

บทที่ 4

หน่วยประมวลผลกลาง

~~~~~

ซีพียูหรือหน่วยประมวลผลกลาง อาจจะเป็นโปรเซสเซอร์เพียงซีพียูเดียวหรืออาจเป็นโปรเซสเซอร์ต่อร่วมกับชิ้นส่วนไอซีอื่น ๆ ก็ได้ ในบทนี้จะได้อธิบายถึงโครงสร้างตลอดจนกลไกภายในของซีพียู เพื่อเป็นพื้นฐานที่จะนำไปใช้ในการศึกษาสถาปัตยกรรมของซีพียูในระดับสูงต่อไป เนื้อหาจะเริ่มต้นด้วย ส่วนประกอบพื้นฐานของซีพียู การประมวลผลของคอมพิวเตอร์ ปัจจัยที่มีผลต่อความเร็วในการประมวลผล สถาปัตยกรรมของระบบไมโครโปรเซสเซอร์ เทคโนโลยีของหน่วยประมวลผลกลาง การจัดการของโปรเซสเซอร์ และสุดท้ายจะอธิบายการจัดการของรีจิสเตอร์ เพื่อเสริมความเข้าใจการทำงานภายในของซีพียูให้มากขึ้น

คำว่า การประมวลผล (Process) หมายถึง ขั้นตอนการทำงานที่เปลี่ยนจากข้อมูลดิบไปเป็นสารสนเทศที่นำไปใช้ประโยชน์ ซึ่งขั้นตอนการทำงานนี้คอมพิวเตอร์จะต้องใช้ส่วนประกอบของคอมพิวเตอร์ 2 ส่วน คือ หน่วยประมวลผลและหน่วยความจำ

หน่วยประมวลผล เหมือนกับเป็นสมองของคอมพิวเตอร์ เป็นส่วนที่รวบรวมคำสั่งซึ่งได้มาจากผู้ใช้หรือโปรแกรมคอมพิวเตอร์ ในเครื่องคอมพิวเตอร์ส่วนบุคคลหน่วยประมวลผลจะประกอบไปด้วยหน่วยประมวลผลขนาดเล็ก (Microprocessor) 1 ตัว หรือมากกว่าขึ้นไป ซึ่งหน่วยประมวลผลขนาดเล็ก อาจจะทำด้วยซิลิกอนขนาดเล็กหรือวัสดุชนิดอื่น และมีวงจรรีเลย์เล็กทรอนิกส์เล็ก ๆ มากมายอีกด้วย หน่วยประมวลผลขนาดเล็กจะถูกเสียบเข้ากับแผงวงจร (Circuit board) ซึ่งแผงวงจรคือแผ่นสี่เหลี่ยมแข็งที่บรรจุไปด้วยวงจรที่เชื่อมต่อระหว่างหน่วยประมวลผลกับส่วนประกอบอื่น ๆ แผงวงจรที่หน่วยประมวลผลขนาดเล็กเชื่อมต่อนี้เรียกว่า มาร์เทอร์บอร์ด (Motherboard) หรือ เมนบอร์ด (Mainboard)

ในเครื่องคอมพิวเตอร์ที่มีประสิทธิภาพนั้น หน่วยประมวลผลประกอบด้วยชิปจำนวนมากบนแผงวงจร เมื่อพูดถึงซีพียู (CPU : Central Processing Unit) จะหมายถึงหน่วยประมวลผลของเครื่องคอมพิวเตอร์ อาจประกอบไปด้วยชิปเพียงตัวเดียว หรือแผงวงจรหลาย ๆ แผงก็ได้ ซึ่งซีพียูนี้เป็นส่วนที่สำคัญ และใช้เนื้อที่เพียงชนิดเดียวในเครื่องคอมพิวเตอร์

#### 4.1 ส่วนประกอบพื้นฐาน

ซีพียู (CPU) เป็นเหมือนสมองของเครื่องคอมพิวเตอร์เป็นที่ซึ่งข้อมูลจะถูกทำการประมวลผล ในไมโครคอมพิวเตอร์ (Microcomputer) ตัวซีพียูจะบรรจุชิปขนาดเล็ก ๆ ไว้ซึ่งเรียกว่าไมโครโปรเซสเซอร์

(Microprocessor) ซึ่งชิปเหล่านี้จะถูกบรรจุอยู่ในชิ้นคล้ายพลาสติกขนาดเล็กและมีการลากสายไฟเชื่อมต่อเข้าด้วยกัน ทุก ๆ ชิพียูจะมีส่วนประกอบพื้นฐานอยู่สองส่วนเสมอ คือ ส่วนควบคุม (Control Unit) และส่วนคำนวณและเปรียบเทียบ (Arithmetic Logic Unit)

หน่วยควบคุม (Control Unit) ทำหน้าที่ควบคุมการทำงานของเครื่องคอมพิวเตอร์ทั้งระบบ เช่น ควบคุมการทำงานของหน่วยความจำหลัก หน่วยรับข้อมูล หน่วยคำนวณและตรรกะ หน่วยแสดงผล และที่เก็บข้อมูลต่าง ๆ ดังนั้นการทำงานของหน่วยนี้จึงเปรียบเสมือนศูนย์กลางระบบประสาทที่ทำหน้าที่ควบคุมการทำงานของส่วนประกอบต่าง ๆ ของเครื่องคอมพิวเตอร์ โดยที่หน่วยควบคุมและชิพียูจะรับรู้คำสั่งต่าง ๆ ในรูปของคำสั่งภาษาเครื่องเท่านั้น ถ้าผู้ใช้เขียนโปรแกรมโดยใช้ภาษาระดับสูง (High Level Language) ก่อนที่จะสั่งให้คอมพิวเตอร์ทำงานจะต้องมีการแปลงเป็นภาษาระดับต่ำ (Low Level Language) เสียก่อน

หน่วยคำนวณและตรรกะ (Arithmetic and Logic Unit) หรือเรียกสั้น ๆ ว่า เอแอลยู (ALU) ทำหน้าที่ประมวลผลการคำนวณทางคณิตศาสตร์ ตลอดจนการเปรียบเทียบทางตรรกะทั้งหมด

เราได้ทราบมาแล้วว่าหน้าที่ของชิพียูคือการประมวลผลคำสั่งต่าง ๆ แต่ไม่สามารถเก็บโปรแกรมใหญ่ ๆ ได้ ถึงแม้ว่าชิพียูจะมีรีจิสเตอร์สำหรับเก็บข้อมูลได้ แต่ก็ยังมีปัญหาว่าข้อมูลที่รีจิสเตอร์จะสามารถเก็บได้นั้นมีขนาดเพียงไม่กี่ไบต์เท่านั้น ดังนั้นจึงจำเป็นต้องมีเนื้อที่มากเพียงพอที่จะเก็บข้อมูลจำนวนมากที่จะใช้ในการประมวลผล เนื้อที่ที่กล่าวถึงก็คือเมมโมรี (Memory) หรือหน่วยความจำ โดยลักษณะภายนอกของเมมโมรีจะประกอบไปด้วยชิป (Chip) บนมาเธอร์บอร์ด (Mother Board) หรืออาจจะเป็นแผงวงจรเล็ก ๆ ติดอยู่กับมาเธอร์บอร์ด ซึ่งจะทำให้ชิพียูสามารถเก็บและดึงข้อมูลจากเมมโมรีมาได้อย่างรวดเร็ว

## 4.2 การประมวลผลของคอมพิวเตอร์

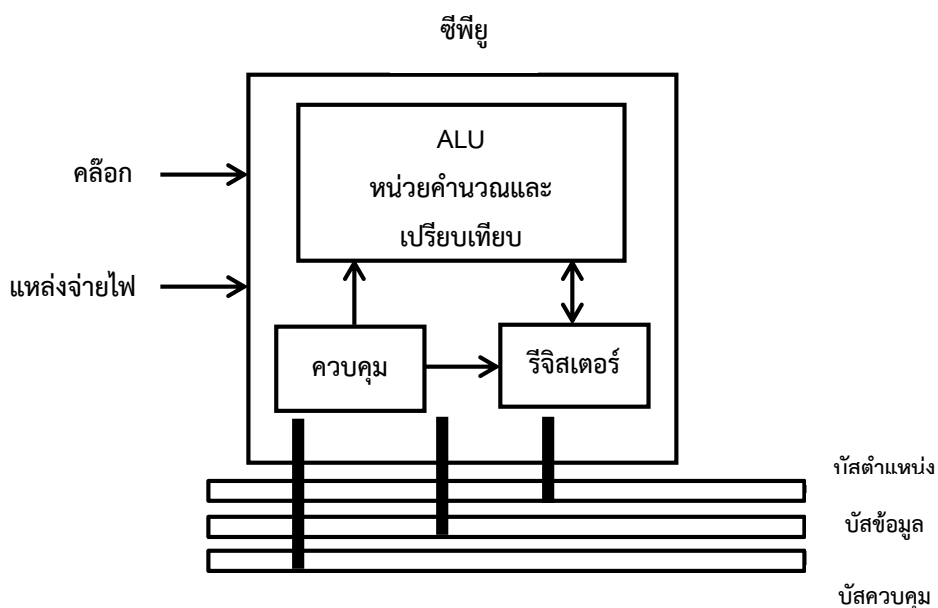
การประมวลผลของไมโครโพรเซสเซอร์อยู่ในตัวประมวลผลกลางหรือชิพียู ซึ่งแบ่งเป็นสองส่วนคือ คอนโทรลยูนิต (Control Unit) และอริทเมติกลอจิกยูนิต (Arithmetic Logic Unit) ภายในชิพียู คำสั่งจากโปรแกรมต่าง ๆ จะถูกแปลเปลี่ยนให้ตรงกับคำสั่งในอินสตรักชันเซต (Instruction Set) และกลายมาเป็นไมโครโค้ด (Micro Code) ชิพียูในตระกูลเดียวกันจะถูกสร้างให้เป็นแบบรองรับรุ่นก่อนหน้าหรือแบ็กเวิร์ดคอมแพทิบิลิตี (Backward Compatibility) ในส่วนของอินสตรักชันเซต (Instruction Set) ดังนั้นชิพียูใหม่ ๆ จะสามารถทำงานกับโปรแกรมเดียวกันที่ทำงานในชิพียูเก่า การจัดการเกี่ยวกับข้อมูลจะดำเนินการในเอแอลยู (ALU) ซึ่งมีรีจิสเตอร์ที่ใช้ในการพักข้อมูล หน่วยความจำแบบรวมเป็นหน่วยความจำแบบถาวร ซึ่งเก็บข้อมูลที่เป็นคำสั่งในหน้าที่ต่าง ๆ ไว้และมักถูกใช้ในขณะที่เปิดเครื่องครั้งแรก หน่วยความจำแบบแรมเป็นหน่วยความจำแบบชั่วคราว โปรแกรมและข้อมูลสามารถที่จะเขียนและลบข้อมูลในแรมได้ทุกเมื่อ ชิพียูจะเข้าถึงตำแหน่งต่าง ๆ ในหน่วยความจำได้โดยการใช้ตัวเลขที่เรียกว่าเมมโมรีแอดเดรส

ชิพียูจะทำงานตามคำสั่งที่ผู้ใช้ป้อนเข้าไป ซึ่งคำสั่งดังกล่าวจะถูกเก็บในหน่วยความจำหลัก ดังนั้นชิพียูจะต้องทำงานดังต่อไปนี้ (สตอลลิงค์ วิลเลียม, 2546, หน้า 463)

- 1) ดึงคำสั่งเข้ามา (Fetch Instruction) คือการที่ซีพียูอ่านคำสั่งมาจากหน่วยความจำหลักเข้ามาเก็บไว้ในตัวซีพียู
- 2) แปลความหมายของคำสั่ง (Interpret Instruction) คำสั่งจะถูกแปลความหมาย เพื่อจะได้ทราบว่าคำสั่งนั้นต้องการให้ทำงานอย่างไร
- 3) ดึงข้อมูล (Fetch Data) การประมวลผลคำสั่งเครื่องหนึ่งคำสั่งอาจจำเป็นต้องอ่านข้อมูลจากหน่วยความจำหลักหรืออุปกรณ์ไอโอเข้ามาด้วย
- 4) ประมวลผลข้อมูล (Process Data) การประมวลผลคำสั่งอาจเกี่ยวข้องกับการทำงานทางคณิตศาสตร์หรือตรรกะ
- 5) การบันทึกข้อมูล (Write Data) ผลการประมวลผลข้อมูลอาจต้องการให้บันทึกผลลัพธ์ที่ได้ไว้ในหน่วยความจำหลักหรืออุปกรณ์ไอโอ

