

## บทที่ 1 : ระบบคอมพิวเตอร์

คอมพิวเตอร์เป็นเครื่องใช้ไฟฟ้าชนิดหนึ่งซึ่งสามารถทำงานตามคำสั่งของผู้ใช้งานได้ องค์ประกอบของคอมพิวเตอร์มีอยู่สองส่วนคือ

- ฮาร์ดแวร์ (Hardware) เป็นส่วนประกอบที่เป็นรูปธรรมสามารถจับต้องได้ ได้แก่ตัวเครื่องคอมพิวเตอร์ จอภาพ และคีย์บอร์ด เป็นต้น
- ซอฟต์แวร์ (Software) เป็นส่วนประกอบที่เป็นนามธรรม อยู่ในรูปของคำสั่งต่าง ๆ ที่จะให้ฮาร์ดแวร์ปฏิบัติตาม

### 1.1 โครงสร้างของฮาร์ดแวร์ของระบบคอมพิวเตอร์

โดยทั่วไปแล้วคอมพิวเตอร์แต่ละรุ่นแต่ละยี่ห้อ ก็จะมีการวางส่วนประกอบต่าง ๆ ไม่เหมือนกัน แต่ถ้าเรามองโครงสร้างของคอมพิวเตอร์ในรูปแบบ Module แล้วจะเห็นส่วนประกอบต่าง ๆ ที่มีรูปแบบเหมือนกันดังต่อไปนี้ (COAL1)

#### 1.1.1 หน่วยประมวลผลกลาง (Central Processing Unit : CPU)

หน่วยประมวลผลกลางจัดได้ว่าเป็นส่วนที่สำคัญที่สุดของคอมพิวเตอร์ เปรียบเสมือนเป็นสมองของเครื่องคอมพิวเตอร์ โดยทำหน้าที่ในการคำนวณค่าต่าง ๆ ตามคำสั่งที่ได้รับ และควบคุมการทำงานของส่วนประกอบอื่น ๆ ทั้งหมด ในระบบไมโครคอมพิวเตอร์ หน่วยประมวลผลกลางจะถูกสร้างให้อยู่ในรูปวงจรรวม (Integrated Circuit: IC) เพียงตัวเดียวทำให้ง่ายในการนำไปใช้งาน ภายในหน่วยประมวลผลกลางมีส่วนประกอบย่อย ๆ สามส่วนคือ

- หน่วยคำนวณทางคณิตศาสตร์และตรรกะ (Arithmetic and Logic Unit : ALU) เป็นหน่วยที่ทำหน้าที่ประมวลผลโดยใช้วิธีที่คณิตศาสตร์ เช่น บวก ลบ คูณ หาร หรือ ทำหน้าที่ประมวลผลทางตรรกะ เช่น AND OR NOT COMPLEMENT เป็นต้น รวมทั้งยังทำหน้าที่ในการเปรียบเทียบค่าต่าง ๆ อีกด้วย
- หน่วยเก็บข้อมูลชั่วคราว (Register) เป็นหน่วยความจำขนาดเล็ก ทำหน้าที่เป็นที่พักข้อมูล ชั่วคราวก่อนที่จะถูกนำไปประมวลผล โดยปกติแล้วในหน่วยประมวลผลกลางจะมี Register สำหรับเก็บข้อมูลไม่เกิน 64 ตัว การอ้างอิงข้อมูลของ Register จะมีความเร็วเท่ากับความเร็วของหน่วยประมวลผลกลาง เพราะเป็นหน่วยความจำส่วนที่อยู่ภายในตัวหน่วยประมวลผลกลางจึงไม่ต้องไปอ้างอิงถึงภายนอกหน่วยประมวลผล
- หน่วยควบคุม (Control Unit) เป็นเสมือนหน่วยบัญชาการของระบบคอมพิวเตอร์ทั้งหมด ทำหน้าที่กำหนดจังหวะการทำงานต่าง ๆ ของคอมพิวเตอร์ไม่เว้นแม้แต่ส่วนประกอบอื่น ๆ ของ CPU นอกจากนี้ยังทำหน้าที่ควบคุมการส่งข้อมูลระหว่างหน่วยต่าง ๆ ในคอมพิวเตอร์

### 1.1.2 หน่วยความจำ (Memory Unit)

เป็นหน่วยที่ทำหน้าที่เก็บข้อมูลของระบบคอมพิวเตอร์ ไม่ว่าจะเป็นตัวเลขหรือข้อความแม้กระทั่งคำสั่งต่าง ๆ ในโปรแกรมที่จะใช้สั่งงานระบบคอมพิวเตอร์ โดยทั่วไปแล้วหน่วยความจำจะถูกสร้างมาบน IC เพื่อให้มีความจุสูงแต่มีขนาดเล็ก ข้อมูลที่เก็บในหน่วยความจำจะมีสถานะเพียงแค่เปิดวงจร (0) หรือปิดวงจร (1) เท่านั้น หน่วยความจำสามารถแบ่งออกได้เป็นสองประเภทใหญ่ ๆ คือ

- ROM (Read Only Memory) เป็นหน่วยความจำส่วนที่ CPU สามารถอ่านข้อมูลออกมาใช้งานได้ตามกรรมวิธีปกติ แต่เมื่อต้องการจะเขียนข้อมูลลงไปจะต้องใช้วิธีพิเศษ ทำให้ต้องมีวงจรในการเขียนข้อมูลโดยเฉพาะ ข้อดีของหน่วยความจำแบบนี้ก็คือ ข้อมูลที่เขียนลงไปแล้วจะคงอยู่ไปตลอดแม้ว่าจะปิดเครื่องคอมพิวเตอร์ไปแล้วก็ตาม ดังนั้น ROM จึงมักจะถูกใช้เก็บโปรแกรมสำหรับเริ่มต้นการทำงานของเครื่องคอมพิวเตอร์
- RAM (Random Access Memory) เป็นหน่วยความจำที่ CPU สามารถอ่านเขียนข้อมูลได้ด้วยกรรมวิธีปกติของระบบ ทำให้สามารถเปลี่ยนแปลงข้อมูลได้ตลอดเวลา RAM จึงมักจะถูกนำไปใช้เก็บข้อมูลระหว่างการทำงานของระบบ แต่ RAM ก็มีข้อเสียคือข้อมูลที่เก็บไว้ทั้งหมดจะสูญหายไปทันทีที่หยุดจ่ายไฟให้กับหน่วยความจำ

