

# Lab 5 report

---

## Contents

<b>“Check your understanding”</b>	<b>1</b>
<b>Simulation Results (8 bit)</b>	<b>1</b>
<b>Test Results (5 bit)</b>	<b>4</b>
<b>Pin Assignments</b>	<b>7</b>
<b>Verilog Code</b>	<b>10</b>

---

Note : if you are to build the Quartus project there is a “N” macro defined at the top of the “lab5.sv” file, changing that changes the number of bits handled. E.g : “`define N 8” for 8 bit.

## “Check your understanding”

**A :** The circuit uses the fact that taking the 2’s complement of a number gets its negative value to find the difference, then because the highest carryout indicates A is less than B and so the result is negative, another 2’s comp conversion can be done to flip the sign of the negative result.

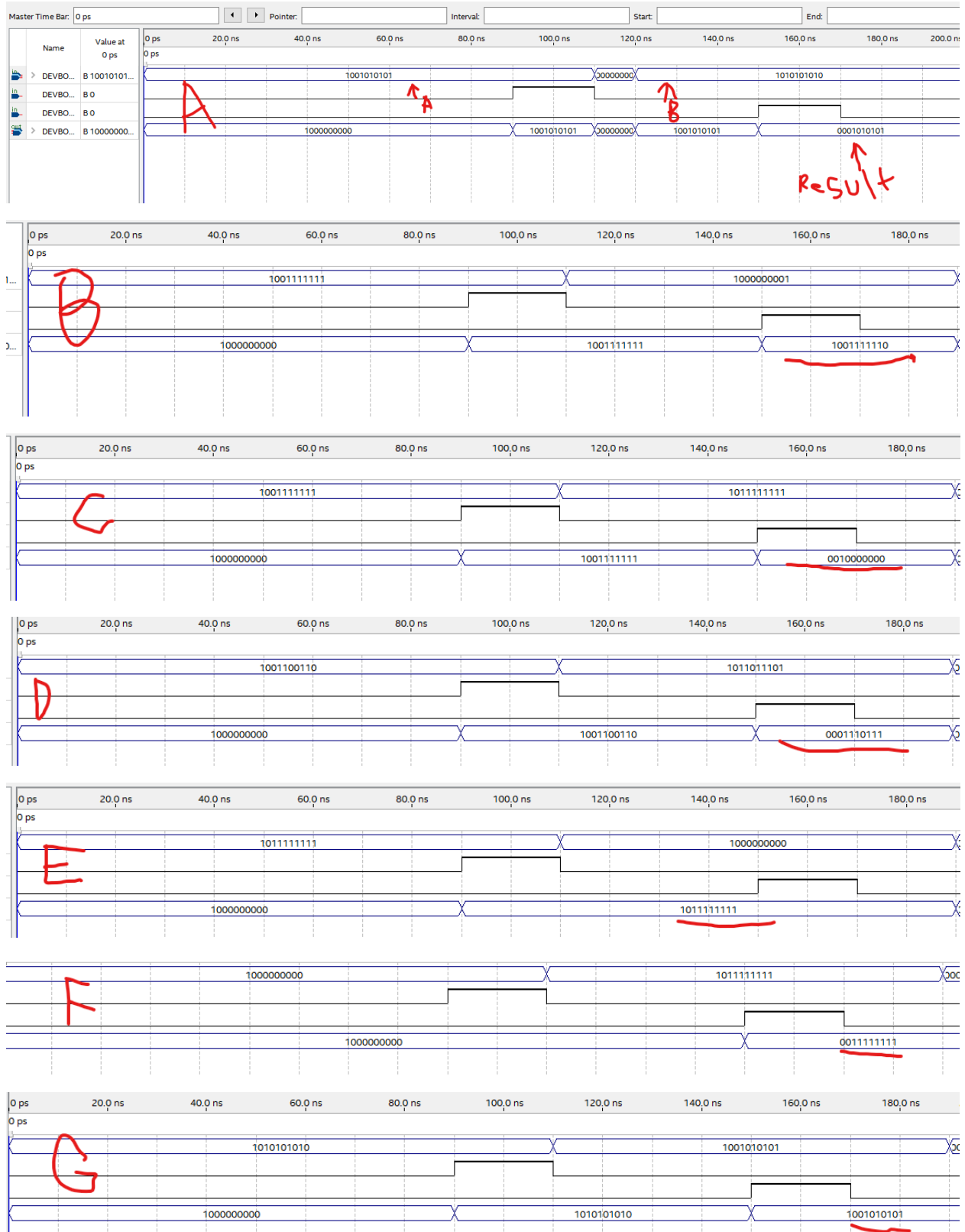
**B :** cout high indicates value A was less than value B. since the carries are effectively ‘borrows’ so it's just the same as having an extra ‘borrow’. cout low indicates A is greater than B since it was able to satisfy all the ‘borrows’.

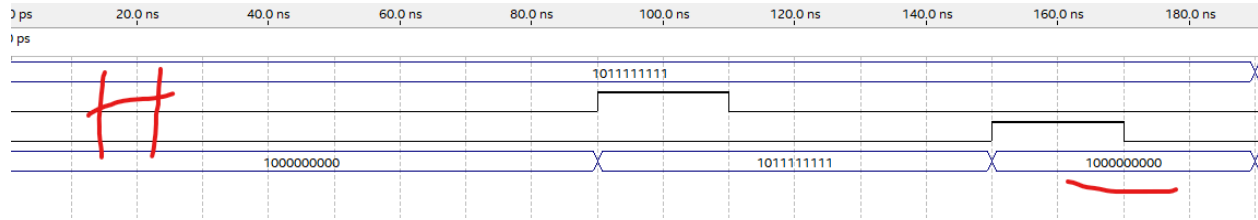
## Simulation Results (8 bit)

context: A & B values are input from switches, loaded on button press. Switch 9 is held high to show results, but is not part of the input value. Red LED9 indicates Cout.

