

CS231 ระบบคอมพิวเตอร์และภาษาแอสเซมบลี

บทที่ 2: การกำหนดค่าและสำเนาค่า



ผู้ช่วยศาสตราจารย์ ดร. ปวีณ เชื้อนแก้ว

สาขาวิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์ มหาวิทยาลัยแม่โจ้



CS231: บทที่ 2 : การกำหนดค่าและสำเนา

Mnemonic : MOV

ความหมาย: Move data

การใช้งาน: MOV ปลายทาง, ต้นทาง

รูปแบบที่สามารถใช้ได้

MOV REG, REG

MOV REG, IMMEDIATE

MOV REG, MEMORY

MOV MEMORY, IMMEDIATE

MOV MEMORY, REG

รูปแบบที่สามารถใช้ร่วมกับ SREG

MOV SREG, MEMORY

MOV SREG, REG

MOV REG, SREG

MOV MEMORY, SREG

* ใช้กำหนดค่า CS และ IP ไม่ได้

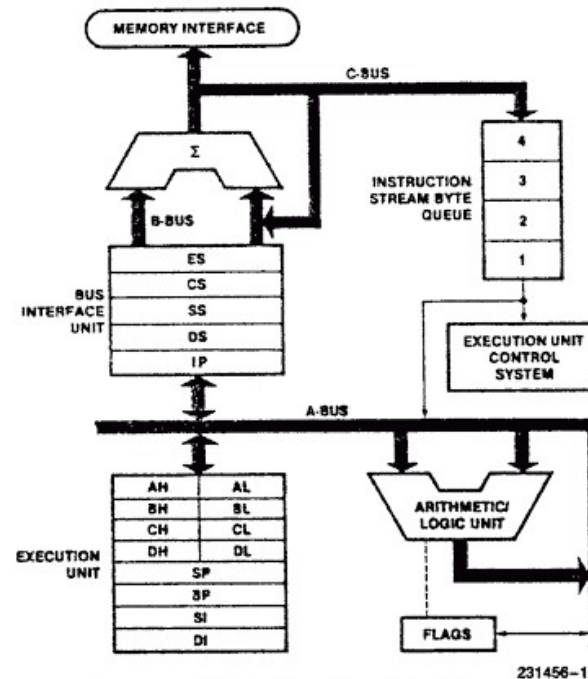


Figure 1. 8088 CPU Functional Block Diagram

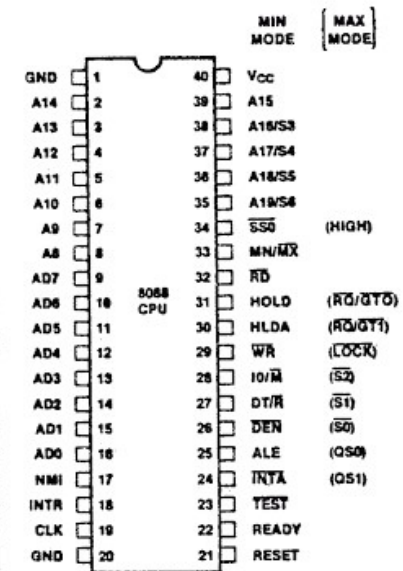


Figure 2. 8088 Pin Configuration

ขนาดของข้อมูล ต้นทางและปลายทางต้องเท่ากัน
คำสั่ง MOV ไม่ได้ย้ายข้อมูล แต่เป็นการสำเนาข้อมูล
ดังนั้นข้อมูลที่ต้นทางจะไม่หาย

CS231: บทที่ 2 : การกำหนดค่าและสำเนาค่า

Mnemonic : MOV

ความหมาย: Move data

การใช้งาน: MOV ปลายทาง, ต้นทาง

รูปแบบที่สามารถใช้ได้

MOV REG, IMMEDIATE

MOV REG, REG

MOV REG, MEMORY

MOV MEMORY, IMMEDIATE

MOV MEMORY, REG

รูปแบบที่สามารถใช้ได้กับ SREG

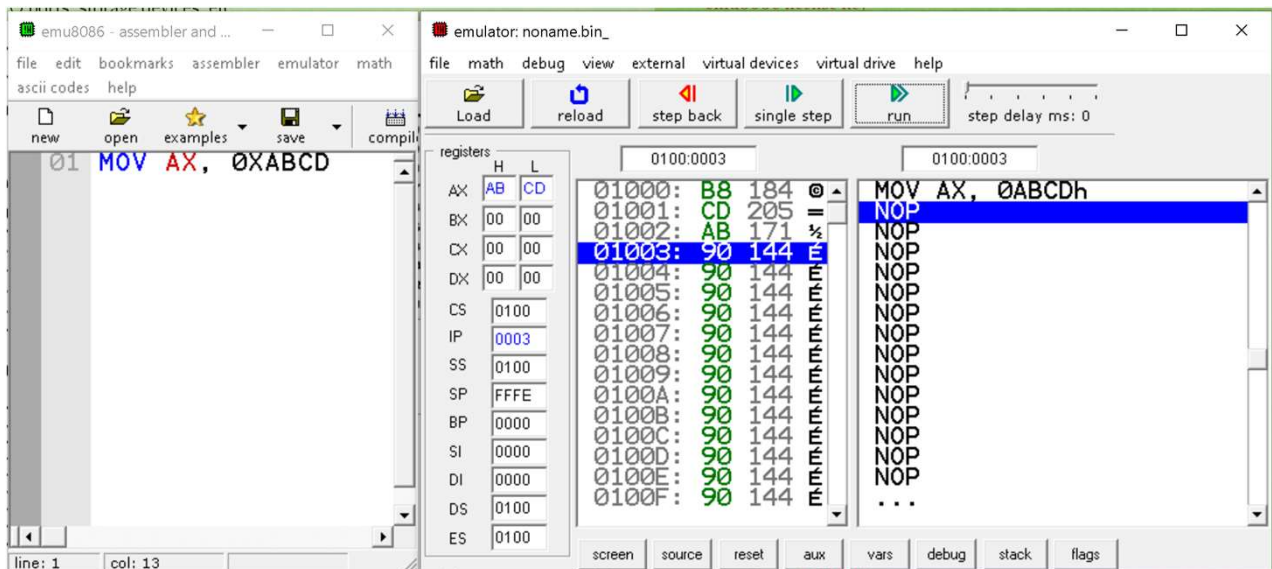
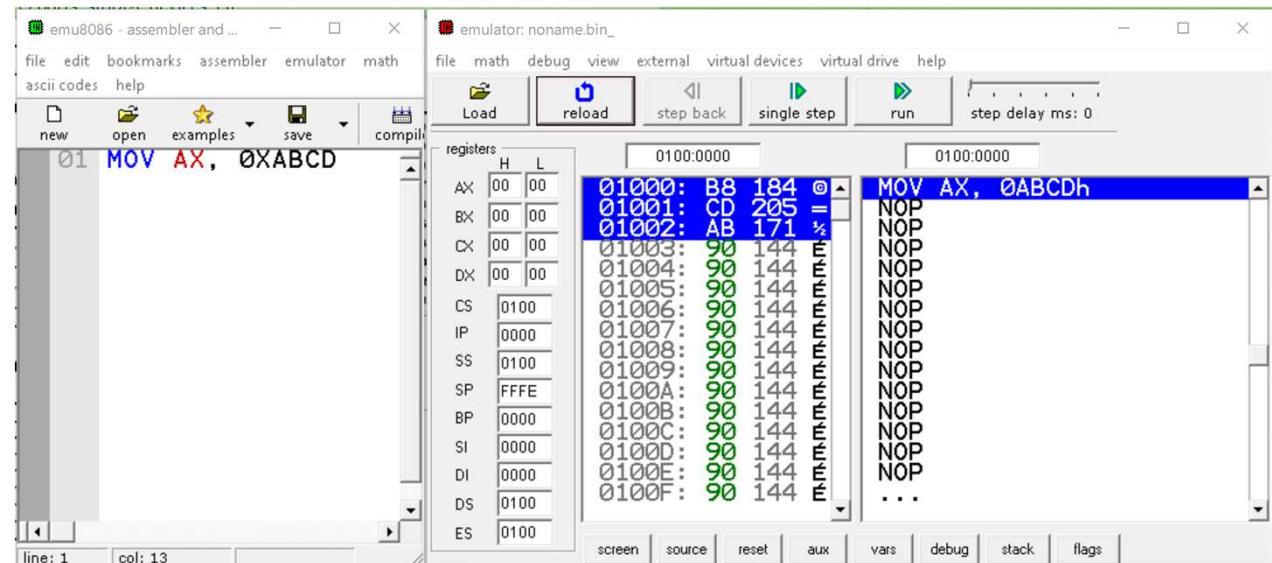
MOV SREG, MEMORY

