```
/*      AND_2   */
module  AND_2   ( IN1, IN2, OUT );
input   IN1, IN2;
output  OUT;
        and     U1      ( OUT, IN1, IN2 );
endmodule
```

```
/*      AND_2   */
module  AND_2   ( IN1, IN2, OUT );
input   IN1, IN2;
output  OUT;
        assign  OUT = IN1 & IN2;
endmodule
```

```
/*      OR_3    */
module  OR_3    ( IN1, IN2, IN3, OUT );
input   IN1, IN2, IN3;
output  OUT;
        or      U2      ( OUT, IN1, IN2, IN3 );
endmodule
```

```
/*      OR_3    */
module  OR_3    ( IN1, IN2, IN3, OUT );
input   IN1, IN2, IN3;
output  OUT;
        assign  OUT = IN1 | IN2 | IN3;
endmodule
```

```
/*      NAND_2  */
module  NAND_2  ( IN1, IN2, OUT );
input   IN1, IN2;
output  OUT;
        nand    U3      ( OUT, IN1, IN2 );
endmodule
```

```
/*      NAND_2  */
module  NAND_2  ( IN1, IN2, OUT );
input   IN1, IN2;
output  OUT;
        assign  OUT = ~( IN1 & IN2 );
endmodule
```

```
/*      NOR_2   */
module  NOR_2   ( IN1, IN2, OUT );
input   IN1, IN2;
output  OUT;
        nor     U4      ( OUT, IN1, IN2 );
endmodule
```

```
/*      NOR_2   */
module  NOR_2   ( IN1, IN2, OUT );
input   IN1, IN2;
output  OUT;
        assign  OUT = ~( IN1 | IN2 );
endmodule
```

```
/*      NOT     */
module  NOT     ( IN, OUT );
input   IN;
output  OUT;
        not     U5      ( OUT, IN );
endmodule
```

```
/*      NOT     */
module  NOT     ( IN, OUT );
input   IN;
output  OUT;
        assign  OUT = ~IN;
endmodule
```

```
/*      BUF     */
module  BUF     ( IN, OUT );
input   IN;
output  OUT;
        buf     U6      ( OUT, IN );
endmodule
```

```
/*      BUF     */
module  BUF     ( IN, OUT );
input   IN;
output  OUT;
        assign  OUT = IN;
endmodule
```

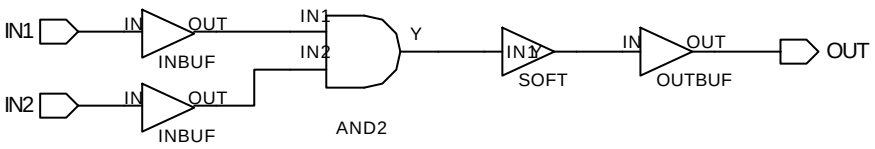
```
/*      GATE    */
module GATE     (IN_1, IN_2, IN_3,
                        OUT_AND, OUT_OR, OUT_NAND, OUT_NOR, OUT_NOT, OUT_BUF );
input   IN_1, IN_2, IN_3;
output  OUT_AND, OUT_OR, OUT_NAND, OUT_NOR, OUT_NOT, OUT_BUF;

        AND_2   AND_2   ( IN_1, IN_2, OUT_AND );
        OR_3    OR_3    ( IN_1, IN_2, IN_3, OUT_OR );
        NAND_2  NAND_2  ( IN_1, IN_2, OUT_NAND );
        NOR_2   NOR_2   ( IN_1, IN_2, OUT_NOR );
        NOT     NOT     ( IN_1, OUT_NOT );
        BUF     BUF     ( IN_1, OUT_BUF );
endmodule

/*      AND_2   */
module  AND_2   ( IN1, IN2, OUT );
input   IN1, IN2;
output  OUT;
        assign  OUT = IN1 & IN2;
endmodule

/*      OR_3    */
module  OR_3    ( IN1, IN2, IN3, OUT );
input   IN1, IN2, IN3;
output  OUT;
        assign  OUT = IN1 | IN2 | IN3;
endmodule

/*      NAND_2  */
module  NAND_2  ( IN1, IN2, OUT );
input   IN1, IN2;
output  OUT;
        assign  OUT = ~( IN1 & IN2 );
endmodule
```

