

Primer Laboratorio GRUPO MACACO

Estudiantes:

Luis Mario Toscano Palomino.

Jhon Anderson Vargas Gómez.

Jeider Torres Martínez.

Samuel David Traslaviña Mateus.

Luis Alejandro Vecino Daza.

Profesor:

Carlos Jaime Barrios Hernández

Grupo:

A1

Tabla de contenido

Introducción.....	3
Compuerta NOT	3
Compuerta AND.....	3
Compuerta OR.....	4
Compuerta XOR.....	5
MUX.....	5
DMUX.....	6
Preguntas adicionales.....	7
Consideraciones importantes para Nand2Tetris	7
Herramientas similares a Nand2Tetris	8

Introducción

En este documento, encontrarás toda la información necesaria para crear desde cero las compuertas lógicas, demultiplexores y multiplexores. Además, se explican los fundamentos teóricos basados en las tablas de verdad de cada una de las compuertas, y se incluyen comentarios sobre los códigos utilizados para facilitar su comprensión.

Compuerta NOT

INPUT A	OUTPUT
0	1
1	0

$$\overline{A \cdot A} = \overline{A}$$

Código:

```
CHIP Not {
  IN in;
  OUT out;

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

Compuerta AND

INPUT A	INPUT B	OUTPUT
0	0	0
0	1	0
1	0	0
1	1	1

$$A \cdot B = \overline{\overline{A \cdot B}}$$

Código:

```
CHIP And {
    IN a, b;
    OUT out;

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

Compuerta OR

INPUT A	INPUT B	OUTPUT
0	0	0
0	1	1
1	0	1
1	1	1

$$A + B = \overline{\overline{A \cdot B}}$$

Código:

```
CHIP Or {
    IN a, b;
    OUT out;

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

Compuerta XOR

INPUT A	INPUT B	OUTPUT
0	0	0
0	1	1
1	0	1
1	1	0

$$A \oplus B = A \cdot \overline{B} + \overline{A} \cdot B$$

Código:

```
CHIP Xor {  
    IN a, b;  
    OUT out;  
  
    PARTS:  
        //(-a) , (-b)  
        Nand( a = a, b = a, out = nota );  
        Nand( a = b, b = b, out = notb );  
  
        // aux1 = -(a*(-b)) , aux2 = -((-a)*b)  
        Nand( a = a, b = notb, out = aux1 );  
        Nand( a = nota, b = b, out = aux2 );  
  
        // xor = -aux1 + -aux2 = -(aux1 * aux2)  
        Nand( a = aux1, b = aux2, out = out);  
}
```

MUX

S	OUTPUT
0	A
1	B

$$\bar{S} \cdot A + S \cdot B$$

Código:

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

    PARTS:
        //notSel
        Nand( a = sel, b = sel, out = notsel) ;

        //aux1 = -(notsel * a) , aux2 = -(sel * b)
        Nand( a = notsel , b = a, out = aux1 );
        Nand( a = sel , b = b, out = aux2 );

        //mux = (notsel * a) + (sel * b) = -(aux1 * aux2)
        Nand( a = aux1, b = aux2, out = out);
}
```

DMUX

S	IN	OUTPUT A	OUTPUT B
0	0	0	0
0	1	1	0
1	0	0	0
1	1	0	1

$$\bar{S} \cdot IN = A$$

$$S \cdot IN = B$$

Código:

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

    PARTS:
        //notSel
        Nand( a = sel, b = sel, out = notsel );

        //aux1 = -(-s * in)
        Nand( a = notsel , b = in , out = aux1);

        //aux2 = -(s * in)
        Nand( a = sel , b = in , out = aux2);

        //output a = -aux1
        Nand( a = aux1, b = aux1, out = a );

        //output b = - aux2
        Nand( a = aux2, b = aux2, out = b );
}
```

Preguntas adicionales

Consideraciones importantes para Nand2Tetris

1. Entender lo básico: Antes de empezar, debes de saber qué son las puertas lógicas, como las NAND, y cómo se combinan para hacer cosas más complejas. Esto es clave porque vas a construir desde cero, y sin entender lo básico, te puedes perder rápido y no llegas al resultado que esperas.

2. Experimentar mucho: Nand2Tetris es una oportunidad para aprender, así que no dudes en probar diferentes cosas. Juega con el código, cambia parámetros, y ve qué pasa. Así es como más vas a aprender.

Herramientas similares a Nand2Tetris

1. Logisim: Es como un simulador de circuitos. Te permite hacer circuitos lógicos dibujando y conectando componentes. Es útil si quieres ver cómo funcionan los circuitos sin tener que programar tanto.

2. Digital: Similar a Logisim, pero con otra interfaz. También te deja crear y probar circuitos lógicos, y es genial si quieres entender mejor cómo se conectan las cosas a nivel de hardware antes de pasar a algo más avanzado como Nand2Tetris.