```
module  HalfAdder       (A, B, SUM, CY_OUT);
input   A, B;
output  SUM, CY_OUT;
        assign SUM    = A ^ B;
        assign CY_OUT = A & B;
endmodule
```

```
module  FullAdder       (A, B, CY_IN, SUM, CY_OUT);
input   A, B, CY_IN;
output  SUM, CY_OUT;
        assign SUM    = A ^ B ^CY_IN;
        assign CY_OUT = (A & B) | (A & CY_IN) | (B & CY_IN);
endmodule
```

```
module  FullAdder       (A, B, CY_IN, SUM_OUT, CY_OUT);
input   A, B, CY_IN;
output  SUM_OUT, CY_OUT;
wire    SUM1, CY1, CY2;
        HalfAdder       HA1     (A, B, SUM1, CY1);
        HalfAdder       HA2     (SUM1, CY_IN, SUM_OUT, CY2);
        or      OR      (CY_OUT, CY1, CY2);
endmodule

module  HalfAdder       (A, B, SUM, CY_OUT);
input   A, B;
output  SUM, CY_OUT;
        assign SUM    = A ^ B;
        assign CY_OUT = A & B;
endmodule
```

```
module  HalfSubtractor  (A, B, DIF, BR_OUT);
input   A, B;
output  DIF, BR_OUT;
        assign DIF    =  A ^ B;
        assign BR_OUT = (~A) & B;       //      *1
endmodule
```

```
module  FullSubtractor  (A, B, BR_IN, DIF, BR_OUT);
input   A, B, BR_IN;
output  DIF, BR_OUT;
        assign DIF    = A ^ B ^ BR_IN;
        assign BR_OUT = (~A & B) | (~A & BR_IN) | (B & BR_IN);
endmodule
```

```
module  FullSubtractor  (A, B, BR_IN, DIF_OUT, BR_OUT);
input   A, B, BR_IN;
output  DIF_OUT, BR_OUT;
wire    DIF1, BR1, BR2;
        HalfSubstractor HS1     (A, B, DIF1, BR1);
        HalfSubstractor HS2     (DIF1, BR_IN, DIF_OUT, BR2);
        or      OR      (BR_OUT, BR1, BR2);
endmodule

module  HalfSubstractor (A, B, DIF, BR_OUT);
input   A, B;
output  DIF, BR_OUT;
        assign DIF    =  A ^ B;
        assign BR_OUT = (~A) & B;       //      *1
endmodule
```

```
module  ADDER4P ( A_IN, B_IN, SUM, OVF);
input           [3:0]   A_IN, B_IN;
output  [3:0]   SUM;
wire            [2:0]   CY;
output  OVF;

        FullAdder       FA0     (A_IN[0], B_IN[0],     0, SUM[0], CY[0]);
        FullAdder       FA1     (A_IN[1], B_IN[1], CY[0], SUM[1], CY[1]);
        FullAdder       FA2     (A_IN[2], B_IN[2], CY[1], SUM[2], CY[2]);
        FullAdder       FA3     (A_IN[3], B_IN[3], CY[2], SUM[3], OVF);
endmodule

module  FullAdder       (A, B, CY_IN, SUM, CY_OUT);
input   A, B, CY_IN;
output  SUM, CY_OUT;

        assign #15  SUM  = A ^ B ^ CY_IN;
        assign #15  CY_OUT = (A & B) | (A & CY_IN) | (B & CY_IN);
endmodule
```

```
module  ADDER4HSP       ( A_IN, B_IN, SUM, OVF);
  input         [3:0]   A_IN, B_IN;
  output        [3:0]   SUM;
  output        OVF;
  wire          CIN;
  wire          S0, S1, S2, S3, C0, C1, C2, C3;
  wire          N1, N2, N3, N4, N5, N6, N7, N8, N9, N10;
  wire          R1, R2, R3;
  wire          Y0, Y1, Y2, Y3;

        HalfAdder       HA0     (A_IN[0], B_IN[0], S0, C0);
        HalfAdder       HA1     (A_IN[1], B_IN[1], S1, C1);
        HalfAdder       HA2     (A_IN[2], B_IN[2], S2, C2);
        HalfAdder       HA3     (A_IN[3], B_IN[3], S3, C3);
        HalfAdder       HA4     (S0, CIN, SUM[0], Y0);
        HalfAdder       HA5     (S1,  R1, SUM[1], Y1);
        HalfAdder       HA6     (S2,  R2, SUM[2], Y2);
        HalfAdder       HA7     (S3,  R3, SUM[3], Y3);

        and     #15  AND1  (N1, S0, CIN);
        and     #15  AND2  (N2, C0, S1);
        and     #15  AND3  (N3, CIN, S0, S1);
        and     #15  AND4  (N4, C1, S2);
        and     #15  AND5  (N5, C0, S1, S2);
        and     #15  AND6  (N6, CIN, S0, S1, S2);
        and     #15  AND7  (N7, C2, S3);
        and     #15  AND8  (N8, C1, S2, S3);
        and     #15  AND9  (N9, C0, S1, S2, S3);
        and     #15  AND10 (N10, CIN, S0, S1, S2, S3);

        or      #15  OR1   (R1, C0, N1);
        or      #15  OR2   (R2, C1, N2, N3);
        or      #15  OR3   (R3, C2, N4, N5, N6);
        or      #15  OR4   (OVF, C3, N7, N8, N9, N10);

        assign CIN = 0;
endmodule

module  HalfAdder       (A, B, SUM, CY_OUT);
input   A, B;
output  SUM, CY_OUT;
        assign  #15  SUM    = A ^ B;
        assign  #15  CY_OUT = A & B;
endmodule
```

```
module  ADDER4  ( A_IN, B_IN, DATA_LOAD, CLK, RST, Q);
input   [3:0]   A_IN, B_IN;
input   DATA_LOAD, CLK, RST;
wire    A, B, CY_IN, CY_OUT, SUM;
output  [4:0]  Q;

        PIN_SOUT_4R_SHIFT       REG_A   (DATA_LOAD, A_IN, A, CLK);
        PIN_SOUT_4R_SHIFT       REG_B  (DATA_LOAD, B_IN, B, CLK);
        SIN_POUT_5R_SHIFT       REG_S  (RST, SUM, CLK, Q);
        FullAdder       FA      (A, B, CY_IN, SUM, CY_OUT);
        R_SYDFF CY_REG  (RST, CY_OUT, CLK, CY_IN);
endmodule

module  PIN_SOUT_4R_SHIFT       (DATA_LOAD, P_IN, S_OUT, CLK);
input   DATA_LOAD, CLK;
input   [3:0] P_IN;
output  S_OUT;
reg     [3:0] Q;

        assign  S_OUT = Q[0];

        always  @(posedge CLK or posedge DATA_LOAD) begin
                if      (DATA_LOAD)
                                Q = P_IN;
                else
                                Q = Q >> 1;
        end
endmodule

module  SIN_POUT_5R_SHIFT       (RST, S_IN, CLK, P_OUT);
input   CLK, S_IN, RST;
output  [4:0]   P_OUT;
reg     [4:0]   P_OUT;
        always  @(posedge CLK or posedge RST)
                if      (RST)
                        P_OUT <= 0;
                else
                        P_OUT <= {S_IN, P_OUT} >> 1;
endmodule

module  FullAdder       (A, B, CY_IN, SUM, CY_OUT);
input   A, B, CY_IN;
output  SUM, CY_OUT;
        assign SUM    = A ^ B ^ CY_IN;
        assign CY_OUT = (A & B) | (A & CY_IN) | (B & CY_IN);
endmodule
```

```
module  R_SYDFF (R, D, CLK, Q);
input   R, D, CLK;
output  Q;
reg     Q;
        always  @(posedge CLK or posedge R)
                if ( R ) Q = 0;
                else     Q = D;
endmodule
```

```
module  BCD_ADDER ( A, B, BCD_CY_IN, BCD_SUM, BCD_CY_OUT );
input   [3:0]   A, B;
input   BCD_CY_IN;
output  [3:0]   BCD_SUM;
output  BCD_CY_OUT;
wire    [3:0]   S;
wire    CY1, CY2, N1, N2;

        ADDER4P ADDR1   (A, B, BCD_CY_IN, S, CY1);
        ADDER4P ADDR2   (S, {1'b0, BCD_CY_OUT, BCD_CY_OUT, 1'b0}, 0, BCD_SUM, CY2);
        //*1

        and     AND1    (N1, S[3], S[2]);
        and     AND2    (N2, S[3], S[1]);
        or      OR1     (BCD_CY_OUT, CY1, N1, N2);
endmodule

module  ADDER4P ( A_IN, B_IN, CY_IN, SUM, CY_OUT);
input   [3:0]   A_IN, B_IN;
input   CY_IN;
output  [3:0]   SUM;
output  CY_OUT;
wire    [2:0]   CY;

        FullAdder       FA0     (A_IN[0], B_IN[0], CY_IN, SUM[0], CY[0]);
        FullAdder       FA1     (A_IN[1], B_IN[1], CY[0], SUM[1], CY[1]);
        FullAdder       FA2     (A_IN[2], B_IN[2], CY[1], SUM[2], CY[2]);
        FullAdder       FA3     (A_IN[3], B_IN[3], CY[2], SUM[3], CY_OUT);
endmodule

module  FullAdder       (A, B, CY_IN, SUM, CY_OUT);
input   A, B, CY_IN;
output  SUM, CY_OUT;
        assign  SUM  = A ^ B ^ CY_IN;
        assign  CY_OUT = (A & B) | (A & CY_IN) | (B & CY_IN);
endmodule
```

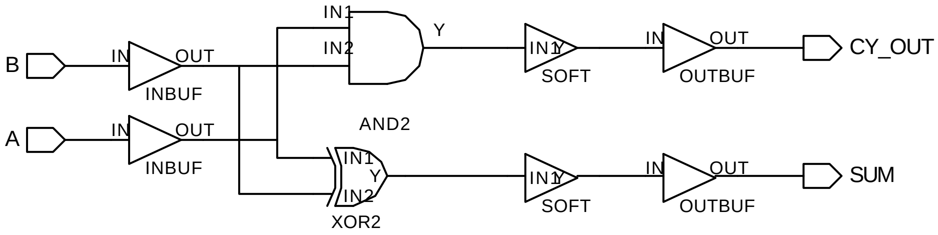
```
module  SUB4P   ( A_IN, B_IN, DIF, DOWN_FLW);
output  DOWN_FLW;
input   [3:0]   A_IN, B_IN;
output  [3:0]   DIF;
wire    [2:0]   BR;
        FullSubstractor FS3     (A_IN[3], B_IN[3], BR[2], DIF[3], DOWN_FLW),
                        FS2     (A_IN[2], B_IN[2], BR[1], DIF[2], BR[2]),
                        FS1     (A_IN[1], B_IN[1], BR[0], DIF[1], BR[1]),
                        FS0     (A_IN[0], B_IN[0], 0, DIF[0], BR[0]);
endmodule

module  FullSubstractor (A, B, BR_IN, DIF, BR_OUT);
input   A, B, BR_IN;
output  DIF, BR_OUT;
        assign DIF    = A ^ B ^BR_IN;
        assign BR_OUT = (~A & B) | (~A & BR_IN) | (B & BR_IN);
endmodule
```

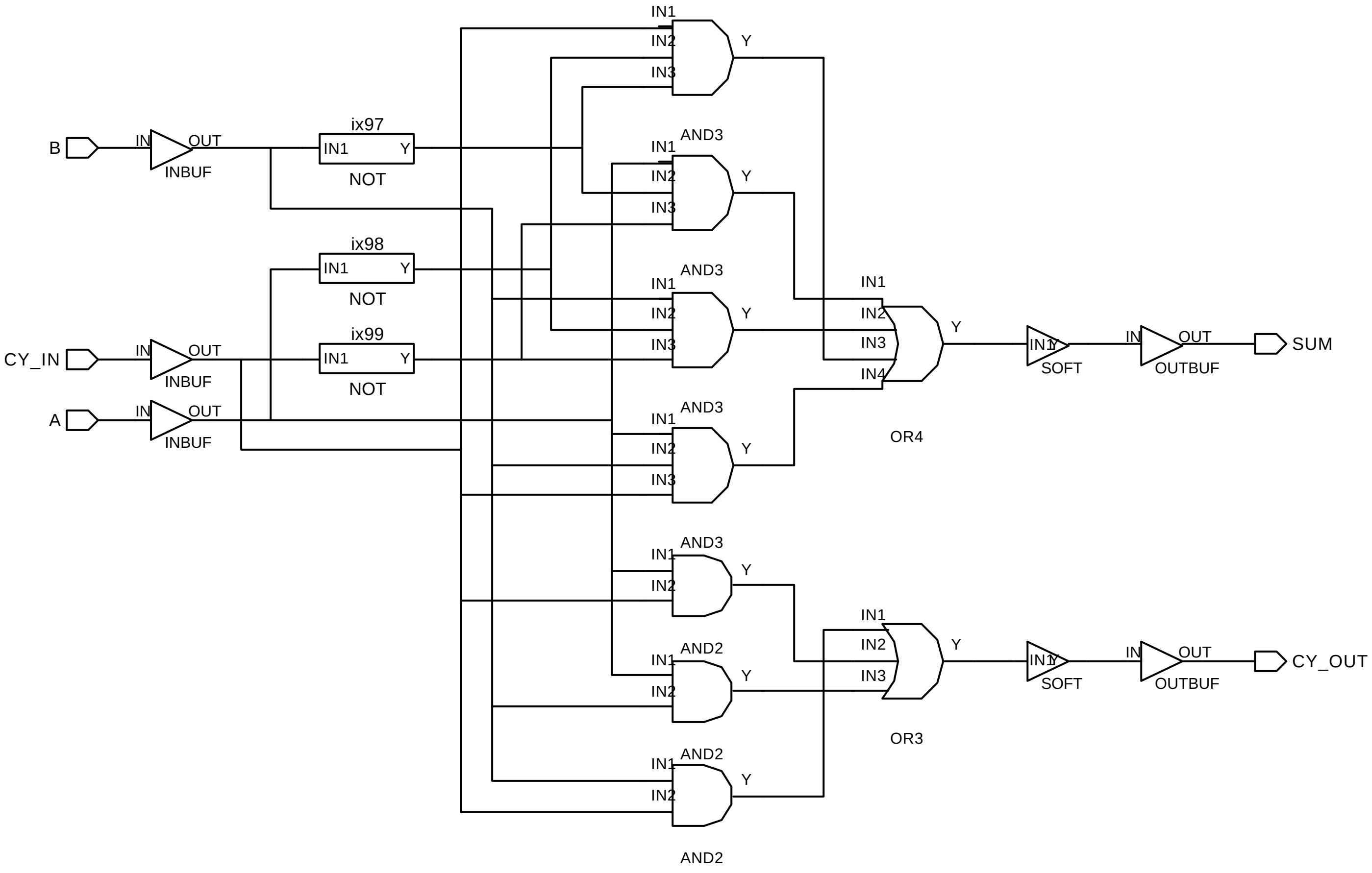
```
module ADSUB4P  ( A, B, AS, S_S, S_NS, OV_DW_S, OV_DW_NS);
  input  [3:0]  A, B;           // A + B or A - B => RESULT
  input         AS;             // ADD:0, SUB:1
  output [3:0]  S_S, S_NS;
  output        OV_DW_S, OV_DW_NS;
  wire   [3:0]  C, COMP;
  wire          SUM3, X1, X2, NX1, ND1, ND2;

  FullAdder     FA0  (A[0], COMP[0],   AS, S_S[0], C[0]);
  FullAdder     FA1  (A[1], COMP[1], C[0], S_S[1], C[1]);
  FullAdder     FA2  (A[2], COMP[2], C[1], S_S[2], C[2]);
  FullAdder     FA3  (A[3], COMP[3], C[2], SUM3, C[3]);
  xor           EX00 (COMP[0], B[0], AS);
  xor           EX01 (COMP[1], B[1], AS);
  xor           EX02 (COMP[2], B[2], AS);
  xor           EX03 (COMP[3], B[3], AS);
  xor           EX04 (X1, A[3], COMP[3]);
  xor           EX05 (X2, C[3], SUM3);
  xor           EX06 (OV_DW_NS, C[3], AS);
  and           AND1 (OV_DW_S, X2, NX1);
  and           AND2 (ND1, SUM3, X1);
  and           AND3 (ND2, C[3], NX1);
  or            OR1  (S_S[3], ND2, ND1);
  not           NOT1 (NX1, X1);
  assign        S_NS[2:0] = S_S[2:0];
  assign        S_NS[3]   = SUM3;
endmodule

module FullAdder (A, B, Cy_IN, SUM, CY_OUT);
input A, B, Cy_IN;
output SUM, CY_OUT;

    assign      SUM  = A ^ B ^ Cy_IN;
    assign      CY_OUT = (A & B) | (A & Cy_IN) | (B & Cy_IN);
endmodule
```