### 1.1.3 หน่วยนำข้อมูลเข้าและหน่วยแสดงผล (I/O Unit)

เป็นหน่วยที่เปรียบเสมือนประสาทของเครื่องคอมพิวเตอร์ ทำหน้าที่รับการติดต่อจากภายนอกเข้าสู่ระบบ และแสดงผลที่ได้จากการทำงานของระบบออกสู่ภายนอก เช่น คีย์บอร์ด, จอภาพ, ลำโพง, Disk Drive เป็นต้น โดยปกติแล้วเรามักจะแบ่งความหรรษาของเครื่องคอมพิวเตอร์โดยดูจากหน่วยนำข้อมูลเข้าและหน่วยแสดงผลเป็นส่วนใหญ่ เช่น ขนาดของจอภาพ, ความจุของ Disk เป็นต้น

## 1.2 ระบบซอฟต์แวร์ของคอมพิวเตอร์

เมื่อคอมพิวเตอร์ไม่มีซอฟต์แวร์ก็ไม่ต่างไปจากเครื่องประดับราคาสูงชิ้นหนึ่งเท่านั้น โดยปกติแล้วราคาของระบบคอมพิวเตอร์ส่วนใหญ่จะเป็นราคาของซอฟต์แวร์ และเป็นส่วนที่จะกำหนดขนาดของระบบคอมพิวเตอร์ว่าจะอยู่ในตลาดได้นานเท่าใด

ความหมายของซอฟต์แวร์มีได้หลายแบบตามแต่จะใช้คำพูดใด ซึ่งความหมายโดยรวมก็คือ “ซอฟต์แวร์หมายถึงกลุ่มของคำสั่งที่กำหนดการทำงานของคอมพิวเตอร์เพื่อให้บรรลุจุดประสงค์ที่ต้องการ” การพัฒนาระบบซอฟต์แวร์ให้กับคอมพิวเตอร์เครื่องหนึ่ง มักจะมีการกำหนดประเภทของซอฟต์แวร์ออกเป็นสามประเภทคือ (COAL2)

### 1.2.1 ซอฟต์แวร์ระบบ

เป็นซอฟต์แวร์ส่วนที่ทำหน้าที่ควบคุมการทำงานของฮาร์ดแวร์โดยตรงเพื่อให้ระบบคอมพิวเตอร์โดยรวมมีประสิทธิภาพสูงสุด และยังทำหน้าที่ให้บริการการสั่งงานฮาร์ดแวร์ของซอฟต์แวร์ประเภทอื่น ๆ เพื่ออำนวยความสะดวกในเรื่องของการรวมรายละเอียดการสั่งงานที่ซับซ้อนให้เหลือเพียงการสั่งงานง่าย ๆ เช่น การอ่านไฟล์จากดิสก์ก็สามารถสั่งงานด้วยคำสั่งง่าย ๆ เพียงคำสั่งเดียวให้ซอฟต์แวร์ระบบสั่งงานฮาร์ดแวร์ส่วนที่ทำหน้าที่จัดการกับดิสก์ในเรื่องรายละเอียดต่อไป

### 1.2.2 System Library

เป็นส่วนที่รวบรวมการคำสั่งการทำงานที่สลับซับซ้อนขึ้นมาจากซอฟต์แวร์ระบบ ทำหน้าที่ให้บริการการทำงานกับซอฟต์แวร์ประยุกต์ต่อไป เช่น ในซอฟต์แวร์ระบบอาจจะมีคำสั่งแค่อ่านเขียนไฟล์ แต่ใน System Library อาจจะมีคำสั่งที่จัดการกับไฟล์หลาย ๆ ไฟล์ในรูปแบบของฐานข้อมูล เป็นต้น

### 1.2.3 ซอฟต์แวร์ประยุกต์

เป็นซอฟต์แวร์ที่พัฒนาขึ้นเพื่อใช้กับงานด้านใดด้านหนึ่งโดยเฉพาะ ซอฟต์แวร์ประยุกต์อาจจะเป็นซอฟต์แวร์ที่พัฒนาขึ้นมาเองหรือซื้อมาในรูปโปรแกรมสำเร็จรูปก็ได้ ปัจจุบันโปรแกรมสำเร็จรูปของระบบไมโครคอมพิวเตอร์มีอยู่มากมายในหลายด้าน เช่น Microsoft Office เป็นต้น

## 1.3 ความสัมพันธ์กันของส่วนประกอบต่าง ๆ

ความสัมพันธ์กันระหว่างฮาร์ดแวร์และซอฟต์แวร์ประเภทต่าง ๆ สามารถที่จะแสดงอยู่ในรูปของแผนภูมิหัวหอมได้ (COAL3) โดยฮาร์ดแวร์จะอยู่ชั้นในสุด ถัดมาจะเป็นส่วนของซอฟต์แวร์ระบบ System Library และซอฟต์แวร์ประยุกต์ตามลำดับ โดยระบบในชั้นนอกจะเรียกใช้การบริการของระบบในชั้นในกว่าลงไปและคอยให้บริการแก่ระบบในชั้นที่อยู่ถัดออกมาจากตัวเอง

