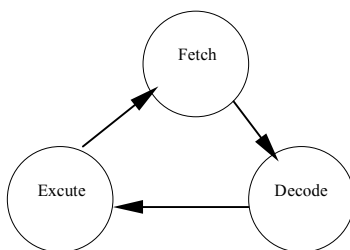


## บทที่ 4 : สถาปัตยกรรมคอมพิวเตอร์เบื้องต้น

ระบบคอมพิวเตอร์ประกอบด้วยหน่วยประมวลผลกลาง (CPU) หน่วยความจำ (Memory) อุปกรณ์รับข้อมูล (Input Units) และอุปกรณ์แสดงผล (Output Unit) ในบทนี้เราจะเริ่มพิจารณาการเชื่อมโยงของอุปกรณ์ต่าง ๆ และโครงสร้างภายในขององค์ประกอบเหล่านั้น

### หน่วยประมวลผลกลาง

หน่วยประมวลผลกลางมีหน้าที่ประมวลผลข้อมูลต่าง ๆ ในระบบคอมพิวเตอร์ โดยหน่วยประมวลผลกลางจะทำงานตามโปรแกรมที่ระบุโดยผู้ใช้ ขั้นตอนการทำงานของหน่วยประมวลผลกลางมีลักษณะเป็นวงรอบ โดยขั้นแรกหน่วยประมวลผลกลางจะอ่านคำสั่งจากหน่วยความจำ (fetch) จากนั้นหน่วยประมวลผลกลางจะตีความคำสั่งนั้น (decode) และในขั้นสุดท้ายหน่วยประมวลผลกลางก็จะประมวลผลตามคำสั่งที่อ่านเข้ามา (execute) เมื่อทำงานเสร็จหน่วยประมวลผลก็จะเริ่มอ่านคำสั่งเข้ามาอีกครั้ง ขั้นตอนดังกล่าวมีลักษณะดังรูปที่ 4.1



รูป 4.1 ขั้นตอนการทำงานของหน่วยประมวลผล

หน่วยประมวลผลกลางจะทำงานตามชุดคำสั่ง (instructions) ที่อ่านเข้ามาจากหน่วยความจำหลักเท่านั้น เราจะเรียกสถาปัตยกรรมของระบบคอมพิวเตอร์ที่มีการเก็บโปรแกรมและข้อมูลไว้ในหน่วยความจำหลัก โดยที่หน่วยประมวลผลจะทำงานกับหน่วยความจำเท่านั้น ว่า *Stored Program Architecture* หรือคอมพิวเตอร์แบบวอนนอยแมน (von Neumann Computer) โดยตั้งเป็นเกียรติให้กับ John von Neumann<sup>1</sup>

ชุดคำสั่งของคอมพิวเตอร์ โดยทั่วไปจะประกอบด้วยส่วนย่อย ๆ 2 ส่วนคือ ส่วน Opcode ซึ่งเป็นส่วนที่ระบุประเภทของการประมวลผล และส่วน Operand ซึ่งเป็นส่วนที่ระบุข้อมูลสำหรับการประมวลผลตามที่ระบุใน opcode

โดยปกติแล้ว เรานิยมใช้ไมโครโปรเซสเซอร์ทำหน้าที่เป็นหน่วยประมวลผลกลางในระบบคอมพิวเตอร์ ดังนั้นการที่เราอ้างถึงไมโครโปรเซสเซอร์เราจะอ้างถึงในหน้าที่ที่เป็นหน่วยประมวลผลกลาง โดยคำสองคำนี้อาจใช้แทนกันได้

### หน่วยความจำ

การเก็บข้อมูลในหน่วยความจำมีหน่วยที่เล็กที่สุดในการเก็บข้อมูลคือบิต แต่ในการเรียกข้อมูลจากหน่วยความจำนั้นจะกระทำในรูปของข้อมูลที่มีขนาดใหญ่กว่า คือจะมีขนาด 8 บิต หรือ 1 ไบท์ ภายในหน่วยความจำของระบบคอมพิวเตอร์หนึ่ง ๆ จะประกอบด้วยหน่วยย่อย ๆ ขนาด 8 บิตเหล่านี้อยู่มากมาย หน่วยย่อย ๆ เหล่านี้จะมีหมายเลขเฉพาะตัว เพื่อให้หน่วยประมวลผลสามารถใช้อ้างถึงเมื่อจะอ่าน หรือเขียนข้อมูลลงไปหน่วยย่อยหน่วยนั้น หมายเลขนี้เราเรียกว่า *แอดเดรส (Address)* ดังนั้นการ

