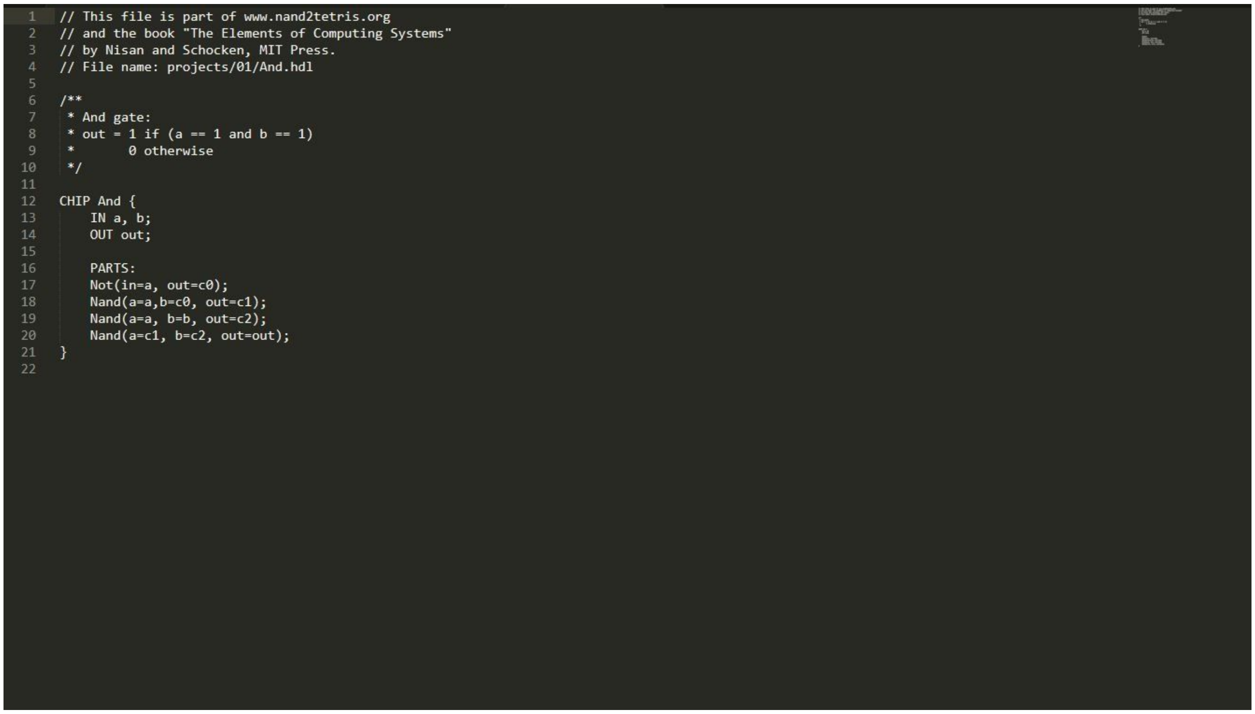
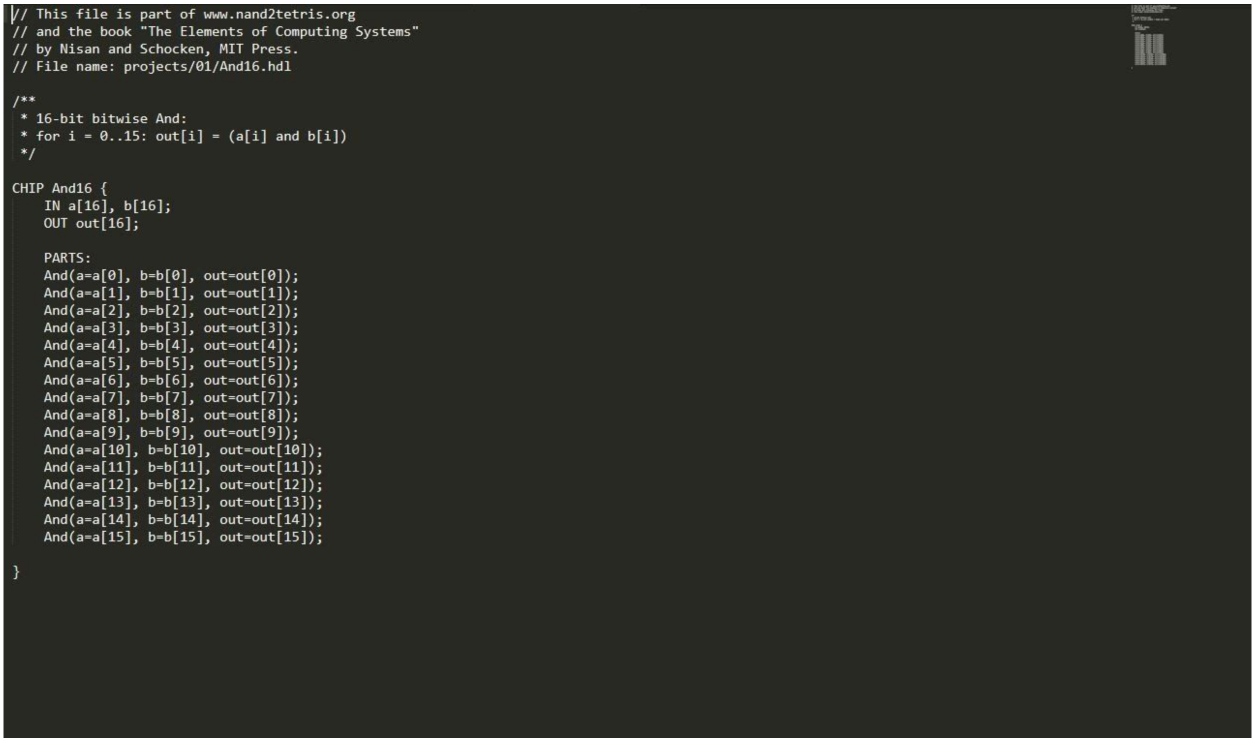
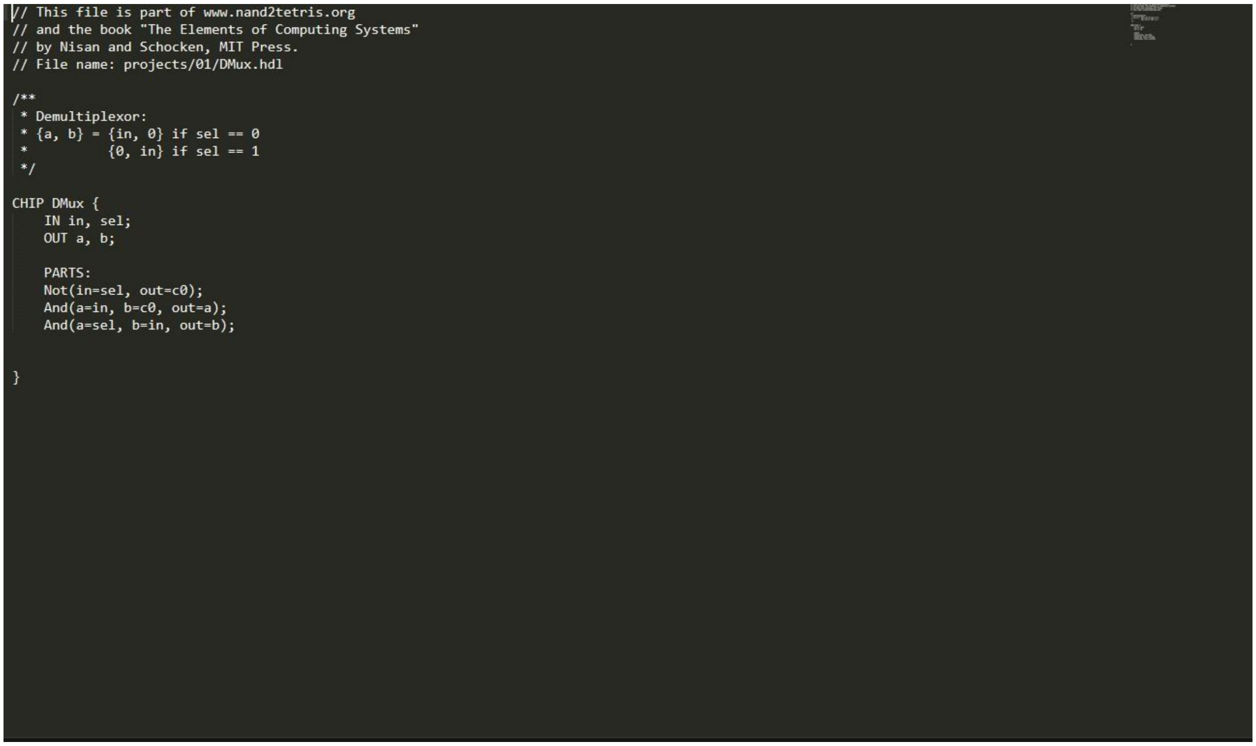
Hdl file screenshots



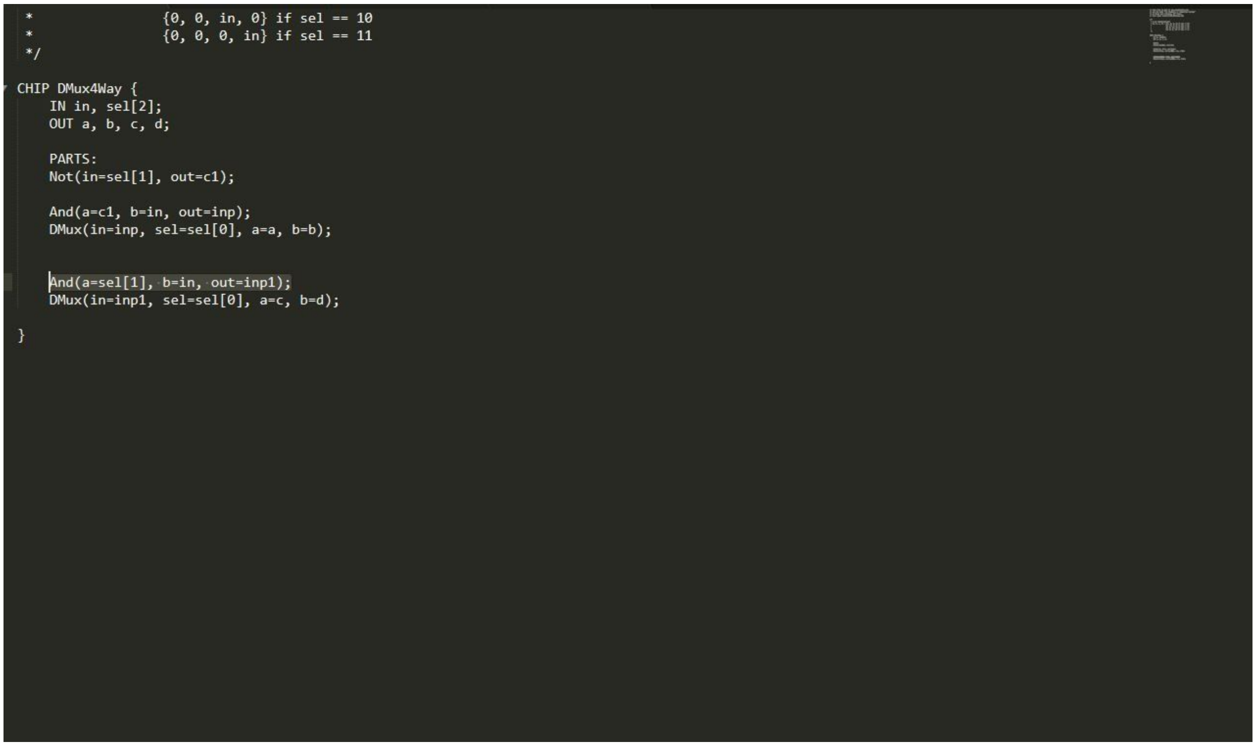
```
// This file is part of www.nand2tetris.org
// and the book "The Elements of Computing Systems"
// by Nisan and Schocken, MIT Press.
// File name: projects/01/And.hdl

/**
 * And gate:
 * out = 1 if (a == 1 and b == 1)
 *       0 otherwise
 */

CHIP And {
    IN a, b;
    OUT out;

    PARTS:
    Not(in=a, out=c0);
    Nand(a=a,b=c0, out=c1);
    Nand(a=a, b=b, out=c2);
    Nand(a=c1, b=c2, out=out);
}
```



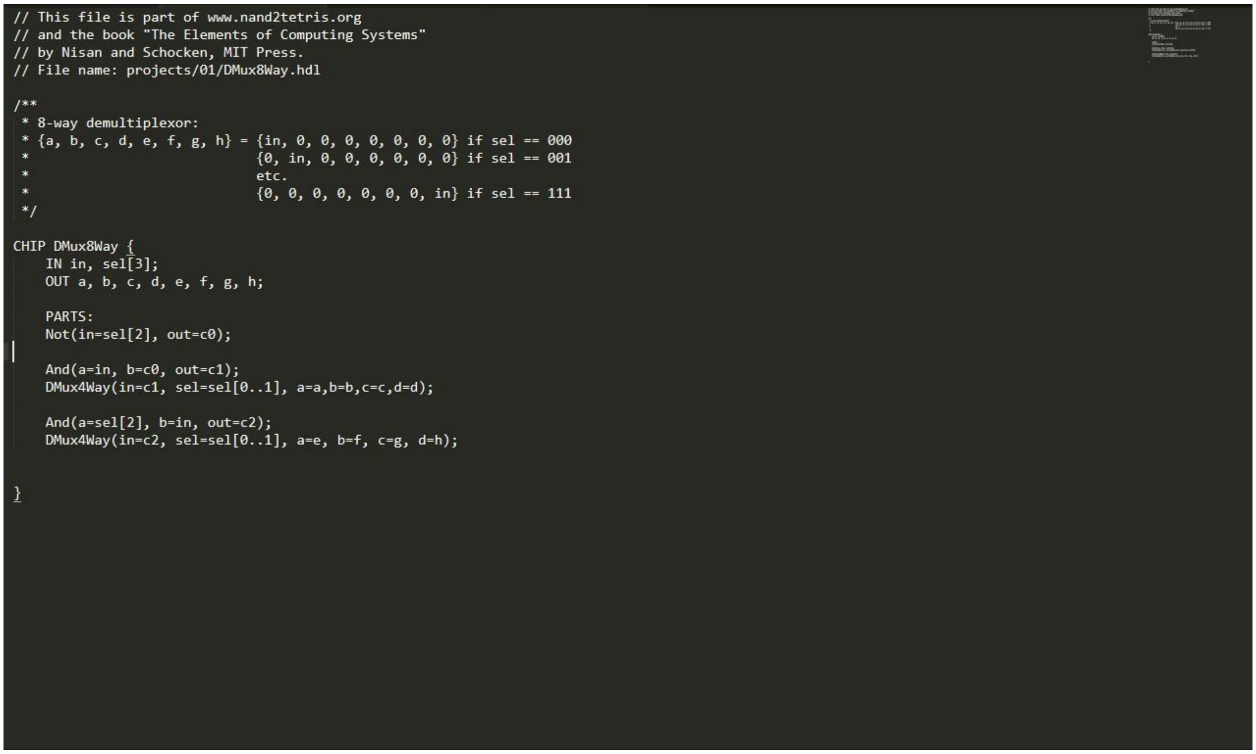
```
// This file is part of www.nand2tetris.org
// and the book "The Elements of Computing Systems"
// by Nisan and Schocken, MIT Press.
// File name: projects/01/And16.hdl

/**
 * 16-bit bitwise And:
 * for i = 0..15: out[i] = (a[i] and b[i])
 */

CHIP And16 {
    IN a[16], b[16];
    OUT out[16];

    PARTS:
    And(a=a[0], b=b[0], out=out[0]);
    And(a=a[1], b=b[1], out=out[1]);
    And(a=a[2], b=b[2], out=out[2]);
    And(a=a[3], b=b[3], out=out[3]);
    And(a=a[4], b=b[4], out=out[4]);
    And(a=a[5], b=b[5], out=out[5]);
    And(a=a[6], b=b[6], out=out[6]);
    And(a=a[7], b=b[7], out=out[7]);
    And(a=a[8], b=b[8], out=out[8]);
    And(a=a[9], b=b[9], out=out[9]);
    And(a=a[10], b=b[10], out=out[10]);
    And(a=a[11], b=b[11], out=out[11]);
    And(a=a[12], b=b[12], out=out[12]);
    And(a=a[13], b=b[13], out=out[13]);
    And(a=a[14], b=b[14], out=out[14]);
    And(a=a[15], b=b[15], out=out[15]);

}
```