## บทที่ 2 : ระบบเลขจำนวน

คอมพิวเตอร์เป็นเครื่องจักรที่กลไกการทำงานพื้นฐานเป็นสองสถานะ (Binary) คือเปิดวงจรกับปิดวงจร ซึ่งสามารถแทนสถานะดังกล่าวได้ด้วยตัวเลขโคตสองตัวคือ 0 กับ 1 ข้อมูลแบบอื่นของคอมพิวเตอร์จะเกิดจากการประกอบรวมกันของเลข 0 กับ 1 เท่านั้น เราเรียกระบบเลขจำนวนที่ประกอบด้วยตัวเลข 0 กับ 1 เท่านั้นว่า “เลขฐาน 2”

ส่วนการนับของมนุษย์โดยปกติแล้ว เราจะมีตัวเลขโคตอยู่สิบตัวคือ 0, 1, 2, 3, 4, 5, 6, 7, 8, และ 9 ซึ่งจะประกอบรวมกันเป็นระบบเลขจำนวนที่เรียกกันว่า “เลขฐาน 10” จะเห็นว่าระบบเลขจำนวนที่ใช้ในคอมพิวเตอร์มีความแตกต่างจากระบบเลขจำนวนที่มนุษย์ใช้กันโดยปกติ ดังนั้นเราจะต้องเรียนรู้ถึงทักษะในการคำนวณของระบบเลขจำนวนทั้งสองแบบรวมถึงวิธีการเปลี่ยนระบบเลขจำนวนไปมา

### 2.1 ความหมายของตัวเลขในหลักต่าง ๆ

ในระบบเลขฐานสิบนั้น ค่าของเลขโคต ณ ตำแหน่งใด ก็คือค่าของเลขโคตนั้นคูณด้วยสิบยกกำลังของตำแหน่งนั้น เช่น 12345 หมายความว่า ค่า 5 อยู่ในตำแหน่งหลักหน่วยซึ่งค่าของสิบยกกำลังของหลักหน่วยคือ  $10^0$  ค่า 4 อยู่ในตำแหน่งของหลักสิบ ( $10^1$ ) ค่า 3 อยู่ในตำแหน่งของหลักร้อย ( $10^2$ ) ค่า 2 อยู่ในตำแหน่งของหลักพัน ( $10^3$ ) และค่า 1 อยู่ในตำแหน่งของหลักหมื่น ( $10^4$ ) ซึ่ง 12345 สามารถเขียนอยู่ในรูปผลบวกทางคณิตศาสตร์ได้ (COAL4)

$$\begin{aligned}12345 &= (1 \times 10^4) + (2 \times 10^3) + (3 \times 10^2) + (4 \times 10^1) + (5 \times 10^0) \\ &= 10000 + 2000 + 300 + 40 + 5\end{aligned}$$

จะเห็นว่าเลขกำลังของสิบจะเริ่มต้นจากศูนย์ที่หลักหน่วย แล้วเพิ่มขึ้นหนึ่งทุกครั้งในหลักถัดมาทางด้านซ้ายมือ ในกรณีที่เลขเป็นจำนวนทศนิยม ให้เริ่มกำลังศูนย์ที่หลักหน่วย แล้วลดกำลังลงหนึ่งทุกครั้งในหลักถัดไปทางด้านขวามือ ส่วนทางด้านซ้ายมือก็จะเข้าไปในรูปแบบเดิม เช่น 12.34 จะสามารถเขียนได้เป็น

$$\begin{aligned}12.34 &= (1 \times 10^1) + (2 \times 10^0) + (3 \times 10^{-1}) + (4 \times 10^{-2}) \\ &= 10 + 2 + 0.3 + 0.04\end{aligned}$$

เราสามารถใช้หลักการเดียวกันนี้กับเลขฐานสองเพื่อหาค่าของจำนวนดังกล่าวในรูปของเลขฐานสิบ (ในความเป็นจริงแล้วสามารถที่จะนำไปใช้ได้กับเลขทุกฐาน) เช่น  $1011.01_2$  จะเขียนได้เป็น

$$\begin{aligned}1011.01_2 &= (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0) + (0 \times 2^{-1}) + (1 \times 2^{-2}) \\ &= 8 + 0 + 2 + 1 + 0 + 0.25 \\ &= 11.25\end{aligned}$$

## 2.2 การแปลงค่าจากเลขฐานสิบเป็นเลขฐานสอง

การแปลงเลขฐานสิบเป็นฐานสองจะมีขั้นตอนอยู่สองขั้นตอนคือ การแปลงเลขส่วนที่อยู่หน้าทศนิยมและการแปลงเลขส่วนที่อยู่หลังทศนิยม (COAL5)

การแปลงเลขในส่วนที่อยู่หน้าทศนิยม ให้นำเลขดังกล่าวมาหารด้วยสองไปเรื่อย ๆ จนกว่าจะได้ผลลัพธ์เป็นศูนย์ โดยการหารแต่ละครั้งจะได้เศษเป็น 0 หรือ 1 ลำดับของเศษที่เกิดขึ้นก็คือกำลังของเลขสอง กล่าวคือ เศษที่ได้จากการหารครั้งแรกจะเป็นเลขในหลัก  $2^0$ , เศษที่เกิดจากการหารครั้งที่สองจะเป็นเลขในหลัก  $2^1$  เรื่อยไป

ตัวอย่าง จงเปลี่ยนค่า  $13_{10}$  ให้เป็นเลขฐานสอง

$$\begin{array}{r} 2 \overline{) 13} \\ 6 \quad \text{เศษ } 1 \\ 3 \quad \text{เศษ } 0 \\ 1 \quad \text{เศษ } 1 \\ 0 \quad \text{เศษ } 1 \end{array}$$

$$\therefore 13_{10} = 1101_2$$

ส่วนการแปลงเลขหลังจุดทศนิยมนั้น จะใช้วิธีคูณตัวเลขนั้นด้วยสองไปเรื่อย ๆ จนกว่าจะมีเลขหลังจุดทศนิยมเป็นศูนย์ ซึ่งในการคูณแต่ละครั้งอาจจะมีการทดค่าหลังจุดทศนิยมขึ้นมาเป็นตัวเลข 1 หน้าจุดทศนิยมหรือไม่ก็ได้ ในการคูณแต่ละครั้งก็เท่ากับว่าเราเลื่อนการคำนวณจากหลักแรกหลังจุดทศนิยม ( $2^{-1}$ ) ไปยังหลักต่อไป