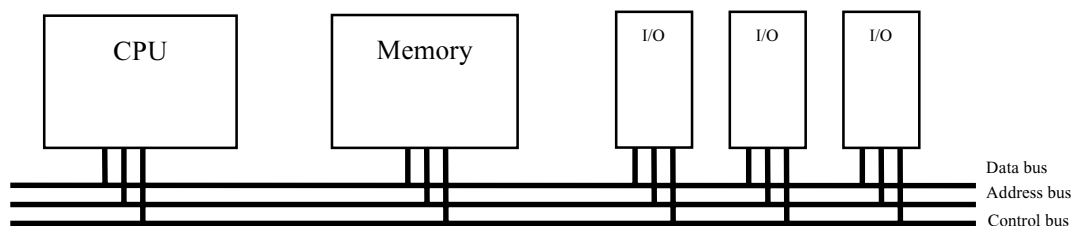
<sup>1</sup> ผู้บุกเบิกเทคโนโลยีคอมพิวเตอร์เชื่อว่าชื่อของคอมพิวเตอร์แบบวอนนอยแมน ให้เกียรติกับ von Neumann ผู้เขียนแนวคิดเกี่ยวกับคอมพิวเตอร์รูปแบบนี้มากไป โดยให้ความสำคัญกับ J.Presper Eckert และ John Mauchly วิศวกรผู้สร้างเครื่องคอมพิวเตอร์ลักษณะนี้ขึ้นมาอยู่เกินไป

ที่หน่วยประมวลผลจะ อ้างถึง ข้อมูลข้อมูลหนึ่งที่เก็บอยู่ในหน่วยความจำได้นั้น หน่วยประมวลผลจะต้องระบุ *แอดเดรส* ของข้อมูล ชื่อนั้นให้ได้ด้วย

## การเชื่อมต่อระหว่างอุปกรณ์ต่าง ๆ

### บัส (Bus) : ช่องทางสื่อสาร

อุปกรณ์ต่าง ๆ จะเชื่อมต่อกับโดยผ่านทางกลุ่มของสายสัญญาณ ที่เราเรียกว่า “บัส” อุปกรณ์ต่าง ๆ จะส่งและรับสัญญาณ ผ่านทางกลุ่มสายสัญญาณชุดเดียวกัน ดังรูป



รูปที่ 4.2 การเชื่อมต่อของอุปกรณ์ต่าง ๆ ผ่านระบบบัส

เราสามารถแบ่งกลุ่มของบัสออกเป็น 3 กลุ่ม คือ

- บัสข้อมูล (Data bus) ใช้สำหรับส่งรับข้อมูล
- บัสตำแหน่ง หรือ แอดเดรสบัส (Address bus) ใช้สำหรับส่งรับตำแหน่งสำหรับอ้างอิงข้อมูลจากหน่วยความจำ หรือ จากอุปกรณ์ I/O
- บัสควบคุม (Control bus) ใช้สำหรับส่งสัญญาณควบคุม

### แอดเดรสบัส (Address bus)

ในระบบคอมพิวเตอร์ที่หน่วยประมวลผลเชื่อมต่อกับอุปกรณ์อื่น ๆ ผ่านทางบัส ข้อมูลต่าง ๆ ที่ส่ง/รับกันระหว่างอุปกรณ์ต่าง ๆ นั้นจะส่งผ่านทางบัสข้อมูล ดังนั้นการที่หน่วยประมวลผลจะติดต่อกับหน่วยความจำหรืออุปกรณ์รับและแสดงผลข้อมูลที่ต้องการได้นั้น ทั้งหน่วยความจำ และ อุปกรณ์รับหรือแสดงผลข้อมูลทุกอุปกรณ์ จะต้องมีความหมายเฉพาะ หมายเลขนี้สำหรับหน่วยความจำก็คือแอดเดรส ส่วนอุปกรณ์อินพุตและอุปกรณ์เอาต์พุตก็มีความหมายเฉพาะสำหรับอุปกรณ์หนึ่ง ๆ เช่นเดียวกัน โดยเรียกว่า หมายเลข I/O (I/O address) เมื่อหน่วยประมวลผลต้องการติดต่อกับหน่วยความจำที่ตำแหน่งใด หรือติดต่อกับอุปกรณ์ใด ก็จะส่งแอดเดรสของหน่วยความจำนั้น หรือของอุปกรณ์นั้นมา ในการเลือกว่าหมายเลขแอดเดรสที่ส่งมาเป็นของหน่วยความจำ หรือของอุปกรณ์อินพุตเอาต์พุต หน่วยประมวลผลจะส่งสัญญาณระบุมาทางสัญญาณในบัสควบคุม

## สถาปัตยกรรมของระบบไมโครโปรเซสเซอร์ตระกูล 80x86

### ความเป็นมา

ไมโครโปรเซสเซอร์ตระกูล 80x86 เป็นไมโครโปรเซสเซอร์ที่พัฒนาขึ้นโดยบริษัท Intel โดยมีการพัฒนามาตั้งแต่รุ่น 4040 ซึ่งเป็นไมโครโปรเซสเซอร์ขนาด 4 บิต จนกระทั่งในปัจจุบันได้พัฒนาเป็นไมโครโปรเซสเซอร์รุ่น Pentium รายละเอียดคร่าว ๆ ของไมโครโปรเซสเซอร์รุ่นต่าง ๆ เป็นดังต่อไปนี้

