

# LAB 1: CÁCH MÁY TÍNH XỬ LÝ MỘT CHƯƠNG TRÌNH

## I. LÀM QUEN VỚI CPU SIM

### 1. CPU Sim là gì?

CPU Sim là gói phần mềm được viết bằng Java. Nó cung cấp cho sinh viên một môi trường học tập tích cực, trong đó sinh viên có thể thiết kế, sửa đổi và so sánh các kiến trúc máy tính khác nhau ở mức thanh ghi và cao hơn. Sinh viên cũng có thể tạo một Machine với các thông số của CPU và Main memory, viết một chương trình (text) theo các lệnh của hợp ngữ (Assembly) (đã được định nghĩa cho Machine) và chạy mô phỏng một chương trình trên máy đã tạo.

Tham khảo thêm về CPUSim:

<https://drive.google.com/open?id=1uR4eAkRgbdC7QLAsuI-bVQ00MT5x3d5X>

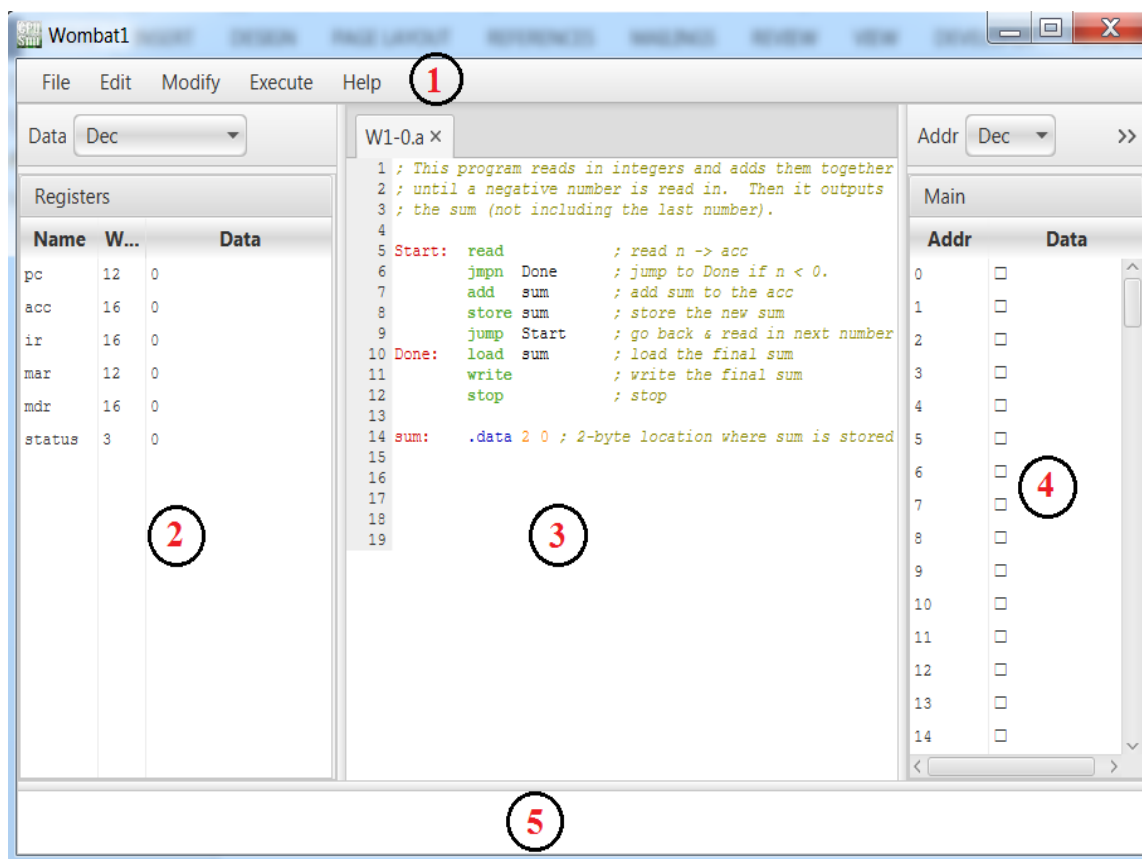
### 2. Cài đặt CPU Sim

Link tải CPU Sim:

<https://drive.google.com/open?id=1jk8p9LDzBygZcgDnRZjhrYL7V41PiK8g>

### 3. Giao diện CPU Sim

Chạy CPUSim (Cpusim4.bat)



- Vùng 1 là thanh thực đơn (menu).