MOV SREG, REG

MOV REG, SREG

MOV MEMORY, SREG

* ใช้กำหนดค่า CS และ IP ไม่ได้



CS231: บทที่ 2 : การกำหนดค่าและสำเนาค่า

Mnemonic : MOV

ความหมาย: Move data

การใช้งาน: MOV ปลายทาง, ต้นทาง

รูปแบบที่สามารถใช้ได้

MOV REG, IMMEDIATE

MOV REG, REG

MOV REG, MEMORY

MOV MEMORY, IMMEDIATE

MOV MEMORY, REG

รูปแบบที่สามารถใช้ได้กับ SREG

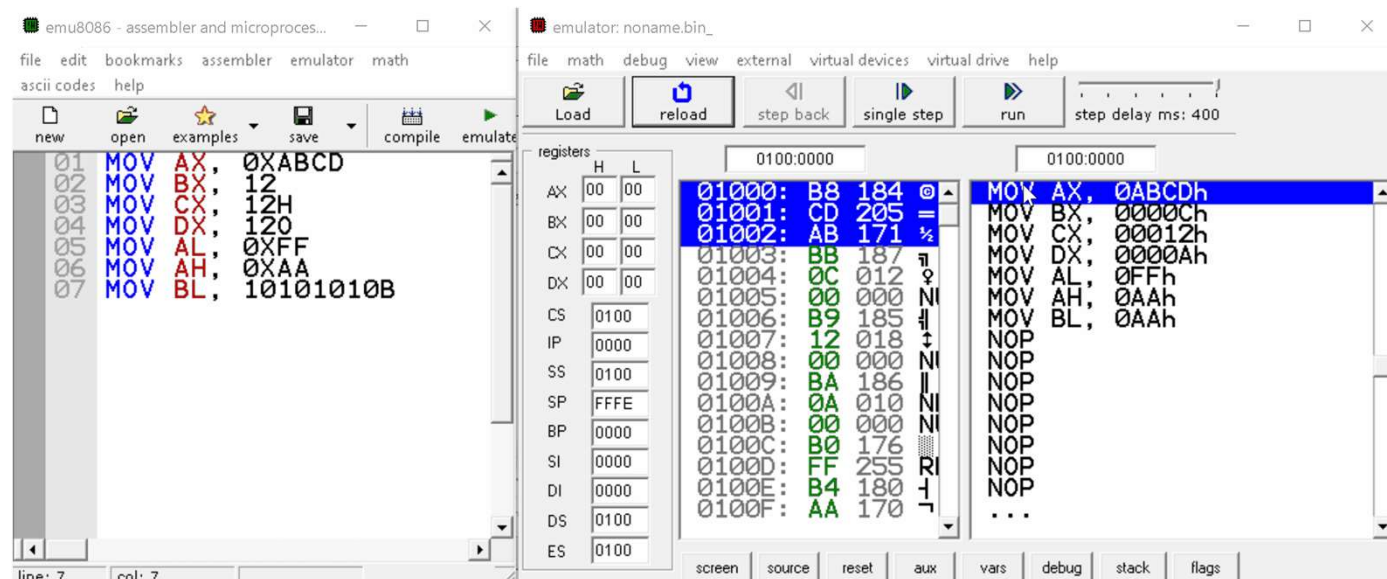
MOV SREG, MEMORY

MOV SREG, REG

MOV REG, SREG

MOV MEMORY, SREG

* ใช้กำหนดค่า CS และ IP ไม่ได้



CS231: บทที่ 2 : การกำหนดค่าและสำเนาค่า

Mnemonic : MOV

ความหมาย: Move data

การใช้งาน: MOV ปลายทาง, ต้นทาง

รูปแบบที่สามารถใช้ได้

MOV REG, IMMEDIATE

MOV REG, REG

MOV REG, MEMORY

MOV MEMORY, IMMEDIATE

MOV MEMORY, REG

รูปแบบที่สามารถใช้ได้กับ SREG

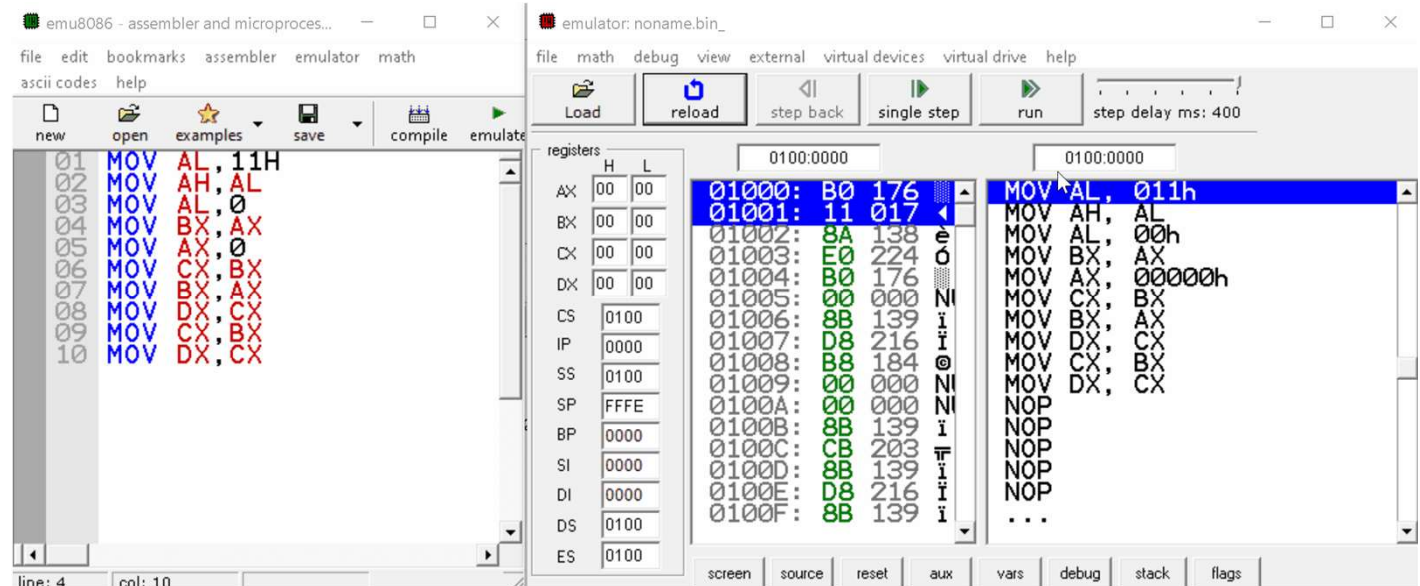
MOV SREG, MEMORY

MOV SREG, REG

MOV REG, SREG

MOV MEMORY, SREG

* ใช้กำหนดค่า CS และ IP ไม่ได้



CS231: บทที่ 2 : การกำหนดค่าและสำเนาค่า

Mnemonic : MOV

ความหมาย: Move data

การใช้งาน: MOV ปลายทาง, ต้นทาง

รูปแบบที่สามารถใช้ได้

MOV REG, IMMEDIATE

MOV REG, REG

MOV REG, MEMORY

MOV MEMORY, IMMEDIATE

MOV MEMORY, REG

รูปแบบที่สามารถใช้ได้กับ SREG

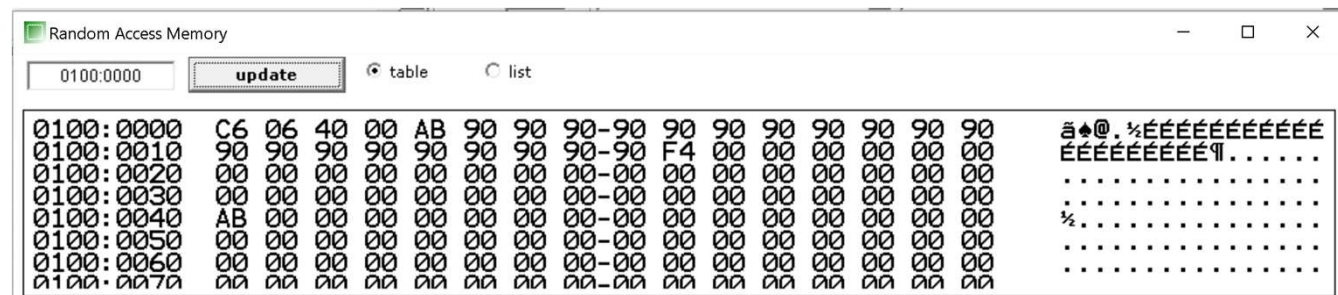
MOV SREG, MEMORY

MOV SREG, REG

MOV REG, SREG

MOV MEMORY, SREG

* ใช้กำหนดค่า CS และ IP ไม่ได้



CS231: บทที่ 2 : การกำหนดค่าและสำเนาค่า

0 1 2 3 4 5 6 7 8 9 A B C D E F

Address	Hex	ASCII
0100:0000	C6 06 40 00 AB 90 90 90-90 90 90 90 90 90 90 90	ล๑@.½EEEEEEEEEEEEEEEE