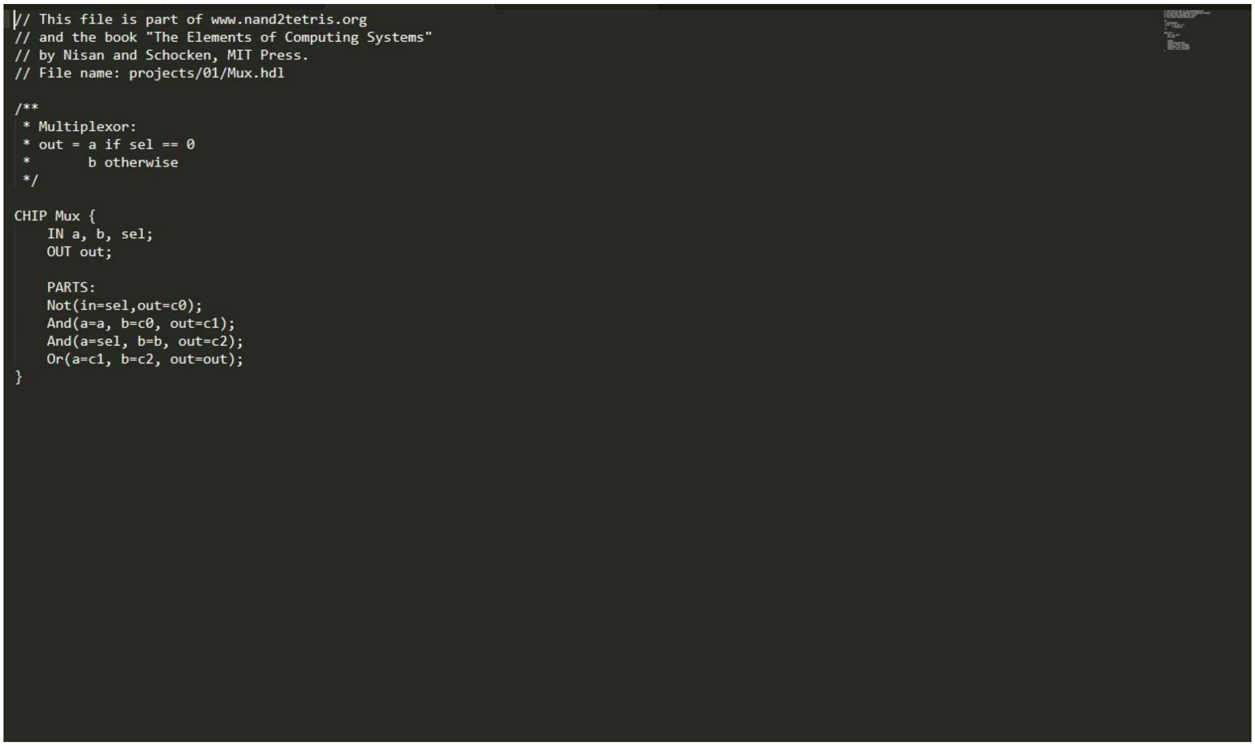
```
// This file is part of www.nand2tetris.org
// and the book "The Elements of Computing Systems"
// by Nisan and Schocken, MIT Press.
// File name: projects/01/DMux.hdl

/**
 * Demultiplexor:
 * {a, b} = {in, 0} if sel == 0
 *          {0, in} if sel == 1
 */

CHIP DMux {
    IN in, sel;
    OUT a, b;

    PARTS:
    Not(in=sel, out=c0);
    And(a=in, b=c0, out=a);
    And(a=sel, b=in, out=b);


}
```



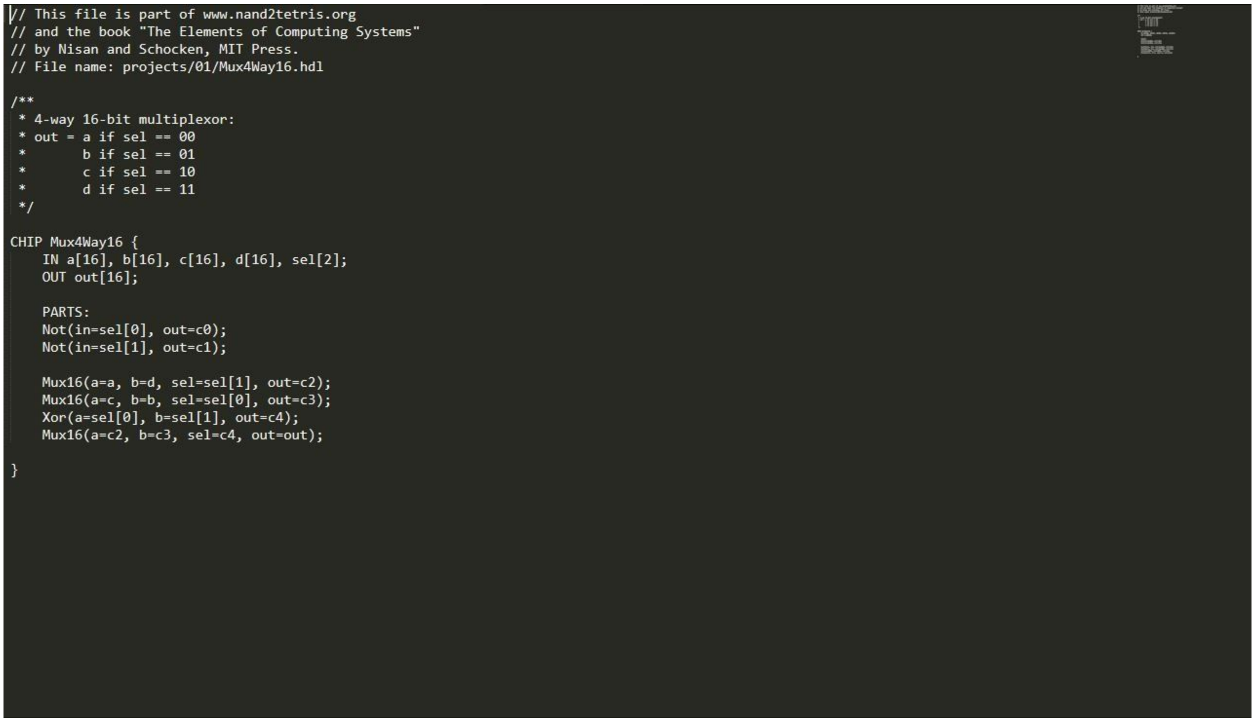
```
 *                {0, 0, in, 0} if sel == 10
 *                {0, 0, 0, in} if sel == 11
 */

CHIP DMux4Way {
    IN in, sel[2];
    OUT a, b, c, d;

    PARTS:
    Not(in=sel[1], out=c1);

    And(a=c1, b=in, out=inp);
    DMux(in=inp, sel=sel[0], a=a, b=b);


    And(a=sel[1], b=in, out=inp1);
    DMux(in=inp1, sel=sel[0], a=c, b=d);

}
```



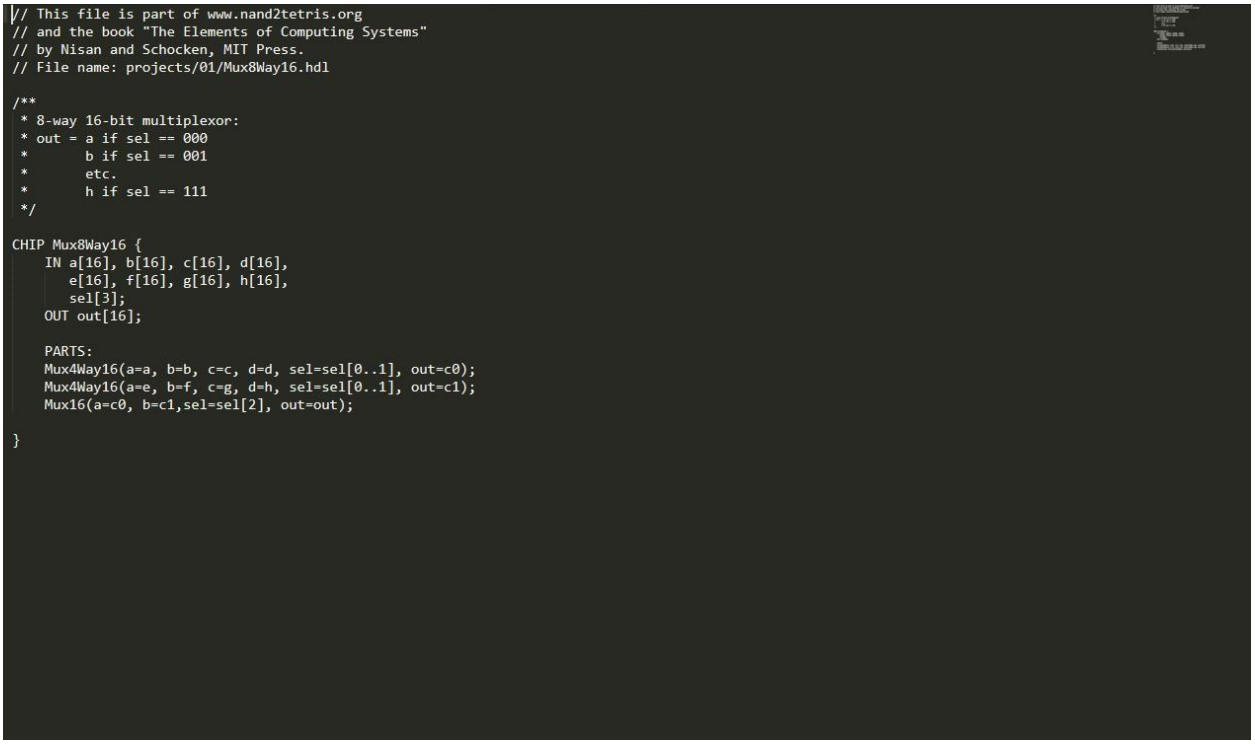
```
// This file is part of www.nand2tetris.org
// and the book "The Elements of Computing Systems"
// by Nisan and Schocken, MIT Press.
// File name: projects/01/DMux8Way.hdl

/**
 * 8-way demultiplexor:
 * {a, b, c, d, e, f, g, h} = {in, 0, 0, 0, 0, 0, 0, 0} if sel == 000
 *                            {0, in, 0, 0, 0, 0, 0, 0} if sel == 001
 *                            etc.
 *                            {0, 0, 0, 0, 0, 0, 0, in} if sel == 111
 */

CHIP DMux8Way {
    IN in, sel[3];
    OUT a, b, c, d, e, f, g, h;

    PARTS:
    Not(in=sel[2], out=c0);

    And(a=in, b=c0, out=c1);
    DMux4Way(in=c1, sel=sel[0..1], a=a,b=b,c=c,d=d);

    And(a=sel[2], b=in, out=c2);
    DMux4Way(in=c2, sel=sel[0..1], a=e, b=f, c=g, d=h);


}
```



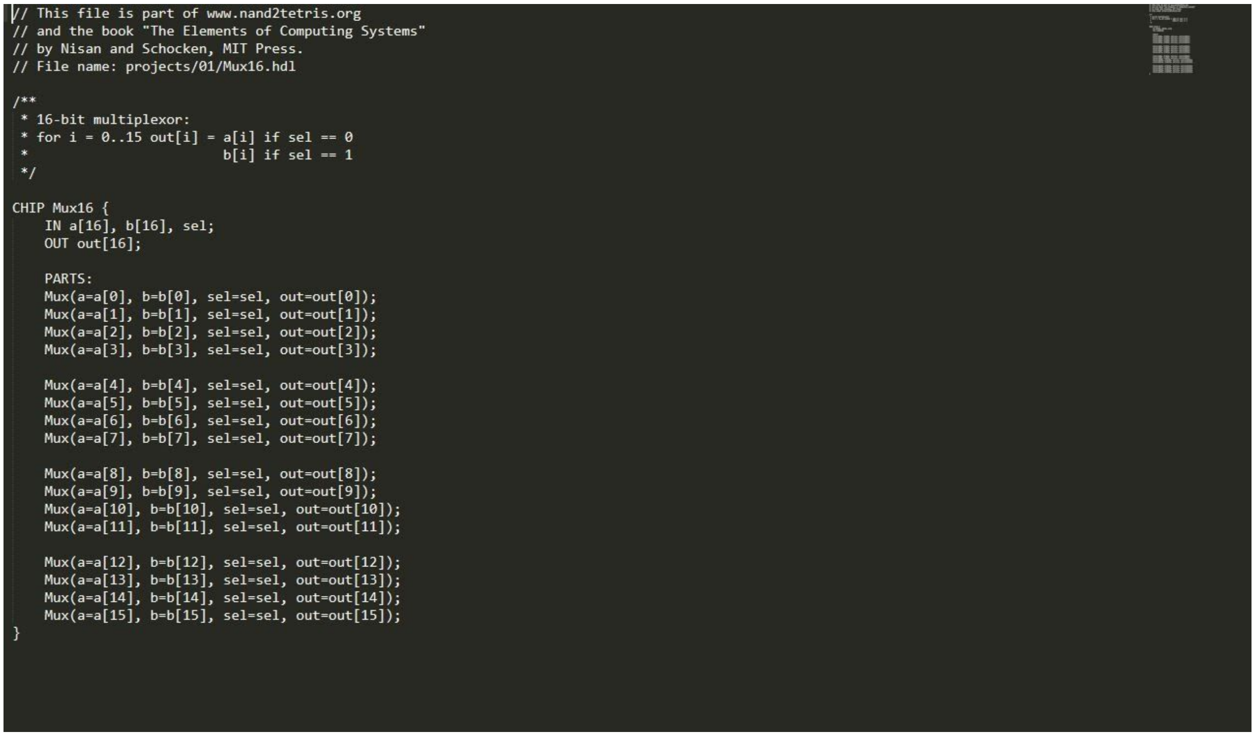
```
// This file is part of www.nand2tetris.org
// and the book "The Elements of Computing Systems"
// by Nisan and Schocken, MIT Press.
// File name: projects/01/Mux.hdl

/**
 * Multiplexor:
 * out = a if sel == 0
 *       b otherwise
 */

CHIP Mux {
    IN a, b, sel;
    OUT out;

    PARTS:
    Not(in=sel,out=c0);
    And(a=a, b=c0, out=c1);
    And(a=sel, b=b, out=c2);
    Or(a=c1, b=c2, out=out);
}
```



