บทที่ 1

โครงสร้างสถาปัตยกรรมของคอมพิวเตอร์

PC ARCHITECTURE

วัตถุประสงค์

หลังจากที่ท่านศึกษาบทนี้ท่านจะมีความเข้าใจดังต่อไปนี้

- โครงสร้างฮาร์ดแวร์ของไมโครคอมพิวเตอร์
- การจัดองค์กรของโปรแกรม
- รีจีสเตอร์ต่างๆที่เกี่ยวข้องกับการเขียนโปรแกรม
- การคำนวนทางคณิตศาสตร์ของคอมพิวเตอร์
- การจัดการหน่วยความจำ
- การกำหนดแอดเดรสในการเข้าถึงข้อมูลของหน่วยความจำ

บทที่ 1
โครงสร้างสถาปัตยกรรมของคอมพิวเตอร์
PC ARCHITECTURE

บทนำ

การเขียนโปรแกรมภาษาแอสแซมบลี จะต้องมีความรู้เกี่ยวกับชุดคำสั่งของคอมพิวเตอร์ กฎเกณฑ์การใช้ชุดคำสั่ง และฮาร์ดแวร์ที่เกี่ยวข้องกับคำสั่งของคอมพิวเตอร์และการพัฒนาการเขียนคำสั่ง หนังสือเล่มนี้จะเริ่มต้นอธิบายพื้นฐานของฮาร์ดแวร์ (โครงสร้างสถาปัตยกรรม) บิต ไบต์ รีจีสเตอร์ หน่วยความจำ ตัวประมวลผล และบัสข้อมูล

โครงสร้างพื้นฐานของระบบคอมพิวเตอร์คือ บิต ไบต์ พื้นฐานเหล่านี้จะเป็นตัวกำหนดความหมายของคอมพิวเตอร์ ซึ่งสามารถแทนข้อมูลและคำสั่งในหน่วยความจำ

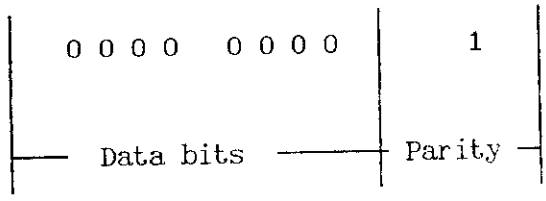
ฮาร์ดแวร์หลักคือ หน่วยความจำภายใน ไมโครโปรเซสเซอร์ และรีจีสเตอร์ คอมพิวเตอร์จะต้องเก็บโปรแกรมไว้ภายในหน่วยความจำหลักที่อยู่ภายในสำหรับการเอ็กซีคิวส์ ไมโครโปรเซสเซอร์จะทำการเอ็กซีคิวส์คำสั่งในโปรแกรมร่วมทำงานกับรีจีสเตอร์ในหน่วยคำนวนทางคณิตศาสตร์ การเคลื่อนย้ายข้อมูลและการกำหนดแอดเดรสของหน่วยความจำ

การเขียนโปรแกรมภาษาแอสแซมบลีประกอบไปด้วยหนึ่งหรือมากกว่าหนึ่งเซคเม้นต์สำหรับการกำหนดข้อมูล สำหรับคำสั่งภาษาเครื่อง และการกำหนดชื่อของเซคเม้นต์

บิต และ ไบท์ (BITS AND BYTES)

บิตเป็นหน่วยความจำที่เล็กที่สุดในการเก็บข้อมูลของคอมพิวเตอร์ บิตหมายถึงอำนาจแม่เหล็กที่เป็น (OFF) ซึ่งมีค่าเป็น 0 หรือ ON ซึ่งมีค่าเป็น 1 เราสามารถนำกลุ่มของบิตมาเป็นข้อมูลหรือข่าวสารได้

กลุ่มของบิต 9 บิตเรานำมารวมกันเราเรียกว่า ไบท์ (BYTE) คือ 8 บิตสำหรับข้อมูลและอีก 1 บิต สำหรับบิตตรวจสอบ (PARITY BIT)



บิตข้อมูลจำนวน 8 บิตซึ่งเป็นพื้นฐานของหน่วยคำนวนทางคณิตศาสตร์ ซึ่งนำมาแทนตัวอักขระ เช่น ตัวอักษร A และสัญญลักษณ์ * กลุ่มของบิต 8 บิตสามารถแทนการกำหนดค่าที่แตกต่างกันได้ 256 ชนิด ในเงื่อนไขของการ ON/OFF จากค่า OFF ทั้งหมดคือ 00000000 จนกระทั่งเป็นค่า ON คือ 11111111 ตัวอย่างการแทนค่าตัวอักษร A ด้วยค่า ON และ OFF คือ 0100 0001 และค่า * คือ (Asterisk) 0010 1010 ซึ่งเป็นข้อมูลขนาด 1 ไบท์เก็บไว้ในหน่วยความจำ

บิตตรวจสอบหรือเรียกว่าพาริตี้บิตมีขนาด 1 บิตสำหรับข้อมูลแต่ละไบท์ พาริตี้บิตจะต้องกำหนดโดยอัตโนมัติที่เป็น ODD PARITY ฉนั้นตัวอักษร A มีเลข 1 จำนวน 2 ตัว โปรเซสเซอร์จะเซ็ทค่าพาริตี้บิตเป็น 1 คือ 0100 0001-1 และข้อมูลของ Asterrisk ซึ่งมีเลข 1 จำนวน 3 ตัวโปรเซสเซอร์จะเซ็ทพาริตี้บิตเป็น 0 ดังนี้ 0010 1010-0 เมื่อคำสั่งมีการอ้างถึงข้อมูลในหน่วยความจำ คอมพิวเตอร์จะทำการตรวจสอบพาริตี้บิต ถ้ากำหนดพาริตี้บิตเป็นแบบ EVEN PARITY การตรวจสอบแบบคู่ ข้อมูลที่กำหนดก็เกิดการผิดพลาด ข้อผิดพลาดของพาริตี้บิตอาจมีผลลัพธ์มาจากสัญญานทางไฟฟ้าได้

ท่านจะมีความเข้าใจการทำงานของคอมพิวเตอร์ได้ดี ถ้าท่านเข้าใจพื้นฐานของคำว่า บิต หมายถึงอะไร ถ้าท่านพิมพ์ตัวอักษร A จากคีย์บอร์ด ผลลัพธ์ที่เก็บในหน่วยความจำแต่ละบิตของตัวอักษร A คือ 0100 0001 ระบบคอมพิวเตอร์ก็จะรับสัญญานจากการทำงานของคีย์บอร์ดไปเก็บไว้ในหน่วยความจำ แล้วทำการเซ็ทข้อมูลที่รับเข้ามาเป็น 0100 0001 แล้วก็นำข้อมูลที่รับเข้ามาที่อยู่ในหน่วยความจำไปแสดงผลที่จอภาพ เป็นตัวอักษร A

ตำแหน่งของข้อมูลแต่ละบิตขนาด 1 ไบท์ของตัวอักษร A มีการกำหนดตำแหน่งดังนี้

หมายเลขบิต	7	6	5	4	3	2	1	0
ข้อมูลบิตค่า A	0	1	0	0	0	0	0	1

ค่าของ 2 ยกกำลัง 10 มีค่าเท่ากับ 1024 ซึ่งหมายถึงค่า K หรือเรียกว่า กิโลไบท์ (KILOBYTE) ถ้าขนาดของคอมพิวเตอร์ 640 K หมายถึง 640x1024 = 65535 ไบท์

เลขขนาด 16 บิต หรือ 2 ไบท์ เราเรียกว่าเวิร์ด (WORD) และขนาด 32 บิตเราเรียกว่า 2 เวิร์ด (DOUBLEWORD) ตำแหน่งของบิตในแต่ละเวิร์ดเหล่านี้จะมีหมายเลขบิต 0 - 15 จากซ้ายไปขวา ที่ใช้ในการแทนตัวอักษร PC ดังนี้

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0	1	1
			P								C				

ระบบเลขฐานสอง (BINARY SYSTEM)

