บทที่ 2

การเอ็กซีคิวส์ของเครื่อง

MACHINE EXECUTION

วัตถุประสงค์

เมื่อท่านศึกษาบทนี้แล้วท่านจะเข้าใจดังต่อไปนี้

- วิธีการบูตของเครื่องคอมพิวเตอร์
- ตารางการบริการอินเตอร์รัพท์
- การกำหนดแอดเดรสของหน่วยความจำ

CT 215 23

บทที่ 2
การเอ็กซีคิวส์ของเครื่อง
MACHINE EXECUTION

บทนำ

บทนี้จะเริ่มต้นอธิบายในส่วนทั่วๆไปของกรรมวิธีการเริ่มต้นทำงาน (BOOT PROCESS) ที่เอ็กซีคิวส์คำสั่งที่อยู่ในหน่วยความจำที่เขียนอยู่ใน ROM การอธิบายในบทนี้จะใช้โปรแกรมของ DOS ที่เรียกว่า DEBUG ซึ่งเป็นการแสดงข้อมูลของหน่วยความจำเป็นผลที่เกิดจากกระทำของโปรแกรม การเอ็กซ์ที่ละคำสั่ง เพื่อดูการทำงานของคำสั่ง หรืออาจจะใช้โปรแกรมอีกตัวหนึ่งเรียกว่า CODEVIEW หรือ TURBO DEBUGGER

ในการใช้โปรแกรม DEBUG ซึ่งเป็นตัวอย่างที่เข้าใจง่ายๆ ในการแสดงการทำงานของคำสั่งและแสดงผลลัพธ์ที่เกิดขึ้นในหน่วยความจำ โปรแกรมแรกเป็นการคำใช้ที่กำหนดข้อมูลให้โดยตรงมาพร้อมคำสั่ง ซึ่งข้อมูลเหล่านี้จะแยกส่วนกับข้อมูลใน DATA SEGMENT โดยใช้คำสั่ง แทรต (T) เพื่อการเอ็กซีคิวส์คำสั่งภาษาเครื่อง (MACHINE EXECUTE) ที่จะทำให้เห็นการทำงานของคอมพิวเตอร์ และผลลัพธ์ในรีจีสเตอร์ โปรแกรมนี้ท่านจะต้องศึกษาก่อนที่ท่านจะเขียนโปรแกรมด้วยภาษาแอสแซมบลี

ขบวนการบูต BOOTING PROCESS

ในการใช้เครื่องคอมพิวเตอร์ เราจะต้องเปิดสวิทช์เพื่อจ่ายกระแสไฟฟ้าให้กับเครื่องคอมพิวเตอร์ เรียกว่า COLD BOOT คอมพิวเตอร์จะอยู่ในสภาวะรีเซ็ท (RESET) จะทำการเคลียตำแหน่งแอดเดรสของหน่วยความจำให้เริ่มที่แอดเดรส 0 คือเป็นการเซ็ทค่ารีจีสเตอร์ CS มีค่าเป็น FFFF[0]H และค่าของ IP = 0 คำสั่งแรกที่เอ็กซีคิวส์จะอยู่ที่ตำแหน่ง FFFF[0]H ซึ่งเป็นจุดเริ่มต้น BIOS ของโปรแกรมที่อยู่ใน ROM

โปรแกรม BIOS จะทำการตรวจสอบ PORT ต่างๆ และตรวจสภาพความพร้อมของอุปกรณ์ของระบบ โปรแกรม BIOS จะใช้การบริการอินเตอร์รัพธ์ ตามข้อมูลในแอดเดรสของตารางอินเตอร์รัพธ์ ตารางนี้จะเริ่มต้นที่แอดเดรส 0 ในหน่วยความจำ การทำงานของโปรแกรม BIOS ต่อไปจะโหลดข้อมูลของ DOS ที่อยู่ในแผ่นดิสก์ โปรแกรม DOS ที่โหลดออกมาคือ IBMBIO.COM , IBMDOS.COM และ COMMAND.COM มายังหน่วยความจำของคอมพิวเตอร์ ถ้าในการโหลดของ MS-DOS จะเป็นการโหลด IO.SYS และ MSDOS.SYS การทำงานจะเกิดขึ้นตามขั้นตอนต่อไปนี้

INTERRUPT SERVICES TABLE

BIOS DATA AREA

IBMBIO.COM AND IBMDOS.COM

COMMAND.COM resident portion

AVAILABLE FOR PROGRAM'S USE

COMMAND.COM transident portion

ROM BIOS

แอดเดรสของเครื่อง MACHINE ADDRESSING

ในส่วนของบทที่ 1 ได้อธิบายเกี่ยวกับรีจีสเตอร์ CS เป็นตัวเก็บข้อมูลแอดเดรสที่เริ่มแรกของโปรแกรมใน CODE SEGMENT และรีจีสเตอร์ DS เป็นตัวเก็บข้อมูลที่แอดเดรสเริ่มแรกของ DATA - SEGMENT ส่วนของ CODE SEGMENT จะเป็นตัวเก็บรหัสคำสั่งที่ใช้ในการเอ็กซีคิวส์ ขณะที่ DATA SEGMENT จะเก็บข้อมูลของรหัสคำสั่ง ส่วนรีจีสเตอร์ IP เป็นตัวชี้แอดเดรสของคำสั่งในปัจจุบันใน CODE SEGMENT ที่จะนำมาเอ็กซีคิวส์ รีเปอร์รานด์ของคำสั่งจะเป็นค่าของ OFFSET ADDRESS ในส่วนของ DATA - SEGMENT ที่อ้างอิงโดยคำสั่ง

ถ้าพิจารณาจากตัวอย่างในรีจีสเตอร์ CS มีข้อมูล 04AF[0]H และรีจีสเตอร์ DS มีข้อมูล 04b1[0]H และค่าของ IP มีข้อมูล 0013H ฉนั้นแอดเดรสของคำสั่งต่อไปที่จำนำมาเอ็กซีคิวส์คือ

CS : 04AF0

IP : + 0013

INSTRUCTION ADDRESS : +04B03

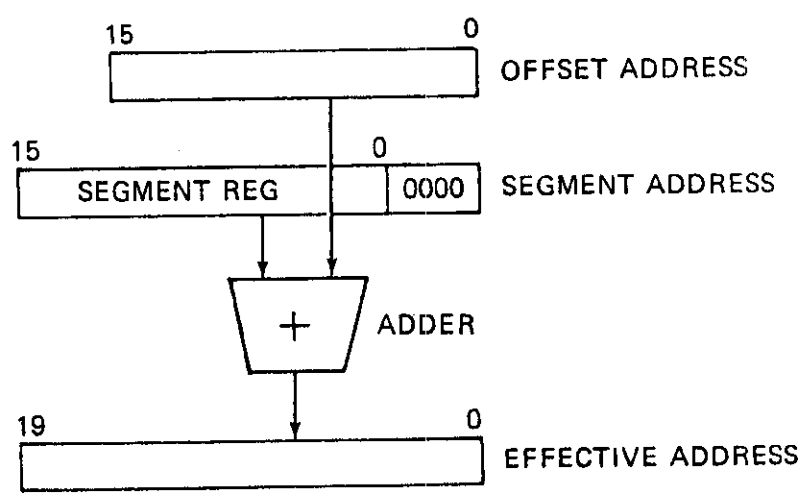
ถ้าคำสั่งเริ่มต้นที่ 04B03 ก็จะเป็นการเคลื่อนย้ายข้อมูลขนาด 1 ไบท์จากหน่วยความจำไปไว้ที่รีจีสเตอร์ AL ค่า OFFSET คือ 0012 ที่ชี้ในส่วนของ DATA SEGMENT รหัสของภาษาเครื่องและสัญญลักษณ์ของคำสั่งมีดังนี้

```
A01200          MOV     AL,[0012]
  ¦
```