- Các vùng 2, 3, 4, 5 là các cửa sổ.

### 3.1. Menu

#### File

- Các thao tác với file chương trình/mã nguồn (text)
- Các thao tác với file Máy tính (machine)

New text	Ctrl+N
Open text...	Ctrl+O
Reopen text	▶
Close text	Ctrl+W
Save text	Ctrl+S
Save text as...	Ctrl+Shift+S
Open RAM	▶
Save RAM	▶
New machine	Ctrl+Shift+N
Open machine...	Ctrl+Shift+O
Reopen machine	▶
Save machine	Ctrl+B
Save machine as...	Ctrl+Shift+B
Save machine in HTML...	Ctrl+Alt+B
Page setup...	
Print...	Ctrl+P
Quit	Ctrl+Q

#### Edit

- Thao tác với chương trình/mã nguồn (text).

Undo	Ctrl+Z
Redo	Ctrl+Shift+Z
Cut	Ctrl+X
Copy	Ctrl+C
Paste	Ctrl+V
Select All	Ctrl+A
Toggle Comment	Ctrl+/
Find...	Ctrl+F
Preferences...	Ctrl+,

## Modify

- Chỉnh sửa các thông số của CPU, RAM

Machine instructions...	Ctrl+M
Microinstructions...	Ctrl+Shift+M
Hardware Modules...	Ctrl+K
EQUs...	Ctrl+E
Fetch Sequence...	Ctrl+T

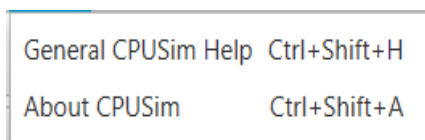
## Execute

Debug Mode	Ctrl+D
Assemble	Ctrl+1
Assemble & load	Ctrl+2
Assemble, load & run	Ctrl+3
Clear, assemble, load & run	Ctrl+G
Run	Ctrl+R
Stop	Ctrl+.
Reset everything	Ctrl+Shift+R
Clear console	Ctrl+L
Options...	Ctrl+I

- Hợp dịch (Assemble)
- Nạp chương trình vào bộ nhớ (Load)
- Chạy chương trình
  - o Run
  - o Debug Mode
    - Chạy từng lệnh (Step by Instr)
    - Chạy từng vi lệnh (Step by Micro)

