

Visual Studio Code interface showing the HDL code for a 1-bit register (Bit.hdl) and the Hardware Simulator (2.5) window.

Bit.hdl Code:

```
1 CHIP Bit {
2   IN in, load;
3   OUT out;
4
5   PARTS:
6     // Put your code here:
7     DFF(in = muxout,out = ffout,out = out);
8     Mux(a = ffout,b = in,sel = load,out = muxout);
9
10 }
```

Hardware Simulator (2.5) - Bit (Clocked)

Time: 3

Input pins		Output pins	
Name	Value	Name	Value
in	0	out	0
load	1		

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: tools/builtIn/Bit.hdl

/**
 * 1-bit register.
 * If load[t] == 1 then out[t+1] = in[t]
 * else out[t+1] = out[t]
 */

CHIP Bit {
    IN in, load;
    OUT out;
```

End of script

Selección Ver Ir Ejecutar Terminal Ayuda Registro8.hdl - Ejercicios Arqui - Visual Studio

... Bit.hdl OchoPos.hdl **Registro8.hdl** SixTeenPos.hdl SixtyFourPos.hdl

Lab 04 Registros en HDL > Registro8.hdl

```
1 CHIP Registro8 {
2     IN in[8], flag;
3     OUT out[8];
4
5     PARTS:
6         // 8 Bits de Memoria
7         Bit(in = in[0], load = flag, out = out[0]);
8         Bit(in = in[1], load = flag, out = out[1]);
9         Bit(in = in[2], load = flag, out = out[2]);
10        Bit(in = in[3], load = flag, out = out[3]);
11        Bit(in = in[4], load = flag, out = out[4]);
12        Bit(in = in[5], load = flag, out = out[5]);
13        Bit(in = in[6], load = flag, out = out[6]);
14        Bit(in = in[7], load = flag, out = out[7]);
15
16 }
```

Hardware Simulator (2.5) - D:\NAND2TETRIS\Ejercicios Arqui\Lab 04 - Registros en HDL\Registro8.hdl

File View Run Help

Chip Name: **Registro8 (Clocked)** Time: 4

Input pins		Output pins	
Name	Value	Name	Value
in[8]	11111111	out[8]	11111111
flag	1		

HDL		Internal pins	
		Name	Value
CHIP Registro8 { IN in[8], flag; OUT out[8]; PARTS: // 8 Bits de Memoria Bit(in = in[0], load = flag, Bit(in = in[1], load = flag, Bit(in = in[2], load = flag, Bit(in = in[3], load = flag, Bit(in = in[4], load = flag, Bit(in = in[5], load = flag, Bit(in = in[6], load = flag, Bit(in = in[7], load = flag, Bit(in = in[7], load = flag, ✓			

```
set in %B00111111,
set flag 1,
// set address 0,
tick,
output;
tock,
output;

set in %B11001101,
set flag 0,
// set address 0,
tick,
output;
tock,
output;

set in %B00000000,
set flag 1,
// set address 0,
tick,
output;
tock,
output;

set in %B11111111,
set flag 1,
// set address 0,
tick,
output;
tock,
output;
output;
```

End of script

er Ir Ejecutar Terminal Ayuda OchoPos.hdl - Ejercicios Arqui - Visual Studio Code

Bit.hdl OchoPos.hdl X Registro8.hdl SixTeenPos.hdl SixtyFourPos.hdl

Lab 04 Registros en HDL > OchoPos.hdl

```

1 CHIP OchoPos {
2     IN in[8], flag, address[3];
3     OUT out[8];
4
5     PARTS:
6         // Put your code here:
7         DMux8Way(in = flag, sel = address, a = a, b = b, c = c, d = d, e = e, f = f, g = g, h = h);
8         Registro8(in = in, flag = a, out = outa);
9         Registro8(in = in, flag = b, out = outb);
10        Registro8(in = in, flag = c, out = outc);
11        Registro8(in = in, flag = d, out = outd);
12        Registro8(in = in, flag = e, out = oute);
13        Registro8(in = in, flag = f, out = outf);
14        Registro8(in = in, flag = g, out = outg);
15        Registro8(in = in, flag = h, out = outh);
16
17        Mux8Way8(a = outa, b = outb, c = outc, d = outd, e = oute, f = outf, g = outg, h = outh, sel = address, out = out);
18
19
20 }

```

Hardware Simulator (2.5) - D:\NAND2TETRIS\Ejercicios Arqui\Lab 04 Registros en HDL\OchoPos.hdl

File View Run Help

Chip Name : OchoPos (Clocked) Time : 4

Input pins		Output pins	
Name	Value	Name	Value
in[8]	01011010	out[8]	01011010
flag	1		
address[3]	110		

Internal pins	
Name	Value
a	0
b	0
c	0
d	0
e	0
f	0
g	1
h	0
outa[8]	00000000
outb[8]	01101100
outc[8]	00110000
outd[8]	11111111
oute[8]	00000000
outf[8]	00000000
outg[8]	00000000
outh[8]	00000000