คอมพิวเตอร์จะรับข้อมูลที่เป็น 0 และ 1 ซึ่งลักษณะเหมือนกับระบบเลขฐานสอง หรือเรียกว่าระบบไบนารี (BINARY) ค่าของบิตที่ใช้แทนในแต่ละเวิร์ดจะอยู่ในรูปของเลขฐานสอง กลุ่มของเลขฐานสองจะนำมารวมกันเพื่อใช้แทนค่าต่างๆ เพราะแต่ละบิตจะมีค่าเป็น 1 หรือ 0 ก็ได้ ถ้าค่าของเลขฐานสองจำนวน 8 บิต ที่มีค่าเป็น 1 ทั้งหมดจะได้ค่าดังนี้

ค่าของตำแหน่ง	128	64	32	16	8	4	2	1
บิต	1	1	1	1	1	1	1	1

บิตขวามือสุดเป็นบิตที่มีค่าน้อยที่สุดซึ่งมีค่าเท่ากับ 1 บิตถัดมาจะมีค่าเป็น 2 และมีค่าเป็น 4 ตามลำดับตามค่ายกกำลังสอง จำนวนค่า 1 ในแต่ละบิตคือ 1+2+4+.....+128 หรือเท่ากับ 255 (2 ยกกำลัง 8 - 1) สำหรับเลขไบนารี 0100 0001 ค่าของเลข 1 ในแต่ละบิตจะมีค่า 1 + 64 จะได้ผลรวมเท่ากับ 65

ค่าของตำแหน่ง	128	64	32	16	8	4	2	1
บิต	0	1	0	0	0	0	0	1

แต่ค่าของ 0100 0001 ไม่ใช่ตัวอักษร A แต่เป็นค่าของจำนวนตัวเลขคือ 65 แต่ถ้าแทนเป็นตัวอักขระก็สามารถกำหนดเป็นตัว A ได้

- ถ้าโปรแกรมกำหนดข้อมูลสำหรับการคำนวนทางคณิตศาสตร์ค่า 0100 0001 จะเป็นค่าของเลขฐานสองเมื่อเทียบกับเลขฐานสิบจะได้ 65
- ถ้าโปรแกรมกำหนดข้อมูล (1 หรือมากกว่า 1 ไบท์) ค่าที่อธิบายนี้หมายถึงส่วนต้นของข้อมูลค่า 0100 0001 จะใช้แทนตัวอักษรหรือข้อมูลที่เป็นตัวสตริง (STRING)

เมื่อท่านจะเขียนโปรแกรมท่านจะต้องทำความเข้าใจเกี่ยวกับข้อมูลในการกำหนดค่าของรายการข้อมูลให้ชัดเจน ในส่วนของการเขียนโปรแกรม การกำหนดจำนวนค่าในไบนารีอาจจะเป็น 8 บิตหรือ 16 บิต หรืออาจจะเป็น 32 บิตขึ้นอยู่กับไมโครโปรเซสเซอร์ ถ้าโครงสร้างเป็น 16 บิต ค่าของจำนวนที่ได้จะเป็น 65535 หรือโครงสร้างเป็น 32 บิตจะได้ค่า 4,294,967,295

การคำนวนเลขฐานสอง (BINARY ARITHMETIC)

รูปแบบการคำนวนทางคณิตศาสตร์ของไมโครคอมพิวเตอร์ จะใช้รูปแบบการคำนวนของเลขฐานสองเพียงอย่างเดียว ผู้เขียนโปรแกรมภาษาแอสแซมบลีจะต้องทำความเข้าใจการคำนวนทางคณิตศาสตร์เลขฐานสอง เช่นการบวก การลบ ดังตัวอย่างต่อไปนี้

```
   0    0    1    1
+     +    +    +
   0    1    1    1
  ---------------------
   0    1   10   10
                + 1
                11
```

การคำนวนให้สังเกตจากตัวทด 1 + 1 และ 1 + 1 + 1 ซึ่งผลที่ได้จากบวกจะมีตัวทด จากตัวอย่างการคำนวนเลขฐานสองถ้าเราต้องการบวก 0100 0001 กับค่า 0010 1010 ตัวอักษร A และตัว Asterisk ซึ่งค่าทั้งสองเปลี่ยนเป็นเลขฐานสิบได้ 65 และ 42

เลขฐานสิบ		เลขฐานสอง
65		0100 0001
+ 42	+	0010 1010
107		0110 1011

การตรวจสอบผลลัพธ์ที่ได้จากการคำนวนเลขฐานสอง 0110 1011 จะมีค่าเท่ากับ 107 ถ้าเรานำเลขชุดอื่นๆ ก็จะได้ผลลัพธ์ที่ถูกต้องเช่นกัน

เลขฐานสิบ		เลขฐานสอง
60		0011 1100
+ 53	+	0011 0101
113		0111 0001

ค่าเลขจำนวนลบ (NEGATIVE NUMBERS)

ตัวอย่างของการคำนวนเลขฐานสองที่กล่าวมาเป็นการคำนวนค่าบวกทั้งหมด เพราะบิตซ้ายสุดของเลขฐานสองมีค่าเป็น 0 ถ้าเป็นเลขจำนวนติดลบบิตซ้ายสุดจะเป็นค่า 1 อย่างไรก็ตามจะไม่มีการเปลี่ยนบิตซ้ายสุดเป็น 1 เช่น 0100 0001 (+65) เป็น 1100 0001 การคำนวนค่าลบของเลขฐานสองจะต้องใช้กฎเกณฑ์ของ 2' คอมพลีเม้นต์ นั่นคือ ค่าลบของเลขฐานสองจะต้องเปลี่ยนทุกบิตเป็นตรงกันข้ามแล้วบวกด้วย 1 ตัวอย่างเราจะใช้เลขฐานสอง 0100 0001 ดังนี้

ค่า 65	0100 0001	
เปลี่ยนค่าเป็นตรงกันข้าม	1011 1110	
บวก 1	1011 1111	(-65)

ระบบเลขฐานสองจะเข้าใจว่า ค่าติดลบบิตซ้ายสุดจะเป็นค่า 1 แต่ถ้าท่านบวก 1 จะได้ 1011 1111 ถ้าท่านต้องการเปลี่ยนกลับเป็นค่าเดิม โดยกลับค่าทุกบิตเป็นตรงกันข้ามแล้วบวก 1

เลขฐานสอง	1011 1111	
เปลี่ยนค่าตรงข้าม	0100 0000	
บวก 1	0100 0001	(+65)

ฉนั้นถ้าเรานำ +65 และ -65 มารวมกันเป็น 0

+ 65	0100	0001
- 65	+ 1011	1111
00	(1) 0000	0000

ค่าของ 8 บิตหลัก จะเป็นค่า 0 บิตตัวทดจะมีค่าเป็น 1 อยู่ซ้ายสุด อย่างไรก็ตามถ้ามีตัวทดในบิตเครื่องหมาย ตัวทดจะถูกตัดทิ้ง ดังตัวอย่างและที่เหลือคือผลลัพธ์ที่ถูกต้อง

การลบเลขฐานสองก็เป็นอย่างหนึ่ง จะต้องมีการเปลี่ยนค่าที่จะนำมาลบให้อยู่ในรูปของ 2'คอมพลีเม้นต์แล้วบวกเข้ากับจำนวนที่เป็นตัวตั้ง เช่นการลบ 42 จาก 65 ค่าเลขฐานสองของ 42 คือ 00101010 และค่า 2' คอมพลีเม้นต์คือ 1101 0110

65	0100 0001
+ (- 42)	+ 1101 0110
23	(1) 0001 0111

จะได้ผลลัพธ์คือ 23 มีตัวทดออกมา 1 ตัว แล้วให้ตัดทิ้ง ที่เหลือคือคำตอบ ถ้ามีการเปลี่ยนแปลงการทำ 2' คอมพลีเม้นต์ผลลัพธ์ที่ได้ออกมาจะไม่ชัดเจน เราจะพิจารณาดังต่อไปนี้ ค่าอะไรที่บวกกับค่าของเลขฐานสองได้ 0000 0001 ซึ่งมีค่าเท่ากับ 0000 0000 ในเทอมของเลขฐานสิบคำตอบคือ -1 ค่า 2' คอมพลีเม้นต์ของ -1 คือ 1111 1111 ค่าบวก 1 และลบ 1 จะบวกได้ดังนี้