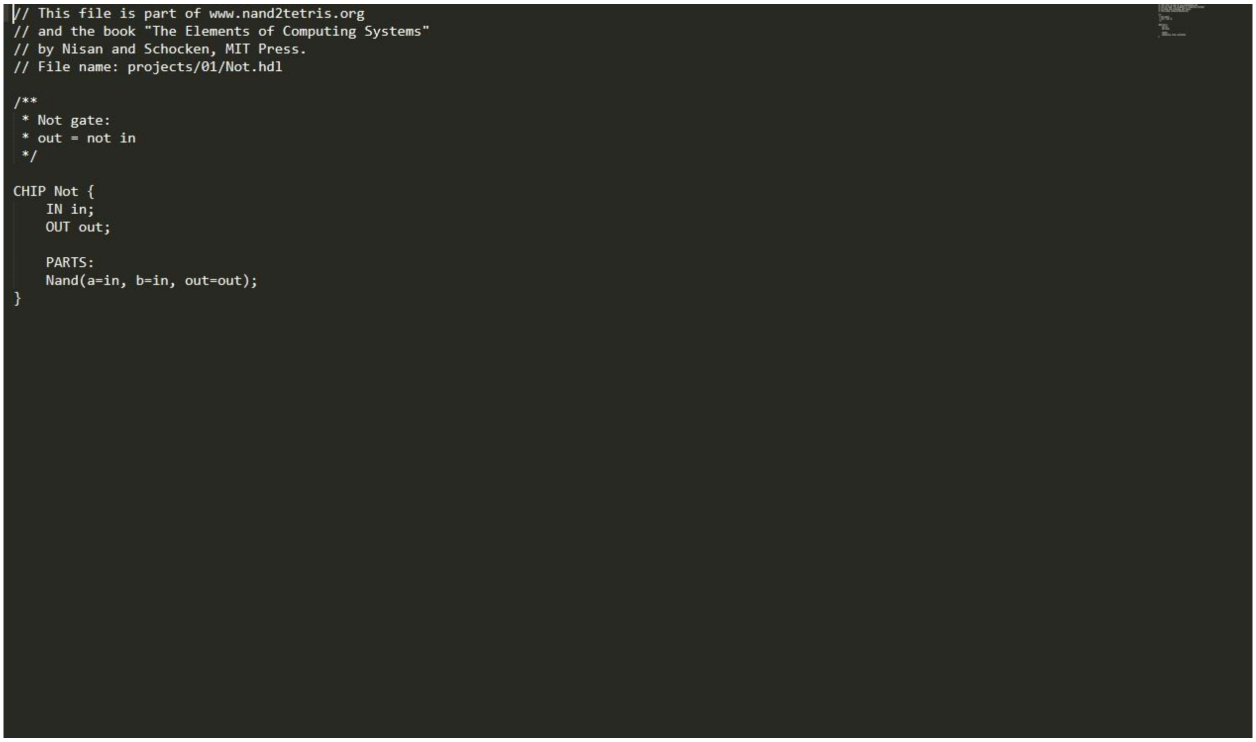
```
// This file is part of www.nand2tetris.org
// and the book "The Elements of Computing Systems"
// by Nisan and Schocken, MIT Press.
// File name: projects/01/Mux4Way16.hdl

/**
 * 4-way 16-bit multiplexor:
 * out = a if sel == 00
 *       b if sel == 01
 *       c if sel == 10
 *       d if sel == 11
 */

CHIP Mux4Way16 {
    IN a[16], b[16], c[16], d[16], sel[2];
    OUT out[16];

    PARTS:
    Not(in=sel[0], out=c0);
    Not(in=sel[1], out=c1);

    Mux16(a=a, b=d, sel=sel[1], out=c2);
    Mux16(a=c, b=b, sel=sel[0], out=c3);
    Xor(a=sel[0], b=sel[1], out=c4);
    Mux16(a=c2, b=c3, sel=c4, out=out);

}
```



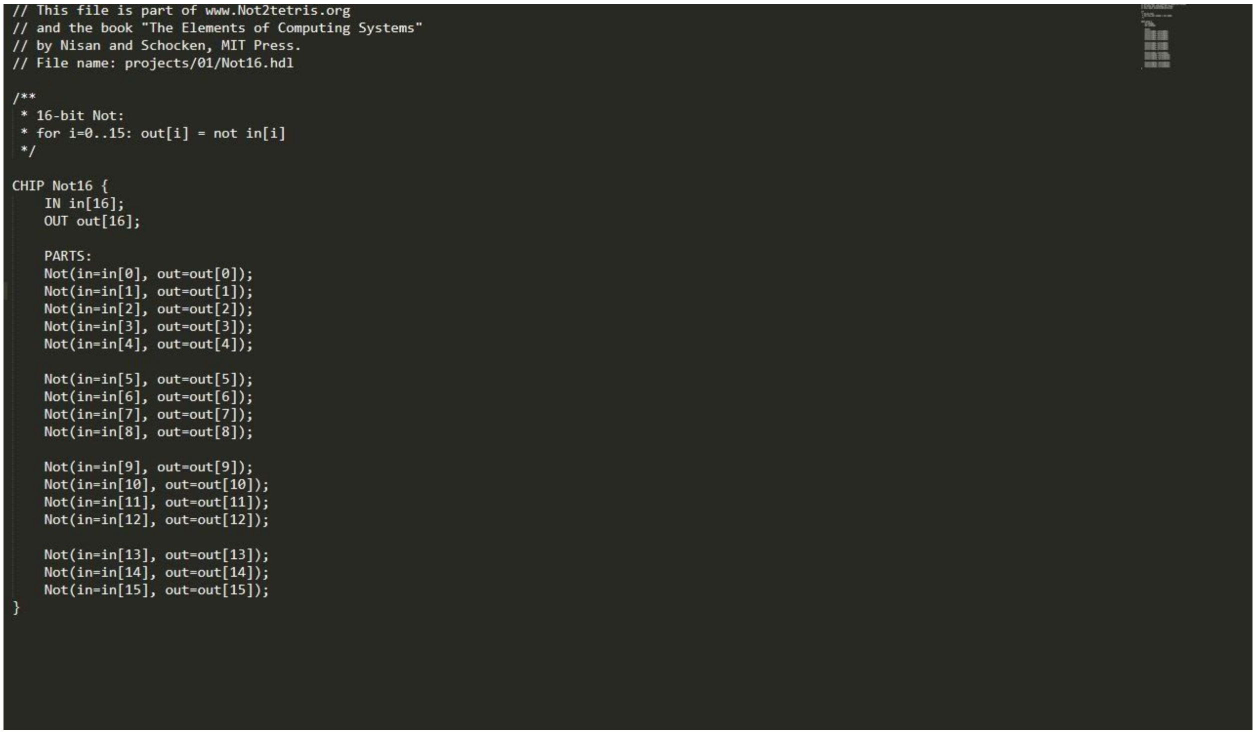
```
// This file is part of www.nand2tetris.org
// and the book "The Elements of Computing Systems"
// by Nisan and Schocken, MIT Press.
// File name: projects/01/Mux8Way16.hdl

/**
 * 8-way 16-bit multiplexor:
 * out = a if sel == 000
 *       b if sel == 001
 *       etc.
 *       h if sel == 111
 */

CHIP Mux8Way16 {
    IN a[16], b[16], c[16], d[16],
       e[16], f[16], g[16], h[16],
       sel[3];
    OUT out[16];

    PARTS:
    Mux4Way16(a=a, b=b, c=c, d=d, sel=sel[0..1], out=c0);
    Mux4Way16(a=e, b=f, c=g, d=h, sel=sel[0..1], out=c1);
    Mux16(a=c0, b=c1,sel=sel[2], out=out);

}
```



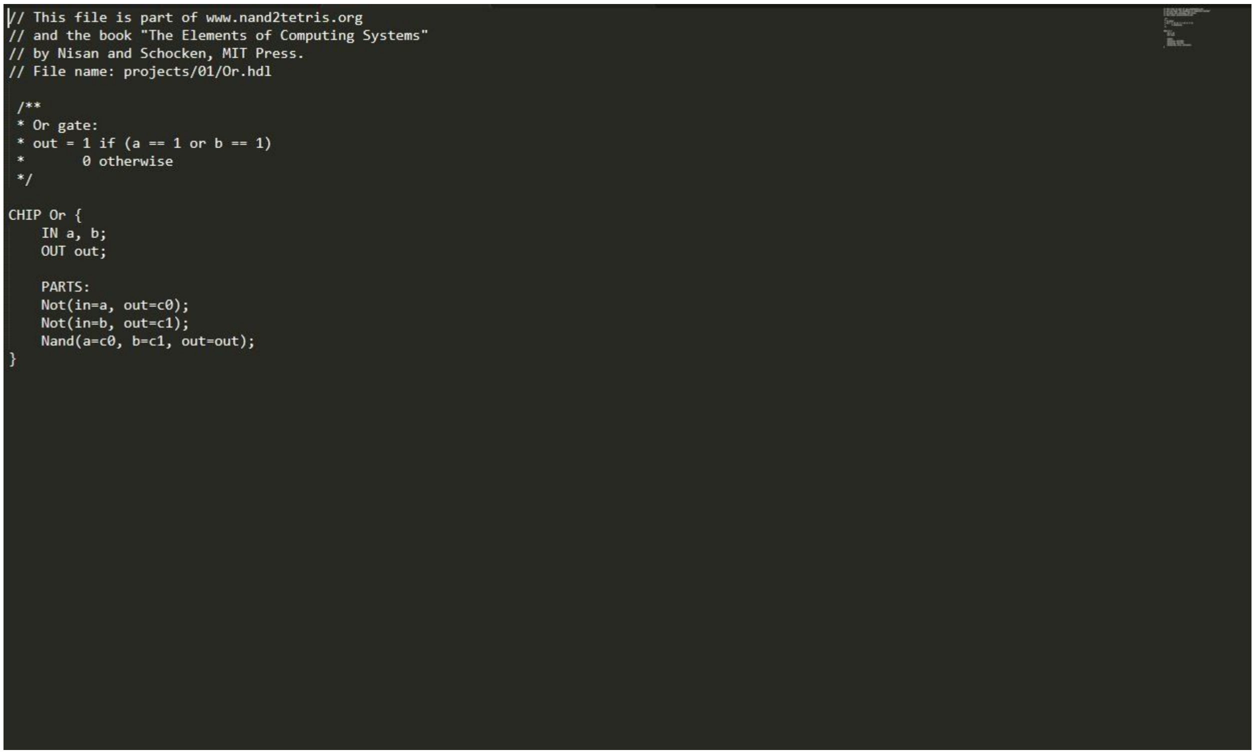
```
// This file is part of www.nand2tetris.org
// and the book "The Elements of Computing Systems"
// by Nisan and Schocken, MIT Press.
// File name: projects/01/Mux16.hdl

/**
 * 16-bit multiplexor:
 * for i = 0..15 out[i] = a[i] if sel == 0
 *                        b[i] if sel == 1
 */

CHIP Mux16 {
    IN a[16], b[16], sel;
    OUT out[16];

    PARTS:
    Mux(a=a[0], b=b[0], sel=sel, out=out[0]);
    Mux(a=a[1], b=b[1], sel=sel, out=out[1]);
    Mux(a=a[2], b=b[2], sel=sel, out=out[2]);
    Mux(a=a[3], b=b[3], sel=sel, out=out[3]);

    Mux(a=a[4], b=b[4], sel=sel, out=out[4]);
    Mux(a=a[5], b=b[5], sel=sel, out=out[5]);
    Mux(a=a[6], b=b[6], sel=sel, out=out[6]);
    Mux(a=a[7], b=b[7], sel=sel, out=out[7]);

    Mux(a=a[8], b=b[8], sel=sel, out=out[8]);
    Mux(a=a[9], b=b[9], sel=sel, out=out[9]);
    Mux(a=a[10], b=b[10], sel=sel, out=out[10]);
    Mux(a=a[11], b=b[11], sel=sel, out=out[11]);

    Mux(a=a[12], b=b[12], sel=sel, out=out[12]);
    Mux(a=a[13], b=b[13], sel=sel, out=out[13]);
    Mux(a=a[14], b=b[14], sel=sel, out=out[14]);
    Mux(a=a[15], b=b[15], sel=sel, out=out[15]);
}
```