0100:0010	90 90 90 90 90 90 90 90-90 F4 00 00 00 00 00 00	EEEEEEEEEEEEEEEEEEEE
0100:0020	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	
0100:0030	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	
0100:0040	AB 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	½.....
0100:0050	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	
0100:0060	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	
0100:0070	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00	

ตำแหน่ง

Segment Offset

ค่าที่แสดงเป็นเลขฐาน16

ข้อมูล (เลขฐาน16)

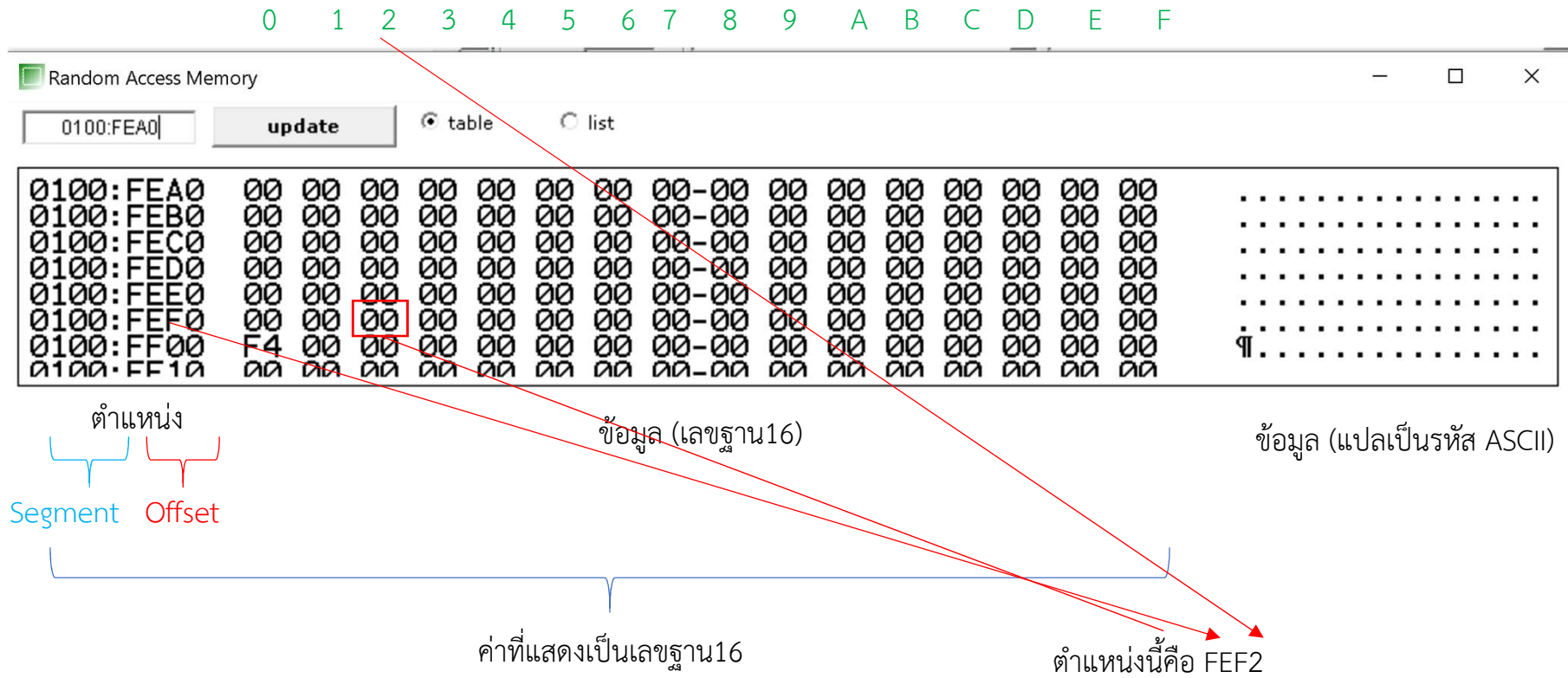
ข้อมูล (แปลเป็นรหัส ASCII)

ตำแหน่งนี้คือ 3 B

*ในบทนี้จะไม่กล่าวถึง Segment

เราจะพิจารณตำแหน่งหน่วยความจำเฉพาะส่วน Offset เท่านั้น

CS231: บทที่ 2 : การกำหนดค่าและสำเนาค่า



*ในบทนี้จะไม่กล่าวถึง Segment

เราจะพิจารณาตำแหน่งหน่วยความจำเฉพาะส่วน Offset เท่านั้น

CS231: บทที่ 2 : การกำหนดค่าและสำเนา

ฐาน16 ฐาน10 ASCII

The screenshot shows the emu8086 emulator interface. The assembly window displays the instruction `MOV AX, 0xABCD`. The registers window shows the state of various registers, with AX containing 0000. The memory dump window shows the memory contents, with the first three bytes (0100:0000) highlighted in blue and containing the values B8, CD, and AB. The random access memory window shows a table of memory addresses and their corresponding values, with the first three bytes (0100:0000) highlighted in blue and containing the values B8, CD, and AB.