	1	0000 0001
	+(-1)	1111 1111
ผลลัพธ์	(1)	0000 0000

ตัวทดที่มีค่าเป็น 1 เราตัดทิ้ง ท่านก็จะได้ค่าผลลัพธ์เป็น 1111 1111 มีค่าเป็นลบ 1 จะมีรูปแบบของเลขฐานสองดังต่อไปนี้

+3	0000 0011
+2	0000 0010
+1	0000 0001
0	0000 0000
-1	1111 1111
-2	1111 1110
-3	1111 1101

ดูจากรูปแบบของเลขฐานสองแล้วบิต 0 ของค่าติดลบนั้นเลข 0 ภายในข้อมูลเปรียบเสมือนค่าติดลบโดยการบวก 1 ให้กับค่า 0

การแทนด้วยเลขฐานสิบหก

ในการแทนเลขฐานสองด้วยเลขฐานสิบหก ในการเก็บข้อมูลชนิด 4 ไบท์ เราจะเห็นว่ามีแต่เลข 1 และเลข 0 ถึง 32 ตัว แต่ในบางครั้งเราอาจใช้ข้อมูลถึง 256 บิตก็ตาม ความยุ่งยากในการแสดงผลลัพธ์ทางจอภาพหรือเครื่องพิมพ์ เราใช้รหัสมาตราฐาน ASCII เป็นตัวกำหนดในการแสดงผล ไม่ว่าจะเป็นรหัสควบคุมหรือรหัสตัวอักขระต่างๆ นักออกแบบคอมพิวเตอร์ก็จะต้องมีการพัฒนา วิธีการที่แทนรหัสเลขฐานสองให้เหลือสั้นที่สุดในรูปข้อมูล การแบ่งข้อมูลเลขฐานสองในแต่ละไบท์มี 8 บิต เราจะแบ่งทีละครึ่งไบท์ดังต่อไปนี้

ฐานสอง:	0101	1001	0011	0101	1011	1100	1110	1001	1100	1110
ฐานสิบ :	5	9	3	5	11	12	14	9	12	14

ค่าของเลขฐานสองจำนวน 4 บิตจะต้องใช้เลข 2 ตัว ดังนั้นจำนวนในระบบเลขฐานสิบหกจึงกำหนดค่า 10 = A , 11 = B , 12 = C , 13 = D , 14 = E , และ 15 = F ซึ่งเป็นการแทนข้อมูล 4 ตัวเหลือเพียงตัวเดียว

59 35 BC E9

จำนวนตัวเลขในระบบจะมีตัวเลข 0 - F ซึ่งมีตัวเลขอยู่ 16 ตัวเราเรียกว่า HEXADECIMAL REPRESENTATION ตามรูป 1-1 ซึ่งแสดงเลขฐานสอง ฐานสิบ และฐานสิบหกของเลข 0 -15

BINARY	DECIMAL	HEXADECIMAL	BINARY	DECIMAL	HEXADECIMAL
0000	0	0	1000	8	8
0001	1	1	1001	9	9
0010	2	2	1010	10	A
0011	3	3	1011	11	B
0100	4	4	1100	12	C
0101	5	5	1101	13	D
0110	6	6	1110	14	E
0111	7	7	1111	15	F

รูป 1-1 Binary , Decimal , และ Hexadecimal Representation

ภาษาแอสแซมบลีเราจะพิจารณาใช้เลขฐานสิบหก (Hex) หลังจากที่เราแปลเป็นภาษาเครื่องแล้ว ทั้งหมดข้อมูล คำสั่ง ค่าคงที่ และแอดเดรสจะอยู่ในรูปเลขฐานสิบหก เราสามารถใช้โปรแกรมดีบักแสดงค่าแอดเดรสข้อมูลในเลขฐานสิบหก

ท่านสามารถใช้รูปแบบของเลขฐานสิบหกในการทำงาน เช่นค่าของ F ในเลขฐานสิบหก ซึ่งเป็นค่าสูงสุดและตามด้วย 10 ซึ่งเทียบกับเลขฐานสิบคือ 16 ดังตัวอย่างการคำนวนทางคณิตศาสตร์ของเลขฐานสิบหก มีการคำนวนดังต่อไปนี้

6	5	F	F	10	FF
+ 4	+ 8	+ 1	+ F	+ 10	+ 1
A	D	10	1E	20	100

สังเกตุว่า 20H มีค่าเท่ากับ 32 ในเลขฐานสิบและค่า 100H มีค่าเท่ากับ 256 และ 1000H มีค่าเท่ากับ 4096 ตัวที่กำหนดค่าเลขฐานสิบหกในโปรแกรมเราต้องใช้ตัวอักษร H ตามหลังเลขจำนวนนั้น เช่น 25H (เลขฐานสิบเท่ากับ 37) จำนวนค่าของเลขฐานสิบจะเริ่มต้นที่ 0 - 9 เสมอ ถ้าเลขฐานสิบหกเท่ากับ B8H เราจะต้องเริ่มต้นด้วย 0B8H ถ้าเป็นค่าของเลขฐานสองเช่น 0100 1100 หรือ 0100 1100B ฉนั้นระบบเลขฐานสองจะต้องมีอักษร B กำกับต่อท้ายค่าคงที่

รหัสแอสกี ASCII CODE

มาตราฐานการใช้รหัสแทนข้อมูลในไมโครโปรเซสเซอร์ จะใช้รหัสมาตราฐานของ ASCII (American Standard Code for Information Interchange) ซึ่งเป็นรหัสมาตราฐานของคอมพิวเตอร์ ASCII เป็นรหัส 8 บิต สามารถกำหนดแทนตัวอักษรได้ทั้งหมด 256 ตัว รวมทั้งสัญญลักษณ์และภาษาต่างๆ ตัวอย่างตัวอักษร A จะมีรหัสเลขฐานสิบหกคือ 41H ดูรายละเอียดจากภาคผนวก

ฮาร์ดแวร์ของพีซี PC HARDWARE

ส่วนประกอบต่างๆที่สำคัญของฮาร์ดแวร์ในระบบ PC หนึ่งระบบประกอบไปด้วย SYSTEM BOARD, POWER SUPPLY , และ EXPLANSION SLOT ส่วน SYSTEM BOARD ใช้ CPU ของ Intel ,ROM,RAM

โปรเซสเซอร์ PROCESSOR

โปรเซสเซอร์หรือซีพียู เป็นสมองของระบบไมโครคอมพิวเตอร์ซึ่งใช้โปรเซสเซอร์ของบริษัท Intel (ใช้เฉพาะการศึกษาของหนังสือเล่มนี้) ที่มีพื้นฐานมาจาก 8086 ซึ่งเป็นรูปแบบข้อมูลหรือคำสั่งที่ใช้ในการประมวลผล การทำงานของตัวโปรเซสเซอร์มีความเร็วสูง หน่วยความจำมีความจุสูง รีจีสเตอร์ และบัสข้อมูล (Data Bus) บัสข้อมูลทำหน้าที่เคลื่อนย้ายข้อมูลระหว่างโปรเซสเซอร์ หน่วยความจำและอุปกรณ์รอบข้าง (Peripheral Devices)

8088/80188 เป็นโปรเซสเซอร์ขนาด 16 บิต มีรีจีสเตอร์ขนาด 16 บิต และบัสข้อมูลขนาด 8 บิต สามารถกำหนดแอดเดรสได้สูงถึง 1 ล้านไบท์ในหน่วยความจำ รีจีสเตอร์สามารถประมวลผลข้อมูลครั้งละ 2 ไบท์ในเวลาเดียวกัน ขณะที่บัสข้อมูลสามารถเคลื่อนย้ายข้อมูลครั้งละ 1 ไบท์ในเวลาเดียวกัน 80188 มีคำสั่งมากกว่า 8088.

