

Raul Aguilar

Professor Paulding

CS 220 2148

October 23, 2020

Homework 6 Computer Architecture

Memory

Hardware Simulator (2.5) - C:\Users\raul\Dropbox\College\CS 220 2148\Week9\05\Memory\Memory.hdl

File View Run Help

Chip Name: **Memory (Clocked)** Time: **8**

Input pins		Output pins	
Name	Value	Name	Value
in[16]	-1	out[16]	0
load	0		
address[15]	24576		

HDL

```
CHIP Memory {
    IN in[16], load, address[15]
    OUT out[16];

    PARTS:
        // Put your code here:
        DMux4Way(in=load, sel=address[15],
            Or(a=rama, b=ramb, out=ram);
        RAM16K(in=in, address=address[0:15],
            Screen(in=in, address=address[0:15],
            Keyboard(out=kout);
        Mux4Way16(a=rout, b=rout, c=sout,
            out=out[16]);
}
```

Internal pins

Name	Value
rama	0
ramb	0
screen	0
key	0
ram	0
rout[16]	2222
sout[16]	0
kout[16]	0

RAM 16K:

8189	0
8190	0
8191	0
8192	2222
8193	0
8194	0
8195	0

End of script - Comparison ended successfully

CPU

Hardware Simulator (2.5) - C:\Users\raula\Dropbox\College\CS 220 2148\Week9\05\CPU\CPU.hdl

File View Run Help

Chip Name: CPU (Clocked) Time: 46

Input pins		Output pins	
Name	Value	Name	Value
inM[16]	11111	outM[16]	1
instruction[16]	32767	writeM	0
reset	0	addressM[15]	32767
		pc[15]	1

HDL		Internal pins	
Name	Value	Name	Value
nin	1	aluout[16]	1
inA[16]	32767	regA	1
aout[16]	32767	AM	0
AMout[16]	32767	D[16]	1
zr	0	ng	0
regD	0	jgt	0
jgt	0	jeq	0
jeq	0	...	0

```

* CPU jumps to address 0 (i.e. 0)
* than to the address resulting
*/
CHIP CPU {
    IN inM[16], // M v
    instruction[16], // Inst
    reset; // Sigr
    // proq
    // the

    OUT outM[16], // M v
    writeM, // Writ

```

```

set instruction %B1110001100000110, // D:JLE
tick, output, tock, output;

set instruction %B1110001100000111, // D:JMP
tick, output, tock, output;

set instruction %B1110111111010000, // D=1
tick, output, tock, output;

set instruction %B1110001100000001, // D:JGT
tick, output, tock, output;

set instruction %B1110001100000010, // D:JEQ
tick, output, tock, output;

set instruction %B1110001100000011, // D:JGE
tick, output, tock, output;

set instruction %B1110001100000100, // D:JLT
tick, output, tock, output;

set instruction %B1110001100000101, // D:JNE
tick, output, tock, output;

set instruction %B1110001100000110, // D:JLE
tick, output, tock, output;

set instruction %B1110001100000111, // D:JMP
tick, output, tock, output;

set reset 1;
tick, output, tock, output;

set instruction %B0111111111111111, // @32767
set reset 0;
tick, output, tock, output;

```

End of script - Comparison ended successfully

CPU External

Hardware Simulator (2.5) - C:\Users\raula\Dropbox\College\CS 220 2148\Week9\05\CPU\CPU.hdl

File View Run Help

Chip Name: CPU (Clocked) Time: 46

Input pins		Output pins	
Name	Value	Name	Value
inM[16]	11111	outM[16]	1
instruction[16]	32767	writeM	0
reset	0	addressM[15]	32767
		pc[15]	1

HDL		Internal pins	
Name	Value	Name	Value
nin	1	aluout[16]	1
inA[16]	32767	regA	1
regA	1	acut[16]	32767
AM	0	AMout[16]	32767
D[16]	1	zr	0
nr	0	ng	0
regD	0	jgt	0
jgt	0	jeq	0
jeq	0	alt	0

```
* CPU jumps to address 0 (i.e. ^
* than to the address resulting
*/
CHIP CPU {
    IN inM[16],          // M v
    instruction[16],     // Inst
    reset;              // Sigr
                      // prog
                      // the

    OUT outM[16],        // M v
    writeM,             // Writ

    set instruction %B1110001100000110, // D:JLE
    tick, output, tock, output;

    set instruction %B1110001100000111, // D:JMP
    tick, output, tock, output;

    set instruction %B1110111111010000, // D=1
    tick, output, tock, output;

    set instruction %B1110001100000001, // D:JGT
    tick, output, tock, output;

    set instruction %B1110001100000010, // D:JEQ
    tick, output, tock, output;

    set instruction %B1110001100000011, // D:JGE
    tick, output, tock, output;

    set instruction %B1110001100000100, // D:JLT
    tick, output, tock, output;

    set instruction %B1110001100000101, // D:JNE
    tick, output, tock, output;

    set instruction %B1110001100000110, // D:JLE
    tick, output, tock, output;

    set instruction %B1110001100000111, // D:JMP
    tick, output, tock, output;

    set reset 1;
    tick, output, tock, output;

    set instruction %B0111111111111111, // @32767
    set reset 0;
    tick, output, tock, output;
}
```