ตัวอย่าง จงเปลี่ยนค่า  $0.25_{10}$  ให้เป็นเลขฐานสอง

$$\begin{array}{r} 0.25 \\ \times \quad 2 \\ \hline 0.50 \\ \times \quad 2 \\ \hline 1.00 \\ \hline .01 \end{array}$$

$$\therefore 0.25_{10} = 0.01_2$$

ดังนั้นจากตัวอย่างข้างต้นสามารถสรุปได้ว่า  $13.25_{10} = 1101.01_2$

### 2.3 การบวกและการลบเลขฐานสอง

การบวกเลขฐานสองมีหลักการเหมือนกับการบวกเลขฐานสิบ การบวกเลขในฐานสิบนั้นเมื่อผลบวกในหลักใดมีค่ามากกว่า 9 ก็จะต้องมีการทดเลข 1 ไปยังหลักถัดไป ซึ่งหลักเกณฑ์การทดเลขนี้ยังสามารถใช้ได้กับเลขฐานสอง เพียงแต่ว่าเลขโดดที่สูงที่สุดของเลขฐานสองคือ 1 ดังนั้นถ้าผลบวกมีค่าเกิน 1 ก็จะมีการทดไปยังหลักถัดไปทางซ้าย(COAL6) รูปแบบการบวกเป็นดังนี้

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 0 \text{ ทด } 1$$

ตัวอย่าง จบบวกเลข  $1011.101_2$  กับ  $110.011_2$

$$\begin{array}{r} 1011.101 \\ + \quad 110.011 \\ \hline 10010.000 \end{array}$$

การลบเลขเป็นการดำเนินการที่ผกผันกับการบวก ในการลบ ถ้ามีการลบเลขที่มากกว่าจากเลขที่น้อยกว่า ต้องมีการขอยืมจากเลขในหลักถัดไปทางซ้ายมา 1 รูปแบบการลบเป็นดังนี้

$$0 - 0 = 0$$

$$0 - 1 = 1 \text{ ขอยืม } 1$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

ตัวอย่าง จงลบเลข  $1001.11$  กับ  $101.1$

$$\begin{array}{r} 1001.11 \\ - \quad 101.10 \\ \hline 100.01 \end{array}$$

## 2.4 การคูณและการหารเลขฐานสอง

การคูณและการหารของเลขฐานสอง ก็มีหลักการเช่นเดียวกับเลขฐานสิบ เพียงแต่มีสูตรคูณแค่แม่ 0 กับ 1 เท่านั้น (COAL7) อีกทั้งการหารด้วยศูนย์ก็ไม่มี ความหมายเช่นเดียวกับการหารในระบบเลขฐานสิบ ตารางการคูณ และการหารของระบบเลขฐานสองคือ

$$0 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 0 = 0$$

$$1 \times 1 = 1$$

$$0 \div 1 = 0$$

$$1 \div 1 = 1$$

ตัวอย่าง จงคูณเลขฐานสอง  $1.01 \times 10.1$

$$\begin{array}{r} \phantom{x}\phantom{0}1\phantom{0}01 \\ x\phantom{0}1\phantom{0}01\phantom{0}0 \\ \hline \phantom{x}\phantom{0}1\phantom{0}01 \\ \phantom{x}000 \\ \phantom{x}101 \\ \hline \phantom{x}11\phantom{0}001 \end{array}$$

จงหารเลขฐานสอง  $11001 \div 101$

$$\begin{array}{r} 101 \\ 101 \overline{) 11001} \\ \underline{101} \phantom{00} \\ 101 \phantom{00} \end{array}$$

## 2.5 เลขฐานแปดและเลขฐานสิบหก

ถึงแม้ว่าระบบคอมพิวเตอร์จะเข้าใจแต่ระบบเลขฐานสองเพียงอย่างเดียว แต่ในทางปฏิบัติจะเกิดปัญหากับผู้ใช้งานคอมพิวเตอร์เป็นอย่างมาก เนื่องจากระบบเลขฐานสองมีจำนวนเลขโคคน้อยจึงต้องมีจำนวนหลักมากขึ้นเพื่อแทนค่าตัวเลขต่าง ๆ ทำให้จดจำได้ยาก จึงมีความพยายามในการรวมหลักของเลขฐานสองหลาย ๆ หลักเข้าด้วยกันเป็นเลขฐานที่ใหญ่ขึ้น เพื่อให้ง่ายต่อการจดจำ ซึ่งการแปลงเลขฐานที่ได้นี้กลับเป็นเลขฐานสองจะทำได้ง่ายจากเนื่องจากแต่ละหลักของเลขฐานดังกล่าวแทนเลขฐานสองที่มีจำนวนหลักแน่นอน

โดยปกติแล้วเรามักจะรวมเลขฐานสองจำนวนสามหรือสี่หลักเป็นเลขฐานใหม่ (COAL8) เมื่อเรารวมเลขฐานสอง 3 หลักจะได้เลขที่มี 8 รูปแบบแตกต่างกันคือ 000, 001, 010, 011, 100, 101, 110, และ 111 ซึ่งเราสามารถใช้ตัวเลขโคค 0 ถึง 7 แทนเลขฐานสองในแต่ละแบบได้ ซึ่งระบบเลขจำนวนที่มีตัวเลขโคค 8 ตัวก็คือเลขฐานแปดนั่นเอง โดยที่เลขโคคแต่ละเลขของระบบเลขฐานแปดจะแทนรูปแบบของเลขฐานสองจำนวนสามหลักที่มีค่าเท่ากัน