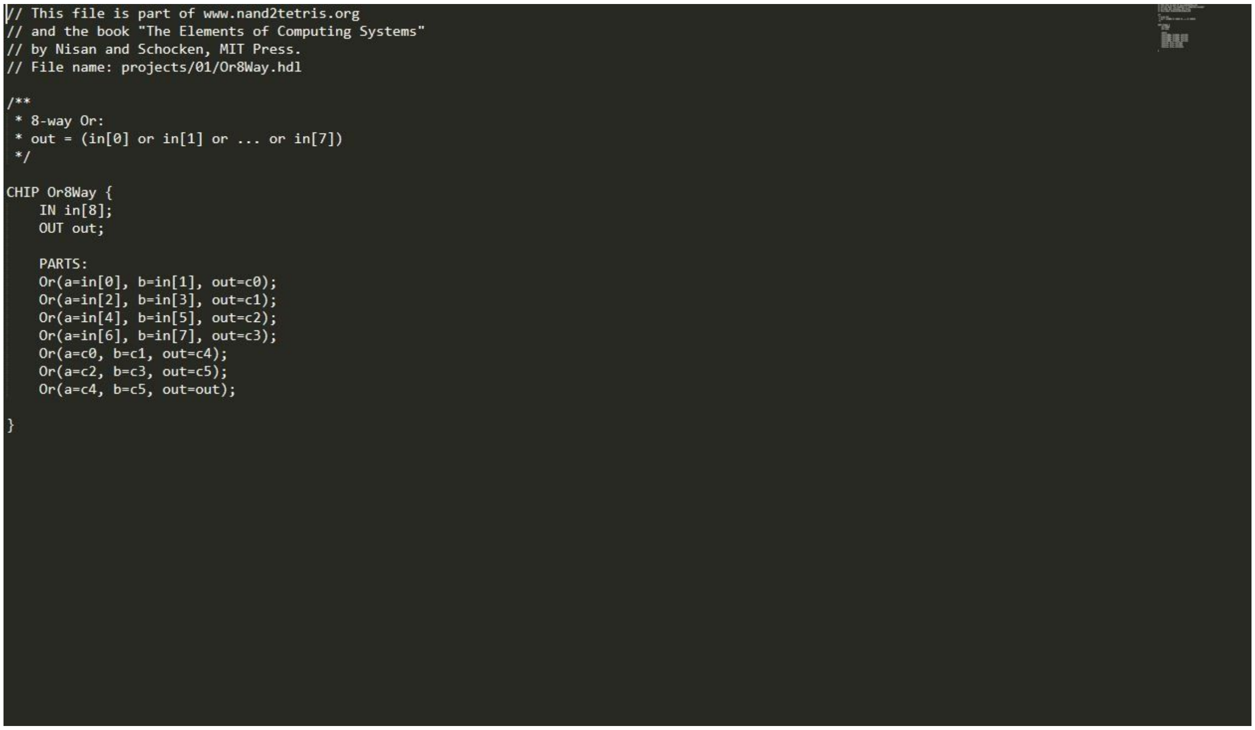
```
// This file is part of www.nand2tetris.org
// and the book "The Elements of Computing Systems"
// by Nisan and Schocken, MIT Press.
// File name: projects/01/Not.hdl

/**
 * Not gate:
 * out = not in
 */

CHIP Not {
    IN in;
    OUT out;

    PARTS:
    Nand(a=in, b=in, out=out);
}
```



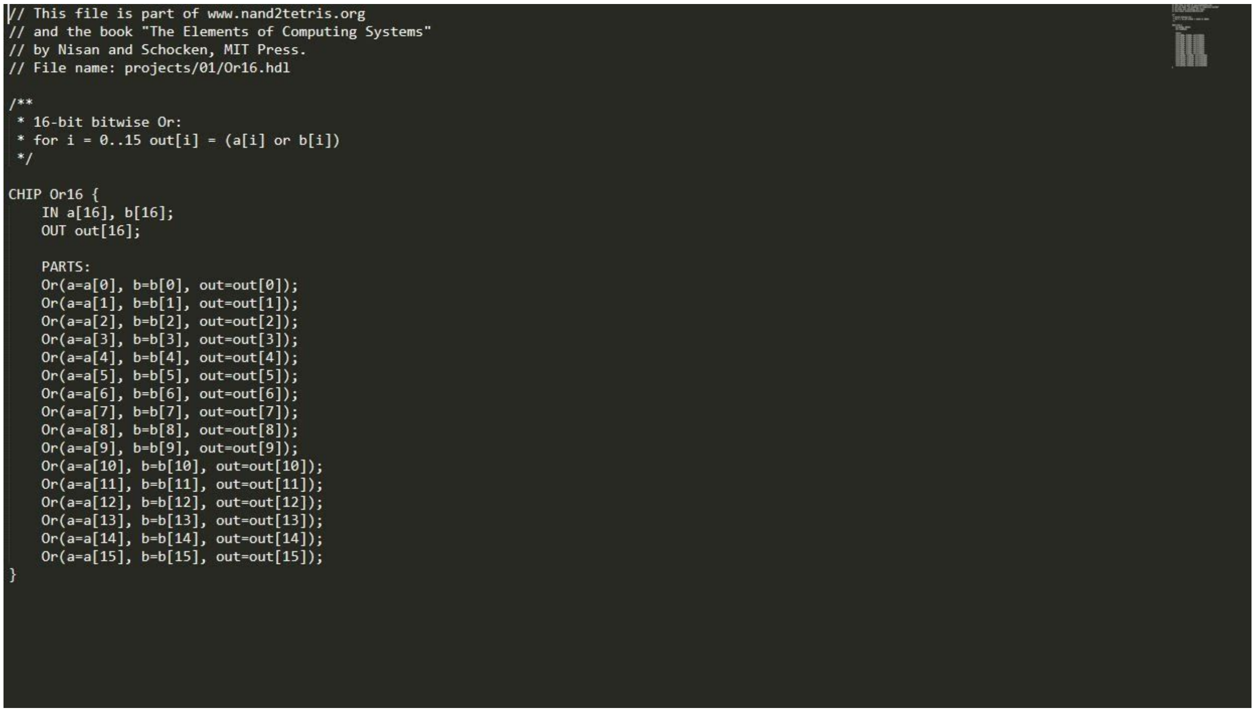
```
// This file is part of www.Not2tetris.org
// and the book "The Elements of Computing Systems"
// by Nisan and Schocken, MIT Press.
// File name: projects/01/Not16.hdl

/**
 * 16-bit Not:
 * for i=0..15: out[i] = not in[i]
 */

CHIP Not16 {
    IN in[16];
    OUT out[16];

    PARTS:
    Not(in=in[0], out=out[0]);
    Not(in=in[1], out=out[1]);
    Not(in=in[2], out=out[2]);
    Not(in=in[3], out=out[3]);
    Not(in=in[4], out=out[4]);

    Not(in=in[5], out=out[5]);
    Not(in=in[6], out=out[6]);
    Not(in=in[7], out=out[7]);
    Not(in=in[8], out=out[8]);

    Not(in=in[9], out=out[9]);
    Not(in=in[10], out=out[10]);
    Not(in=in[11], out=out[11]);
    Not(in=in[12], out=out[12]);

    Not(in=in[13], out=out[13]);
    Not(in=in[14], out=out[14]);
    Not(in=in[15], out=out[15]);
}
```



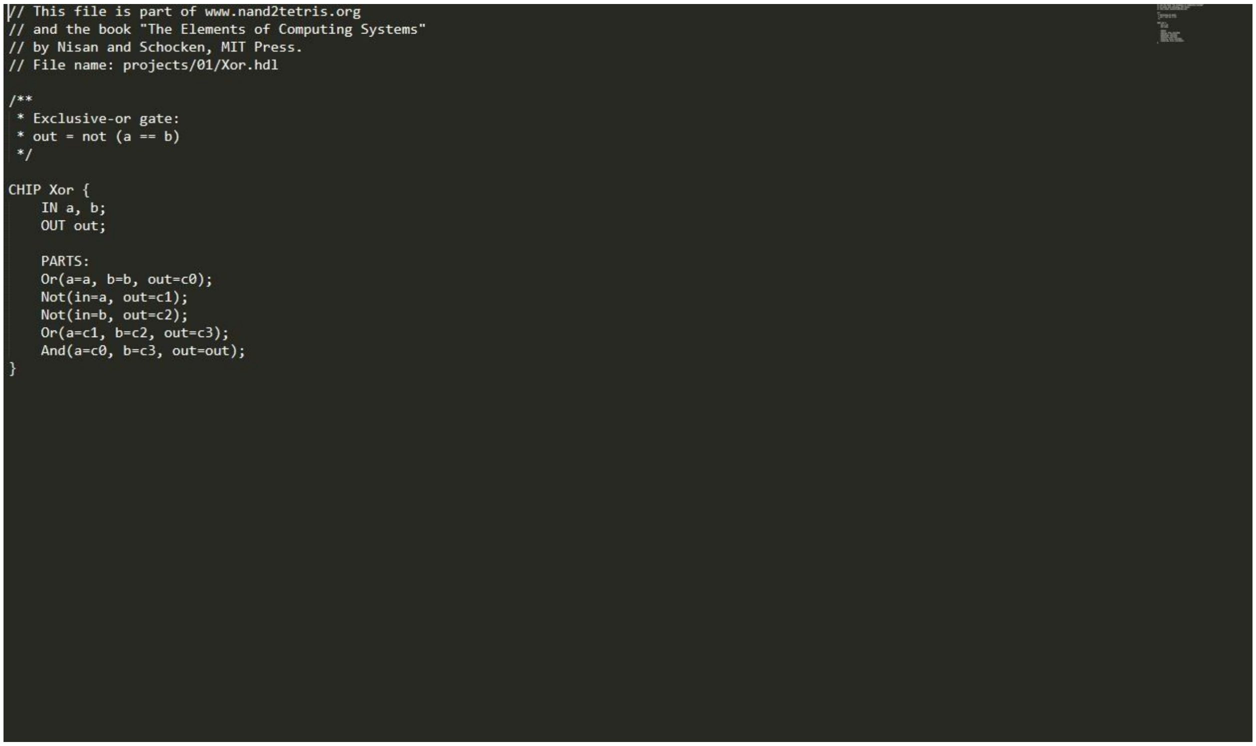
```
// This file is part of www.nand2tetris.org
// and the book "The Elements of Computing Systems"
// by Nisan and Schocken, MIT Press.
// File name: projects/01/Or.hdl

 /**
 * Or gate:
 * out = 1 if (a == 1 or b == 1)
 *       0 otherwise
 */

CHIP Or {
    IN a, b;
    OUT out;

    PARTS:
    Not(in=a, out=c0);
    Not(in=b, out=c1);
    Nand(a=c0, b=c1, out=out);
}
```



```
// This file is part of www.nand2tetris.org
// and the book "The Elements of Computing Systems"
// by Nisan and Schocken, MIT Press.
// File name: projects/01/Or8Way.hdl

/**
 * 8-way Or:
 * out = (in[0] or in[1] or ... or in[7])
 */

CHIP Or8Way {
    IN in[8];
    OUT out;

    PARTS:
    Or(a=in[0], b=in[1], out=c0);
    Or(a=in[2], b=in[3], out=c1);
    Or(a=in[4], b=in[5], out=c2);
    Or(a=in[6], b=in[7], out=c3);
    Or(a=c0, b=c1, out=c4);
    Or(a=c2, b=c3, out=c5);
    Or(a=c4, b=c5, out=out);

}
```

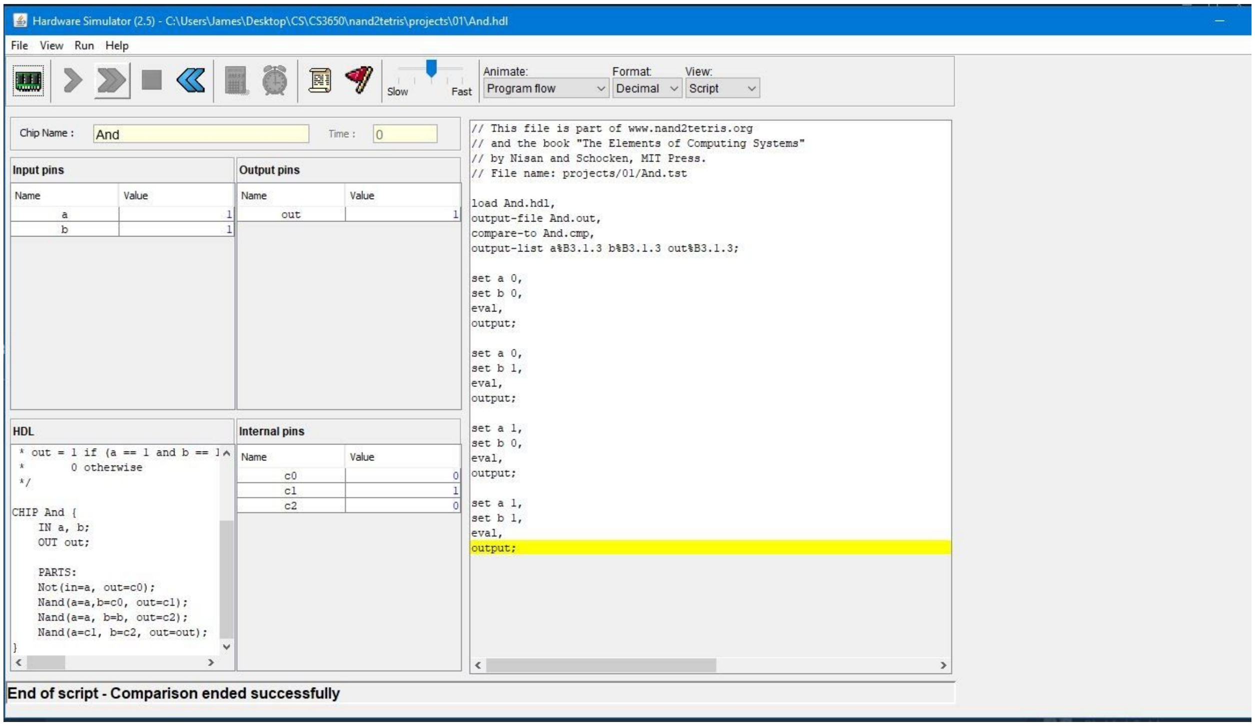


```
// This file is part of www.nand2tetris.org
// and the book "The Elements of Computing Systems"
// by Nisan and Schocken, MIT Press.
// File name: projects/01/Or16.hdl

/**
 * 16-bit bitwise Or:
 * for i = 0..15 out[i] = (a[i] or b[i])
 */

CHIP Or16 {
    IN a[16], b[16];
    OUT out[16];

    PARTS:
    Or(a=a[0], b=b[0], out=out[0]);
    Or(a=a[1], b=b[1], out=out[1]);
    Or(a=a[2], b=b[2], out=out[2]);
    Or(a=a[3], b=b[3], out=out[3]);
    Or(a=a[4], b=b[4], out=out[4]);
    Or(a=a[5], b=b[5], out=out[5]);
    Or(a=a[6], b=b[6], out=out[6]);
    Or(a=a[7], b=b[7], out=out[7]);
    Or(a=a[8], b=b[8], out=out[8]);
    Or(a=a[9], b=b[9], out=out[9]);
    Or(a=a[10], b=b[10], out=out[10]);
    Or(a=a[11], b=b[11], out=out[11]);
    Or(a=a[12], b=b[12], out=out[12]);
    Or(a=a[13], b=b[13], out=out[13]);
    Or(a=a[14], b=b[14], out=out[14]);
    Or(a=a[15], b=b[15], out=out[15]);
}
```



```
// This file is part of www.nand2tetris.org
// and the book "The Elements of Computing Systems"
// by Nisan and Schocken, MIT Press.
// File name: projects/01/Xor.hdl

/**
 * Exclusive-or gate:
 * out = not (a == b)
 */

CHIP Xor {
    IN a, b;
    OUT out;

    PARTS:
    Or(a=a, b=b, out=c0);
    Not(in=a, out=c1);
    Not(in=b, out=c2);
    Or(a=c1, b=c2, out=c3);
    And(a=c0, b=c3, out=out);
}
```