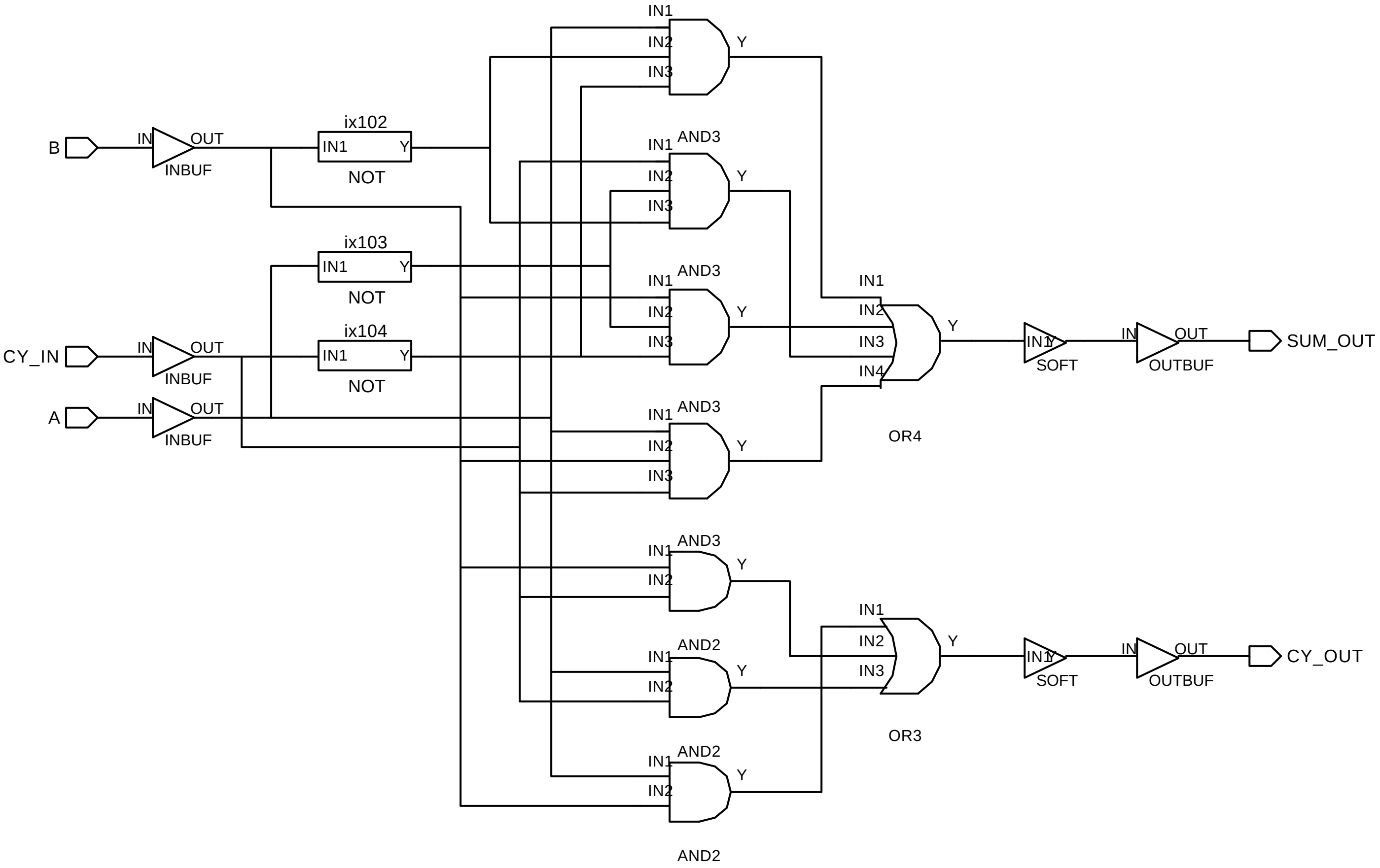
```
module  ADDER4  ( A_IN, B_IN, SUM, OVF);
input   [3:0]   A_IN, B_IN;
output  [3:0]   SUM;
wire    [2:0]   CY;
output  OVF;

        FullAdder       FA0     (A_IN[0], B_IN[0],     0, SUM[0], CY[0]);
        FullAdder       FA1     (A_IN[1], B_IN[1], CY[0], SUM[1], CY[1]);
        FullAdder       FA2     (A_IN[2], B_IN[2], CY[1], SUM[2], CY[2]);
        FullAdder       FA3     (A_IN[3], B_IN[3], CY[2], SUM[3], OVF);
endmodule

module  FullAdder       (A, B, CY_IN, SUM_OUT, CY_OUT);
input   A, B, CY_IN;
output  SUM_OUT, CY_OUT;
wire    SUM1, CY1, CY2;
        HalfAdder HA1 (A, B, SUM1, CY1);
        HalfAdder HA2 (SUM1, CY_IN, SUM_OUT, CY2);
        or      #15     OR      (CY_OUT, CY1, CY2);
endmodule

module  HalfAdder       (A, B, SUM, CY_OUT);
input   A, B;
output  SUM, CY_OUT;
        assign  #15     SUM    = A ^ B;
        assign  #15     CY_OUT = A & B;
endmodule
```

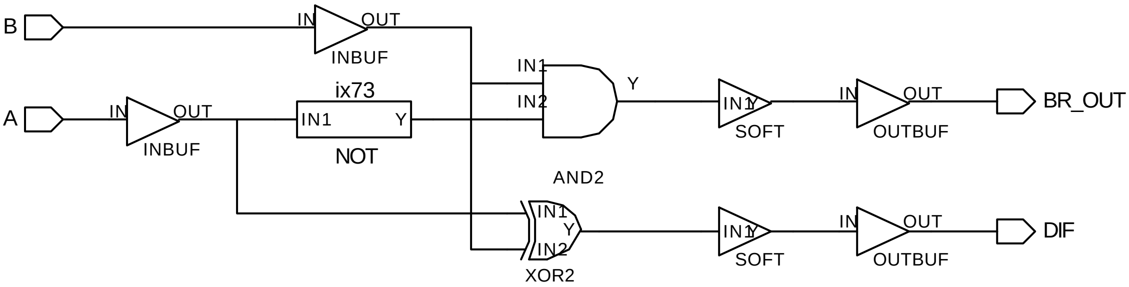
```
module  ADDER4  ( A_IN, B_IN, SUM, OVF);
input           [3:0]   A_IN, B_IN;
output  [3:0]   SUM;
wire            [2:0]   CY;
output  OVF;

        FullAdder       FA0     (A_IN[0], B_IN[0],     0, SUM[0], CY[0]);
        FullAdder       FA1     (A_IN[1], B_IN[1], CY[0], SUM[1], CY[1]);
        FullAdder       FA2     (A_IN[2], B_IN[2], CY[1], SUM[2], CY[2]);
        FullAdder       FA3     (A_IN[3], B_IN[3], CY[2], SUM[3], OVF);
endmodule

module  FullAdder       (A, B, CY_IN, SUM, CY_OUT);
input   A, B, CY_IN;
output  SUM, CY_OUT;

        assign #15  SUM  = A ^ B ^ CY_IN;
        assign #15  CY_OUT = (A & B) | (A & CY_IN) | (B & CY_IN);
endmodule
```

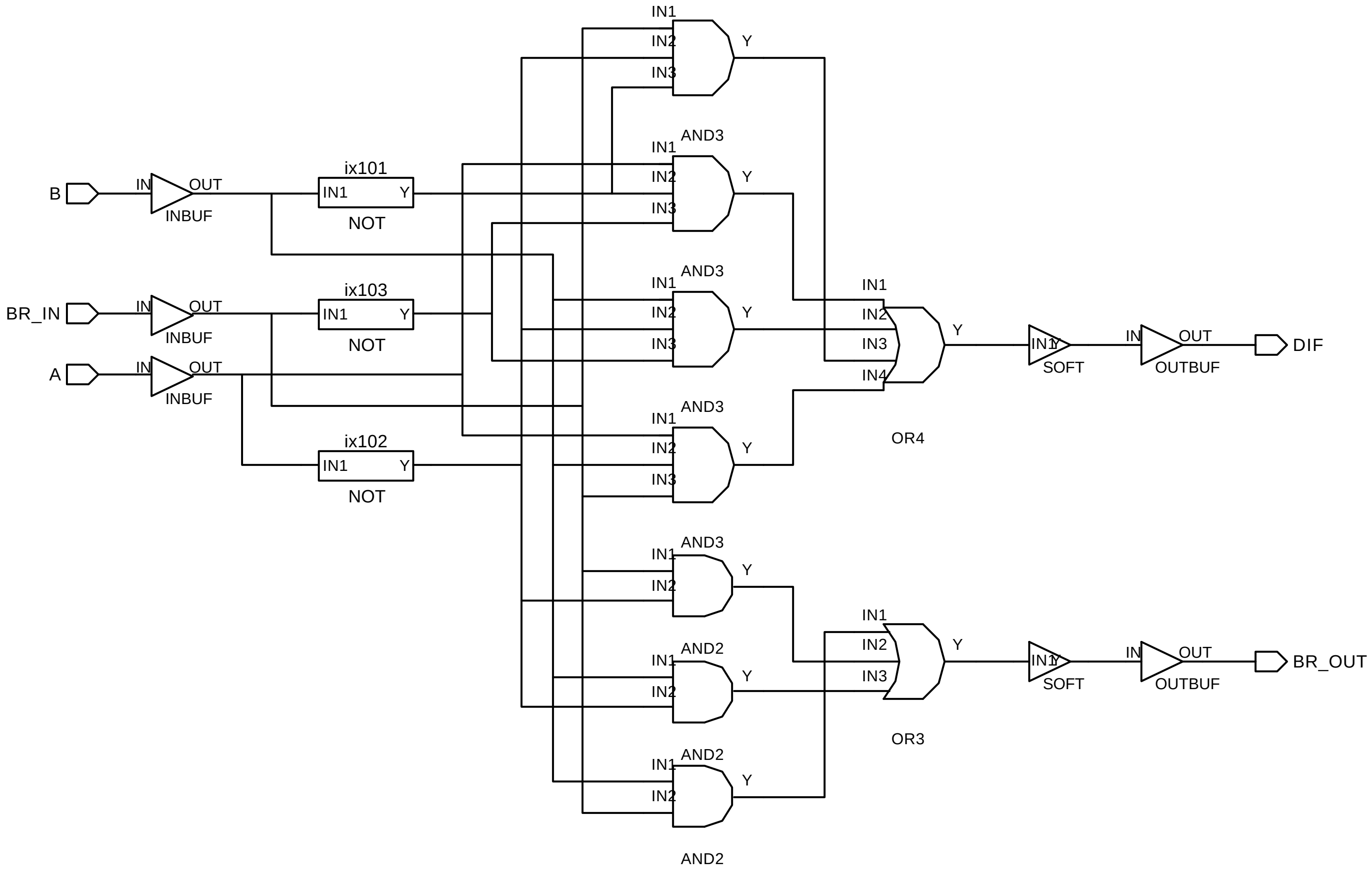
```
module  ADDER4  ( A_IN, B_IN, SUM, OVF);
  input         [3:0]   A_IN, B_IN;
  output        [3:0]   SUM;
  output        OVF;
  wire          CIN;
  wire          S0, S1, S2, S3, C0, C1, C2, C3;
  wire          N1, N2, N3, N4, N5, N6, N7, N8, N9, N10;
  wire          R1, R2, R3;
  wire          Y0, Y1, Y2, Y3;

        HalfAdder       HA0     (A_IN[0], B_IN[0], S0, C0);
        HalfAdder       HA1     (A_IN[1], B_IN[1], S1, C1);
        HalfAdder       HA2     (A_IN[2], B_IN[2], S2, C2);
        HalfAdder       HA3     (A_IN[3], B_IN[3], S3, C3);
        HalfAdder       HA4     (S0, CIN, SUM[0], Y0);
        HalfAdder       HA5     (S1,  R1, SUM[1], Y1);
        HalfAdder       HA6     (S2,  R2, SUM[2], Y2);
        HalfAdder       HA7     (S3,  R3, SUM[3], Y3);

        and     #15  AND1  (N1, S0, CIN);
        and     #15  AND2  (N2, C0, S1);
        and     #15  AND3  (N3, CIN, S0, S1);
        and     #15  AND4  (N4, C1, S2);
        and     #15  AND5  (N5, C0, S1, S2);
        and     #15  AND6  (N6, CIN, S0, S1, S2);
        and     #15  AND7  (N7, C2, S3);
        and     #15  AND8  (N8, C1, S2, S3);
        and     #15  AND9  (N9, C0, S1, S2, S3);
        and     #15  AND10 (N10, CIN, S0, S1, S2, S3);

        or      #15  OR1   (R1, C0, N1);
        or      #15  OR2   (R2, C1, N2, N3);
        or      #15  OR3   (R3, C2, N4, N5, N6);
        or      #15  OR4   (OVF, C3, N7, N8, N9, N10);

        assign CIN = 0;
endmodule

module  HalfAdder       (A, B, SUM, CY_OUT);
input   A, B;
output  SUM, CY_OUT;
        assign  #15  SUM    = A ^ B;
        assign  #15  CY_OUT = A & B;
endmodule
```


B
A
IN
OUT
INBUF
IN
OUT
INBUF
IN1
IN2
Y
AND2
IN1
IN2
Y
XOR2
IN1
Y
SOFT
IN1
Y
SOFT
IN
OUT
OUTBUF
IN
OUT
OUTBUF
CY_OUT
SUM

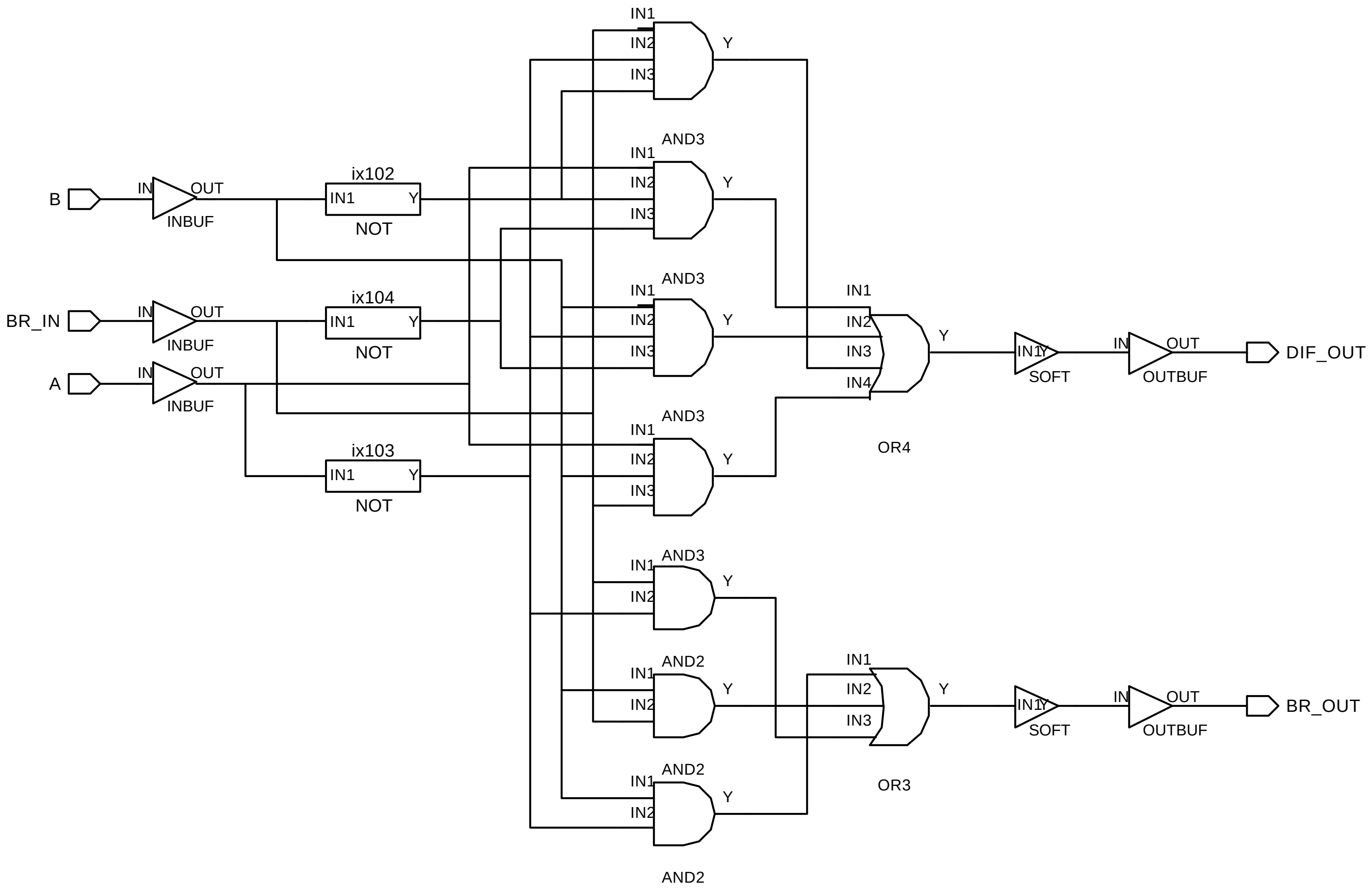

IN1
IN2
IN3
Y
AND3
B
IN
OUT
INBUF
ix97
IN1
Y
NOT
IN1
IN2
IN3
Y
AND3
ix98
IN1
Y
NOT
IN1
IN2
IN3
Y
IN1
IN2
IN3
IN4
Y
ix99
IN1
Y
NOT
CY_IN
IN
OUT
INBUF
IN1
Y
SOFT
IN
OUT
OUTBUF
SUM
AND3
A
IN
OUT
INBUF
OR4
IN1
IN2
IN3
Y
AND3
IN1
IN2
Y
AND2
IN1
IN2
IN3
Y
IN1
Y
SOFT
IN
OUT
OUTBUF
CY_OUT
IN1
IN2
Y
OR3
AND2
IN1
IN2
Y
AND2