8086/80186 ตัวโปรเซสเซอร์เบอร์นี้มีการทำงานเหมือนกับ 8088/80188 แต่มีบัสข้อมูลขนาด 16 บิต การเคลื่อนย้ายข้อมูลทำงานได้เร็วกว่า 80186 มีคำสั่งมากกว่า 8086

80286 ตัวโปรเซสเซอร์ตัวนี้ทำงานได้เร็วกว่าตัวโปรเซสเซอร์ที่กล่าวมาแล้ว สามารถกำแอดเดรสเดรสได้ถึง 16 ล้านไบท์ ทำงานในโหมด REAL MODE และ PROTECTED MODE สามารถทำงานแบบ Multitasking

80386 ตัวโปรเซสเซอร์ตัวนี้ มีรีจีสเตอร์ขนาด 32 บิต และบัสข้อมูล 32 บิต และสามารถกำหนดแอดเดรสในหน่วยความจำได้ 4 Billion Byte ทำงานในส่วนของ REAL MODE และ PROTECTED MODE สามารถทำงานแบบ Multitasking

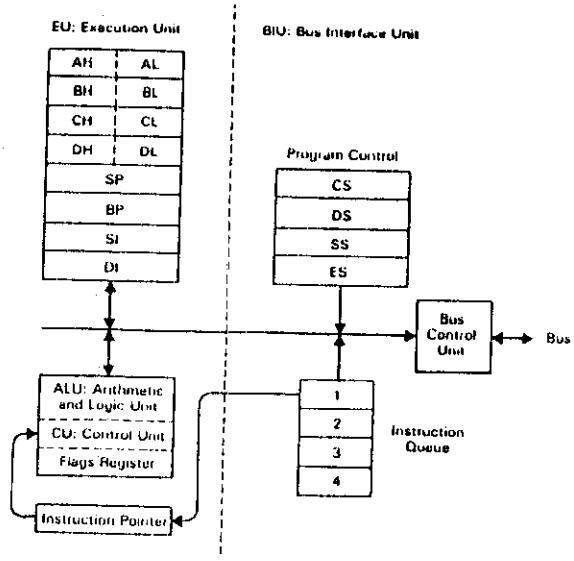
80486 ตัวประมวลผลตัวนี้มีรีจีสเตอร์ขนาด 32 บิต และบัสข้อมูลขนาด 32 บิต ซึ่งการทำงานดีกว่า 80386 ทำงานใน REAL MODE และ PROTECTED MODE.

EXECUTION UNIT AND BUS INTERFACE UNIT

ตัวประมวลผลจะแบ่งทำงานออกเป็น 2 ส่วน คือ EXECUTION UNIT (EU) และ BUS INTERFACE UNIT (BIU) ตามที่แสดงในรูป 1-2 การทำงานของ EU คือการเอ็กซีคิวส์คำสั่งและข้อมูล ที่ส่งมาจาก BIU ภายในส่วนของ EU จะประกอบไปด้วย หน่วยคำนวนทางคณิตศาสตร์และลอจิก (ALU) หน่วยควบคุม (CU) และกลุ่มของรีจีสเตอร์ต่างๆ ซึ่งรีจีสเตอร์เหล่านี้ช่วยในการคำนวนทางคณิตศาสตร์และลอจิก ส่วน BIU ที่สำคัญก็คือหน่วยควบคุมบัส (BUS CONTROL UNIT) รีจีสเตอร์เซคเม้นต์ (SEGMENT REGISTER) และคิวคำสั่ง (INSTRUCTION QUEUE) ขั้นตอนแรก BIU จะทำหน้าที่ควบคุมบัสในการเคลื่อนย้ายข้อมูลไปยัง EU , หน่วยความจำ และอุปกรณ์ I/O ภายนอก ขั้นตอนที่ 2 รีจีสเตอร์เซคเม้นต์จะควบการทำงานของแอดเดรสในหน่วยความจำ

ขั้นตอนที่ 3 ของ BIU คือการเข้าถึงคำสั่งทั้งหมดที่อยู่ในหน่วยความจำ โดยการอ่านคำสั่งที่อยู่ในหน่วยความจำไปยัง INSTRUCTION QUEUE ขนาดของ QUEUE มีขนาด 4 ไบท์หรือมากกว่า 4 ไบท์ ขึ้นอยู่กับตัวไมโครโปรเซสเซอร์ BIU สามารถที่จะทำการเฟทซ์คำสั่งได้ล่วงหน้า ฉนั้นคำสั่งที่อยู่ใน QUEUE พร้อมที่จะเอ็กซีคิวส์

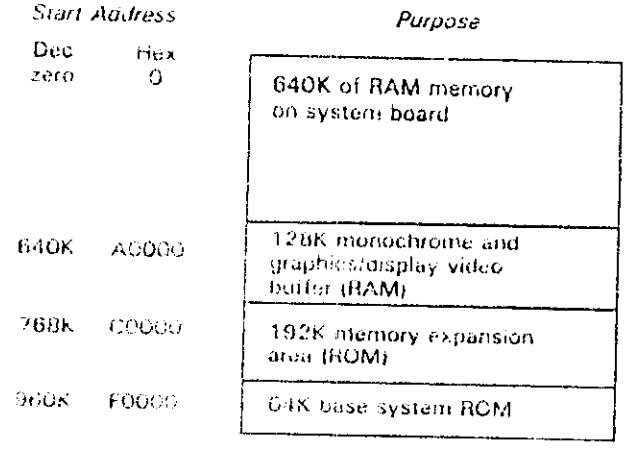
หน่วย EU และ BIU จะทำงานขนานกันไป ซึ่ง BIU จะทำงานล่วงหน้า 1 ขั้นตอน หน่วย EU จะทำงานตามหลังหน่วย BIU ซึ่งหน่วยนี้จะทำหน้าที่เข้าถึงข้อมูลจากหน่วยความจำและ I/O ดังนั้นหน่วย EU จะต้องรอคำสั่งที่มาจาก EIU เพื่อทำการเอ็กซีคิวส์



รูป 1-2 Execute unit and Bus interface unit

INTERNAL MEMORY

หน่วยความจำภายในของไมโครคอมพิวเตอร์แบ่งออกเป็น 2 ชนิดคือ RAM (RANDOM ACCESS MEMORY) และ ROM (READ ONLY MEMORY) แอดเดรสของหน่วยความจำทั้งสองจะเริ่มต้นที่ 00 จนถึงค่าสูงสุด ส่วนของ RAM และ RAM ขึ้นอยู่กับโครงสร้างของการจัดการหน่วยความจำ ดังตัวอย่าง 8086 ตามรูป 1-3 ซึ่งมี RAM ขนาด 640 K.



รูป 1-3 Physical Memory map

READ ONLY MEMORY

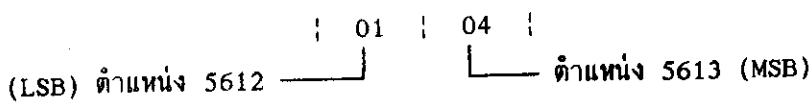
หน่วยความจำชนิด ROM เป็นหน่วยความจำชนิดพิเศษที่ใช้อ่านข้อมูลได้อย่างเดียว ข้อมูลและคำสั่งที่อยู่ในหน่วยความจำชนิดนี้จะถูกเก็บไว้อย่างถาวรไม่มีสูญหาย และไม่สามารถเปลี่ยนแปลงข้อมูลได้ โปรแกรม BIOS (BASIC INPUT OUTPUT SYSTEM) ที่เก็บไว้ใน ROM จะมีแอดเดรสเริ่มแรกที่ 768K ซึ่งเป็นคำสั่งควบคุมอุปกรณ์ I/O เช่นควบคุมฮาร์ดดิสก์ ส่วนแอดเดรสของ ROM ที่ 960K ควบคุมการทำงานของเครื่องคอมพิวเตอร์ เช่นเปิดเครื่อง ตรวจสอบตัวเอง เป็นต้น