- 8086 เป็นไมโครโปรเซสเซอร์ขนาด 16 บิต รุ่นแรกที่ Intel ผลิตขึ้น สามารถอ้างหน่วยความจำได้ 1 MB มีรีจิสเตอร์ภายในขนาด 16 บิต

- 8088 เนื่องจากในขณะนั้นอุปกรณ์ต่าง ๆ โดยมากเป็นอุปกรณ์ขนาด 8 บิต บริษัท Intel จึงได้ผลิตไมโครโปรเซสเซอร์ 8088 ซึ่งมีสถาปัตยกรรมภายในเหมือน 8086 แต่มีการติดต่อกับระบบภายนอกเป็นแบบ 8 บิต
- 80186 เป็นหน่วยประมวลผลซึ่งเพิ่มการจัดการเกี่ยวกับอุปกรณ์รอบข้างเข้าไป เพื่อให้เป็นหน่วยประมวลผลสำหรับงานควบคุมต่าง ๆ
- 80286 เป็นหน่วยประมวลผลขนาด 16 บิตที่มีความเร็วในการทำงานสูงขึ้น ขยายขอบเขตการอ้างหน่วยความจำเป็น 24 เมกะไบต์ และมีการเพิ่มความสามารถในการจัดการหน่วยความจำ
- 80386 เป็นหน่วยประมวลผลขนาด 32 บิต อ้างหน่วยความจำได้ถึง 4 กิกะไบต์ มีความสามารถในการจัดการหน่วยความจำที่ซับซ้อน และสามารถใช้อย่างมีประสิทธิภาพแบบเสมือนได้ มาตรฐานของชุดคำสั่งและกรรมวิธีในการจัดการหน่วยความจำของหน่วยประมวลผลรุ่นนี้ยังคงใช้เป็นมาตรฐานอยู่จนถึงปัจจุบันนี้ ยกตัวอย่างเช่น แม้แต่ระบบปฏิบัติการ Windows 95 ยังสามารถนำมาทำงานได้บนหน่วยประมวลผลรุ่นนี้ แต่จะทำงานได้ช้ามากเท่านั้นเอง
- 80386SX เป็นหน่วยประมวลผลซึ่งมีสถาปัตยกรรมภายในเหมือน 80386 แต่มีระบบบัสภายนอกเป็น 16 บิต หน่วยประมวลผลรุ่นนี้ได้รับการออกแบบขึ้นด้วยสาเหตุคล้ายกับ 8088
- 80486 ได้รับการพัฒนาจาก 80386 โดยการที่ทางบริษัท Intel เพิ่มหน่วยประมวลผลเลขทศนิยมเข้าไป
- 80486SX เป็นหน่วยประมวลผลรุ่น 80486 ซึ่งตัดความสามารถในส่วนของการประมวลผลเลขทศนิยมออก
- Pentium, Pentium Pro, Pentium II เป็นหน่วยประมวลผลรุ่นล่าสุดของบริษัท Intel มีการเพิ่มความสามารถในการประมวลผลให้สูงขึ้น โดยใช้เทคโนโลยีต่าง ๆ เช่นการประมวลผลแบบไปป์ไลน์ การประมวลผลแบบซูเปอร์สเกลาร์ เป็นต้น

นอกจากไมโครโปรเซสเซอร์ตระกูล 80x86 แล้ว ยังมีหน่วยประมวลผลกลางที่ผลิตและพัฒนาโดยบริษัทต่าง ๆ อีกหลายตระกูลเช่น ตระกูล 68000 และ PowerPC ของบริษัทโมโตโรล่า ตระกูล Alpha ของบริษัท DEC และตระกูล Sparc ของบริษัท Sun Microsystems เป็นต้น

ในการศึกษาบทนี้ เราจะศึกษาสถาปัตยกรรมของระบบไมโครโปรเซสเซอร์ 8086 เท่านั้น

## ลักษณะทั่วไปของไมโครโปรเซสเซอร์ 8086

### ระบบบัส

ไมโครโปรเซสเซอร์ 8086 มีแอดเดรสบัสขนาด 20 บิต ทำให้สามารถอ้างแอดเดรสได้ 1 เมกะไบต์ และมีบัสข้อมูลขนาด 16 บิต ซึ่งทำให้การอ่านและเขียนข้อมูลทำได้ครั้งละ 2 ไบต์ หน่วยประมวลผลทางคณิตศาสตร์และตรรกศาสตร์ภายใน 8086 สามารถประมวลผลได้กับข้อมูลขนาด 16 บิต รีจิสเตอร์ภายในไมโครโปรเซสเซอร์ 8086 มีขนาด 16 บิต