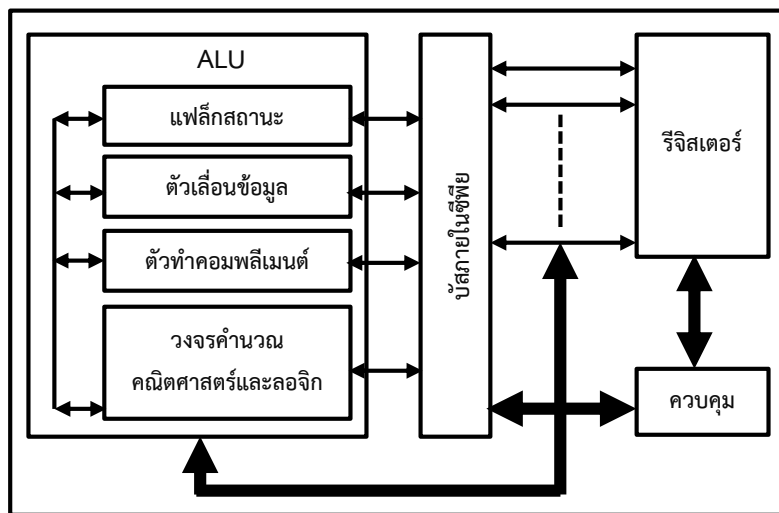
จากการทำงานดังกล่าว ซีพียูจะมีการติดต่อกับอุปกรณ์ภายนอก รอบ ๆ ตัว โดยจะติดต่อผ่านระบบบัสซึ่งได้แก่ บัสตำแหน่ง บัสข้อมูล และบัสควบคุม ในการประมวลผลคำสั่งแต่ละครั้งจะต้องดึงคำสั่งเข้ามาเก็บในตัวซีพียู หรืออาจต้องดึงข้อมูลเข้ามา ซีพียูอาจต้องจำตำแหน่งที่อยู่ของคำสั่งหรือข้อมูลปัจจุบัน ดังนั้นซีพียูจะต้องจัดเก็บคำสั่งข้อมูลและตำแหน่งไว้ชั่วคราวขณะที่คำสั่งนั้นกำลังได้รับการประมวลผล นั่นคือ ซีพียูมีความจำเป็นที่จะต้องมียูนิทความจำไว้ในตัวซีพียู ซึ่งเราเรียกหน่วยความจำนี้ว่า รีจิสเตอร์

สำหรับการประมวลผลข้อมูลก็จะมีหน่วยประมวลผล ซึ่งเรียกว่า หน่วยคำนวณและเปรียบเทียบหรือ เอแอลยู (ALU : Arithmetic and Logic Unit) ทำหน้าที่คำนวณทางคณิตศาสตร์และตรรกะ ในขณะที่ซีพียูมีการทำงานต่าง ๆ ดังกล่าว ก็จะมีหน่วยควบคุมทำหน้าที่ควบคุมการเคลื่อนย้ายข้อมูลและคำสั่งเข้าและออกจากซีพียู และควบคุมการทำงานของหน่วยเอแอลยู ดังได้แสดงโครงสร้างพื้นฐานของซีพียูในภาพที่ 4.1



ภาพที่ 4.1 การจัดองค์ประกอบพื้นฐานภายในซีพียู

จากภาพที่ 4.1 แสดงให้เห็นส่วนประกอบที่มีรายละเอียดเพิ่มขึ้นเล็กน้อยของซีพียู แสดงให้เห็นเส้นทางการส่งผ่านข้อมูล และการควบคุมทางตรรกะ รวมทั้งส่วนที่เรียกว่าบัสภายในซีพียู ซึ่งมีไว้ในการถ่ายเทข้อมูลระหว่างรีจิสเตอร์ต่าง ๆ และหน่วยเอแอลยู จะสังเกตเห็นความคล้ายคลึงกันระหว่างโครงสร้างภายในเครื่องคอมพิวเตอร์และโครงสร้างภายในตัวซีพียู ในทั้งสองกรณี ประกอบด้วยส่วนประกอบขนาดเล็กต่าง ๆ (สำหรับคอมพิวเตอร์ เช่น ซีพียู อุปกรณ์ไอโอ และหน่วยความจำ สำหรับซีพียู เช่น หน่วยควบคุม หน่วยเอแอลยู และรีจิสเตอร์) ที่เชื่อมต่อกันด้วยบัส



ภาพที่ 4.2 แสดงโครงสร้างภายในซีพียู

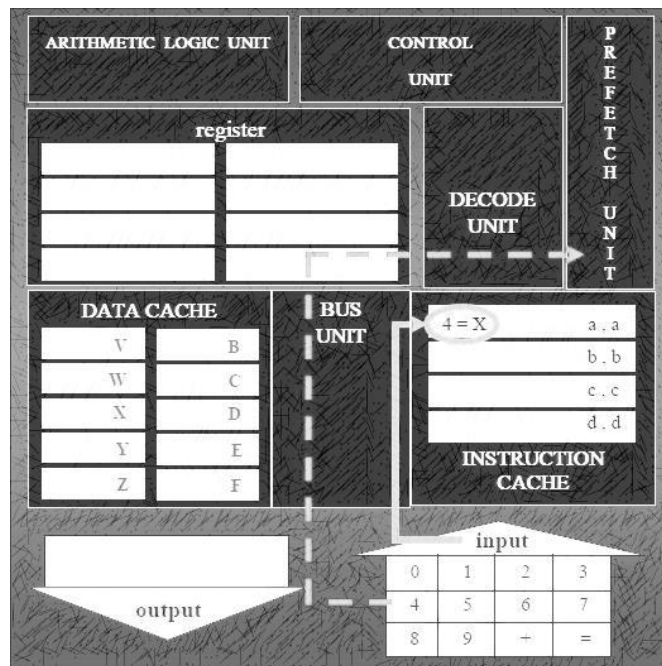
โดยทั่วไประบบการประมวลผลของซีพียูจะมีความซับซ้อนมากหรือน้อยขึ้นอยู่กับความสามารถ เช่น ความเร็วในการประมวลผล ขนาดหรือจำนวนบิตของสายสัญญาณข้อมูล และขนาดของหน่วยความจำ

ตัวอย่างต่อไปนี้เป็นกำลองวิธีการประมวลผลของการบวกเลขสองจำนวน (4+5) ซึ่งเป็นหลักการพื้นฐานของระบบการทำงานภายในของซีพียูทั่ว ๆ ไป

ขั้นตอนการทำงานจะเริ่มจากการป้อนข้อมูลจากส่วนรับข้อมูลอินพุต และวิธีการนำข้อมูลไปประมวลผลภายในซีพียู พร้อมทั้งผลลัพธ์ในแต่ละขั้นตอนของคำสั่งที่กระทำดังต่อไปนี้

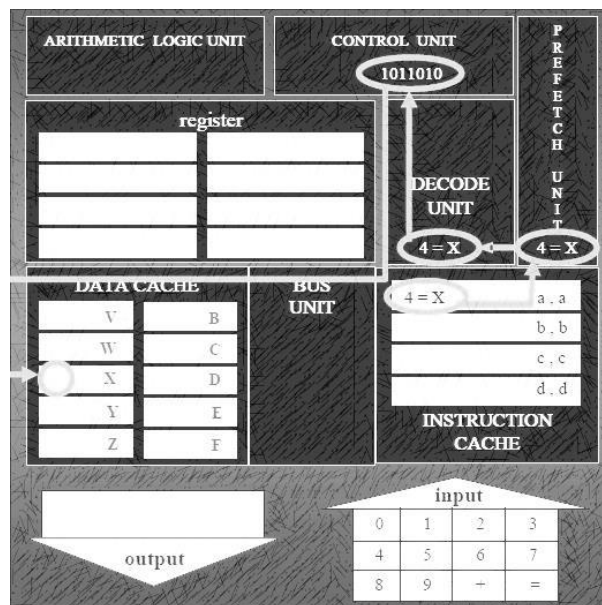
1) ขั้นตอนการรับค่า “4”

เมื่อป้อนค่า 4 จากแป้นพิมพ์ ส่วนการอ่านและตรวจสอบข้อมูล (pre-fetch) จะติดต่อไปยังหน่วยความจำหลักเพื่ออ่านข้อมูลนั้นเข้ามาในซีพียูผ่านทางส่วนควบคุมเส้นทางข้อมูล จากนั้นจะนำไปเก็บที่หน่วยความจำที่ใช้เก็บคำสั่ง และกำหนดให้มีค่ารหัสเป็น  $4=X$  เนื่องจากเป็นการเริ่มต้นทำงานใหม่ซีพียูจึงไม่สนใจหน่วยความจำแคช



ภาพที่ 4.3 ขั้นตอนการรับค่า “4” จากแป้นพิมพ์

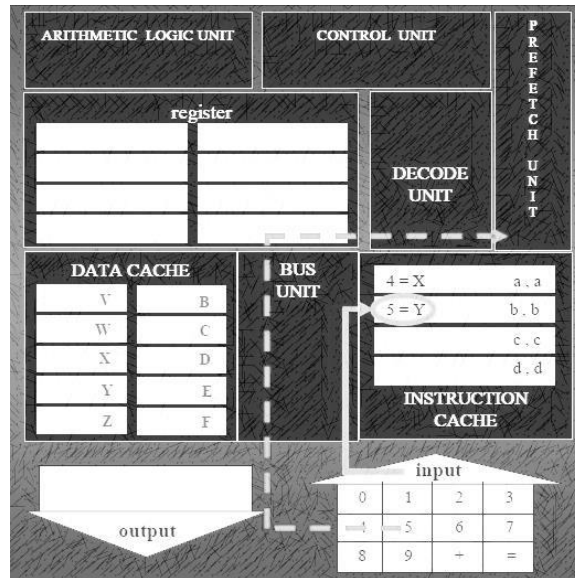
ในการประมวลผล ข้อมูลจะถูกส่งมายังส่วนการอ่านและตรวจสอบข้อมูล (pre-fetch) จากนั้นจะส่งผ่านไปให้ส่วนถอดรหัส (decoder) ซึ่งคำสั่ง 4=X จะถูกแปล (translated) หรือทำการเข้ารหัสข้อมูลตัวอักษรให้เป็นรหัสเลขฐานสอง (1011010) ที่มีความหมาย 4=X จากนั้นนำไปเก็บไว้ที่หน่วยความจำแคชที่ตำแหน่ง 4=X เพื่อรอคำสั่งในการประมวลผล



ภาพที่ 4.4 ขั้นตอนการรับค่า “4” จากแป้นพิมพ์ แล้วทำการเข้ารหัสข้อมูลตัวอักษร

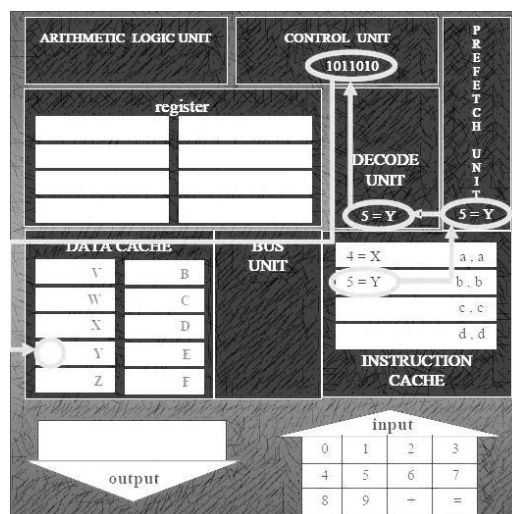
2) ขั้นตอนการรับค่า “5”

เมื่อกดแป้นพิมพ์เลข 5 ส่วนของการอ่านและตรวจสอบข้อมูล (pre-fetch) จะส่งสัญญาณสอบถามไปยังหน่วยความจำที่ใช้เก็บคำสั่งและหน่วยความจำหลัก ถ้าข้อมูลที่ป้อนเข้ามาเป็นข้อมูลที่แตกต่างจากที่มีในหน่วยความจำที่ใช้เก็บคำสั่งจะถือว่าเป็นข้อมูลคำสั่งใหม่ ซีพียูก็จะอ่านข้อมูลจากหน่วยความจำหลักโดยผ่านทางส่วนควบคุมเส้นทางข้อมูลไปเก็บไว้ในส่วนของหน่วยความจำที่ใช้เก็บคำสั่ง และกำหนดให้มีค่ารหัสเป็น 5=Y



ภาพที่ 4.5 ขั้นตอนการรับค่า “5” จากแป้นพิมพ์

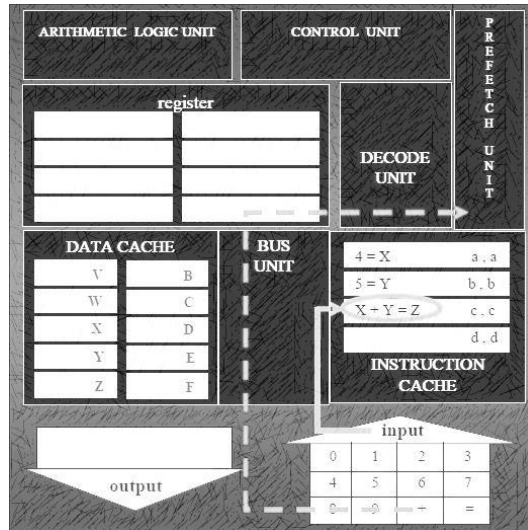
ขั้นตอนในการอ่านและตรวจสอบข้อมูลจะกำหนดให้ส่วนของหน่วยความจำที่ใช้เก็บข้อมูล คัดลอกรหัส “5=Y” และหน่วยความจำแคชจะเก็บค่าข้อมูลคือ 5 ที่ตำแหน่ง Y เอาไว้ จะเห็นได้ว่าที่ส่วนควบคุม (Control unit) จะมีค่ารหัสเลขฐานสองที่เหมือนกันกับครั้งที่แล้ว เพราะมีวิธีการเหมือนกัน จะต่างกันเฉพาะค่าข้อมูล



ภาพที่ 4.6 ขั้นตอนการรับค่า “5” จากแป้นพิมพ์ แล้วทำการเข้ารหัสข้อมูลตัวอักษร

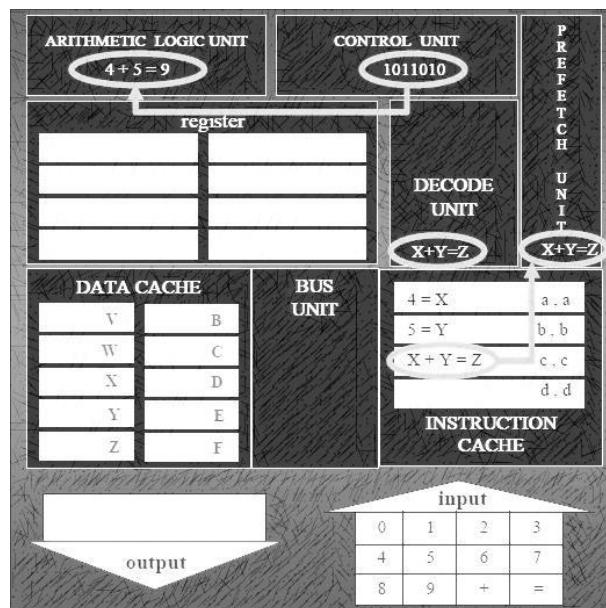
3) ขั้นตอนการรับค่าเครื่องหมายบวก “+”

เมื่อป้อนเครื่องหมายบวก “+” ส่วนรับข้อมูลแรกจะทำการตรวจสอบ หรือค้นหาข้อมูลที่ป้อนเข้ามาที่หน่วยความจำที่ใช้เก็บคำสั่งและหน่วยความจำหลัก ถ้าเป็นข้อมูลที่เป็นคำสั่งใหม่จะอ่านจากหน่วยความจำหลัก แล้วนำไปเก็บไว้ที่หน่วยความจำที่ใช้เก็บคำสั่ง และจะกำหนดเป็นรหัสคือ “X+Y=Z” ที่ตำแหน่งถัดมา



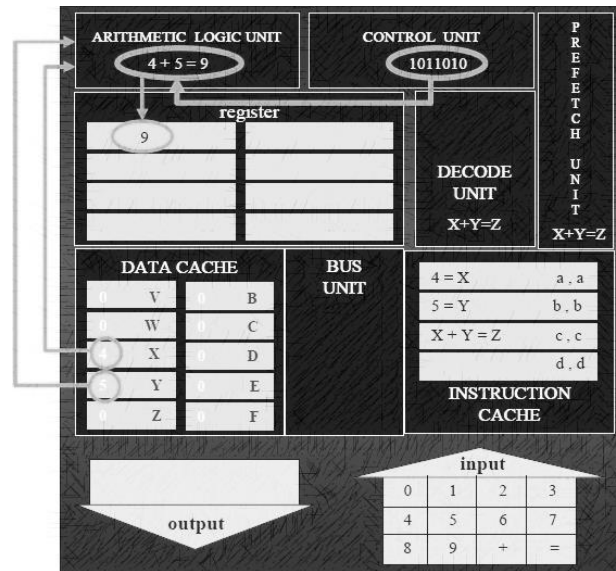
ภาพที่ 4.7 ขั้นตอนการรับค่าเครื่องหมายบวก “+” จากแป้นพิมพ์

ส่วนที่รับข้อมูลครั้งแรกจะทำหน้าที่อ่านข้อมูล “X+Y=Z” จากหน่วยความจำที่ใช้เก็บคำสั่ง และส่งไปยังส่วนถอดรหัส (Decoder) จากนั้นจะส่งออกไปให้ส่วนควบคุมเพื่อบอกให้ทราบและสั่งการให้ส่วนต่าง ๆ ที่รับผิดชอบทำงาน เช่น นำค่าข้อมูลไปเก็บไว้ที่หน่วยความจำแคช พร้อมกับคัดลอกและส่งไปยังส่วนของการคำนวณ (ALU)



ภาพที่ 4.8 การสั่งการให้ส่วนต่าง ๆ ที่รับผิดชอบทำงาน

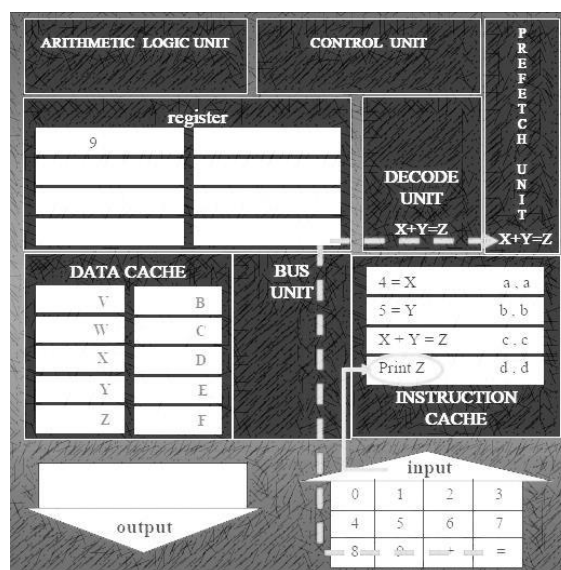
เมื่อส่วนควบคุมได้รับคำสั่งให้กระทำการบวก ก็จะส่งข้อมูลคำสั่งไปยังส่วนของการคำนวณ ซึ่งจะทำการอ่านค่าของ “X” และ “Y” และทำการบวกกัน หลังจากนั้นส่วนการคำนวณจะส่งผลลัพธ์ที่ได้จากการบวกคือ “9” ไปเก็บไว้ที่หน่วยความจำรีจิสเตอร์ (Register)



ภาพที่ 4.9 ขั้นตอนการบวกค่า “X” และ “Y”

4) ขั้นตอนการรับค่าเครื่องหมายเท่ากับ “=”

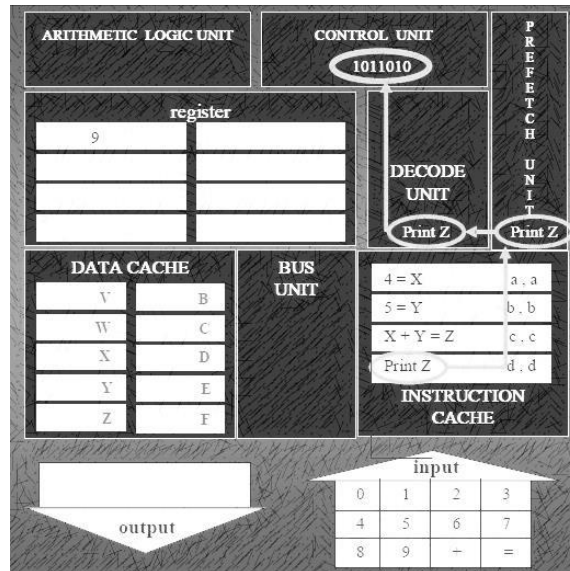
เมื่อป้อนเครื่องหมายเท่ากับ “=” ซีพียูจะส่งส่วนรับข้อมูลแรกให้ทำการหาข้อมูลก่อนว่าหน่วยความจำที่ใช้เก็บคำสั่งมีข้อมูลนี้อยู่หรือไม่ ถ้าไม่มีก็ให้ไปอ่านจากหน่วยความจำหลักผ่านเส้นทางสัญญาณข้อมูล แล้วนำไปเก็บไว้ที่หน่วยความจำแคชที่ตำแหน่งรหัส “Print Z”