## Help

- Trợ giúp sử dụng

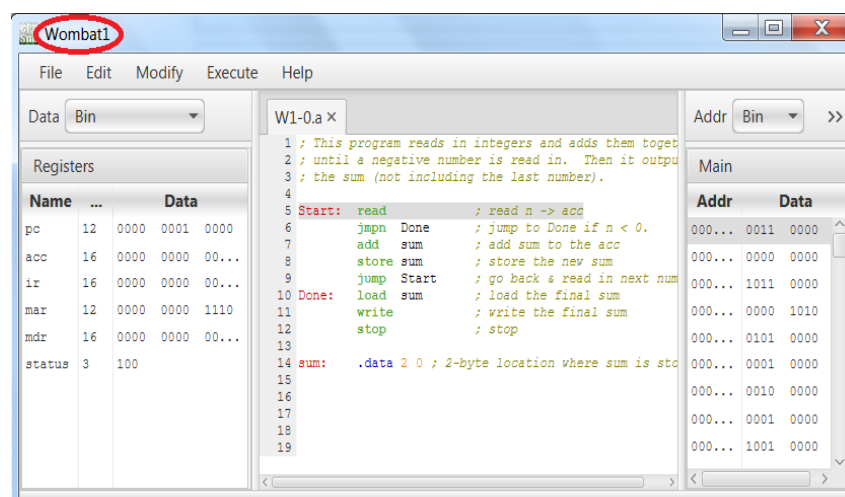
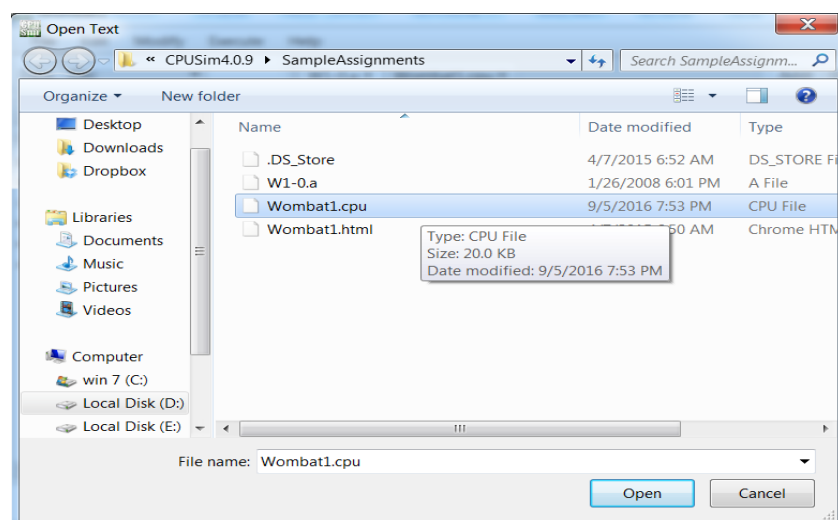


### 3.2. Các cửa sổ

- Vùng 2 (Register) chứa trạng thái các thanh ghi đại diện trong CPU.
- Vùng 3 là vùng soạn thảo viết bằng ngôn ngữ assembly (\*.a).
- Vùng 4 hiển thị trạng thái bộ nhớ chính (Main Memory).
- Vùng 5 là cửa sổ hiển thị (Console).