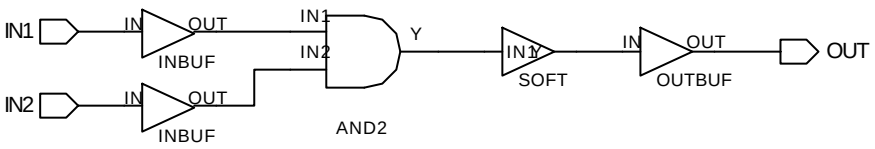
```
/*      NOR_2   */
module  NOR_2   ( IN1, IN2, OUT );
input   IN1, IN2;
output  OUT;
        assign  OUT = ~( IN1 | IN2 );
endmodule

/*      NOT     */
module  NOT     ( IN, OUT );
input   IN;
output  OUT;
        assign  OUT = ~IN;
endmodule

/*      BUF     */
module  BUF     ( IN, OUT );
input   IN;
output  OUT;
        assign  OUT = IN;
endmodule
```

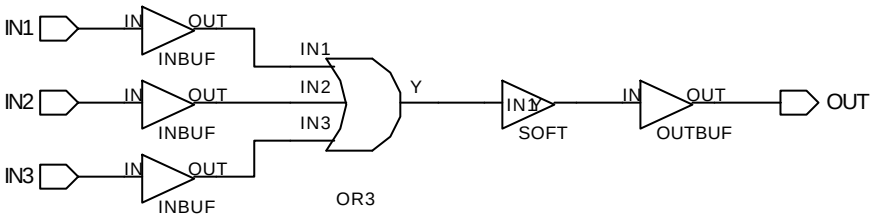
```
/*      DE_MORGAN_1     */
module  DE_MORGAN_1     ( A, B, OUT_LHS, OUT_RHS );
input   A, B;
output  OUT_LHS, OUT_RHS;
        assign  OUT_LHS = ~( A & B );
        assign  OUT_RHS =  ~A | ~B;
endmodule
```

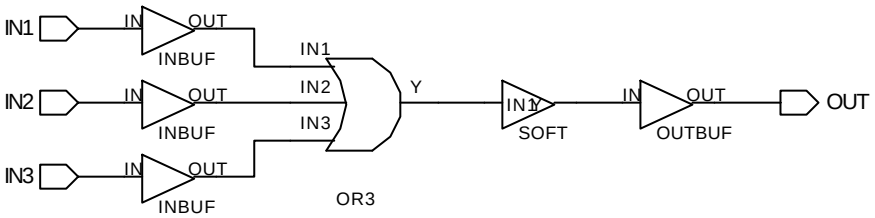
```
/*      DE_MORGAN_2     */
module  DE_MORGAN_2     ( A, B, OUT_LHS, OUT_RHS );
input   A, B;
output  OUT_LHS, OUT_RHS;
        assign  OUT_LHS = ~( A | B );
        assign  OUT_RHS =  ~A & ~B;
endmodule
```

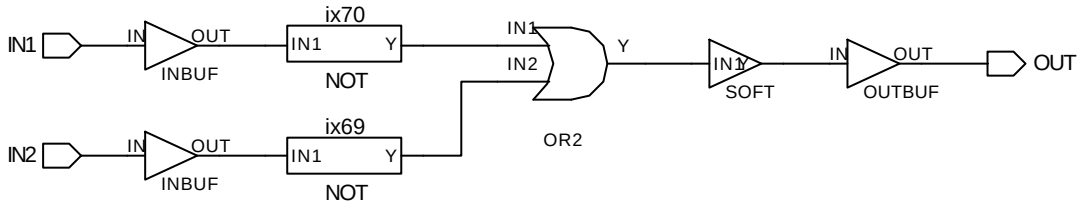
```
/*      PROBLEM_2       */
module  PROBLEM_2       ( A, B, C, OUT );
input   A, B, C;
output  OUT;
        assign  OUT = ( ~C & A ) | ( C & B ) | ( C & ~A );
endmodule
```