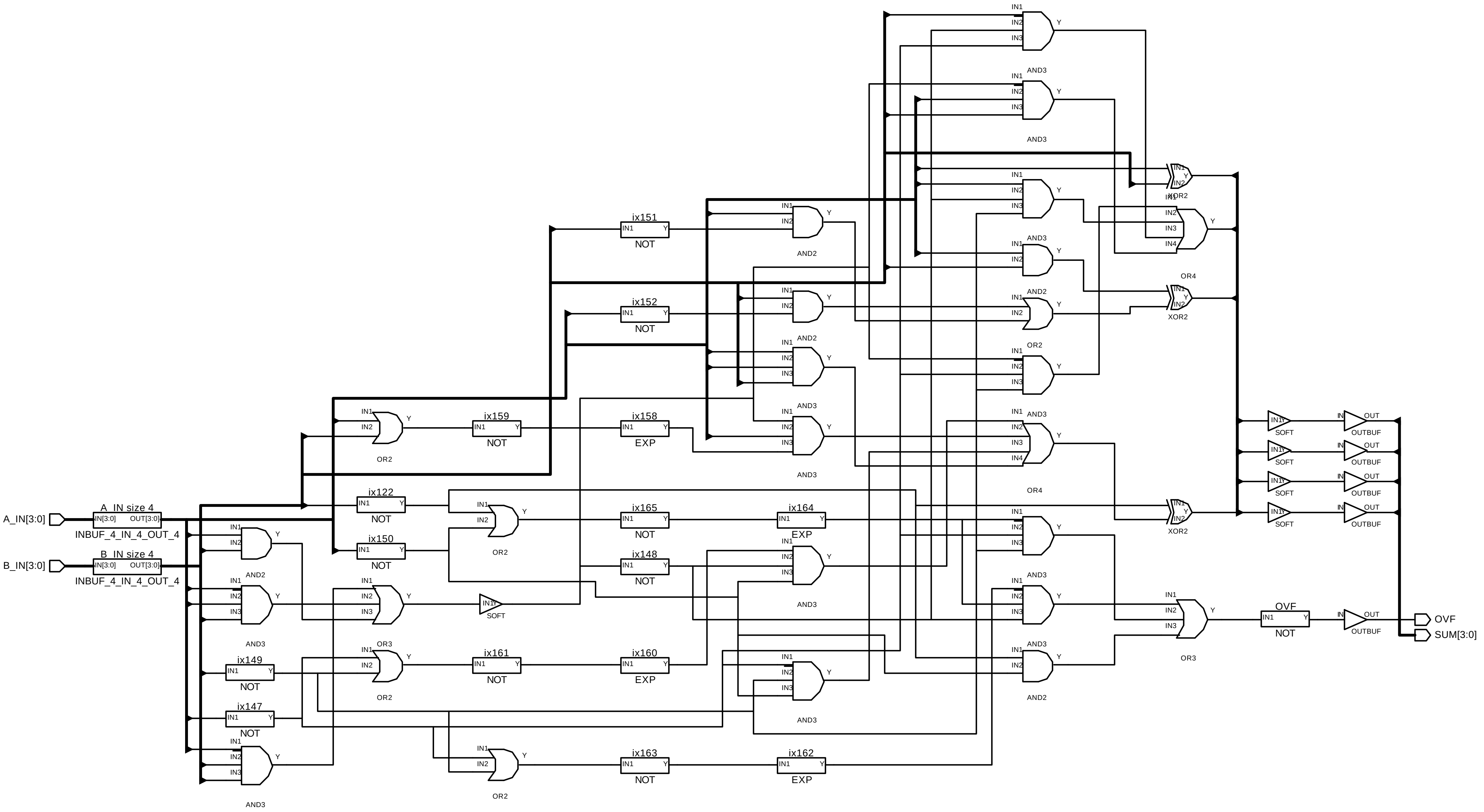

B
IN
OUT
INBUF
ix102
IN1
Y
NOT
ix103
IN1
Y
NOT
CY_IN
IN
OUT
INBUF
ix104
IN1
Y
NOT
A
IN
OUT
INBUF
IN1
IN2
IN3
Y
AND3
IN1
IN2
IN3
Y
AND3
IN1
IN2
IN3
Y
AND3
IN1
IN2
IN3
Y
AND3
IN1
IN2
Y
AND2
IN1
IN2
Y
AND2
IN1
IN2
Y
AND2
IN1
IN2
IN3
IN4
Y
OR4
IN1
IN2
IN3
Y
OR3
IN1
Y
SOFT
IN
OUT
OUTBUF
SUM_OUT
IN1
Y
SOFT
IN
OUT
OUTBUF
CY_OUT

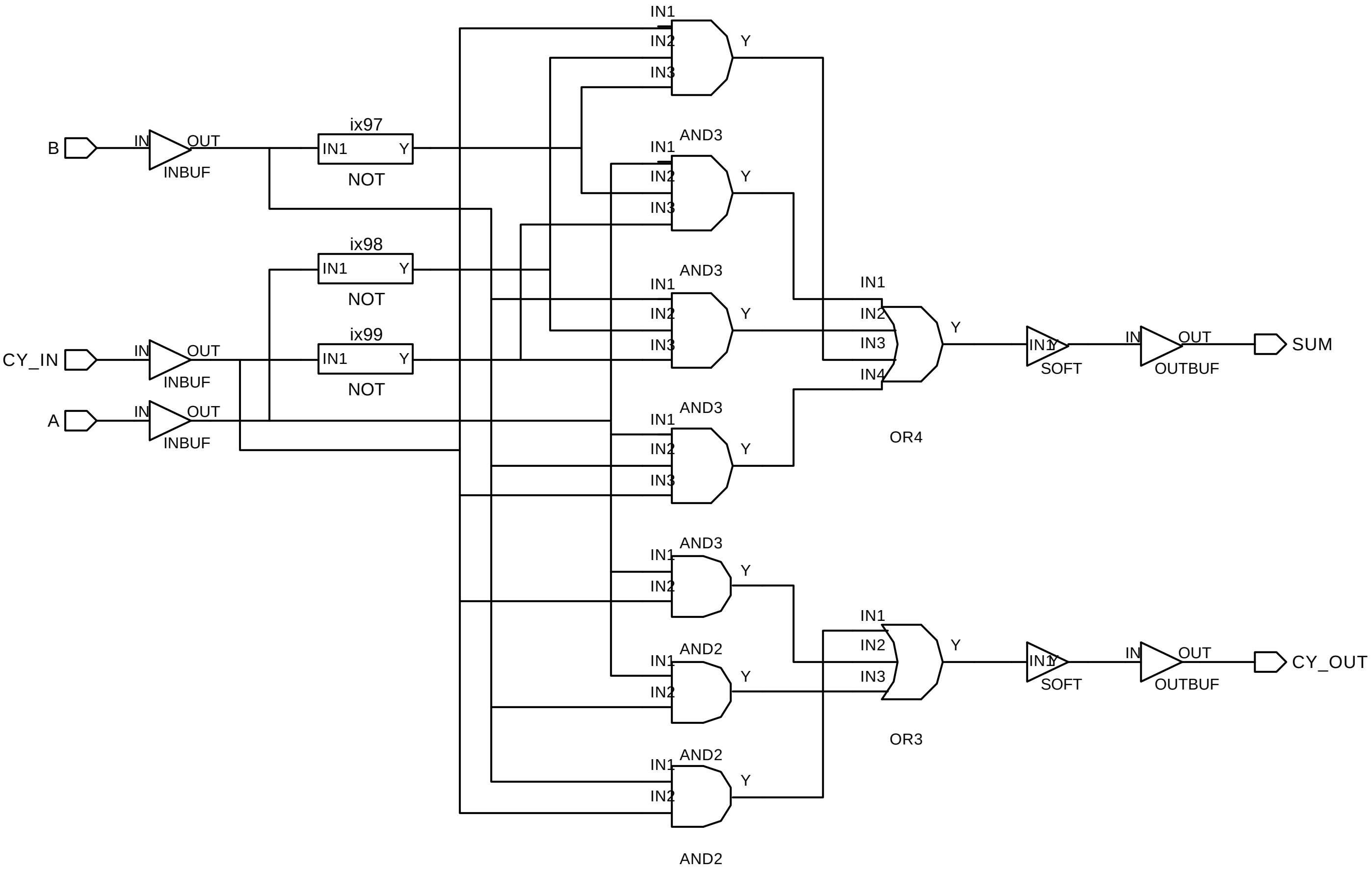

B
IN
OUT
INBUF
A
IN
OUT
INBUF
ix73
IN1
Y
NOT
IN1
IN2
Y
AND2
IN1
Y
SOFT
IN
OUT
OUTBUF
BR_OUT
IN1
IN2
Y
XOR2
IN1
Y
SOFT
IN
OUT
OUTBUF
DIF

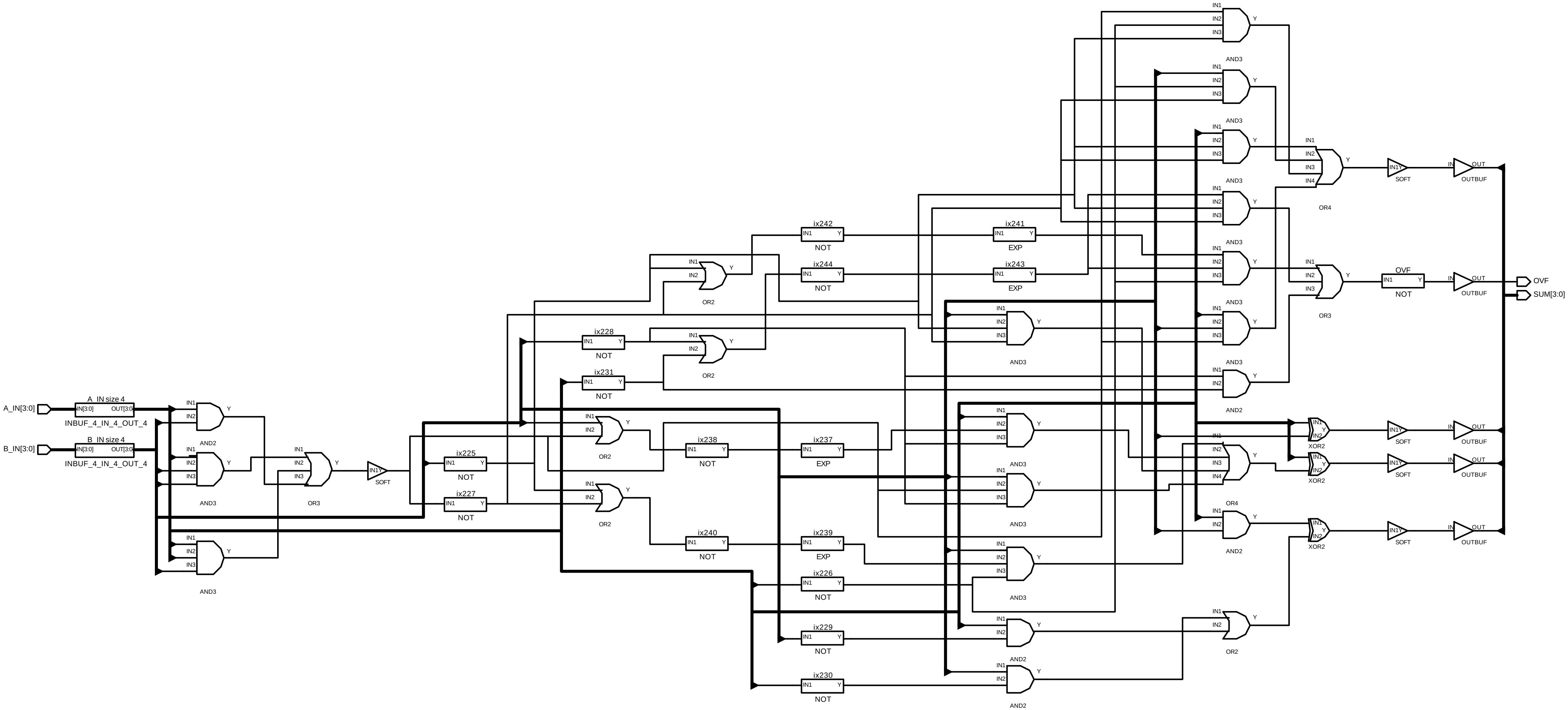


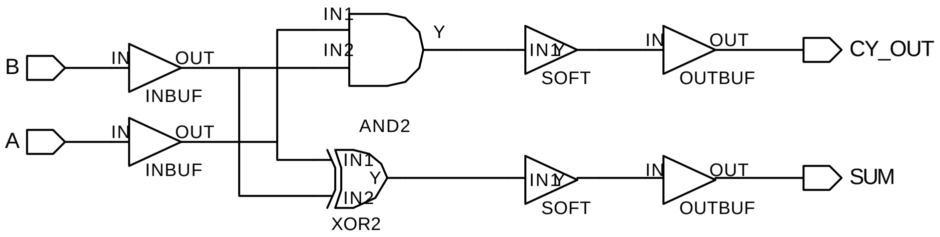
IN1
IN2
IN3
Y
AND3
IN1
IN2
IN3
Y
AND3
B
IN
OUT
INBUF
ix101
IN1
Y
NOT
BR_IN
IN
OUT
INBUF
ix103
IN1
Y
NOT
A
IN
OUT
INBUF
ix102
IN1
Y
NOT
IN1
IN2
IN3
Y
AND3
IN1
IN2
IN3
Y
AND3
IN1
IN2
Y
AND2
IN1
IN2
Y
AND2
IN1
IN2
Y
AND2
IN1
IN2
IN3
IN4
Y
OR4
IN1
Y
SOFT
IN
OUT
OUTBUF
DIF
IN1
IN2
IN3
Y
OR3
IN1
Y
SOFT
IN
OUT
OUTBUF
BR_OUT

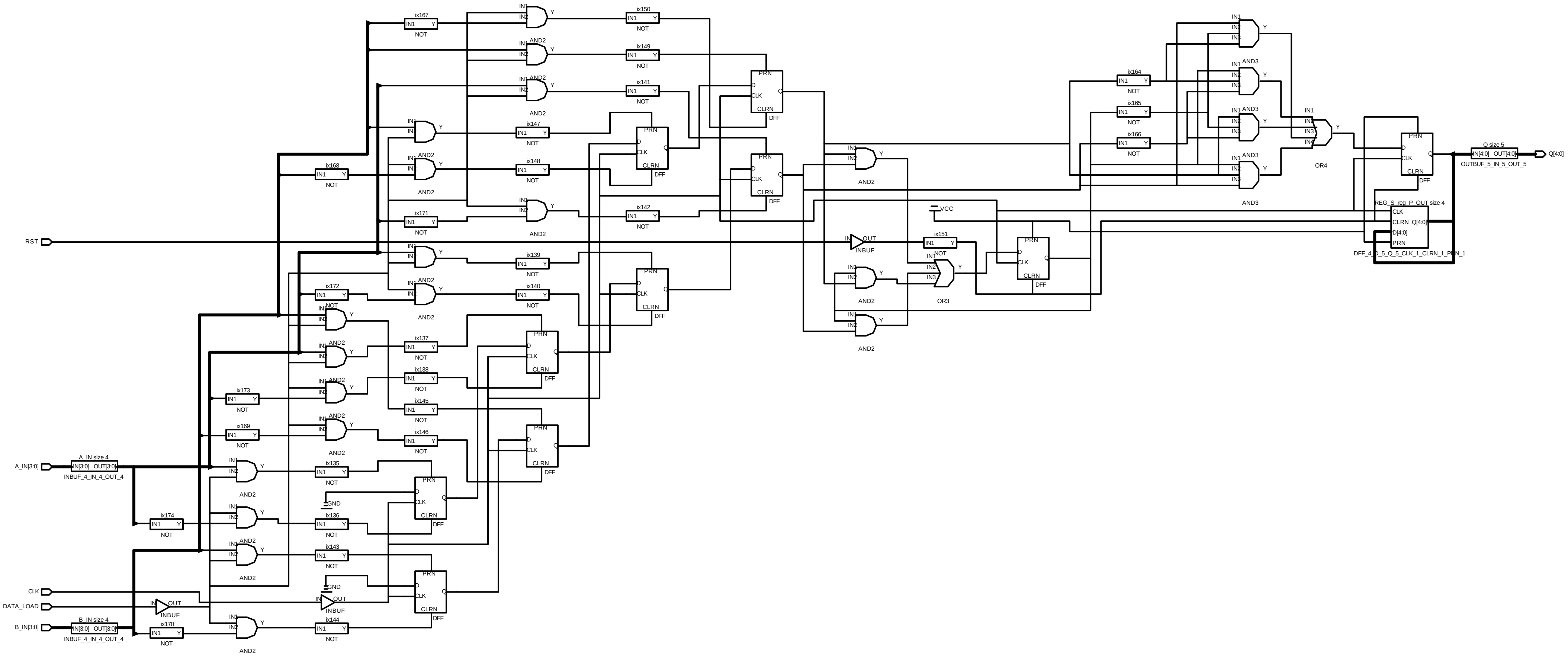

B
BR_IN
A
IN
OUT
INBUF
IN
OUT
INBUF
IN
OUT
INBUF
ix102
IN1
Y
NOT
ix104
IN1
Y
NOT
ix103
IN1
Y
NOT
IN1
IN2
IN3
Y
AND3
IN1
IN2
IN3
Y
AND3
IN1
IN2
IN3
Y
AND3
IN1
IN2
IN3
Y
AND3
IN1
IN2
Y
AND2
IN1
IN2
Y
AND2
IN1
IN2
Y
AND2
IN1
IN2
IN3
IN4
Y
OR4
IN1
IN2
IN3
Y
OR3
IN1
Y
SOFT
IN
OUT
OUTBUF
DIF_OUT
IN1
Y
SOFT
IN
OUT
OUTBUF
BR_OUT


A_IN[3:0]
B_IN[3:0]
A_IN size 4
B_IN size 4
IN[3:0]
OUT[3:0]
INBUF_4_IN_4_OUT_4
INBUF_4_IN_4_OUT_4
ix122
NOT
ix150
NOT
ix159
NOT
ix158
EXP
ix151
NOT
ix152
NOT
ix165
NOT
ix164
EXP
ix148
NOT
ix149
NOT
ix147
NOT
ix161
NOT
ix160
EXP
ix163
NOT
ix162
EXP
IN1
IN2
IN3
IN4
Y
AND2
AND3
OR2
OR3
OR4
XOR2
SOFT
OUTBUF
IN
OUT
OVF
NOT
OVF
SUM[3:0]