Registers:

Register	H	L
AX	00	00
BX	00	00
CX	00	00
DX	00	00
CS	0100	
IP	0000	
SS	0100	
SP	FFFE	
BP	0000	
SI	0000	

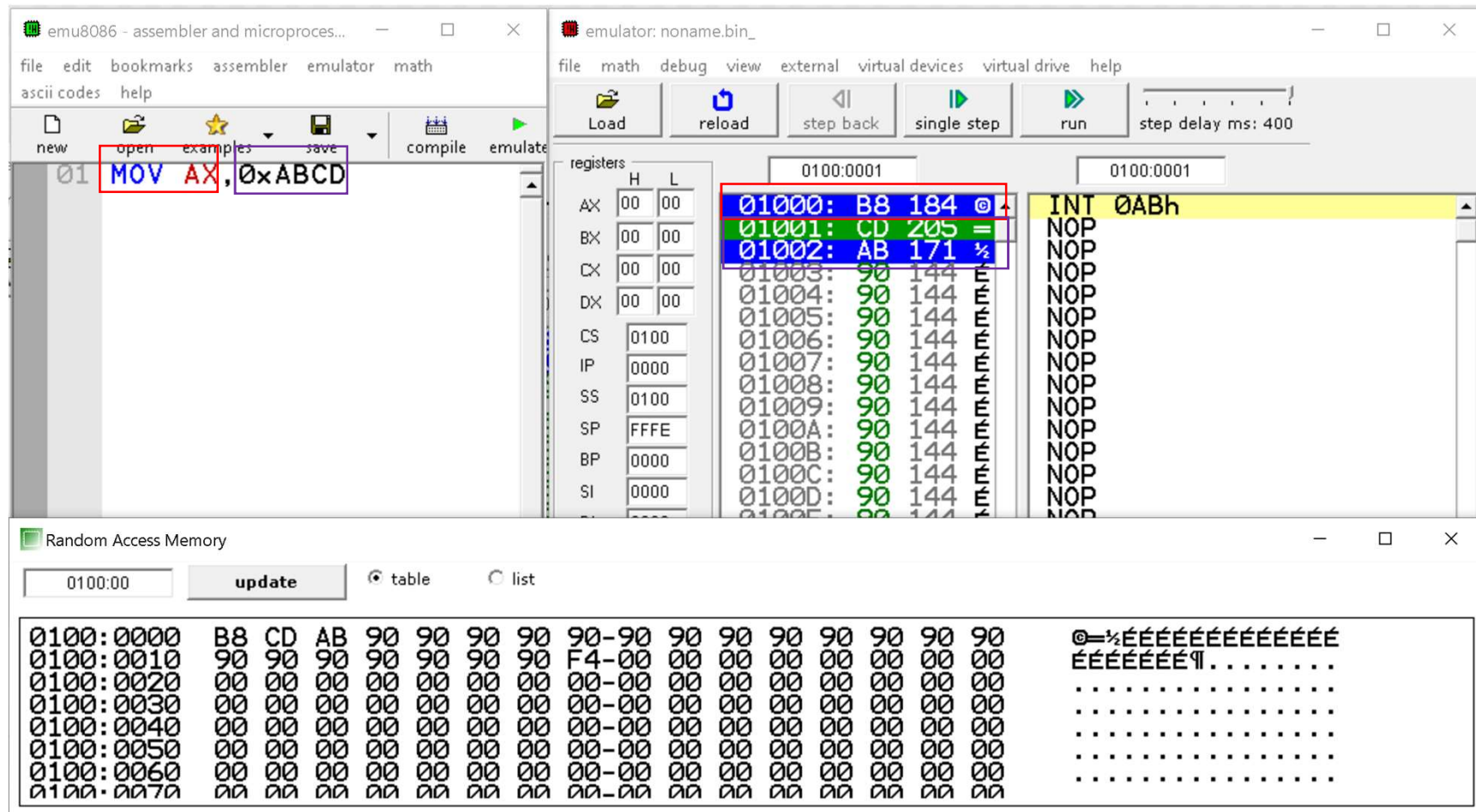
Memory Dump:

Address	Value
0100:0000	B8 CD AB
0100:0010	90 90 90
0100:0020	00 00 00
0100:0030	00 00 00
0100:0040	00 00 00
0100:0050	00 00 00
0100:0060	00 00 00
0100:0070	00 00 00

Random Access Memory:

Address	Value
0100:0000	B8 CD AB
0100:0010	90 90 90
0100:0020	00 00 00
0100:0030	00 00 00
0100:0040	00 00 00
0100:0050	00 00 00
0100:0060	00 00 00
0100:0070	00 00 00

CS231: บทที่ 2 : การกำหนดค่าและสำเนา



Intel 8086 บันทึกข้อมูลลงหน่วยความจำแบบ Little-endian (LE) least-significant byte จึงถูกเขียนไว้ก่อน most-significant byte
ABCD เมื่อถูกเขียนลงหน่วยความจำจึงกลายเป็น CDAB
ข้อมูลขนาด 1 byte เรียกว่า byte และ ข้อมูลขนาด 2 bytes จะเรียกอีกชื่อว่า word
1 word = 2 bytes