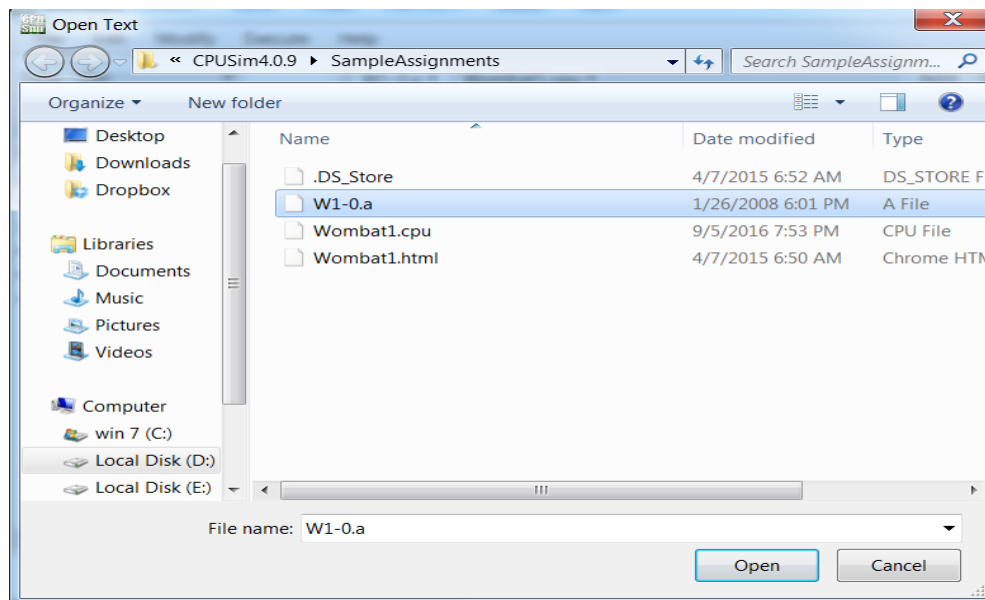
### 4. Đặc điểm của CPU Sim

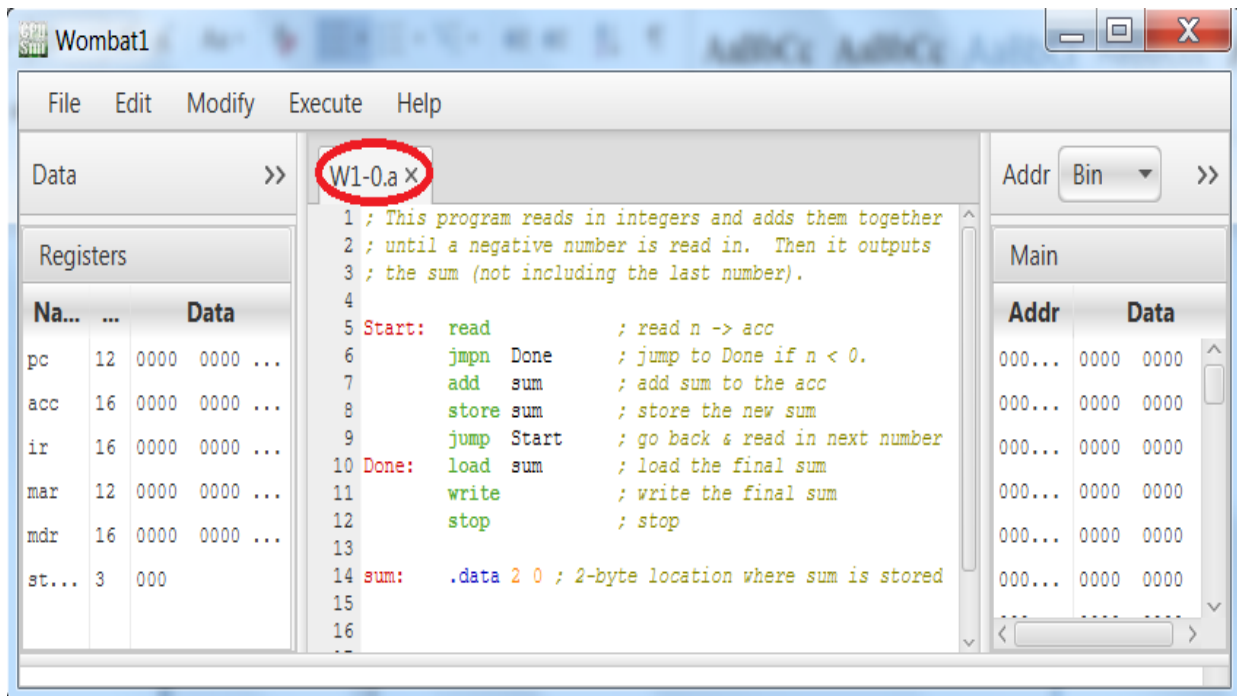
- Phần mềm cho phép tạo một Machine với các thông số của CPU và Main memory



- File cấu hình của Machine được lưu dạng \*.cpu

- File mặc định được nạp luôn khi khởi động CPUSim là **Wombat1.cpu** (xem phần title (góc trên, trái) của cửa sổ chính)
- Các thông số của CPU gồm:
  - o Các thanh ghi: Name, Width
  - o Các lệnh (Instruction)
  - o Các vi lệnh (Microinstruction)
  - o Cấu trúc một lệnh: mặc định dài 16 bit (4 bit opcode, 12 bit address)
- Các thông số của Main memory gồm:
  - o Tên: Mặc định là Main
  - o Dung lượng (Size): Mặc định là 128 từ nhớ (word)
  - o Độ rộng một từ nhớ: Mặc định là 1 Byte (8 bit)
- **Phần mềm cho phép tạo một Chương trình (Text) theo các lệnh của Assembly (đã được định nghĩa cho Machine).**





- File Text mặc định được lưu dạng \*.a (Assembly)
- Chương trình mặc định được nạp luôn khi khởi động CPUSim là **W1-0.a**. Chương trình này có nhiệm vụ: Yêu cầu nhập vào các số nguyên dương và cộng dồn đến khi nào gặp một số âm thì kết thúc, sau đó in ra kết quả (Không cộng dồn số âm cuối vào).