ภาพที่ 4.10 ขั้นตอนการรับเครื่องหมายเท่ากับ “=”

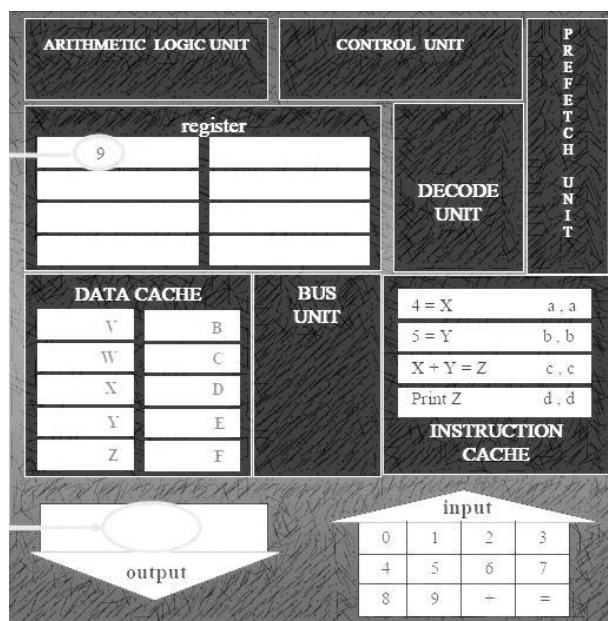


ส่วนรับข้อมูลแรกจะทำการคัดลอกข้อมูลรหัส “Print Z” จากหน่วยความจำแคชส่งไปยังส่วนถอดรหัสคำสั่งเพื่อทำการเข้ารหัสเป็นภาษาเครื่องที่เป็นเลขฐานสอง จากนั้นจะส่งค่าที่ได้ออกไปยังส่วนควบคุมเพื่อให้ส่วนควบคุมทำงานตามคำสั่ง



ภาพที่ 4.11 ทำการคัดลอกข้อมูลรหัส “Print Z” เพื่อทำการเข้ารหัส

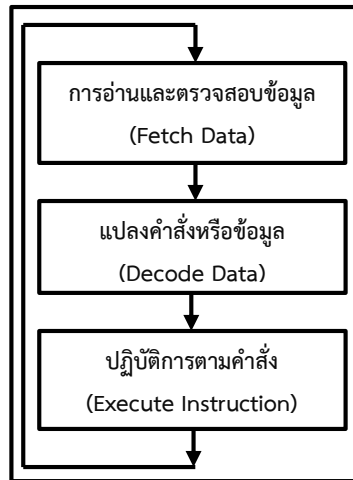
ขั้นตอนสุดท้ายของการบวกเลขจะได้ผลลัพธ์ของการบวกคือ ค่าข้อมูลของ “Z” ซึ่งเป็นผลมาจากการคำนวณและการกระทำตามคำสั่ง คือ การนำค่า “Z” ที่เป็นผลการบวกของ “X+Y” จะมีค่าเท่ากับ “9” ไปเก็บไว้ในหน่วยความจำรีจิสเตอร์ พร้อมกับนำผลที่ได้ออกไปแสดงที่เอาต์พุต



ภาพที่ 4.12 การเก็บค่าผลลัพธ์ไว้ในหน่วยความจำรีจิสเตอร์

สรุปขั้นตอนการทำงานของซีพียู

จากตัวอย่าง แสดงถึงขั้นตอนการทำงานของซีพียู โดยคำสั่งที่ถูกป้อนเข้าไปในระบบแต่ละคำสั่งจะมีกระบวนการอยู่ 3 กระบวนการใหญ่ ๆ ดังภาพที่ 4.13



ภาพที่ 4.13 แสดงขั้นตอนการทำงานของซีพียูในหนึ่งรอบคำสั่ง

- 1) กระบวนการอ่านและตรวจสอบข้อมูล (Fetch data) เป็นส่วนแรกที่ทำกรอ่านข้อมูลจากหน่วยความจำภายใน (Cache memory) หรือหน่วยความจำภายนอก (Main memory)
- 2) หลังจากอ่านข้อมูลเข้ามาแล้วจะส่งไปให้ส่วนแปลความหมายหรือเข้ารหัสคำสั่ง (Decode data) เพื่อแปลให้เป็นภาษาเครื่อง (Binary code)
- 3) หลังจากแปลคำสั่งเสร็จก็จะส่งไปยังส่วนควบคุมการทำงาน เพื่อนำไปปฏิบัติตามคำสั่งที่ได้รับมอบหมาย (Execute instruction) ต่อไป

### 4.3 ปัจจัยที่มีผลต่อความเร็วในการประมวลผล

ในการประมวลผลของซีพียูนับขึ้นกับปัจจัยหลายประการดังต่อไปนี้

- ขนาดของรีจิสเตอร์ หรือเรียกอีกชื่อหนึ่งว่า ขนาดของเวิร์ด เป็นตัววัดขนาดของข้อมูลที่เครื่องคอมพิวเตอร์สามารถที่จะทำงานได้ในหนึ่งช่วงเวลา
- ขนาดของแรม สามารถมีผลต่อความเร็ว เนื่องจากซีพียูจะสามารถเก็บขนาดของโปรแกรมหรือข้อมูลที่กำลังใช้งานอยู่ไว้ในหน่วยความจำได้ทันที โดยไม่ต้องพึ่งดิสก์ต่าง ๆ
- สัญญาณนาฬิกากระบบ เป็นตัวนำในการทำงานสำหรับซีพียู โดยอาศัยการสั้นของผลึกควอตซ์
- บัส มีบัสสองชนิดคือ ดาต้าบัส (Data bus) และแอดเดรสบัส (Address bus) ซึ่งบัสสองชนิดนี้อยู่บนเมนบอร์ด

- ความกว้างขวางของดาต้าบัส เป็นตัววัดว่าในหนึ่งช่วงเวลาสามารถที่จะขนถ่ายข้อมูลระหว่างอุปกรณ์ได้มากน้อยแค่ไหน
- โครงสร้างบัสของพีซี ซึ่งรวมถึงตั้งแต่ไอซาบัส (ISA bus) จนถึงพีซีไอบัส (PCI bus)
- ขนาดของแอดเดรสบัส เป็นตัววัดจำนวนของไบต์ของแรมที่ซีพียูสามารถเข้าถึงได้
- แคชเมมโมรี (Cache memory) เป็นหน่วยความจำที่มีความเร็วสูงที่เก็บข้อมูลและคำสั่งที่ถูกเรียกใช้บ่อยในซีพียู
- ซีพียูใช้แมธโครโพรเซสเซอร์ (Math Co-Processor) หรือการรวมส่วนคำนวณแบบโพลติง – พอยต์ (Floating point) ในการคำนวณ

#### 4.4 สถาปัตยกรรมของระบบไมโครโพรเซสเซอร์

คอมพิวเตอร์จะทำงานตามโปรแกรมหรือชุดคำสั่งที่เราป้อนเข้าไป โดยโปรแกรมหรือชุดคำสั่งดังกล่าวจะเป็นตัวบอกว่าคอมพิวเตอร์จะประมวลผลอย่างไร การที่มันจะทำงานได้นั้นจะต้องประกอบด้วยหน่วยประมวลผล หน่วยความจำ และอุปกรณ์อินพุต/เอาต์พุต สิ่งดังกล่าวเหล่านี้รวมกันเรียกว่า สถาปัตยกรรม (Architecture)

คำว่า “การประมวลผล (Process)” หมายถึง ขั้นตอนการทำงานที่เปลี่ยนจากข้อมูลดิบไปเป็นสารสนเทศที่นำไปใช้ประโยชน์ ซึ่งขั้นตอนการทำงานนี้คอมพิวเตอร์จะต้องใช้ส่วนประกอบของคอมพิวเตอร์ 2 ส่วน คือ หน่วยประมวลผล และหน่วยความจำ

หน่วยประมวลผล เหมือนกับเป็นสมองของคอมพิวเตอร์ เป็นส่วนที่รวบรวมคำสั่งซึ่งได้มาจากผู้ใช้หรือโปรแกรมคอมพิวเตอร์ ในเครื่องคอมพิวเตอร์ส่วนบุคคลนั้นหน่วยประมวลผลจะประกอบไปด้วยหน่วยประมวลผลขนาดเล็ก (Microprocessor) 1 ตัว หรือมากกว่าขึ้นไป ซึ่งหน่วยประมวลผลขนาดเล็กอาจจะทำด้วยซิลิกอนขนาดเล็กหรือวัสดุชนิดอื่น และมีวงจรรีเลย์ทรานซิสเตอร์เล็ก ๆ มากมายอีกด้วย หน่วยประมวลผลขนาดเล็กจะถูกเสียบเข้ากับแผงวงจร (Circuit board) แผงวงจรคือแผ่นสี่เหลี่ยมแข็งที่บรรจุไปด้วยวงจรที่เชื่อมต่อระหว่างหน่วยประมวลผลกับส่วนประกอบอื่น ๆ แผงวงจรที่หน่วยประมวลผลขนาดเล็กเชื่อมต่อนี้เรียกว่า มาร์เทอร์บอร์ด (Motherboard) หรือ เมนบอร์ด (Mainboard)

ในเครื่องคอมพิวเตอร์ที่มีประสิทธิภาพนั้น หน่วยประมวลผลประกอบด้วยชิป (Chip) จำนวนมากบนแผงวงจร เมื่อพูดถึงซีพียู (CPU : Central Processing Unit) จะหมายถึงหน่วยประมวลผลของเครื่องคอมพิวเตอร์ อาจจะประกอบไปด้วยชิปเพียงตัวเดียว หรือแผงวงจรหลาย ๆ แผงก็ได้ ซึ่งซีพียูนี้เป็นส่วนที่สำคัญและใช้เนื้อที่เพียงชนิดเดียวในเครื่องคอมพิวเตอร์

หน่วยประมวลผลผลกลางหรือซีพียู เป็นเหมือนสมองของเครื่องคอมพิวเตอร์เป็นที่ซึ่งข้อมูลจะถูกทำการประมวลผล ในไมโครคอมพิวเตอร์ (Microcomputer) ตัวซีพียูจะบรรจุชิปขนาดเล็ก ๆ ไว้ซึ่งเรียกว่าไมโครโพรเซสเซอร์ (Microprocessor) ซึ่งชิปเหล่านี้จะถูกบรรจุอยู่ในชั้นคล้ำยพลาสติกขนาดเล็กและมีการลาก

สายไฟเชื่อมต่อกันด้วยกัน ทุก ๆ ซีพียูจะมีส่วนประกอบพื้นฐานอยู่สองส่วนเสมอคือ ส่วนควบคุม (Control Unit) และส่วนคำนวณและเปรียบเทียบ (Arithmetic Logic Unit)

#### 1) หน่วยควบคุม (Control Unit)

ทำหน้าที่ควบคุมการทำงานของเครื่องคอมพิวเตอร์ทั้งระบบ เช่น ควบคุมการทำงานของหน่วยความจำหลัก หน่วยรับข้อมูล หน่วยคำนวณและตรรกะ หน่วยแสดงผล และที่เก็บข้อมูลต่าง ๆ ดังนั้นการทำงานของหน่วยนี้จึงเปรียบเสมือนศูนย์กลางระบบประสาทที่ทำหน้าที่ควบคุมการทำงานของส่วนประกอบต่าง ๆ ของเครื่องคอมพิวเตอร์ โดยที่หน่วยควบคุมและซีพียูจะรับรู้คำสั่งต่าง ๆ ในรูปของคำสั่งภาษาเครื่องเท่านั้น ถ้าผู้ใช้เขียนโปรแกรมโดยใช้ภาษาระดับสูง (High Level Language) ก่อนที่จะสั่งให้คอมพิวเตอร์ทำงานจะต้องมีการแปลงเป็นภาษาระดับต่ำ (Low Level Language) เสียก่อน

#### 2) หน่วยคำนวณและตรรกะ (Arithmetic and Logic Unit : ALU)

ทำหน้าที่ประมวลผลการคำนวณทางคณิตศาสตร์ ตลอดจนการเปรียบเทียบทางตรรกศาสตร์ทั้งหมด

