



Introduction to the Computer Simulator SimpSim

Chun-Jen Tsai
National Chiao Tung University
3.12.2012

Computer Simulator

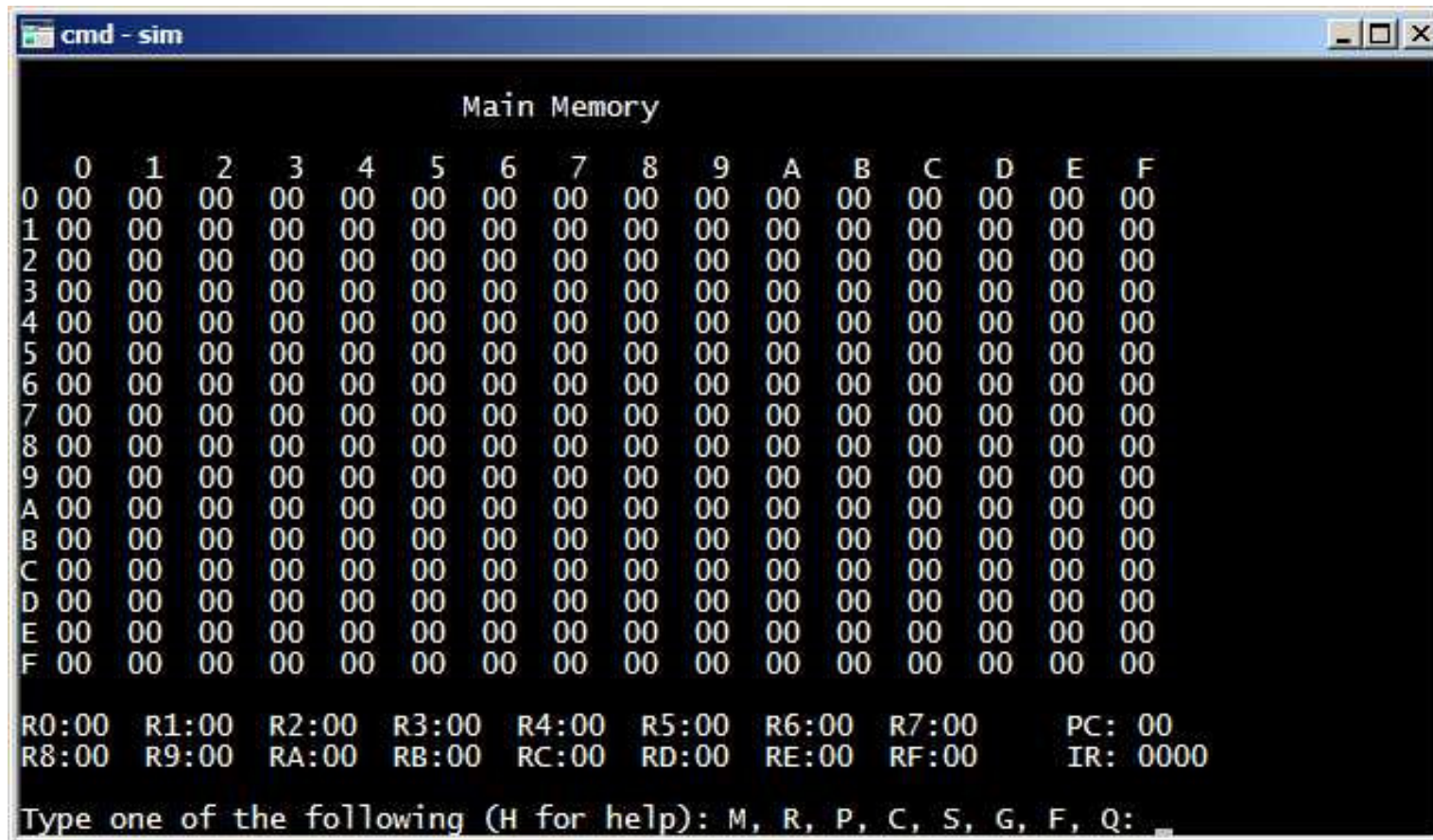
- ❑ A computer simulator mimics the way a computer runs machine language programs
- ❑ A computer simulator must display the status of the system, including
 - The special-purposes registers (e.g. program counter, instruction register)
 - The data registers
 - The main memory cells
 - Simple I/O devices

Simulators of a Simple Computer

- ❑ Our textbook describes a simple computer and the author provides a simple command-line simulator program, **sim.c**, that simulates the operation of this computer
 - You can compile the source code using any C compiler (like Visual C++ or gcc)
- ❑ A more user-friendly simulator, **SimpSim**, of the same machine can be obtained from the University of Twente (<http://wwwes.cs.utwente.nl/software/simpsim/>)

You can download the source code of `sim.c` from the class website.