	A	B	$R =  A - B $	Cout
a	01010101	10101010	01010101	0
b	01111111	00000001	01111110	1
c	01111111	11111111	10000000	0
d	01100110	11011101	01110111	0

e	11111111	00000000	11111111	1
f	00000000	11111111	11111111	0
g	10101010	01010101	01010101	1
h	11111111	11111111	00000000	1

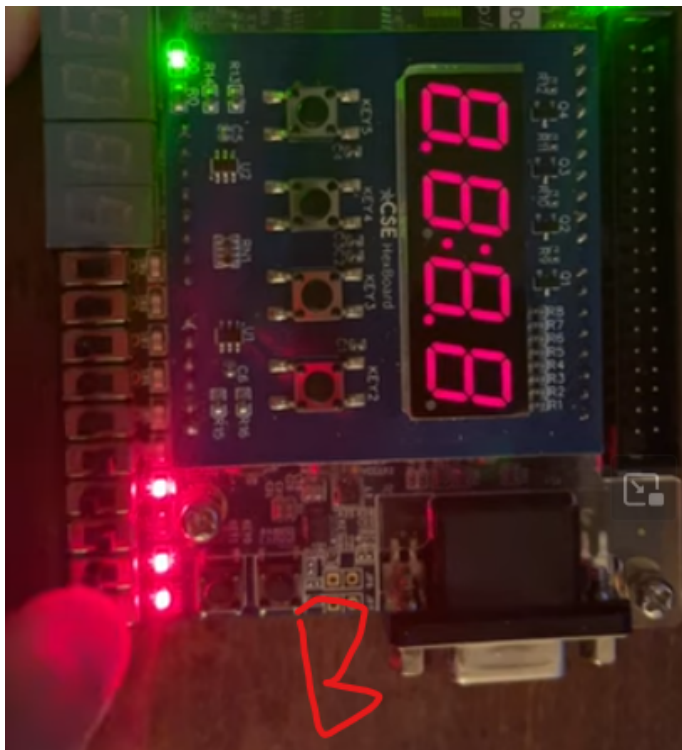
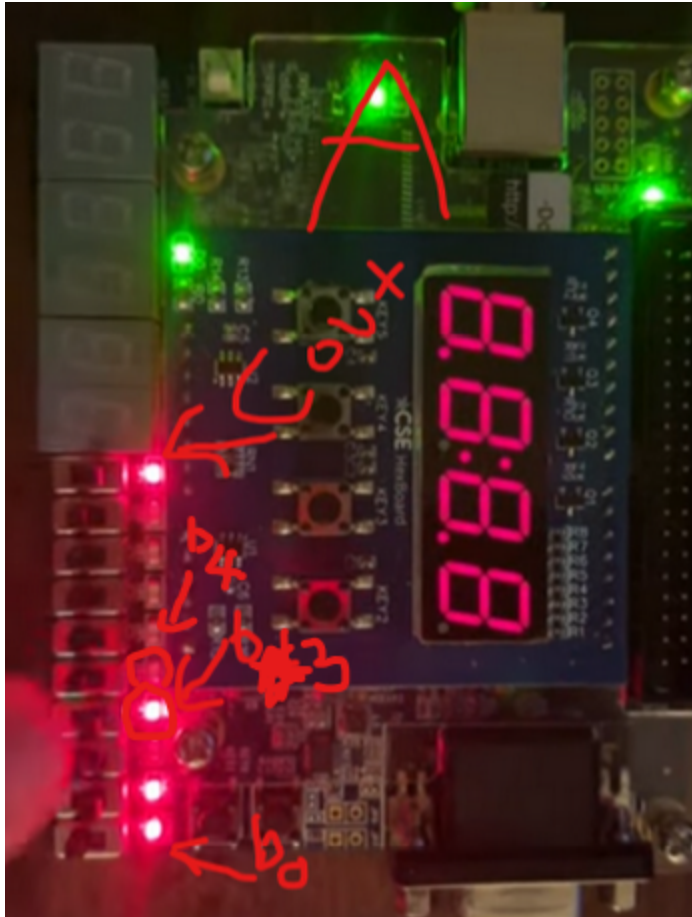