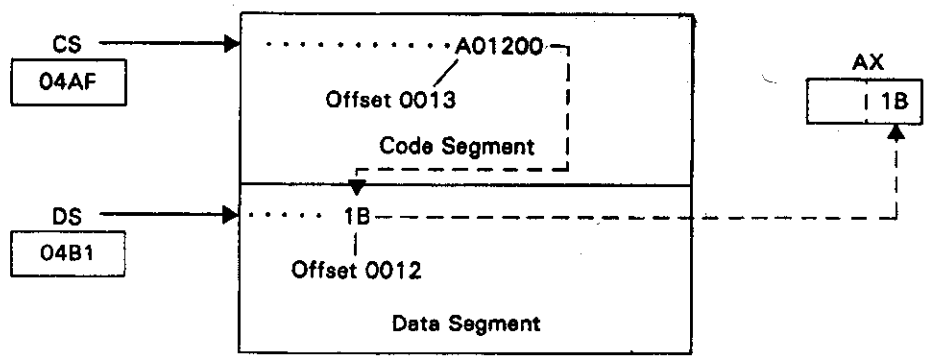
แอดเดรส 04B03 ชี้ที่ A0

แอดเดรสของหน่วยความจำ 04B03 ข้อมูลไบท์แรกเป็นรหัสคำสั่งที่จะมีการอ่านไบท์แรก ส่วนไบท์ที่ 2 ไบท์ที่ 3 เป็นข้อมูลของค่า OFFSET ADDRESS เราจะเห็นว่าข้อมูลในรหัสภาษาเครื่องจะสลับกันระหว่างไบท์สูงกับไบท์ต่ำ การเข้าถึงข้อมูล โปรเซสเซอร์จะกำหนดแอดเดรสจากข้อมูลในรีจีสเตอร์ DS บวกกับข้อมูล OFFSET (0012) ที่อยู่ในโอเปอร์รานด์ของคำสั่ง ข้อมูลใน DS มีค่า 04B10H ตำแหน่งที่อ้างอิงแอดเดรสในรายการข้อมูลคือ

```
                DS  :     04B10
            OFFSET  :   + 0012
ADDRESS OF DATA ITEM :   + 04B22
```



เราจะเห็นว่าแอดเดรส 04B22 มีข้อมูล 1BH โปรเซสเซอร์จะทำการอ่านข้อมูล 1BH ที่แอดเดรส 04B22 และนำไปไว้ที่รีจีสเตอร์ AL ดังรูป 2.1



Segments and Offsets

รูป 2.1 เซคเม้นต์และออฟเซ็ท

โปรเซสเซอร์จะทำการเฟทช์คำสั่งที่ละไบท์ตามคำสั่ง ค่าของ IP จะเพิ่มขึ้นที่ละหนึ่งดังนั้นข้อมูลของค่า OFFSET (0016) สำหรับการเฟทช์คำสั่งถัดไป เพื่อให้คอมพิวเตอร์พร้อมที่จะเอ็กซิคิวส์คำสั่งต่อไป ซึ่งข้อมูลของ CS (04AF0) บวกกับค่า OFFSET ปัจจุบันใน IP (0016)

คำสั่งอาจจะมีการเข้าถึงข้อมูลมากกว่า 1 ไบท์ในแต่ละครั้งดังตัวอย่าง คำสั่งที่ใช้ในการเก็บข้อมูลของ AX (0123H) ซึ่งมีข้อมูลขนาด 2 ไบท์ ที่อยู่ในส่วนของ DATA SEGMENT เริ่มต้นที่ค่า OFFSET คือ 0012 จากคำสั่ง MOV [0012],AX โปรเซสเซอร์จะเก็บข้อมูลสลับไบท์กัน

ข้อมูลของไบท์	23	01
	¦	¦
Offset ของ Data Segment	0012	0013

คำสั่ง MOV AX,[0012] เป็นการอ่านข้อมูลจากหน่วยความจำไปเก็บไว้ใน AX การทำงานจะสลับข้อมูลใน AX เป็นดังนี้ 0123

EVEN – NUMBERED ADDRESSES

8086,80286,80386 และ 80486 ตัวโปรเซสเซอร์เหล่านี้จะมีการทำงานในการเอ็กซีคิวส์ที่มีประสิทธิภาพมาก ถ้าเวิร์ดการเข้าถึงข้อมูล เริ่มที่แอดเดรสคู่ ในการทำงานคำสั่ง MOV ครั้งก่อนเราจะเห็นว่า โปรเซสเซอร์เข้าถึงข้อมูลเป็นเวิร์ดที่ค่า OFFSET 0012 ที่กำหนดในรีจีสเตอร์ แต่ถ้าเวิร์ดเริ่มต้นที่แอดเดรสคี่ เช่น 0013

Memory contains	:	X X	2 3	0 1	X X
offset	:	0012	0013	0014	0015

ในกรณีนี้ โปรเซสเซอร์จะเข้าถึงข้อมูล 2 ครั้ง ครั้งแรกโปรเซสเซอร์เข้าถึงข้อมูลไบท์ที่ 0012และ 0013 ข้อมูลที่ 0013 เก็บใน AL และจะเข้าถึงข้อไบท์ที่ 0014 และ 0015 ข้อมูล 0014 เก็บใน AH ปัจจุบัน AX จะมีข้อมูล 0123

ภาษาแอสแซมบลีจะมีคำสั่งเทียม EVEN ท่านสามารถนำไปใช้กำหนดข้อมูลของ DATA SEGMENT ข้อมูลตัวเลขจะเริ่มต้นด้วยแอดเดรสคู่

การแสดงตำแหน่งในหน่วยความจำ VIEWING MEMORY LOCATION

แบบฝึกหัดชุดแรก ท่านจะต้องใช้ DOS DEBUG ในการแสดงข้อมูลของหน่วยความจำ โดยใช้โปรแกรม DEBUG ที่อยู่ในแผ่นดิสก์หรือ อยู่ในฮาร์ดดิสก์ ที่เก็บโปรแกรม DEBUG การใช้งานโปรแกรมนี้จะต้องพิมพ์คำว่า DEBUG และกด RETURN โปรแกรม DEBUG จะโหลดจากแผ่นดิสก์ไปที่หน่วยความจำ เมื่อเกิด PROMPT ของ DEBUG ซึ่งเป็นรูป (-) Hyphen บนจอภาพ ก็แสดงว่า DEBUG พร้อมที่จะใช้งานในการรับคำสั่งของ DEBUG ในการแสดงข้อมูลใช้คำสั่ง D (DUMP)