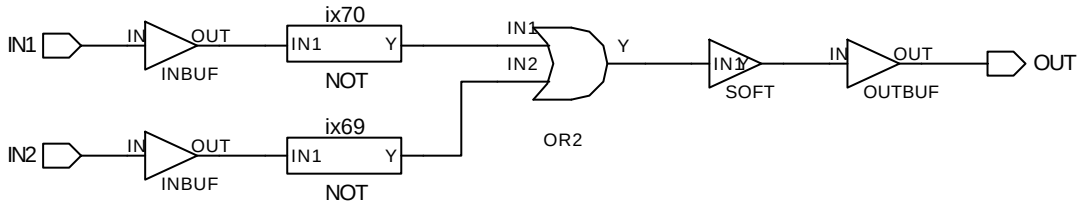

IN1
IN
OUT
INBUF
IN2
IN
OUT
INBUF
IN1
IN2
Y
AND2
IN1
Y
SOFT
IN
OUT
OUTBUF
OUT

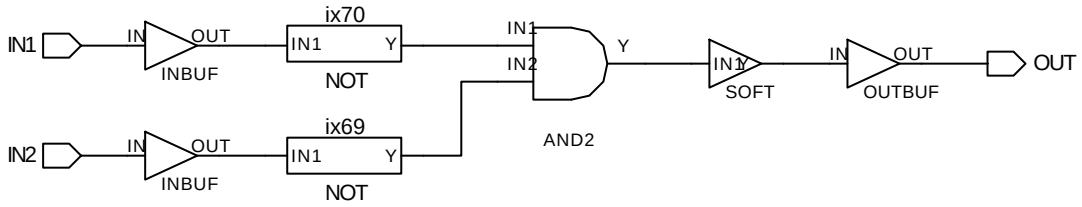

IN1
IN
OUT
INBUF
IN2
IN
OUT
INBUF
IN1
IN2
Y
AND2
IN1
Y
SOFT
IN
OUT
OUTBUF
OUT

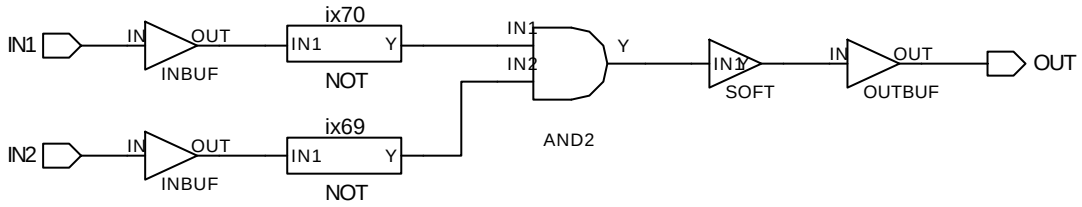

IN1
IN2
IN3
IN
OUT
INBUF
IN1
IN2
IN3
Y
OR3
IN1
Y
SOFT
IN
OUT
OUTBUF
OUT

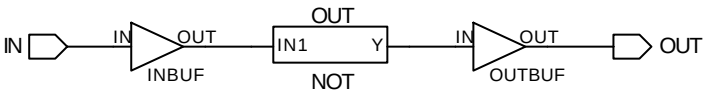

IN1
IN2
IN3
IN
OUT
INBUF
IN1
IN2
IN3
Y
OR3
IN1
Y
SOFT
IN
OUT
OUTBUF
OUT

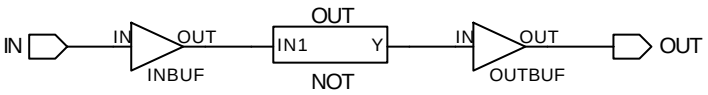

IN1
IN
OUT
INBUF