## II. CÁCH CPU XỬ LÝ MỘT CHƯƠNG TRÌNH

### 1. Các thông số mặc định của CPU và Main memory

- CPU gồm các thanh ghi với số bit tương ứng.

Registers		
Name	Width	Data
pc	12	0
acc	16	0
ir	16	0
mar	12	0
mdr	16	0
status	3	0

- Mặc định Độ dài một từ nhớ Word cho data và instruction là 16 bit. Do đó các thanh ghi acc, ir, mdr dài 16 bit.
  - Một Word lệnh gồm 4 bit opcode và 12 bit address. Do đó, các thanh ghi pc và mar dài 12 bit.
  - Status là thanh ghi trạng thái của CPU.
- Main memory gồm các Byte nhớ với địa chỉ tương ứng.

Main			
Addr		Data	
000	0000	0000	0000
000	0001	0000	0000
000	0010	0000	0000
000	0011	0000	0000
000	0100	0000	0000
000	0101	0000	0000
000	0110	0000	0000
000	0111	0000	0000
000	1000	0000	0000
000	1001	0000	0000
000	1010	0000	0000
000	1011	0000	0000

- Mặc định bộ nhớ dài 128 Byte.
- Từng Byte được đánh địa chỉ riêng.
- Do đó 1 Word (mặc định 16 bit) sẽ chiếm 2 Byte.

## 2. Ý nghĩa của chương trình (W1-0.a)

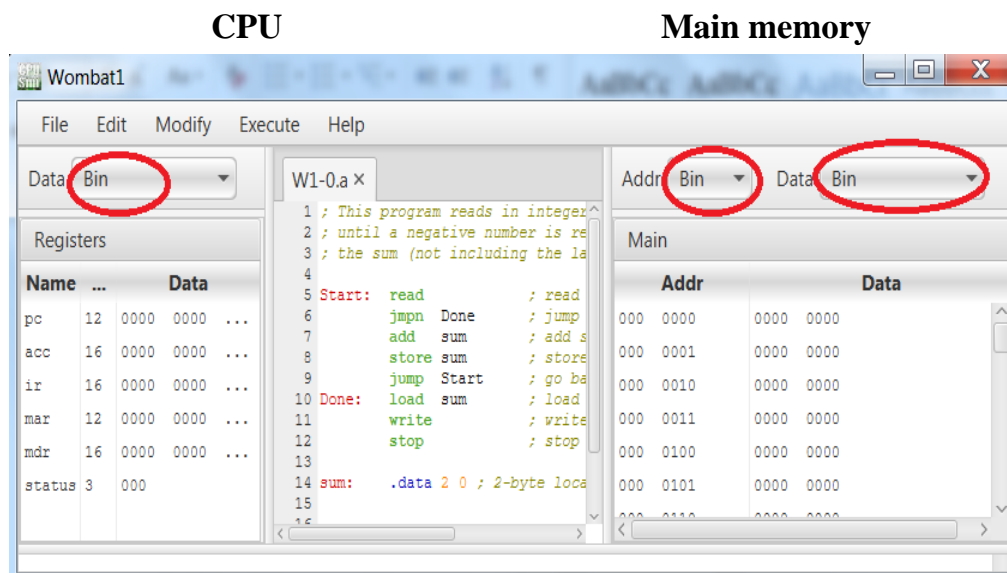
Yêu cầu nhập vào các số nguyên dương và cộng dồn đến khi nào gặp một số âm thì kết thúc. Rồi in ra kết quả (Không cộng dồn số âm cuối vào).

```
Start: read          ; đọc một số nguyên n -> acc
      jmpn Done       ; Nhảy đến Done nếu n < 0 (Kết thúc chương trình và in ra kết quả)
      add sum         ; cộng sum vào acc và lưu vào acc
      store sum        ; Lưu giá trị mới của sum (vào sum) từ acc
      jump Start      ; Quay trở lại và đọc vào số nguyên tiếp theo
Done:  load sum        ; Nạp giá trị cuối cùng của sum vào CPU
      write          ; In giá trị của sum ra màn hình
      stop           ; Kết thúc

sum:   .data 2 0 ; 2-byte location where sum is stored
```

## 3. Chuẩn bị trước khi chạy chương trình

- Thiết lập hiển thị dữ liệu theo dạng nhị phân Bin (Binary) ở cả CPU và Main memory.