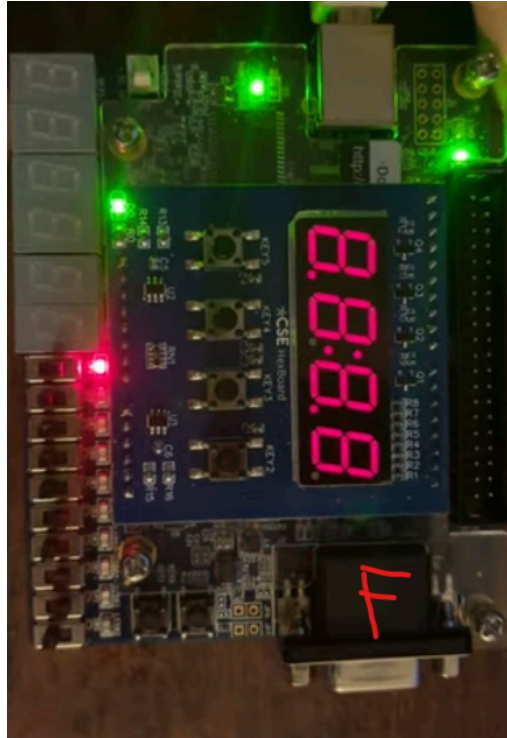
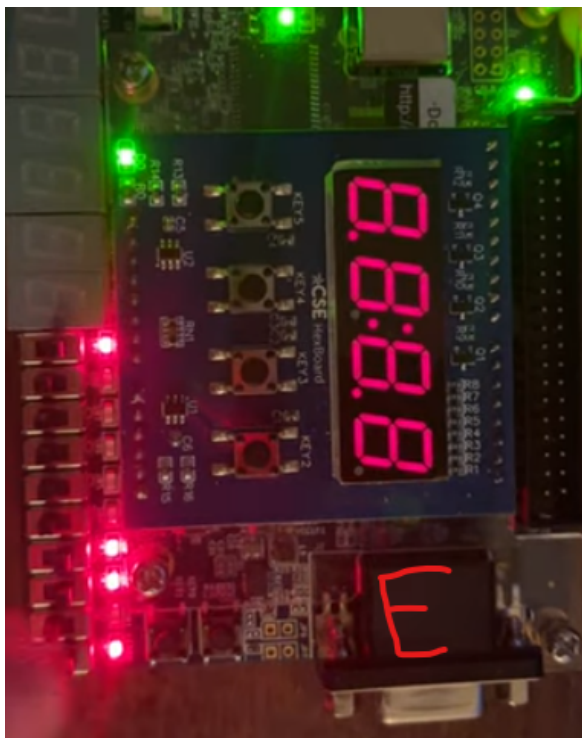
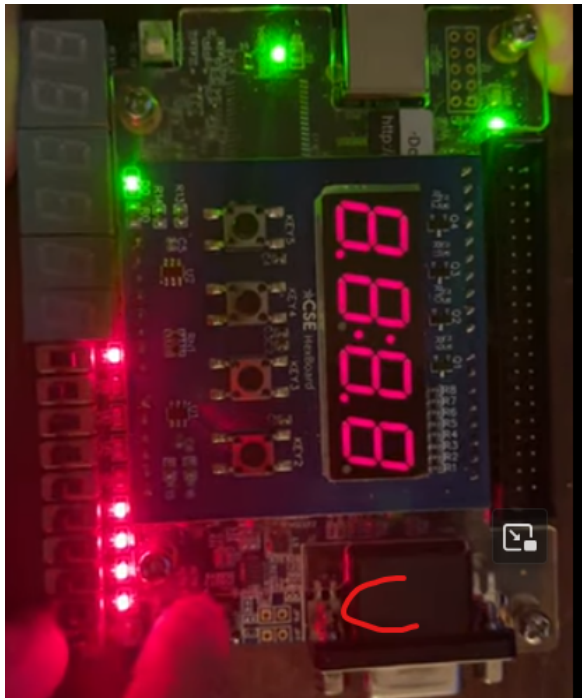


## Test Results (5 bit)

5-bit results recording : <https://youtu.be/OHLeVnUSl3M>

	A	B	$R =  A - B $	cout
a	10101	01010	01011	1
b	01010	10101	01011	0
c	11111	00001	11110	1
d	00001	11111	11110	0
e	10110	00000	01101	1
f	10110	10110	00000	1





## Pin Assignments




































effective assignments (from top level of lab module)

```
assign val_in      = DEVBOARD_SWS[0+:`N];  
assign ina         = DEVBOARD_BTN[0];  
assign inb         = DEVBOARD_BTN[1];  
assign showAelseB  = DEVBOARD_SWS[8];  
assign showResult  = DEVBOARD_SWS[9];  
  
assign DEVBOARD_RLEDs[9] = cout;  
  
//blank the seg displays  
assign DEVBOARD_SEGS = ~0;
```

Actual

	tatu	From	To	Assignment Name	Value	Enabled	Entity	Co
1	✓		in DEVB...N[1]	Location	PIN_A7	Yes		
2	✓		in DEVB...N[0]	Location	PIN_B8	Yes		
3	✓		out DEVB...S[1]	Location	PIN_A9	Yes		
4	✓		out DEVB...S[2]	Location	PIN_A10	Yes		
5	✓		out DEVB...S[3]	Location	PIN_B10	Yes		
6	✓		out DEVB...S[4]	Location	PIN_D13	Yes		
7	✓		out DEVB...S[5]	Location	PIN_C13	Yes		
8	✓		out DEVB...S[6]	Location	PIN_E14	Yes		
9	✓		out DEVB...S[7]	Location	PIN_D14	Yes		
10	✓		out DEVB...S[8]	Location	PIN_A11	Yes		
11	✓		out DEVB...S[9]	Location	PIN_B11	Yes		
12	✓		out DEVB...S[0]	Location	PIN_A8	Yes		
13	✓		in DEVB...S[1]	Location	PIN_C11	Yes		
14	✓		in DEVB...S[2]	Location	PIN_D12	Yes		
15	✓		in DEVB...S[3]	Location	PIN_C12	Yes		
16	✓		in DEVB...S[4]	Location	PIN_A12	Yes		
17	✓		in DEVB...S[5]	Location	PIN_B12	Yes		
18	✓		in DEVB...S[6]	Location	PIN_A13	Yes		
19	✓		in DEVB...S[7]	Location	PIN_A14	Yes		
20	✓		in DEVB...S[8]	Location	PIN_B14	Yes		
21	✓		in DEVB...S[9]	Location	PIN_F15	Yes		
22	✓		in DEVB...S[0]	Location	PIN_C10	Yes		
23	✓		out DEVB...S[1]	Location	PIN_E15	Yes		
24	✓		out DEVB...S[2]	Location	PIN_C15	Yes		
25	✓		out DEVB...S[3]	Location	PIN_C16	Yes		
26	✓		out DEVB...S[4]	Location	PIN_E16	Yes		
27	✓		out DEVB...S[5]	Location	PIN_D17	Yes		
28	✓		out DEVB...S[6]	Location	PIN_C17	Yes		
29	✓		out DEVB...S[7]	Location	PIN_D15	Yes		
30	✓		out DEVB...S[8]	Location	PIN_C18	Yes		
31	✓		out DEVB...S[9]	Location	PIN_D18	Yes		
32	✓		out DEVB...[10]	Location	PIN_E18	Yes		
33	✓		out DEVB...[11]	Location	PIN_B16	Yes		
34	✓		out DEVB...[12]	Location	PIN_A17	Yes		
35	✓		out DEVB...[13]	Location	PIN_A18	Yes		
36	✓		out DEVB...[14]	Location	PIN_B17	Yes		