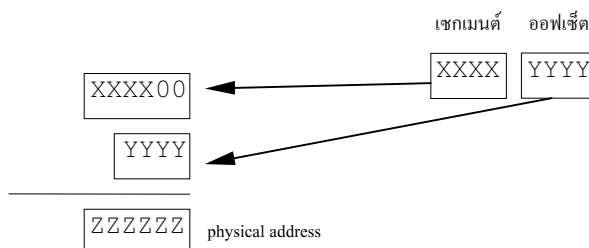
### การจัดการหน่วยความจำ

ภายในหน่วยประมวลผล 8086 มีรีจิสเตอร์ขนาด 16 บิต แต่มีแอดเดรสบัสขนาด 20 บิต ด้วยสาเหตุดังกล่าวหน่วยประมวลผลจะไม่สามารถเก็บตำแหน่งข้อมูลภายในหน่วยความจำได้ภายในรีจิสเตอร์เพียงตัวเดียว ดังนั้นการจัดเก็บตำแหน่งของข้อมูลภายในหน่วยความจำใน 8086 จึงต้องเก็บด้วยรีจิสเตอร์ 2 ตัว โดยมีวิธีการจัดเก็บแบบ **เซกเมนต์ : ออฟเซต (segment : offset)** แอดเดรสที่แท้จริง (physical address) ขนาด 20 บิต จะถูกจัดเก็บด้วยรีจิสเตอร์ขนาด 16 บิต 2 ตัว ค่าที่เก็บในรีจิสเตอร์ตัวแรกเรียกว่าเซกเมนต์ (segment) ส่วนค่าที่เก็บในรีจิสเตอร์อีกตัวเรียกว่าออฟเซต (offset)

การอ้างถึงตำแหน่งภายในหน่วยความจำแบบเซกเมนต์ : ออฟเซตนั้น อาจเปรียบได้เสมือนกับการที่เราแบ่งหน่วยความจำเป็นส่วนย่อย ๆ โดยส่วนย่อยนี้เราจะเรียกว่า เซกเมนต์ การที่เราจะอ้างถึงตำแหน่งใด ๆ เราจะอ้างถึงเซกเมนต์ที่ตำแหน่งข้อมูลนั้นอยู่ และระยะห่างของหน่วยความจำที่คิดเทียบกับจุดเริ่มต้นของเซกเมนต์ที่เราระบุไป

การแปลงแอดเดรสที่เก็บอยู่ในรูปของ เซกเมนต์ : ออฟเซต เป็นแอดเดรสขนาด 20 บิต มีขั้นตอนดังนี้

1. เลื่อนบิตของค่าเซกเมนต์ไปทางซ้าย 4 บิต ดังนั้นจากข้อมูล 16 บิต เราจะได้ข้อมูล 20 บิต ที่มี 4 บิตทางขวาเป็น 0 ในทุกหลัก
  2. นำค่าออฟเซตมาบวกเข้ากับค่าเซกเมนต์ที่เลื่อนบิตแล้ว จะได้แอดเดรสขนาด 20 บิต ที่จะนำไปอ้างตำแหน่งที่แท้จริงของข้อมูล
- ขั้นตอนทั้งสองแสดงได้ดังรูปที่ 4.2



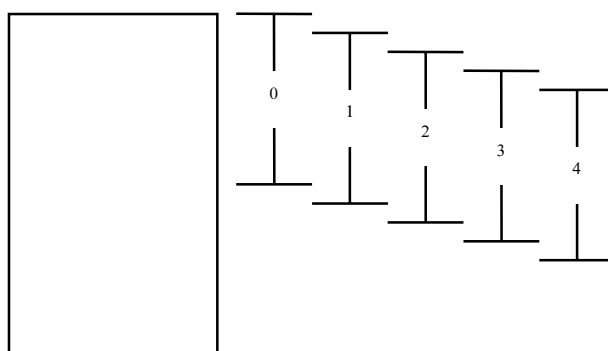
รูปที่ 4.3 การแปลงแอดเดรสจากแบบเซกเมนต์ : ออฟเซต เป็น แอดเดรสขนาด 20 บิต

ตัวอย่างเช่น 123Ah : 22B6h แปลงเป็นแอดเดรสขนาด 20 บิตได้ดังนี้

เลื่อนบิตของ 123Ah ไป 4 บิต ได้	123A0h
นำ 22B6h มาบวก	22B6h
จะได้แอดเดรสขนาด 20 บิตคือ	14656h

