

Project3 for Intro to Computer Systems 2025 spring

Yiming Cheng

12450588

Project3 for Intro to Computer Systems 2025 spring

[Result:](#)

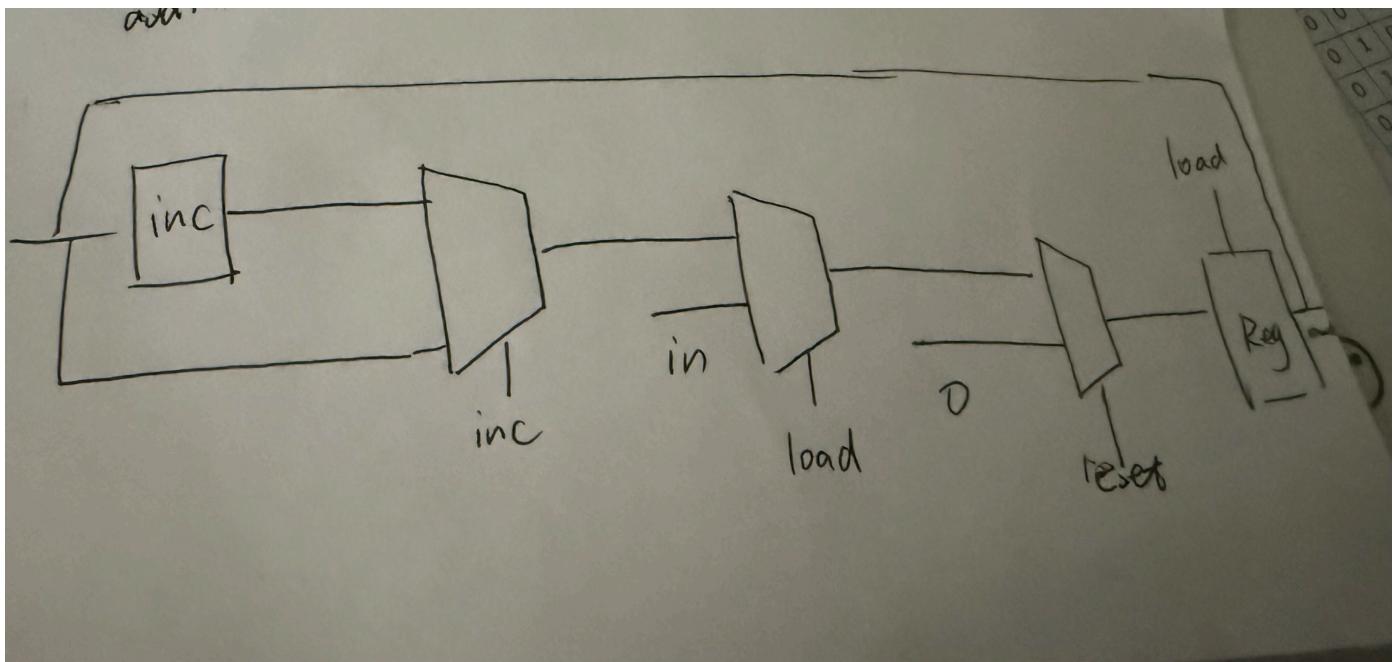
[Details](#)

Result:

1. Bit (folder a) `work`
2. Register (folder a) `work`
3. RAM8 (folder a) `work`
4. RAM64 (folder a) `work`
5. RAM512 (folder b) `work`
6. RAM4K (folder b) `work`

7. RAM16K (folder b) ***work***

8. PC (folder a) ***work***



Details

1. Bit (folder a)

NAND2Tetris / Hardware Simulator / Bit.hdl

HDL Builtin Project 3 Bit Chip Bit Eval Reset Clock: 29+

```

4 // File name: projects/3/a/Bit.hdl
5 /**
6 * 1-bit register:
7 * If Load is asserted, the register's value is set to in;
8 * Otherwise, the register maintains its current value:
9 * if (Load(t)) out(t+1) = in(t), else out(t+1) = out(t)
10 */
11 CHIP Bit {
12     IN in, load;
13     OUT out;
14
15     PARTS:
16         Mux(a= DFFout, b= in , sel= load , out= out1);
17         DFF(in= out1 , out= DFFout,out =out );
18

```

Input pins

- in 0
- load 0

Output pins

- out 1

Internal pins

- DFFout 1
- out1 1

Test Slow Fast

Test Script	Compare File	Output File	Diff Table
1 time in load out			
2 0+ 0 0 0			
3 1 0 0 0			
4 1+ 0 1 0			
5 2 0 1 0			
6 2+ 1 0 0			
7 3 1 0 0			
8 3+ 1 1 0			
9 / 1 1 1			