	tatu	From	To	Assignment Name	Value	Enabled	E
36	✓		 DEVB...[14]	Location	PIN_B17	Yes	
37	✓		 DEVB...[15]	Location	PIN_A16	Yes	
38	✓		 DEVB...[16]	Location	PIN_B20	Yes	
39	✓		 DEVB...[17]	Location	PIN_A20	Yes	
40	✓		 DEVB...[18]	Location	PIN_B19	Yes	
41	✓		 DEVB...[19]	Location	PIN_A21	Yes	
42	✓		 DEVB...[20]	Location	PIN_B21	Yes	
43	✓		 DEVB...[21]	Location	PIN_C22	Yes	
44	✓		 DEVB...[22]	Location	PIN_B22	Yes	
45	✓		 DEVB...[23]	Location	PIN_A19	Yes	
46	✓		 DEVB...[24]	Location	PIN_F21	Yes	
47	✓		 DEVB...[25]	Location	PIN_E22	Yes	
48	✓		 DEVB...[26]	Location	PIN_E21	Yes	
49	✓		 DEVB...[27]	Location	PIN_C19	Yes	
50	✓		 DEVB...[28]	Location	PIN_C20	Yes	
51	✓		 DEVB...[29]	Location	PIN_D19	Yes	
52	✓		 DEVB...[30]	Location	PIN_E17	Yes	
53	✓		 DEVB...[31]	Location	PIN_D22	Yes	
54	✓		 DEVB...[32]	Location	PIN_F18	Yes	
55	✓		 DEVB...[33]	Location	PIN_E20	Yes	
56	✓		 DEVB...[34]	Location	PIN_E19	Yes	
57	✓		 DEVB...[35]	Location	PIN_J18	Yes	
58	✓		 DEVB...[36]	Location	PIN_H19	Yes	
59	✓		 DEVB...[37]	Location	PIN_F19	Yes	
60	✓		 DEVB...[38]	Location	PIN_F20	Yes	
61	✓		 DEVB...[39]	Location	PIN_F17	Yes	
62	✓		 DEVB...[40]	Location	PIN_J20	Yes	
63	✓		 DEVB...[41]	Location	PIN_K20	Yes	
64	✓		 DEVB...[42]	Location	PIN_L18	Yes	
65	✓		 DEVB...[43]	Location	PIN_N18	Yes	
66	✓		 DEVB...[44]	Location	PIN_M20	Yes	
67	✓		 DEVB...[45]	Location	PIN_N19	Yes	
68	✓		 DEVB...[46]	Location	PIN_N20	Yes	
69	✓		 DEVB...[47]	Location	PIN_L19	Yes	
70	✓		 DEVB...S[0]	Location	PIN_C14	Yes	
71		<<new>>	<<new>>	<<new>>			