RANDOM ACCESS MEMORY

หน่วยความจำชนิด RAM นี้เป็นพื้นที่สำหรับโปรแกรมเมอร์ใช้ในการเก็บคำสั่งและข้อมูล ซึ่งหน่วยความจำชนิดนี้สามารถเขียน/อ่านข้อมูลได้ จัดเป็นหน่วยความจำชั่วคราวสำหรับการเอ็กซีคิวส์โปรแกรม ข้อมูลที่เก็บไว้ใน RAM ถ้าเราปิดสวิตหรือไม่จ่ายกระแสไฟฟ้าให้กับวงจรหน่วยความจำข้อมูลเหล่านี้ก็จะสูญหายไป แต่ถ้าต้องการเก็บข้อมูลเราสามารถเก็บข้อมูลเหล่านี้ไว้ในหน่วยความจำสำรองก่อนการปิดสวิต แต่ถ้าเรายังไม่ปิดสวิตข้อมูลก็ยังมีอยู่ไม่สูญหาย

ADDRESSING MEMORY LOCATIONS

การกำหนดแอดเดรสของหน่วยความจำนั้น ขึ้นอยู่กับชนิดของโปรเซสเซอร์แต่ละตัว ในการเข้าถึงข้อมูลจากหน่วยความจำครั้งละ 1 ไบท์ หรือมากกว่า 1 ไบท์ การกำหนดแอดเดรสให้เราพิจารณาจากเลขฐานสิบ 1025 สามารถแทนเป็นเลขฐานสิบหกได้ 0401H ฉนั้นต้องใช้เนื้อที่ 2 ไบท์หรือ 1 เวิร์ด ประกอบด้วยไบท์สูงคือ 04H และไบท์ต่ำคือ 01H ในระบบของไมโครคอมพิวเตอร์จะเก็บข้อมูลในลักษณะตรงกันข้ามคือข้อมูลไบท์ต่ำจะเก็บในแอดเดรสต่ำ ข้อมูลไบท์สูงจะเก็บไว้ในแอดเดรสสูง ตัวโปรเซสเซอร์จะเคลื่อนย้ายข้อมูลจากรีจีสเตอร์ 0401ไปยังหน่วยความจำที่แอดเดรส 5612 และ 5613 ดังนี้



SEGMENTS AND ADDRESSING

เซคเม้นต์คือพื้นที่ในหน่วยความจำที่มีการกำหนดขอบเขต นั่นคือทุกเซคเม้นต์ในแต่ละตำแหน่งจะหารด้วย 16 หรือ 10H เซคเม้นต์ที่กำหนดตำแหน่งใหน่วยความจำจะเป็นการทำงานของ REAL MODE แต่ละเซคเม้นต์มีขนาด 64 KB เพื่อให้โปรแกรมใช้ในการเอ็กซีคิวส์

การทำงานในส่วนของ REAL MODE จะกำหนดเซคเม้นต์ขนาด 64 KB การกำหนดเซตเม้นต์แบ่งออกเป็น 4 ชนิดในส่วนของ RAM คือ 64K x 4 = 256K หน่วยความจำจะมีเซคเม้นต์หลักที่ใช้ในการทำงานคือ CODE SEGMENT , DATA SEGMENT , STACK SEGMENT.

CODE SEGMENT

CODE SEGMENT จะเป็นหน่วยความจำที่ใช้ในการเก็บชุดคำสั่งที่ใช้ในการเอ็กซีคิวส์ คำสั่งที่จะเริ่มต้นเอ็กซีคิวส์คำสั่งแรกอยู่ที่ CODE SEGMENT หน่วยความจำของเซคเม้นต์นี้จะเชื่อมโยงกับระบบสำหรับการเอ็กซีคิวส์โปรแกรม โดยมีรีจีสเตอร์ CS เป็นตัวเก็บแอดเดรสฐานเซคเม้นต์

DATA SEGMENT

DATA SEGMENT จะเป็นส่วนที่ใช้ในการเก็บข้อมูลที่จะนำมาเอ็กซีคิวส์ของโปรแกรม ค่าคงที่ แอดเดรสฐานของเซคเม้นต์นี้จะอยู่ในรีจีสเตอร์ DS

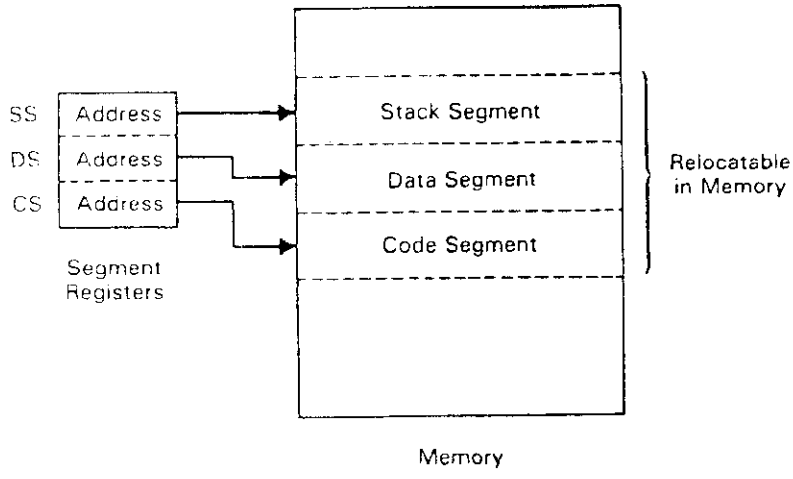
STACK SEGMENT

STACK SEGMENT ทำหน้าที่เก็บข้อมูลและแอดเดรสที่ท่านต้องการไว้ชั่วคราว ในกรณีที่ท่านเรียกใช้โปรแกรมย่อย หรือย่อนกลับมาที่โปรแกรมหลัก แอดเดรสฐานของเซคเม้นต์นี้จะอยู่ในรีจีสเตอร์ SS

SEGMENT BOUNDARIES

เซคเม้นต์รีจีสเตอร์แต่ละตัวจะมีข้อมูลฐานแอดเดรสเริ่มต้นของแต่ละเซคเม้นต์ ตามรูป 1 - 4 ซึ่งแสดงฐานแอดเดรสของแต่ละกลุ่มมี CS , DS , SS ส่วน ES (EXTRA SEGMENT) ถ้าเป็นไมโครโปรเซสเซอร์ 80386/80486 จะมีเซคเม้นต์รีจีสเตอร์ FS และ GS เพิ่มขึ้น

ในส่วนแรกจะอธิบายเกี่ยวกับขอบเขตที่เริ่มต้นของเซคเม้นต์ ซึ่งฐานแอดเดรสของเซคเม้นต์จะหารด้วย 16 หรือ 10H สมมุติว่า DATA SEGMENT ฐานแอดเดรสอยู่ที่ 0405F0 ในกรณีนี้หรือกรณีอื่นใดก็ตาม ตัวเลขฐานสิบหกขวามือสุดเป็นเลข 0 ผู้ออกแบบคอมพิวเตอร์ค่าของเลขฐานสิบหกขวามือสุดไม่จำเป็นต้องเก็บไว้ในเซคเม้นต์รีจีสเตอร์ 0405F0H จะเก็บข้อมูลเพียง 045FH ในหนังสือเล่มนี้จะใช้ดังนี้ 045F[0]



รูป 1-4 Segment and registers

SEGMENT OFFSETS

ในการเขียนโปรแกรมภาษาแอสแซมบลี ตำแหน่งของหน่วยความจำทั้งหมดจะมีความสัมพันธ์กับแอดเดรสเริ่มแรกของเซคเม้นต์ ค่า Offset จะกำหนดระยะข้อมูลขนาด 1 ไบท์ (หรือค่า Displacement) จากจุดเริ่มต้นของเซคเม้นต์ ข้อมูล Offset ขนาด 2 ไบท์ (16 บิต) จะอยู่ในช่วง 0000H ถึง FFFFH หรือ 0 - 65535 ไบท์แรกของ CS คือค่า Offset เป็น 00 ไบท์ที่ 2 คือ 01 และค่า Offset จะเพิ่มไปเรื่อยๆจนค่า Offset เป็น 65535 การอ้างอิงแอดเดรสทุกตำแหน่งในหน่วยความจำของแต่ละเซคเม้นต์การอ้างอิงแอดเดรสคือการเอาผลรวมของค่า Offset กับค่าฐานแอดเดรสในรีจีสเตอร์