ในทางกลับกันที่ตำแหน่ง แอดเดรส 14656h ก็จะสามารถอ้างแบบ เซกเมนต์ : ออฟเซต ได้เป็น 123Ah : 22B6h เช่นเดียวกัน แต่ ตำแหน่งแอดเดรส 14656h สามารถอ้างแบบ เซกเมนต์ : ออฟเซต ค่าอื่นก็ได้เช่น 1465h : 0006h หรือ 1460h : 0056h และยังอ้างโดยใช้คู่เซกเมนต์ : ออฟเซต คู่อื่น ๆ ได้อีกหลายคู่

ผลจากการอ้างแอดเดรสแบบ เซกเมนต์ : ออฟเซต ทำให้ลักษณะของหน่วยความจำที่ 8086 มองเห็นจะมีลักษณะเป็นส่วน ๆ ที่อ้างอิงตามค่าของเซกเมนต์ โดยแต่ละส่วนนี้จะมีขนาดส่วนละ 64 กิโลไบต์ ส่วนของหน่วยความจำขนาด 64 กิโลไบต์นี้เรียกว่าเซกเมนต์ การจัดเรียงตัวของเซกเมนต์ต่าง ๆ ในหน่วยความจำจะจัดเรียงเป็นส่วน ๆ ที่มีการเหลื่อมกันดังรูป



รูปที่ 4.4 ลักษณะการเหลื่อมกันของเซกเมนต์

เนื่องจากการอ้างถึงข้อมูลใด ๆ ในหน่วยความจำของไมโครโปรเซสเซอร์ 8086 จะต้องอ้างตำแหน่งเป็นคู่เซกเมนต์ : ออฟเซต การอ้างถึงข้อมูลในตำแหน่งต่าง ๆ อาจทำได้ยุ่งยากเพราะต้องมีการระบุทั้งเซกเมนต์และออฟเซต ในไมโครโปรเซสเซอร์ 8086 จึงได้ออกแบบรีจิสเตอร์พิเศษขึ้น 4 ตัวเพื่อใช้เก็บค่าของเซกเมนต์ต่าง ๆ ที่กำลังใช้งานอยู่ในขณะนั้น กลุ่มของรีจิสเตอร์นั้น

เรียกว่า **เซกเมนต์รีจิสเตอร์** ซึ่งได้แก่ CS (Code segment) DS (Data segment) ES (Extra segment) และ SS (Stack segment) เซกเมนต์รีจิสเตอร์ทั้ง 4 ตัวนี้จะใช้ประกอบกับค่าออฟเซตต่าง ๆ เพื่อระบุตำแหน่งของ โปรแกรม (code) ข้อมูล (data) และแอสตีก (stack) ส่วนรีจิสเตอร์ ES มีหน้าที่เก็บเซกเมนต์ของข้อมูลที่ใช้ในการส่งงานคำสั่งพิเศษบางประเภท เช่น คำสั่งเกี่ยวกับข้อความ

ดังนั้นเราจะเห็นได้ว่า ถึงแม้ไมโครโปรเซสเซอร์ 8086 จะสามารถมองหน่วยความจำได้รวมถึง 1 เมกะไบต์ แต่โปรแกรมที่ทำงานอยู่จะมองเห็นหน่วยความจำได้พร้อม ๆ กันแค่เพียง 4 เซกเมนต์เท่านั้น นั่นคือ เซกเมนต์ของโปรแกรม เซกเมนต์ของข้อมูล 2 เซกเมนต์ และ เซกเมนต์ของแอสตีก

### แอสตีก (Stack)

ภายในหน่วยความจำของระบบไมโครโปรเซสเซอร์ 8086 จะมีหน่วยความจำส่วนหนึ่งที่ถูกกันเอาไว้สำหรับเป็น แอสตีก โดยเซกเมนต์ของแอสตีกจะถูกชี้โดยรีจิสเตอร์ SS ลักษณะเฉพาะของหน่วยความจำแบบแอสตีกคือการที่ระบบจะเก็บข้อมูลและอ่านข้อมูลออกไปแบบ เข้าก่อน ออกทีหลัง (First In Last Out : FILO) โดยอาจมองลักษณะเหมือนการวางซ้อนข้อมูลเหมือนซ้อนจานข้อมูลที่ถูกนำมาเก็บก่อนจะอยู่ทางด้านล่าง ข้อมูลถัดไปจะวางซ้อนอยู่ด้านบน ข้อมูลที่อยู่ทางด้านล่างจะไม่สามารถอ่านออกไปได้ ถ้ามีข้อมูลอื่นที่เก็บทีหลังและยังไม่ได้อ่านออกไป ระบบจะใช้แอสตีกในการเรียกโปรแกรมย่อย และเราจะศึกษาเกี่ยวกับแอสตีกโดยละเอียดอีกครั้งในส่วนของโปรแกรมย่อย