End of script - Comparison ended successfully

Computer Add

Hardware Simulator (2.5) - C:\Users\raula\Dropbox\College\CS 220 2148\Week9\05\Computer\Computer.hdl

File View Run Help

Chip Name: **Computer (Clocked)** Time: **13**

Input pins		Output pins	
Name	Value	Name	Value
reset	0		

Internal pins	
Name	Value
pc[15]	6
instruction[16]	0
data[16]	5
load	0
value[16]	0
address[15]	0

HDL

```
* screen may show some output : ^
* with the computer via the key
*/
CHIP Computer {
    IN reset;

    PARTS:
        // Put your code here:
        ROM32K(address=pc, out=instruction)
        CPU(inM=data, instruction=instruction,
            Memory(address=address, in=value))
}
```

Script

```
// This file is part of www.nand2tetrtris.org
// and the book "The Elements of Computing Systems"
// by Nisan and Schocken, MIT Press.
// File name: projects/05/ComputerAdd.tst

load Computer.hdl,
output-file ComputerAdd.out,
compare-to ComputerAdd.cmp,
output-list time%$1.4.1 reset%B2.1.2 ARegister[0]%$1.7.1 DRegister[0]%$1.7.1

// Load a program written in the Hack machine language.
// The program adds the two constants 2 and 3 and writes the result in RAM.
ROM32K load Add.hack,
output;

// First run (at the beginning PC=0)
repeat 6 {
    tick, tock, output;
}

// Reset the PC
set reset 1,
set RAM16K[0] 0,
tick, tock, output;

// Second run, to check that the PC was reset correctly.
set reset 0,

repeat 6 {
    tick, tock, output;
}
```

End of script - Comparison ended successfully

Computer Max

Hardware Simulator (2.5) - C:\Users\raula\Dropbox\College\CS 220 2148\Week9\05\Computer\Computer.hdl

File View Run Help

Chip Name: Computer (Clocked) Time: 25

Input pins		Output pins	
Name	Value	Name	Value
reset	0		

Internal pins	
Name	Value
pc[15]	14
instruction[16]	14
data[16]	23456
load	0
value[16]	0
address[15]	2

HDL

```
* screen may show some output : ^
* with the computer via the key
*/
CHIP Computer {
    IN reset;

    PARTS:
        // Put your code here:
        ROM32K(address=pc, out=instruction);
        CPU(inM=data, instruction=instruction);
        Memory(address=address, in=value);
}
```

Internal pins

```
// File name: projects/05/ComputerMax.tst

load Computer.hdl,
output-file ComputerMax.out,
compare-to ComputerMax.cmp,
output-list time%$1.4.1 reset%B2.1.2 ARegister[%D1.7.1 DRegister[%D1.

// Load a program written in the Hack machine language.
// The program computes the maximum of RAM[0] and RAM[1]
// and writes the result in RAM[2].

ROM32K load Max.hack,

// first run: compute max(3,5)
set RAM16K[0] 3,
set RAM16K[1] 5,
output;

repeat 14 {
    tick, tock, output;
}

// reset the PC
set reset 1,
tick, tock, output;

// second run: compute max(23456,12345)
set reset 0,
set RAM16K[0] 23456,
set RAM16K[1] 12345,
output;

// The run on these inputs needs less cycles (different branching)
repeat 10 {
    tick, tock, output;
}
```

End of script - Comparison ended successfully

Computer Rect

Hardware Simulator (2.5) - C:\Users\raula\Dropbox\College\CS 220 2148\Week9\05\Computer\Computer.hdl

File View Run Help

Chip Name: Computer (Clocked) Time: 63

Input pins

Name	Value
reset	0

Output pins

Name	Value
------	-------

HDL

```
* screen may show some output : ^
* with the computer via the key
*/
CHIP Computer {
    IN reset;

    PARTS:
        // Put your code here:
        ROM32K(address=pc, out=instruction);
        CPU(inM=data, instruction=instruction);
        Memory(address=address, in=value);
}
```

Internal pins

Name	Value
pc[15]	24
instruction[16]	-5497
data[16]	0
load	0
value[16]	0
address[15]	23

RAM 16K:

Address	Value
20	0
21	0
22	0
23	0
24	0
25	0
26	0

ROM:

Address	Value
18	1110001100001000
19	0000000000001000
20	1111110010011000
21	0000000000001010
22	1110001100000001
23	0000000000001011
24	1110101010000111

A: 23 D: 0 PC: 24

ALU

D Input: 0

M/A Input: 23

ALU output: 0

End of script - Comparison ended successfully