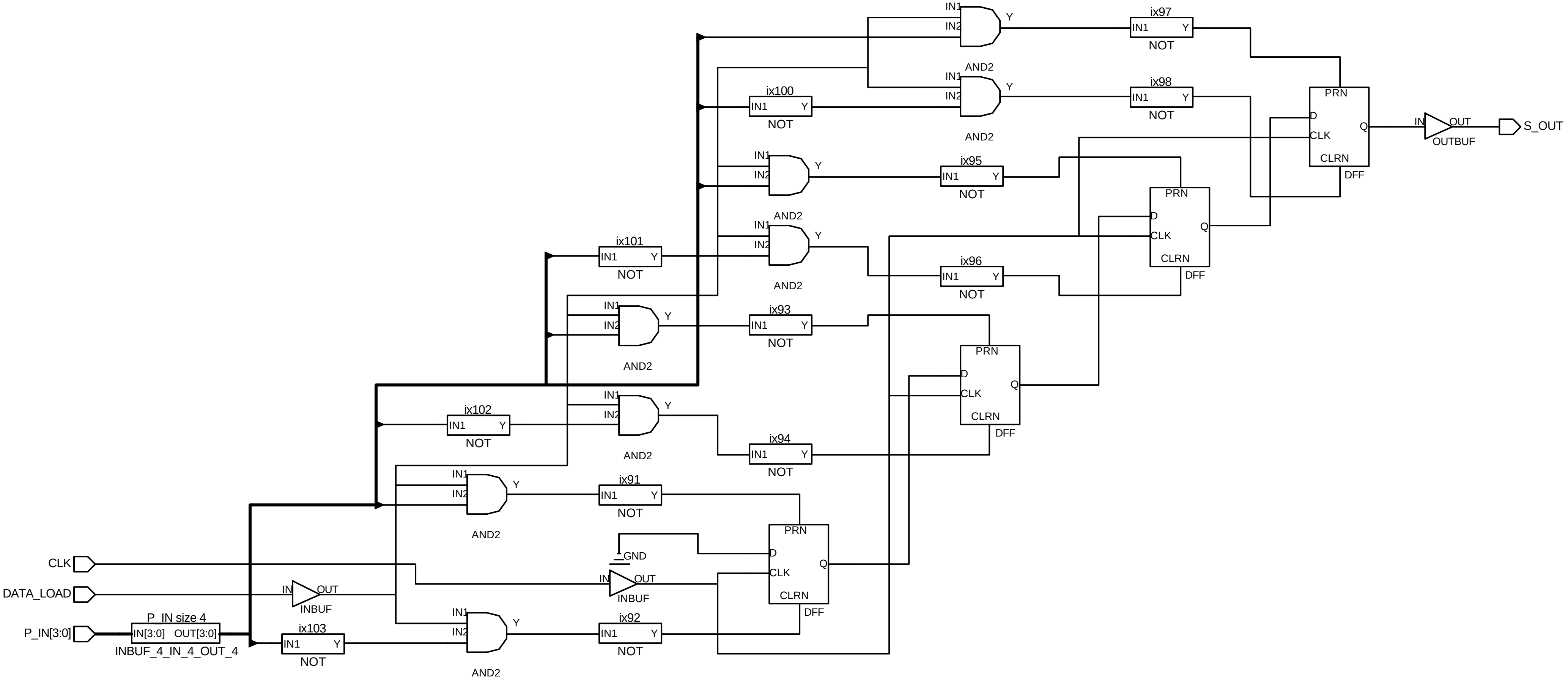

B
IN
OUT
INBUF
ix97
IN1
Y
NOT
ix98
IN1
Y
NOT
ix99
IN1
Y
NOT
CY_IN
IN
OUT
INBUF
A
IN
OUT
INBUF
IN1
IN2
IN3
Y
AND3
IN1
IN2
IN3
Y
AND3
IN1
IN2
IN3
Y
AND3
IN1
IN2
IN3
Y
AND3
IN1
IN2
Y
AND2
IN1
IN2
Y
AND2
IN1
IN2
Y
AND2
IN1
IN2
IN3
IN4
Y
OR4
IN1
IN2
IN3
Y
OR3
IN1
Y
SOFT
IN
OUT
OUTBUF
SUM
IN1
Y
SOFT
IN
OUT
OUTBUF
CY_OUT

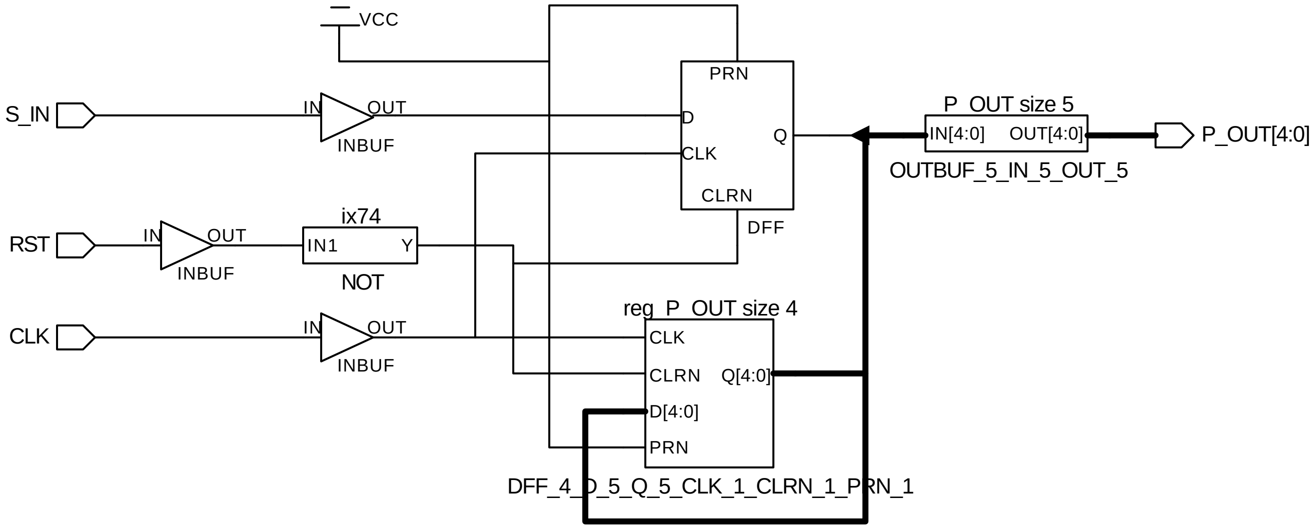

A_IN[3:0]
B_IN[3:0]
A_IN size 4
B_IN size 4
IN[3:0]
OUT[3:0]
INBUF_4_IN_4_OUT_4
INBUF_4_IN_4_OUT_4
IN1
IN2
IN3
IN4
Y
AND2
AND3
OR2
OR3
OR4
XOR2
SOFT
OUTBUF
OUT
ix225
NOT
ix227
NOT
ix228
NOT
ix231
NOT
ix238
NOT
ix240
NOT
ix237
EXP
ix239
EXP
ix226
NOT
ix229
NOT
ix230
NOT
ix242
NOT
ix244
NOT
ix241
EXP
ix243
EXP
OVF
NOT
OVF
SUM[3:0]

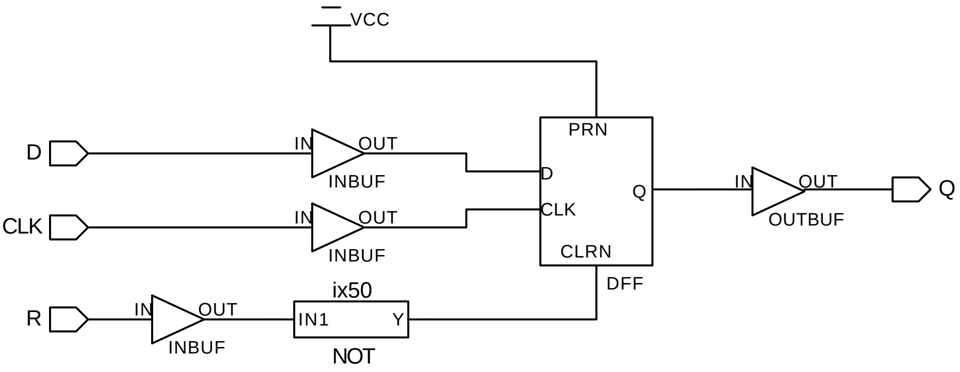

B
A
IN
OUT
INBUF
IN
OUT
INBUF
IN1
IN2
Y
AND2
IN1
IN2
Y
XOR2
IN1
Y
SOFT
IN1
Y
SOFT
IN
OUT
OUTBUF
IN
OUT
OUTBUF
CY_OUT
SUM

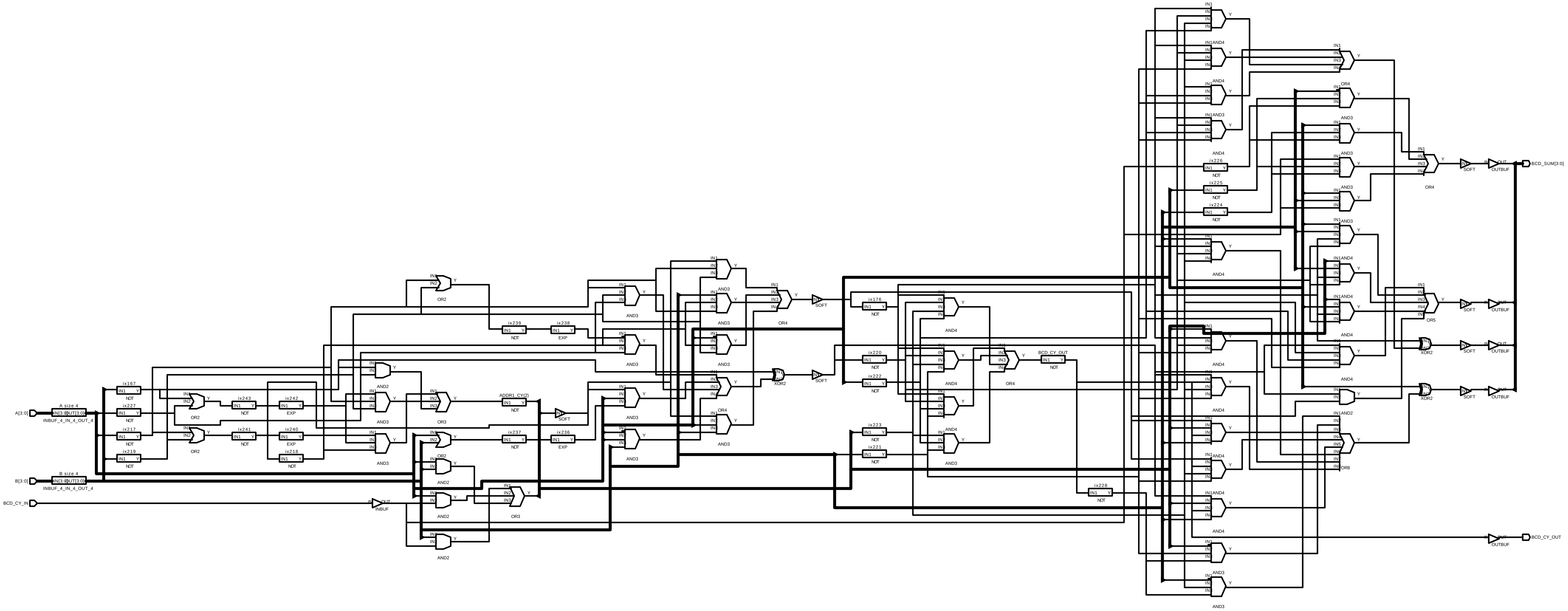


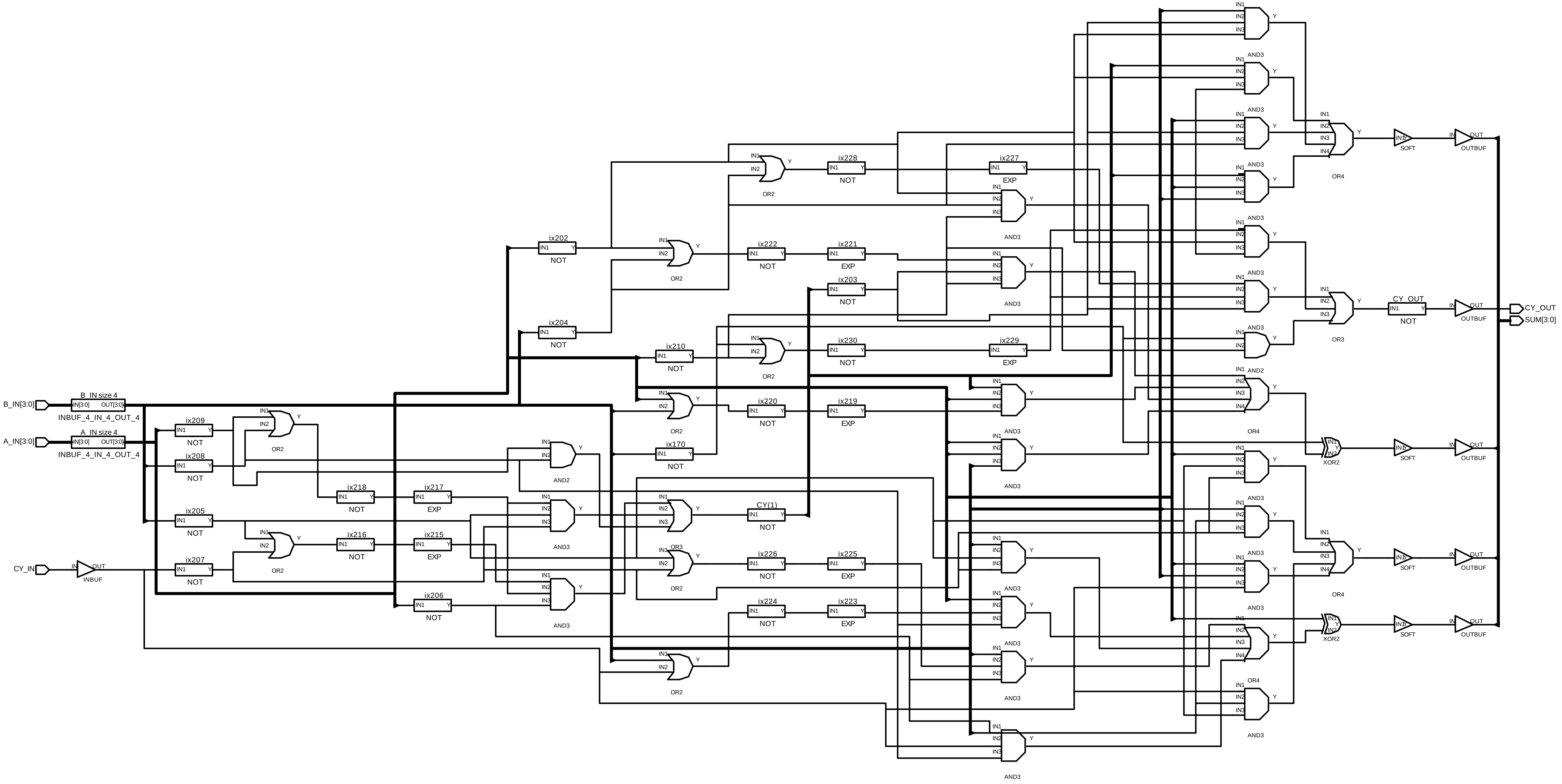
RST
A_IN[3:0]
A_IN size 4
IN[3:0] OUT[3:0]
INBUF_4_IN_4_OUT_4
CLK
DATA_LOAD
B_IN[3:0]
B_IN size 4
INBUF_4_IN_4_OUT_4
ix167
ix168
ix169
ix170
ix171
ix172
ix173
ix174
ix135
ix136
ix137
ix138
ix139
ix140
ix141
ix142
ix143
ix144
ix145
ix146
ix147
ix148
ix149
ix150
ix151
ix164
ix165
ix166
IN1
IN2
IN3
IN4
Y
NOT
AND2
AND3
OR3
OR4
PRN
D
CLK
CLRN
Q
DFF
GND
VCC
IN
OUT
INBUF
REG_S_reg_P_OUT size 4
CLK
CLRN Q[4:0]
D[4:0]
PRN
DFF_4_D_5_Q_5_CLK_1_CLRN_1_PRN_1
Q size 5
IN[4:0] OUT[4:0]
OUTBUF_5_IN_5_OUT_5
Q[4:0]