ix70
IN1
Y
NOT
IN2
IN
OUT
INBUF
ix69
IN1
Y
NOT
IN1
IN2
Y
OR2
IN1
Y
SOFT
IN
OUT
OUTBUF
OUT

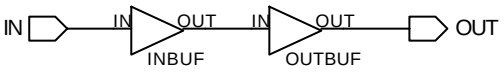

IN1
IN
OUT
INBUF
ix70
IN1
Y
NOT
IN2
IN
OUT
INBUF
ix69
IN1
Y
NOT
IN1
IN2
Y
OR2
IN1
Y
SOFT
IN
OUT
OUTBUF
OUT

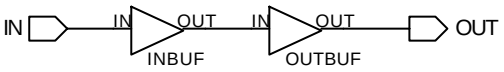

IN1
IN
OUT
INBUF
ix70
IN1
Y
NOT
IN1
IN2
Y
AND2
IN1
Y
SOFT
IN
OUT
OUTBUF
OUT
IN2
IN
OUT
INBUF
ix69
IN1
Y
NOT

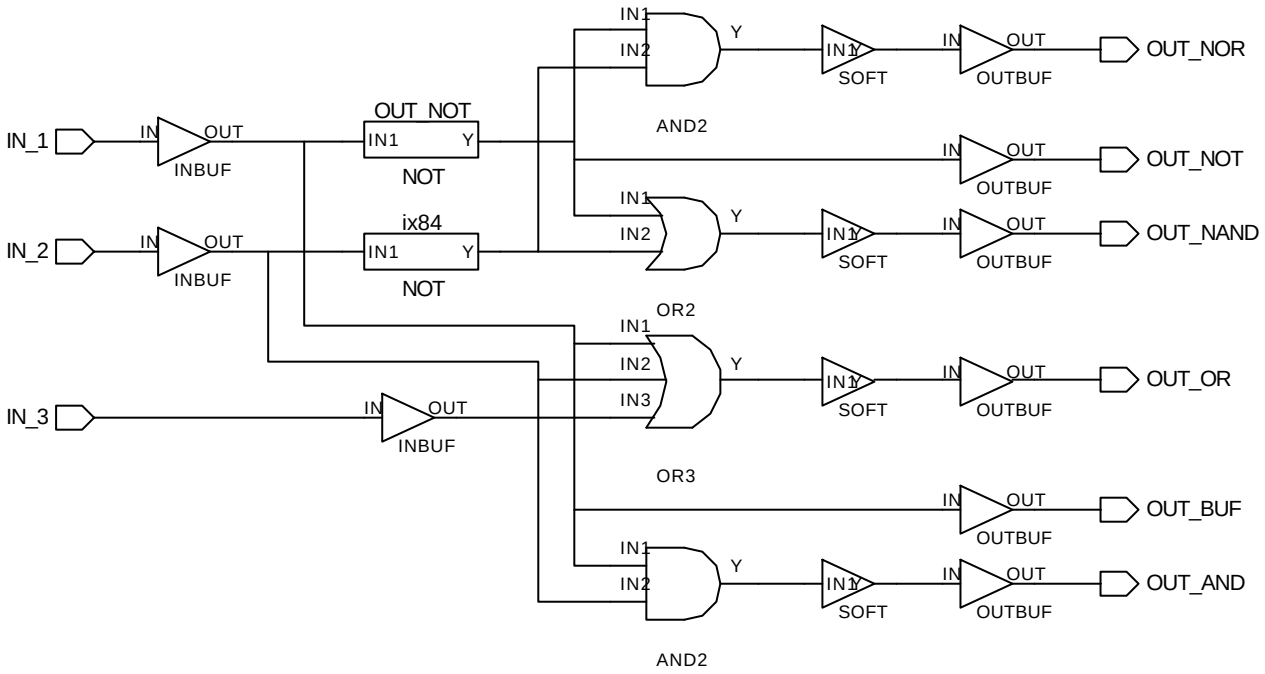

IN1
IN
OUT
INBUF
ix70
IN1
Y
NOT
IN1
IN2
Y
AND2
IN1
Y
SOFT
IN
OUT
OUTBUF
OUT
IN2
IN
OUT
INBUF