## Verilog Code

Top level lab

```
/**
    cse3341, digital logic 2, lab 5
    George Boone
    1002055713

    modular exponent subtractor
    this module is not part of the logic, it only serves to demo it
*/

`define N 8

module lab5(
    // standard in. i got tired of the assignment editor, see immediately after wire declarations for
    assignments
    input [9:0] DEVBOARD_SWS,
    input [1:0] DEVBOARD_BTN,
    output[9:0] DEVBOARD_RLEDS,
    output[47:0]DEVBOARD_SEGS
);

//-----
//      INPUT ASSIGNMENTS \ EXTERNAL WIRING
//-----

    wire [`N-1:0] val_in;
    wire ina, inb;
    wire cout;
    reg showAelseB, showResult;    //yeah

    assign val_in    = DEVBOARD_SWS[0+:`N];
    assign ina       = DEVBOARD_BTN[0];
    assign inb       = DEVBOARD_BTN[1];
    assign showAelseB = DEVBOARD_SWS[8];
    assign showResult = DEVBOARD_SWS[9];

    assign DEVBOARD_RLEDS[9]      = cout;

    //blank the seg displays
    assign DEVBOARD_SEGS = ~0;

//-----
//      INTERNAL WIRING
//-----
```

```
    reg [`N:0] valA, valB, diff;

    always @ (posedge ina)
    valA = val_in;

    always @ (posedge inb)
    valB = val_in;

    always @ (showAelseB, showResult)
    if(showResult == 1)
    DEVBOARD_RLEDS[0+:`N] = diff;
    else if(showAelseB == 1)
    DEVBOARD_RLEDS[0+:`N] = valA;
    else
    DEVBOARD_RLEDS[0+:`N] = valB;

//-----
//    MODULES
//-----

//there is a bug where valA and/or valB are updated when flicking around the switches.
// i

//works
//    RCTractor(
//    .A(valA),
//    .B(valB),
//
//    .D(diff),
//    .COUT(cout)
//    );

//works
//    BintTwosComp(
//    .BIN(valA),
//
//    .TWOSCOMP(diff),
//    .COUT(cout)
//    );

    ExponentSubtractor#(
    .N(`N)
    )(
    .EA(valA),
    .EB(valB),

    .Cout(cout),
    .ED(diff)
    );
```

```
endmodule
```

```
Top level exponent subtractor
```

```
/**  
cse3341, digital logic 2, lab 5  
George Boone  
1002055713  
  
hardcoded exponent subtractor according to figure 1 of lab document  
*/
```

```
module ExponentSubtractor#(  
    parameter N = 8  
)(  
    input [N-1:0] EA, EB,  
    output Cout,  
    output [N-1:0] ED  
);  
  
    wire [N-1:0] d, invd;    // difference, inverted-difference  
  
    RCTSubtractor #(  
        .N(N)  
    ) _rcs    (  
        .A(EA),  
        .B(EB),  
  
        .D(d),  
        .COUT(Cout)  
    );  
  
    Mux2t1 #(  
        .N(N)  
    ) _m2t1    (  
        .CTRL(Cout),  
        .A(invd),  
        .B(d),  
  
        .OUT(ED)  
    );  
  
    BintTwosComp #(  
        .N(N)  
    ) _btcc    (  
        .A(ED),  
        .B(ED),  
        .C(ED),  
        .D(ED),  
        .E(ED),  
        .F(ED),  
        .G(ED),  
        .H(ED),  
        .I(ED),  
        .J(ED),  
        .K(ED),  
        .L(ED),  
        .M(ED),  
        .N(ED),  
        .O(ED),  
        .P(ED),  
        .Q(ED),  
        .R(ED),  
        .S(ED),  
        .T(ED),  
        .U(ED),  
        .V(ED),  
        .W(ED),  
        .X(ED),  
        .Y(ED),  
        .Z(ED),  
        .AA(ED),  
        .AB(ED),  
        .AC(ED),  
        .AD(ED),  
        .AE(ED),  
        .AF(ED),  
        .AG(ED),  
        .AH(ED),  
        .AI(ED),  
        .AJ(ED),  
        .AK(ED),  
        .AL(ED),  
        .AM(ED),  
        .AN(ED),  
        .AO(ED),  
        .AP(ED),  
        .AQ(ED),  
        .AR(ED),  
        .AS(ED),  
        .AT(ED),  
        .AU(ED),  
        .AV(ED),  
        .AW(ED),  
        .AX(ED),  
        .AY(ED),  
        .AZ(ED),  
        .BA(ED),  
        .BB(ED),  
        .BC(ED),  
        .BD(ED),  
        .BE(ED),  
        .BF(ED),  
        .BG(ED),  
        .BH(ED),  
        .BI(ED),  
        .BJ(ED),  
        .BK(ED),  
        .BL(ED),  
        .BM(ED),  
        .BN(ED),  
        .BO(ED),  
        .BP(ED),  
        .BQ(ED),  
        .BR(ED),  
        .BS(ED),  
        .BT(ED),  
        .BU(ED),  
        .BV(ED),  
        .BW(ED),  
        .BX(ED),  
        .BY(ED),  
        .BZ(ED),  
        .CA(ED),  
        .CB(ED),  
        .CC(ED),  
        .CD(ED),  
        .CE(ED),  
        .CF(ED),  
        .CG(ED),  
        .CH(ED),  
        .CI(ED),  
        .CJ(ED),  
        .CK(ED),  
        .CL(ED),  
        .CM(ED),  
        .CN(ED),  
        .CO(ED),  
        .CP(ED),  
        .CQ(ED),  
        .CR(ED),  
        .CS(ED),  
        .CT(ED),  
        .CU(ED),  
        .CV(ED),  
        .CW(ED),  
        .CX(ED),  
        .CY(ED),  
        .CZ(ED),  
        .DA(ED),  
        .DB(ED),  
        .DC(ED),  
        .DD(ED),  
        .DE(ED),  
        .DF(ED),  
        .DG(ED),  
        .DH(ED),  
        .DI(ED),  
        .DJ(ED),  
        .DK(ED),  
        .DL(ED),  
        .DM(ED),  
        .DN(ED),  
        .DO(ED),  
        .DP(ED),  
        .DQ(ED),  
        .DR(ED),  
        .DS(ED),  
        .DT(ED),  
        .DU(ED),  
        .DV(ED),  
        .DW(ED),  
        .DX(ED),  
        .DY(ED),  
        .DZ(ED),  
        .EA(ED),  
        .EB(ED),  
        .EC(ED),  
        .ED(ED),  
        .EE(ED),  
        .EF(ED),  
        .EG(ED),  
        .EH(ED),  
        .EI(ED),  
        .EJ(ED),  
        .EK(ED),  
        .EL(ED),  
        .EM(ED),  
        .EN(ED),  
        .EO(ED),  
        .EP(ED),  
        .EQ(ED),  