Screenshot of the sim.c Simulator



The screenshot shows a terminal window titled "cmd - sim". The main content is a memory dump labeled "Main Memory". The dump consists of 16 rows, each representing a memory location from 0 to F. Each row contains 16 hexadecimal values, all of which are "00". Below the memory dump, the status of registers is shown: R0:00, R1:00, R2:00, R3:00, R4:00, R5:00, R6:00, R7:00, R8:00, R9:00, RA:00, RB:00, RC:00, RD:00, RE:00, RF:00, PC: 00, and IR: 0000. At the bottom, a prompt asks the user to "Type one of the following (H for help): M, R, P, C, S, G, F, Q:" followed by a cursor.

```
cmd - sim
Main Memory
 0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 1 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 2 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 3 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 4 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 5 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 6 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 7 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 9 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 B 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

R0:00 R1:00 R2:00 R3:00 R4:00 R5:00 R6:00 R7:00 PC: 00
R8:00 R9:00 RA:00 RB:00 RC:00 RD:00 RE:00 RF:00 IR: 0000

Type one of the following (H for help): M, R, P, C, S, G, F, Q: _
```

Screenshot of SimpSim

The screenshot displays the Simple Simulator interface. The window title is "Simple Simulator" and it has a menu bar with "File", "Edit", "Run", and "Help".

Main Memory: A table with 16 columns labeled .0 through .F and 16 rows labeled 00 through F0. All cells contain "00".

Registers: A list of registers R0 through RF, each with a value of "00". To the right, there are input fields for "PC" (value "00") and "IR" (value "0000"). Below these are buttons for "Open...", "Save", "Run", "Step", "Break", and "Clear...".

Assembly Window: A text area at the bottom left showing assembly instructions. The visible lines are:

```
00: 00,00 invalid
02: 00,00 invalid
04: 00,00 invalid
06: 00,00 invalid
08: 00,00 invalid
0A: 00,00 invalid
```

At the bottom of the window, there is an "Address:" field with the value "Ready" and two buttons labeled "Asm" and "Disasm".

Assembly Language Program

- ❑ We have learned about the machine code program in Chapter 2, however, it is not easy to read and write programs in assembly directly
- ❑ A more human readable form of the machine code program is called the ***assembly language program***