- Chọn Execute/ Reset everything để đưa Machine về trạng thái ban đầu.

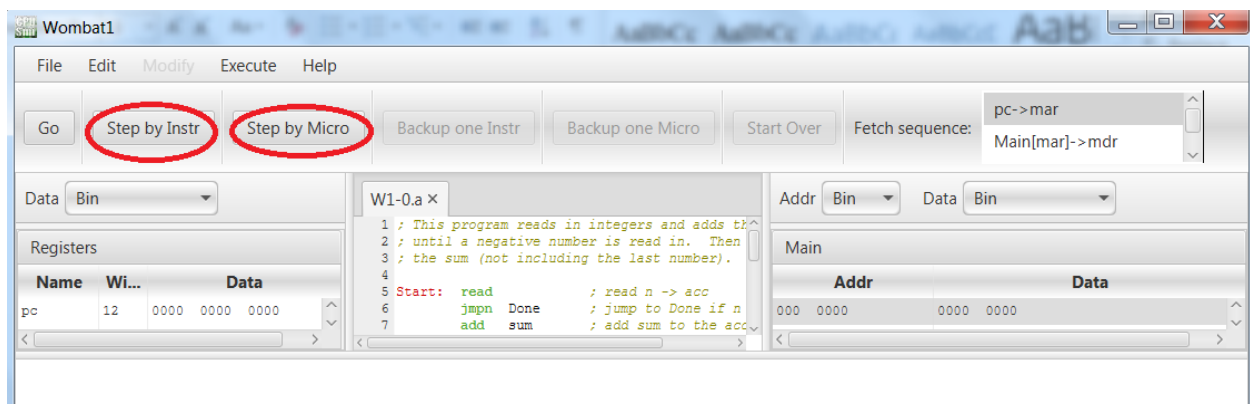
Execute		Help
Debug Mode		Ctrl+D
Assemble		Ctrl+1
Assemble & load		Ctrl+2
Assemble, load & run		Ctrl+3
Clear, assemble, load & run		Ctrl+G
Run		Ctrl+R
Stop		Ctrl+.
Reset everything		Ctrl+Shift+R
Clear console		Ctrl+L
Options...		Ctrl+I



- Chọn Debug Mode để thực hiện từng lệnh/ vi lệnh.

Execute	Help
Debug Mode	Ctrl+D
Assemble	Ctrl+1
Assemble & load	Ctrl+2
Assemble, load & run	Ctrl+3
Clear, assemble, load & run	Ctrl+G
Run	Ctrl+R
Stop	Ctrl+.
Reset everything	Ctrl+Shift+R
Clear console	Ctrl+L
Options...	Ctrl+I

Xuất hiện



- Hợp dịch và nạp chương trình vào Main memory.

Execute	Help
✓ Debug Mode	Ctrl+D
Assemble	Ctrl+1
Assemble & load	Ctrl+2
Assemble, load & run	Ctrl+3
Clear, assemble, load & run	Ctrl+G
Run	Ctrl+R
Stop	Ctrl+.
Reset everything	Ctrl+Shift+R
Clear console	Ctrl+L
Options...	Ctrl+I

- Quan sát sự thay đổi phần Data trong Main memory.