Tst screenshots


Hardware Simulator (2.5) - C:\Users\James\Desktop\CS\CS3650\nand2tetris\projects\01\And.hdl
File View Run Help
Slow
Fast

Animate: Program flow

Format: Decimal

View: Script

Chip Name : And

Time : 0

Input pins

Name	Value
a	1
b	1

Output pins

Name	Value
out	1

HDL

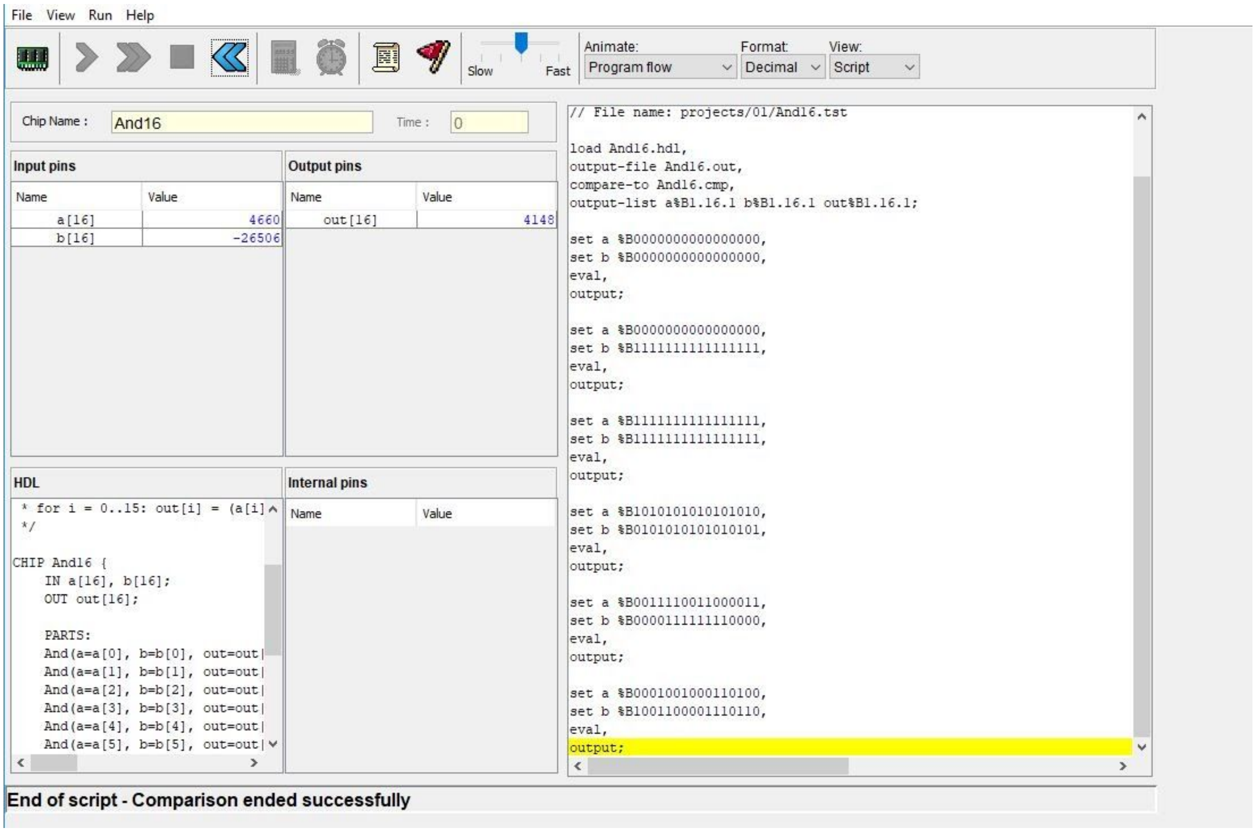
```
 * out = 1 if (a == 1 and b == 1
 *       0 otherwise
 */

CHIP And {
    IN a, b;
    OUT out;

    PARTS:
    Not(in=a, out=c0);
    Nand(a=a,b=c0, out=c1);
    Nand(a=a, b=b, out=c2);
    Nand(a=c1, b=c2, out=out);
}
```

Internal pins

Name	Value
c0	0
c1	1
c2	0

```
// This file is part of www.nand2tetris.org
// and the book "The Elements of Computing Systems"
// by Nisan and Schocken, MIT Press.
// File name: projects/01/And.tst

load And.hdl,
output-file And.out,
compare-to And.cmp,
output-list a%B3.1.3 b%B3.1.3 out%B3.1.3;

set a 0,
set b 0,
eval,
output;

set a 0,
set b 1,
eval,
output;

set a 1,
set b 0,
eval,
output;

set a 1,
set b 1,
eval,
output;
```

End of script - Comparison ended successfully



Animate: Program flow

Format: Decimal

View: Script

Chip Name : And16

Time : 0

Input pins

Name	Value
a[16]	4660
b[16]	-26506

Output pins

Name	Value
out[16]	4148

HDL

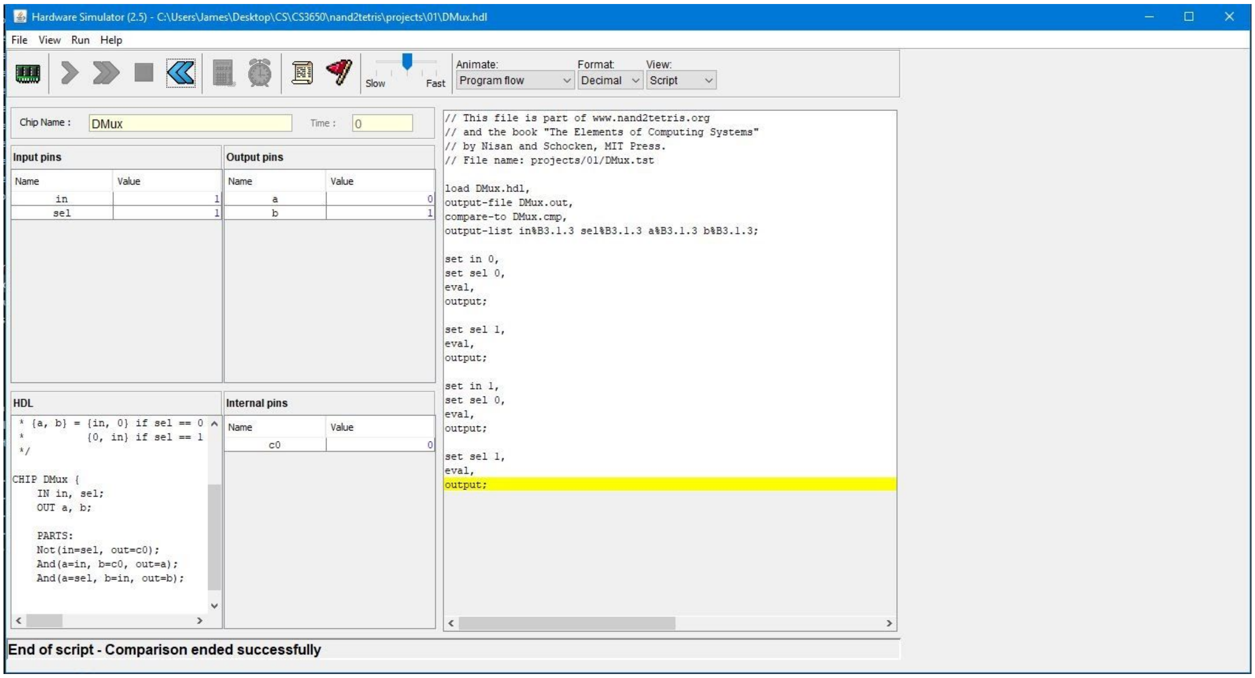
```
 * for i = 0..15: out[i] = (a[i]
 */

CHIP And16 {
    IN a[16], b[16];
    OUT out[16];

    PARTS:
    And(a=a[0], b=b[0], out=out
    And(a=a[1], b=b[1], out=out
    And(a=a[2], b=b[2], out=out
    And(a=a[3], b=b[3], out=out
    And(a=a[4], b=b[4], out=out
    And(a=a[5], b=b[5], out=out
```

Internal pins

Name	Value

```
// File name: projects/01/And16.tst

load And16.hdl,
output-file And16.out,
compare-to And16.cmp,
output-list a%B1.16.1 b%B1.16.1 out%B1.16.1;

set a %B0000000000000000,
set b %B0000000000000000,
eval,
output;

set a %B0000000000000000,
set b %B1111111111111111,
eval,
output;

set a %B1111111111111111,
set b %B1111111111111111,
eval,
output;

set a %B1010101010101010,
set b %B0101010101010101,
eval,
output;

set a %B0011110011000011,
set b %B0000111111110000,
eval,
output;

set a %B0001001000110100,
set b %B1001100001110110,
eval,
output;
```

End of script - Comparison ended successfully


Hardware Simulator (2.5) - C:\Users\James\Desktop\CS\CS3650\nand2tetris\projects\01\DMux.hdl
File
View
Run
Help
Slow
Fast

Animate: Program flow
Format: Decimal
View: Script

Chip Name : DMux
Time : 0

Input pins

Name	Value
in	1
sel	1

Output pins

Name	Value
a	0
b	1

HDL

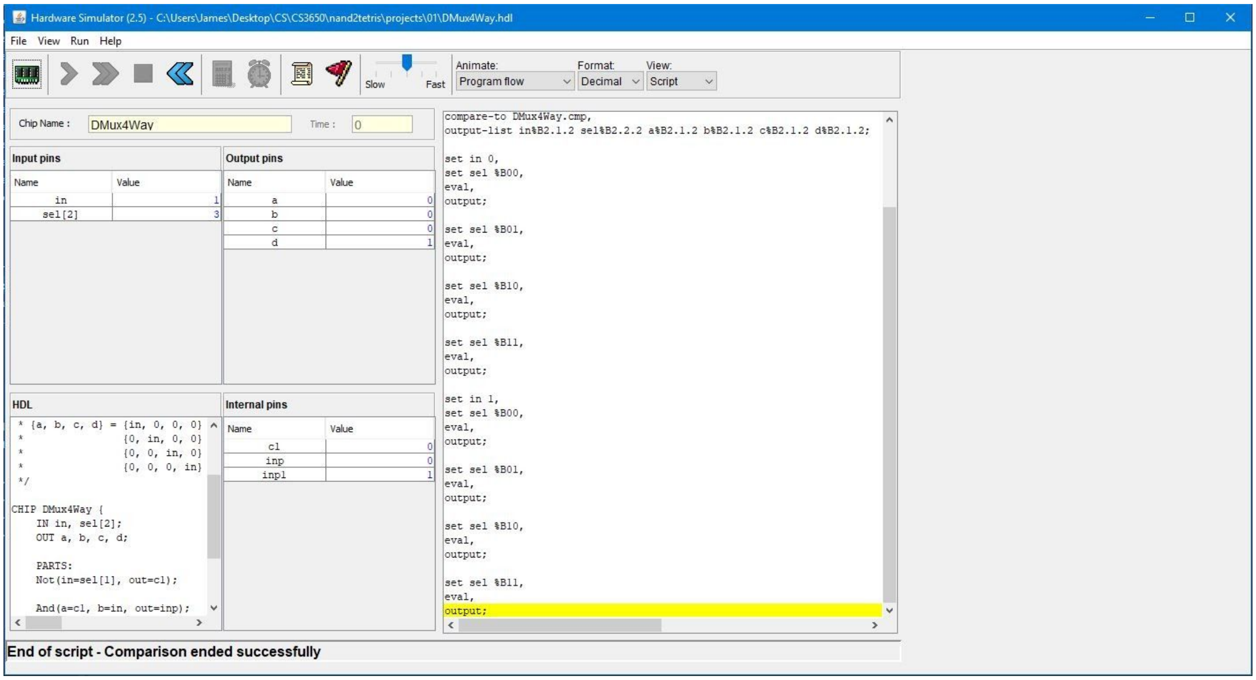
```
 * {a, b} = {in, 0} if sel == 0
 *          {0, in} if sel == 1
 */

CHIP DMux {
    IN in, sel;
    OUT a, b;

    PARTS:
    Not(in=sel, out=c0);
    And(a=in, b=c0, out=a);
    And(a=sel, b=in, out=b);
```

Internal pins

Name	Value
c0	0

```
// This file is part of www.nand2tetris.org
// and the book "The Elements of Computing Systems"
// by Nisan and Schocken, MIT Press.
// File name: projects/01/DMux.tst

load DMux.hdl,
output-file DMux.out,
compare-to DMux.cmp,
output-list in%B3.1.3 sel%B3.1.3 a%B3.1.3 b%B3.1.3;

set in 0,
set sel 0,
eval,
output;

set sel 1,
eval,
output;

set in 1,
set sel 0,
eval,
output;

set sel 1,
eval,
output;
```

End of script - Comparison ended successfully


Hardware Simulator (2.5) - C:\Users\James\Desktop\CS\CS3650\nand2tetris\projects\01\DMux4Way.hdl
File
View
Run
Help
Slow
Fast