ในทำนองเดียวกัน ถ้าเรามีเลขฐานสองจำนวนสี่หลัก ก็จะได้เลขฐานสองที่มี 16 รูปแบบแตกต่างกัน ซึ่งสามารถใช้เลขฐานสิบหกแทนแต่ละรูปแบบของเลขฐานสองได้ แต่เนื่องจากเรามีเลขโคคใช้งานกันแค่สิบตัว ดังนั้นจึงมีการนำตัว A – F มาแทนค่าเลขโคคที่มีค่า 10 – 15 แทน ทำให้เลขฐานสิบหกมีเลขโคคคือ 0 – 9 และ A – F

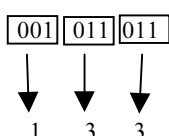
เลขฐานสิบ	เลขฐานสอง	เลขฐานแปด	เลขฐานสิบหก
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F



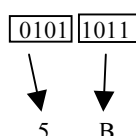
ในการแปลงเลขฐานสองให้เป็นเลขฐานแปดหรือฐานสิบหกนั้น เราจะต้องจับกลุ่มเลขฐานสองให้ได้จำนวนหลักตามที่เลขโคคของฐานที่เราจะแปลงไป เช่น จับกลุ่มสามหลักสำหรับการแปลงเป็นฐานแปด เป็นต้น การจับกลุ่มจะเริ่มจับจากหลักทางด้านขวามือสุดก่อน ในกรณีที่เหลือเลขไม่ครบจำนวนหลักที่ต้องการให้เติมศูนย์ไปทางด้านซ้ายมือเรื่อย ๆ จนกว่าจะได้จำนวนหลักที่ต้องใช้ จากนั้นจึงแปลงฐานเลขโดยแปลงทีละกลุ่มก็จะได้เลขฐานแปดหรือฐานสิบหกตามต้องการ

**ตัวอย่าง** จงแปลงเลขฐานสอง 1011011 ให้เป็นเลขฐานแปดและฐานสิบหก

แปลงเป็นเลขฐานแปด



แปลงเป็นเลขฐานสิบหก



$$\therefore 1011011_2 = 113_8 = 5B_{16}$$

## 2.6 บิต, ไบต์, เวิร์ด และนิบเบิล

ข้อมูลต่าง ๆ ในระบบคอมพิวเตอร์มักจะไม่ได้อ่านจากเลขฐานสองเพียงแค่หลักเดียว เพื่อความสะดวกในการเรียกและความกระชับของจำนวนที่จะต้องใช้เรียก (คงไม่สะดวกนักถ้าต้องมีการเรียกกันว่า เลขฐานสอง 32 หลัก) ดังนั้นจึงได้มีการตั้งชื่อเฉพาะเรียกกลุ่มของเลขฐานสองที่มีจำนวนหลักตั้งแต่หนึ่งหลักขึ้นไป โดยชื่อต่าง ๆ มีดังนี้ (COAL9)

เลขฐานสองหนึ่งหลักเรียก หนึ่ง“บิต” (bit)

4 บิต เท่ากับ หนึ่ง“นิบเบิล” (nibble) [ไม่ค่อยนิยมใช้กันนัก]

2 นิบเบิล เท่ากับ หนึ่ง“ไบต์” (byte)

2 ไบต์ เท่ากับ หนึ่ง“เวิร์ด” (word)

2 เวิร์ด เท่ากับ หนึ่ง“ดับเบิลเวิร์ด” (double word)

## 2.7 ระบบเลข 2' Complement

การคำนวณเลขฐานสองที่เราคุ้นเคยกัน มักจะเป็นเลขฐานสองที่มีค่าเป็นบวกอยู่เสมอ แต่ในความเป็นจริงเรามีการใช้งานจำนวนทั้งที่เป็นบวกและลบ ดังนั้นระบบคอมพิวเตอร์จึงต้องมีกลไกบางอย่างเพื่อระบุเครื่องหมายของตัวเลขต่าง ๆ ที่อยู่ในระบบ หนึ่งในวิธีการระบุเครื่องหมายของตัวเลขในระบบคอมพิวเตอร์ที่ได้รับความนิยมมากที่สุดก็คือ วิธี 2' Complement (COAL10)

วิธี 2' Complement จะเริ่มจากการกำหนดจำนวนหลักสูงสุดของตัวเลข (จำนวนบิตสูงสุด) กระบวนการทางคณิตศาสตร์ใด ๆ ก็ตาม ถ้าทำให้เกิดการทดเลขเลยบิตซ้ายสุดที่กำหนด เลขทดดังกล่าวจะหายไป เช่น ถ้า

กำหนดให้ตัวเลขทั้งหมด 4 บิต 1111 บวกกับ 0001 จะเท่ากับ 0000 ไม่ใช่ 10000 เป็นต้น แต่ถ้ามีการขอยืมจากหลักหน้าสุดที่เป็นศูนย์ ให้ถือเสมือนว่ามีหลักที่เป็นค่า 1 อยู่ถัดออกไปจากบิตสูงสุดที่กำหนดไว้ แล้วทำการขอยืมตามปกติ เช่น ถ้าต้องการลบตัวเลขขนาด 4 บิต 0100 ด้วย 0111 จะถือว่าเลข 0100 เสมือนเป็น (1)0100 แล้วจึงลบกันตามปกติ เป็นต้น (COAL11)