### 4.5 เทคโนโลยีของหน่วยประมวลผลกลาง

ในช่วงกลางปี ค.ศ.1990 โพรเซสเซอร์มีการออกแบบสถาปัตยกรรมแยกออกเป็นสองรูปแบบคือ สถาปัตยกรรมแบบ CISC (Complex Instruction Set Computer) และ สถาปัตยกรรมแบบ RISC (Reduced Instruction Set Computer) โดยสถาปัตยกรรมการออกแบบซีพียูนั้นมีผลต่อการทำงานของคอมพิวเตอร์อย่างมาก ซึ่งจะกล่าวถึงสถาปัตยกรรมแต่ละแบบดังนี้

#### 4.5.1 สถาปัตยกรรมคอมพิวเตอร์แบบ CISC (Complex Instruction Set Computer)

##### 1) ประวัติและการพัฒนาระบบคอมพิวเตอร์แบบ CISC

ในยุคแรกของเครื่องคอมพิวเตอร์นั้น หน่วยความจำ (Memory) จัดว่าเป็นส่วนประกอบที่มีราคาแพงมาก จึงเป็นเหตุผลให้เกิดความคิดในหมู่นักออกแบบคอมพิวเตอร์ว่า สถาปัตยกรรมที่ดีที่สุดนั้นจะต้องเป็นสถาปัตยกรรมที่สามารถรองรับการใช้งานของโปรแกรมที่มีความยาวนานที่สุดเท่าที่จะทำได้ โดยการนำความยาวของโปรแกรมนั้นมาเทียบกับหน่วยความจำที่มีอยู่ ปัจจัยสำคัญที่ถูกนำมาใช้ในการวัดประสิทธิภาพของสถาปัตยกรรมคอมพิวเตอร์ที่วิศวกรคอมพิวเตอร์ออกแบบมานั้นจะพิจารณาจากความเร็ว (Speed) ของซีพียู และเวลาตอบสนอง (Response time) จากการประมวลผล โดยเฉพาะการทำงานของซีพียูนั้นจะวัดที่ความถี่ของสัญญาณนาฬิกา ดังนั้นสถาปัตยกรรมที่แตกต่างกันย่อมมีผลต่อประสิทธิภาพของระบบคอมพิวเตอร์โดยรวม อย่างไรก็ตามในปัจจุบันได้มีสถาปัตยกรรมคอมพิวเตอร์ที่สำคัญอยู่สามชนิดคือ แบบ CISC (Complex instruction set computer), RISC (Reduced instruction set computer) และ DSP (Digital signal processor)

เนื่องจากสถาปัตยกรรมคอมพิวเตอร์นั้นจะเกี่ยวข้องกับโครงสร้างภายในตัวซีพียู และลักษณะการทำงานของเครื่องคอมพิวเตอร์ที่ผู้ใช้เห็นได้ทั่วไป นอกจากนั้นรูปแบบ (Format) ในการจัดโครงสร้างข้อมูลและจำนวนชุดคำสั่ง ชนิดของการอ้างอิงตำแหน่งข้อมูลในหน่วยความจำรวมทั้งความสัมพันธ์ระหว่างอุปกรณ์ต่าง ๆ นั้นจะเป็นกรอบแนวความคิดให้พิจารณาถึงคอมพิวเตอร์ทั้งแบบ CISC และ แบบ RISC

## 2) คุณสมบัติของสถาปัตยกรรมคอมพิวเตอร์แบบ CISC

วิศวกรคอมพิวเตอร์มีความเชื่อว่าการออกแบบสถาปัตยกรรมคอมพิวเตอร์นั้นควรจะมีชุดคำสั่งและการอ้างอิงตำแหน่งข้อมูลในหน่วยความจำให้มากเข้าไว้เพื่อความพร้อมและรวดเร็วในการประมวลผลคำสั่งต่าง ๆ จึงเป็นที่มาของสถาปัตยกรรมคอมพิวเตอร์แบบ CISC ซึ่งมีคุณสมบัติที่สำคัญ ๆ ดังนี้

- ชุดคำสั่งมีขั้นตอนซับซ้อนมาก
- ยอมให้มีชุดคำสั่งการเรียก (Call), การกลับค่า (Return) และการส่งผ่านค่าพารามิเตอร์ (Parameter) ได้
- ชุดคำสั่งมีการอ้างอิงตำแหน่งในหน่วยความจำ (Addressing mode) มากกว่า 1 โหมด
- ชุดคำสั่งมีความยาวไม่คงที่
- ชุดคำสั่งมีรูปแบบ (Instruction format) มากกว่า 1 แบบ
- การย้าย (Transfer) ชุดคำสั่งและข้อมูล ระหว่างหน่วยความจำหลักกับซีพียูนั้นจะอยู่ห่างไกลกัน
- ระบบโครงสร้างของ CISC ในวงจรของชิป จะมีขนาดใหญ่ และมักจะไม่ใช้ระบบไปป์ไลน์ (Pipeline)
- ใช้เทคโนโลยีเกี่ยวกับหน่วยความจำหลัก (Main memory) แบบดั้งเดิม
- สามารถปฏิบัติงานคำนวณแบบซ้ำ ๆ ได้ดี
- สนับสนุนซีพียูแบบระบบหลายตัว (Multiprocessor)

ชุดคำสั่งของระบบ CISC นอกจากจะมีจำนวนมากแล้ว ยังมีขั้นตอนที่ซับซ้อนมากด้วย ที่จริงแล้วการที่มีชุดคำสั่งจำนวนมากนั้นจะทำให้ระบบ CISC ทำงานได้สะดวกและง่ายตายขึ้น ระบบ CISC ยอมให้มีการเรียก (Call), การกลับค่า (Return) และการส่งผ่านค่าของตัวแปรที่เปลี่ยนแปลงค่าได้ (Parameter) โดยจะเกิดขึ้นในขณะที่ระบบทำงาน เราเรียกการทำงานของระบบ CISC แบบนี้ว่า เป็นการทำงานแบบหลายชุดคำสั่ง (Multiple operation) ซึ่งทำให้ระบบ CISC ค่อนข้างจะมีความซับซ้อนมากยิ่งขึ้น

ชุดคำสั่งที่มีการอ้างอิงในหน่วยความจำ (Memory addressing mode) มากกว่า 1 โหมดจะทำให้สถาปัตยกรรมแบบนี้สามารถทำการคำนวณเพื่อหาตำแหน่งของข้อมูลได้หลายวิธี เช่น MC68000 และ MC68020 มีการอ้างอิงตำแหน่งในหน่วยความจำ (Addressing mode) 12 และ 18 แบบตามลำดับ ถึงแม้ว่าคอมพิวเตอร์จะมีวิธีการเข้าถึงข้อมูลได้หลายทาง แต่ก็จะเกิดปัญหาในเรื่องของเวลาในการรอข้อมูล และเวลาที่ใช้ในการคำนวณ

### 3) คุณลักษณะของสถาปัตยกรรมคอมพิวเตอร์แบบ CISC

การอธิบายสถาปัตยกรรมคอมพิวเตอร์แบบ CISC กับแบบ RISC ให้สามารถเข้าใจได้ง่าย จะใช้ตัวอย่างของไมโครโพรเซสเซอร์ตระกูลต่าง ๆ แต่โดยส่วนใหญ่แล้วจะพิจารณาในเรื่องของชุดคำสั่ง (Instruction set) การอ้างอิงตำแหน่งในหน่วยความจำ (Addressing mode) และจำนวนสัญญาณนาฬิกา (Clock) ต่อคำสั่ง (Instruction) ในการทำงาน

ในปี ค.ศ.1971 บริษัทอินเทล ได้ผลิตไมโครโพรเซสเซอร์ตัวแรกชื่อว่า 4004 ลักษณะของชิป จะมีหน่วยคำนวณทางคณิตศาสตร์และตรรกศาสตร์ (ALU) ขนาด 4 บิต และต่อมาบริษัทก็ได้ผลิตชิป 8086, 8088, 80186, 80286, 80386 และ 80486 ซึ่งเป็นชิปขนาด 32 บิต นอกจากนั้นยังได้มีการผลิตชิป 80586 หรือ เพนเทียม (Pentium) และ เมอเซด (Merced) ซึ่งเป็นไมโครโพรเซสเซอร์ขนาด 64 บิต

ส่วนบริษัทโมโตโรล่า ได้ผลิตไมโครโพรเซสเซอร์ขนาด 8 บิตตัวแรก ชื่อว่า MC6800 เมื่อปี ค.ศ.1974 ต่อมาในปี ค.ศ.1979 ได้ผลิตขนาด 16 บิต คือ 68000 ปี ค.ศ.1984 ผลิตขนาด 32 บิต คือ 68020 และตั้งแต่ปี ค.ศ.1988 ได้ผลิตตระกูล MC680X0 ได้แก่ MC68030 และ MC68040

ซึ่งโดยทั่วไปไมโครโพรเซสเซอร์แบบ CISC มักจะใช้ในอุตสาหกรรมของคอมพิวเตอร์ส่วนบุคคล (Personal computer : PC) เป็นส่วนใหญ่

#### ตารางที่ 4.1 คุณลักษณะบางอย่างของสถาปัตยกรรมแบบ CISC

| ลักษณะ          | IBM 370/168 | VAX 11/780 | IAPX -432 |
|-----------------|-------------|------------|-----------|
| ปีเริ่มต้น      | 1973        | 1978       | 1982      |
| จำนวนคำสั่ง     | 208         | 303        | 222       |
| ขนาดหน่วยความจำ | 420 Kbyte   | 480 Kbyte  | 64 Kbyte  |
| ขนาดชุดคำสั่ง   | 16-48       | 16-456     | 16-321    |

#### ตารางที่ 4.2 ลักษณะสถาปัตยกรรมของ CISC

| ลักษณะของสถาปัตยกรรม                                       | ชุดคำสั่งของ CISC Computer                                                       |
|------------------------------------------------------------|----------------------------------------------------------------------------------|