การตรวจสอบหน่วยความจำ CHECKING MEMORY SIZE

ขั้นตอนแรกของโปรแกรมจะทำการตรวจสอบจำนวนหน่วยความจำ โดยการใช้ DOS นั้นขึ้นอยู่กับโมเดลของคอมพิวเตอร์ ค่าของมันขึ้นอยู่กับการเซ็ทสวิทช์ภายในระบบ และอาจจะชี้ว่าหน่วยความจำน้อยกว่าความเป็นจริงก็ได้ ค่าของ ROM BIOS อยู่ที่ตำแหน่ง 413H และ 414H สามารถแสดงให้เห็นได้จากโปรแกรมดีบัก ในส่วนแอดเดรสจะแบ่งออกเป็น 2 ส่วน 400 คือเซคเม้นต์แอดเดรส ซึ่งพิมพ์เป็น 40 (0 ตัวสุดท้ายคือค่าสมมุติ) และ 13 เป็นค่า OFFSET จากเซคเม้นต์แอดเดรส

D 40:13 [กด ENTER]

ข้อมูล 2 ไบท์แรกแสดงจานวนกิโลไบท์ของหน่วยความจำในรูปเลขฐานสิบหก ซึ่งข้อมูลที่แสดงจะมีค่าสลับกัน ดังตัวอย่างที่แสดงค่าสลับกันของเลขฐานสิบหก

REVERSED HEX	CORRECTED HEX	DECIMAL(K)
0001	0100	256
8001	0180	384
0002	0200	512
8002	0280	640

CHECKING SERAIL NUMBER

คอมพิวเตอร์ที่จะแสดง Seria' number จะอยู่ในส่วนของ ROM ที่แอดเดรส FE00

D FE00:0 [กด ENTER]

จอภาพจะแสดงเลข 7 ตัว ตามด้วยวันที่อยู่ในรูปเลขฐานสิบหก

CHECKING ROM BIOS DATE

ตรวจสอบวันที่ของ ROM BIOS เริ่มที่แอดเดรส FFFF5H

D FFFF:05 [กด ENTER]

เมื่อเราทราบวันที่เราก็ทราบอายุของคอมพิวเตอร์และโมเดล

การกำหนดข้อมูลทันที MACHINE LANGUAGE EXAMPLE: IMMEDIATE DATA

เมื่อเราทราบถึงวิธีการป้อนโปรแกรมของดีบัก เข้าไปยังหน่วยความจำและจะทำการเอ็กซีคิวส์ที่ละคำสั่งโดยใช้คำสั่ง TRACE ตัวอย่างต่อไปนี้จะแสดงโปรแกรมภาษาเครื่องที่เก็บไว้ในหน่วยความจำหลักและแสดงผลของการเอ็กซีคิวส์ ข้อมูลในโปรแกรมดีบักจะอยู่ในรูปของเลขฐานสิบหก

INSTRUCTION	EXPLANATION
B82301	เซ็ทค่า 0123 ไว้ในรีจีสเตอร์ AX
052500	บวกค่า 0025 กับค่าใน AX
8BD8	บวกค่า AX กับค่า BX
8BCB	เก็บค่าของ BX ไปที่ CX
2BC8	ลบค่า AX จาก CX
2BC0	ลบข้อมูลของ AX จาก AX
90	ไม่มีการทำงาน
C3	กลับสู่ DOS

ท่านจะสังเกตุจากรหัสภาษาเครื่องอาจจะมีความยาว 1 , 2 หรือ 3 ไบท์ ไบท์แรกจะเป็นรหัสการทำงาน (Operation code) ส่วนไบท์อื่นๆเป็นตัวประกอบการทำงานของไบท์แรกหรือเรียกว่า Operand จะอ้างถึงข้อมูลในลักษณะ Immedaitely รีจีสเตอร์ หรือตำแหน่งในหน่วยความจำ ดังนั้นคำสั่งในภาษาเครื่องจะต้องอยู่ในหน่วยความจำก่อนการเอ็กซีคิวส์เริ่มต้น การเอ็กซีคิวส์จะเริ่มจากคำสั่งแรกและลำดับต่อไปที่ละคำสั่ง ตัวอย่างการเคลื่อนย้ายกรณีหนึ่งใช้รหัสภาษาเครื่อง B8H และการเคลื่อนย้ายอีกกรณีหนึ่งใช้รหัส 8BH

ท่านสามารถป้อนโปรแกรมโดยตรงในหน่วยความจำและทำการเอ็กซีคิวส์ทีละคำสั่งได้ ในเวลาเดียวกันท่านก็สามารถแสดงข้อมูลของรีจีสเตอร์ของการทำงานแต่ละคำสั่ง เริ่มต้นในการฝึกงานเพียงแต่ท่านโหลดโปรแกรมดีบัก โดยการพิมพ์คำว่า DEBUG และตามด้วย ENTER เมื่อดีบักโหลดเรียบร้อยแล้วจะปรากฏ Prompt (-) ก็สามารถเริ่มพิมพ์คำสั่งได้ การป้อนคำสั่งภาษาเครื่องโดยตรง ท่านต้องใช้คำสั่ง E ดังนี้

```
E CS:100 B8 23 01 05 25 00  [ENTER]
```

CS:100 เป็นตัวแอดเดรสเริ่มต้นของหน่วยความจำ ขณะที่ข้อมูลเก็บไว้เรียบร้อยแล้วที่ 100H(256) ของ Code Segment ส่วนคำสั่ง E ในดีบัก จะเก็บค่าของเลขฐานสิบหกที่ละคู่ในแต่ละไบท์ของหน่วยความจำ เริ่มต้นที่ CS:100 ถึง CS:106 คำสั่ง E จะเก็บข้อมูล 6 ไบท์เริ่มต้นที่แอดเดรส CS:106 ถึง 107 108,109,10A และ 10B

```
E CS:106 8B D8 03 D8 8B CB  [ENTER]
```

คำสั่ง E จะเก็บค่าทั้งหมด 6 ไบท์เริ่มต้นที่ CS:10C ถึง 10D,10E,10F,110 และ 111 ดังนี้

```
E CS:10C 2B C8 2B C0 90 C3  [ENTER]
```

ถ้าท่านป้อนข้อมูลผิดท่านสามารถทำซ้ำใหม่ได้ง่าย ดูจากตัวอย่างในรูป 2.2 แสดงถึงขั้นตอนในการใช้คำสั่ง E ดูจากจอภาพของท่านที่ป้อนข้อมูลผ่านดีบัก

คำสั่ง R ในดีบัก จะเป็นการแสดงข้อมูลของรีจีสเตอร์และแฟลก ดีบักจะแสดงข้อมูลของรีจีสเตอร์ในรูปของเลขฐานสิบหกมีรูปแบบดังนี้

```
AX = 0000   BX = 0000   CX = 0000 ......
```

เพราะว่าความแตกต่างของ DOS แต่ละรุ่น ข้อมูลในรีจีสเตอร์แสดงบนจอภาพในรูป 2.2 ใช้ DOS รุ่น 4.01 รีจีสเตอร์ IP มีค่าเป็น IP = 0100 เป็นตัวชี้คำสั่งแรกในการเอ็กซีคิวส์ของ Code Segment ส่วนในรูป 2.2 แฟลกรีจีสเตอร์ แสดงดังต่อไปนี้

```
                    NV  UP  EI  PL  NZ  NA  PO  NC
NV = NO OVERFLOW        UP = UP(RIGHT)      EI = ENABLE INTERRUPT
PL = PLUS SIGN          NZ = NON ZERO       NA = NO AUXILIARY
PO = PARITY ODD         NC = NO CARRY
```