Machine Code Program

```
156C
166D
5056
306E
C000
```

Assembly Program

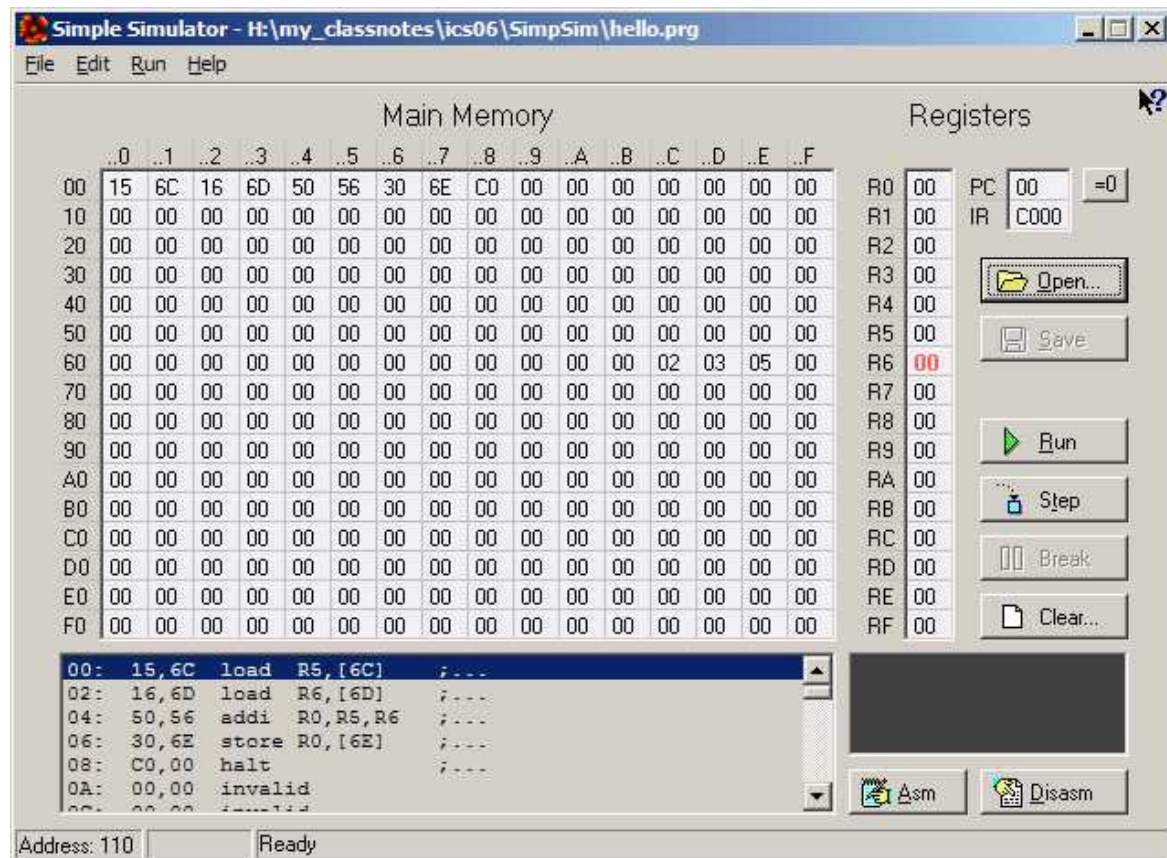
```
load R5, [$6C]
load R6, [$6D]
addi R0, R5, R6
Store [$6E], R0
halt
```