Animate: Program flow
Format: Decimal
View: Script

Chip Name : DMux4Way
Time : 0

Input pins

Name	Value
in	1
sel[2]	3

Output pins

Name	Value
a	0
b	0
c	0
d	1

HDL

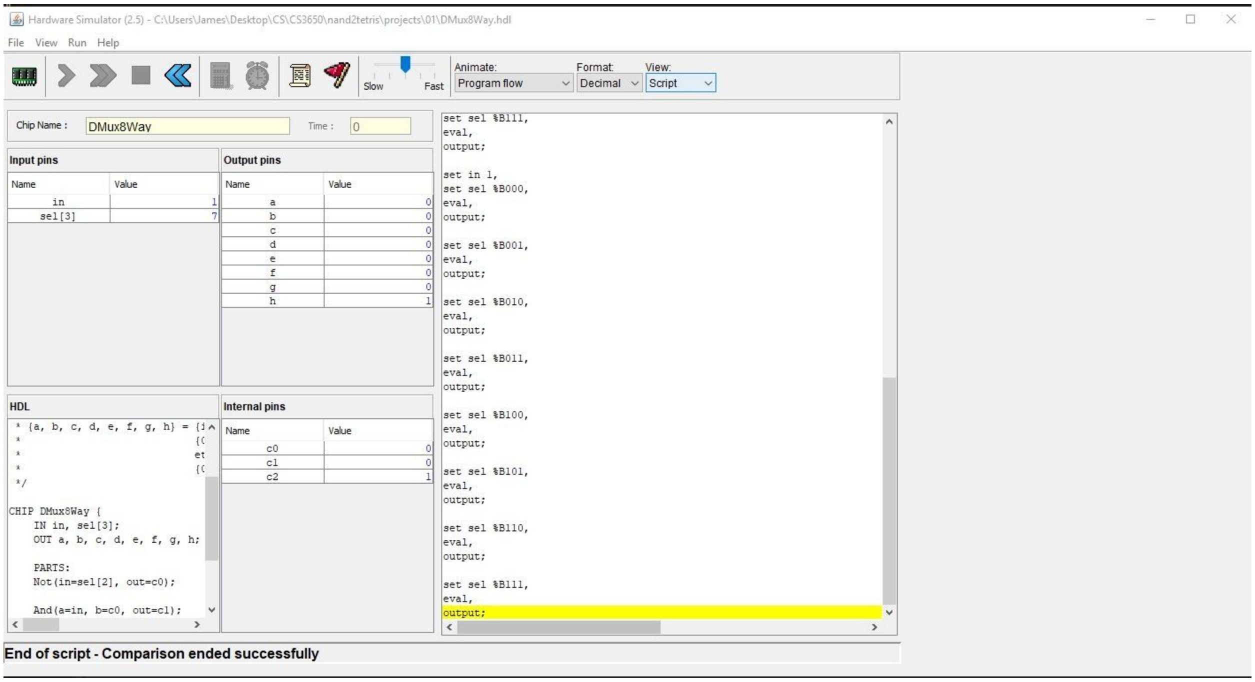
```
 * {a, b, c, d} = {in, 0, 0, 0}
 *                {0, in, 0, 0}
 *                {0, 0, in, 0}
 *                {0, 0, 0, in}
 */

CHIP DMux4Way {
    IN in, sel[2];
    OUT a, b, c, d;

    PARTS:
    Not(in=sel[1], out=c1);

    And(a=c1, b=in, out=inp);
```

Internal pins

Name	Value
c1	0
inp	0
inp1	1

```
compare-to DMux4Way.cmp,
output-list in%B2.1.2 sel%B2.2.2 a%B2.1.2 b%B2.1.2 c%B2.1.2 d%B2.1.2;

set in 0,
set sel %B00,
eval,
output;

set sel %B01,
eval,
output;

set sel %B10,
eval,
output;

set sel %B11,
eval,
output;

set in 1,
set sel %B00,
eval,
output;

set sel %B01,
eval,
output;

set sel %B10,
eval,
output;

set sel %B11,
eval,
output;
```

End of script - Comparison ended successfully


Hardware Simulator (2.5) - C:\Users\James\Desktop\CS\CS3650\nand2tetris\projects\01\DMux8Way.hdl
File View Run Help
Slow
Fast

Animate: Program flow
Format: Decimal
View: Script

Chip Name : DMux8Way
Time : 0

Input pins

Name	Value
in	1
sel[3]	7

Output pins

Name	Value
a	0
b	0
c	0
d	0
e	0
f	0
g	0
h	1

HDL

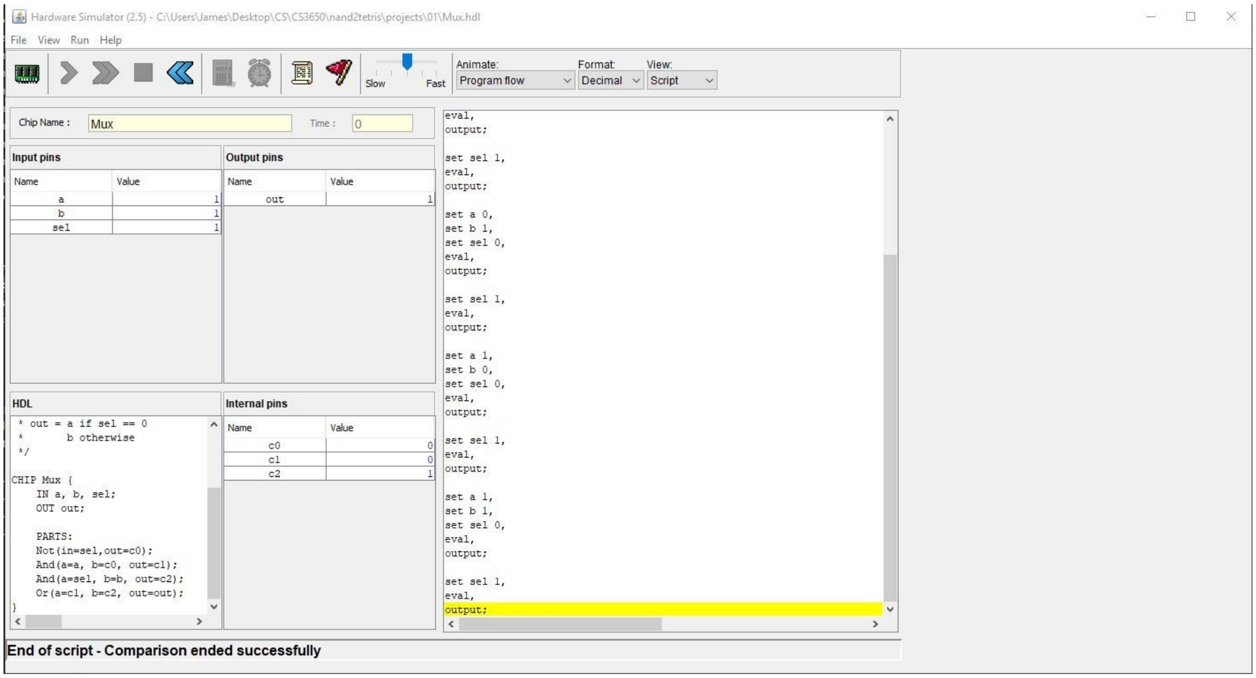
```
 * {a, b, c, d, e, f, g, h} = {i
 *                             {(
 *                             et
 *                             {(
 */

CHIP DMux8Way {
    IN in, sel[3];
    OUT a, b, c, d, e, f, g, h;

    PARTS:
    Not(in=sel[2], out=c0);

    And(a=in, b=c0, out=c1);
```

Internal pins

Name	Value
c0	0
c1	0
c2	1

```
set sel %B111,
eval,
output;

set in 1,
set sel %B000,
eval,
output;

set sel %B001,
eval,
output;

set sel %B010,
eval,
output;

set sel %B011,
eval,
output;

set sel %B100,
eval,
output;

set sel %B101,
eval,
output;

set sel %B110,
eval,
output;

set sel %B111,
eval,
output;
```

End of script - Comparison ended successfully


Hardware Simulator (2.5) - C:\Users\James\Desktop\CS\CS3650\nand2tetris\projects\01\Mux.hdl
File View Run Help
Slow
Fast

Animate: Program flow
Format: Decimal
View: Script

Chip Name : Mux
Time : 0

Input pins

Name	Value
a	1
b	1
sel	1

Output pins

Name	Value
out	1

HDL

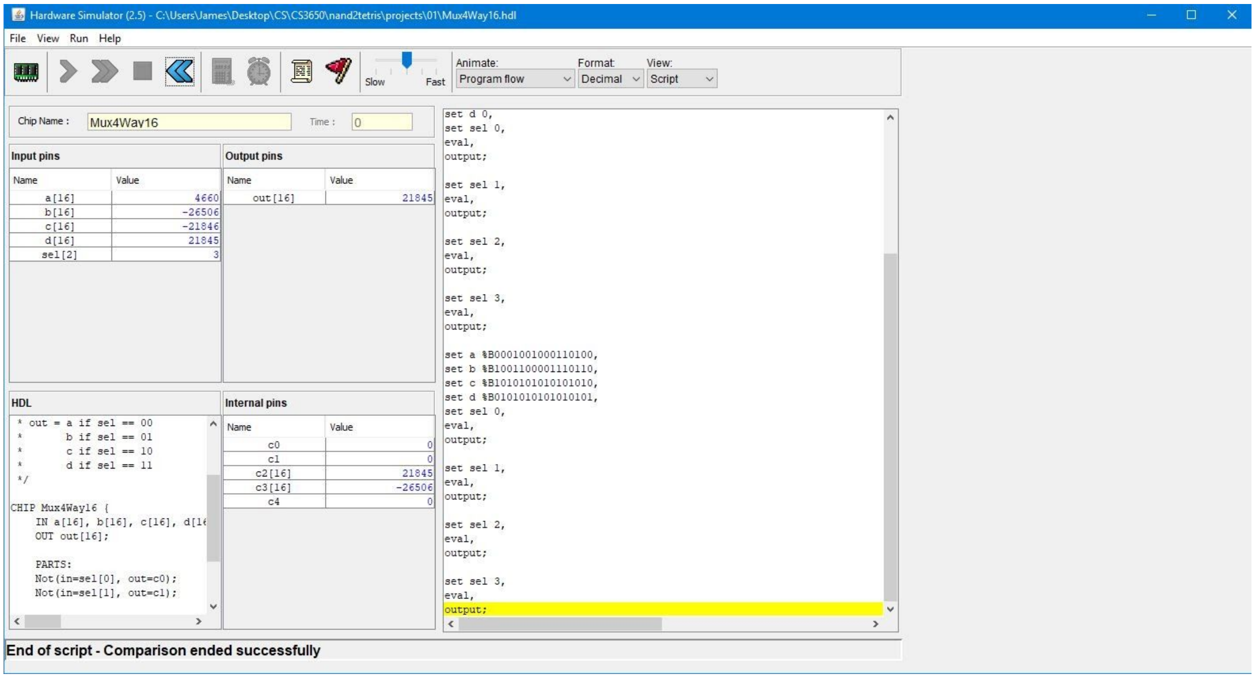
```
 * out = a if sel == 0
 *       b otherwise
 */

CHIP Mux {
    IN a, b, sel;
    OUT out;

    PARTS:
    Not(in=sel,out=c0);
    And(a=a, b=c0, out=c1);
    And(a=sel, b=b, out=c2);
    Or(a=c1, b=c2, out=out);
}
```

Internal pins

Name	Value
c0	0
c1	0
c2	1

```
eval,
output;

set sel 1,
eval,
output;

set a 0,
set b 1,
set sel 0,
eval,
output;

set sel 1,
eval,
output;

set a 1,
set b 0,
set sel 0,
eval,
output;

set sel 1,
eval,
output;

set a 1,
set b 1,
set sel 0,
eval,
output;

set sel 1,
eval,
output;
```

End of script - Comparison ended successfully



Hardware Simulator (2.5) - C:\Users\James\Desktop\CS\CS3650\nand2tetris\projects\01\Mux4Way16.hdl

Animate: Program flow   Format: Decimal   View: Script

Chip Name : Mux4Way16   Time : 0

Input pins

Name	Value
a[16]	4660
b[16]	-26506
c[16]	-21846
d[16]	21845
sel[2]	3

Output pins

Name	Value
out[16]	21845

HDL

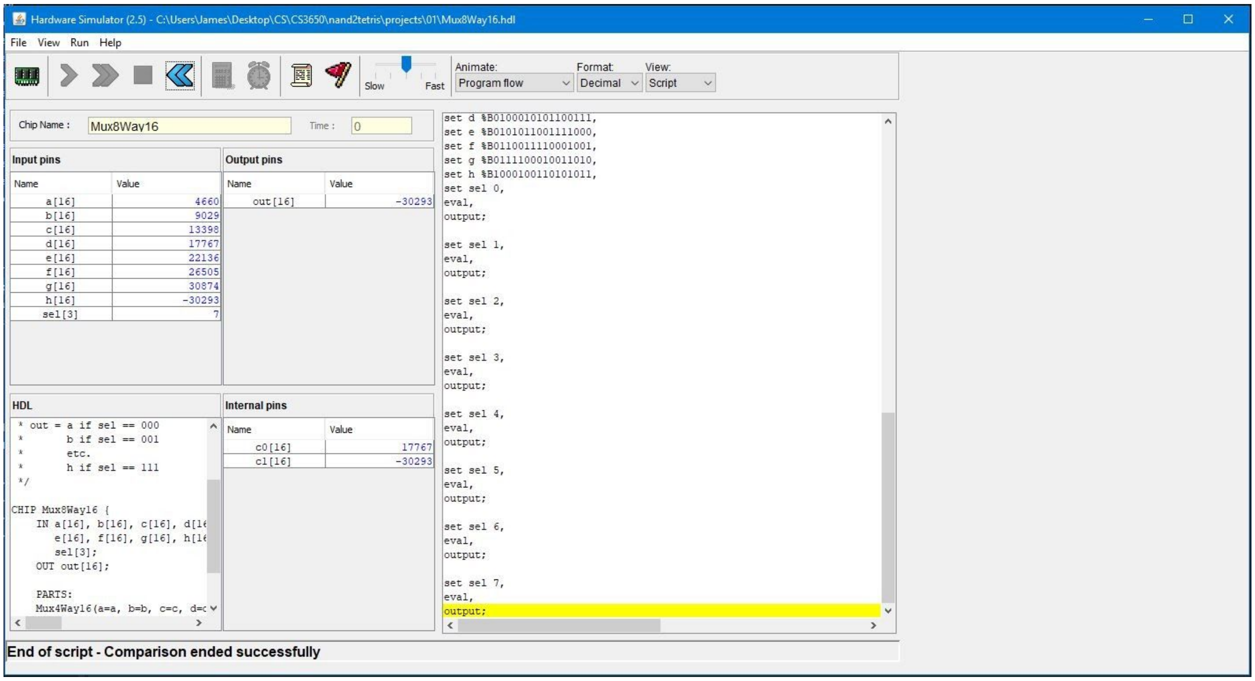
```
 * out = a if sel == 00
 *       b if sel == 01
 *       c if sel == 10
 *       d if sel == 11
 */

CHIP Mux4Way16 {
    IN a[16], b[16], c[16], d[16
    OUT out[16];

    PARTS:
    Not(in=sel[0], out=c0);
    Not(in=sel[1], out=c1);
```

Internal pins

Name	Value
c0	0
c1	0
c2[16]	21845
c3[16]	-26506
c4	0

```
set d 0,
set sel 0,
eval,
output;

set sel 1,
eval,
output;

set sel 2,
eval,
output;

set sel 3,
eval,
output;

set a %B0001001000110100,
set b %B1001100001110110,
set c %B1010101010101010,
set d %B0101010101010101,
set sel 0,
eval,
output;

set sel 1,
eval,
output;

set sel 2,
eval,
output;

set sel 3,
eval,
output;
```