ข้อกำหนดข้อที่สองของ 2' Complement คือบิตที่ถูกกำหนดให้เป็นบิตนัยสำคัญสูงสุดที่สุดจะเป็นบิตที่บอกเครื่องหมายของตัวเลข เช่น กำหนดให้ใช้ตัวเลขขนาด 6 บิต ดังนั้นบิตที่ 5 (เริ่มนับทางขวาสุดเป็นบิตที่ 0) จะเป็นบิตที่ระบุเครื่องหมายของตัวเลขนั้น เป็นต้น โดยการระบุเครื่องหมายจะใช้มาตรฐานว่า “ถ้าบิตหน้าสุดเป็น 0 หมายถึงเป็นเลขบวก ถ้าบิตหน้าสุดเป็นหนึ่งหมายถึงเป็นเลขลบ” ดังนั้นจึงไม่สามารถใช้บิตทุกบิตที่กำหนดให้ในการเก็บค่าตัวเลขได้

ข้อกำหนดข้อที่สามของ 2' Complement คือ ค่าตัวเลขที่เก็บอยู่เมื่อบวกกับค่าที่มีเครื่องหมายตรงกันข้ามกันตามแบบวิธีการบวกเลขฐานสองปกติ จะต้องได้ผลลัพธ์เป็นศูนย์ (แต่จะมีบิตทดที่เหลือหายทางซ้ายมือสุด) เช่น ถ้ากำหนดให้เป็นตัวเลขขนาด 4 บิต 1011 จะเท่ากับค่า -5 เพราะถ้าบวกกับ 0101 แล้วจะได้ (1)0000 โดยเลขทศด้านหน้าสุดจะหายไป เป็นต้น

จากข้อกำหนดทั้งหมดข้างต้น ถ้ากำหนดให้ตัวเลขที่ใช้มีขนาด 1 ไบต์ จะสามารถเก็บค่าบวกระหว่าง 0 (00000000) ถึง 127 (01111111) และค่าลบระหว่าง -1 (11111111) ถึง -128 (10000000)

การนำเลข 2' Complement ไปบวกหรือลบกัน จะสามารถทำได้ตามวิธีการปกติ แต่จะผลลัพธ์สุดท้ายจะต้องเป็นไปตามข้อกำหนดของ 2' Complement ทั้งสามข้อ สำหรับการคูณและการหารนั้นจะต้องแยกค่าสัมบูรณ์ (Absolute) ออกมาคูณหารกัน แล้วจึงนำเครื่องหมายมาคิดภายหลัง แต่มีข้อควรระวังคือ ถ้าเป็นการคูณกันระหว่างตัวเลขขนาด 1 ไบต์ด้วยกัน ผลลัพธ์ที่ได้จะต้องใช้ที่เก็บขนาด 2 ไบต์ (ดูตัวอย่างในเรื่องการคูณเลขฐานสอง) ซึ่งมากกว่าที่จะเก็บได้ ดังนั้นถ้าต้องการเก็บผลลัพธ์ไว้เป็นขนาด 1 ไบต์ ก็จะต้องให้ตัวตั้งและตัวคูณมีขนาดไม่เกิน 1 นิบเบิล (4 บิต)

## 2.8 ระบบเลข BCD (Binary Code Decimal)

ระบบเลข BCD มีชื่อเรียกอีกชื่อหนึ่งว่า Packed Decimal เป็นวิธีการหนึ่งที่จะทำให้การแปลงเลขระหว่างฐานสองกับฐานสิบง่ายขึ้นโดยใช้วิธีเดียวกับการสร้างเลขฐานสิบหก โดยจัดกลุ่มให้เลขฐานสอง 4 บิต เป็นเลขฐานสิบ 1 หลัก แต่จำนวนเลขโดดของเลขฐานสิบมีค่าน้อยกว่ารูปแบบของเลขฐานสองที่เป็นไปได้ใน 4 บิต ดังนั้นจึงมีเลขฐานสองบางรูปแบบที่ไม่ได้ถูกใช้งาน สาเหตุที่ต้องจัดกลุ่มเลขฐานสองให้เป็น 4 บิตแทนที่จะเป็นค่าอื่น ๆ เพราะจำนวนเลขโดดของเลขฐานสองมีค่าเท่ากับ 10 ซึ่งอยู่ระหว่างเลขฐานสอง 3 บิตกับ 4 บิต นั่นเอง (COAL12)

ปกติแล้วเรามักจะไม่นำเลข BCD ไปคำนวณแบบซับซ้อน มักจะนำไปบวกลบกันเท่านั้นซึ่งการบวกลบกันของเลข BCD จะต้องทำทีละ 1 นิบเบิล และต้องปรับค่าผลลัพธ์เพื่อเลี้ยงเลขฐานสองที่ไม่ได้ใช้งานด้วย

## 2.9 มาตรฐาน IEEE 754

มาตรฐาน IEEE754 เป็นมาตรฐานที่ถูกกำหนดขึ้นโดยหน่วยงานชื่อ IEEE (...) เพื่อเป็นมาตรฐานในการเก็บเลขทศนิยม (Floating Point) ในระบบคอมพิวเตอร์ โดยมาตรฐาน IEEE754 จะเก็บเลขทศนิยมฐานสิบในรูปแบบ (COAL13)

$$(-1 \times S)1.M \times 2^{E-B}$$

โดยที่	S	เป็นตัวระบุเครื่องหมายของตัวเลข ถ้าเป็น 0 จะเป็นบวก ถ้าเป็น 1 จะเป็นลบ
	1.M	เป็นฐานของเลขยกกำลัง อยู่ในรูป 1.XXXX
	E	เป็นตัวยกกำลังของสอง ไว้สำหรับระบุตำแหน่งของทวินิยม
	B	เป็นค่า BIAS เพื่อให้ค่าของ E ไม่ติดลบ จะมีค่าคงที่สำหรับทุกหมายเลข (ค่าของ B จะถูกกำหนดมาแล้วคือ 127 สำหรับ IEEE754 ขนาด 32 bit)