คำสั่ง R จะแสดงค่าผลที่ค่าออฟเซ็ท 0100 ของคำสั่งแรก สังเกตุจากในรูป 2.2 รีจีสเตอร์ CS จะมีข้อมูล 21C1 แต่ค่าของรีจีสเตอร์ CS ไม่เปลี่ยนแปลง ดังที่แสดงต่อไปนี้ด้วยค่า XXXX

```
XXXX:0100 B82301  MOV  AX,0123
```

- XXXX เป็นตัวชี้เริ่มแรกของ CODE SEGMENT ที่ XXXX[0] ค่าของ XXXX:0100 หมายถึงเลขฐานสิบหก 100H ที่ตามหลังแอดเดรสของ CS คือ XXXX[0]
- B82301 เป็นรหัสภาษาเครื่องที่ป้อนเข้าในส่วนของ CS:100
- MOV AX,0123 เป็นคาสั่งภาษาแอสแซมบลีสำหรับภาษาเครื่อง คำสั่งนี้หมายถึงการเคลื่อนย้ายข้อมูล 0123 ทันทีเข้าไปไว้ในรีจีสเตอร์ AX

```
-E CS:100 B8 23 01 05 25 00
-E CS:106 8B D8 03 D8 8B CB
-E CS:10C 2B C8 2B C0 90 C3
-R
AX=0000  BX=0000  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=21C1  ES=21C1  SS=21C1  CS=21C1  IP=0100   NV UP EI PL NZ NA PO NC
21C1:0100 B82301        MOV     AX,0123
-T

AX=0123  BX=0000  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=21C1  ES=21C1  SS=21C1  CS=21C1  IP=0103   NV UP EI PL NZ NA PO NC
21C1:0103 052500        ADD     AX,0025
-T

AX=0148  BX=0000  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=21C1  ES=21C1  SS=21C1  CS=21C1  IP=0106   NV UP EI PL NZ NA PE NC
21C1:0106 8BD8          MOV     BX,AX
-T

AX=0148  BX=0148  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=21C1  ES=21C1  SS=21C1  CS=21C1  IP=0108   NV UP EI PL NZ NA PE NC
21C1:0108 03D8          ADD     BX,AX
-T

AX=0148  BX=0290  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=21C1  ES=21C1  SS=21C1  CS=21C1  IP=010A   NV UP EI PL NZ AC PE NC
21C1:010A 8BCB          MOV     CX,BX
-T

AX=0148  BX=0290  CX=0290  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=21C1  ES=21C1  SS=21C1  CS=21C1  IP=010C   NV UP EI PL NZ AC PE NC
21C1:010C 2BC8          SUB     CX,AX
-T

AX=0148  BX=0290  CX=0148  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=21C1  ES=21C1  SS=21C1  CS=21C1  IP=010E   NV UP EI PL NZ AC PE NC
21C1:010E 2BC0          SUB     AX,AX
-T

AX=0000  BX=0290  CX=0148  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=21C1  ES=21C1  SS=21C1  CS=21C1  IP=0110   NV UP EI PL ZR NA PE NC
21C1:0110 90            NOP
-T

AX=0000  BX=0290  CX=0148  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=21C1  ES=21C1  SS=21C1  CS=21C1  IP=0111   NV UP EI PL ZR NA PE NC
21C1:0111 C3            RET
-
```

รูป 2.2 Trace of Machine Instructions

การทำงานของคำสั่ง MOV หลังจากเปลี่ยนเป็นรหัสภาษาเครื่องแล้วจะได้ B8 และตามด้วย 2301 การทำงานจะเคลื่อนย้ายข้อมูล 23 ไบท์ที่ AL และข้อมูล 01 ไบท์ที่ AH

AX: |01|23|

โปรแกรมดีบักจะแสดงผลลัพธ์ของรีจีสเตอร์ และข้อมูลใน IP จะเป็น 0103 เพื่อชี้ค่าตำแหน่ง Offset ในส่วนของ Code Segment ของคำสั่งต่อไปที่นำมาเอ็กซีคิวส์

XXXX:0103 052500 ADD AX,0025

การเอ็กซีคิวส์คำสั่ง โดยการป้อนคำสั่ง T คำสั่ง ADD จะเป็นการบวกค่า 25 เข้ากับ AL ของรีจีสเตอร์ AX และค่า 00 เข้ากับ AH ผลของการบวก 0025 เข้ากับ AX จะได้ข้อมูลใหม่คือ 0148 และ IP จะมีค่า 0106 สำหรับการเอ็กซีคิวส์คำสั่งถัดไป

XXXX:0106 8BD8 MOV BX,AX

ท่านสามารถใช้คำสั่ง T ในการเคลื่อนย้ายข้อมูลของ AX ไปยัง BX ฉนั้น BX จะมีข้อมูลใหม่เป็น 0148 และ AX ยังคงมีค่าเดิมคือ 0148 ถ้าท่านต้องการเอ็กซีคิวส์คำสั่งเหล่านี้ใหม่ ให้ท่านรีเซ็ทค่า IP และเริ่มต้นการใช้คำสั่ง T ใหม่ โดยการป้อน R IP และตามด้วยค่า 100 เขียนดังนี้

R IP 100

ถึงแม้ว่าท่านจะกดคำสั่ง T สำหรับคำสั่งสุดท้าย NOP (No Operation) และคำสั่ง RET(Return) มันจะออกจากโปรแกรมดีบัก เราสามารถแสดงโปรแกรมภาษาเครื่องในส่วนของ Code Segment โดยใช้คำสั่ง D ดังนี้

D CS:100

โปรแกรมดีบักจะแสดงข้อมูล 16 ไบท์ จากซ้ายไปขวาในแต่ละบรรทัดทางด้านซ้าย ส่วนทางด้านขวาจะแสดงรหัส ASCII ของแต่ละไบท์ บรรทัดแรกของการแสดงผลที่ได้จากคำสั่ง D จะเริ่มที่ 00 และแทนข้อมูลในตำแหน่ง CS:100 ถึง CS:10F บรรทัดที่ 2 จะแทนข้อมูลของ CS:110 ถึง CS:11F ถึงแม้ว่าโปรแกรมของท่านจะสิ้นสุดที่ CS:111 คำสั่ง D ก็จะแสดงข้อมูล 8 บรรทัดจาก CS:100 ถึง CS:170 ตามรูป 2.3 แสดงข้อมูลของ DS,ES,SS,CS ในแอดเดรสเดียวกัน โปรแกรมดีบักจะแสดง