        .ER(ED),  
        .ES(ED),  
        .ET(ED),  
        .EU(ED),  
        .EV(ED),  
        .EW(ED),  
        .EX(ED),  
        .EY(ED),  
        .EZ(ED),  
        .FA(ED),  
        .FB(ED),  
        .FC(ED),  
        .FD(ED),  
        .FE(ED),  
        .FF(ED),  
        .FG(ED),  
        .FH(ED),  
        .FI(ED),  
        .FJ(ED),  
        .FK(ED),  
        .FL(ED),  
        .FM(ED),  
        .FN(ED),  
        .FO(ED),  
        .FP(ED),  
        .FQ(ED),  
        .FR(ED),  
        .FS(ED),  
        .FT(ED),  
        .FU(ED),  
        .FV(ED),  
        .FW(ED),  
        .FX(ED),  
        .FY(ED),  
        .FZ(ED),  
        .GA(ED),  
        .GB(ED),  
        .GC(ED),  
        .GD(ED),  
        .GE(ED),  
        .GF(ED),  
        .GG(ED),  
        .GH(ED),  
        .GI(ED),  
        .GJ(ED),  
        .GK(ED),  
        .GL(ED),  
        .GM(ED),  
        .GN(ED),  
        .GO(ED),  
        .GP(ED),  
        .GQ(ED),  
        .GR(ED),  
        .GS(ED),  
        .GT(ED),  
        .GU(ED),  
        .GV(ED),  
        .GW(ED),  
        .GX(ED),  
        .GY(ED),  
        .GZ(ED),  
        .HA(ED),  
        .HB(ED),  
        .HC(ED),  
        .HD(ED),  
        .HE(ED),  
        .HF(ED),  
        .HG(ED),  
        .HH(ED),  
        .HI(ED),  
        .HJ(ED),  
        .HK(ED),  
        .HL(ED),  
        .HM(ED),  
        .HN(ED),  
        .HO(ED),  
        .HP(ED),  
        .HQ(ED),  
        .HR(ED),  
        .HS(ED),  
        .HT(ED),  
        .HU(ED),  
        .HV(ED),  
        .HW(ED),  
        .HX(ED),  
        .HY(ED),  
        .HZ(ED),  
        .IA(ED),  
        .IB(ED),  
        .IC(ED),  
        .ID(ED),  
        .IE(ED),  
        .IF(ED),  
        .IG(ED),  
        .IH(ED),  
        .II(ED),  
        .IJ(ED),  
        .IK(ED),  
        .IL(ED),  
        .IM(ED),  
        .IN(ED),  
        .IO(ED),  
        .IP(ED),  
        .IQ(ED),  
        .IR(ED),  
        .IS(ED),  
        .IT(ED),  
        .IU(ED),  
        .IV(ED),  
        .IW(ED),  
        .IX(ED),  
        .IY(ED),  
        .IZ(ED),  
        .JA(ED),  
        .JB(ED),  
        .JC(ED),  
        .JD(ED),  
        .JE(ED),  
        .JF(ED),  
        .JG(ED),  
        .JH(ED),  
        .JI(ED),  
        .JJ(ED),  
        .JK(ED),  
        .JL(ED),  
        .JM(ED),  
        .JN(ED),  
        .JO(ED),  
        .JP(ED),  
        .JQ(ED),  
        .JR(ED),  
        .JS(ED),  
        .JT(ED),  
        .JU(ED),  
        .JV(ED),  
        .JW(ED),  
        .JX(ED),  
        .JY(ED),  
        .JZ(ED),  
        .KA(ED),  
        .KB(ED),  
        .KC(ED),  
        .KD(ED),  
        .KE(ED),  
        .KF(ED),  
        .KG(ED),  
        .KH(ED),  
        .KI(ED),  
        .KJ(ED),  
        .KK(ED),  
        .KL(ED),  
        .KM(ED),  
        .KN(ED),  
        .KO(ED),  
        .KP(ED),  
        .KQ(ED),  
        .KR(ED),  
        .KS(ED),  
        .KT(ED),  
        .KU(ED),  
        .KV(ED),  
        .KW(ED),  
        .KX(ED),  
        .KY(ED),  
        .KZ(ED),  
        .LA(ED),  
        .LB(ED),  
        .LC(ED),  
        .LD(ED),  
        .LE(ED),  
        .LF(ED),  
        .LG(ED),  
        .LH(ED),  
        .LI(ED),  
        .LJ(ED),  
        .LK(ED),  
        .LL(ED),  
        .LM(ED),  
        .LN(ED),  
        .LO(ED),  
        .LP(ED),  
        .LQ(ED),  
        .LR(ED),  
        .LS(ED),  
        .LT(ED),  
        .LU(ED),  
        .LV(ED),  
        .LW(ED),  
        .LX(ED),  
        .LY(ED),  
        .LZ(ED),  
        .MA(ED),  
        .MB(ED),  
        .MC(ED),  
        .MD(ED),  
        .ME(ED),  
        .MF(ED),  
        .MG(ED),  
        .MH(ED),  
        .MI(ED),  
        .MJ(ED),  
        .MK(ED),  
        .ML(ED),  
        .MM(ED),  
        .MN(ED),  
        .MO(ED),  
        .MP(ED),  
        .MQ(ED),  
        .MR(ED),  
        .MS(ED),  
        .MT(ED),  
        .MU(ED),  
        .MV(ED),  
        .MW(ED),  
        .MX(ED),  
        .MY(ED),  
        .MZ(ED),  
        .NA(ED),  
        .NB(ED),  
        .NC(ED),  
        .ND(ED),  
        .NE(ED),  
        .NF(ED),  
        .NG(ED),  
        .NH(ED),  
        .NI(ED),  
        .NJ(ED),  
        .NK(ED),  
        .NL(ED),  
        .NM(ED),  
        .NN(ED),  
        .NO(ED),  
        .NP(ED),  
        .NQ(ED),  
        .NR(ED),  
        .NS(ED),  
        .NT(ED),  
        .NU(ED),  
        .NV(ED),  
        .NW(ED),  
        .NX(ED),  
        .NY(ED),  
        .NZ(ED),  
        .OA(ED),  
        .OB(ED),  
        .OC(ED),  
        .OD(ED),  
        .OE(ED),  
        .OF(ED),  
        .OG(ED),  
        .OH(ED),  
        .OI(ED),  
        .OJ(ED),  
        .OK(ED),  
        .OL(ED),  
        .OM(ED),  
        .ON(ED),  
        .OO(ED),  
        .OP(ED),  
        .OQ(ED),  
        .OR(ED),  
        .OS(ED),  
        .OT(ED),  
        .OU(ED),  
        .OV(ED),  
        .OW(ED),  
        .OX(ED),  
        .OY(ED),  
        .OZ(ED),  
        .PA(ED),  
        .PB(ED),  
        .PC(ED),  
        .PD(ED),  
        .PE(ED),  
        .PF(ED),  
        .PG(ED),  
        .PH(ED),  
        .PI(ED),  
        .PJ(ED),  