จากตัวอย่างต่อไปนี้ รีจีสเตอร์ DS จะมีข้อมูลแอดเดรสเริ่มแรกของ DATA SEGMENT ที่ 045F[0]H และคำสั่งอ้างอิงของค่า Offset 0032 ที่อยู่ภายใน DATA SEGMENT ฉนั้นเราจะได้ตำแหน่งที่อ้างอิงโดยคำสั่งกำหนดที่ตำแหน่ง 04622H

แอดเดรสของ DS	0	4	5	F	0
ค่า Offset	+	0	0	3	2
แอดเดรสที่คำนวนได้	0	4	6	2	2

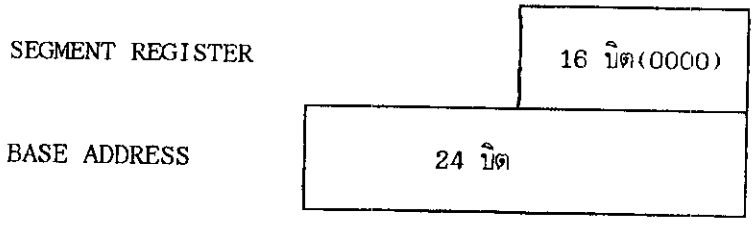
สังเกตุจากในโปรแกรมหนึ่งๆจะมีเซคเม้นต์ตั้งแต่ 2 เซคเม้นต์หรือมากกว่าและอาจจะเริ่มต้นทุกๆ ตำแหน่งในหน่วยความจำ

MEMORY CAPACITY

ระบบ PC ที่ใช้ไมโครโปรเซสเซอร์ของตระกูล Intel สามารถกำหนดแอดเดรสที่มีขนาดต่างกัน และหน่วยความจำต่างกัน

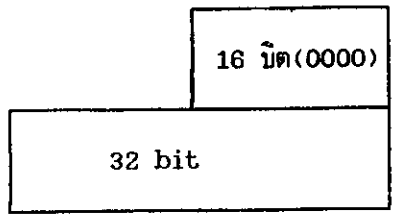
8086/8088 ADDRESSING รีจีสเตอร์ของโปรเซสเซอร์ 8086/8088 ซึ่งมีขนาด 16 บิต ขอบเขตของแอดเดรสเซคเม้นต์นี้จะมี 4 บิตทางขวามือของแอดเดรสเท่ากับ 0 ดังที่อธิบายไปแล้วว่าแอดเดรสที่เก็บในรีจีสเตอร์เซคเม้นต์ที่คอมพิวเตอร์ให้ 4 บิตขวามือเป็น 0 ,[0] ค่าของแอดเดรสสูงสุดคือ FFFF[0]H หรือ 1,048,560 ไบท์

80286 ADDRESSING ในส่วนของการทำงานที่เรียกว่า REAL MODE ไมโครโปรเซสเซอร์ 80286 จะทำงานเหมือนกับ 8086 ยกเว้น 80286 ทำงานได้เร็วกว่า ในส่วนของ PROTECTED MODE 80286 มีแอดเดรสขนาด 24 บิต ดังนั้น ค่าแอดเดรสสูงสุดคือ FFFFF[0]H หรือ 16 ล้านไบท์ รีจีสเตอร์เซคเม้นต์จะทำการใช้ตัวเลือกสำหรับการอ่าน/เขียนข้อมูล 24 บิตของฐานแอดเดรสจากหน่วยความจำและบวกกับค่า Offset address ขนาด 16 บิต



80386/80486 ADDRESSING

ไมโครโปรเซสเซอร์ 80386/80486 การทำงานในส่วนของ REAL MODE จะทำงานเหมือนกับ 8086 แต่มีความเร็วสูงกว่า การทำงานของ PROTECTED MODE มีแอดเดรสทั้งหมด 48 บิตสามารถกำหนดแอดเดรสของเซคเม้นต์ได้สูงถึง 4 BILLION BYTE เซคเม้นต์รีจีสเตอร์มีขนาด 16 บิต ซึ่งทำหน้าที่เลือกฐานแอดเดรส (BASE ADDRESS) ขนาด 32 บิตจากหน่วยความจำ และบวกค่า Offset ซึ่งมีขนาด 32 บิต



รีจีสเตอร์ REGISTER

รีจีสเตอร์ที่อยู่ภายในตัวโปรเซสเซอร์ที่ใช้ร่วมกับคำสั่งในการเอ็กซีคิวส์ การเก็บค่าของแอดเดรสในหน่วยความจำ และช่วยในการคำนวนทางคณิตศาสตร์และลอจิก รีจีสเตอร์แต่ละตัวจะมีชื่อของตัวเอง ขนาดจำนวนบิตที่แน่นอน การกำหนดตำแหน่งของบิตจะเริ่มจากขวาไปซ้ายดังนี้

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

MSB LSB

เซคเม้นต์รีจีสเตอร์ SEGMENT REGISTER

เซคเม้นต์รีจีสเตอร์มีความยาวขนาด 16 บิต ใช้เก็บข้อมูลของแอดเดรสในหน่วยความจำ ที่เป็นแอดเดรสปัจจุบันของเซคเม้นต์ เซคเม้นต์รีจีสเตอร์แต่ละตัวมีหน้าที่ดังนี้

CS REGISTER รีจีสเตอร์ CS ใช้เก็บแอดเดรสของโปรแกรมใน CODE SEGMENT แอดเดรสจะต้องนำมาบวกกับค่า Offset address ที่อยู่ใน INSTRUCTION POINTER (IP) เพื่อชี้คำสั่งที่จะนำมาเฟทช์และเอ็กซีคิวส์

DS REGISTER รีจีสเตอร์ DS เก็บข้อมูลแอดเดรสในส่วนของที่เป็นข้อมูลของโปรแกรม ที่อยู่ใน DATA SEGMENT ข้อมูลที่อยู่ใน DS จะนำมาบวกกับค่า Offset address ของคำสั่งเพื่ออ้างอิงแอดเดรสของข้อมูลใน DATA SEGMENT

SS REGISTER รีจีสเตอร์ SS เป็นตัวกำหนดในการใช้สแตคของหน่วยความจำ ซึ่งใช้เก็บข้อมูลชั่วคราวของแอดเดรสและข้อมูลที่ต้องการ รีจีสเตอร์ SS เป็นตัวเก็บแอดเดรสเริ่มต้นของโปรแกรมสแตค แอดเดรสที่ถูก PUSH นั้นเป็นค่า Offset ใน STACK POINTER (SP) รีจีสเตอร์ SP จะเป็นตัวชี้เวิร์คปัจจุบันของสแตค ถ้าเขียนโปรแกรมเล็กๆ ไม่ต้องอ้างอิงถึงรีจีสเตอร์ SS ก็ได้

ES REGISTER ข้อมูลที่เป็นสตริง (ตัวอักขระ) การทำงานบางครั้งต้องใช้ EXTRA SEGMENT เพื่อเก็บแอดเดรสที่อ้างอิงของหน่วยความจำ รีจีสเตอร์ ES ใช้ร่วมกับ DI ถ้าโปรแกรมต้องการใช้ ES ต้องกำหนดค่าให้ด้วย

FS and GS REGISTER รีจีสเตอร์ FS และ GS เป็นรีจัสเตอร์ที่เพิ่มขึ้นมาใช้เหมือนกับ ES แต่ใช้กับไมโครโปรเซสเซอร์ 80386/80486

รีจีสเตอร์คำสั่ง INSTRUCTION POINTER REGISTER

รีจีสเตอร์ IP มีขนาด 16 บิต ทำหน้าที่เก็บค่า Offset Address ของคำสั่งที่จะส่งไปเอ็กซีคิวส์ ข้อมูลของรีจีสเตอร์ IP จะนำมาคำนวณร่วมกับรีจีสเตอร์ CS เช่นรีจีสเตอร์ IP กำหนดตำแหน่งปัจจุบันของคำสั่งภายในส่วนของ Code Segment ข้อมูลที่อ้างอิงของรีจีสเตอร์ IP ท่านสามารถเปลี่ยนแปลงได้เมื่อท่านใช้โปรแกรม DEBUG ของ DOS ในการตรวจสอบโปรแกรม 30386/80486 มีรีจีสเตอร์ IP ขนาด 32 บิต เรียกว่า EIP