ทีละหนึ่งเซคเม้นต์ ถ้าคำสั่งและข้อมูลอยู่ในเซคเม้นต์อยู่ในเซคเม้นต์เดียวกัน มันก็จะมีการเก็บที่แตกต่างกัน ถ้าต้องการยกเลิกดีบัก ให้ใช้คำสั่ง Q การทำงานนี้จะกลับเข้าสู่ DOS ท่านสามารถพิมพ์ข้อมูลบนจอภาพโดยใช้คำสั่งพิมพ์จอภาพได้คือ Ctrl/PrtSc

```
-D CS:100
21C1:0100  B8 23 01 05 25 00 8B D8-03 D8 8B C8 2B C8 7B C0  .#..%.......+.{.
21C1:0110  90 C3 8D 46 14 50 51 52-FF 76 28 E8 74 00 8B E5  ...F.PQR.v(.t...
21C1:0120  B8 01 00 50 FF 76 32 FF-76 30 FF 76 2E FF 76 28  ...P.v2.v0.v..v(
21C1:0130  E8 86 15 8B E5 FF 36 18-12 FF 36 16 12 8B 76 28  ......6...6...v(
21C1:0140  FF 74 3A 89 46 06 E8 22-CE 8B E5 30 E4 3D 0A 00  .t:.F..".....0.=..
21C1:0150  75 32 A1 16 12 2D 01 00-8B 1E 18 12 83 DB 00 53  u2...-.........S
21C1:0160  50 8B 76 28 FF 74 3A A3-16 12 89 1E 18 12 E8 FA  P.v(.t:.........
21C1:0170  CD 8B E5 30 E4 3D 0D 00-74 0A 83 06 16 12 01 83  ...0.=..t.......
```

รูป 2.3 Dump of the Code Segment

การกำหนดข้อมูลและโปรแกรมภาษาเครื่อง MACHINE LANGUAGE EXAMPLE: DEFINED DATA

ตัวอย่างการใช้ข้อมูล Immedaite ในการกำหนดข้อมูลโดยตรงให้กับคำสั่ง MOV และ ADD ตัวอย่างต่อไปเป็นการกำหนดข้อมูล 0123 และ 0025 ให้กับโปรแกรม โปรแกรมจะเข้าถึงข้อมูลในหน่วยความจำของค่าเหล่านี้

DS Offset	Hex Value	Bytes Occupied
0200	2301	0 and 1
0202	2500	2 and 3
0204	0000	4 and 5
0206	2A2A2A	6,7, and 8

อย่าลืมว่าตัวอักขระเลขฐานสิบหกมีขนาดครึ่งไบท์ ดังนั้นตัวอย่าง 23 ที่เก็บไว้ในไบท์ 0 (ไบท์แรก) ของพื้นที่ข้อมูล และ 01 ก็เก็บขนาด 1 ไบท์ ไบท์ที่ 2 รหัสคำสั่งภาษาเครื่องจะทำงานต่อไปนี้

Instruction	Explanation
A10002	การเคลื่อนย้ายข้อมูล 1 เวิร์ด ที่ DS แอดเดรสเริ่มต้น 0200 เก็บในรีจีสเตอร์ AX
03060202	การบวกข้อมูล 1 เวิร์ด ของ DS ที่แอดเดรส 0202 กับ รีจีสเตอร์ AX
A30402	การเคลื่อนย้ายข้อมูลของรีจีสเตอร์ AX ไปยัง DSที่ตำแหน่ง 0204
C3	Return to DOS

ท่านจะสังเกตุว่าคำสั่ง MOV จะมีรหัสภาษาเครื่องที่แตกต่างกัน A1 และ A3 การทำงานของคำสั่งที่ใช้ในภาษาเครื่อง รีจีสเตอร์แต่ละตัวจะเป็นอิสระในการอ้างถึง ขนาดของไบท์ข้อมูล ทิศทางการเคลื่อนย้ายข้อมูล (ไปหรือมาของรีจีสเตอร์) และอ้างถึงข้อมูลหรือหน่วยความจำ ท่านสามารถใช้รับรแกรมดีบักตรวจดูการทำงาน เมื่อพบสัญญานดีบักในลักษณะ HYPHEN (-) ท่านสามารถใช้คำสั่ง E สำหรับ DATA SEGMENT

```
E DS:0200 23 01 25 00 00 00  [ENTER]
E DS:0206 2A 2A 2A  [ENTER]
```

คำสั่ง E เป็นการเก็บค่า 3 เวิร์ด (6 ไบท์) ที่จุดเริ่มต้นของพื้นที่ข้อมูล DS:0200 สังเกตุจากการป้อนข้อมูลเป็นชนิดเวิร์ด ท่านจะต้องป้อนข้อมูลสลับกัน คือ 0123 เป็น 2301 และข้อมูล 0025 เป็น 2500 เมื่อคำสั่ง MOV เข้าถึงเวิร์ดเหล่านี้ในรีจีสเตอร์ มันจะได้ข้อมูลถูกต้องคือ 2301 จะกลายเป็น 0123 และ 2500 เป็น 0025

คำสั่ง E คำสั่งที่ 2 จะเก็บค่า ASTERISKS (***) ท่านสามารถแสดงดูข้อมูลภายหลังได้โดยใช้คำสั่ง D (DUMP) ต่อไปนี้เราป้อนคำสั่งลงใน CODE SEGMENT ที่ CS:0100 ดังนี้

```
E CS:100 A1 00 02 03 06 02 02  [ENTER]
E CS:107 A3 04 02 C3 [ENTER]
```

จากรูป 2.4 แสดงขั้นตอน รวมทั้งการใช้คำสั่ง E จอภาพของท่านจะแสดงผลลัพธ์ ผ่านแอดเดรสของ CS และ DS ตรวจสอบการเก็บข้อมูลที่ DS:200 ถึง 208 และตรวจสอบคำสั่งที่ CS:100 ถึง 10A โดยใช้คำสั่ง D

```
-E DS:200 23 01 25 00 00 00
-E DS:206 2A 2A 2A
-E CS:100 A1 00 02 03 06 02 02
-E CS:107 A3 04 02 C3
-D DS:200,208
21C1:0200  23 01 25 00 00 00 2A 2A-2A                          #.%...***
-R
AX=0000  BX=0000  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=21C1  ES=21C1  SS=21C1  CS=21C1  IP=0100   NV UP EI PL NZ NA PO NC
21C1:0100 A10002        MOV     AX,[0200]                      DS:0200=0123
-T

AX=0123  BX=0000  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=21C1  ES=21C1  SS=21C1  CS=21C1  IP=0103   NV UP EI PL NZ NA PO NC
21C1:0103 03060202      ADD     AX,[0202]                      DS:0202=0025
-T

AX=0148  BX=0000  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=21C1  ES=21C1  SS=21C1  CS=21C1  IP=0107   NV UP EI PL NZ NA PE NC
21C1:0107 A30402        MOV     [0204],AX                      DS:0204=0000
-T

AX=0148  BX=0000  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=21C1  ES=21C1  SS=21C1  CS=21C1  IP=010A   NV UP EI PL NZ NA PE NC
21C1:010A C3            RET
-D DS:0200,0208
21C1:0000  23 01 25 00 48 01 2A 2A-2A                          #.%.H.***
-Q
```

รูป 2.2 Trace of Machine Instructions

ดูข้อมูล D DS:200,208 [ENTER]
ดูคำสั่ง D CS:100,10A [ENTER]

ตรวจสอบดูการเก็บข้อมูลของเซคเม้นต์ทั้ง 2 จะได้ตามรูป 2.4 ท่านสามารถตรวจสอบการทำงานโดยใช้คำสั่ง R เพื่อแสดงดูข้อมูลในรีจีสเตอร์และผลที่เกิดขึ้นในแฟลก

XXXX:0100 A1002 MOV AX,[0200]