A number with a '\$' prefix means the number is in radix 16

assemble

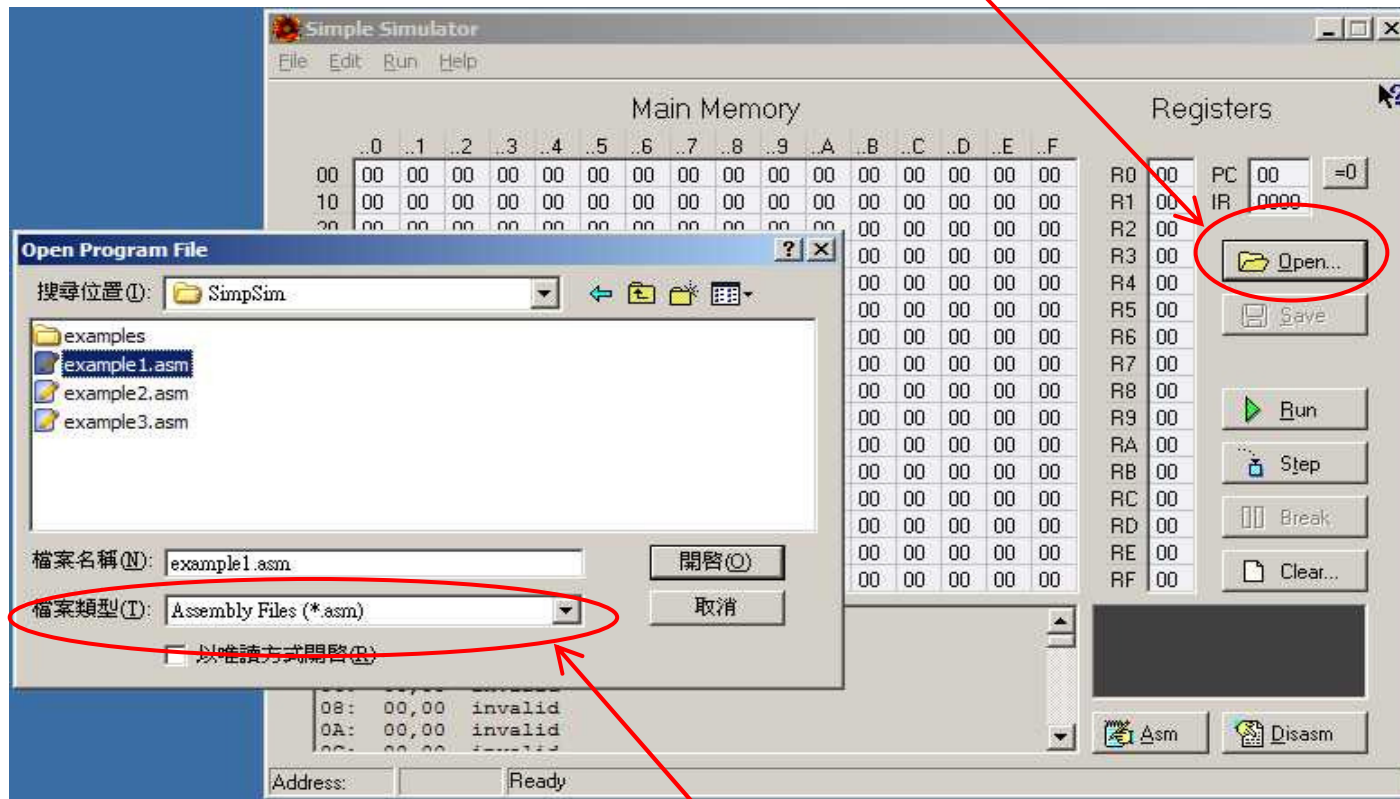
Initialization of Simulated Machine

- ❑ The machine can be initialized by typing machine codes and data into main memory and registers
- ❑ However, this is not very convenient!
- ❑ A faster way is to load a program to initialize the machine



Loading an Assembly Program

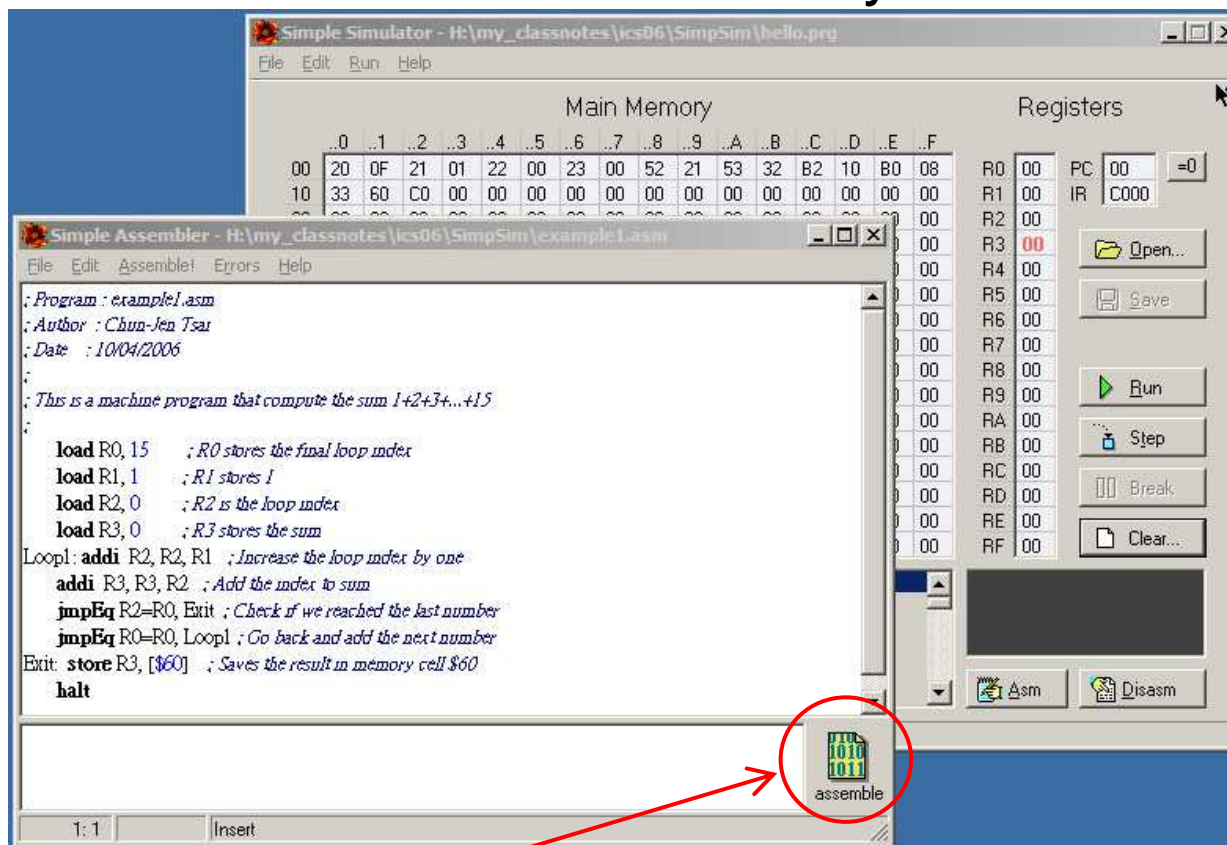
- ❑ Use the “Open” button to load an assembly program into the SimpSim simulator:



Select the “Assembly Files” type

Assembling the Program

- ❑ After loading of an assembly program, you must assemble it into the main memory



Press the “assemble” button to generate machine codes

Executing the Program

- Two different execution modes: Run and Step:

The screenshot shows the Simple Simulator interface. The title bar reads "Simple Simulator - H:\my_classnotes\ics06\SimpSim\hello.prg". The menu bar includes "File", "Edit", "Run", and "Help".

Main Memory: A 16x16 grid showing memory addresses from 00 to F0. The value at address 60 is 78, and the value at address 14 is 00. The value at address 14 is highlighted with a red circle.

Registers: A list of registers R0 through RF. R0 is 0F, R1 is 01, R2 is 0F, R3 is 78, R4 is 00, R5 is 00, R6 is 00, R7 is 00, R8 is 00, R9 is 00, RA is 00, RB is 00, RC is 00, RD is 00, RE is 00, and RF is 00. The PC register is 14 and the IR register is C000. The "Run" button is circled in red.

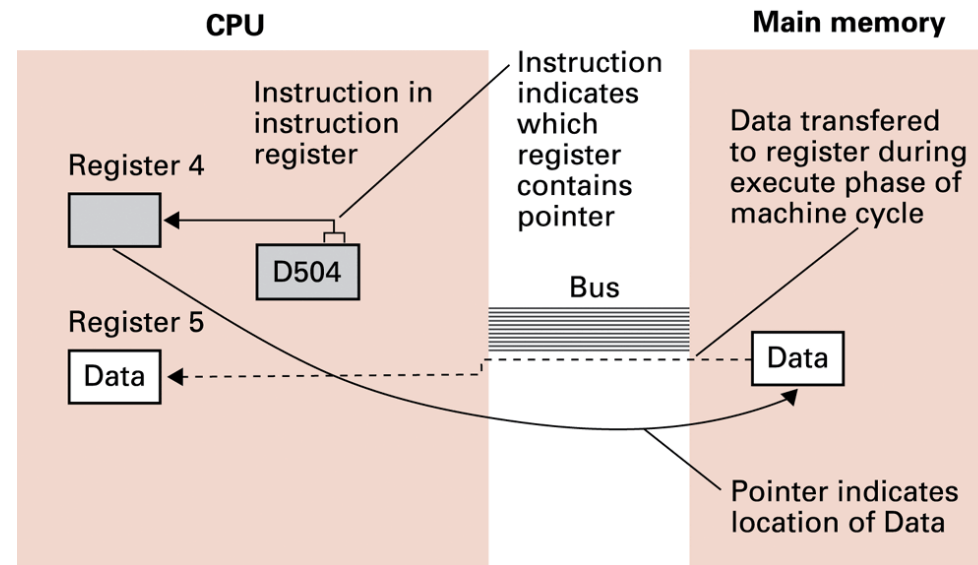
Code Window: A list of instructions with their addresses and values. The instruction at address 14 is highlighted in blue.

Address	Value	Instruction
0E	B0,08	jmpEQ R0=R0,08
10	33,60	store R3,[60]
12	C0,00	halt
14	00,00	invalid
16	00,00	invalid
18	00,00	invalid

Address: 110 | Halted

Notes on OP-Code D and E

- ❑ Op-Code 'D' loads a register with data from a memory cell whose address is stored in another register)



- ❑ Similarly, Op-Code 'E' stores a register to a memory cell via indirect addressing

Reading Assignments

- ❑ In order to be familiar with the machine language of the simple computer, you should read the following materials by yourself:
 - SimSim_Tutorial.pdf
 - Chapter 2
 - Appendix C
 - The on-line Help menu of the SimpSim program
 - Section 8.7 of Chapter 8