ix69
IN1
Y
NOT

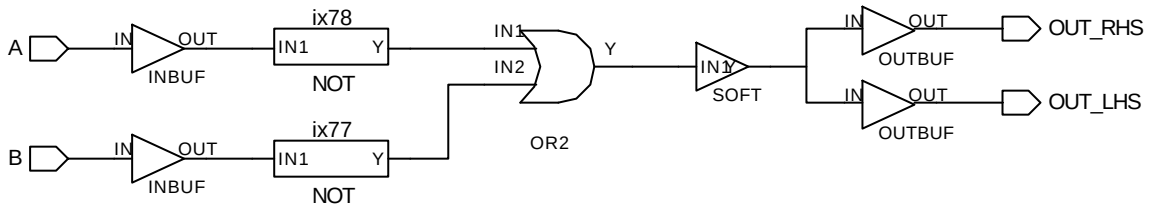

IN
IN
OUT
INBUF
OUT
IN1
Y
NOT
IN
OUT
OUTBUF
OUT

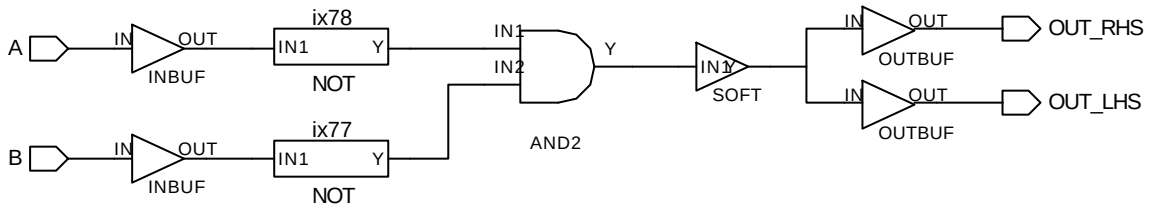

IN
IN
OUT
INBUF
OUT
IN1
Y
NOT
IN
OUT
OUTBUF
OUT

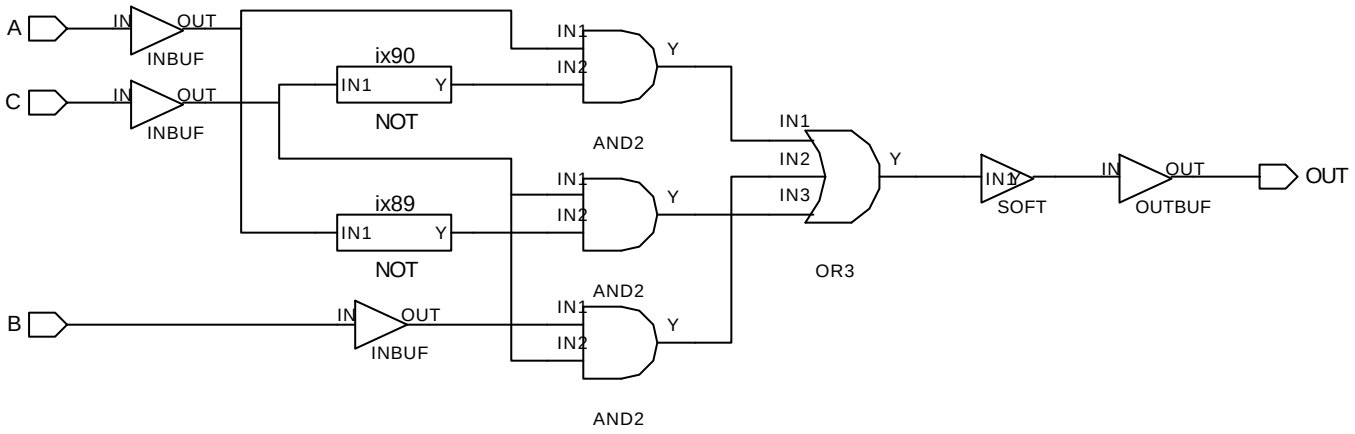

IN
IN
OUT
INBUF
IN
OUT
OUTBUF
OUT

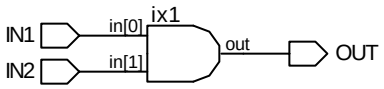

IN
IN
OUT
INBUF
IN
OUT
OUTBUF
OUT

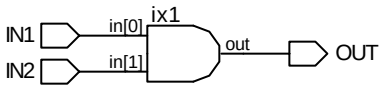

IN_1
IN_2
IN_3
IN
OUT
INBUF
OUT_NOT
IN1
Y
NOT
ix84
AND2
OR2
OR3
IN2
IN3
SOFT
OUTBUF
OUT_NOR
OUT_NOT
OUT_NAND
OUT_OR
OUT_BUF
OUT_AND