จากตัวอย่างต่อไปนี้ รีจีสเตอร์ CS มีข้อมูล 25A4[0] และรีจีสเตอร์ IP มีข้อมูล 412 เพื่อกำหนดคำสั่งต่อไปที่จะนำมาเอ็กซีคิวส์ ตัวโปรเซสเซอร์จะคำนวนแอดเดรสของ CS และ IP ได้ดังนี้

ข้อมูลของรีจีสเตอร์ CS	2 5 A 4 0
ข้อมูลของรีจีสเตอร์ IP	+ 4 1 2
แอดเดรสของคำสั่งถัดไป	2 5 E 5 2

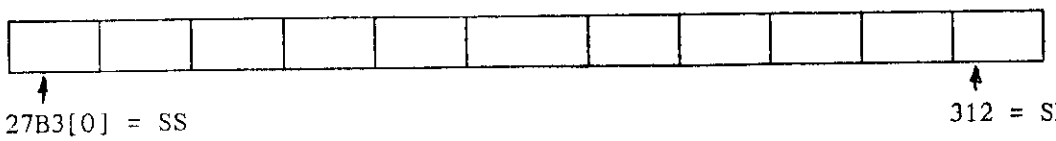
POINTER REGISTER

กลุ่มของรีจีสเตอร์ตัวชี้มี 2 ตัว คือรีจีสเตอร์ SP และ BP จะใช้ร่วมกันกับรีจิสเตอร์ SS ในการกำหนดแอดเดรสเพื่อเข้าถึงข้อมูลของหน่วยความจำที่เรียกว่า STACK MEMORY

รีจีสเตอร์ SP มีขนาด 16 บิตทำงานร่วมกับรีจีสเตอร์ SS ในคำนวณหาแอดเดรส ซึ่งรีจีสเตอร์ SP ทำหน้าที่เก็บค่า Offset Address ที่อ้างถึงแอดเดรสปัจจุบันที่จะเอ็กซีคิวส์ของหน่วยความจำสแตค ในโปรเซสเซอร์ 80386/80486 จะมีรีจีสเตอร์ SP ขนาด 32 บิตที่เรียกว่า ESP สมมุติว่ารีจีสเตอร์ SS มีข้อมูล 27B3[0] และรีจีสเตอร์ SP มีข้อมูล 312 การคำนวนหาแอดเดรสปัจจุบันจะได้ดังนี้

ข้อมูลของรีจีสเตอร์ SS 2 7 B 3 0
ข้อมูลของรีจีสเตอร์ SP + 3 1 2
แอดเดรสของสแตค 2 7 E 4 2

Stack Segment:



รีจีสเตอร์ BP เป็นรีจีสเตอร์ขนาด 16 บิต หรือ Base Pointer ที่เป็นตัวชี้ค่าของแอดเดรสที่อยู่ในสแตค ในโปรเซสเซอร์ 80386/80486 รีจีสเตอร์ตัวนี้มีค่า 32 บิต เรียกว่า EBP

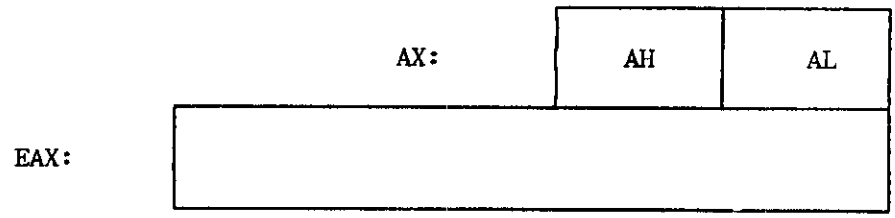
รีจีสเตอร์ทั่วไป GENERAL PURPOSE REGISTER

รีจีสเตอร์ที่ใช้งานทั่วๆไปคือ AX , BX , CX , DX รีจีสเตอร์เหล่านี้สามารถเก็บข้อมูลครั้งละ 1 เวิร์ดหรือข้อมูลครั้งละ 1 ไบท์ บิตที่อยู่ซ้ายสุดเป็นค่าสูงสุด บิตที่อยู่ขวาสุดเป็นค่าต่ำสุด ดังตัวอย่างของรีจีสเตอร์ CX ประกอบด้วย CH กับ CL ท่านสามารถอ้างถึงรีจีสเตอร์นี้ได้ 3 ความหมายหรือ 3 ชื่อ จากตัวอย่างคำสั่งต่อไปนี้

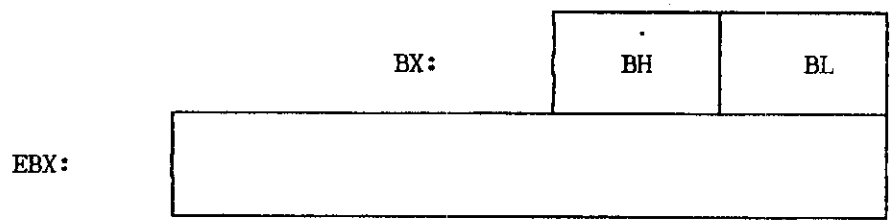
```
MOV     CX,00
MOV     CH,00
MOV     CL,00
```

โปรเซสเซอร์ 80386/80486 รีจีสเตอร์ทั่วไปจะมีขนาด 32 บิต คือ EAX , EBX , ECX , EDX สามารถประมวลผลได้ 3 ขนาดคือ 8 บิต 16 บิต และ 32 บิต

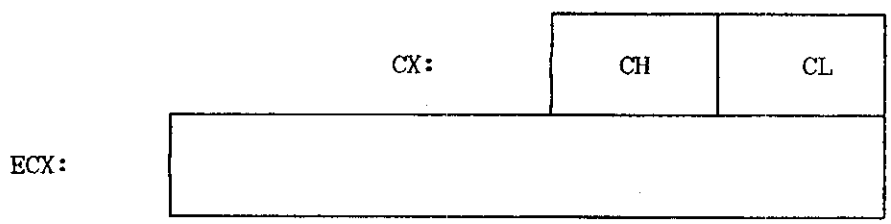
รีจีสเตอร์ AX รีจีสเตอร์ตัวนี้เป็นรีจีสเตอร์หลักที่ใช้ในการประมวลผลเรียกว่า ACCUMULATOR ใช้ทำงานร่วมกับ INPUT/OUTPUT และการคำนวนทางคณิตศาสตร์ เช่นการคูณ การหาร เป็นต้น



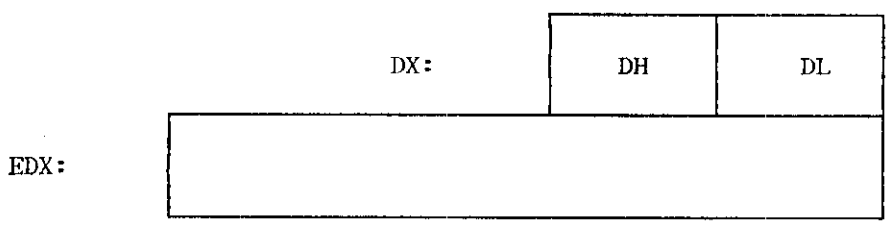
รีจีสเตอร์ BX เราเรียกว่า BASE REGISTER **สำหรับใช้งานทั่วๆไปในการคำนวน และใช้เป็นค่า** INDEX **ในการกำหนดแอดเดรส**



รีจีสเตอร์ CX เราเรียกว่า COUNT REGISTER **ใช้เป็นตัวเก็บค่าการควบคุมของคำสั่งในการวนรอบที่ทำงานซ้ำๆกัน หรือเป็นตัวเก็บค่าของจำนวนที่เลื่อนซ้ายหรือขวา บางครั้งอาจใช้ในการคำนวนก็ได้**



รีจีสเตอร์ DX เราเรียกว่า DATA REGISTER **ใช้เป็นตัวอ้างอิงการทำงานของ** INPUT/OUTPUT **และใช้ในการทำงานการคูณ การหาร**