| ขนาดของชุดคำสั่งและรูปแบบของคำสั่ง                         | ชุดคำสั่งมีขนาดใหญ่ และรูปแบบคำสั่งเปลี่ยนแปลงไม่ได้                             |
| Addressing mode                                            | 12 - 24                                                                          |
| จุดประสงค์โดยทั่วไปของรีจิสเตอร์ และการออกแบบ Cache memory | คำสั่งและข้อมูลใช้หน่วยความจำ (Memory) ร่วมกัน และมีการใช้หน่วยความจำความเร็วสูง |
| Clock Rate และ CPI                                         | 35-50 MHz ในปี ค.ศ.1992<br>CPI อยู่ระหว่าง 2-15 CPI                              |
| CPU Control                                                | ส่วนใหญ่ใช้ไมโครโพรเซสเซอร์ในการควบคุมหน่วยความจำ แต่มักใช้ Hardwired Control    |

#### 4.5.2 สถาปัตยกรรมคอมพิวเตอร์แบบ RISC (Reduced Instruction Set Computer)

##### 1) ประวัติและการพัฒนาสถาปัตยกรรมคอมพิวเตอร์แบบ RISC

สถาปัตยกรรมคอมพิวเตอร์แบบ RISC เป็นสถาปัตยกรรมคอมพิวเตอร์แบบที่นิยมใช้ในปัจจุบัน และมีแนวโน้มในการใช้เพิ่มมากขึ้น ทั้งนี้เพราะสถาปัตยกรรมแบบนี้เป็นการช่วยให้คอมพิวเตอร์ทำงานได้เร็วขึ้น ซึ่งปกติหากพิจารณาจากจำนวนบิตที่เท่ากันระหว่างสถาปัตยกรรมคอมพิวเตอร์แบบ CISC และ แบบ RISC แล้ว จะเห็นได้ว่า คอมพิวเตอร์แบบ RISC ทำงานได้เร็วกว่าคอมพิวเตอร์แบบ CISC ถึงประมาณ 3 เท่า

หลักการอย่างง่าย ๆ ของคอมพิวเตอร์แบบ RISC คือ มีการออกแบบให้ซีพียูทำงานในวงรอบของสัญญาณนาฬิกา (Cycle) ที่แน่นอนโดยพยายามลดจำนวนคำสั่งลงให้เหลือเป็นคำสั่งพื้นฐานมากที่สุด แล้วใช้หลักการไปป์ไลน์ (Pipeline) คือ การทำงานแบบคู่ขนานชนิดเหลื่อมกัน (Overlap) โดยปกติแล้วการทำงานใน 1 ชุดคำสั่งจะใช้เวลามากกว่า 1 วงรอบสัญญาณนาฬิกา โดยในการทำคำสั่งเหล่านั้นมีการทำงานในลักษณะแถว (Pipe) และทำงานขนานกัน จึงทำให้ได้ค่าเฉลี่ยโดยรวมของเวลาเป็นคำสั่งละ 1 วงรอบสัญญาณนาฬิกา

เช่น สถาปัตยกรรมคอมพิวเตอร์แบบ RISC ได้แก่ MIPS รุ่น R2000 ได้มีการออกแบบชุดคำสั่งไว้อย่างชัดเจนว่ามีการทำงานแบบ 5 ขั้นตอน นั่นคือเป็นการทำงานแบบขนาน 5 ระดับด้วยกันคือ

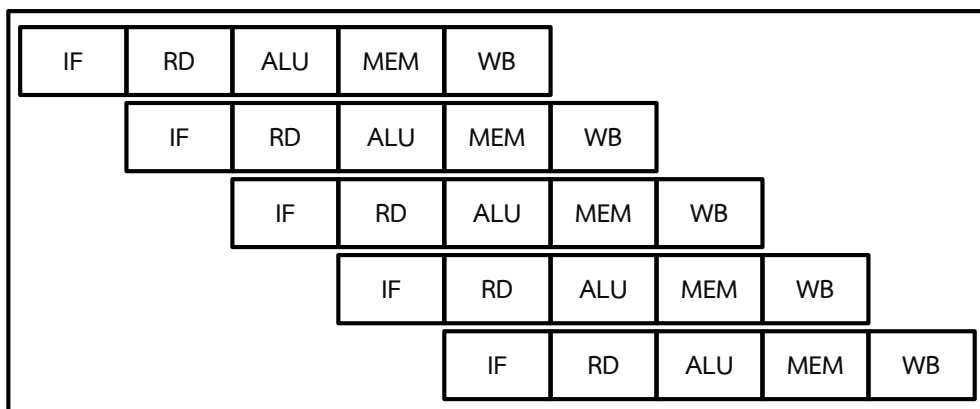
IF Instruction Fetch เป็นการเฟตช์คำสั่ง หรือดึงข้อมูลจากหน่วยความจำ

RDRead เป็นการอ่านโอเปอเรนด์ (Operand) จากรีจิสเตอร์ในซีพียู

ALU เป็นการคำนวณในหน่วยคำนวณและตรรกศาสตร์ตามคำสั่งนั้น ๆ

MEM การติดต่อกับหน่วยความจำ

WB การเขียนข้อมูลผลลัพธ์ลงในรีจิสเตอร์



ภาพที่ 4.14 แสดงการใช้คำสั่งแบบ 5 ระดับใน R2000

วิธีการทำงานของคำสั่งแบบคู่ขนานเช่นนี้ แสดงได้จากภาพที่ 4.14 ซึ่งจะมีผลทำให้การทำงานของคำสั่งเสร็จสิ้นได้ใน 1 วงรอบของสัญญาณนาฬิกา (Cycle) จากภาพที่ 4.14 จะใช้เวลาถึง 5 วงรอบ

สัญญาณนาฬิกา (Cycle) แต่ทุกช่วงการทำงานซีพียูจะทำงานขนานกับทั้ง 5 ส่วนนี้ด้วย ขณะที่เฟลท์คำสั่งหนึ่ง อยู่ภายในก็มีการทำ RD, ALU, MEM และ WB ของแต่ละคำสั่งที่ผ่านมา ดังนั้นการเฟลท์จึงเกิดขึ้นได้ทุกรอบ การออกแบบสถาปัตยกรรมโปรเซสเซอร์แบบ RISC นั้นจะเน้นการลดจำนวน และความซับซ้อนของคำสั่งภายใน (Instruction) ซึ่งในความเป็นจริงแล้วยังมีอีกหลายอย่างที่เป็นลักษณะของการออกแบบสถาปัตยกรรมแบบ RISC ได้แก่ การจัดการเกี่ยวกับ Cache, การทำ Pipeline, Superscalar และอื่นๆ

## 2) กฎเกณฑ์การออกแบบสถาปัตยกรรมแบบ RISC

การออกแบบสถาปัตยกรรมแบบ RISC โดยทั่วไปมีกฎเกณฑ์คือ ใช้ 1 ชุดคำสั่งต่อ 1 รอบสัญญาณนาฬิกา (One instruction per cycle) นับเป็นสิ่งสำคัญอันดับแรกในการออกแบบสถาปัตยกรรมแบบ RISC โดยจะพยายามทำงานให้เสร็จภายใน 1 รอบสัญญาณนาฬิกา (1 Clock cycle) ซึ่งก็มีการใช้ไปป์ไลน์ (Pipeline) มาช่วยในการทำงาน ดังนั้น ในการทำงานจริง ๆ อาจไม่ใช่ทำงานแค่ 1 คำสั่งเสร็จภายใน 1 รอบสัญญาณนาฬิกา แต่เป็นการนับเวลาการทำงานของโปรแกรม ซึ่งถ้าหากมีการทำงานหลายคำสั่งและหลายขั้นตอน แล้วคิดเป็นเวลาเฉลี่ยโดยรวม ก็จะได้ประมาณ 1 คำสั่งต่อ 1 รอบสัญญาณนาฬิกา วิธีทำเพื่อให้ได้ 1 คำสั่งต่อ 1 รอบสัญญาณนาฬิกาจะทำโดยใช้คำสั่งแบบธรรมดาต่าง ๆ แต่จะไม่ใช้เป็นการเพิ่มรอบสัญญาณนาฬิกาให้นานขึ้น วิธีการคือ กำหนดขนาดของชุดคำสั่ง (Instruction) ให้มีขนาดที่แน่นอน (Fixed instruction length) นั่นเอง ถ้าหากว่าจะทำให้การทำงานแต่ละคำสั่งเสร็จภายใน 1 รอบสัญญาณนาฬิกาได้นั้น ผู้ออกแบบ RISC ก็จะต้องจำกัดขนาดของคำสั่งด้วย โดยไม่ให้มีขนาดที่ยาวเกินไป ซึ่งโดยปกติแล้วจะใช้ขนาด 1 เวิร์ด (Word) ซึ่งขนาด 1 เวิร์ดนั้นจะมีขนาดไม่แน่นอนแล้วแต่เครื่อง แต่โดยทั่ว ๆ ไปซีพียูแบบ RISC จะมีขนาดของเวิร์ดเท่ากับ 32 บิต โดยใน 1 เวิร์ดก็จะกำหนดทุก ๆ อย่างรวมทั้งคำสั่งด้วยว่าจะทำการดึงโอเปอเรนด์ (Operand) จากไหน จะให้เก็บผลลัพธ์ไว้ที่ใด และคำสั่งถัดไปอยู่ที่ไหน เช่น  $Z = X + Y$ ; “X และ Y” คือ โอเปอเรนด์ (Operand), “+” คือ โอเปอเรชัน (Operation) และ Z คือ ผลลัพธ์ (Result)

คำสั่งในการเข้าถึงหน่วยความจำหลักจะใช้แค่การดึงข้อมูล (Load) และเก็บข้อมูล (Store) เท่านั้น สถาปัตยกรรมแบบ RISC เมื่อจะทำการจัดการกับคำสั่งต่าง ๆ จำเป็นจะต้องนำโอเปอเรนด์มาเก็บไว้ในรีจิสเตอร์ก่อน และในแต่ละคำสั่งจะถูกจำกัดไว้ที่ 1 เวิร์ด ซึ่งก็ไม่เพียงพอต่อการเก็บค่าของโอเปอเรนด์ต่าง ๆ ดังนั้น จึงใช้การอ้างอิงตำแหน่งในหน่วยความจำแทนการเข้าถึงหน่วยความจำนั้น แต่ต้องเสียเวลาอยู่พอสมควร ด้วยเหตุผลนี้จึงมีการกำหนดให้ใช้เพียงแค่ 2 คำสั่งเท่านั้นคือ การดึงข้อมูล และการเก็บข้อมูล เพื่อลดความหนาแน่นของการส่งผ่านข้อมูล (Traffic) ระหว่างหน่วยประมวลผลและหน่วยความจำ สำหรับเวลาในการดึงข้อมูล และเก็บข้อมูลนั้นก็ขึ้นอยู่กับขนาดของโอเปอเรนด์ที่ใช้ในการอ้างอิงตำแหน่งต่าง ๆ ส่วนใหญ่การอ้างอิงตำแหน่งข้อมูลสำหรับระบบ RISC นิยมใช้ 2 แบบด้วยกันคือ อ้างอิงผ่านรีจิสเตอร์โดยอ้อม (Register indirect) และการชี้ (Index) ซึ่งเป็นการอ้างอิงตำแหน่งผ่านรีจิสเตอร์ที่เป็นตัวชี้ (Index register)