End of script - Comparison ended successfully



Hardware Simulator (2.5) - C:\Users\James\Desktop\CS\CS3650\nand2tetris\projects\01\Mux8Way16.hdl

Animate: Program flow   Format: Decimal   View: Script

Chip Name : Mux8Way16   Time : 0

Input pins

Name	Value
a[16]	4660
b[16]	9029
c[16]	13398
d[16]	17767
e[16]	22136
f[16]	26505
g[16]	30874
h[16]	-30293
sel[3]	7

Output pins

Name	Value
out[16]	-30293

HDL

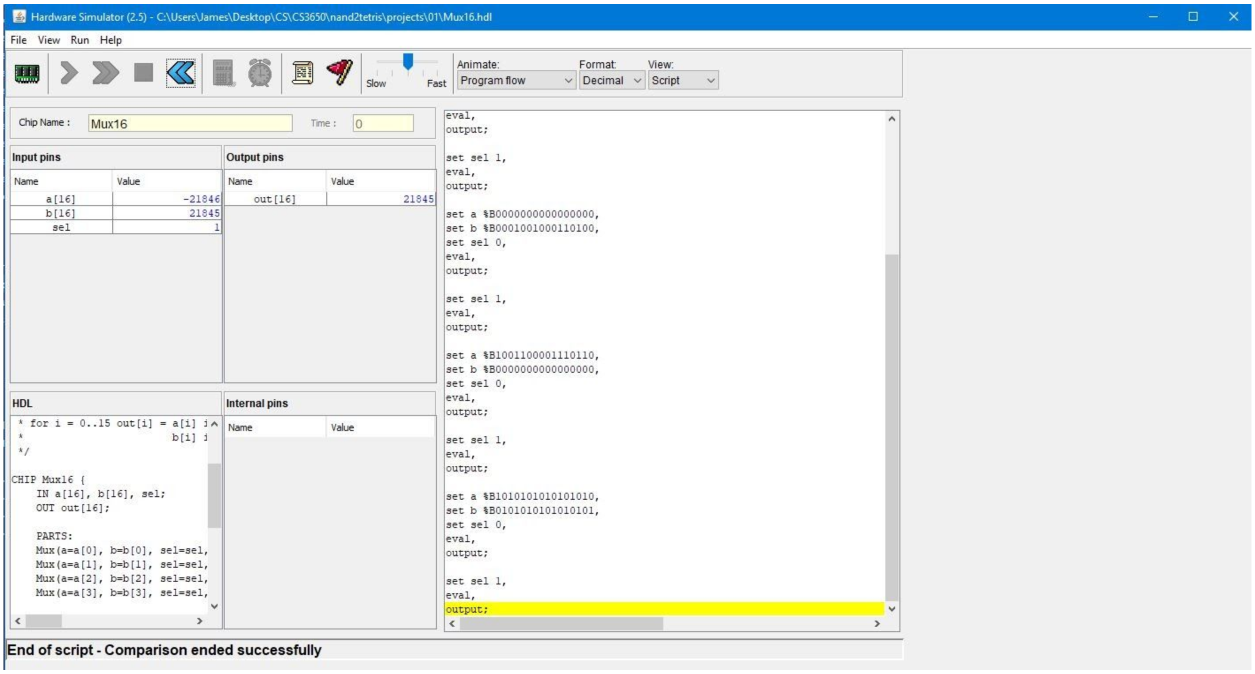
```
 * out = a if sel == 000
 *       b if sel == 001
 *       etc.
 *       h if sel == 111
 */

CHIP Mux8Way16 {
    IN a[16], b[16], c[16], d[16
       e[16], f[16], g[16], h[16
       sel[3];
    OUT out[16];

    PARTS:
    Mux4Way16(a=a, b=b, c=c, d=c
```

Internal pins

Name	Value
c0[16]	17767
c1[16]	-30293

```
set d %B0100010101100111,
set e %B0101011001111000,
set f %B0110011110001001,
set g %B0111100010011010,
set h %B1000100110101011,
set sel 0,
eval,
output;

set sel 1,
eval,
output;

set sel 2,
eval,
output;

set sel 3,
eval,
output;

set sel 4,
eval,
output;

set sel 5,
eval,
output;

set sel 6,
eval,
output;

set sel 7,
eval,
output;
```

End of script - Comparison ended successfully



Hardware Simulator (2.5) - C:\Users\James\Desktop\CS\CS3650\nand2tetris\projects\01\Mux16.hdl

Animate: Program flow
Format: Decimal
View: Script

Chip Name : Mux16
Time : 0

Input pins

Name	Value
a[16]	-21846
b[16]	21845
sel	1

Output pins

Name	Value
out[16]	21845

HDL

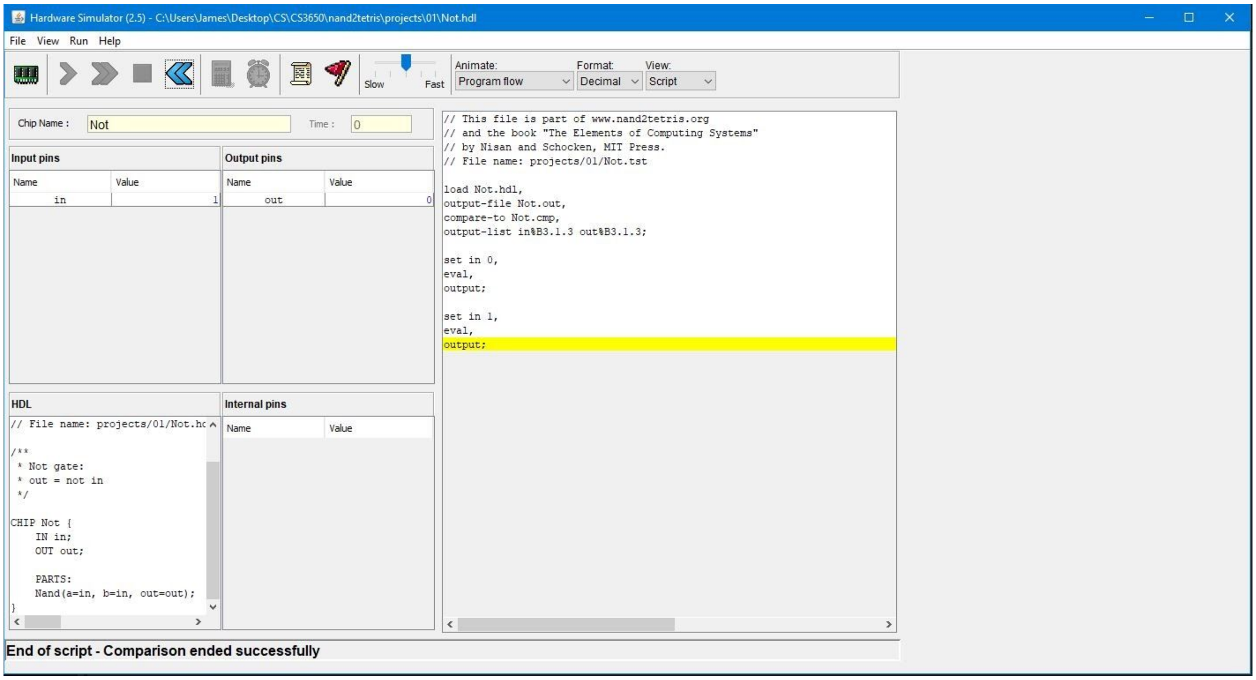
```
 * for i = 0..15 out[i] = a[i] i
 *                         b[i] i
 */

CHIP Mux16 {
    IN a[16], b[16], sel;
    OUT out[16];

    PARTS:
    Mux(a=a[0], b=b[0], sel=sel,
    Mux(a=a[1], b=b[1], sel=sel,
    Mux(a=a[2], b=b[2], sel=sel,
    Mux(a=a[3], b=b[3], sel=sel,
```

Internal pins

Name	Value

```
eval,
output;

set sel 1,
eval,
output;

set a %B0000000000000000,
set b %B0001001000110100,
set sel 0,
eval,
output;

set sel 1,
eval,
output;

set a %B1001100001110110,
set b %B0000000000000000,
set sel 0,
eval,
output;

set sel 1,
eval,
output;

set a %B1010101010101010,
set b %B0101010101010101,
set sel 0,
eval,
output;

set sel 1,
eval,
output;
```

End of script - Comparison ended successfully



Hardware Simulator (2.5) - C:\Users\James\Desktop\CS\CS3650\nand2tetris\projects\01\Not.hdl

Animate: Program flow
Format: Decimal
View: Script

Chip Name : Not
Time : 0

Input pins

Name	Value
in	1

Output pins

Name	Value
out	0

HDL

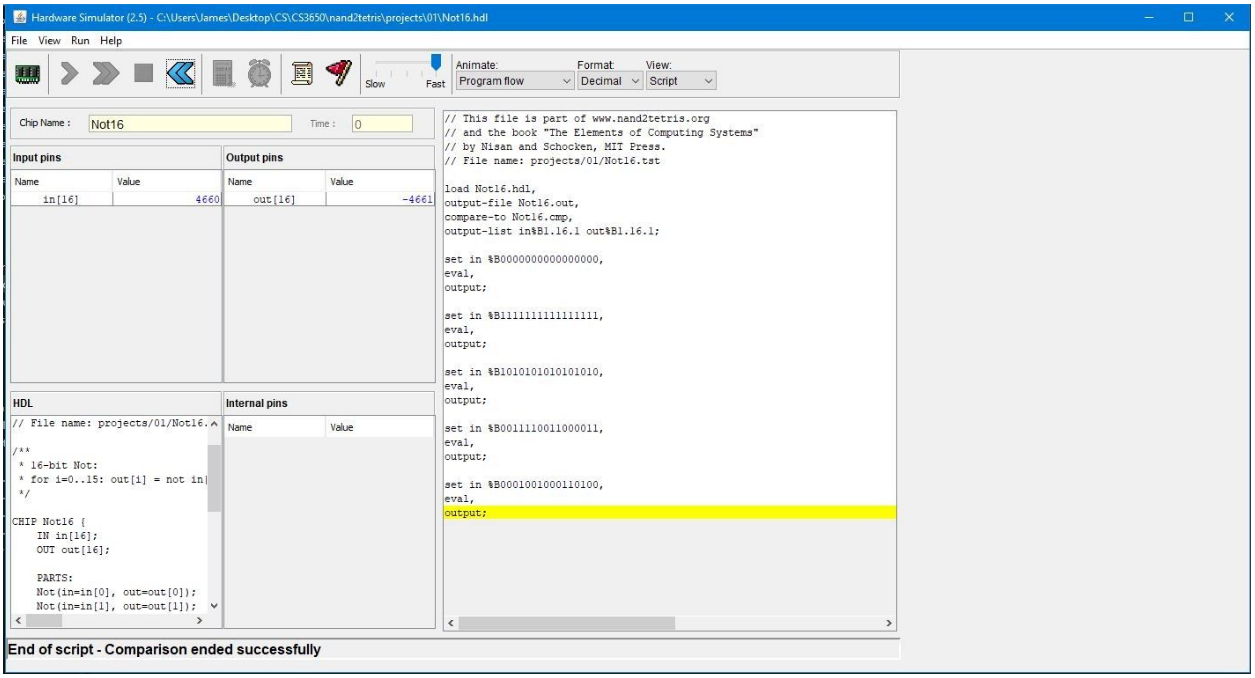
```
// File name: projects/01/Not.hc

/**
 * Not gate:
 * out = not in
 */

CHIP Not {
    IN in;
    OUT out;

    PARTS:
    Nand(a=in, b=in, out=out);
}
```

Internal pins

Name	Value

```
// This file is part of www.nand2tetris.org
// and the book "The Elements of Computing Systems"
// by Nisan and Schocken, MIT Press.
// File name: projects/01/Not.tst

load Not.hdl,
output-file Not.out,
compare-to Not.cmp,
output-list in%B3.1.3 out%B3.1.3;

set in 0,
eval,
output;

set in 1,
eval,
output;
```

End of script - Comparison ended successfully


Hardware Simulator (2.5) - C:\Users\James\Desktop\CS\CS3650\nand2tetris\projects\01\Not16.hdl
File View Run Help
Slow
Fast

Animate: Program flow
Format: Decimal
View: Script

Chip Name : Not16
Time : 0

Input pins

Name	Value
in[16]	4660

Output pins

Name	Value
out[16]	-4661

HDL

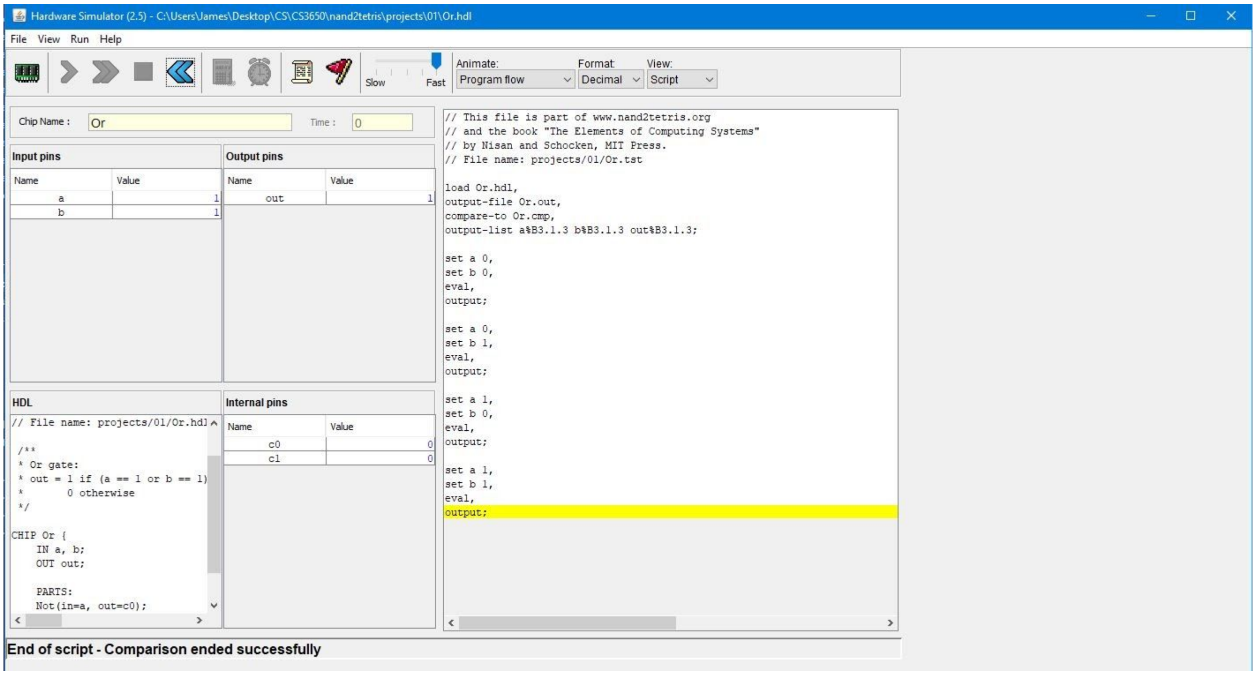
```
// File name: projects/01/Not16.

/**
 * 16-bit Not:
 * for i=0..15: out[i] = not in|
 */

CHIP Not16 {
    IN in[16];
    OUT out[16];

    PARTS:
    Not(in=in[0], out=out[0]);
    Not(in=in[1], out=out[1]);
```

