Contents

1 Basic Test Results 2

2 README 3

3 And.hdl 4

4 And16.hdl 5

5 DMux.hdl 6

6 DMux4Way.hdl 7

7 DMux8Way.hdl 8

8 Mux.hdl 9

9 Mux16.hdl 10

10 Mux4Way16.hdl 11

11 Mux8Way16.hdl 12

12 Not.hdl 13

13 Not16.hdl 14

14 Or.hdl 15

15 Or16.hdl 16

16 Or8Way.hdl 17

17 Xor.hdl 18

1 Basic Test Results

```
********** TESTING FOLDER STRUCTURE START **********
Running test1.sh:
Your logins are: muaz.abdeen, is that ok?

********** TESTING FOLDER STRUCTURE END **********

********** PROJECT TEST START **********

And passed test
Not passed test
Or passed test
Or8Way passed test

********** PROJECT TEST END **********
```

2 README

```
muaz.abdeen
===============================================================================
Muaz Abdeen, ID 300575297, muaz.abdeen@mail.huji.ac.il
===============================================================================

                           Project 1 - Boolean Logic
                           -------------------------


Submitted Files
---------------
(0)  README         - This file.
(1)  And.hdl         - And logic gate implementation.
(2)  And16.hdl         - And16 logic gate implementation.
(3)  DMux.hdl         - DMux logic gate implementation.
(4)  DMux4Way.hdl     - DMux4Way logic gate implementation.
(5)  DMux8Way.hdl     - DMux8Way logic gate implementation.
(6)  Mux.hdl         - Mux logic gate implementation.
(7)  Mux4Way16.hdl     - Mux4Way16 logic gate implementation.
(8)  Mux8Way16.hdl     - Mux8Way16 logic gate implementation.
(9)  Mux16.hdl         - Mux16 logic gate implementation.
(10) Not.hdl         - Not logic gate implementation.
(11) Not16.hdl        - Not16 logic gate implementation.
(12) Or.hdl         - Or logic gate implementation.
(13) Or8Way.hdl     - Or8Way logic gate implementation.
(14) Or16.hdl         - Or16 logic gate implementation.
(15) Xor.hdl         - Xor logic gate implementation.


Remarks
-------
* ...
```

3 And.hdl

```
// 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:
    Nand(a=a ,b=b, out=res);
    Not(in=res , out=out);
}
```

4 And16.hdl

```
// 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]);
}
```

5 DMux.hdl

```
// 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:
    Nand(a=in ,b=sel, out=aNandSel);
    And(a=in, b=aNandSel, out=a);
    Not(in=aNandSel ,out=b);
}
```

6 DMux4Way.hdl

```
// 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/DMux4Way.hdl

/**
 * 4-way demultiplexor:
 * {a, b, c, d} = {in, 0, 0, 0} if sel == 00
 *                {0, in, 0, 0} if sel == 01
 *                {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:
    DMux(in=in ,sel=sel[1] ,a=AB ,b=CD);
    DMux(in=AB ,sel=sel[0] ,a=a ,b=b);
    DMux(in=CD ,sel=sel[0] ,a=c ,b=d);
}
```

7 DMux8Way.hdl

```
// 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:
    DMux(in=in ,sel=sel[2] ,a=ABCD ,b=EFGH);
    DMux4Way(in=ABCD ,sel=sel[0..1] ,a=a ,b=b ,c=c ,d=d);
    DMux4Way(in=EFGH ,sel=sel[0..1] ,a=e ,b=f ,c=g ,d=h);
}
```

8 Mux.hdl

```
// 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=NotSel);
    Nand(a=NotSel, b=a, out=aNandNotSel);
    Nand(a=sel, b=b, out=bNandSel);
    Nand(a=aNandNotSel, b=bNandSel, out=out);
}
```

9 Mux16.hdl

```
// 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]);
}
```

10 Mux4Way16.hdl

```
// 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:
    Mux16(a=a ,b=b ,sel=sel[0] ,out=aMux16b);
    Mux16(a=c ,b=d ,sel=sel[0] ,out=cMux16d);
    Mux16(a=aMux16b ,b=cMux16d ,sel=sel[1] ,out=out);
}
```

11 Mux8Way16.hdl

```
// 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=MuxAtoD);
    Mux4Way16(a=e ,b=f ,c=g ,d=h ,sel=sel[0..1] ,out=MuxEtoH);
    Mux16(a=MuxAtoD ,b=MuxEtoH ,sel=sel[2] ,out=out);

}
```

12 Not.hdl

```
// 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);
}
```

13 Not16.hdl

```
// 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.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]);
}
```

14 Or.hdl

```
// 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=aOut);
    Not(in=b ,out=bOut);
    Nand(a=aOut, b=bOut, out=out);
}
```

15 Or16.hdl

```
// 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]);
}
```

16 Or8Way.hdl

```
// 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=res1);
    Or(a=in[2] ,b=in[3] ,out=res2);
    Or(a=in[4] ,b=in[5] ,out=res3);
    Or(a=in[6] ,b=in[7] ,out=res4);
    Or(a=res1, b=res2, out=res5);
    Or(a=res3, b=res4, out=res6);
    Or(a=res5, b=res6, out=out);
}
```

17 Xor.hdl

```
// 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:
    Nand(a=a ,b=b, out=aNandb);
    Nand(a=aNandb, b=a, out=res1);
    Nand(a=aNandb, b=b, out=res2);
    Nand(a=res1, b=res2, out=out);
}
```