### 4.5.3 การเปรียบเทียบสถาปัตยกรรมคอมพิวเตอร์แบบ CISC และ RISC

สถาปัตยกรรมคอมพิวเตอร์แบบ CISC นั้นจะมีจุดเด่นในเรื่องของการเขียนโปรแกรมเพราะผู้ใช้สามารถเขียนโปรแกรมได้ง่ายกว่า เนื่องจากโปรแกรมสามารถใช้คำสั่งที่มีอยู่มากมายในสถาปัตยกรรมคอมพิวเตอร์แบบ CISC นอกจากนี้ ขนาดของโปรแกรมบน CISC ก็มีขนาดเล็กกว่า RISC

การประมวลผลในสถาปัตยกรรมคอมพิวเตอร์ทั้ง 2 แบบ เช่น การหาผลลัพธ์ของ  $7*9$  ถ้าเป็นระบบ RISC จะมีเพียงคำสั่งบวก และลบ ในการหาผลลัพธ์ต้องทำการบวก 7 เข้าด้วยกัน 9 ครั้ง สมมติว่าแต่ละครั้งของการบวกใช้เวลา 1 สัญญาณนาฬิกา ดังนั้น จะใช้เวลาถึง 9 สัญญาณนาฬิกา และถึงแม้ว่าคอมพิวเตอร์จะเอา 9 มาบวกกัน 7 ครั้ง ก็ยังต้องใช้สัญญาณนาฬิกาถึง 7 ครั้ง แต่ถ้าเป็นสถาปัตยกรรมคอมพิวเตอร์แบบ CISC ซึ่งมีคำสั่งคูณไว้ให้แล้วและใช้สัญญาณนาฬิกา 5 ครั้งในการคูณแต่ละครั้ง การหาผลลัพธ์ก็จะทำเสร็จภายในสัญญาณนาฬิกาเพียงแค่ 5 ครั้ง แต่ในบางกรณีเช่น  $4*3$  สถาปัตยกรรมคอมพิวเตอร์แบบ RISC ใช้เวลาเพียง 3 หรือ 4 สัญญาณนาฬิกา แต่แบบ CISC ยังต้องใช้เวลาลงถึง 5 สัญญาณนาฬิกาอยู่ดี

ในการเปรียบเทียบความเร็วและประสิทธิภาพของ RISC และ CISC ทำได้ยาก เนื่องจากขึ้นอยู่กับงานที่ใช้ ซึ่งการเปรียบเทียบอย่างง่ายในตัวอย่างข้างต้นนั้น เพียงแค่เปลี่ยนค่าโอเปอเรนด์เท่านั้น ผลลัพธ์ที่ได้ก็ต่างกันแล้ว

ปัจจุบันนี้ไมโครโพรเซสเซอร์แบบ CISC มีความซับซ้อนของคำสั่งเพิ่มมากขึ้น และมีรูปแบบเปลี่ยนไปทาง RISC มากขึ้น โดยส่วนหนึ่งมาจากการลดปัญหาเรื่องความเข้ากันได้ของไมโครโพรเซสเซอร์รุ่นใหม่ ๆ กับไมโครโพรเซสเซอร์รุ่นเก่า ซึ่งดูเหมือนว่าการตัดแปลงคำสั่งสำหรับไมโครโพรเซสเซอร์ตระกูล 80X86 (CISC) ให้เป็นชุดคำสั่งแบบ RISC ที่เล็กกว่า เร็วกว่า และง่ายขึ้น สิ่งที่ได้ก็คือชุดคำสั่งของ RISC สำหรับไมโครโพรเซสเซอร์ตระกูล 80X86 (CISC) ที่เรียกชื่อว่า ชุดคำสั่ง RISC86 และยิ่งกว่านั้นไมโครโพรเซสเซอร์แบบ CISC ก็มีการทำงานแบบไปป์ไลน์ (Pipeline) และแบบการคาดเดาล่วงหน้า (Speculative Execution) ด้วย ซึ่งคุณสมบัติเหล่านี้แต่เดิมพัฒนาอยู่บนสถาปัตยกรรมแบบ RISC เท่านั้น ส่วนในทางกลับกันไมโครโพรเซสเซอร์แบบ RISC เองนั้นก็เริ่มมีการที่จะใช้คำสั่งที่ซับซ้อนและมากขึ้น ซึ่งก็คือมีแนวโน้มไปทางสถาปัตยกรรมแบบ CISC

สำหรับไมโครโพรเซสเซอร์ที่ใช้สถาปัตยกรรมแบบ RISC นั้น ได้แก่ SUN Sparc, IBM RS/6000, Motorola 88000 (เป็นไมโครโพรเซสเซอร์แบบ RISC ตัวแรกของบริษัท Motorola) ส่วนไมโครโพรเซสเซอร์ที่ใช้สถาปัตยกรรมแบบ CISC เช่น อินเทลตระกูล 80X86 ของบริษัท Motorola ตระกูล 68XXX เป็นต้น

### 4.5.4 ความเป็นมาของสถาปัตยกรรมระบบไมโครโพรเซสเซอร์ตระกูล 80X86

ไมโครโพรเซสเซอร์ตระกูล 80X86 เป็นไมโครโพรเซสเซอร์ที่พัฒนาขึ้นโดยบริษัท Intel โดยมีการพัฒนามาตั้งแต่รุ่น 4040 ซึ่งเป็นไมโครโพรเซสเซอร์ขนาด 4 บิต จนกระทั่งได้พัฒนาเป็นไมโครโพรเซสเซอร์รุ่น Pentium รายละเอียดคร่าว ๆ ของไมโครโพรเซสเซอร์รุ่นต่าง ๆ เป็นดังต่อไปนี้

- 1) 8086 เป็นไมโครโพรเซสเซอร์ขนาด 16 บิต รุ่นแรกที่ Intel ผลิตขึ้น สามารถอ้างหน่วยความจำได้ 1 MB มีรีจิสเตอร์ภายในขนาด 16 บิต
- 2) 8088 เนื่องจากในขณะนั้นอุปกรณ์ต่าง ๆ โดยมากเป็นอุปกรณ์ขนาด 8 บิต บริษัท Intel จึงได้ผลิตไมโครโพรเซสเซอร์ 8088 ซึ่งมีสถาปัตยกรรมภายในเหมือน 8086 แต่มีการติดต่อกับระบบภายนอกเป็นแบบ 8 บิต
- 3) 8086 เป็นหน่วยประมวลผลซึ่งเพิ่มการจัดการเกี่ยวกับอุปกรณ์รอบข้างเข้าไป เพื่อให้เป็นหน่วยประมวลผลสำหรับงานควบคุมต่าง ๆ
- 4) 80286 เป็นหน่วยประมวลผลขนาด 16 บิตที่มีความเร็วในการทำงานสูง ขยายขอบเขตการอ้างหน่วยความจำเป็น 24 เมกะไบต์ และมีการเพิ่มความสามารถในการจัดการหน่วยความจำ
- 5) 80386 เป็นหน่วยประมวลผลขนาด 32 บิต อ้างหน่วยความจำได้ถึง 4 กิกะไบต์ มีความสามารถในการจัดการหน่วยความจำที่ซับซ้อน และสามารถใช้หน่วยความจำแบบเสมือนได้ มาตรฐานชุดคำสั่งและกรรมวิธีในการจัดการหน่วยความจำของหน่วยประมวลผลรุ่นนี้ยังคงใช้เป็นมาตรฐานอยู่จนถึงปัจจุบัน ยกตัวอย่างเช่น แม้แต่ระบบปฏิบัติการ Windows 95 ยังสามารถนำมาทำงานได้บนหน่วยประมวลผลรุ่นนี้ แต่จะทำงานได้ช้ามากเท่านั้นเอง
- 6) 80386SX เป็นหน่วยประมวลผลซึ่งมีสถาปัตยกรรมภายในเหมือน 80386 แต่มีระบบบัสภายนอกเป็น 16 บิตหน่วยประมวลผลรุ่นนี้ได้รับการออกแบบขึ้นด้วยสาเหตุคล้ายกับ 8088
- 7) 80486 ได้รับการพัฒนาจาก 80386 โดยการที่บริษัท Intel เพิ่มหน่วยประมวลผลเลขทศนิยมเข้าไป
- 8) 80486SX เป็นหน่วยประมวลผลรุ่น 80486 ซึ่งตัดความสามารถของการประมวลผลเลขทศนิยมออก
- 9) Pentium, Pentium Pro, Pentium II เป็นหน่วยประมวลผลรุ่นล่าสุดของบริษัท Intel มีการเพิ่มความสามารถในการประมวลผลให้สูงขึ้น โดยเทคโนโลยีต่าง ๆ เช่น การประมวลผลแบบไปป์ไลน์ การประมวลผลแบบซูเปอร์สเกลาร์ เป็นต้น

#### 4.5.5 ลักษณะทั่วไปของไมโครโพรเซสเซอร์ 80X86

##### 1) ระบบบัส

ไมโครโพรเซสเซอร์ 80X86 มีแอดเดรสบัสขนาด 20 บิต ทำให้สามารถอ้างแอดเดรสได้ 1 เมกะไบต์ และมีบัสข้อมูลขนาด 16 บิต ซึ่งทำให้การอ่านและเขียนข้อมูลทำได้ครั้งละ 2 ไบต์ หน่วยประมวลผลทางคณิตศาสตร์และตรรกศาสตร์ภายใน 80X86 สามารถประมวลผลข้อมูลขนาด 16 บิต รีจิสเตอร์ภายในไมโครโพรเซสเซอร์ 80X86 มีขนาด 160 บิต

## 2) การจัดการหน่วยความจำ

ภายในหน่วยประมวลผล 80X86 มีรีจิสเตอร์ขนาด 16 บิต แต่มีแอดเดรสบัสนขนาด 20 บิต จึงทำให้หน่วยประมวลผลไม่สามารถเก็บตำแหน่งข้อมูลภายในหน่วยความจำได้ในรีจิสเตอร์เพียงตัวเดียว ดังนั้นการจัดเก็บตำแหน่งของข้อมูลภายในหน่วยความจำของ 8086 จึงต้องเก็บด้วยรีจิสเตอร์ 2 ตัว โดยใช้วิธีการจัดเก็บแบบเซกเมนต์ : ออฟเซต (segment : offset) แอดเดรสที่แท้จริง (Physical address) ขนาด 20 บิต จะถูกจัดเก็บด้วยรีจิสเตอร์ขนาด 16 บิต 2 ตัว ค่าที่เก็บในรีจิสเตอร์แรกเรียกว่า เซกเมนต์ ส่วนค่าที่เก็บในรีจิสเตอร์อีกตัวเรียกว่า ออฟเซต

การอ้างถึงตำแหน่งภายในหน่วยความจำแบบเซกเมนต์ : ออฟเซตนั้นอาจเปรียบได้เสมือนกับการที่เราแบ่งหน่วยความจำเป็นส่วนย่อย ๆ โดยส่วนย่อยนี้เราเรียกว่าเซกเมนต์ การที่เราอ้างถึงตำแหน่งใด ๆ เราจะอ้างถึงเซกเมนต์ที่ตำแหน่งข้อมูลนั้นอยู่ และระยะห่างของหน่วยความจำที่คิดเทียบกับจุดเริ่มต้นของเซกเมนต์ที่เราระบุไป