Internal pins

Name	Value

```
// This file is part of www.nand2tetris.org
// and the book "The Elements of Computing Systems"
// by Nisan and Schocken, MIT Press.
// File name: projects/01/Not16.tst

load Not16.hdl,
output-file Not16.out,
compare-to Not16.cmp,
output-list in%B1.16.1 out%B1.16.1;

set in %B0000000000000000,
eval,
output;

set in %B1111111111111111,
eval,
output;

set in %B1010101010101010,
eval,
output;

set in %B0011110011000011,
eval,
output;

set in %B0001001000110100,
eval,
output;
```

End of script - Comparison ended successfully


Hardware Simulator (2.5) - C:\Users\James\Desktop\CS\CS3650\nand2tetris\projects\01\Or.hdl
File View Run Help
Slow
Fast

Animate: Program flow
Format: Decimal
View: Script

Chip Name : Or
Time : 0

Input pins

Name	Value
a	1
b	1

Output pins

Name	Value
out	1

HDL

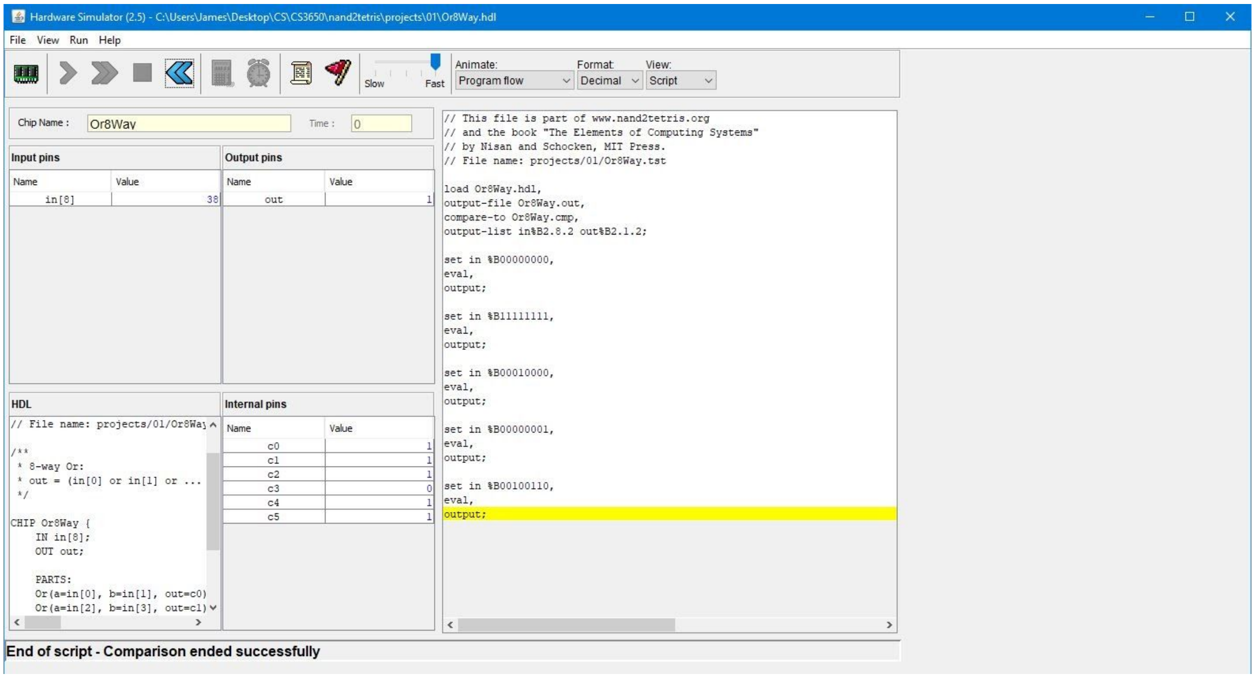
```
// File name: projects/01/Or.hdl

 /**
 * Or gate:
 * out = 1 if (a == 1 or b == 1)
 *       0 otherwise
 */

CHIP Or {
    IN a, b;
    OUT out;

    PARTS:
    Not(in=a, out=c0);
```

Internal pins

Name	Value
c0	0
c1	0

```
// This file is part of www.nand2tetris.org
// and the book "The Elements of Computing Systems"
// by Nisan and Schocken, MIT Press.
// File name: projects/01/Or.tst

load Or.hdl,
output-file Or.out,
compare-to Or.cmp,
output-list a%B3.1.3 b%B3.1.3 out%B3.1.3;

set a 0,
set b 0,
eval,
output;

set a 0,
set b 1,
eval,
output;

set a 1,
set b 0,
eval,
output;

set a 1,
set b 1,
eval,
output;
```

End of script - Comparison ended successfully



Hardware Simulator (2.5) - C:\Users\James\Desktop\CS\CS3650\nand2tetris\projects\01\Or8Way.hdl

Animate: Program flow

Format: Decimal

View: Script

Chip Name : Or8Way

Time : 0

Input pins

Name	Value
in[8]	38

Output pins

Name	Value
out	1

HDL

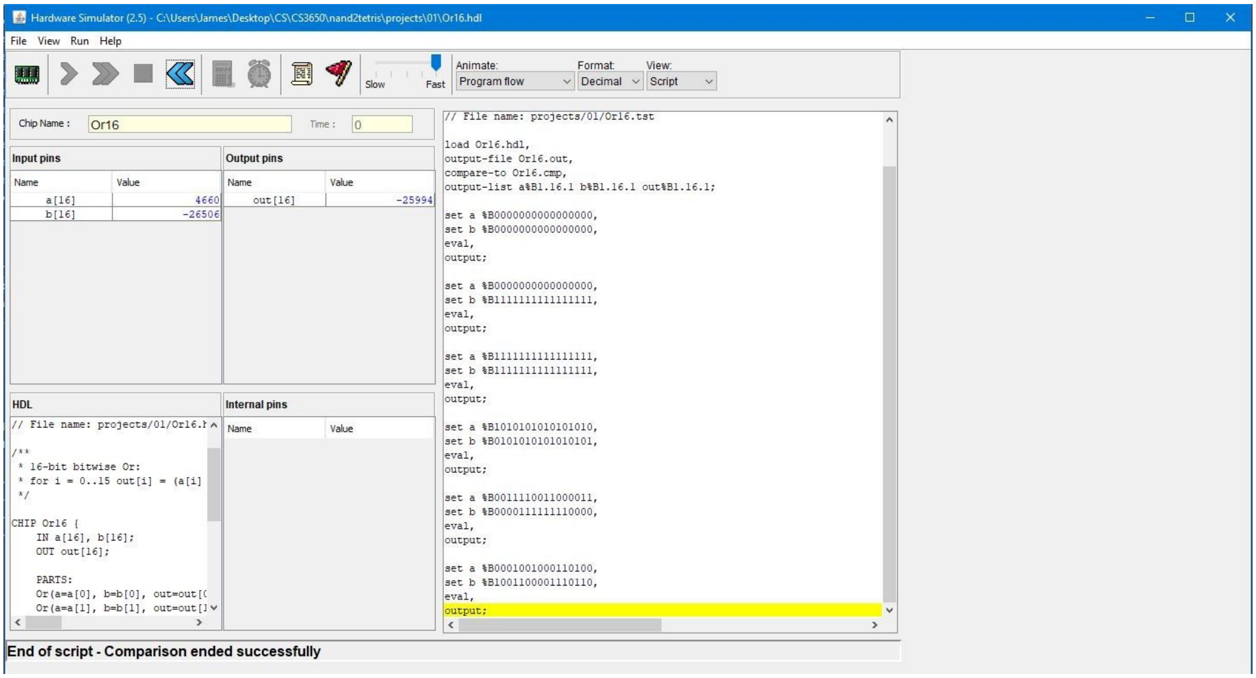
```
// File name: projects/01/Or8Way

/**
 * 8-way Or:
 * out = (in[0] or in[1] or ...
 */

CHIP Or8Way {
    IN in[8];
    OUT out;

    PARTS:
    Or(a=in[0], b=in[1], out=c0)
    Or(a=in[2], b=in[3], out=c1)
```

Internal pins

Name	Value
c0	1
c1	1
c2	1
c3	0
c4	1
c5	1

```
// This file is part of www.nand2tetris.org
// and the book "The Elements of Computing Systems"
// by Nisan and Schocken, MIT Press.
// File name: projects/01/Or8Way.tst

load Or8Way.hdl,
output-file Or8Way.out,
compare-to Or8Way.cmp,
output-list in%B2.8.2 out%B2.1.2;

set in %B00000000,
eval,
output;

set in %B11111111,
eval,
output;

set in %B00010000,
eval,
output;

set in %B00000001,
eval,
output;

set in %B00100110,
eval,
output;
```

End of script - Comparison ended successfully



Hardware Simulator (2.5) - C:\Users\James\Desktop\CS\CS3650\nand2tetris\projects\01\Or16.hdl

Animate: Program flow

Format: Decimal

View: Script

Chip Name : Or16

Time : 0

Input pins

Name	Value
a[16]	4660
b[16]	-26506

Output pins

Name	Value
out[16]	-25994

HDL

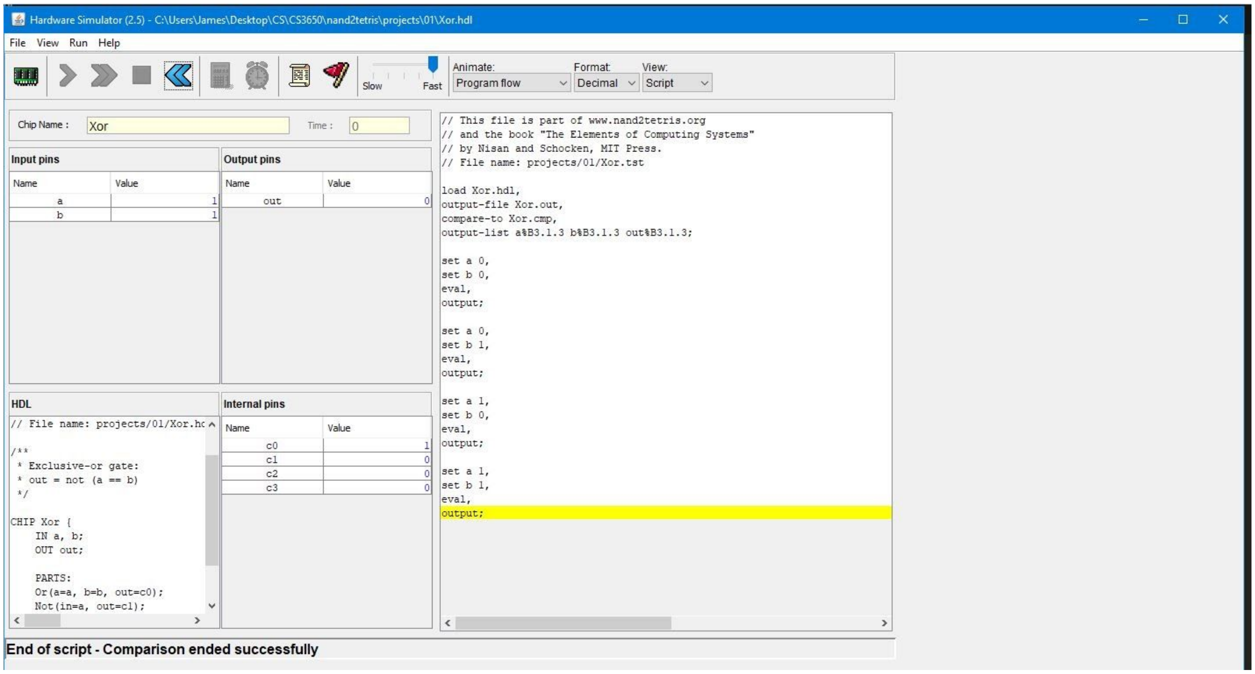
```
// File name: projects/01/Or16.h

/**
 * 16-bit bitwise Or:
 * for i = 0..15 out[i] = (a[i]
 */

CHIP Or16 {
    IN a[16], b[16];
    OUT out[16];

    PARTS:
    Or(a=a[0], b=b[0], out=out[(
    Or(a=a[1], b=b[1], out=out[1
```

Internal pins

Name	Value

```
// File name: projects/01/Or16.tst

load Or16.hdl,
output-file Or16.out,
compare-to Or16.cmp,
output-list a%B1.16.1 b%B1.16.1 out%B1.16.1;

set a %B0000000000000000,
set b %B0000000000000000,
eval,
output;

set a %B0000000000000000,
set b %B1111111111111111,
eval,
output;

set a %B1111111111111111,
set b %B1111111111111111,
eval,
output;

set a %B1010101010101010,
set b %B0101010101010101,
eval,
output;

set a %B0011110011000011,
set b %B0000111111110000,
eval,
output;

set a %B0001001000110100,
set b %B1001100001110110,
eval,
output;
```

End of script - Comparison ended successfully



Hardware Simulator (2.5) - C:\Users\James\Desktop\CS\CS3650\nand2tetris\projects\01\Xor.hdl

Animate: Program flow

Format: Decimal

View: Script

Chip Name : Xor

Time : 0

Input pins

Name	Value
a	1
b	1

Output pins

Name	Value
out	0

HDL

```
// File name: projects/01/Xor.hc

/**
 * Exclusive-or gate:
 * out = not (a == b)
 */

CHIP Xor {
    IN a, b;
    OUT out;

    PARTS:
    Or(a=a, b=b, out=c0);
    Not(in=a, out=c1);
```

Internal pins

Name	Value
c0	1
c1	0
c2	0
c3	0

```
// This file is part of www.nand2tetris.org
// and the book "The Elements of Computing Systems"
// by Nisan and Schocken, MIT Press.
// File name: projects/01/Xor.tst

load Xor.hdl,
output-file Xor.out,
compare-to Xor.cmp,
output-list a%B3.1.3 b%B3.1.3 out%B3.1.3;

set a 0,
set b 0,
eval,
output;

set a 0,
set b 1,
eval,
output;

set a 1,
set b 0,
eval,
output;

set a 1,
set b 1,
eval,
output;
```

End of script - Comparison ended successfully