**Trước khi nạp**

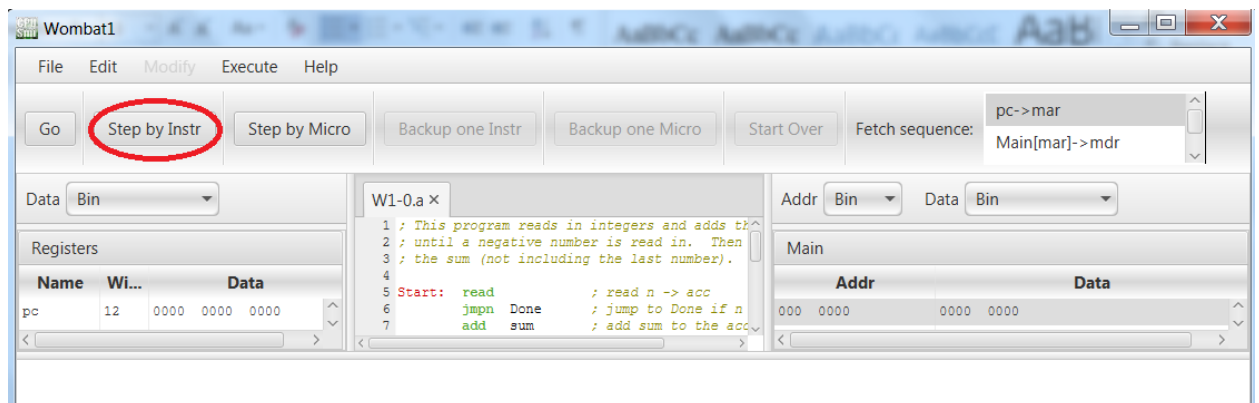
Main			
Addr		Data	
000	0000	0000	0000
000	0001	0000	0000
000	0010	0000	0000
000	0011	0000	0000
000	0100	0000	0000
000	0101	0000	0000
000	0110	0000	0000
000	0111	0000	0000
000	1000	0000	0000
000	1001	0000	0000
000	1010	0000	0000
000	1011	0000	0000

**Chương trình đã được nạp**

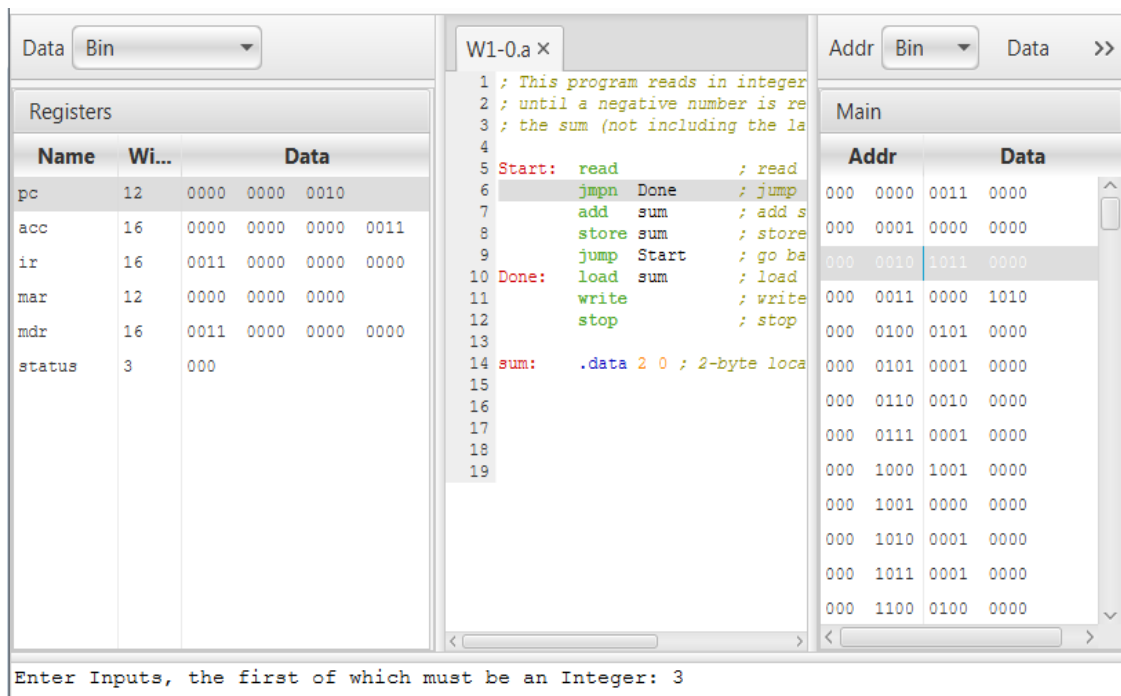
Main			
Addr		Data	
000	0010	1011	0000
000	0011	0000	1010
000	0100	0101	0000
000	0101	0001	0000
000	0110	0010	0000
000	0111	0001	0000
000	1000	1001	0000
000	1001	0000	0000
000	1010	0001	0000
000	1011	0001	0000