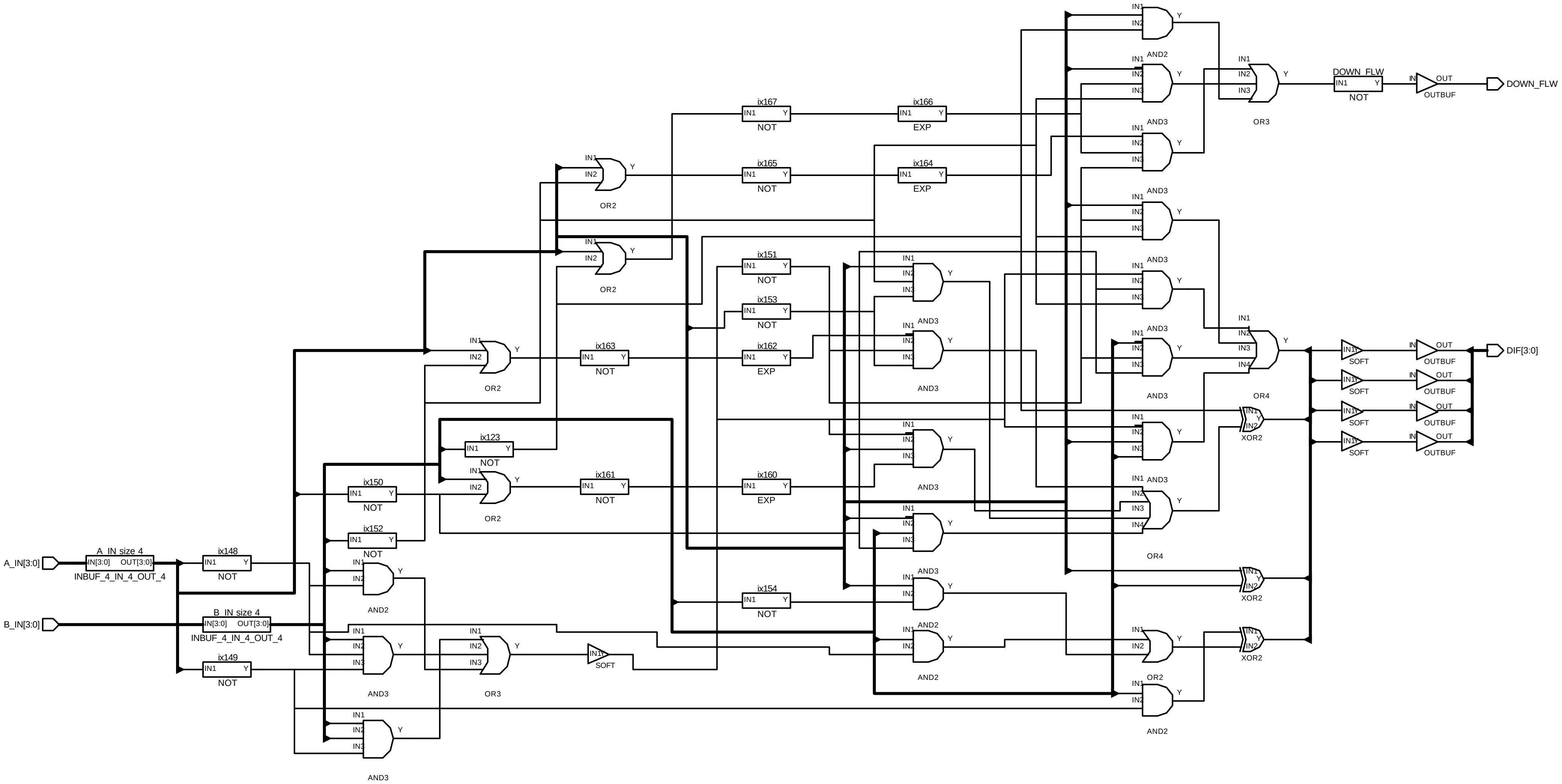

IN1
IN2
Y
AND2
ix97
IN1 Y
NOT
IN1
IN2
Y
AND2
ix98
IN1 Y
NOT
ix100
IN1 Y
NOT
PRN
D
CLK
Q
CLRN
DFF
IN
OUT
OUTBUF
S_OUT
IN1
IN2
Y
AND2
ix95
IN1 Y
NOT
PRN
D
CLK
Q
CLRN
DFF
IN1
IN2
Y
AND2
ix96
IN1 Y
NOT
ix101
IN1 Y
NOT
IN1
IN2
Y
AND2
ix93
IN1 Y
NOT
PRN
D
CLK
Q
CLRN
DFF
ix102
IN1 Y
NOT
IN1
IN2
Y
AND2
ix94
IN1 Y
NOT
IN1
IN2
Y
AND2
ix91
IN1 Y
NOT
GND
PRN
D
CLK
Q
CLRN
DFF
CLK
IN
OUT
INBUF
DATA_LOAD
IN
OUT
INBUF
P_IN[3:0]
P_IN size 4
IN[3:0] OUT[3:0]
INBUF_4_IN_4_OUT_4
ix103
IN1 Y
NOT
IN1
IN2
Y
AND2
ix92
IN1 Y
NOT

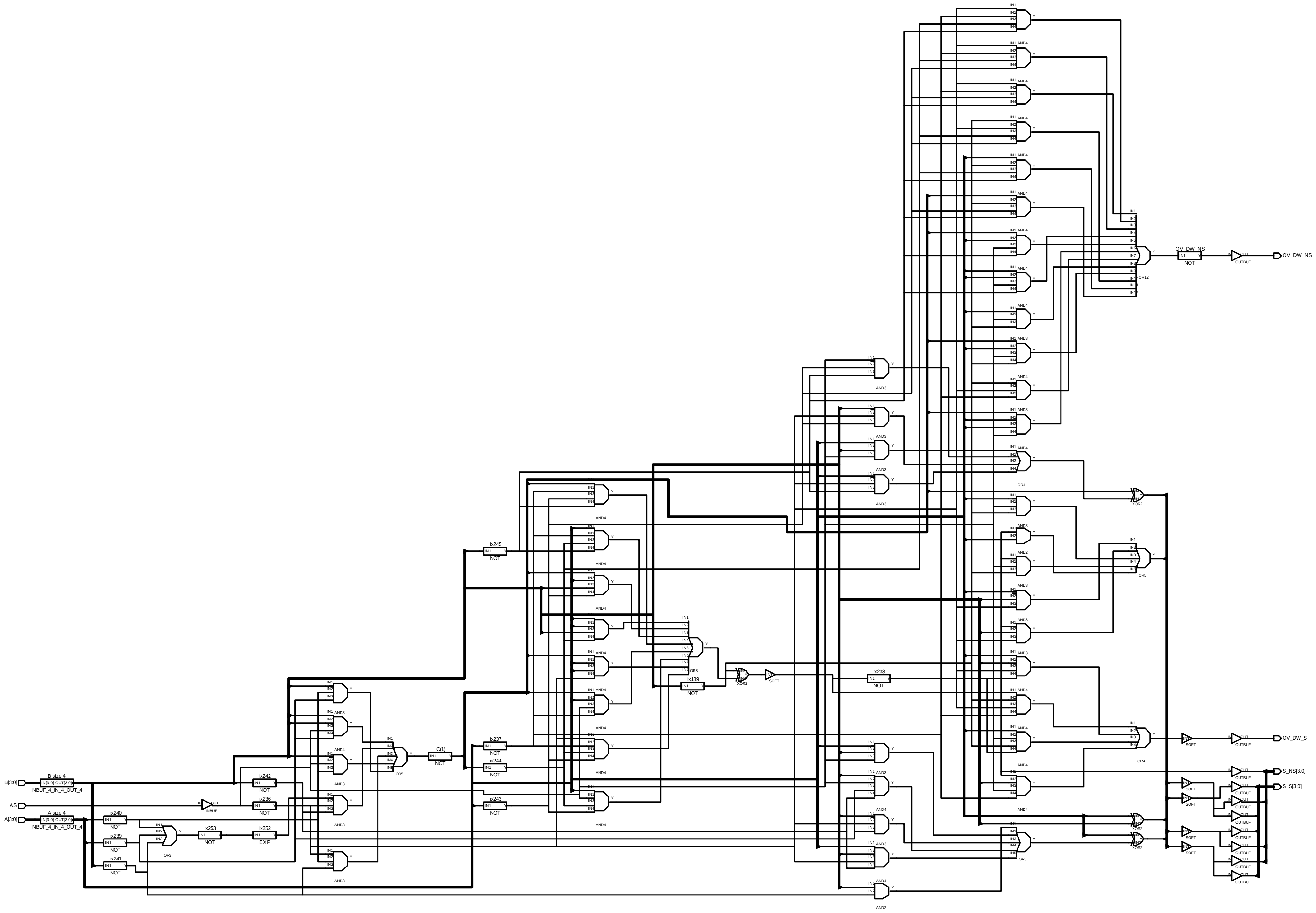

VCC
S_IN
IN
OUT
INBUF
PRN
D
CLK
Q
CLRN
DFF
P_OUT size 5
IN[4:0]
OUT[4:0]
P_OUT[4:0]
OUTBUF_5_IN_5_OUT_5
RST
IN
OUT
INBUF
ix74
IN1
Y
NOT
CLK
IN
OUT
INBUF
reg_P_OUT size 4
CLK
CLRN
Q[4:0]
D[4:0]
PRN
DFF_4_D_5_Q_5_CLK_1_CLRN_1_PRN_1


VCC
D
IN
OUT
INBUF
CLK
IN
OUT
INBUF
R
IN
OUT
INBUF
ix50
IN1
Y
NOT
PRN
D
CLK
Q
CLRN
DFF
IN
OUT
OUTBUF
Q

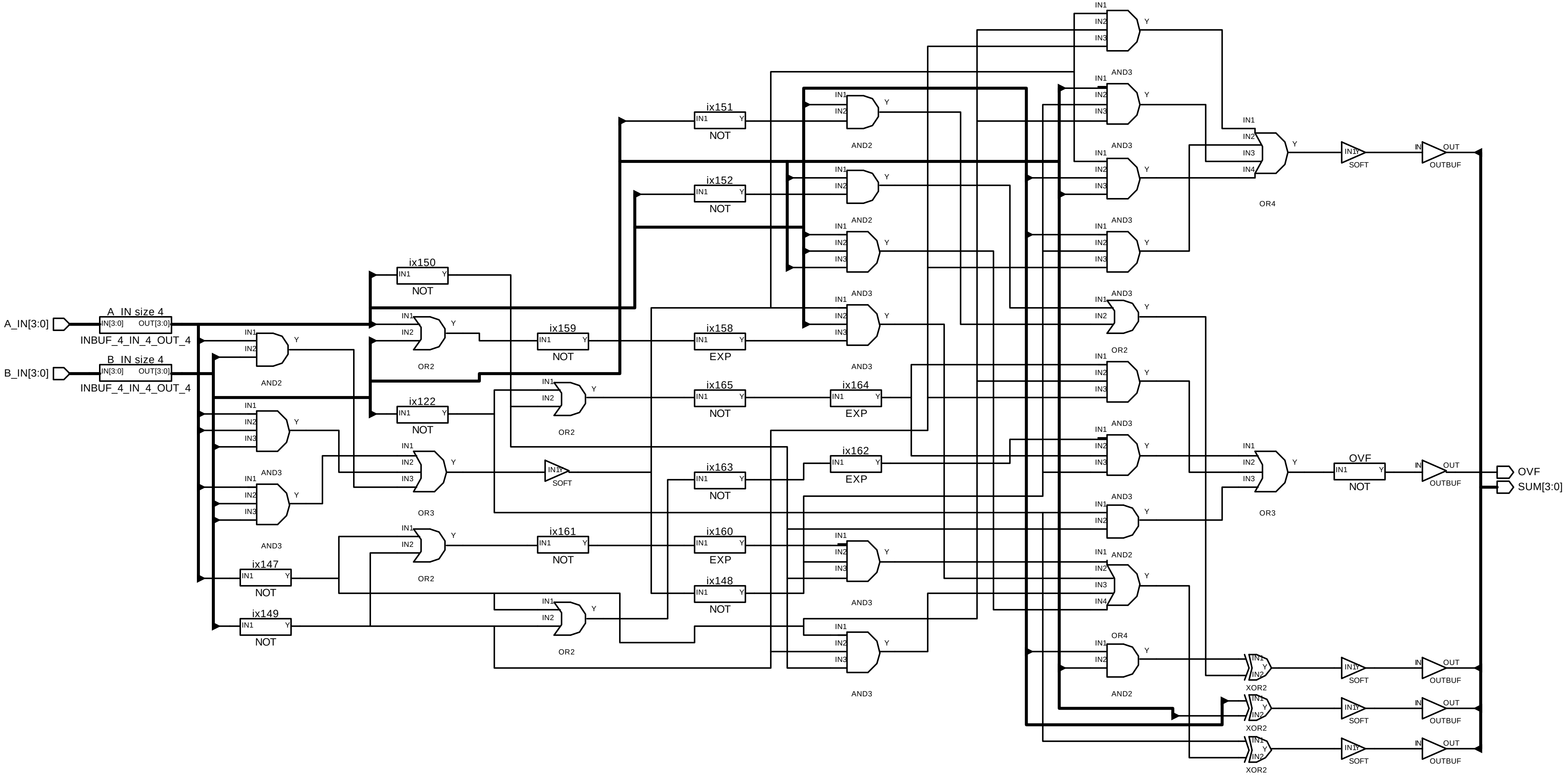

A[3:0]
A size 4
IN[3:0] OUT[3:0]
INBUF_4_IN_4_OUT_4
B[3:0]
B size 4
IN[3:0] OUT[3:0]
INBUF_4_IN_4_OUT_4
BCD_CY_IN
INBUF
ix167
ix227
ix217
ix219
ix243
ix242
ix241
ix240
ix218
ix239
ix238
ix237
ix236
ADDR1_CY(2)
ix176
ix220
ix222
ix223
ix221
BCD_CY_OUT
ix228
ix226
ix225
ix224
NOT
EXP
SOFT
OR2
OR3
OR4
OR5
OR8
AND2
AND3
AND4
XOR2
IN1
IN2
IN3
IN4
IN5
Y
OUT
OUTBUF
BCD_SUM[3:0]
BCD_CY_OUT

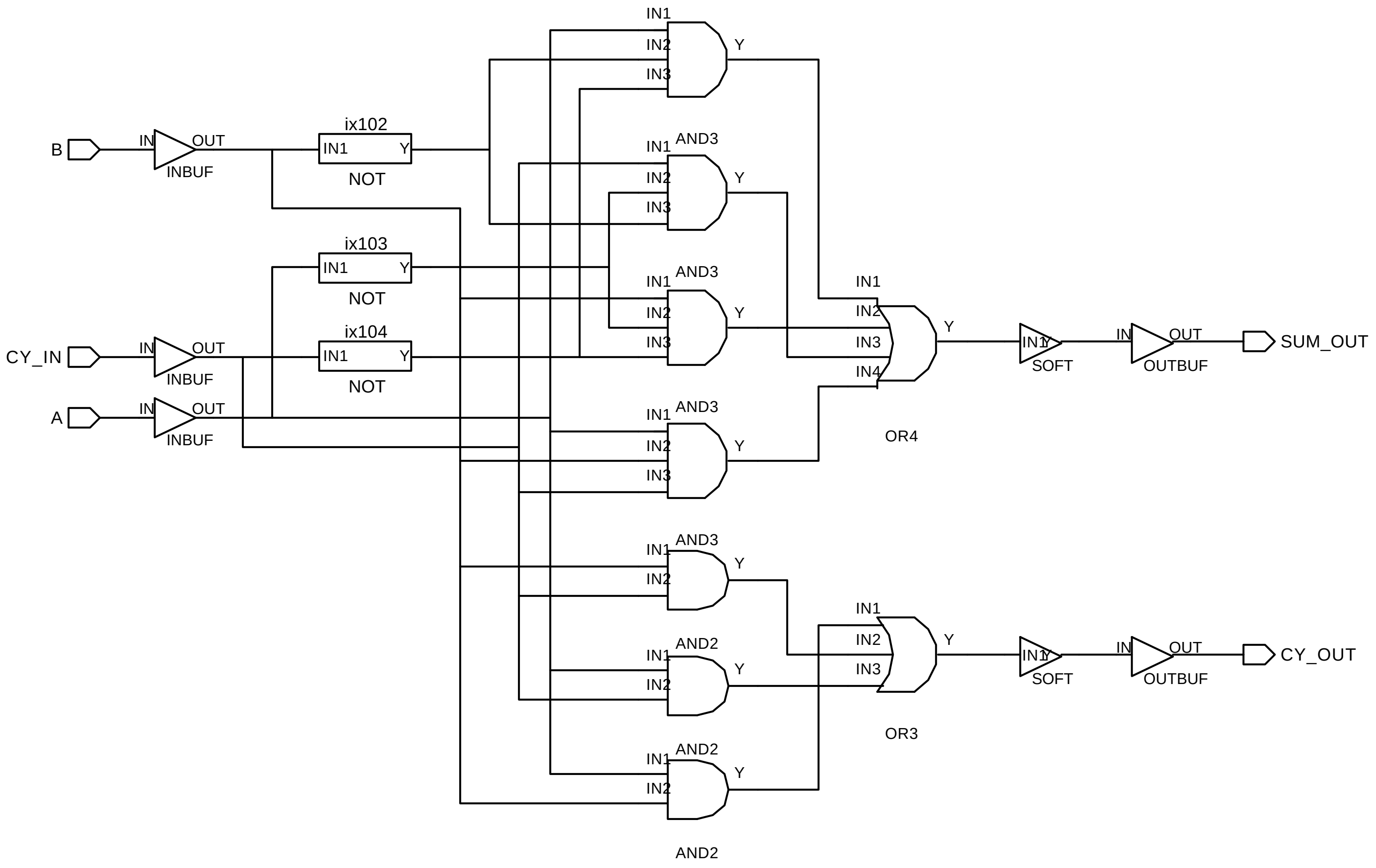

B_IN[3:0]
B_IN size 4
INBUF_4_IN_4_OUT_4
A_IN[3:0]
A_IN size 4
INBUF_4_IN_4_OUT_4
CY_IN
INBUF
ix209
ix208
ix205
ix207
ix218
ix217
ix216
ix215
ix206
ix202
ix204
ix210
ix170
ix228
ix227
ix222
ix221
ix203
ix230
ix229
ix220
ix219
CY(1)
ix226
ix225
ix224
ix223
NOT
EXP
OR2
OR3
AND2
AND3
OR4
XOR2
SOFT
OUTBUF
CY_OUT
SUM[3:0]

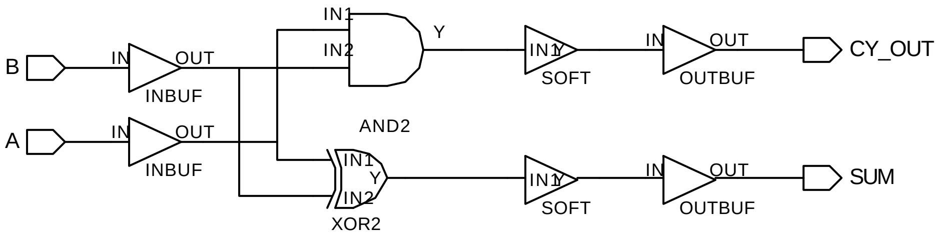

A_IN[3:0]
A_IN size 4
IN[3:0]
OUT[3:0]
INBUF_4_IN_4_OUT_4
B_IN[3:0]
B_IN size 4
IN[3:0]
OUT[3:0]
INBUF_4_IN_4_OUT_4
ix148
NOT
ix149
NOT
ix150
NOT
ix152
NOT
ix123
NOT
ix161
NOT
ix163
NOT
ix167
NOT
ix165
NOT
ix151
NOT
ix153
NOT
ix154
NOT
ix166
EXP
ix164
EXP
ix162
EXP
ix160
EXP
IN1
IN2
IN3
IN4
Y
AND2
AND3
OR2
OR3
OR4
XOR2
SOFT
OUTBUF
IN
OUT
DOWN_FLW
NOT
DOWN_FLW
DIF[3:0]