CS231: บทที่ 2 : การกำหนดค่าและสำเนาค่า

Mnemonic : MOV

ความหมาย: Move data

การใช้งาน: MOV ปลายทาง, ต้นทาง

รูปแบบที่สามารถใช้ได้

MOV REG, IMMEDIATE

MOV REG, REG

MOV REG, MEMORY

MOV MEMORY, IMMEDIATE

MOV MEMORY, REG

รูปแบบที่สามารถใช้ได้กับ SREG

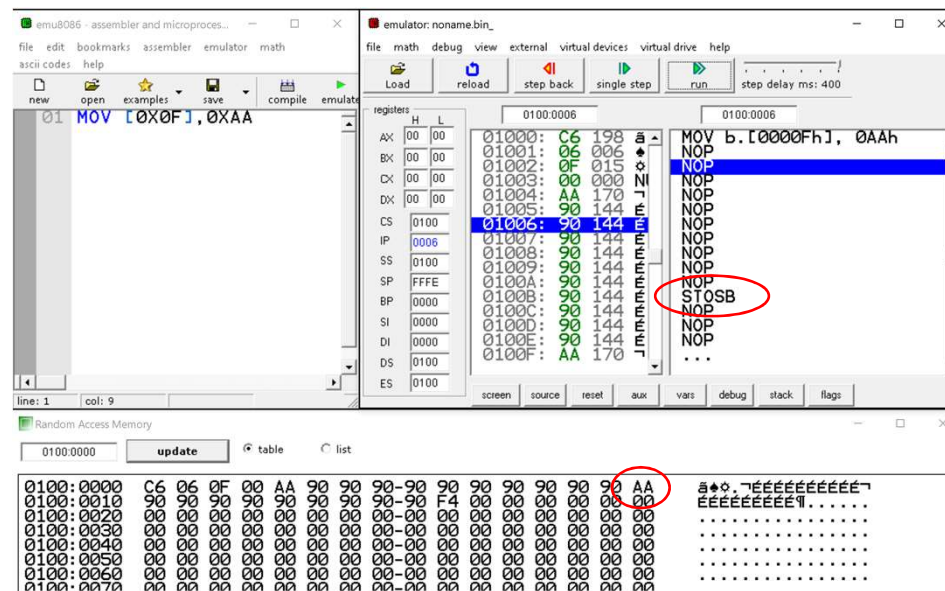
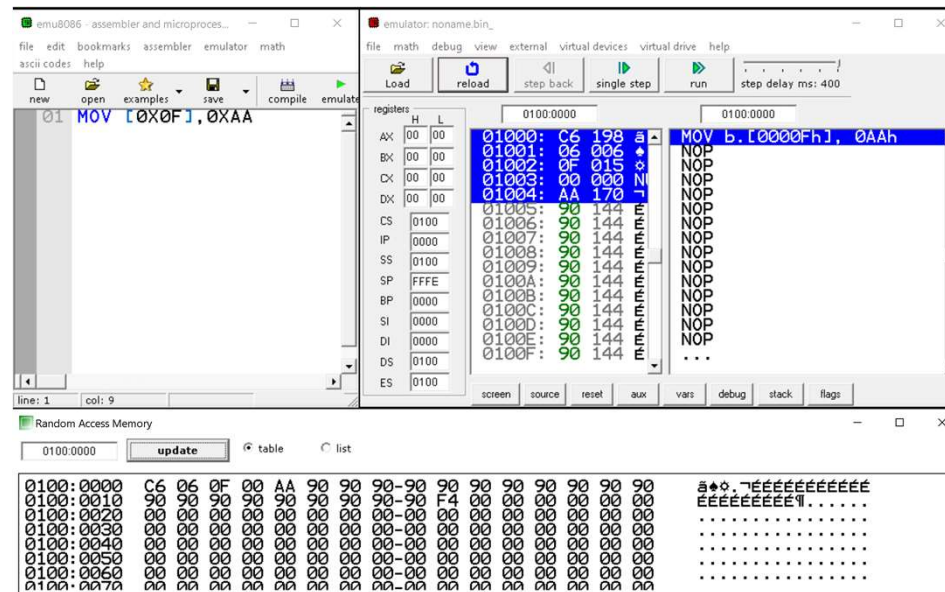
MOV SREG, MEMORY

MOV SREG, REG

MOV REG, SREG

MOV MEMORY, SREG

* ใช้กำหนดค่า CS และ IP ไม่ได้



CS231: บทที่ 2 : การกำหนดค่าและสำเนา

ไมโครโพรเซสเซอร์ 8086 มีสถาปัตยกรรมแบบ

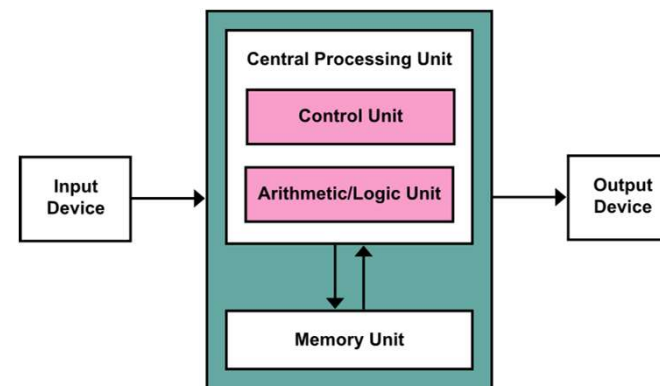
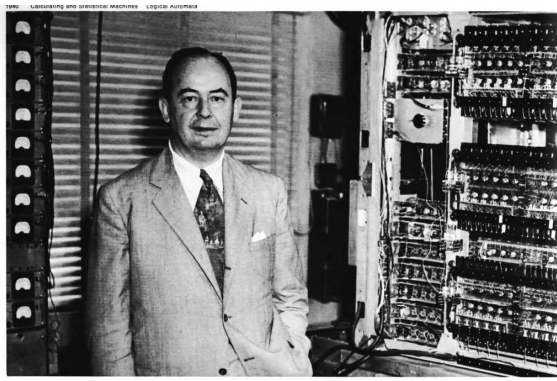
von Neumann architecture

โปรแกรมและข้อมูล จะเก็บไว้ในหน่วยความจำเดียวกัน เข้าถึงได้แบบเดียวกัน

มีข้อดีคือ สามารถเปลี่ยนโปรแกรมได้โดยง่าย (แค่โหลดโปรแกรมเข้าสู่หน่วยความจำก็รันได้เลย)

ข้อเสียคือ หน่วยประมวลผลไม่มีทางรู้ว่าข้อมูลนั้นคือโปรแกรมจริง ๆ หรือไม่
hacker ก็ใช้วิธีนี้ในการส่งโปรแกรมเข้าไปรันโดยไม่ได้รับอนุญาต

เป็นหน้าที่ของโปรแกรมเมอร์ในการติดตามตำแหน่งและข้อมูลต่าง ๆ ในหน่วยความจำ (Memory Tracking) และป้องกันไม่ให้เขียนข้อมูลลงไปทับตำแหน่งที่มีโปรแกรมเราติดตั้งอยู่