## รายละเอียดของส่วนประกอบภายในไมโครโปรเซสเซอร์

### หน่วยประมวลผลทางคณิตศาสตร์และตรรกศาสตร์ (ALU)

ไมโครโปรเซสเซอร์ 8086 มี ALU ที่สามารถประมวลผลได้ครั้งละ 16 บิต ด้วยความสามารถในการประมวลผลที่ละ 16 บิตนี้ ทำให้เราเรียกไมโครโปรเซสเซอร์ 8086 ว่าเป็นไมโครโปรเซสเซอร์ขนาด 16 บิต

### หน่วยความจำชั่วคราว (รีจิสเตอร์ --- Register)

รีจิสเตอร์ใน 8086 มีทั้งขนาด 16 บิต และ 8 บิต โดยจะแบ่งเป็นกลุ่ม ๆ ได้ดังนี้

#### 1.รีจิสเตอร์สำหรับใช้งานทั่วไป (General-Purpose Registers)

รีจิสเตอร์ในกลุ่มนี้ ผู้เขียนโปรแกรมสามารถนำไปใช้งานได้ตามความต้องการ โดยในกลุ่มนี้จะมี รีจิสเตอร์ขนาด 16 บิต อยู่ 4 ตัว โดยรีจิสเตอร์ขนาด 16 บิตทั้ง 4 ตัวจะแบ่งได้เป็นรีจิสเตอร์ขนาด 8 บิตอีก 8 ตัว รีจิสเตอร์ขนาด 16 บิต และคู่รีจิสเตอร์ขนาด 8 บิต มีดังต่อไปนี้

##### 1.1. รีจิสเตอร์ AX (Accumulator Register)

AX	
AH	AL

##### 1.2. รีจิสเตอร์ BX (Base Register)

BX	
BH	BL

##### 1.3. รีจิสเตอร์ CX (Counter Register)

CX	
CH	CL

#### 1.4. รีจิสเตอร์ DX (Data Register)

DX	
DH	DL

สำหรับรีจิสเตอร์ทั้ง 4 ตัวนี้ นอกจากจะมีไว้สำหรับใช้งานทั่วไปได้แล้ว ยังมีหน้าที่เฉพาะอื่น ๆ อีก ซึ่งจะกล่าวถึงในบทต่อไป

#### 2.รีจิสเตอร์สำหรับอ้างอิง (Index Register)

ไมโครโปรเซสเซอร์ 8086 มีรีจิสเตอร์สำหรับอ้างอิง 2 ตัว คือ SI (Source Index) และ DI (Destination Index) รีจิสเตอร์ในกลุ่มนี้ใช้สำหรับการอ้างตำแหน่งแบบอ้างอิง และใช้ในคำสั่งที่เกี่ยวกับข้อความ แต่ผู้ใช้สามารถนำไปใช้งานทั่วไปได้ด้วยเช่นกัน

#### 3.รีจิสเตอร์สำหรับการชี้ (Pointer Register)

รีจิสเตอร์กลุ่มนี้คือ SP และ BP รีจิสเตอร์ SP ใช้ประกอบกับรีจิสเตอร์ SS มีหน้าที่ชี้ตำแหน่งปัจจุบันของแอสตัก รีจิสเตอร์ BP ส่วนใหญ่จะใช้เพื่อชี้ตำแหน่งของแอสตักเช่นเดียวกัน แต่นิยมใช้ในส่วนของการส่งพารามิเตอร์ในโปรแกรมย่อย

#### 4.เซกเมนต์รีจิสเตอร์ (segment register)

เซกเมนต์รีจิสเตอร์ทั้ง 4 ตัวคือ CS DS ES และ SS ใช้ประกอบกับค่าของออฟเซต เพื่อชี้ตำแหน่งของโปรแกรม ข้อมูลปกติ ข้อมูลพิเศษ และ แอสตัก ตามลำดับ

#### 5.แฟล็ก (flag)

ไมโครโปรเซสเซอร์ 8086 จะเก็บลักษณะของผลลัพธ์ของการคำนวณทางคณิตศาสตร์ไว้ใน แฟล็ก

#### 6.รีจิสเตอร์อื่น ๆ ของระบบ

นอกเหนือจากรีจิสเตอร์ต่าง ๆ ที่ผู้ใช้สามารถกำหนดและใช้งานได้แล้ว ยังมีรีจิสเตอร์อีกกลุ่มหนึ่งซึ่งผู้เขียนโปรแกรมไม่สามารถเรียกใช้ได้ รีจิสเตอร์ในกลุ่มนี้ เช่น IP ซึ่งเป็นรีจิสเตอร์ที่ใช้ประกอบกับ CS เพื่อชี้ตำแหน่งของคำสั่งที่จะทำงานต่อไป หรือ IR ซึ่งเป็นรีจิสเตอร์ที่เก็บคำสั่งปัจจุบันที่ไมโครโปรเซสเซอร์อ่าน (fetch) ขึ้นมาจากหน่วยความจำ