        .PK(ED),  
        .PL(ED),  
        .PM(ED),  
        .PN(ED),  
        .PO(ED),  
        .PP(ED),  
        .PQ(ED),  
        .PR(ED),  
        .PS(ED),  
        .PT(ED),  
        .PU(ED),  
        .PV(ED),  
        .PW(ED),  
        .PX(ED),  
        .PY(ED),  
        .PZ(ED),  
        .QA(ED),  
        .QB(ED),  
        .QC(ED),  
        .QD(ED),  
        .QE(ED),  
        .QF(ED),  
        .QG(ED),  
        .QH(ED),  
        .QI(ED),  
        .QJ(ED),  
        .QK(ED),  
        .QL(ED),  
        .QM(ED),  
        .QN(ED),  
        .QO(ED),  
        .QP(ED),  
        .QQ(ED),  
        .QR(ED),  
        .QS(ED),  
        .QT(ED),  
        .QU(ED),  
        .QV(ED),  
        .QW(ED),  
        .QX(ED),  
        .QY(ED),  
        .QZ(ED),  
        .RA(ED),  
        .RB(ED),  
        .RC(ED),  
        .RD(ED),  
        .RE(ED),  
        .RF(ED),  
        .RG(ED),  
        .RH(ED),  
        .RI(ED),  
        .RJ(ED),  
        .RK(ED),  
        .RL(ED),  
        .RM(ED),  
        .RN(ED),  
        .RO(ED),  
        .RP(ED),  
        .RQ(ED),  
        .RR(ED),  
        .RS(ED),  
        .RT(ED),  
        .RU(ED),  
        .RV(ED),  
        .RW(ED),  
        .RX(ED),  
        .RY(ED),  
        .RZ(ED),  
        .SA(ED),  
        .SB(ED),  
        .SC(ED),  
        .SD(ED),  
        .SE(ED),  
        .SF(ED),  
        .SG(ED),  
        .SH(ED),  
        .SI(ED),  
        .SJ(ED),  
        .SK(ED),  
        .SL(ED),  
        .SM(ED),  
        .SN(ED),  
        .SO(ED),  
        .SP(ED),  
        .SQ(ED),  
        .SR(ED),  
        .SS(ED),  
        .ST(ED),  
        .SU(ED),  
        .SV(ED),  
        .SW(ED),  
        .SX(ED),  
        .SY(ED),  
        .SZ(ED),  
        .TA(ED),  
        .TB(ED),  
        .TC(ED),  
        .TD(ED),  
        .TE(ED),  
        .TF(ED),  
        .TG(ED),  
        .TH(ED),  
        .TI(ED),  
        .TJ(ED),  
        .TK(ED),  
        .TL(ED),  
        .TM(ED),  
        .TN(ED),  
        .TO(ED),  
        .TP(ED),  
        .TQ(ED),  
        .TR(ED),  
        .TS(ED),  
        .TT(ED),  
        .TU(ED),  
        .TV(ED),  
        .TW(ED),  
        .TX(ED),  
        .TY(ED),  
        .TZ(ED),  
        .UA(ED),  
        .UB(ED),  
        .UC(ED),  
        .UD(ED),  
        .UE(ED),  
        .UF(ED),  
        .UG(ED),  
        .UH(ED),  
        .UI(ED),  
        .UJ(ED),  
        .UK(ED),  
        .UL(ED),  
        .UM(ED),  
        .UN(ED),  
        .UO(ED),  
        .UP(ED),  
        .UQ(ED),  
        .UR(ED),  
        .US(ED),  
        .UT(ED),  
        .UU(ED),  
        .UV(ED),  
        .UW(ED),  
        .UX(ED),  
        .UY(ED),  
        .UZ(ED),  
        .VA(ED),  
        .VB(ED),  
        .VC(ED),  
        .VD(ED),  
        .VE(ED),  
        .VF(ED),  
        .VG(ED),  
        .VH(ED),  
        .VI(ED),  
        .VJ(ED),  
        .VK(ED),  
        .VL(ED),  
        .VM(ED),  
        .VN(ED),  
        .VO(ED),  
        .VP(ED),  
        .VQ(ED),  
        .VR(ED),  
        .VS(ED),  
        .VT(ED),  
        .VU(ED),  
        .VV(ED),  
        .VW(ED),  
        .VX(ED),  
        .VY(ED),  
        .VZ(ED),  
        .WA(ED),  
        .WB(ED),  
        .WC(ED),  
        .WD(ED),  
        .WE(ED),  
        .WF(ED),  
        .WG(ED),  
        .WH(ED),  
        .WI(ED),  
        .WJ(ED),  
        .WK(ED),  
        .WL(ED),  
        .WM(ED),  
        .WN(ED),  
        .WO(ED),  
        .WP(ED),  
        .WQ(ED),  
        .WR(ED),  
        .WS(ED),  
        .WT(ED),  
        .WU(ED),  
        .WV(ED),  
        .WW(ED),  
        .WX(ED),  
        .WY(ED),  
        .WZ(ED),  
        .XA(ED),  
        .XB(ED),  
        .XC(ED),  
        .XD(ED),  
        .XE(ED),  
        .XF(ED),  
        .XG(ED),  
        .XH(ED),  
        .XI(ED),  
        .XJ(ED),  
        .XK(ED),  
        .XL(ED),  
        .XM(ED),  
        .XN(ED),  
        .XO(ED),  
        .XP(ED),  
        .XQ(ED),  
        .XR(ED),  
        .XS(ED),  
        .XT(ED),  
        .XU(ED),  
        .XV(ED),  
        .XW(ED),  
        .XX(ED),  
        .XY(ED),  
        .XZ(ED),  
        .YA(ED),  
        .YB(ED),  
        .YC(ED),  
        .YD(ED),  
        .YE(ED),  
        .YF(ED),  
        .YG(ED),  
        .YH(ED),  
        .YI(ED),  
        .YJ(ED),  
        .YK(ED),  
        .YL(ED),  
        .YM(ED),  
        .YN(ED),  
        .YO(ED),  
        .YP(ED),  
        .YQ(ED),  
        .YR(ED),  
        .YS(ED),  
        .YT(ED),  
        .YU(ED),  
        .YV(ED),  
        .YW(ED),  
        .YX(ED),  
        .YY(ED),  
        .YZ(ED),  
        .ZA(ED),  
        .ZB(ED),  
        .ZC(ED),  
        .ZD(ED),  
        .ZE(ED),  
        .ZF(ED),  
        .ZG(ED),  
        .ZH(ED),  
        .ZI(ED),  
        .ZJ(ED),  
        .ZK(ED),  
        .ZL(ED),  
        .ZM(ED),  
        .ZN(ED),  
        .ZO(ED),  
        .ZP(ED),  
        .ZQ(ED),  
        .ZR(ED),  
        .ZS(ED),  
        .ZT(ED),  
        .ZU(ED),  
        .ZV(ED),  
        .ZW(ED),  
        .ZX(ED),  
        .ZY(ED),  
        .ZZ(ED)  
    );
```

```
.BIN(d),  
  
.TWOSCOMP(inv d)  
);  
  
endmodule
```