A von Neumann architecture scheme

CS231: บทที่ 2 : การกำหนดค่าและสำเนาค่า

Mnemonic : MOV

ความหมาย: Move data

การใช้งาน: MOV ปลายทาง, ต้นทาง

รูปแบบที่สามารถใช้ได้

MOV REG, IMMEDIATE

MOV REG, REG

MOV REG, MEMORY

MOV MEMORY, IMMEDIATE

MOV MEMORY, REG

รูปแบบที่สามารถใช้ได้กับ SREG

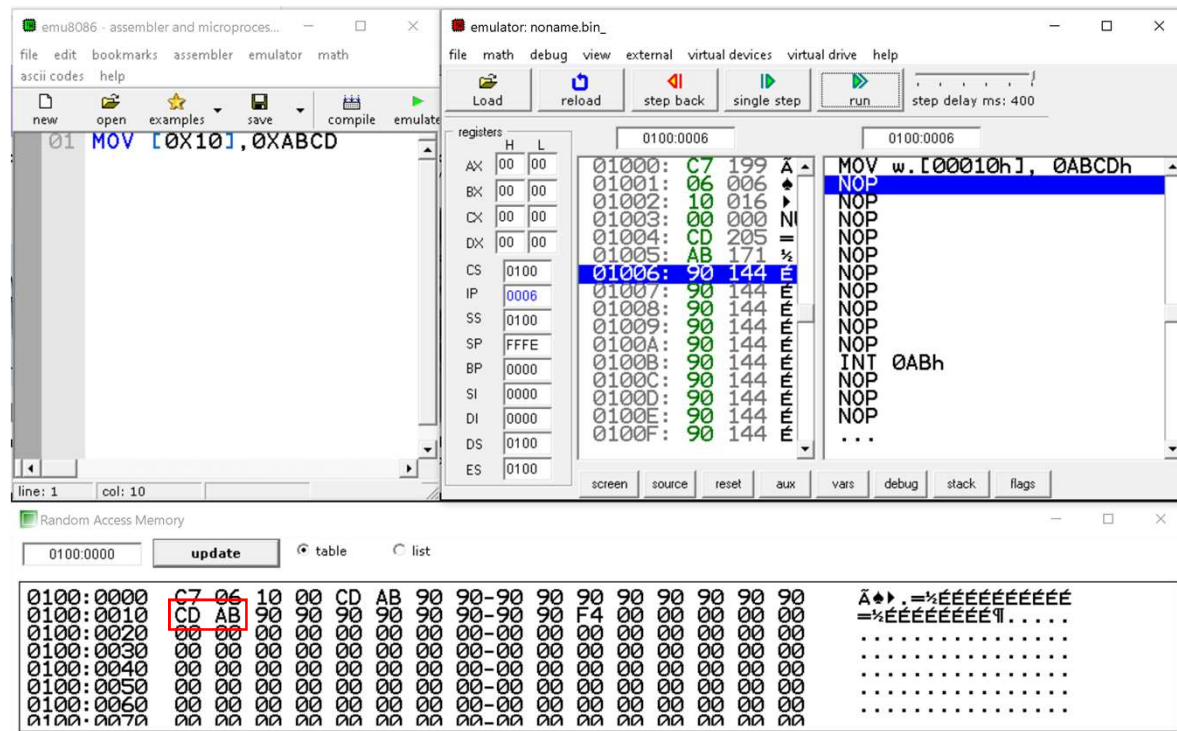
MOV SREG, MEMORY

MOV SREG, REG

MOV REG, SREG

MOV MEMORY, SREG

* ใช้กำหนดค่า CS และ IP ไม่ได้



CS231: บทที่ 2 : การกำหนดค่าและสำเนาค่า

Mnemonic : MOV

ความหมาย: Move data

การใช้งาน: MOV ปลายทาง, ต้นทาง

รูปแบบที่สามารถใช้ได้

MOV REG, IMMEDIATE

MOV REG, REG

MOV REG, MEMORY

MOV MEMORY, IMMEDIATE

MOV MEMORY, REG

รูปแบบที่สามารถใช้ได้กับ SREG

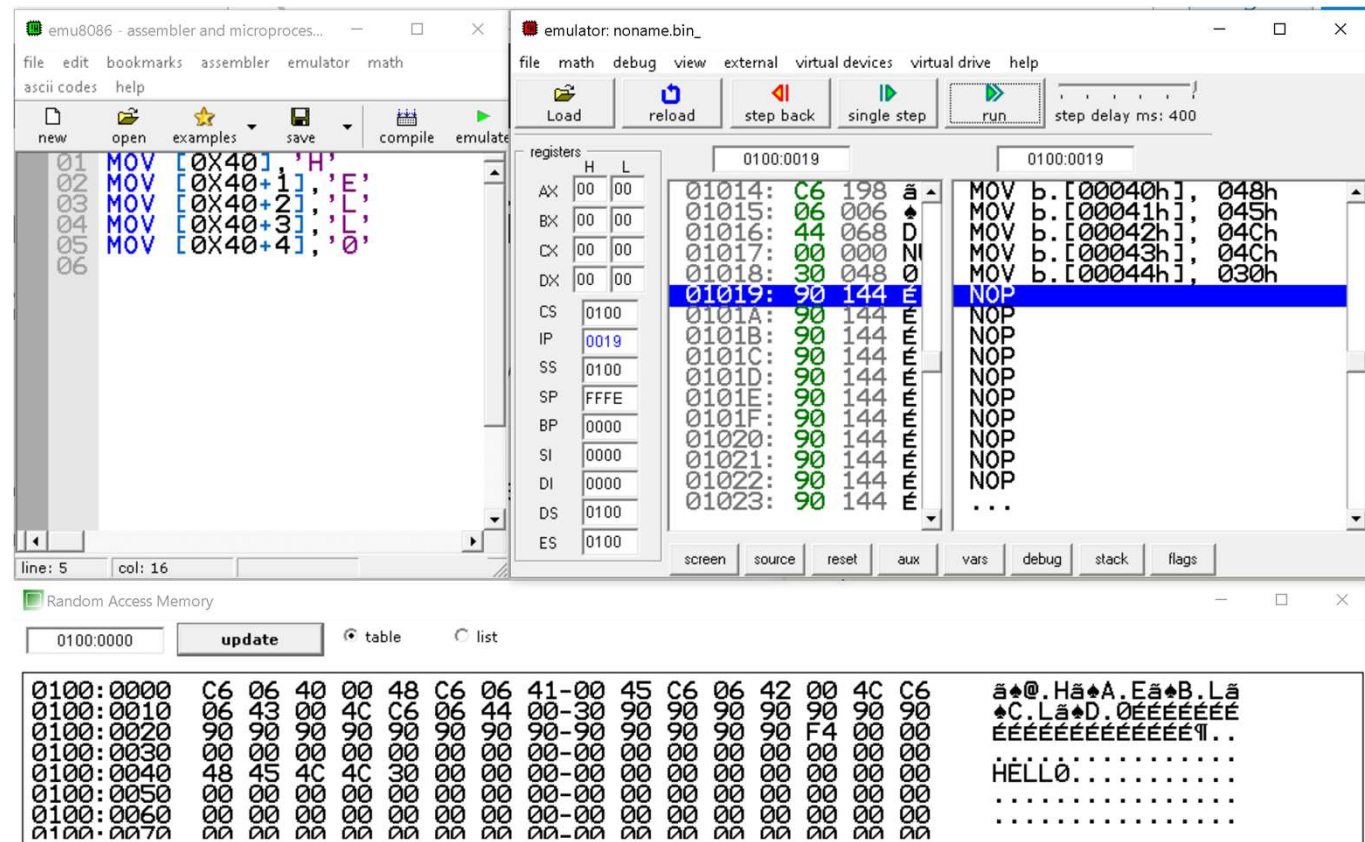
MOV SREG, MEMORY

MOV SREG, REG

MOV REG, SREG

MOV MEMORY, SREG

* ใช้กำหนดค่า CS และ IP ไม่ได้



CS231: บทที่ 2 : การกำหนดค่าและสำเนาค่า

Mnemonic : MOV

ความหมาย: Move data

การใช้งาน: MOV ปลายทาง, ต้นทาง

รูปแบบที่สามารถใช้ได้

MOV REG, IMMEDIATE