### โหมดการอ้างอิงแอดเดรส

ไมโครโปรเซสเซอร์ 8086 สามารถอ้างอิงข้อมูลได้หลายแบบ โดยวิธีการต่าง ๆ ที่ 8086 อ้างอิงข้อมูลนั้น เรารวมเรียกว่า **โหมดการอ้างอิงแอดเดรส (Addressing mode)** ซึ่งรูปแบบที่ 8086 อ้างอิงข้อมูลนั้นแบ่งเป็นกลุ่ม ๆ ได้ 3 กลุ่มใหญ่ ๆ คือ กลุ่มที่อ้างอิงข้อมูลในรีจิสเตอร์ กลุ่มที่อ้างอิงข้อมูลที่ระบุในคำสั่ง และกลุ่มที่อ้างอิงข้อมูลในหน่วยความจำ เราจะศึกษาเรื่องของโหมดการอ้างอิงแอดเดรสอย่างละเอียดอีกครั้งในบทถัด ๆ ไป

### การอินเทอร์รัพท์ (Interrupt)

การอินเทอร์รัพท์ หรือการขัดจังหวะ คือการสั่งให้หน่วยประมวลผลหยุดการทำงานชั่วคราว แล้วกระโดดไปทำงานบางอย่างเพื่อตอบสนองการขัดจังหวะนั้น ตัวอย่างของการขัดจังหวะ เช่น อุปกรณ์บางชิ้นได้รับข้อมูล หรือ ข้อมูลได้รับเขียนเก็บลงในฮาร์ดดิสก์เรียบร้อยแล้ว เป็นต้น เมื่อหน่วยประมวลผลตอบสนองการขัดจังหวะเรียบร้อยแล้ว ก็จะคืนสู่สถานะเดิมและกลับไปประมวลผลงานเก่าที่ประมวลผลค้างไว้ เสมือนไม่มีอะไรเกิดขึ้น การขัดจังหวะนี้มีสองประเภทคือ ซอฟต์แวร์อินเทอร์รัพท์ และ ฮาร์ดแวร์อินเทอร์รัพท์ เรานิยมใช้ซอฟต์แวร์อินเทอร์รัพท์ในการเรียกใช้การบริการต่าง ๆ ของระบบ ส่วนฮาร์ดแวร์อินเทอร์รัพท์จะนิยมใช้ในการแจ้งการเปลี่ยนสถานะของอุปกรณ์อินพุตเอาต์พุต ต่าง ๆ

## สถาปัตยกรรมของระบบคอมพิวเตอร์สมัยใหม่

เทคโนโลยีของสถาปัตยกรรมคอมพิวเตอร์สมัยใหม่ได้พัฒนาไปอย่างรวดเร็วมาก เครื่องคอมพิวเตอร์ที่เราใช้ในปัจจุบันมีประสิทธิภาพมากกว่าเครื่องคอมพิวเตอร์ทั่วไปเมื่อ 2-3 ปีก่อนหลายเท่า ทั้งนี้เนื่องจากการวิจัยและสร้างหน่วยประมวลผลกลางและระบบคอมพิวเตอร์ที่มีประสิทธิภาพสูงขึ้นมาก