#### Full adder

```
/**  
cse3341, digital logic 2, lab 5  
George Boone  
1002055713  
*/  
  
module FullAdder(  
    input A, B, CIN,  
    output S, COUT  
);  
  
    wire axb;  
  
    assign axb = A ^ B;  
  
    assign S = axb ^ CIN;  
    assign COUT = ((axb) & CIN) | (A & B);  
  
endmodule
```

#### Ripple carry subtractor

```
/**  
cse3341, digital logic 2, lab 5  
George Boone  
1002055713  
  
accepts 2 raw binary nums  
*/  
  
module RCSubtractor#(  
    parameter N = 8  
)(  
    input [N-1:0] A, B,  
    output [N-1:0] D,  
    output COUT
```

```
);

wire [N-1:0] _carries;

assign COUT = _carries[N-1];

// ripple carry design
genvar i;
generate
for(i = N-1; i >= 0; i = i-1) begin : adder_array
FullAdder _fa(
.A(A[i]),
.B(~B[i]),
.CIN(i == 0 ? 1 : _carries[i-1]),
.S(D[i]),
.COUT(_carries[i])
);
end
endgenerate

endmodule
```

### 2 to 1 multiplexer

```
/**
cse3341, digital logic 2, lab 5
George Boone
1002055713
*/

module Mux2t1#(
    parameter N = 8
)(
    input CTRL,           // 0:A, 1:B
    input [N-1:0] A, B,
    output wire [N-1:0] OUT
);

    assign OUT = CTRL ? B : A;

endmodule
```

### Binary to 2's complement converter

```
/**
```

```
cse3341, digital logic 2, lab 5
George Boone
1002055713
*/

module BintTwosComp#(
    parameter N = 8
)(
    input [N-1:0] BIN,
    output reg [N-1:0] TWOSCOMP,

    // debug
    output reg COUT
);

    wire [N-1:0] _carries;

    assign COUT = ~_carries[N-1];

    // ripple carry design
    genvar i;
    generate
    for(i = N-1; i >= 0; i = i-1) begin : adder_array
        FullAdder _fa(
            .A(~BIN[i]),
            .B(0),
            .CIN(i == 0 ? 1 : _carries[i-1]),
            .S(TWOSCOMP[i]),
            .COUT(_carries[i])
        );
    end
    endgenerate

endmodule
```