CS:0100 จะอ้างถึงคำสั่งแรก A10002 โปรแกรมดีบักจะทำการแปลคำสั่ง MOV และอ้างถึงข้อมูลของหน่วยความจำใน DS (DATA SEGMENT) ที่แอดเดรส [0200] วงเล็บสี่เหลี่ยมเป็นการอ้างอิงแอดเดรสของหน่วยความจำ ไม่ใช้ข้อมูลที่ป้อนให้กับคำสั่ง ถ้าเป็นข้อมูลที่ป้อนให้กับคำสั่งจะไม่มีวงเล็บสี่เหลี่ยม เช่น MOV AX,0200

คำสั่ง T ที่ใช้ในโปรแกรมดีบัก สังเกตุจากการทำงานของคำสั่ง MOV AX,[0200] เป็นการเคลื่อนย้ายข้อมูลจาก DS ที่แอดเดรส 200 มายังรีจีสเตอร์ AX ข้อมูลที่แอดเดรส 0200 คือ 2301 ซึ่งการทำงาน

จะกลับข้อมูล AX คือ 0123 ส่วนการป้อนคำสั่ง T ต่อไปจะเป็นการเอ็กซีคิวส์คำสั่งที่อยู่ถัดไป คือคำสั่ง ADD ซึ่งเป็นการทำงานในการบวกข้อมูลของเวิร์ดข้อมูลในหน่วยความจำที่แอดเดรส 0202 เข้ากับ AX ผลลัพธ์จะเก็บไว้ในรัจีสเตอร์ AX ผลรวมจะเป็น 0123 และ 0025 หรือ 0148

คำสั่งต่อไปคือ MOV [0240],AX คำสั่งนี้จะทำงานโดยใช้คำสั่ง T ของโปรแกรมดีบัก สำหรับการเอ็กซีคิวส์ คำสั่งนี้จะเป็นการทำงานในการย้ายข้อมูลของ AX ไปยังหน่วยความจำที่แอดเดรส DS:0204 แล้วแสดงการเปลี่ยนข้อมูลในแอดเดรส 200 - 208

```
D DS:200,208 [ENTER]
```

ข้อมูลต่างๆจะแสดงดังนี้

```
ค่าใน    DS: 23  01  25  00  48  01  2A  2A  2A
             ¦   ¦   ¦   ¦   ¦   ¦   ¦   ¦   ¦
ค่า  OFFSET  200 201 202 203 204 205 206 207 208
```

ข้อมูล 1048 จะถูกเคลื่อนย้ายจาก AX ไปยังหน่วยความจำ DS ที่แอดเดรส 204 และแอดเดรส 205 ข้อมูลจะเก็บในหน่วยความจำสลับกันคือ 4810 ส่วนคำสั่ง Q เป็นการสิ้นสุดการทำงานของโปรแกรม DEBUG เพื่อกลับสู่ DOS

ขนาดของหน่วยความจำและโปรแกรมภาษาเครื่อง

แบบฝึกหัดแรกของบทนี้ ให้ท่านใช้โปรแกรมดีบักในการตรวจสอบตำแหน่ง 413 และ 414 ที่อยู่ในหน่วยความจำของ RAM ข้อมูลในคอมพิวเตอร์ในส่วนของ ROM BIOS เป็นโปรแกรมในการตรวจสอบขนาดของหน่วยความจำ การเข้าถึงข้อมูลในส่วนของ BIOS นั้นจะต้องผ่านทางคำสั่ง INT (INTERRUPT) ในกรณีใช้ INT 12H คำสั่งของ BIOS จะเก็บค่าเข้าใน AX การโหลดดีบักลงในหน่วยความจำ และป้อนคำสั่งภาษาเครื่อง สำหรับการใช้ INT 21H และ คำสั่ง RET ดังนี้

```
E CS:100 CD 12 C3 [ENTER]
```

การพิมพ์คำสั่ง R และตามด้วย ENTER เป็นการแสดงข้อมูลในรีจีสเตอร์ จะได้ค่า AX=0000 IP = 0100 คำสั่ง INT 12H จะเป็นทางผ่านการควบคุมเข้าสู่ BIOS เพื่อส่งขนาดของหน่วยความจำไปที่ AX และกดอักษร T ตามด้วย ENTER จะเห็นการทำงานของคำสั่ง BIOS โปรแกรมดีบักแสดงคำสั่งอยู่ในรูปของสัญญลักษณ์ ถึงแม้จะมีการเอ็กซีคิวส์เป็นภาษาเครื่อง

```
STI                 ;SET INTERRUPT
PUSH   DS           ;SAVE DS ADDRESS IN STACK
MOV    AX,0040      ;SEGMENT  40[0]
MOV    DS,AX        ;PLUS
MOV    AX,[0013]    ;OFFSET 0013
POP    DS           ;RESTORE ADDRESS IN DS
IRET                ;RETURN FROM INTERRUPT
```

ณ.จุดนี้ข้อมูลบอกขนาดของหน่วยความจำใน AX จะอยู่ในรูปของเลขฐานสิบหก ถ้าเราป้อนคำสั่ง T เพื่อออกจาก BIOS เพื่อกลับสู่โปรแกรมเดิมของท่าน โดยใช้คำสั่ง RET สำหรับภาษาเครื่องคือ C3

ดีบัก DEBBUG FEATURES

ท่านสามารถใช้ดีบัก ในการเขียนคำสั่งภาษาแอสแซมบลี หรือคำสั่งภาษาเครื่อง ท่านจะต้องใช้คำสั่งดีบักต่อไปนี้

THE A COMMAND

คำสั่ง A (ASSEMBLE) เป็นการบอกดีบักถึงจุดเริ่มต้นในการรับข้อมูลที่เป็นสัญญลักษณ์ภาษาแอสแซมบลี และเปลี่ยนให้อยู่ในรูปของภาษาเครื่อง การกำหนดแอดเดรสเริ่มแรกของโปรแกรมดีบักมีดังนี้

A 100 [ENTER]

โปรแกรมดีบักจะแสดงค่าของ CODE SEGMENT และค่าของ OFFSET คือ XXXX:0100 ของแต่ละคำสั่งที่ป้อนเข้ามา เมื่อท่านป้อนโปรแกรมแต่ละคำสั่งและกด ENTER ถ้ากด ENTER สองครั้งก็จะเป็นการออกจากโปรแกรมดีบัก ดังนี้

```
MOV   AL,20   [ENTER]
MOV   BL,32   [ENTER]
MOV   AL,BL   [ENTER]
RET    [ENETR]
```

จอภาพจะแสดงผลลัพธ์ดังนี้

```
XXXX:0100    MOV    AL,25
XXXX:0102    MOV    BL,32
XXXX:0104    ADD    AL,BL
XXXX:0106    RET
```

THE U COMMAND

คำสั่ง U (UNASSEMBLE) เป็นการแสดงคำสั่งภาษาเครื่องพร้อมกับภาษาแอสแซมบลี้ ท่านจะต้องบอกดีบักถึงตำแหน่งแอดเดรสแรกและแอดเดรสสุดท้าย ดังตัวอย่าง

```
U 100,106    [ENTER]
```

จอภาพจะแสดงผลดังนี้

```
XXXX:0100    B025    MOV   AL,25
XXXX:0102    B332    MOV   BL,32
XXXX:0104    00D8    ADD    AL,BL
XXXX:0106    C3      RET
```

จากตัวอย่างโปรแกรมเราจะเอ็กซีคิวส์ 3 คำสั่ง โดยใช้คำสั่ง R แสดงข้อมูลของรีจีสเตอร์ และเอ็กซีคิวส์คำสั่งแรกใช้คำสั่ง T ไปที่ละคำสั่ง

