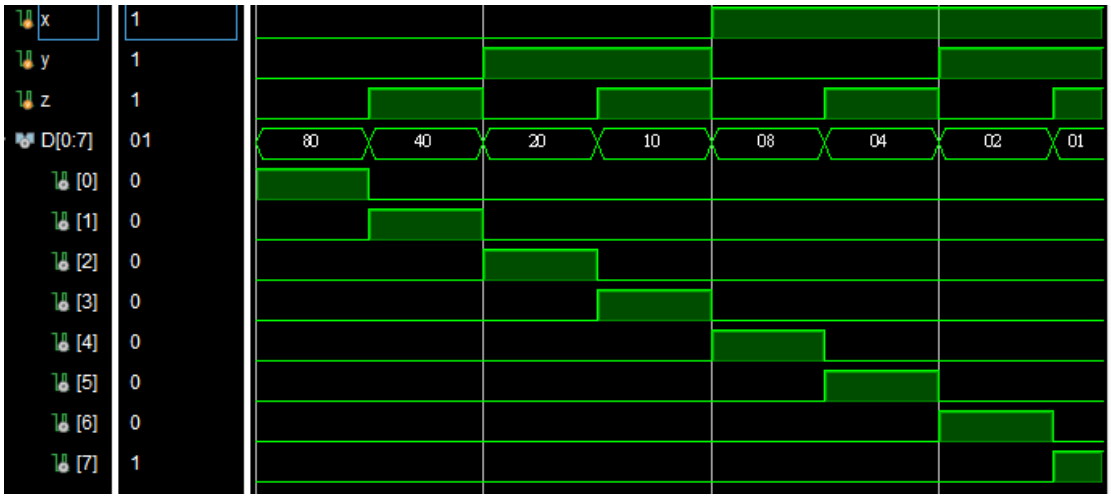
Code:

```
module decoder_2x4_gates (D, A, B, enable);
output [0:3] D;
input A, B;
input enable;
wire A_not, B_not, enable_not;
not
G1 (A_not, A),
G2 (B_not, B),
G3 (enable_not, enable);
and
G4 (D[0], A_not, B_not, enable_not),
G5 (D[1], A_not, B, enable_not),
G6 (D[2], A, B_not, enable_not),
G7 (D[3], A, B, enable_not);
endmodule
module e2(x,y,z,D);
output[7:0] D;
input x,y,z;
wire n1;
not (n1,x);
decoder_2x4_gates d1(D[3:0],y,z,n1);
decoder_2x4_gates d2(D[7:4],y,z,x);
endmodule
```

Testbench:

```
module tb;
reg x,y,z;
wire[0:7] D;
e2 UUT(.x(x),.y(y),.z(z),.D(D));
initial begin
    x=1'b0; y=1'b0;z=1'b0; #10
    x=1'b0; y=1'b0;z=1'b1; #10
    x=1'b0; y=1'b1;z=1'b0; #10
    x=1'b0; y=1'b1;z=1'b1; #10
    x=1'b1; y=1'b0;z=1'b0; #10
    x=1'b1; y=1'b0;z=1'b1; #10
    x=1'b1; y=1'b1;z=1'b0; #10
    x=1'b1; y=1'b1;z=1'b1; #10
    $finish;
    end
endmodule
```

波形圖:



Exercise2 實驗結果與分析:

用兩個預先設計好的 2x4 decoders 拼出 3x8 decoder.

功能正常。

Exercise3:

Boolean Function Implementation

Design and verify the 8-to-1 multiplexer using Verilog HDL (behavioral level modeling)

Implement and verify the following Boolean function of 4 input variable using 8-to-1 MUX F(A, B, C, D) = Σ(1, 2, 5, 8, 9, 10, 12, 13)

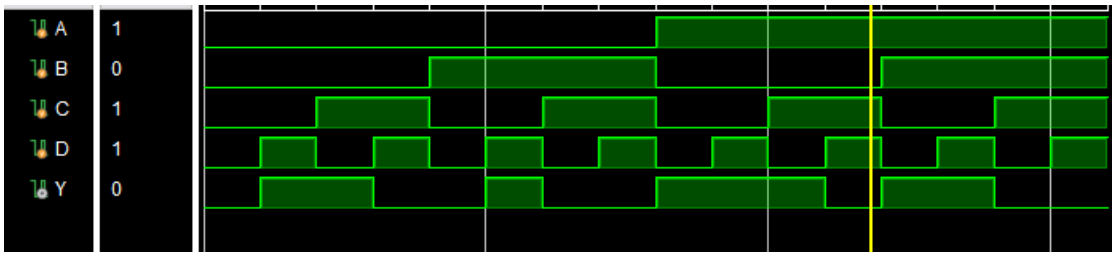
Code:

```
module mux_8x1_beh(A,B,C,D,Y);
output reg Y;
input A,B,C,D;
always @ (A,B,C,D)
case ({A,B,C})
3'b000: Y = D;
3'b001: Y = !D;
3'b010: Y = D;
3'b011: Y = 1'b0;
3'b100: Y = 1'b1;
3'b101: Y = !D;
3'b110: Y = 1'b1;
3'b111: Y = 1'b0;
endcase
endmodule
```

Testbench:

```
module tb();
reg A, B, C, D;
wire Y;
mux_8x1_beh u1(.A(A), .B(B), .C(C), .D(D), .Y(Y));
initial begin
{A, B, C,D} = 4'd0;#10
{A, B, C,D} = 4'd1;#10
{A, B, C,D} = 4'd2;#10
{A, B, C,D} = 4'd3;#10
{A, B, C,D} = 4'd4;#10
{A, B, C,D} = 4'd5;#10
{A, B, C,D} = 4'd6;#10
{A, B, C,D} = 4'd7;#10
{A, B, C,D} = 4'd8;#10
{A, B, C,D} = 4'd9;#10
{A, B, C,D} = 4'd10;#10
{A, B, C,D} = 4'd11;#10
{A, B, C,D} = 4'd12;#10
{A, B, C,D} = 4'd13;#10
{A, B, C,D} = 4'd14;#10
{A, B, C,D} = 4'd15;#10
$finish;
end
endmodule
```

波形圖:



Exercise3 實驗結果與分析:

多工器功能正常，boolean function 順利 implemented.

實驗心得

這次設計的數位電路都很實用，包括 comparator, decoder, multiplexer. 在設計這些電路時，我覺得有一個關鍵是它們都是由較簡單的電路拼成大的電路，我認為這項技巧不只會運用在這裡，將來在設計大型數位系統時，肯定也是由一個個小的稅為電路組成，因此熟悉、靈活運用基本電路的能力很重要，而這次的實驗課就給了我一個機會。老師給了學生適切的練習方向，相信只要持續磨練就能走在正確的道路上有效率地進步。