Lab09

實驗主題:Adders

實驗日期:2022/10/31

學號姓名:B103040009 尹信淳

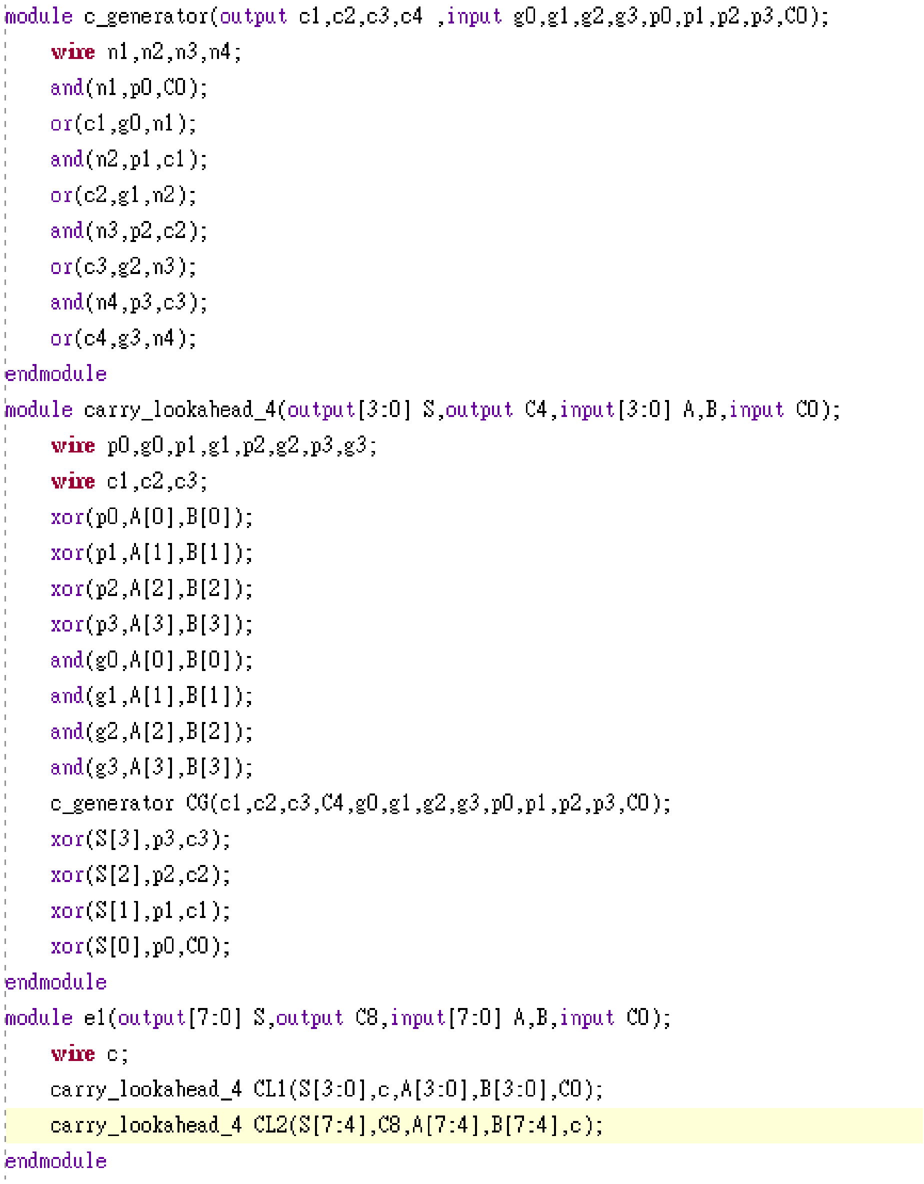
實驗內容: Design and Verify the following circuits using Verilog HDL and Schematic

Exercise1:

8-bit Carry-look Ahead Adder

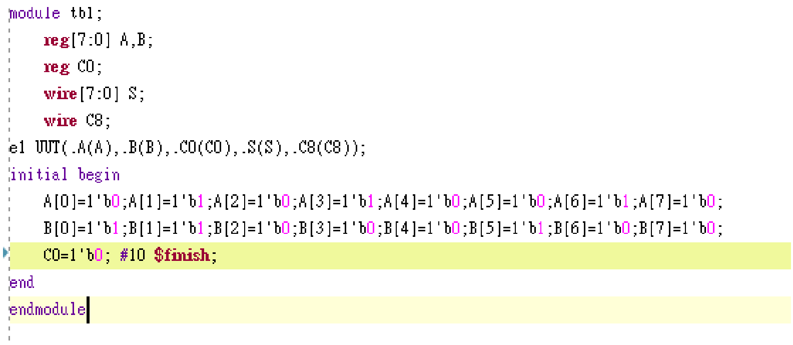
Design and verify the 8-bit carry-look ahead adder composed of two 4-bit carry-look ahead adders (using Verilog Structural level modeling)

Code:



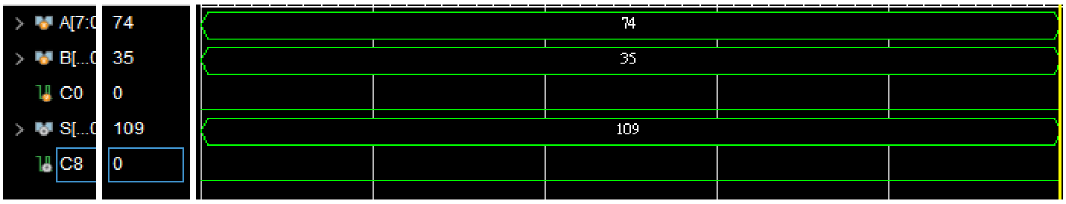
```
module c_generator(output c1,c2,c3,c4 ,input g0,g1,g2,g3,p0,p1,p2,p3,C0);
    wire n1,n2,n3,n4;
    and(n1,p0,C0);
    or(c1,g0,n1);
    and(n2,p1,c1);
    or(c2,g1,n2);
    and(n3,p2,c2);
    or(c3,g2,n3);
    and(n4,p3,c3);
    or(c4,g3,n4);
endmodule
module carry_lookahead_4(output[3:0] S,output C4,input[3:0] A,B,input C0);
    wire p0,g0,p1,g1,p2,g2,p3,g3;
    wire c1,c2,c3;
    xor(p0,A[0],B[0]);
    xor(p1,A[1],B[1]);
    xor(p2,A[2],B[2]);
    xor(p3,A[3],B[3]);
    and(g0,A[0],B[0]);
    and(g1,A[1],B[1]);
    and(g2,A[2],B[2]);
    and(g3,A[3],B[3]);
    c_generator CG(c1,c2,c3,C4,g0,g1,g2,g3,p0,p1,p2,p3,C0);
    xor(S[3],p3,c3);
    xor(S[2],p2,c2);
    xor(S[1],p1,c1);
    xor(S[0],p0,C0);
endmodule
module e1(output[7:0] S,output C8,input[7:0] A,B,input C0);
    wire c;
    carry_lookahead_4 CL1(S[3:0],c,A[3:0],B[3:0],C0);
    carry_lookahead_4 CL2(S[7:4],C8,A[7:4],B[7:4],c);
endmodule
```

Testbench:



```
module tb1;
    reg[7:0] A,B;
    reg C0;
    wire[7:0] S;
    wire C8;
e1 UUT(.A(A),.B(B),.C0(C0),.S(S),.C8(C8));
initial begin
    A[0]=1'b0;A[1]=1'b1;A[2]=1'b0;A[3]=1'b1;A[4]=1'b0;A[5]=1'b0;A[6]=1'b1;A[7]=1'b0;
    B[0]=1'b1;B[1]=1'b1;B[2]=1'b0;B[3]=1'b0;B[4]=1'b0;B[5]=1'b1;B[6]=1'b0;B[7]=1'b0;
    C0=1'b0; #10 $finish;
end
endmodule
```

波形圖:



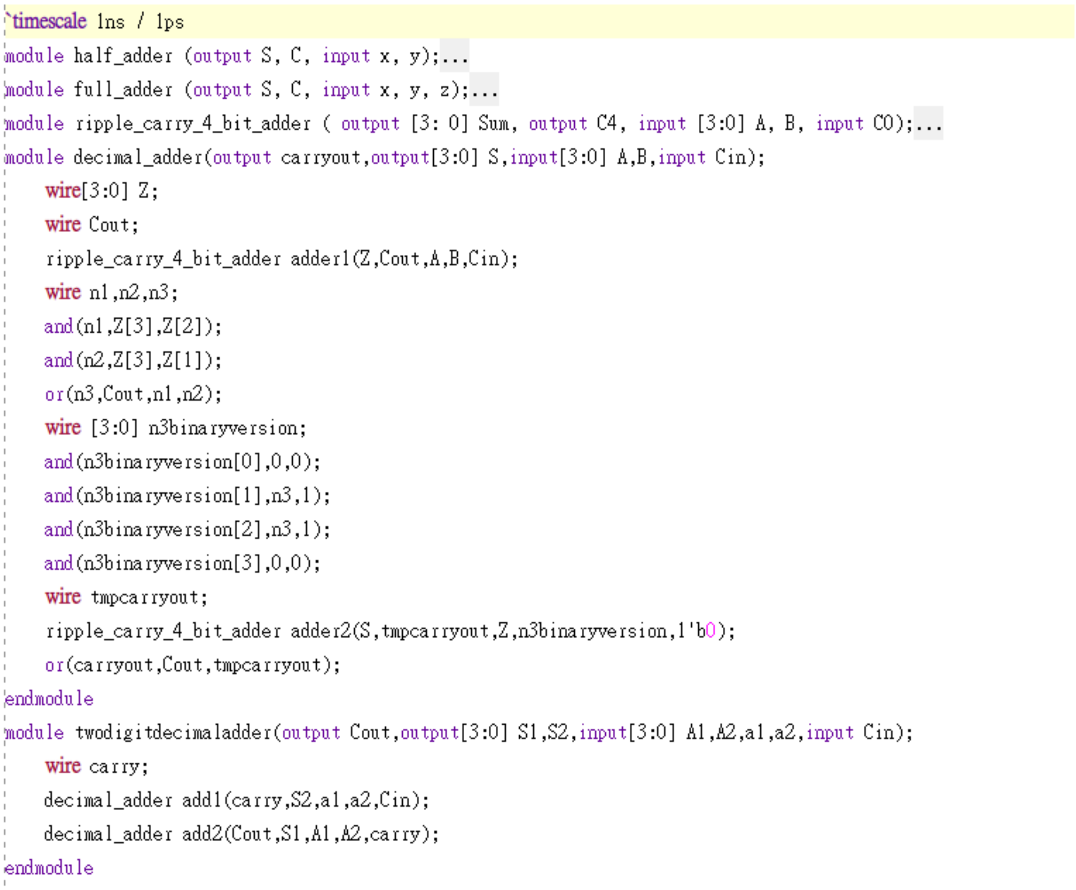
Exercise1 實驗結果與分析: Adder 功能正常。

Exercise2:

Decimal Adder

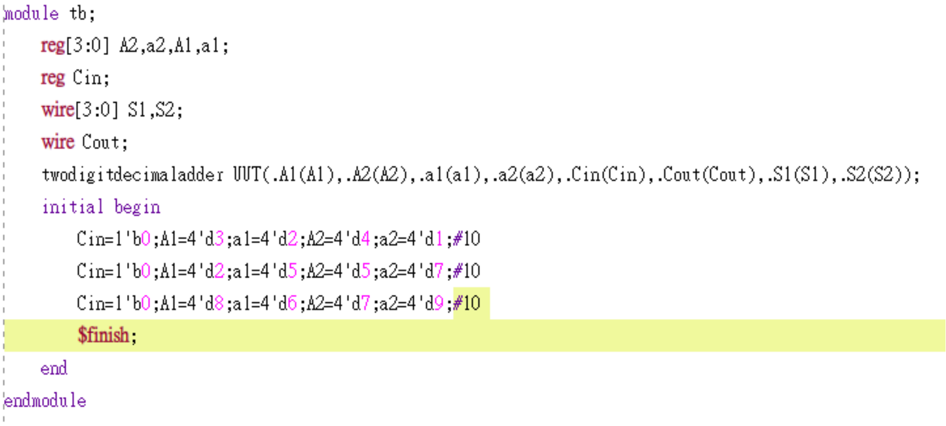
Design and verify the 2-digit decimal adder

Code:



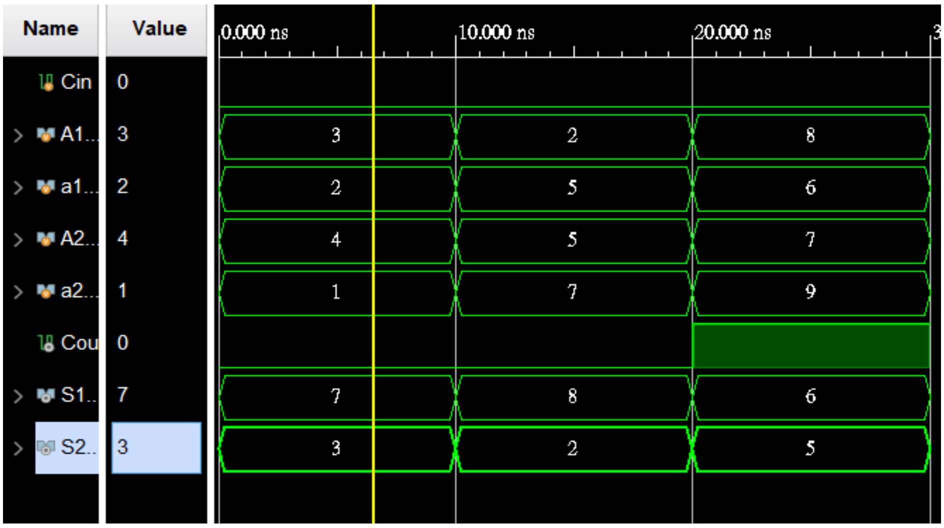
```
`timescale 1ns / 1ps
module half_adder (output S, C, input x, y);...
module full_adder (output S, C, input x, y, z);...
module ripple_carry_4_bit_adder ( output [3: 0] Sum, output C4, input [3:0] A, B, input C0);...
module decimal_adder(output carryout,output[3:0] S,input[3:0] A,B,input Cin);
    wire[3:0] Z;
    wire Cout;
    ripple_carry_4_bit_adder adder1(Z,Cout,A,B,Cin);
    wire n1,n2,n3;
    and(n1,Z[3],Z[2]);
    and(n2,Z[3],Z[1]);
    or(n3,Cout,n1,n2);
    wire [3:0] n3binaryversion;
    and(n3binaryversion[0],0,0);
    and(n3binaryversion[1],n3,1);
    and(n3binaryversion[2],n3,1);
    and(n3binaryversion[3],0,0);
    wire tmpcarryout;
    ripple_carry_4_bit_adder adder2(S,tmpcarryout,Z,n3binaryversion,1'b0);
    or(carryout,Cout,tmpcarryout);
endmodule
module twodigitdecimaladder(output Cout,output[3:0] S1,S2,input[3:0] A1,A2,a1,a2,input Cin);
    wire carry;
    decimal_adder add1(carry,S2,a1,a2,Cin);
    decimal_adder add2(Cout,S1,A1,A2,carry);
endmodule
```

Testbench:



```
module tb;
    reg[3:0] A2,a2,A1,a1;
    reg Cin;
    wire[3:0] S1,S2;
    wire Cout;
    twodigitdecimaladder UUT(.A1(A1),.A2(A2),.a1(a1),.a2(a2),.Cin(Cin),.Cout(Cout),.S1(S1),.S2(S2));
    initial begin
        Cin=1'b0;A1=4'd3;a1=4'd2;A2=4'd4;a2=4'd1;#10
        Cin=1'b0;A1=4'd2;a1=4'd5;A2=4'd5;a2=4'd7;#10
        Cin=1'b0;A1=4'd8;a1=4'd6;A2=4'd7;a2=4'd9;#10
        $finish;
    end
endmodule
```

波型:



Exercise2 實驗結果與分析:

decimal adder 功能正常。

實驗心得

這次實驗有兩題，上課實作時因為第一題的 testbench 忘記加#10，沒給他時間處理輸入就結束，因此輸入都有進去，但是輸出都是 unknown，在這邊浪費了不少時間。因此第二題我是回家才完成的。這兩題的關鍵我覺得是要懂得靈活運用之前學的簡單電路，由這些簡單的小電路拼出較複雜的大電路。因此除了對於小電路的輸入、輸出以及運作原理要足夠熟悉之外，對於大電路的設計、樣貌也要足夠了解。如此一來，這兩題實作就會比較清楚明瞭。