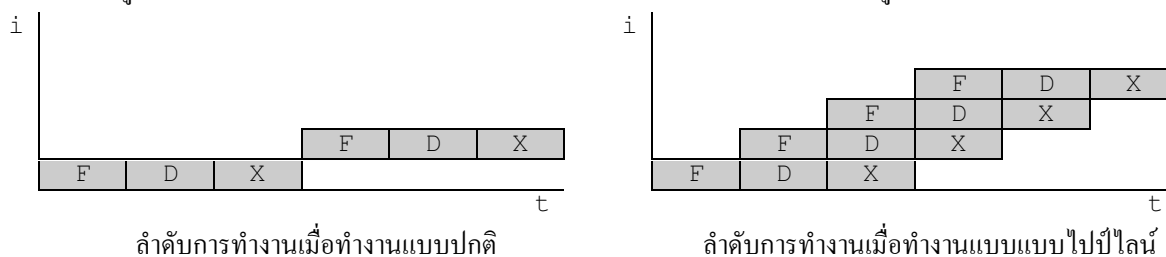
### เทคโนโลยีของหน่วยประมวลผลกลาง

- หน่วยประมวลผลแบบ RISC

ชุดคำสั่งของหน่วยประมวลผลยุคเก่ามีลักษณะเป็นแบบ CISC : Complex Instruction Set Computer นั่นคือชุดคำสั่งจะหนึ่ง ๆ จะมีความซับซ้อนมาก การที่ชุดคำสั่งซับซ้อนทำให้การออกแบบส่วนควบคุมภายในหน่วยประมวลผลทำได้ยาก ในปัจจุบันหน่วยประมวลผลต่าง ๆ ได้เปลี่ยนแนวทางในการพัฒนาไปเป็นแบบ RISC : Reduced Instruction Set Computer โดยเน้นชุดคำสั่งที่มีความซับซ้อนน้อยลง แต่มีความเร็วในการทำงานสูงขึ้น การทำให้ชุดคำสั่งมีรูปแบบที่ง่ายขึ้นทำให้การออกแบบส่วนควบคุมทำได้ง่ายขึ้น และยังทำให้สามารถใช้วิธีการแบบไปป์ไลน์ (Pipeline) และซูเปอร์สเกลาร์ (Superscalar) ในการเพิ่มประสิทธิภาพของหน่วยประมวลผลได้ง่ายขึ้นด้วย

- ไปป์ไลน์ (Pipeline)

หน่วยประมวลผลรุ่นใหม่จะมีการประมวลผลแบบไปป์ไลน์ กล่าวคือจะมีการ fetch decode และ execute คำสั่งเหลื่อมกันดังรูปที่ 4.5 การประมวลผลเหลื่อมกันนี้ทำให้ประสิทธิภาพของการประมวลผลสูงขึ้นมาก



รูปที่ 4.5 แสดงการทำงานแบบไปป์ไลน์เทียบกับการทำงานแบบปกติ

- ซูเปอร์สเกลาร์ (Superscalar)

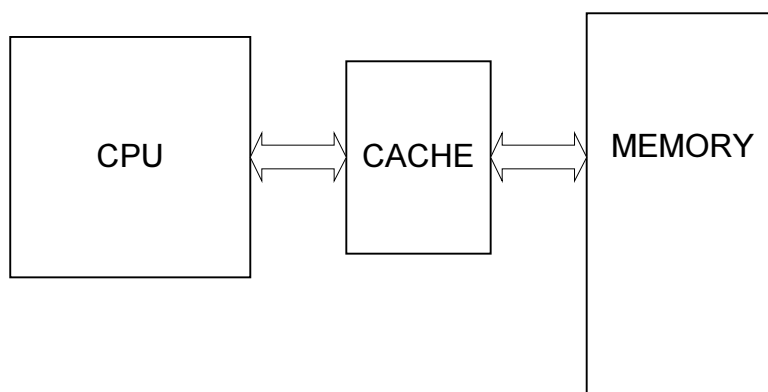
ในหน่วยประมวลผลที่มีประสิทธิภาพสูงบางรุ่น จะประมวลผลชุดคำสั่งหลายชุดคำสั่งได้พร้อมกัน การที่หน่วยประมวลผลประมวลผลคำสั่งได้หลายชุดพร้อมกันนี้เรียกว่า ซูเปอร์สเกลาร์

### ระบบบัสสมัยใหม่

ระบบคอมพิวเตอร์สมัยก่อน หน่วยประมวลผลมีความเร็วในการประมวลผลไม่มากนักทำให้การโอนย้ายข้อมูลระหว่างหน่วยประมวลผลกับหน่วยความจำกระทำไม่ได้โดยไม่ก่อให้เกิดการเสียเวลา แต่การพัฒนาของหน่วยประมวลผลเป็นไปอย่างรวดเร็วกว่าการพัฒนาของหน่วยความจำมากทำให้ปัจจุบันอัตราการประมวลผลของหน่วยประมวลผลสูงกว่าอัตราการโอนย้ายข้อมูลระหว่างหน่วยประมวลผลกับหน่วยความจำมาก สถานะการณ์เช่นนี้ก่อให้เกิดปัญหาในรูปแบบคอขวด (Bottleneck problem) ขึ้น นั่นคือจุดเชื่อมต่อระหว่างหน่วยความจำกับหน่วยประมวลผลทำให้ประสิทธิภาพของระบบลดลง เราเรียกคอขวดระหว่างหน่วยประมวลผลกับหน่วยความจำว่า คอขวดของวอนนอยแมน (von Neumann Bottleneck)

วิธีการแก้ปัญหานี้คือการใช้หน่วยความจำที่มีความเร็วสูงมากขนาดเล็กมาเป็นบัฟเฟอร์ (ที่พักข้อมูลชั่วคราว) ระหว่างหน่วยความจำและหน่วยประมวลผล หน่วยความจำที่มีความเร็วสูงนี้เรียกว่า หน่วยความจำแคช (cache memory) ในหน่วย

ประมวลผลปัจจุบันหลายรุ่น ได้มีการบรรจุหน่วยความจำแคชลงไปภายในไมโครโปรเซสเซอร์ด้วย ลักษณะของบัสที่มีการใช้หน่วยความจำแคชเป็นดังรูปที่ 4.6



รูปที่ 4.6 ลักษณะของบัสที่มีการใช้หน่วยความจำแคช