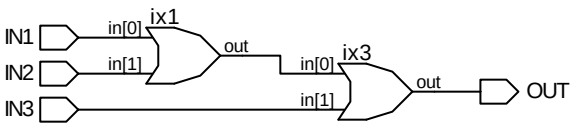

A
IN
OUT
INBUF
ix78
IN1
Y
NOT
B
IN
OUT
INBUF
ix77
IN1
Y
NOT
IN1
IN2
Y
OR2
IN1
Y
SOFT
IN
OUT
OUTBUF
OUT_RHS
IN
OUT
OUTBUF
OUT_LHS

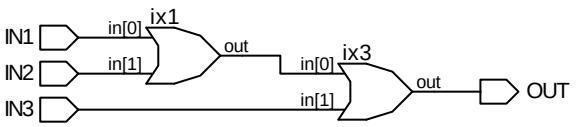

A
B
IN
OUT
INBUF
ix78
ix77
IN1
Y
NOT
IN1
IN2
Y
AND2
IN1
Y
SOFT
IN
OUT
OUTBUF
OUT_RHS
OUT_LHS

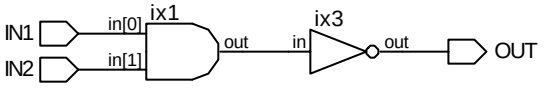

A
IN
OUT
INBUF
C
IN
OUT
INBUF
B
IN
OUT
INBUF
ix90
IN1
Y
NOT
ix89
IN1
Y
NOT
IN1
IN2
Y
AND2
IN1
IN2
Y
AND2
IN1
IN2
Y
AND2
IN1
IN2
IN3
Y
OR3
IN1
Y
SOFT
IN
OUT
OUTBUF
OUT