เช่น 2.5 จะสามารถเขียนได้เป็น  $(-1 \times 0)1.010 \times 2^{128-B}$

ในการเก็บค่าต่าง ๆ จะเก็บอยู่ในรูป S-M-E โดย S จะมีขนาด 1 บิต ส่วน M และ E จะมีขนาดแตกต่างกันไปหลายขนาด ขึ้นอยู่กับความละเอียดที่ต้องการ

มาตรฐาน IEEE754 นอกจากจะสามารถแทนค่าเลขทศนิยมปกติแล้ว ยังมีข้อกำหนดที่สามารถใช้แทนสิ่งที่เรียกกันว่า NaN (Not a Number) เช่น ค่าอินฟินิตี้ ด้วย

การคำนวณเลขตามมาตรฐาน IEEE754 มีความซับซ้อนมาก ดังนั้นจึงไม่ขอกล่าวในที่นี้

## บทที่ 3 : ภาษาสำหรับเขียนโปรแกรมคอมพิวเตอร์

ดังที่ได้กล่าวมาแล้วว่า คอมพิวเตอร์สามารถทำงานได้กับระบบเลขฐานสองเพียงอย่างเดียว ซึ่งเลขฐานสองดังกล่าว ไม่ได้หมายถึงข้อมูลเท่านั้น แต่ยังหมายถึงคำสั่งต่าง ๆ ของเครื่องคอมพิวเตอร์ด้วย ดังนั้นถ้าเราต้องการเขียนโปรแกรมขึ้นมาสักหนึ่งโปรแกรม เราจะต้องเรียบเรียงคำสั่งต่าง ๆ ให้อยู่ในรูปของลำดับของเลขฐานสองต่าง ๆ ตามแต่ผู้ที่ผลิตเครื่องคอมพิวเตอร์นั้นจะกำหนดไว้ จะเห็นว่าการเขียนโปรแกรมแบบนี้ไม่มีความสะดวกเลย จึงมีผู้คิดค้นภาษาสำหรับเขียน โปรแกรมขึ้นมาเพื่อให้ง่ายในการเขียน โปรแกรม

### 3.1. ภาษาต่าง ๆ สำหรับคอมพิวเตอร์

ในบรรดาภาษาสำหรับเขียนโปรแกรมนั้น ภาษาสำหรับการเขียนโปรแกรมด้วยเลขฐานสองหรือฐานสิบหก เป็นภาษาที่เก่าแก่ที่สุดซึ่งเครื่องคอมพิวเตอร์สามารถจะเข้าใจและปฏิบัติตามได้โดยทันที เราเรียกภาษาแบบนี้ว่า “ภาษาเครื่อง” ซึ่งภาษาดังกล่าวไม่สื่อความหมายที่ชัดเจนกับมนุษย์ซึ่งเป็นผู้ที่เขียน โปรแกรม จึงมีการคิดค้นสัญลักษณ์ต่าง ๆ ขึ้นมาแทนที่ภาษาเครื่อง โดยสัญลักษณ์ที่คิดขึ้นในตอนแรกจะเป็นการแทนค่าภาษาเครื่องหนึ่งคำสั่งต่อสัญลักษณ์หนึ่งตัว เช่น 00111110 แทนว่า LD A เป็นต้น ซึ่งเรียกภาษาสัญลักษณ์แบบนี้ว่า “ภาษาโมนิค (Mnemonic)” หรือ “ภาษาแอสเซมบลีพื้นฐาน” แต่ทว่าสัญลักษณ์ที่ได้นี้ ก็ยังยากที่จะใช้เขียนโปรแกรม จึงได้มีการสร้างสัญลักษณ์คำสั่งใหม่ให้มีความง่ายขึ้น โดยเทียบเคียงสัญลักษณ์กับภาษาอังกฤษเพื่อให้สื่อความหมายมากที่สุด จึงเกิดเป็นภาษาสำหรับเขียน โปรแกรมขึ้นมากมายหลายภาษาขึ้นอยู่กับสัญลักษณ์และรูปแบบที่ใช้ เช่น **Fortran, C, Pascal, และ Basic** เป็นต้น (COAL14)

### 3.2. ระดับของภาษาสำหรับเขียนโปรแกรม

เมื่อมีการสร้างสัญลักษณ์แทนภาษาเครื่อง ทำให้มีการแบ่งระดับของภาษาสำหรับเขียนโปรแกรมออกเป็นระดับต่าง ๆ สองระดับคือ ภาษาระดับสูง และภาษาระดับต่ำ

ภาษาระดับสูงเป็นภาษาที่เน้นการสื่อความหมายกับผู้เขียน โปรแกรมเป็นหลัก ทำให้ภาษาในระดับนี้มีความง่ายในการเขียนและทำความเข้าใจ ส่วนภาษาระดับต่ำเน้นที่ความง่ายที่เครื่องคอมพิวเตอร์จะนำคำสั่งต่าง ๆ ไปปฏิบัติซึ่งมีอยู่ด้วยกันแค่สองภาษาคือ ภาษาเครื่องกับภาษาแอสเซมบลี

ในตำราบางเล่มได้กล่าวถึงภาษาระดับกลาง ซึ่งเป็นระดับพิเศษอีกระดับ โดยเป็นระดับที่มีความสมดุลกันระหว่างความง่ายในการเขียนกับความง่ายในการนำไปใช้เครื่องคอมพิวเตอร์ปฏิบัติ แต่ในทางปฏิบัติจริงเราไม่สามารถหาจุดสมดุลกันจริง ๆ ของคุณสมบัติทั้งสองได้ ดังนั้นจึงไม่สามารถแบ่งแยกระดับกลางออกมาได้อย่างเด็ดขาด ซึ่งภาษาที่ได้รับการกล่าวถึงว่าเป็นภาษาระดับกลางก็คือภาษา C