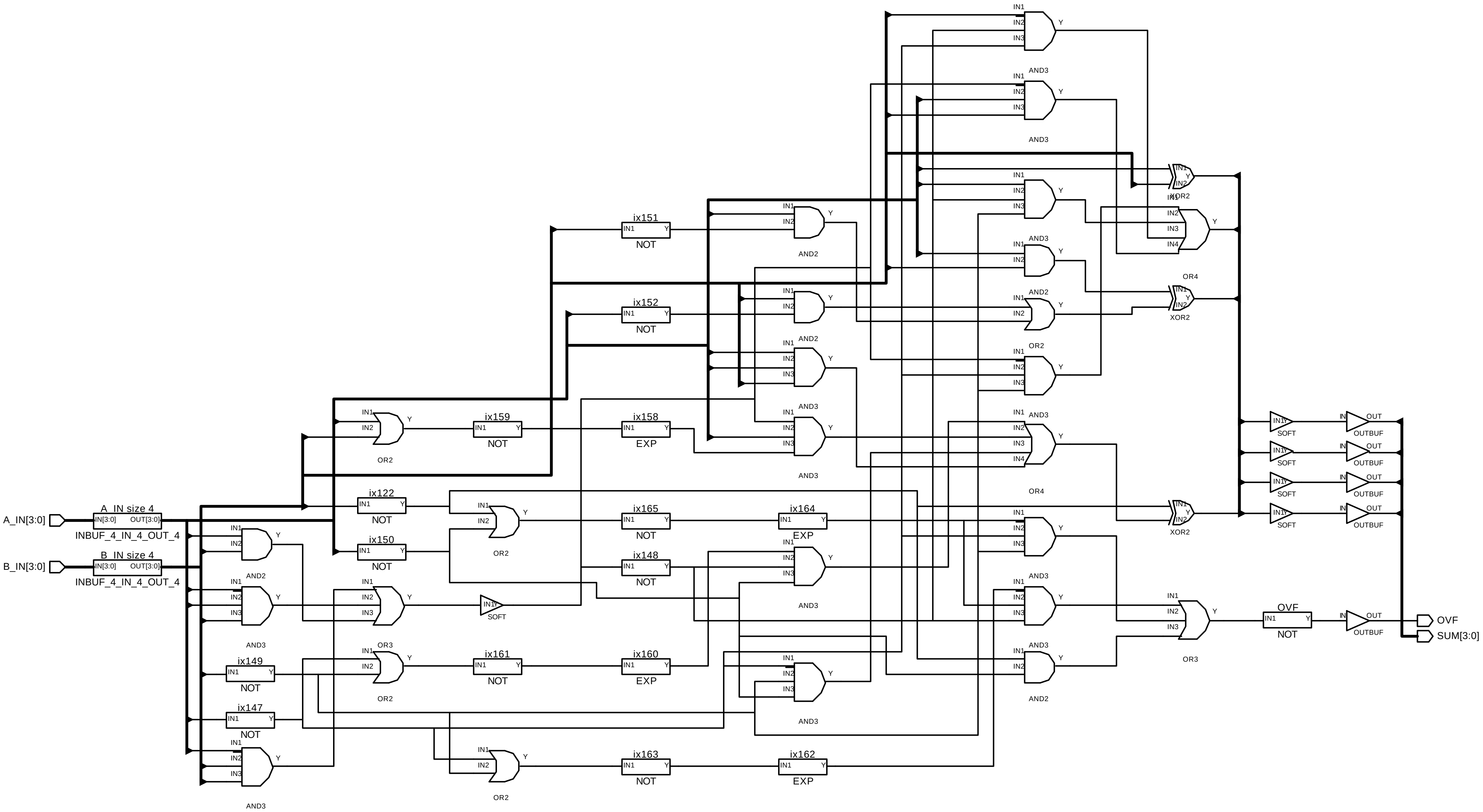


B[3:0]
B size 4
IN[3:0] OUT[3:0]
INBUF_4_IN_4_OUT_4
AS
A[3:0]
A size 4
IN[3:0] OUT[3:0]
INBUF_4_IN_4_OUT_4
IN OUT
INBUF
ix240
NOT
ix239
NOT
ix241
NOT
OR3
ix253
NOT
ix252
EXP
ix242
NOT
ix236
NOT
AND3
AND4
AND3
AND3
OR5
C(1)
NOT
ix245
NOT
ix237
NOT
ix244
NOT
ix243
NOT
AND4
OR8
ix189
NOT
XOR2
SOFT
ix238
NOT
AND3
AND2
OR12
OV_DW_NS
NOT
OUTBUF
OV_DW_NS
OR4
XOR2
OR5
OR4
SOFT
OUTBUF
OV_DW_S
S_NS[3:0]
S_S[3:0]
OR5
XOR2
SOFT
OUTBUF
IN1
IN2
IN3
IN4
Y

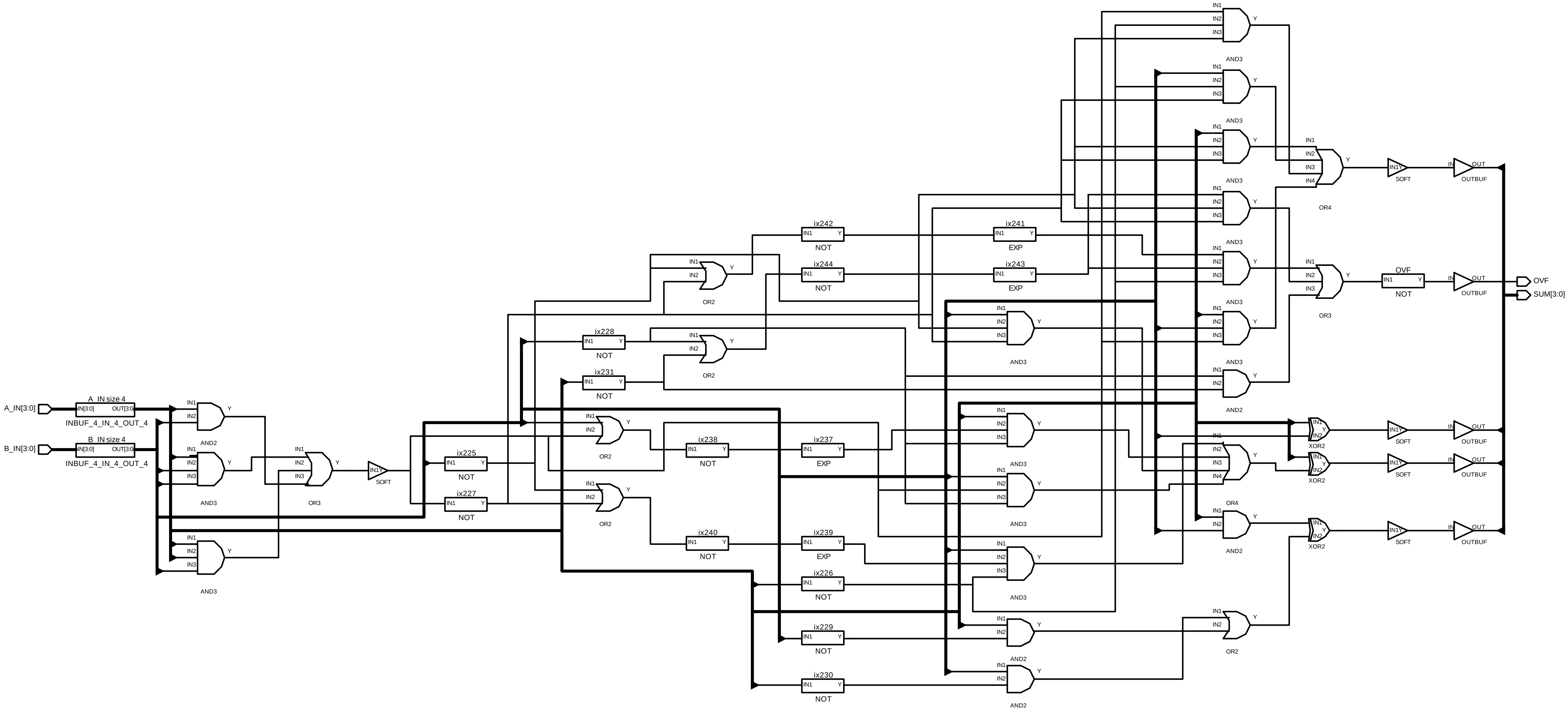

A_IN[3:0]
A_IN size 4
IN[3:0]
OUT[3:0]
INBUF_4_IN_4_OUT_4
B_IN[3:0]
B_IN size 4
IN[3:0]
OUT[3:0]
INBUF_4_IN_4_OUT_4
AND2
AND3
AND3
ix147
NOT
ix149
NOT
ix150
NOT
OR2
ix122
NOT
OR3
OR2
ix159
NOT
OR2
SOFT
ix161
NOT
OR2
ix151
NOT
ix152
NOT
ix158
EXP
ix165
NOT
ix163
NOT
ix160
EXP
ix148
NOT
AND2
AND2
AND3
AND3
ix164
EXP
ix162
EXP
AND3
AND3
AND3
AND3
AND3
AND3
OR2
AND3
AND3
AND2
OR4
AND2
OR4
OR3
OVF
NOT
SOFT
OUTBUF
XOR2
XOR2
XOR2
IN1
IN2
IN3
IN4
Y
IN
OUT
OVF
SUM[3:0]

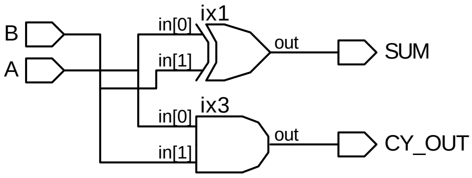

B
IN
OUT
INBUF
ix102
IN1
Y
NOT
CY_IN
IN
OUT
INBUF
ix103
IN1
Y
NOT
ix104
IN1
Y
NOT
A
IN
OUT
INBUF
IN1
IN2
IN3
Y
AND3
IN1
IN2
IN3
Y
AND3
IN1
IN2
IN3
Y
AND3
IN1
IN2
IN3
Y
AND3
IN1
IN2
Y
AND2
IN1
IN2
Y
AND2
IN1
IN2
Y
AND2
IN1
IN2
IN3
IN4
Y
OR4
IN1
IN2
IN3
Y
OR3
IN1
Y
SOFT
IN
OUT
OUTBUF
SUM_OUT
IN1
Y
SOFT
IN
OUT
OUTBUF
CY_OUT

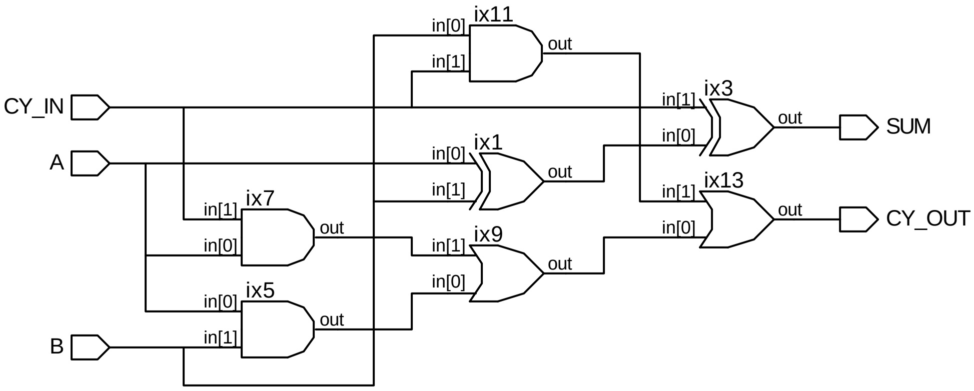

B
IN
OUT
INBUF
A
IN
OUT
INBUF
IN1
IN2
Y
AND2
IN1
IN2
Y
XOR2
IN1
Y
SOFT
IN1
Y
SOFT
IN
OUT
OUTBUF
IN
OUT
OUTBUF
CY_OUT
SUM

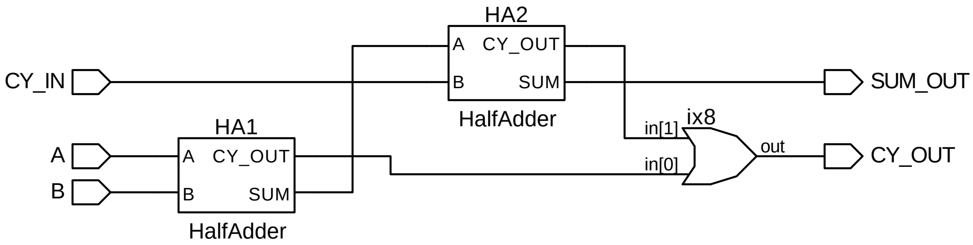


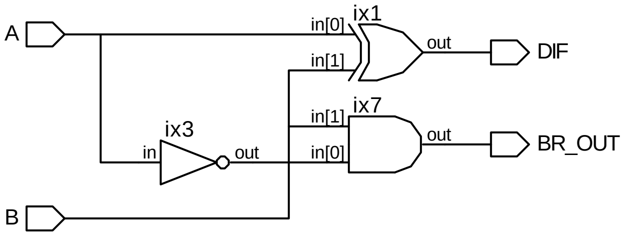
A_IN[3:0]
B_IN[3:0]
A_IN size 4
B_IN size 4
IN[3:0]
OUT[3:0]
INBUF_4_IN_4_OUT_4
INBUF_4_IN_4_OUT_4
AND2
AND3
AND3
OR2
OR3
OR2
SOFT
ix122
ix150
ix149
ix147
ix159
ix161
ix151
ix152
ix158
ix165
ix148
ix160
ix163
ix164
ix162
NOT
EXP
OR2
AND2
AND3
OR4
XOR2
OR3
OVF
OUTBUF
SOFT
IN1
IN2
IN3
IN4
Y
OUT
OVF
SUM[3:0]

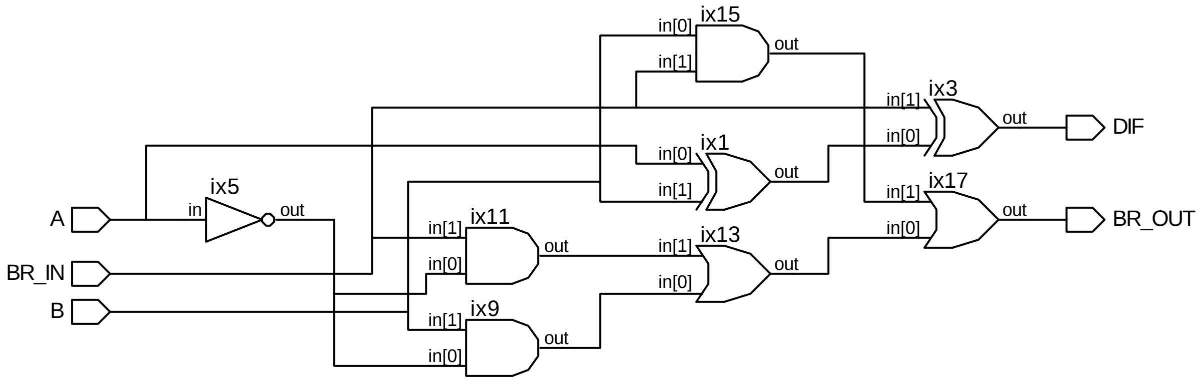

A_IN[3:0]
A_IN size 4
IN[3:0]
OUT[3:0]
INBUF_4_IN_4_OUT_4
B_IN[3:0]
B_IN size 4
IN[3:0]
OUT[3:0]
INBUF_4_IN_4_OUT_4
IN1
IN2
IN3
IN4
Y
AND2
AND3
OR3
SOFT
ix225
NOT
ix227
NOT
ix228
NOT
ix231
NOT
OR2
ix238
NOT
ix237
EXP
ix240
NOT
ix239
EXP
ix226
NOT
ix229
NOT
ix230
NOT
ix242
NOT
ix244
NOT
ix241
EXP
ix243
EXP
OR4
OR3
OVF
NOT
XOR2
OUT
OUTBUF
OVF
SUM[3:0]

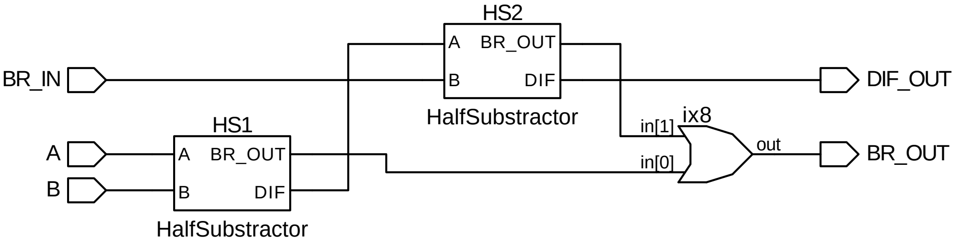

B
A
ix1
in[0]
in[1]
out
SUM
ix3
in[0]
in[1]
out
CY_OUT

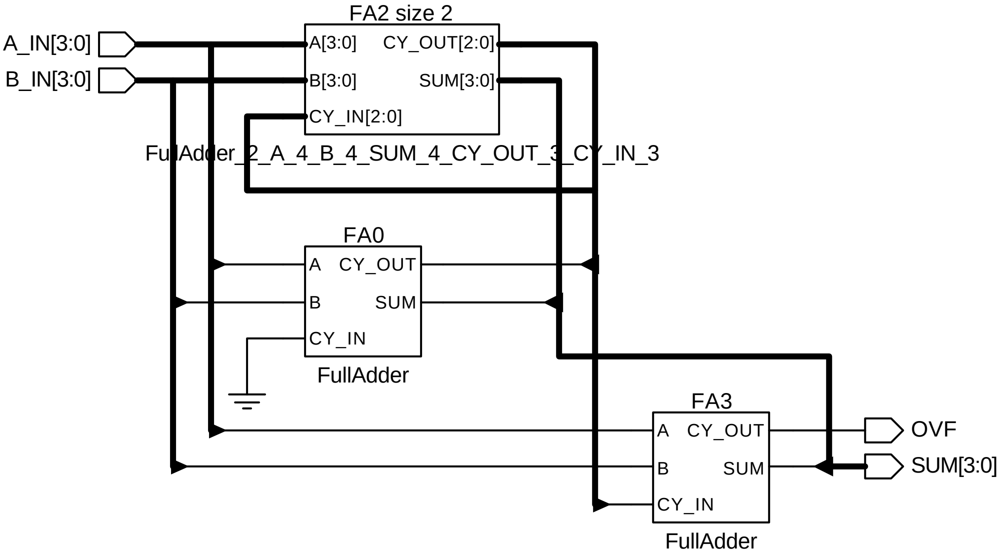

ix11
in[0]
in[1]
out
CY_IN
in[1]
ix3
in[0]
out
SUM
A
in[0]
ix1
in[1]
out
in[1]
ix13
in[0]
out
CY_OUT
ix7
in[1]
in[0]
out
in[1]
ix9
in[0]
out
ix5
in[0]
in[1]
out
B