SAVING A PROGRAM FROM WITH IN DEBUG

การเขียนโปรแกรมด้วยดีบัก จะมีการเก็บโปรแกรมลงในแผ่นดิสก์ได้ 2 วิธี

1. ท่านจะอ่านโปรแกรม แก้ไขโปรแกรม และต้องการเก็บโปรแกรมมีขั้นตอนดังนี้
   - อ่านโปรแกรมภายใต้ชื่อดังนี้ DEBUG n: FILENAME
   - คำสั่ง D แสดงรายการของโปรแกรม และคำสั่ง E เพื่อแก้ไข
   - คำสั่ง W ในการเขียนโปรแกรมลงในแผ่นดิสก์

2. ท่านจะใช้ดีบัก เขียนโปรแกรมขนาดเล็กๆ แล้วท่านจะต้องการเก็บโปรแกรมมีขั้นตอนดังนี้

- ใช้โปรแกรม DEBUG
- คำสั่ง A (ASSEMBLE) และคำสั่ง E ในการสร้างโปรแกรม
- ชื่อโปรแกรม ใช้คำสั่ง N FILENAME.COM โปรแกรมที่เขียนภายใต้ดีบักต้องใช้นามสกุล .COM
- ท่านจะต้องรู้จุดสุดท้ายของโปรแกรม แล้วบอกดีบักถึงความยาวของโปรแกรมมีกี่ไบท์ ดังตัวอย่าง

```
XXXX: 0106    C3    RET
```

คำสั่งสุดท้ายขนาด 1 ไบท์ ฉนั้นโปรแกรมมีขนาดแอดเดรสตั้งแต่ 100 - 106 หรือมีขนาด 7 ไบท์

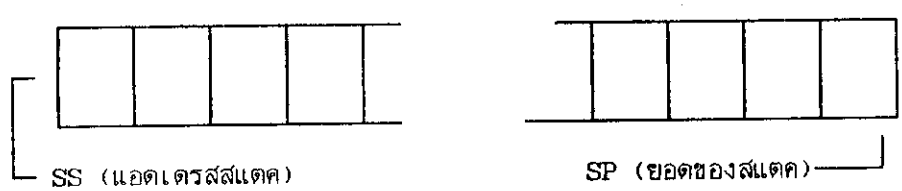
- ครั้งแรกจะใช้ CX โดยใช้คำสั่ง R CX [enter]
- โปรแกรมดีบักจะให้ค่า CX = 0000
- ท่านจะเห็นความยาวโปรแกรมใน CX เท่ากับ 7 ไบท์
- เขียนโปรแกรมโดยใช้คำสั่ง W [enter]

สำหรับทั้ง 2 วิธี โปรแกรมดีบักจะแสดงข่าวสารในการเขียนว่า "WRITING nnnn BYTES" ถ้าค่าเป็น 0 แสดงว่าผิดพลาด ต้องใส่ความยาวโปรแกรมแล้วพยายามอีกครั้ง ดูรายละเอียดเพิ่มเติมจากโปรแกรมดีบักในภาคผนวก

THE STACK

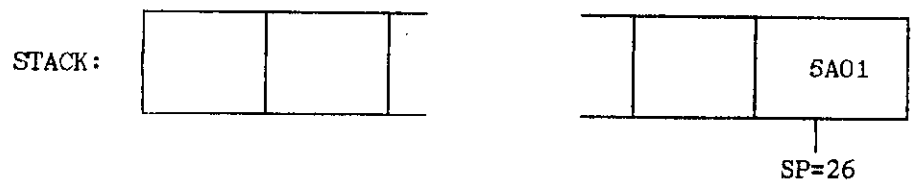
โปรแกรมที่สามารถเอ็กซีคิวส์มีนามสกุลอยู่ 2 ชนิดคือ NAME.COM และ NAME.EXE ซึ่งทั้ง 2 ชนิดต้องการพื้นที่สำรองในการเก็บค่าลงในสแตค จุดประสงค์ของการใช้สแตคคือการสำรองพื้นที่ไว้ชั่วคราวสำหรับการเก็บข้อมูลและแอดเดรส

ท่านจะต้องกำหนดค่าของสแตคของโปรแกรมนามสกุล .EXE ขณะที่ DOS กำหนดค่าสแตคของโปรแกรมนามสกุล COM รายการข้อมูลแต่ละรายการในสแตคคือขนาด 1 เวิร์ด (2 ไบท์) รีจีสเตอร์ SS จะกำหนดค่าเริ่มแรกโดย DOS เป็นข้อมูลเริ่มต้นแอดเดรสแรกของสแตคและค่าที่อยู่ใน SP เป็นค่า OFFSET ชี้ที่ยอดของสแตค

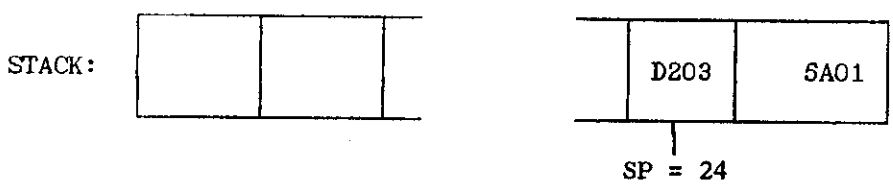


คำสั่ง PUSH ,CALL และ INT จะเป็นการลดค่า SP ที่ละ 2 เพื่อชี้หน่วยความจำแอดเดรสต่ำถัดไปในสแตค และเก็บค่าในสแตค ส่วนคำสั่ง POP,RET และ IRET จะเพิ่มค่า SP ที่ละ 2 และดึงข้อมูลจากสแตค ดังตัวอย่างคำสั่ง PUSH AX และค่า BX ลงในสแตค การใช้คำสั่ง POP จะนำมาเก็บไว้ที่เดิม สมมุติว่า AX = 015A และ BX = 03D2 และ SP มีค่า 28

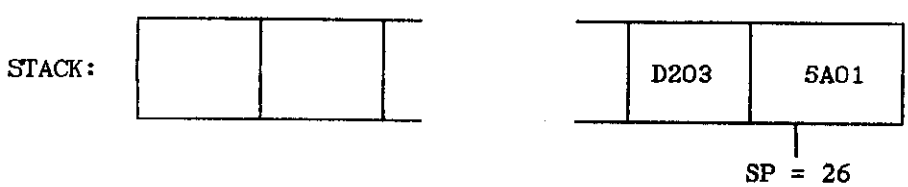
1) การ PUSH ค่า AX คำสั่ง PUSH AX จะลดค่า SP ลง 2 และเก็บข้อมูล AX =015A ลงในสแตค สังเกตุจากค่าที่เก็บในสแตคจะเก็บค่าสลับกัน



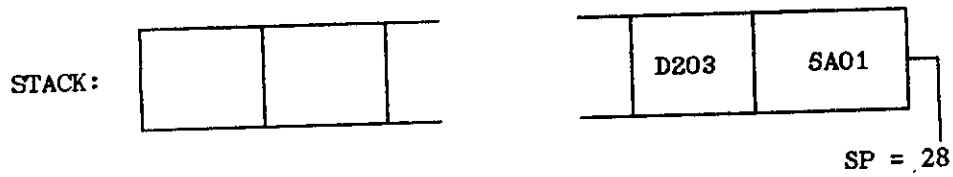
2) การ PUSH BX คำสั่ง PUSH BX จะลดค่า SP ลง 2 และเก็บข้อมูลของ BX 03D2 ลงในสแตค