การแปลงแอดเดรสที่เก็บในรูปของเซกเมนต์ : ออฟเซต เป็นแอดเดรสขนาด 20 บิต มีขั้นตอนดังนี้

- เลื่อนบิตของค่าเซกเมนต์ไปทางซ้าย 4 บิต ดังนั้นจากข้อมูล 16 บิต เราจะได้ข้อมูล 20 บิต ที่มี 4 บิตทางขวาเป็น 0 ในทุกหลัก
- นำค่าออฟเซตมาบวกเข้ากับค่าเซกเมนต์ที่เลื่อนบิตแล้ว จะได้แอดเดรสขนาด 20 บิต ที่จะนำไปอ้างตำแหน่งที่แท้จริงของข้อมูล

ตัวอย่างเช่น 123Ah : 22B6h แปลงเป็นแอดเดรสขนาด 20 บิตได้ดังนี้

เลื่อนบิตของ 123Ah ไปทางซ้าย 4 บิต ได้ 123A0h

นำ 22B6h มาบวก 22B6h

จะได้แอดเดรสขนาด 20 บิตคือ 14656h

ในทางกลับกันที่ตำแหน่ง แอดเดรส 14656h ก็จะสามารถอ้างแบบเซกเมนต์ : ออฟเซต ได้เป็น 123Ah : 22B6h เช่นเดียวกัน แต่ตำแหน่งแอดเดรส 14656h สามารถอ้างแบบ เซกเมนต์ : ออฟเซต ค่าอื่นก็ได้เช่น 1465h : 0006h หรือ 1460h : 0056h และยังอ้างโดยใช้คู่ เซกเมนต์ : ออฟเซตคู่อื่น ๆ ได้อีกหลายคู่

ผลจากการอ้างแอดเดรสแบบ เซกเมนต์ : ออฟเซต ทำให้ลักษณะของหน่วยความจำที่ 8086 มองเห็นจะมีลักษณะเป็นส่วน ๆ ที่อ้างอิงตามค่าของเซกเมนต์ การจัดเรียงตัวของเซกเมนต์ต่าง ๆ ในหน่วยความจำจะจัดเรียงเป็นส่วน ๆ ที่มีการเหลื่อมกัน

เนื่องจากการอ้างถึงข้อมูลใด ๆ ในหน่วยความจำของไมโครโปรเซสเซอร์ 80X86 จะต้องอ้างตำแหน่งเป็นคู่เซกเมนต์ : ออฟเซต การอ้างถึงข้อมูลในตำแหน่งต่าง ๆ อาจทำให้ยุ่งยากเพราะต้องมีการระบุทั้งเซกเมนต์ : ออฟเซต ในไมโครโปรเซสเซอร์ 80X86 จึงได้ออกแบบรีจิสเตอร์พิเศษขึ้น 4 ตัวเพื่อใช้เก็บค่าเซกเมนต์ต่าง ๆ ที่กำลังใช้งานอยู่ในขณะนั้น กลุ่มของรีจิสเตอร์นั้นเรียกว่า เซกเมนต์รีจิสเตอร์ ซึ่งได้แก่ CS

(Code segment), DS (Data segment), ES (Extra segment) และ SS (Stack segment) เซกเมนต์ รีจิสเตอร์ ทั้ง 4 ตัวนี้จะใช้ประกอบกับค่าออฟเซตต่าง ๆ เพื่อระบุตำแหน่งของโปรแกรม (Code), ข้อมูล (Data) และสแต็ก (Stack) ส่วนรีจิสเตอร์ ES มีหน้าที่เก็บเซกเมนต์ของข้อมูลที่ใช้ในการส่งงานคำสั่งพิเศษบางประเภท เช่น คำสั่งเกี่ยวกับข้อความ

ดังนั้นเราจะเห็นได้ว่า ถึงแม้ไมโครโปรเซสเซอร์ 80X86 จะสามารถมองหน่วยความจำได้รวมถึง 1 เมกะไบต์ แต่โปรแกรมที่ทำงานอยู่จะมองเห็นหน่วยความจำได้พร้อม ๆ กันแค่เพียง 4 เซกเมนต์เท่านั้น นั่นคือ เซกเมนต์ของโปรแกรม เซกเมนต์ของข้อมูล 2 เซกเมนต์ และเซกเมนต์ของสแต็ก

ภายในหน่วยความจำของระบบไมโครโปรเซสเซอร์ 80X86 จะมีหน่วยความจำส่วนหนึ่งที่ถูกกันเนื้อที่ไว้สำหรับเป็นสแต็ก โดยเซกเมนต์ของสแต็กจะถูกชี้โดยรีจิสเตอร์ SS ลักษณะเฉพาะของหน่วยความจำแบบสแต็กคือการที่ระบบจะเก็บข้อมูลและอ่านข้อมูลออกไปแบบ เข้าก่อน ออกทีหลัง (First In Last Out : FILO) โดยอาจมองลักษณะเหมือนการวางซ้อนข้อมูลเหมือนซ้อนจาน ข้อมูลที่ถูกนำมาเก็บก่อนจะอยู่ทางด้านล่าง ข้อมูลถัดไปจะวางซ้อนอยู่ด้านบน ข้อมูลด้านล่างจะไม่สามารถอ่านออกไปได้ถ้ามีข้อมูลอื่นที่เก็บทีหลัง และยังไม่ได้อ่านออกไป ระบบจะใช้สแต็กในการเรียกโปรแกรมย่อย

#### 4.5.6 รายละเอียดของส่วนประกอบภายในไมโครโปรเซสเซอร์ 80X86

##### 4.5.6.1 หน่วยประมวลผลทางคณิตศาสตร์และตรรกศาสตร์ (ALU)

ไมโครโปรเซสเซอร์ 80X86 มี ALU ที่สามารถประมวลผลได้ครั้งละ 16 บิต ด้วยความสามารถในการประมวลผลทีละ 16 บิตนี้ ทำให้เราเรียกไมโครโปรเซสเซอร์ 80X86 ว่าเป็นไมโครโปรเซสเซอร์ขนาด 16 บิต

4.5.6.2 หน่วยความจำชั่วคราวรีจิสเตอร์ (Register) รีจิสเตอร์ใน 80X86 มีทั้งขนาด 16 บิต และ 8 บิต โดยจะแบ่งเป็นกลุ่ม ๆ ได้ดังนี้

##### 1) รีจิสเตอร์สำหรับใช้งานทั่วไป (General purpose Register)

รีจิสเตอร์ในกลุ่มนี้ ผู้เขียนโปรแกรมสามารถนำไปใช้งานได้ตามความต้องการ โดยในรีจิสเตอร์กลุ่มนี้จะมีรีจิสเตอร์ขนาด 16 บิต อยู่ 4 ตัวจะแบ่งได้เป็นรีจิสเตอร์ขนาด 8 บิตอีก 8 ตัว รีจิสเตอร์ขนาด 16 บิตและคูรีจิสเตอร์ขนาด 8 บิต มีดังต่อไปนี้

##### - รีจิสเตอร์ AX (Accumulator Register)

|    |    |
|----|----|
| AX |    |
| AH | AL |

##### - รีจิสเตอร์ BX (Base Register)

|    |    |
|----|----|
| BX |    |
| BH | BL |

- รีจิสเตอร์ CX (Counter Register)

|    |    |
|----|----|
| CX |    |
| CH | CL |

- รีจิสเตอร์ DX (Data Register)

|    |    |
|----|----|
| DX |    |
| DH | DL |

## 2) รีจิสเตอร์สำหรับการอ้างอิง (Index Register)

ไมโครโปรเซสเซอร์ 80X86 มีรีจิสเตอร์สำหรับการอ้างอิง 2 ตัว คือ SI (Source Index) และ DI (Destination) รีจิสเตอร์ในกลุ่มนี้ใช้สำหรับการอ้างอิงตำแหน่งแบบอ้างอิง และใช้ในคำสั่งที่เกี่ยวข้องกับข้อความ แต่ผู้ใช้สามารถนำไปใช้งานทั่วไปได้ด้วยเช่นกัน

## 3) รีจิสเตอร์สำหรับการชี้ (Pointer Register)

รีจิสเตอร์กลุ่มนี้คือ SP และ BP รีจิสเตอร์ SP ใช้ประกอบกับรีจิสเตอร์ SS มีหน้าที่ชี้ตำแหน่งปัจจุบันของสแต็ก รีจิสเตอร์ BP ส่วนใหญ่จะใช้เพื่อชี้ตำแหน่งของสแต็กเช่นเดียวกันแต่นิยมใช้ในส่วนของการส่งพารามิเตอร์ในโปรแกรมย่อย

## 4) เซกเมนต์รีจิสเตอร์ (Segment Register)

เซกเมนต์รีจิสเตอร์ทั้ง 4 ตัว คือ CS DS ES และ SS ใช้ประกอบกับค่าของออฟเซตเพื่อชี้ตำแหน่งของโปรแกรม ข้อมูลปกติ ข้อมูลพิเศษ และ สแต็ก ตามลำดับ

## 5) แฟล็ก (Flag)

ไมโครโปรเซสเซอร์ 80X86 จะเก็บลักษณะของผลลัพธ์ของการคำนวณทางคณิตศาสตร์ไว้ในแฟล็ก

## 6) รีจิสเตอร์อื่น ๆ ของระบบ

นอกจากรีจิสเตอร์ต่าง ๆ ที่ผู้ใช้สามารถกำหนดและใช้งานได้แล้ว ยังมีรีจิสเตอร์อีกกลุ่มหนึ่งซึ่งผู้เขียนโปรแกรมไม่สามารถเรียกใช้ได้ รีจิสเตอร์ในกลุ่มนี้ เช่น IP ซึ่งเป็นรีจิสเตอร์ที่ใช้ประกอบกับ CS เพื่อชี้ตำแหน่งของคำสั่งที่จะทำงานต่อไป หรือ IR ซึ่งเป็นรีจิสเตอร์ที่เก็บคำสั่งปัจจุบันที่ไมโครโปรเซสเซอร์อ่าน (Fetch) ขึ้นมาจากหน่วยความจำ

## 4.6 การจัดการของโปรเซสเซอร์

โปรเซสเซอร์มีหน้าที่หลักโดยทั่วไปดังนี้

- 1) Fetch Instructions : CPU จะต้องอ่านคำสั่งจากหน่วยความจำ
- 2) Interpret Instructions : คำสั่งที่อ่านมาจะถูก decode หรือแปลว่าจะทำอะไร
- 3) Fetch Data : อ่านข้อมูลจากหน่วยความจำ หรือ จาก I/O module เพื่อใช้ในการ execute คำสั่ง
- 4) Process Data : ต้องการการทำงานของ ALU เช่น การทำวงกลมคูณหาร หรือการเปรียบเทียบทางลอจิกต่าง ๆ
- 5) Write Data : ผลจากการ execute จะต้องบันทึกไว้ที่หน่วยความจำ หรือ I/O module