#### 4. Chạy chương trình

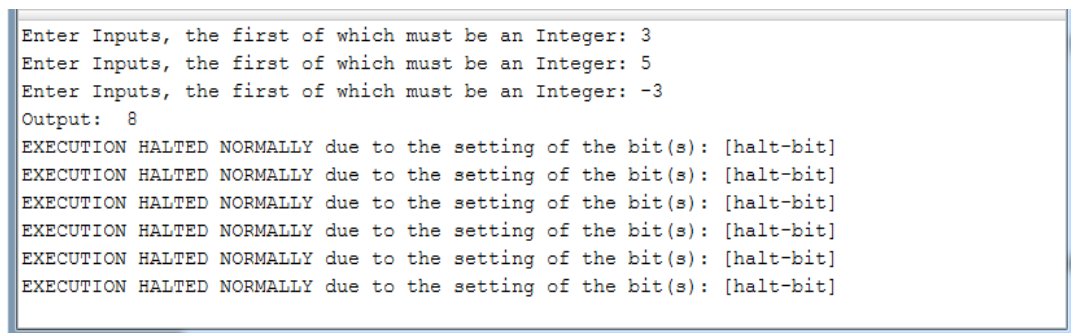
- Chạy theo từng lệnh



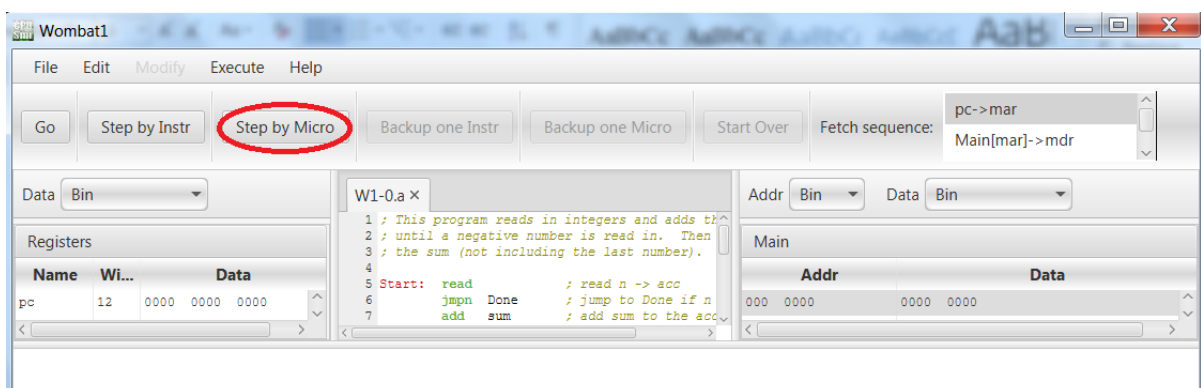
- Click vào Step by Instr,
- Quan sát trình tự thực hiện lệnh ở cửa sổ giữa và cửa sổ dưới,
- Quan sát giá trị thay đổi trong các thanh ghi của CPU và Data trong Main memory,



- Lặp lại cho đến khi nhập vào một số âm (ví dụ: -1) để kết thúc chương trình và in ra tổng.



- Chạy theo từng vi lệnh



- Bỏ chọn Debug Mode,
- Chọn Reset everything để đưa Machine về trạng thái ban đầu,
- Chọn lại Debug Mode,

- Chọn Assemble & Load để Hợp dịch và nạp lại chương trình vào bộ nhớ,
- Click vào Step by Micro để thực hiện từng vi lệnh,
- Quan sát từng lệnh được thực hiện ở cửa sổ giữa (Chương trình) và Các vi lệnh được thực hiện trong từng lệnh ở Fetch Sequence.