3) การ POP BX คำสั่ง POP BX เป็นการดึงเวิร์ดข้อมูลจากหน่วยความจำสแตคที่กำหนดแอดเดรสโดย SP ไปเก็บไว้ใน BX และเพิ่มค่า SP ขึ้น 2 และ BX จะมีค่า 03D2 ดึงค่าออกมาสลับกัน



4) การ POP AX คำสั่ง POP AX เป็นการดึงเวิร์ดข้อมูลจากหน่วยความจำสแตคที่แอดเดรส กำหนดโดย SP ไปเก็บไว้ใน AX และเพิ่มค่า SP ขึ้น 2 และ AX มีค่า 015A



คำสั่งอินเตอร์รัพท์ THE INTERRUPT INSTRUCTION INT

คำสั่ง INT จะเป็นการประมวลผลในการเข้าถึงการบริการอินเตอร์รัพท์ (INTERRUPT SERVICE TABLE) การทำงานเป็นการเคลื่อนย้ายไป DOS หรือ BIOS สำหรับกำหนดการทำงาน และย้อนกลับมาโปรแกรมเดิมเพื่อทำงานต่อไป คำสั่ง INT จะถูกใช้งานบ่อยๆเกี่ยวกับการทำงานของ INPUT OUTPUT คำสั่ง INT มีรูปแบบการทำงานดังต่อไปนี้

- ลดค่า SP ลง 2 และ PUSH ค่าแฟลกลงในสแตค
- เคลียอินเตอร์รัพท์และ TRAP FLAGS
- ลดค่า SP ลง 2 และ PUSH ค่า CS ลงในสแตค
- ลดค่า SP ลง 2 และ PUSH ค่า DS ลงในสแตค
- ทำงานตามความต้องการ

การย้อนกลับจากคำสั่ง INT โดยใช้คำสั่ง IRET ซึ่งเป็นการ POP ค่ารีจีสเตอร์จากสแตค ไว้ตามเดิมทั้งหมด

บทสรุป

- การจ่ายกำลังไฟฟ้าให้เครื่องเราเรียกว่า COLD BOOT โปรเซสเซอร์จะเริ่มต้นทำงานในการเคลียแอดเดรสของหน่วยความจำไปที่แอดเดรส 0 รูปแบบของการตรวจเช็คหน่วยความจำ และเซ็ตค่ารีจีสเตอร์ CX และรีจีสเตอร์ IP ชี้ตำแหน่งของ BIOS ในหน่วยความจำชนิด ROM
- โปรเซสเซอร์จะทำการเฟทช์แต่ละไบท์ของคำสั่งและเพิ่มค่าของ IP เพื่อชี้คำสั่งต่อไป
- DOS DEBUG เป็นโปรแกรมที่ใช้ในการแก้ไขภาษาแอสแซมบลี

- หน่วยความจำสแตคใช้เป็นหน่วยความจำชั่วคราวสำหรับเก็บแอดเดรสของข้อมูล การเก็บข้อมูลในสแตคจะเก็บทีละเวิร์ด
- คำสั่ง INT ใช้ในการทำงานร่วมกับ DOS หรือ BIOS สำหรับกำหนดการทำงาน
- ถ้าท่านป้อนข้อมูลผิดให้ DATA SEGMENT หรือ CODE SEGMENT เราสามารถแก้ไขข้อมูลให้ถูกได้ การเอ็กซีคิวส์คำสั่งต้องเอ็กซีคิวส์คำสั่งแรก โดยการเซ็ตค่าของ IP และรอจนกว่าเราจะป้อนข้อมูล เราป้อนค่า 0100 ตามด้วย ENTER การตรวจสอบผลลัพธ์ใช้คำสั่ง R ดีบักจะแสดงค่าของรีจีสเตอร์ทั้งหมดและแฟลก คำสั่งแรกที่จะนำมาเอ็กซีคิวส์ใช้คำสั่ง T ซึ่งเป็นการเอ็กซีคิวส์ทีละคำสั่ง ผลลัพธ์อยู่ใน AX

แบบฝึกหัด

2-1. อธิบายถึงจุดประสงค์ในการใช้งานของสแตค

2-2. ในช่วงของการเอ็กซีคิวส์โปรแกรม ถ้า CS = 5A2B[0] และ SS = 5B53[0] ค่าของ IP=52 และ SP = 48 จงคำนวนหาค่าแอดเดรสต่อไปที่ต้องการเอ็กซีคิวส์ และตำแหน่งปัจจุบันของสแตค

2-3. ข้อมูลของ DS = 5B24[0] และคำสั่งที่เคลื่อนย้ายข้อมูลจากหน่วยความจำไป AL คือ A03A01 จงคำนวนหาค่าแอดเดรสในหน่วยความจำ

2-4. สมมุติว่าท่านใช้โปรแกรมดีบักในการป้อนข้อมูลต่อไปนี้

E CS:100 B8 04 30 05 00 30 C3

a. 3 คำสั่งแรกหมายถึงคำสั่งอะไร

b. การเอ็กซีคิวส์โปรแกรม AX จะมีค่าอะไร

c. ถ้าท่านป้อนคำสั่งผิดให้แก้ไขแล้วเริ่มต้นเอ็กซีคิวส์ใหม่

2-5. สมมุติว่าท่านใช้โปรแกรมดีบักในการป้อนข้อมูลต่อไปนี้

E CS:100 B8 45 01 05 25 00

ถ้าค่าเลขฐานสิบหก 45 เป็น 54 รหัสที่ป้อนด้วยคำสั่ง E เราสามารถแก้ไขข้อมูลที่ผิดได้อย่างไร

2-6. โปรแกรมภาษาเครื่องต่อไปนี้

B0 25 D0 E0 B3 15 F6 E3 C3

การทำงานของโปรแกรมมีดังต่อไปนี้

- เคลื่อนย้ายค่า 25H ไปที่ AL
- เลื่อนข้อมูล AL ไปทางซ้าย 1 บิต
- เคลื่อนย้ายค่า 15H ไป BL

- คูณ AL ด้วย BL

ใช้โปรแกรมดีบักด้วยการใช้คำสั่ง E ป้อนโปรแกรมที่ D CS:100 แสดงข้อมูล แล้วคีย์คำสั่ง R ตามด้วยคำสั่ง T เพื่อแสดงการทำงานของโปรแกรมจนถึงคำสั่ง RET อยากทราบว่าผลลัพธ์ครั้งสุดท้ายใน AX คืออะไร

2-7. โปรแกรมดีบักในการใช้คำสั่ง A ป้อนคำสั่งต่อไปนี้

```
MOV  BX,25
ADD  BX,30
SHL  BX,01
SUB  BX,22
NOP
RET
```

2-8. จงอธิบายถึงคำสั่ง PUSH , CALL , INT มีผลต่อสแตคอย่างไร

2-9. จงอธิบายคำสั่ง POP , RET , IRET มีผลต่อสแตคอย่างไร

2-10. โปรแกมที่นามสกุล COM กับนามสกุล EXE ต่างกันอย่างไร