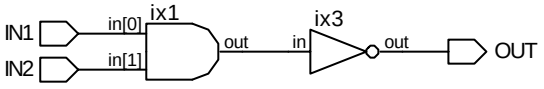

IN1
IN2
in[0]
in[1]
ix1
out
OUT

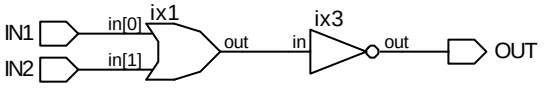

IN1
IN2
in[0]
in[1]
ix1
out
OUT

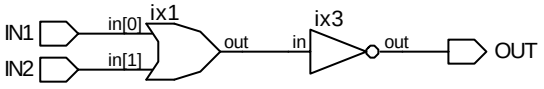

IN1
IN2
IN3
in[0]
in[1]
ix1
out
in[0]
in[1]
ix3
out
OUT

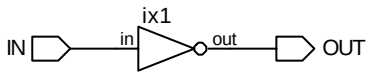

IN1
IN2
IN3
in[0]
in[1]
ix1
out
in[0]
in[1]
ix3
out
OUT

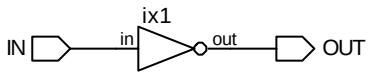

IN1
IN2
in[0]
in[1]
ix1
out
in
ix3
out
OUT

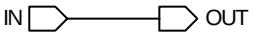

IN1
IN2
in[0]
in[1]
ix1
out
in
ix3
out
OUT

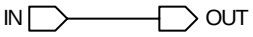

IN1
IN2
in[0]
in[1]
ix1
out
ix3
in
out
OUT

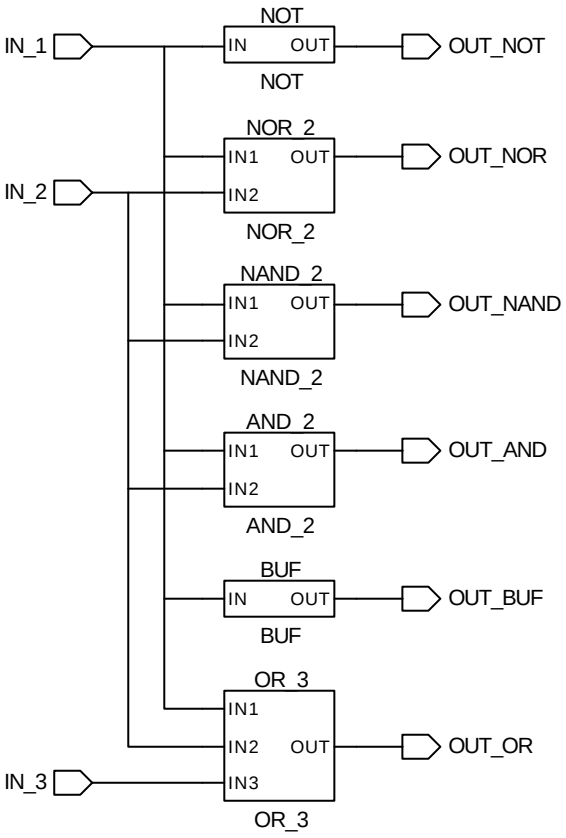

IN1
IN2
in[0]
in[1]
ix1
out
ix3
in
out
OUT

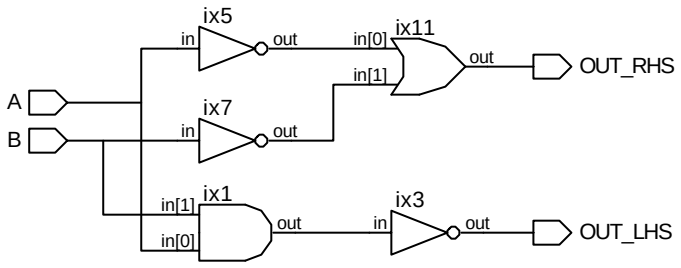

ix1
IN
in
out
OUT

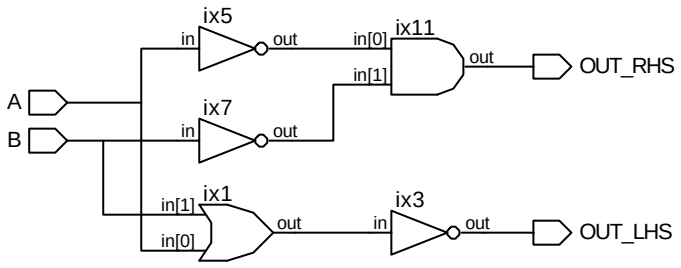

ix1
IN
in
out
OUT

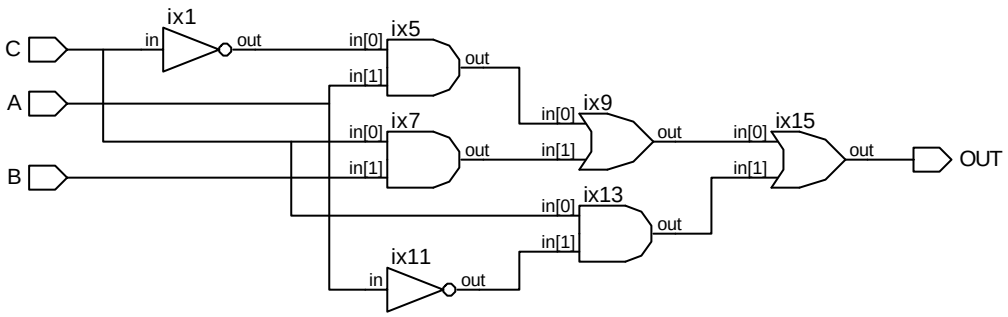

IN
OUT


IN
OUT


NOT
IN_1
IN
OUT
OUT_NOT
NOT
NOR_2
IN1
OUT
OUT_NOR
IN_2
IN2
NOR_2
NAND_2
IN1
OUT
OUT_NAND
IN2
NAND_2
AND_2
IN1
OUT
OUT_AND
IN2
AND_2
BUF
IN
OUT
OUT_BUF
BUF
OR_3
IN1
IN2
OUT
OUT_OR
IN_3
IN3
OR_3


ix5
in
out
ix11
in[0]
in[1]
out
OUT_RHS
A
B
ix7
in
out
ix1
in[1]
in[0]
out
ix3
in
out
OUT_LHS


ix5
in
out
ix11
in[0]
in[1]
out
OUT_RHS
A
B
ix7
in
out
ix1
in[1]
in[0]
out
ix3
in
out
OUT_LHS


C
A
B
ix1
in
out
ix5
in[0]
in[1]
out
ix7
in[0]
in[1]
out
ix9
in[0]
in[1]
out
ix11
in
out
ix13
in[0]
in[1]
out
ix15
in[0]
in[1]
out
OUT