INDEX REGISTER

รีจิสเตอร์อินเดคเป็นรีจีสเตอร์ที่ช่วยในการกำหนดแอดเดรสในการเพิ่มหรือลดแอดเดรส

SI REGISTER เป็นอินเดครีจีสเตอร์ขนาด 16 บิตหรือเรียกว่า Source Index register สำหรับการทำงานเกี่ยวกับสตริง และทำงานร่วมกับรีจีสเตอร์ DS

DI REGISTER เป็นอินเดครีจีสเตอร์ขนาด 16 บิตหรือเรียกว่า Destination Index register สำหรับการทำงานเกี่ยวกับสตริง และทำงานร่วมกับรีจีสเตอร์ ES

FLAGS REGISTER

แฟลกรีจีสเตอร์มีขนาด 16 บิต โปรเซสเซอร์ตระกูล 8086 จะเป็นตัวแสดงสภาวะการทำงานของเครื่องคอมพิวเตอร์ที่ได้จากการเอ็กซีคิวส์ แต่ละบิตของแฟลกจะมีการทำงานดังต่อไปนี้

OF (overflow) เป็นตัวชี้ในกรทำงานเกินขนาด (overflow) ของบิตสูงสุดทางคณิตศาสตร์
DF (direction) เป็นตัวกำหนดทิศทางการเคลื่อนที่จากซ้ายไปขวาหรือขวาไปซ้ายของข้อมูลสตริง
IF (interrupt) เป็นตัวชี้อินเตอร์รัพท์ภายนอกเช่นคีย์บอร์ด ถ้า 1 ยอมรับ 0 ไม่สนใจ
TF (trap) บอกให้โปรเซสเซอร์ทำงานที่ละขั้นตอน ในส่วนของโปรแกรมดีบัก
SF (sign) เป็นตัวแสดงผลลัพธ์เครื่องหมายที่ได้จากการคำนวนทางคณิตศาสตร์ 0 = + ,1 = -
ZF (zero) เป็นตัวแสดงผลลัพธ์ที่ได้จากการคำนวนทางคณิตศาสตร์และลอจิกว่าเป็น 0 หรือไม่
AF (auxiliary) เป็นตัวทดช่วยหรือเก็บค่าตัวทดจากบิต 3 ไป บิต 4 การคำนวนแบบ BCD
PF (parity) เป็นตัวชี้ parity ของ 8 บิตต่ำของการทำงาน even = 1 , odd = 0
CF (carry) เป็นแฟลกตัวทดที่เก็บข้อมูลจากบิตสูงสุด (บิตซ้ายสุด) ตามการทำงานทางคณิตศาสตร์

BIT NO.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLAG					O	D	I	T	S	Z		A		P		C

สรุป

- คอมพิวเตอร์จะเก็บข้อมูลแตกต่างกัน 2 ชนิด คือ 0 (OFF) และ 1 (ON) รูปแบบการคำนวนทางคณิตศาสตร์ใช้กฎเกณฑ์ของไบนารี่
- ค่าของไบนารี่ที่แทนค่าในแต่ละบิตมีน้ำหนักดังนี้ $1111 = 2^3 + 2^2 + 2^1 + 2^0$ มีค่าเท่ากับ 15
- ค่าลบของไบนารี่ใช้หลักการของ 2's คอมพลีเม้นต์ โดยเปลี่ยนค่าเดิมเป็นตรงกันข้ามและบวกด้วย 1
- 1 K มีค่าเท่ากับ 1024 ไบท์
- ตัวอักขระ 1 ตัวมีค่าเท่ากับ 1 ไบท์ ประกอบด้วยเลขฐานสอง 8 ตัว และบิตตรวจสอบอีก 1 บิต
- เลขฐานสิบหกสามารถแทนเลขฐานสองได้ 4 บิต คือ 0-9 และ A-F แทนด้วยค่า 0000 - 1111
- รหัส ASCII เป็นรหัสแทนข้อมูลที่ใช้ในไมโครคอมพิวเตอร์
- หัวใจของระบบ PC คือไมโครโปรเซสเซอร์ สามารถเข้าถึงข้อมูลเป็นไบท์หรือเวิร์ด
- หน่วยความจำหลักของไมโครคอมพิวเตอร์มี 2 ชนิดคือ ROM , RAM
- โปรเซสเซอร์จะเก็บข้อมูลตัวเลขขนาด 1 เวิร์ดในลักษณะสลับไบท์สูงกับไบท์ต่ำ
- โปรแกรมภาษาแอสแซมบลี จะประกอบด้วยเซคเม้นต์ 1 หรือมากกว่า 1 เซคเม้นต์ สแตคเซคเม้นต์สำหรับเก็บค่าแอดเดรสย้อนกลับ ดาต้าเซคเม้นต์สำหรับข้อมูลและ Working area และโค๊ดเซคเม้นต์สำหรับคำสั่งในการเอ็กซีคิวส์
- รีจีสเตอร์ CS,DS,SS จะเป็นตัวจัดการของแอดเดรสของ CODE,DATA,STACK SEGMENT
- รีจีสเตอร์ IP จะเก็บข้อมูลของ OFFSET ADDRESS ของคำสั่งที่จะเอ็กซีคิวส์ต่อไป
- รีจีสเตอร์ SP และ BP ใช้งานร่วมกับรีจีสเตอร์ SS เพื่อเข้าถึงข้อมูลในหน่วยความจำสแตค
- รีจีสเตอร์ที่ใช้งานทั่วๆไปคือ AXZ , BX , DX ,CX ไบท์ซ้ายมือสุดเป็นไบท์สูงสุด และไบท์ขวามือสุดเป็นไบท์ต่ำสุด รีจีสเตอร์ AX คือ ACCUMULATOR ใช้สำหรับอินพุตเอ้าพุต และเป็นตัวหลักในการประมวลผล รีจีสเตอร์ BX (BASE REGISTER) รีจีสเตอร์ CX (COUNT REGISTER) DX (DATA REGISTER)
- อินเดครีจีสเตอร์สำหรับการขายแอดเดรสและใช้ในการเพิ่มหรือลดรีจีสเตอร์ SI และรีจีสเตอร์ DI ใช้สำหรับการทำงานของสตริง

- แฟลกรีจีสเตอร์ เป็นตัวชี้สภาวะการทำงานของคอมพิวเตอร์และผลลัพธ์ที่ได้จากการเอ็กซีคิวส์

แบบฝึกหัด

1-1. จงแทนค่าเลขฐานสิบต่อไปนี้ด้วยเลขฐานสอง

a) 5 b) 13 c) 21 d) 27 e) 32

1-2. จงบวกเลขฐานสองต่อไปนี้

a) 00010101 + 00001101 b) 00111110 + 00101000

c) 00011111 + 00000001 d) 01010101 + 00111111

1-3. จงหาค่า 2's คอมพลีเม้นต์ของเลขฐานสองต่อไปนี้

a) 00010011 b) 00111100 c) 00111001

1-4. จงนำค่าของเลขฐานสิบหกแทนข้อมูลดังต่อไปนี้

a) ASCII LETTER Q b) ASCII LETTER M

c) 01011101 d) 11110111

1-5. จงบวกเลขฐานสิบหกต่อไปนี้

a) 23A6 + 0022 b) 51FD + 26C4

c) 7779 + 0887 d) EABE + 26C7

1-6. จงอธิบายการทำงานของตัวโปรเซสเซอร์มีการทำงานอย่างไรในการประมวลผลข้อมูล

1-7. จงอธิบายการเข้าถึงหน่วยความจำของ PC มีเทคนิคอย่างไร

1-8. จงอธิบายความหมายของคำต่อไปนี้

a) SEGMENT b) OFFSET

c) ACCUMULATOR d) INDEX

e) BASE ADDRESS f) PHYSICAL ADDRESS

1-9. จงอธิบายการทำงานของแต่ละเซคเม้นต์ต่างๆของระบบ PC มีกี่ชนิด อะไรบ้าง

1-10. จงอธิบายการทำงานของรีจีสเตอร์แต่ละชนิดแตกต่างกันอย่างไร