ส่วนประกอบหลักของ CPU คือ ALU CU และ Register โดย ALU จะทำงานด้านการประมวลผล โดย CU ทำหน้าที่ควบคุมการเคลื่อนไหวของข้อมูล และคำสั่งที่จะเข้าหรือออกจาก CPU นอกจากนี้ใน CPU ยังต้องการหน่วยความจำภายในสำหรับที่จะเก็บพักข้อมูลชั่วคราว จะเห็นว่า CPU จะเชื่อมต่อกับระบบภายนอกด้วย System bus ต่าง ๆ และภายใน CPU เองก็จะติดต่อกันด้วย Internal CPU Bus เช่นกัน

## 4.7 การจัดการของรีจิสเตอร์

ภายใน CPU นั้นจะมีกลุ่มของ register ที่แบ่งประเภทตามการทำงานดังนี้คือ

- 1) User – Visible Register : เป็น register ที่โปรแกรมเมอร์รับรู้ และนำไปใช้ในการเขียนโปรแกรมภาษาเครื่องหรือภาษาแอสเซมบลีได้
- 2) Control and status register : เป็น register ที่ถูกใช้โดย CU เพื่อควบคุมการทำงานของ CPU และกำหนดสิทธิในการครอบครอง CPU

ในที่นี้เราจะพิจารณาถึง Control และ Status Register ซึ่งเป็น register ที่ผู้เขียนโปรแกรมทั่วไปไม่สามารถจะใช้ได้ มันจะถูกใช้โดยคำสั่งเครื่องซึ่งจะถูก execute ในขั้นตอนการทำงานของระบบปฏิบัติการเท่านั้น

อย่างไรก็ตามในเครื่องที่แตกต่างกันก็จะมีจัดการ register ที่แตกต่างกันด้วย อย่างไรก็ตามเราสามารถแบ่งประเภท register ออกเป็น 4 register ที่จำเป็นสำหรับการ Execute คำสั่งดังนี้คือ

- Program Counter (PC) : เก็บ address ของคำสั่งที่จะถูก fetch ต่อไป
- Instruction Register (IR) : เก็บคำสั่งที่ถูก fetch มา
- Memory Address Register (MAR) : เก็บตำแหน่ง หรือ Address ของหน่วยความจำ
- Memory Buffer Register (MBR) : เก็บข้อมูลที่จะเขียนลงหน่วยความจำหรือข้อมูลที่อ่านมาจากหน่วยความจำ ซึ่ง MAR จะต้องต่อโดยตรงเข้ากับ Address bus และ MBR ก็จะต้องต่อตรงเข้ากับ data bus

นอกจากนี้ CPU ทุกตัวจะต้องมี register ที่ชื่อว่า Program Status Word (PSW) ที่เก็บสถานะของข้อมูลหรือตัวแปร ณ ขณะนั้น เช่น

- Sign : เก็บ Sign bit ของผลลัพธ์ในการประมวลผลครั้งล่าสุด
- Zero : จะถูก Set เมื่อผลลัพธ์ของการประมวลผลเป็น 0
- Carry : จะถูก Set เมื่อ operation เป็นการบวกแล้วได้ผลลัพธ์ที่ต้องทดไปบิตข้างหน้า หรือจะถูก Set เมื่อ operation เป็นการลบแล้วต้องมีการยืมจากบิตข้างหน้ามา
- Equal : จะถูก Set เมื่อผลการเปรียบเทียบระหว่างตัวแปร 2 ตัวเท่ากัน
- Overflow : ใช้ในกรณีที่มีการคำนวณต่าง ๆ แล้วเกิด Overflow ขึ้น
- Interrupt Enable/ Disable : ใช้สำหรับกำหนดว่ายอมหรือไม่ยอมให้ Interrupt
- Supervisor : ระบุว่าจะให้ CPU execute คำสั่งในส่วนของผู้ควบคุมระบบ (Supervisor) หรือผู้ใช้ (user) หรือเป็นการกำหนดสิทธิในการใช้ CPU นั้นเอง

## 4.8 บทสรุป

หน่วยประมวลผล เหมือนกับเป็นสมองของคอมพิวเตอร์ เป็นส่วนที่รวบรวมคำสั่งซึ่งได้มาจากผู้ใช้หรือโปรแกรมคอมพิวเตอร์

ซีพียู (CPU : Central Processing Unit) หมายถึง หน่วยประมวลผลของเครื่องคอมพิวเตอร์ อาจจะประกอบไปด้วยชิปเพียงตัวเดียว หรือแผงวงจรหลาย ๆ แผงก็ได้ ซึ่งซีพียูนี้เป็นส่วนที่สำคัญ และใช้เนื้อที่เพียงนิดเดียวในเครื่องคอมพิวเตอร์

ซีพียูจะมีส่วนประกอบพื้นฐานอยู่สองส่วนเสมอ คือ ส่วนควบคุม (Control Unit) และส่วนคำนวณและเปรียบเทียบ (Arithmetic Logic Unit)

หน่วยควบคุม (Control Unit) ทำหน้าที่ควบคุมการทำงานของเครื่องคอมพิวเตอร์ทั้งระบบ เช่น ควบคุมการทำงานของหน่วยความจำหลัก หน่วยรับข้อมูล หน่วยคำนวณและตรรกะ หน่วยแสดงผล และที่เก็บข้อมูลต่าง ๆ ดังนั้นการทำงานของหน่วยนี้จึงเปรียบเสมือนศูนย์กลางระบบประสาทที่ทำหน้าที่ควบคุมการทำงานของส่วนประกอบต่าง ๆ ของเครื่องคอมพิวเตอร์ โดยที่หน่วยควบคุมและซีพียูจะรับรู้คำสั่งต่าง ๆ ในรูปของคำสั่งภาษาเครื่องเท่านั้น ถ้าผู้ใช้เขียนโปรแกรมโดยใช้ภาษาระดับสูง (High Level Language) ก่อนที่จะสั่งให้คอมพิวเตอร์ทำงานจะต้องมีการแปลงเป็นภาษาระดับต่ำ (Low Level Language) เสียก่อน

หน่วยคำนวณและตรรกะ (Arithmetic and Logic Unit) หรือเรียกสั้น ๆ ว่า เอแอลยู (ALU) ทำหน้าที่ประมวลผลการคำนวณทางคณิตศาสตร์ ตลอดจนการเปรียบเทียบทางตรรกะทั้งหมด

ขั้นตอนการทำงานของซีพียู มีกระบวนการอยู่ 3 กระบวนการใหญ่ ๆ ได้แก่ กระบวนการอ่านและตรวจสอบข้อมูล (Fetch data) เป็นส่วนแรกที่ทำกรอ่านข้อมูลจากหน่วยความจำภายใน (Cache memory) หรือหน่วยความจำภายนอก (Main memory) หลังจากอ่านข้อมูลเข้ามาแล้วจะส่งไปให้ส่วนแปลความหมาย

หรือเข้ารหัสคำสั่ง (Decode data) เพื่อแปลให้เป็นภาษาเครื่อง (Binary code) หลังจากแปลคำสั่งเสร็จก็จะส่งไปยังส่วนควบคุมการทำงาน เพื่อนำไปปฏิบัติตามคำสั่งที่ได้รับมอบหมาย (Execute instruction) ต่อไป

การประมวลผลของซีพียูนับขึ้นกับปัจจัยหลายประการ ได้แก่ ขนาดของรีจิสเตอร์ ขนาดของแรม สัญญาณนาฬิกา ระบบ บัส ความกว้างขวางของดาต้าบัส โครงสร้างบัสของพีซี ขนาดของแอดเดรสบัส แคชเมมโมรี และ ซีพียูใช้แมธโครโพรเซสเซอร์ (Math Co-Processor) หรือการรวมส่วนคำนวณแบบโพลติง – พอยต์ (Floating point) ในการคำนวณ

โพรเซสเซอร์มีการออกแบบสถาปัตยกรรมแยกออกเป็นสองรูปแบบคือ สถาปัตยกรรมแบบ CISC (Complex Instruction Set Computer) และ สถาปัตยกรรมแบบ RISC (Reduced Instruction Set Computer) สถาปัตยกรรมคอมพิวเตอร์แบบ CISC นั้นจะมีจุดเด่นในเรื่องของการเขียนโปรแกรมเพราะผู้ใช้สามารถเขียนโปรแกรมได้ง่ายกว่า เนื่องจากโปรแกรมสามารถใช้คำสั่งที่มีอยู่มากมายในสถาปัตยกรรมคอมพิวเตอร์แบบ CISC นอกจากนี้ ขนาดของโปรแกรมนบน CISC ก็มีขนาดเล็กกว่า RISC



## คำถามทบทวนประจำบท

~~~~~

คำสั่ง จงตอบคำถามต่อไปนี้ โดยอธิบายให้เข้าใจ

1. ซีพียู จะทำงานอะไรบ้าง ขณะที่ประมวลผลคำสั่ง
2. จงยกตัวอย่างปัจจัยที่มีผลต่อความเร็วในการประมวลผลของซีพียูมา 3 ข้อ พร้อมอธิบายสาเหตุของปัจจัยเหล่านั้น
3. โครงสร้างภายในซีพียูมีส่วนประกอบที่สำคัญอะไรบ้าง จงอธิบายความสัมพันธ์ ของแต่ละส่วน
4. จงอธิบายลักษณะของสถาปัตยกรรมคอมพิวเตอร์แบบ CISC (Complex Instruction Set Computer) และ สถาปัตยกรรมแบบ RISC (Reduced Instruction Set Computer)
5. รีจิสเตอร์ตัวชี้ (Pointer register) มีกี่ชนิด อะไรบ้าง จงอธิบาย
6. จงอธิบายการไหลเวียนข้อมูลภายในวงรอบของการดึงข้อมูล (fetch data)