MOV REG, REG

MOV REG, MEMORY

MOV MEMORY, IMMEDIATE

MOV MEMORY, REG

รูปแบบที่สามารถใช้ได้กับ SREG

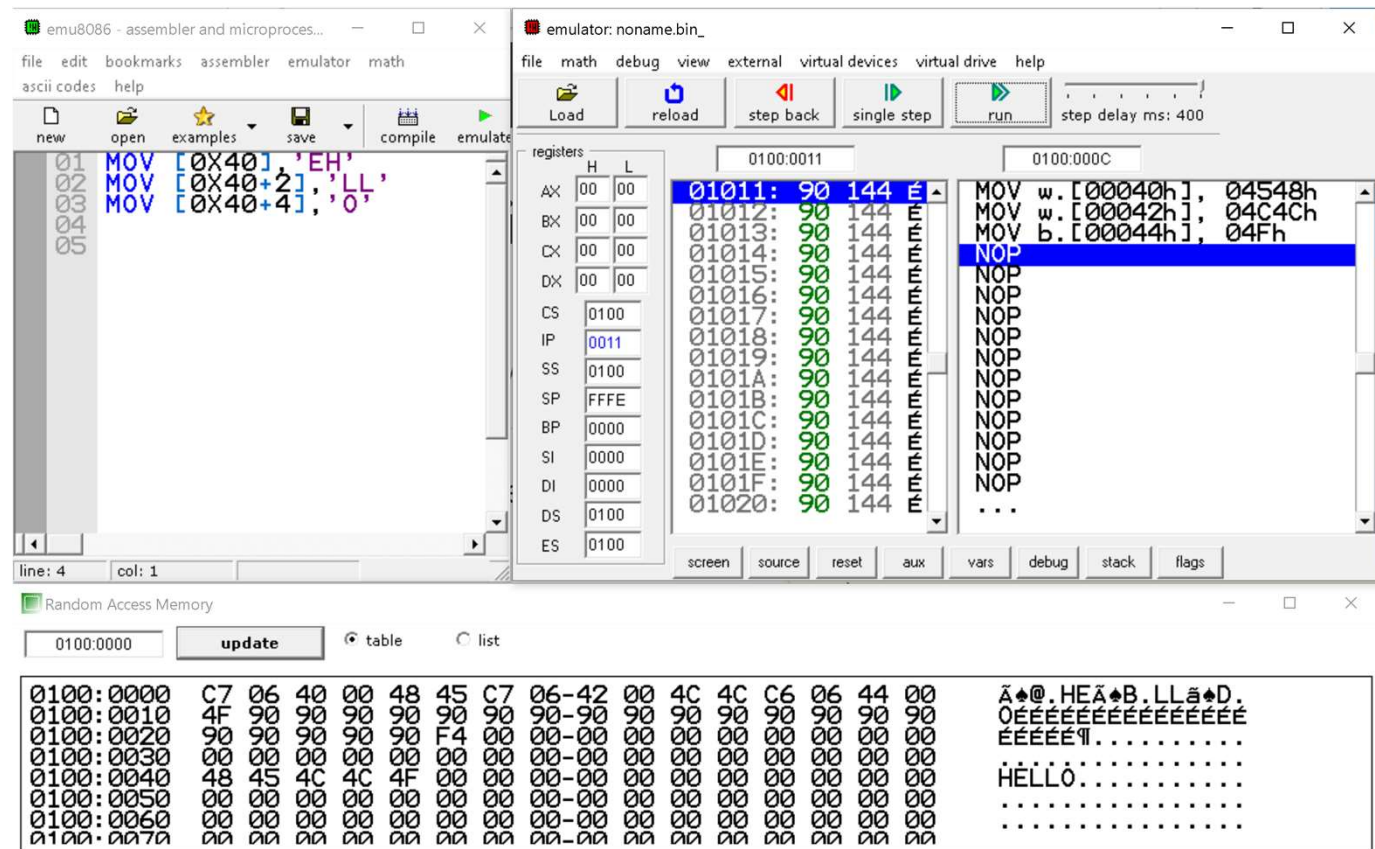
MOV SREG, MEMORY

MOV SREG, REG

MOV REG, SREG

MOV MEMORY, SREG

* ใช้กำหนดค่า CS และ IP ไม่ได้



CS231: บทที่ 2 : การกำหนดค่าและสำเนาค่า

Mnemonic : MOV

ความหมาย: Move data

การใช้งาน: MOV ปลายทาง, ต้นทาง

รูปแบบที่สามารถใช้ได้

MOV REG, IMMEDIATE

MOV REG, REG

MOV REG, MEMORY

MOV MEMORY, IMMEDIATE

MOV MEMORY, REG

รูปแบบที่สามารถใช้ได้กับ SREG

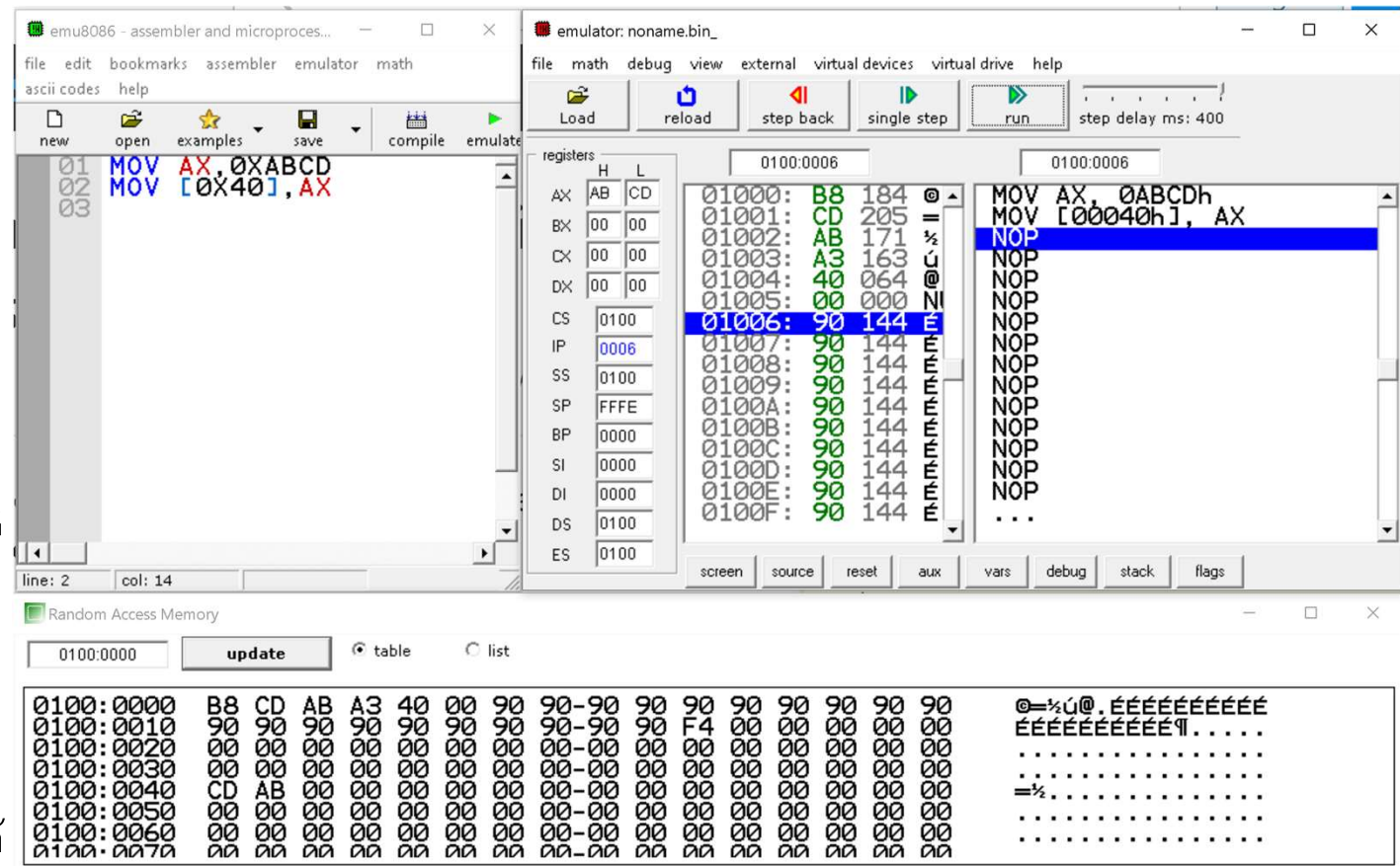
MOV SREG, MEMORY

MOV SREG, REG

MOV REG, SREG

MOV MEMORY, SREG

* ใช้กำหนดค่า CS และ IP ไม่ได้



CS231: บทที่ 2 : การกำหนดค่าและสำเนาค่า

Mnemonic : MOV

ความหมาย: Move data

การใช้งาน: MOV ปลายทาง, ต้นทาง

รูปแบบที่สามารถใช้ได้

MOV REG, IMMEDIATE

MOV REG, REG

MOV REG, MEMORY

MOV MEMORY, IMMEDIATE

MOV MEMORY, REG

Segment Register ใช้งาน MOV ได้เหมือน Register ทั่วไป
แต่จะแบ่งเป็น byte ต่ำ และ byte สูงไม่ได้ (xL, xH)