HDL code: No syntax errors

2. Register (folder a)

NAND2Tetris / Hardware Simulator / Register.hdl

HDL Builtin Project 3 Register Chip Register Eval Reset Clock: 21+

```

11 CHIP Register {
12     IN in[16], load;
13     OUT out[16];
14
15     PARTS:
16         Bit(in=in[0],load=load,out=out[0]);
17         Bit(in=in[1],load=load,out=out[1]);
18         Bit(in=in[2],load=load,out=out[2]);
19         Bit(in=in[3],load=load,out=out[3]);
20         Bit(in=in[4],load=load,out=out[4]);
21         Bit(in=in[5],load=load,out=out[5]);
22         Bit(in=in[6],load=load,out=out[6]);
23         Bit(in=in[7],load=load,out=out[7]);
24         Bit(in=in[8],load=load,out=out[8]);
25         Bit(in=in[9],load=load,out=out[9]);
26         Bit(in=in[10],load=load,out=out[10]);

```

Input pins

- in 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 dec
- load 1

Output pins

- out 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 dec

Test Slow Fast

Test Script	Compare File	Output File	Diff Table
1 time in load out			
2 0+ 0 0 0			
3 1 0 0 0			
4 1+ 0 1 0			
5 2 0 1 0			
6 2+ -32123 0 0			
7 3 -32123 0 0			
8 3+ 11111 0 0			
9 / 11111 0 0			

HDL code: No syntax errors

3. RAM8 (folder a)

The screenshot shows the Nand2Tetris web-based hardware simulator interface. The top navigation bar includes tabs for various projects and components. The main window displays the HDL code for RAM8.hdl, which includes parts like DMux8Way and Register blocks. To the right of the code, there are sections for 'Chip RAM8' showing 'Input pins' (in, load, address) and 'Output pins' (out), and 'Internal pins' (loada, loadb, loadc). Below the code editor is a 'Test' section with a table for a test script, and at the bottom, a message 'HDL code: No syntax errors'.

```
14
15 PARTS:
16 DMux8Way(in = load,sel = address,a= loada, b= loadb, c= loadc , d= loadd, e=
17
18 Register(in= in , load= loada, out=aout );
19 Register(in= in , load= loadb, out=bout );
20 Register(in= in , load= loadc, out=cout );
21 Register(in= in , load= loadd, out=dout );
22 Register(in= in , load= loade, out=eout );
23 Register(in= in , load= loadf, out=fout );
24 Register(in= in , load= loadg, out=gout );
25 Register(in= in , load= loadh, out=hout );
26
27 Mux8Way16(a= aout, b= bout, c= cout, d= dout, e= eout, f= fout, g= gout, h= g
28
```

time	in	load	address	out
0+	0	0	0	0
1	0	0	0	0
1+	0	1	0	0
2	0	1	0	0
2+	11111	0	0	0
3	11111	0	0	0
3+	11111	1	1	0
4	11111	1	1	11111

HDL code: No syntax errors

4. RAM64 (folder a)

NAND2Tetris / Hardware Simulator / RAM64.hdl

HDL Builtin Project 3 RAM64 Chip RAM64 Eval Reset Clock: 16

```

14
15
16 s[3..5],a= loada, b= loadb, c= loadc , d= loadd, e= loade, f= loadf , g= loadg, h=
17
18 ss= address[0..2], out=aout );
19 ss= address[0..2],out=bout );
20 ss= address[0..2],out=cout );
21 ss= address[0..2],out=dout );
22 ss= address[0..2],out=eout );
23 ss= address[0..2],out=fout );
24 ss= address[0..2],out=gout );
25 ss= address[0..2],out=hout );
26
27 cout, d= dout, e= eout, f= fout, g= gout, h= gout, sel= address[3..5], out= out);
28

```

Test Script Compare File Output File Diff Table

4 comparison failures. Scroll down for details

time	in	load	address	out
0+	0	0	0	0
1	0	0	0	0
1+	0	1	0	0
2	0	1	0	0
2+	1313	0	0	0
7	1327	0	0	0

HDL code: No syntax errors

5. RAM512 (folder b)

NAND2Tetris / Hardware Simulator / RAM512.hdl

HDL Builtin Project 3 RAM512 Chip RAM512 Eval Reset Clock: 11

```

14
15
16 6..8],a= loada, b= loadb, c= loadc , d= loadd, e= loade, f= loadf , g= loadg, h=
17
18 ss= address[0..5], out=aout );
19 ss= address[0..5],out=bout );
20 ss= address[0..5],out=cout );
21 ss= address[0..5],out=dout );
22 ss= address[0..5],out=eout );
23 ss= address[0..5],out=fout );
24 ss= address[0..5],out=gout );
25 ss= address[0..5],out=hout );
26
27 ut, d= dout, e= eout, f= fout, g= gout, h= gout, sel= address[6..8], out= out);
28

```

Test Script Compare File Output File Diff Table

6 comparison failures. Scroll down for details

time	in	load	address	out
0+	0	0	0	0
1	0	0	0	0
1+	0	1	0	0
2	0	1	0	0
2+	13099	0	0	0
7	12800	0	0	0