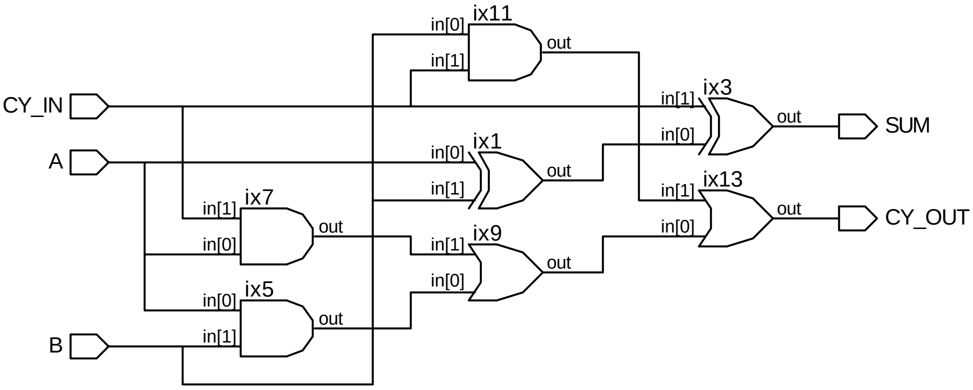

HA2
A
CY_OUT
B
SUM
HalfAdder
CY_IN
SUM_OUT
HA1
A
CY_OUT
B
SUM
HalfAdder
A
B
in[1]
ix8
in[0]
out
CY_OUT

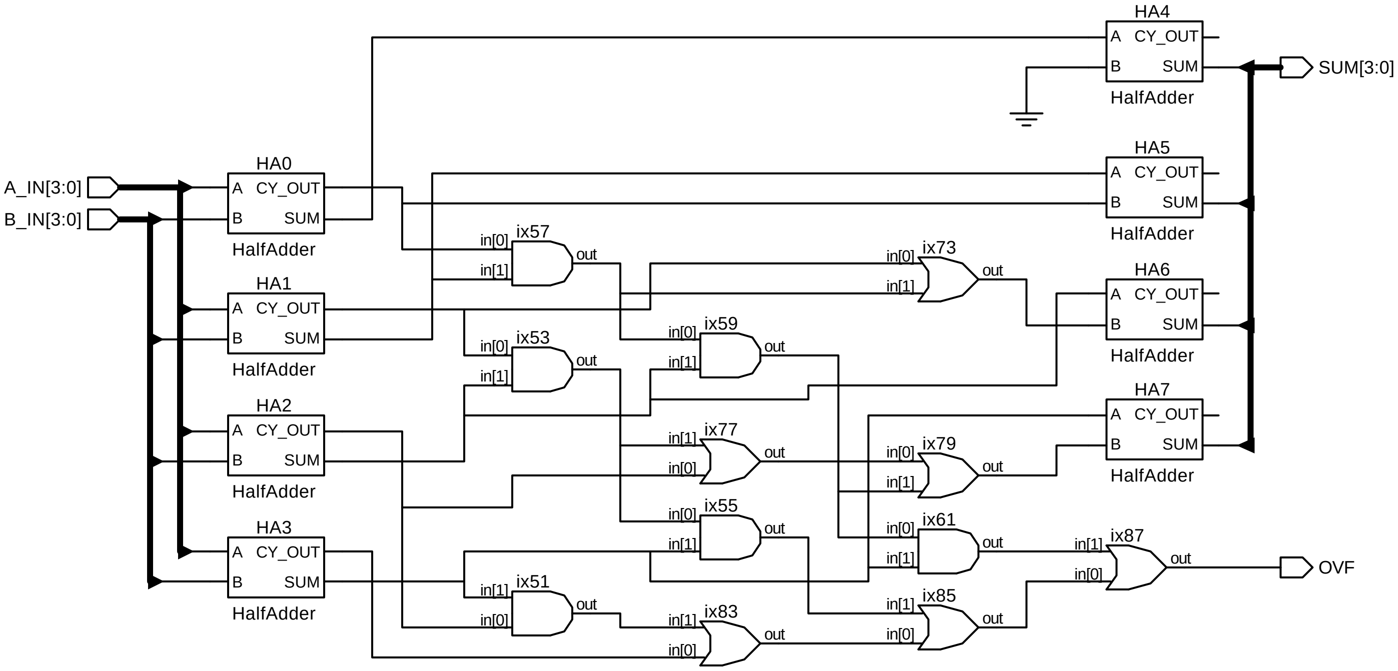

A
in[0]
ix1
out
DIF
in[1]
ix7
in[1]
out
BR_OUT
ix3
in
out
in[0]
B

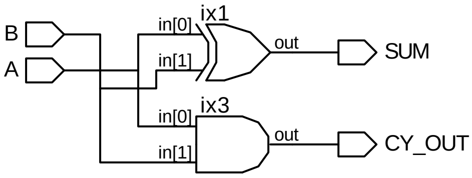

ix15
in[0]
in[1]
out
ix3
in[1]
in[0]
out
DIF
ix1
in[0]
in[1]
out
ix17
in[1]
in[0]
out
BR_OUT
A
ix5
in
out
ix11
in[1]
in[0]
out
ix13
in[1]
in[0]
out
BR_IN
B
ix9
in[1]
in[0]
out

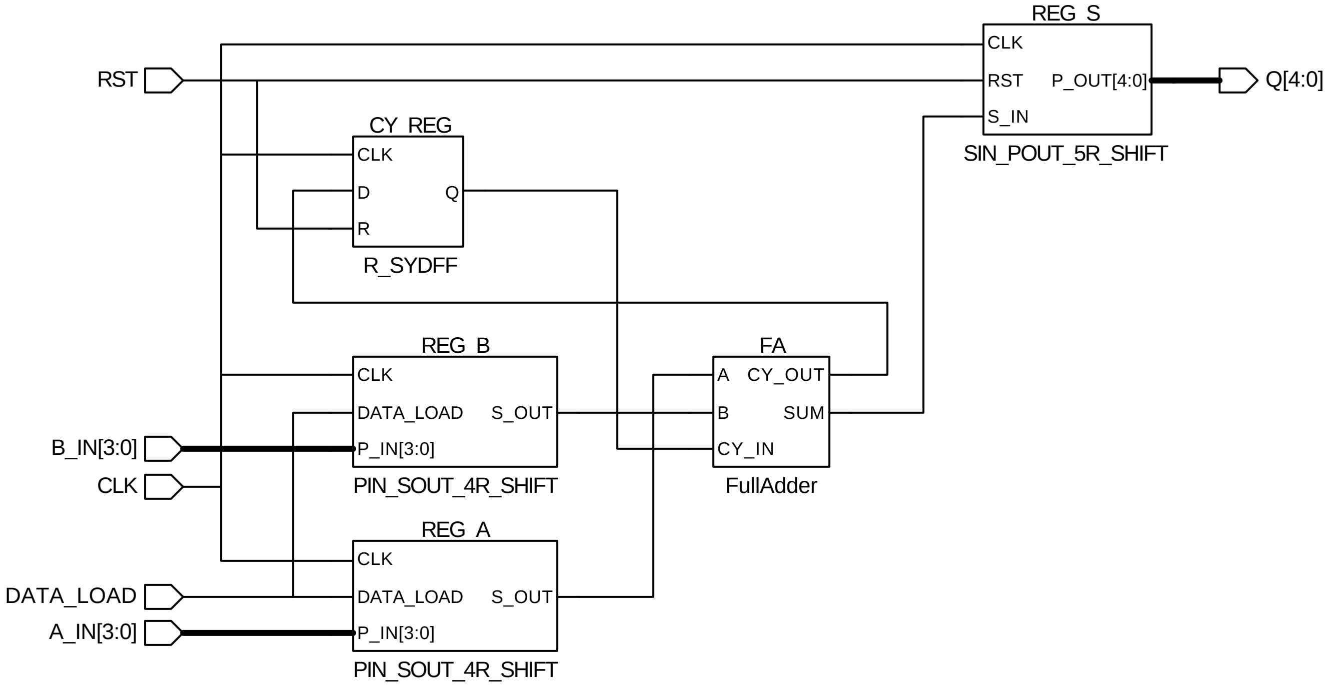

HS2
A
BR_OUT
B
DIF
HalfSubstractor
BR_IN
DIF_OUT
HS1
A
BR_OUT
B
DIF
HalfSubstractor
A
B
in[1]
ix8
in[0]
out
BR_OUT

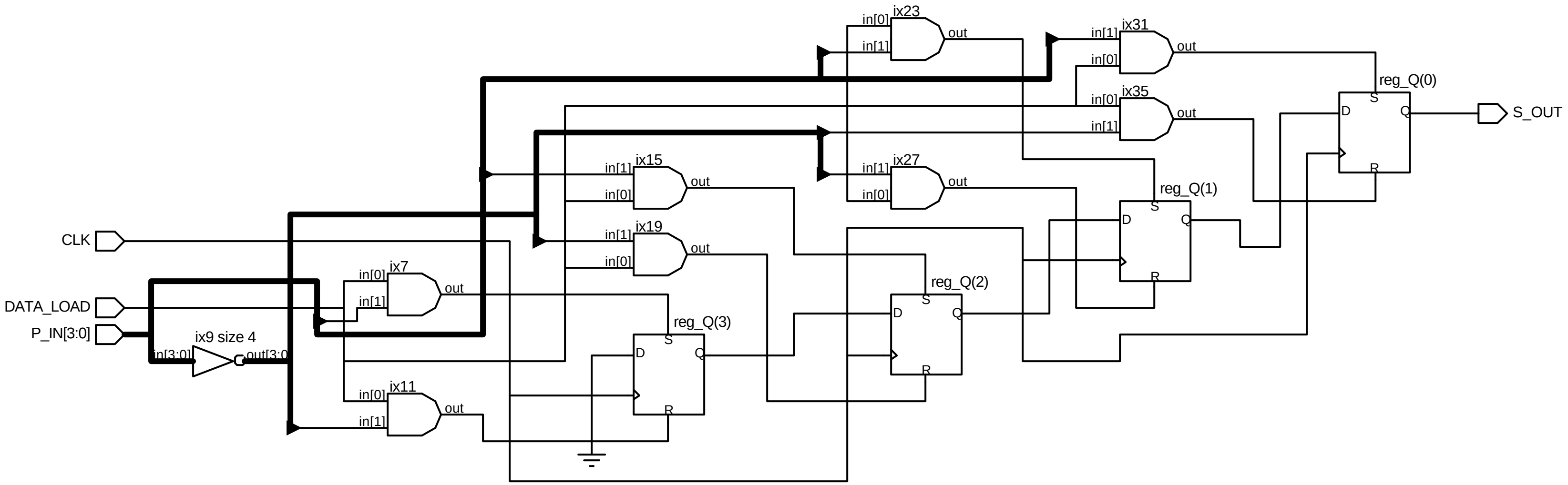

A_IN[3:0]
B_IN[3:0]
FA2 size 2
A[3:0]
B[3:0]
CY_IN[2:0]
CY_OUT[2:0]
SUM[3:0]
FullAdder_2_A_4_B_4_SUM_4_CY_OUT_3_CY_IN_3
FA0
A
B
CY_IN
CY_OUT
SUM
FullAdder
FA3
A
B
CY_IN
CY_OUT
SUM
FullAdder
OVF
SUM[3:0]

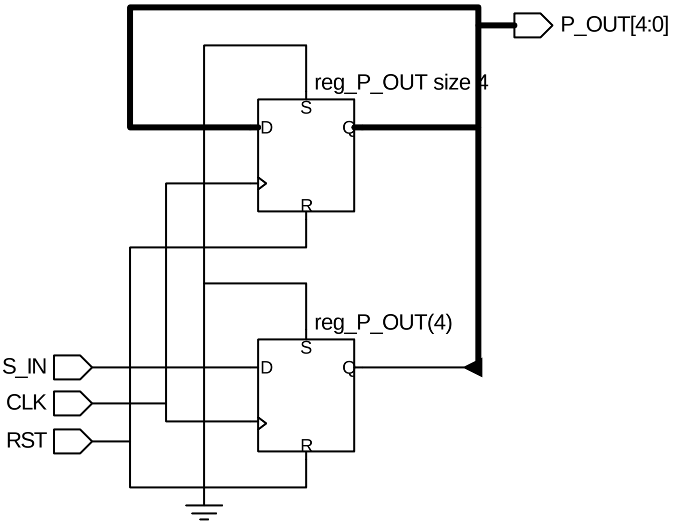

CY_IN
A
B
ix11
in[0]
in[1]
out
ix3
in[1]
in[0]
out
SUM
ix1
in[0]
in[1]
out
ix13
in[1]
in[0]
out
CY_OUT
ix7
in[1]
in[0]
out
ix9
in[1]
in[0]
out
ix5
in[0]
in[1]
out

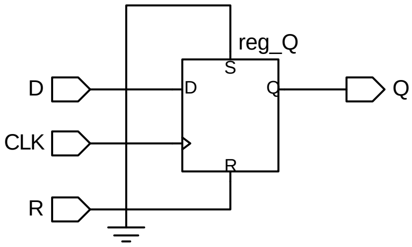

A_IN[3:0]
B_IN[3:0]
HA0
HA1
HA2
HA3
HA4
HA5
HA6
HA7
A
B
CY_OUT
SUM
HalfAdder
ix51
ix53
ix55
ix57
ix59
ix61
ix73
ix77
ix79
ix83
ix85
ix87
in[0]
in[1]
out
SUM[3:0]
OVF

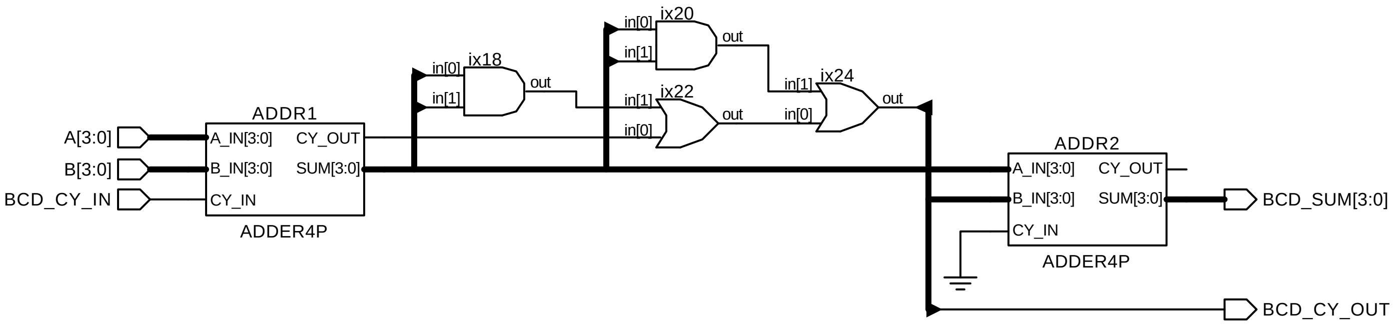

B
A
ix1
in[0]
in[1]
out
SUM
ix3
in[0]
in[1]
out
CY_OUT

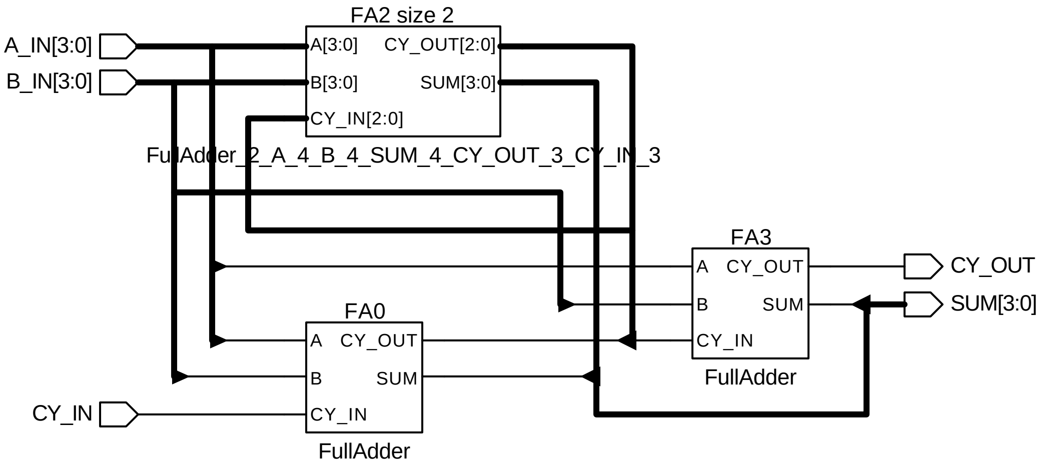

RST
B_IN[3:0]
CLK
DATA_LOAD
A_IN[3:0]
CY_REG
CLK
D
Q
R
R_SYDFF
REG_B
CLK
DATA_LOAD
S_OUT
P_IN[3:0]
PIN_SOUT_4R_SHIFT
REG_A
CLK
DATA_LOAD
S_OUT
P_IN[3:0]
PIN_SOUT_4R_SHIFT
FA
A
CY_OUT
B
SUM
CY_IN
FullAdder
REG_S
CLK
RST
P_OUT[4:0]
S_IN
SIN_POUT_5R_SHIFT
Q[4:0]