### 3.3. การแปลภาษาสำหรับคอมพิวเตอร์

ภาษาสำหรับเขียนโปรแกรมที่อยู่ในรูปของสัญลักษณ์ต่าง ๆ นั้น ไม่สามารถนำไปให้เครื่องคอมพิวเตอร์ปฏิบัติได้ทันที เพราะเครื่องคอมพิวเตอร์เข้าใจแต่ภาษาเครื่องเท่านั้น ดังนั้นเราจะต้องมีขั้นตอนต่าง ๆ ในการแปลสัญลักษณ์ต่าง ๆ ให้กลายเป็นภาษาเครื่อง ซึ่งขั้นตอนการแปลจะมีสองแบบคือ (COAL15)

- **Interpret** เป็นการแปลแบบคำสั่งต่อคำสั่ง โดยดึงสัญลักษณ์ของภาษาระดับสูงมาทีละคำสั่งแล้วแปลคำสั่งนั้น จากนั้นจึงส่งคำสั่งที่แปลได้ไปให้เครื่องคอมพิวเตอร์ทำงานทันที เมื่อคอมพิวเตอร์ทำงานเสร็จแล้วจึง จะเริ่มแปลคำสั่งถัดไปเรื่อย ๆ โดยไม่มีการเก็บภาษาเครื่องของคำสั่งที่แปลไปแล้ว ดังนั้นถ้ามีการวนที่คำสั่งใด ซ้ำแล้วซ้ำอีก ก็จะต้องแปลคำสั่งนั้นใหม่ทุกครั้งที่ทำงานผ่านคำสั่งนั้น ข้อดีของการแปลแบบนี้คือคำสั่งต่าง ๆ จะผ่านขั้นตอนต่าง ๆ อย่างสมบูรณ์ในขั้นตอนการแปลภาษา จึงสามารถพบข้อผิดพลาดได้ง่าย
- **Compile** เป็นกระบวนการแปลสัญลักษณ์ของภาษาระดับสูงเป็นภาษาเครื่องทีเดียวทั้งโปรแกรม แล้วจึงนำผลลัพธ์ที่ได้ทั้งหมดไปให้เครื่องคอมพิวเตอร์ปฏิบัติต่อเนื่องกันไปโดยไม่มีการแปลสัญลักษณ์ใด ๆ อีก ซึ่งภาษาเครื่องที่ได้อาจจะถูกเก็บไว้เพื่อนำกลับมาให้เครื่องคอมพิวเตอร์ทำงานได้อีกโดยไม่ต้องผ่านกระบวนการแปลภาษาอีกครั้ง สัญลักษณ์ของภาษาระดับสูงแต่ละตัวจะถูกแปลเป็นภาษาเครื่องเพียงครั้งเดียว ดังนั้นเมื่อเครื่องคอมพิวเตอร์ทำงานตามโปรแกรมที่เขียนไว้ จึงทำงานด้วยความเร็วสูงกว่าการแปลแบบ Interpret แต่วิธีนี้ก็มีข้อเสียคือ ข้อผิดพลาดบางประการของโปรแกรมจะตรวจพบเมื่อเครื่องคอมพิวเตอร์ทำงานตามโปรแกรมแล้วเท่านั้น ซึ่งเมื่อมีการแก้ไขให้ถูกต้อง ก็จะต้องมีการแปลสัญลักษณ์ทั้งหมดใหม่อีกครั้ง
- **Assembling** จริง ๆ แล้ว Assembling เป็นการ Compile โปรแกรมแบบหนึ่ง แต่ถูกตั้งชื่อใหม่ตามชื่อของภาษาที่ถูกนำมาแปลซึ่งก็คือภาษาแอสเซมบลี การทำงานของ Assembling มักจะอยู่ในรูปการแทนค่าสัญลักษณ์ต่าง ๆ โดยเทียบกับตาราง เพราะภาษาแอสเซมบลีเป็นการแทนค่าภาษาเครื่องในแบบคำสั่งต่อคำสั่งนั่นเอง

### 3.4. ภาษาแอสเซมบลี

ภาษาแอสเซมบลีจริง ๆ แล้วก็คือภาษานีโนนิกที่มีการเพิ่มเติมคำสั่งพิเศษในการจัดการเรื่องการอ้างอิงหน่วยความจำนั่นเอง ภาษาแอสเซมบลีจัดเป็นภาษาระดับต่ำภาษาหนึ่ง ซึ่งสัญลักษณ์ที่ใช้นี้มักจะถูกกำหนดโดยผู้ผลิต IC หน่วยประมวลผลกลาง ดังนั้นเมื่อมีการเปลี่ยนหน่วยประมวลผลกลาง ภาษาแอสเซมบลีที่ใช้ ก็จะต้องเปลี่ยนตามไปด้วย เมื่อเราพัฒนาโปรแกรมเป็นภาษาแอสเซมบลีแล้ว ก็ต้องนำไป Assembling ด้วยตัวแปลภาษาชื่อ Assembler ซึ่งจะแปลภาษาแอสเซมบลีให้เป็นภาษาเครื่องอีกทีหนึ่ง

ข้อดีของการใช้ภาษาแอสเซมบลีคือ (COAL16)

- สามารถสั่งงานระบบคอมพิวเตอร์ได้โดยตรง
- สามารถเขียนให้ได้โปรแกรมภาษาเครื่องที่มีขนาดเล็กที่สุดได้
- โปรแกรมภาษาเครื่องที่ได้จะทำงานได้รวดเร็ว

ข้อเสียของภาษาแอสเซมบลีคือ

- เป็นภาษาระดับต่ำทำให้พัฒนาโปรแกรมได้ยาก
- เหมาะกับการพัฒนาโปรแกรมที่ไม่ใหญ่มาก
- ไม่มีโครงสร้างข้อมูลในระดับสูง
- โปรแกรมที่พัฒนาขึ้นจะไม่สามารถนำไปใช้กับหน่วยประมวลผลกลางแบบอื่นได้

#### **บทที่ 4 : สถาปัตยกรรมของ 80x86**