รูปแบบที่สามารถใช้ได้กับ SREG

MOV SREG, MEMORY

MOV SREG, REG

MOV REG, SREG

MOV MEMORY, SREG

* ใช้กำหนดค่า CS และ IP ไม่ได้

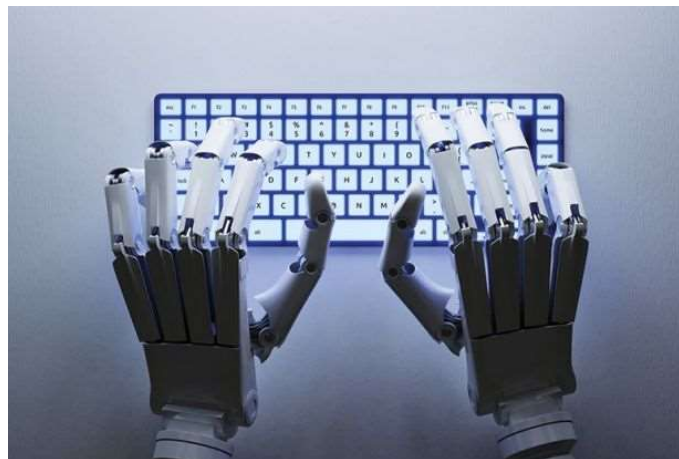
CS231: บทที่ 2 : การกำหนดค่าและสำเนาค่า

Mnemonic : MOV

ความหมาย: Move data

การใช้งาน: MOV ปลายทาง, ต้นทาง

การโปรแกรมเครื่อง von Neumann architecture
โปรแกรมเมอร์รู้แค่คำสั่ง MOV ก็เพียงพอแล้ว



CS231: บทที่ 2 : การกำหนดค่าและสำเนาค่า

Mnemonic : MOV

ความหมาย: Move data

การใช้งาน: MOV ปลายทาง, ต้นทาง

The screenshot displays the emu8086 software interface, which is divided into several panes. The top-left pane shows the assembly code with the following instructions:

```
01 MOV AX, 0xFFFF
02 MOV BX, 0x0000
03 MOV BX, 0xFFFF
04 MOV AX, 0x0000
05 MOV [0x12], 0xEFEb
06
```

The top-right pane shows the registers (AX, BX, CX, DX, CS, IP, SS, SP, BP, SI, DI, DS, ES) and their current values. The bottom-left pane shows the memory dump (Random Access Memory) with the following data:

Address	Hex Data	ASCII Data
0100:0000	B8 FF FF BB 00 00 BB FF FF B8 00 00 C7 06 12 00	0 0
0100:0010	EB EF 90 90 90 90 90 90 90 90 90 90 90 90 90 90	U'EEEEEEEEEEEEEEEE
0100:0020	90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90	EEEEEEEEEEEEEEEE
0100:0030	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	EEEEEEEEEEEEEEEE
0100:0040	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	EEEEEEEEEEEEEEEE
0100:0050	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	EEEEEEEEEEEEEEEE
0100:0060	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	EEEEEEEEEEEEEEEE
0100:0070	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	EEEEEEEEEEEEEEEE

The bottom-right pane shows the disassembly of the memory dump, displaying the instructions corresponding to the hex data.

CS231: บทที่ 2 : การกำหนดค่าและสำเนาค่า

การบ้านที่ 1 เรื่องสำเนาข้อมูล (5 คะแนน)

ให้เขียนโปรแกรมภาษาแอสเซมบลีที่มีการทำงานดังต่อไปนี้

1. ให้บันทึกค่า `a b c d e f g h i j` ลงในหน่วยความจำโดยเริ่มต้นจากตำแหน่ง `0100:0100`
2. ให้สำเนาค่าในหน่วยความจำตำแหน่งที่ `0100:0100` ถึงตำแหน่งที่ `0100:0009` ไปยังตำแหน่ง `0100:0110` ตามลำดับ

หลังจากนั้นให้ทำการเปลี่ยน `a .. i` ให้เป็นรหัสนักศึกษาของตนเอง แล้วรันโปรแกรมอีกครั้งโดยไม่แก้ `code` ในส่วนอื่น

และแสดงให้เห็นว่า ข้อมูลในหน่วยความจำ เปลี่ยนเป็นรหัสนักศึกษาของตนเอง

การส่งงานให้นักศึกษาสำเสนอเป็นวิดีโอคลิปความยาวไม่เกิน 3 นาที โดยมีการแสดง `source code` ผลการทำงาน

และทำการให้คะแนนโปรแกรมของตนเอง โดยมีเกณฑ์ การให้คะแนนดังนี้

หลังจากรันโปรแกรมแล้ว

1. มี `a b c d e f g h i j` อยู่ในหน่วยความจำ (1 คะแนน)
2. มี `a b c d e f g h i j` อยู่ในหน่วยความจำที่ `0100:0100`(2 คะแนน)
3. ค่าที่ `0100:0110` เปลี่ยนไปตามค่า `0100:0100` ที่เปลี่ยนไป(2 คะแนน)
4. หากไม่แสดง `source code` และ อธิบาย `source code` การบ้านนี้จะได้ 0 คะแนน

ให้นักศึกษานำบันทึกวิดีโอด้วยรหัส H.264 รูปแบบแฟ้ม .MP4 หรือ .MKV แล้ว upload เข้ามาใน Microsoft Team

ภายในวันที่ 31 กรกฎาคม 2564

CS231: บทที่ 2 : การกำหนดค่าและสำเนาค่า

การบ้านที่ 1 : การสำเนาข้อมูล

edit: G:\My Drive\Lectures\CS231_Asse... — □ ×

file edit bookmarks assembler emulator math
ascii codes help

new open examples save compile emulate

```
01 ; Lab1 Mov
02 ;
03
04 MOV [0X100], 'a'
05 MOV [0X101], 'b'
06 MOV [0X102], 'c'
07 MOV [0X103], 'd'
08 MOV [0X104], 'e'
09
10 ; COPY DATA FROM 0X100
```

line: 8 col: 15

emulator: LAB1_1.bin_ — □ ×

file math debug view external virtual devices
virtual drive help

Load reload step back single step run

registers H L 0100:0000

AX	00	00
BX	00	00
CX	00	00
DX	00	00
CS	0100	
IP	0000	
SS	0100	

0100:0000 0100:0001 0100:0002 0100:0003 0100:0004 0100:0005 0100:0006

C6 198 ä M
06 006 ♠ M
00 000 N
01 001 ☺
31 049 1
C6 198 ä
06 006 ♠

screen source reset aux vars

Random Access Memory

0100:0000 update table list

0100:0000	C6	06	00	01	31	C6	06	01-01	32	C6	06	02	01	33	C6	ä	♠	☺	1	ä	☺	☺	2	ä	♠	☺	☺	3	ä
0100:0010	06	03	01	34	C6	06	04	01-35	A1	00	01	A3	10	01	A1	♠	♥	☺	4	ä	♠	☺	5	í	.	☺	☺	☺	í
0100:0020	02	01	A3	12	01	A1	04	01-A3	14	01	A1	06	01	A3	16	☺	☺	☺	☺	☺	☺	☺	☺	☺	☺	☺	☺	☺	
0100:0030	01	90	90	90	90	90	90	90-90	90	90	90	90	90	90	90	☺	☺	☺	☺	☺	☺	☺	☺	☺	☺	☺	☺	☺	
0100:0040	90	90	90	90	90	F4	00	00-00	00	00	00	00	00	00	00	É	É	É	É	É	É	É	É	É	É	É	É	É	
0100:0050	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	É	É	É	É	É	É	É	É	É	É	É	É	É	
0100:0060	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	É	É	É	É	É	É	É	É	É	É	É	É	É	
0100:0070	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	É	É	É	É	É	É	É	É	É	É	É	É	É	