HDL code: No syntax errors

6. RAM4K (folder b)

The screenshot shows the NAND2Tetris web IDE interface. On the left, there is a code editor with the file `RAM4K.hdl`. The code is as follows:

```
14
15
16 = loada, b= loadb, c= loadc , d= loadd, e= loade, f= loadf , g= loadg, h= loadh )
17
18 ress[0..8], out=aout );
19 ress[0..8],out=bout );
20 ress[0..8],out=cout );
21 ress[0..8],out=dout );
22 ress[0..8],out=eout );
23 ress[0..8],out=fout );
24 ress[0..8],out=gout );
25 ress[0..8],out=hout );
26
27 out, e= eout, f= fout, g= gout, h= gout, sel= address[9..11], out= out);
```

On the right, there is a simulation interface for the **Chip RAM4K**. It includes sections for **Input pins**, **Output pins**, and **Internal pins**. The **Input pins** section shows the current values for `in`, `load`, and `address`. The **Output pins** section shows the current value for `out`. The **Internal pins** section shows the current values for `loada`, `loadb`, and `loadc`. There are also buttons for **Eval**, **Reset**, and **Clock: I2**.

Below the simulation interface, there is a **Test** section with a slider from **Slow** to **Fast**. Under the **Test** section, there are tabs for **Test Script**, **Compare File**, **Output File**, and **Diff Table**. A message indicates **4 comparison failures. Scroll down for details**. Below this, there is a table with 7 rows and 6 columns, showing data for time, in, load, address, out, and a column labeled with a question mark.

1	time	in	load	address	out	?
2	0+	0 0 0 0 0				
3	1	0 0 0 0 0				
4	1+	0 1 0 0 0				
5	2	0 1 0 0 0				
6	2+	1111 0 0 0 0				
7	?	1111 0 0 0 0				

HDL code: No syntax errors

7. RAM16K (folder b)

NAND2Tetris / Hardware Simulator / RAM16K.hdl

```

HDL Builtin Project 3 RAM16K
11 CHIP RAM16K {
12     IN in[16], load, address[14];
13     OUT out[16];
14
15     PARTS:
16     DMux4Way(in = load,sel = address[12..13],a= loada, b= loadb, c= loadc , d= loadd);
17
18     RAM4K(in= in , load= loada, address= address[0..11], out=aout );
19     RAM4K(in= in , load= loadb, address= address[0..11],out=bout );
20     RAM4K(in= in , load= loadc, address= address[0..11],out=cout );
21     RAM4K(in= in , load= loadd, address= address[0..11],out=dout );
22
23
24     Mux4Way16(a= aout, b= bout, c= cout, d= dout, sel= address[12..13], out= out)
25 }

```

Chip RAM16K

Input pins

- in: 0 0 1 1 0 0 0 0 0 0 1 1 1 0 0 1 (dec)
- load: 1
- address: 1 1 0 0 0 0 0 0 1 1 1 0 0 1 (dec)

Output pins

- out: 0 0 1 1 0 0 0 0 0 0 1 1 1 0 0 1 (dec)

Internal pins

- loada: 0
- loadb: 0
- loadc: 0

Test

Test Script	Compare File	Output File	Diff Table
1 time in load address out			
2 0+ 0 0 0 0			
3 1 0 0 0 0			
4 1+ 0 1 0 0			
5 2 0 1 0 0			
6 2+ 4321 0 0 0			
7 3 4321 0 0 0			
8 3+ 4321 1 4321 0			

HDL code: No syntax errors

8. PC (folder a)

NAND2Tetris / Hardware Simulator / PC.hdl

```

HDL Builtin Project 3 PC
8 * else if load(t): out(t+1) = in(t)
9 * else if inc(t): out(t+1) = out(t) + 1
10 * else: out(t+1) = out(t)
11 */
12 CHIP PC {
13     IN in[16], reset, load, inc;
14     OUT out[16];
15
16     PARTS:
17     Inc16(in=out1, out=out2);
18     Mux16(a=out1, b=out2, sel=inc, out=out3);
19     Mux16(a=out3, b=in, sel=load, out=out4);
20     Mux16(a=out4, b=false, sel=reset, out=out5);
21     Register(in=out5, load=true, out=out1, out=out);
22 }

```

Chip PC

Input pins

- in: 0 0 1 1 0 0 0 0 0 0 1 1 1 0 0 1 (dec)
- reset: 0
- load: 1
- inc: 1

Output pins

- out: 0 0 1 1 0 0 0 0 0 0 1 1 1 0 0 1 (dec)

Internal pins

- out1: 0 0 1 1 0 0 0 0 0 0 1 1 1 0 0 1 (dec)
- out2: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 (dec)

Test

Test Script	Compare File	Output File	Diff Table
1 time in reset load inc out			
2 0+ 0 0 0 0 0 0			
3 1 0 0 0 0 0 0			
4 1+ 0 0 0 0 1 0			
5 2 0 0 0 0 1 1			
6 2+ -32123 0 0 1 1			
7 3 -32123 0 0 1 2			
8 3+ -32123 0 1 1 2			

HDL code: No syntax errors