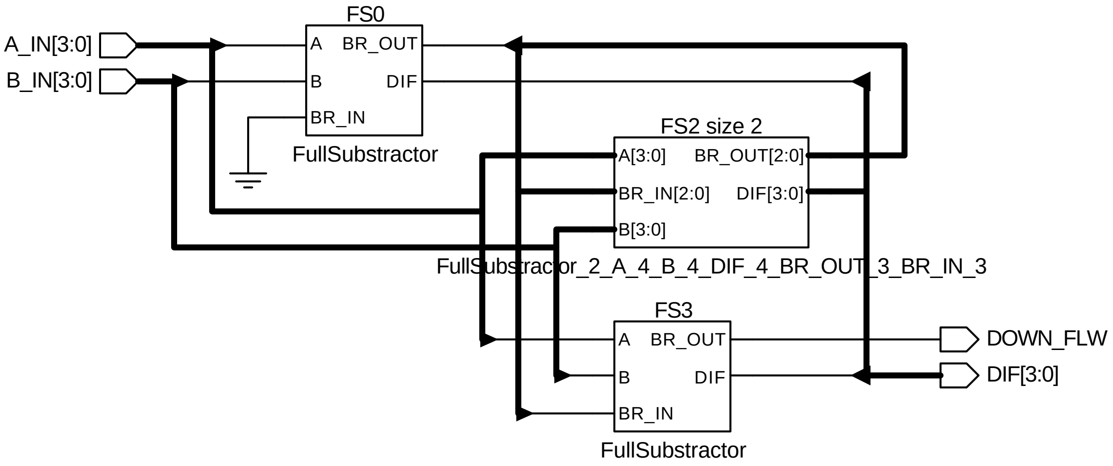

ix23
in[0]
in[1]
out
ix31
in[1]
in[0]
out
reg_Q(0)
ix35
in[0]
in[1]
out
D
S
Q
R
S_OUT
ix15
in[1]
in[0]
out
ix27
in[1]
in[0]
out
reg_Q(1)
D
S
Q
R
ix19
in[1]
in[0]
out
CLK
ix7
in[0]
in[1]
out
reg_Q(2)
D
S
Q
R
DATA_LOAD
P_IN[3:0]
ix9 size 4
in[3:0]
out[3:0]
reg_Q(3)
D
S
Q
R
ix11
in[0]
in[1]
out

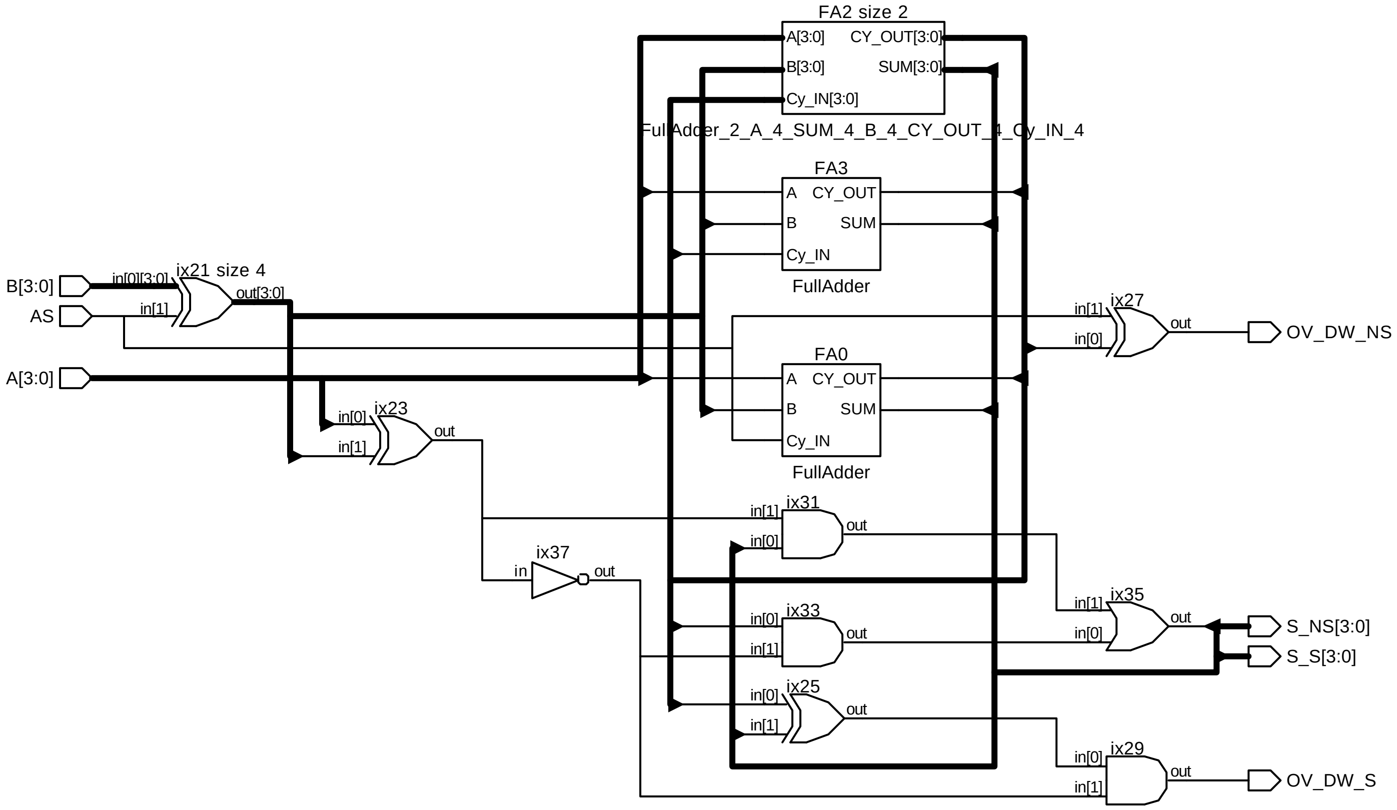

P_OUT[4:0]
reg_P_OUT size 4
S
D
Q
R
reg_P_OUT(4)
S
D
Q
R
S_IN
CLK
RST

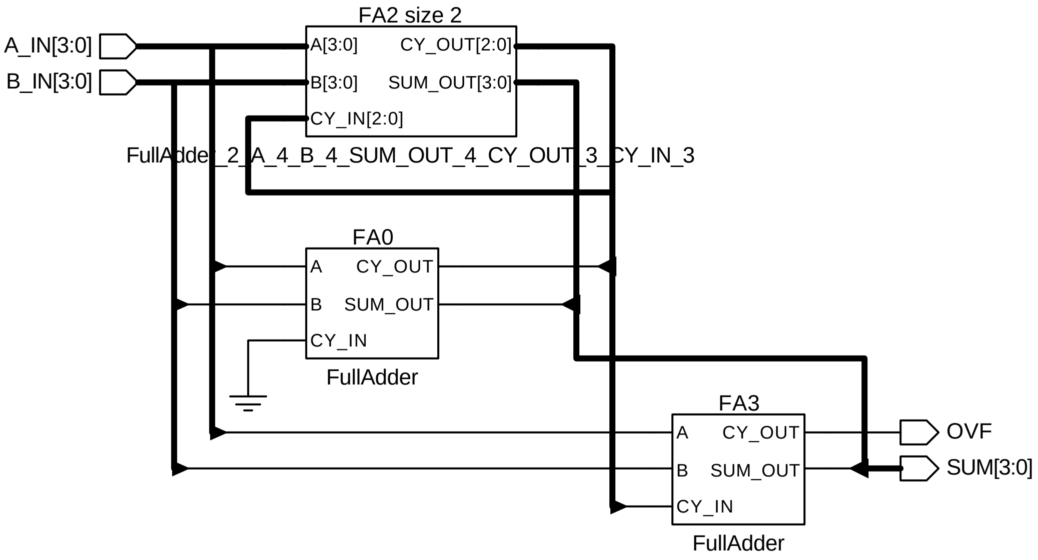

reg_Q
D
CLK
R
S
D
Q
R
Q

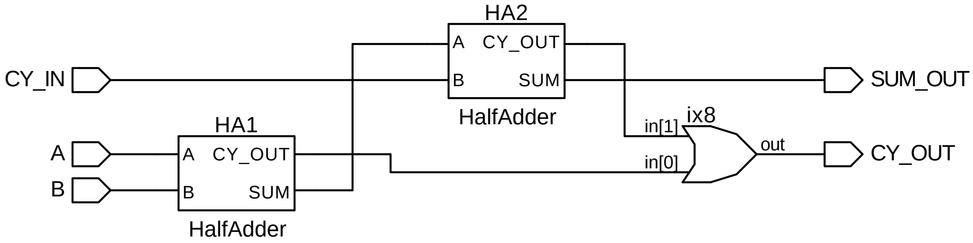

A[3:0]
B[3:0]
BCD_CY_IN
ADDR1
A_IN[3:0]
CY_OUT
B_IN[3:0]
SUM[3:0]
CY_IN
ADDER4P
ix18
in[0]
in[1]
out
ix20
in[0]
in[1]
out
ix22
in[1]
in[0]
out
ix24
in[1]
in[0]
out
ADDR2
A_IN[3:0]
CY_OUT
B_IN[3:0]
SUM[3:0]
CY_IN
ADDER4P
BCD_SUM[3:0]
BCD_CY_OUT

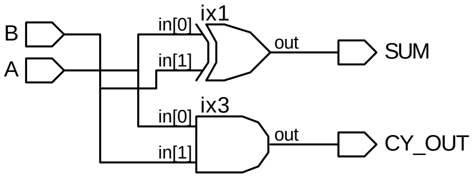

FA2 size 2
A_IN[3:0]
B_IN[3:0]
A[3:0]
B[3:0]
CY_IN[2:0]
CY_OUT[2:0]
SUM[3:0]
FullAdder_2_A_4_B_4_SUM_4_CY_OUT_3_CY_IN_3
FA3
A
B
CY_IN
CY_OUT
SUM
CY_OUT
SUM[3:0]
FullAdder
FA0
A
B
CY_IN
CY_OUT
SUM
CY_IN
FullAdder

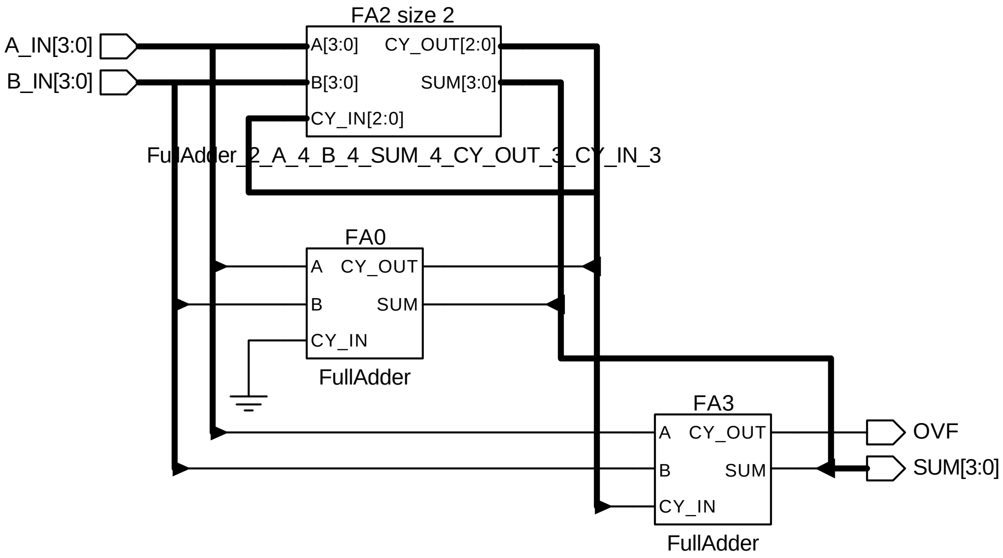

A_IN[3:0]
B_IN[3:0]
FS0
A
BR_OUT
B
DIF
BR_IN
FullSubstractor
FS2 size 2
A[3:0]
BR_OUT[2:0]
BR_IN[2:0]
DIF[3:0]
B[3:0]
FullSubstractor_2_A_4_B_4_DIF_4_BR_OUT_3_BR_IN_3
FS3
A
BR_OUT
B
DIF
BR_IN
FullSubstractor
DOWN_FLW
DIF[3:0]

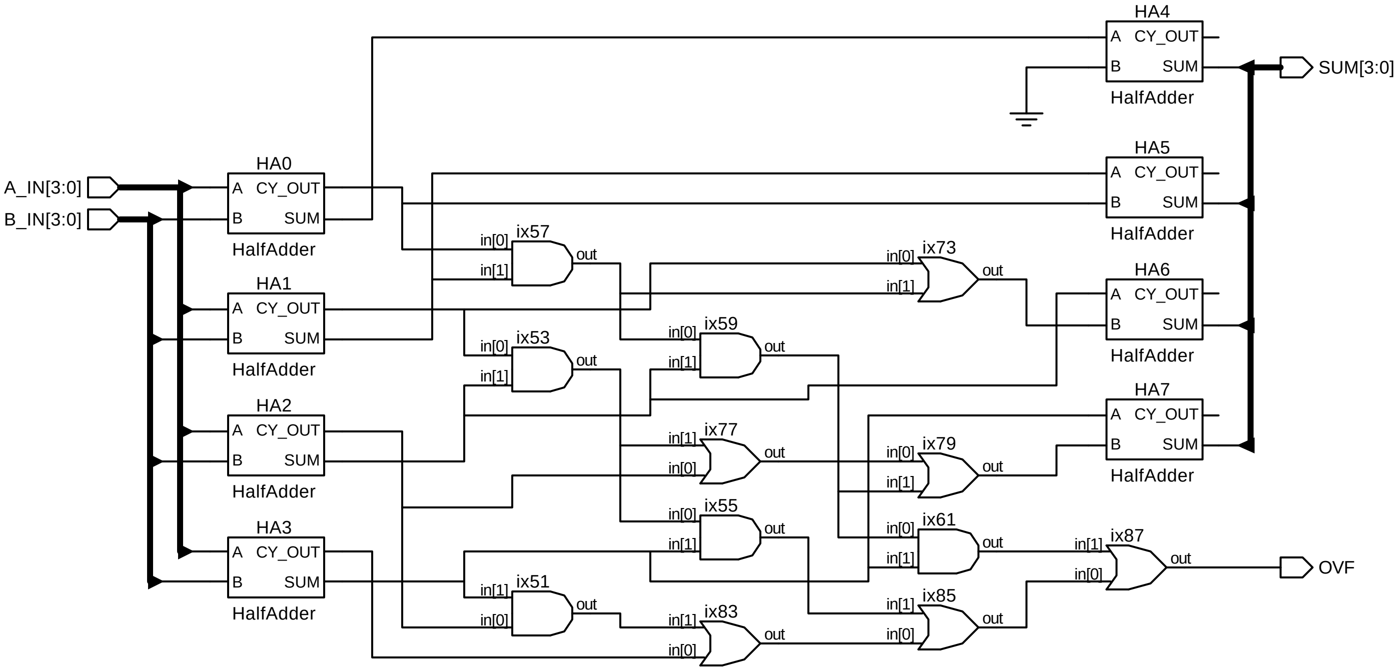

FA2 size 2
A[3:0]
B[3:0]
Cy_IN[3:0]
CY_OUT[3:0]
SUM[3:0]
FullAdder_2_A_4_SUM_4_B_4_CY_OUT_4_Cy_IN_4
FA3
A
CY_OUT
B
SUM
Cy_IN
FullAdder
B[3:0]
AS
in[0][3:0]
ix21 size 4
in[1]
out[3:0]
in[1]
ix27
in[0]
out
OV_DW_NS
FA0
A
CY_OUT
B
SUM
Cy_IN
FullAdder
A[3:0]
in[0]
ix23
in[1]
out
in[1]
ix31
in[0]
out
ix37
in
out
in[1]
ix35
in[0]
out
S_NS[3:0]
S_S[3:0]
in[0]
ix33
in[1]
out
in[0]
ix25
in[1]
out
in[0]
ix29
in[1]
out
OV_DW_S


A_IN[3:0]
B_IN[3:0]
FA2 size 2
A[3:0]
B[3:0]
CY_IN[2:0]
CY_OUT[2:0]
SUM_OUT[3:0]
FullAdder_2_A_4_B_4_SUM_OUT_4_CY_OUT_3_CY_IN_3
FA0
A
B
CY_IN
CY_OUT
SUM_OUT
FullAdder
FA3
A
B
CY_IN
CY_OUT
SUM_OUT
FullAdder
OVF
SUM[3:0]


HA2
A
CY_OUT
B
SUM
HalfAdder
CY_IN
SUM_OUT
HA1
A
CY_OUT
B
SUM
HalfAdder
A
B
in[1]
ix8
in[0]
out
CY_OUT


B
A
ix1
in[0]
in[1]
out
SUM
ix3
in[0]
in[1]
out
CY_OUT


A_IN[3:0]
B_IN[3:0]
FA2 size 2
A[3:0]
B[3:0]
CY_IN[2:0]
CY_OUT[2:0]
SUM[3:0]
FullAdder_2_A_4_B_4_SUM_4_CY_OUT_3_CY_IN_3
FA0
A
B
CY_IN
CY_OUT
SUM
FullAdder
FA3
A
B
CY_IN
CY_OUT
SUM
FullAdder
OVF
SUM[3:0]


A_IN[3:0]
B_IN[3:0]
HA0
HA1
HA2
HA3
HA4
HA5
HA6
HA7
A
B
CY_OUT
SUM
HalfAdder
ix51
ix53
ix55
ix57
ix59
ix61
ix73
ix77
ix79
ix83
ix85
ix87
in[0]
in[1]
out
SUM[3:0]
OVF