```

tick,
output;
tock,
output;

set in %B00110000,
set flag 1,
set address %B010,
// set address 0,
tick,
output;
tock,
output;

set in %B11111111,
set flag 1,
set address %B011,
// set address 0,
tick,
output;
tock,
output;

set in %B01011010,
set flag 1,
set address %B110,
// set address 0,
tick,
output;
tock,
output;

```

End of script

```

SixtyFourPos.hdl  SixTeenPos.hdl X  OchoPos.hdl  Registro8.hdl  Bit.hdl
Lab 04 Registros en HDL > SixTeenPos.hdl
1  CHIP SixTeenPos
2
3      IN in[8], flag, addressBit, address8Pos[3];
4      OUT out[8];
5
6      PARTS:
7          // Put your code here:
8
9      DMux(in=flag, sel=addressBit, a=a, b=b);
10     //2 registros de 8 = 16 Bits
11     OchoPos(in = in, flag = a, address = address8Pos, out = outa);
12     OchoPos(in = in, flag = b, address = address8Pos, out = outb);
13
14     //mux
15     Mux8(a=outa,b=outb,sel=addressBit, out = out );
16
17     // Mux8Way16(a = outa,b = outb,c = outc,d = outd,e = oute,f = outf,g = outg,h = outh,sel = address[3..5],out = out);
18

```

Hardware Simulator (2.5) - D:\NAND2TETRIS\Ejercicios Arqu\Lab 04 Registros en HDL\SixTeenPos.hdl

File View Run Help

Animate: Program flow Format: Binary View: Script

Chip Name: SixTeenPos (Clocked) Time: 8

Input pins		Output pins	
Name	Value	Name	Value
in[8]	11111111	out[8]	00000000
flag	0		
addressBit	1		
address8Pos[3]	101		

Internal pins	
Name	Value
a	0
b	0
outa[8]	00000000
outb[8]	00000000

HDL

```

CHIP SixTeenPos {
    IN in[8], flag, addressBit,
    OUT out[8];

    PARTS:
        // Put your code here:

        DMux(in=flag, sel=addressBit,
        //2 registros de 8 = 16 Bits
        OchoPos(in = in, flag = a, address = address8Pos, out = outa);
        OchoPos(in = in, flag = b, address = address8Pos, out = outb);

        //mux

```

Internal pins

```

set address8Pos %B011,
tick,
output;
tock,
output;

set in %B00000000,
set flag 1,
set addressBit 0,
set address8Pos %B110,
tick,
output;
tock,
output;

// lee datos
set in %B110011,
set flag 0,
set addressBit 1,
set address8Pos %B000,
tick,
output;
tock,
output;

set in %B11111111,
set flag 0,
set addressBit 1,
set address8Pos %B101,
tick,
output;
tock,
output;

```

End of script

```

1  CHIP SixtyFourPos {
2
3      IN in[8], flag, address[3];
4      OUT out[8];
5
6  PARTS:
7      // Put your code here:
8      DMux8Way(in = flag, sel = address, a = a, b = b, c = c, d = d, e = e, f = f, g = g, h = h);
9      OchoPos(in = in, flag = a, out = outa);
10     OchoPos(in = in, flag = b, out = outb);
11     OchoPos(in = in, flag = c, out = outc);
12     OchoPos(in = in, flag = d, out = outd);
13     OchoPos(in = in, flag = e, out = oute);
14     OchoPos(in = in, flag = f, out = outf);
15     OchoPos(in = in, flag = g, out = outg);
16     OchoPos(in = in, flag = h, out = outh);
17
18     Mux8Way8(a = outa, b = outb, c = outc, d = outd, e = oute, f = outf, g = outg, h = outh, sel = address, out = out);
19
20 }

```

Hardware Simulator (2.5) - D:\NAND2TETRIS\Ejercicios Arqui\Lab 04 Registros en HDL\SixteenPos.hdl

File View Run Help

Chip Name : SixteenPos (Clocked) Time : 8

Input pins		Output pins	
Name	Value	Name	Value
in[8]	11111111	out[8]	00000000
flag	0		
addressBit	1		
address8Pos[3]	101		

HDL

```

CHIP SixteenPos {
    IN in[8], flag, addressBit,
    OUT out[8];

    PARTS:
        // Put your code here:

        DMux(in=flag, sel=addressBit,
        // 2 registros de 8 = 16 Bits
        OchoPos(in = in, flag = a, out = outa);
        OchoPos(in = in, flag = b, out = outb);
        //mu

```

Internal pins

Name	Value
a	0
b	0
outa[8]	00000000
outb[8]	00000000

```

set address8Pos $B011,
tick,
output;
tock,
output;

set in $B00000000,
set flag 1,
set addressBit 0,
set address8Pos $B110,
tick,
output;
tock,
output;

// lee datos
set in $B110011,
set flag 0,
set addressBit 1,
set address8Pos $B000,
tick,
output;
tock,
output;

set in $B11111111,
set flag 0,
set addressBit 1,
set address8Pos $B101,
tick,
output;
tock,
output;

```

End of script