

บทที่ 3

โครงสร้างการเชื่อมต่อภายใน

- 3.1 ส่วนประกอบของคอมพิวเตอร์
- 3.2 หน้าที่ของคอมพิวเตอร์
- 3.3 โครงสร้างการเชื่อมต่อภายใน
- 3.4 การเชื่อมต่อแบบบัส
- 3.5 บัสแบบต่างๆ
- 3.6 เอกสารอ้างอิงและเว็บไซต์ที่ควรรู้

วัตถุประสงค์

- ทบทวนส่วนประกอบหลักของคอมพิวเตอร์
- หน้าที่ของส่วนประกอบหลักในระดับบนสุด
- การเชื่อมต่อของส่วนประกอบภายในคอมพิวเตอร์
- ระบบบัส

ค

อมพิวเตอร์ในมุมมองระดับบนสุดประกอบด้วยส่วนประกอบหลัก 3 ส่วน ได้แก่ หน่วยประมวลผลกลาง หน่วยความจำ หน่วยรับและแสดงผล ซึ่งส่วนประกอบแต่ละส่วนอาจมีเพียงหนึ่งหน่วยหรือมากกว่าหนึ่งก็ได้ โดยแต่ละส่วนสามารถติดต่อรับส่งข้อมูลระหว่างกันได้เพื่อให้คอมพิวเตอร์สามารถทำงานได้ตามหน้าที่พื้นฐาน นั่นก็คือ การประมวลผลโปรแกรม (execute program) ดังนั้นในมุมมองระดับบนสุด เราสามารถอธิบายระบบคอมพิวเตอร์ได้โดยการ

- อธิบายว่าแต่ละส่วนประกอบมีการแลกเปลี่ยนข้อมูลและสัญญาณควบคุมระหว่างกันอย่างไร
- อธิบายโครงสร้างการเชื่อมต่อภายใน รวมถึงการควบคุมสำหรับจัดการการใช้งานโครงสร้างนี้

โครงสร้างและหน้าที่ในระดับบนสุดเป็นสิ่งสำคัญ เนื่องจากการอธิบายให้เข้าใจถึงธรรมชาติของคอมพิวเตอร์ ยังรวมถึงความเข้าใจเกี่ยวกับการประเมินประสิทธิภาพของคอมพิวเตอร์ซึ่งเป็นเรื่องที่ค่อนข้างซับซ้อนด้วย อีกทั้งยังทำให้เห็นถึงปัญหาข้อขัดข้องของระบบ ทางเลือกอื่นๆ ปัญหาความล้มเหลวของระบบที่เกิดขึ้นเมื่อส่วนประกอบเสียหาย และการเพิ่มประสิทธิภาพให้กับระบบอย่างง่าย ในหลายๆ กรณี ปัญหาเหล่านี้แก้ไขได้ง่าย เพียงการเปลี่ยนแปลงการออกแบบแทนที่จะเพิ่มความเร็วหรือความน่าเชื่อถือให้กับส่วนประกอบแต่ละส่วน

ในบทนี้เราจะกล่าวถึงโครงสร้างพื้นฐานที่ใช้ในการเชื่อมต่อส่วนประกอบเข้าด้วยกัน โดยจะกล่าวถึงส่วนประกอบหลักและการจัดการสนับสนุนเพื่อเชื่อมต่อกับส่วนประกอบอื่นๆ (interface) จากนั้นจะกล่าวถึงหน้าที่หลักในภาพรวม แล้วแนะนำการใช้งานระบบบัสเพื่อเชื่อมต่อส่วนประกอบต่างๆ ของระบบเข้าด้วยกัน

3.1 ส่วนประกอบของคอมพิวเตอร์

ดังที่ได้กล่าวมาแล้วในบทที่ 2 ถึงโครงสร้างคอมพิวเตอร์ของ John von Neumann ที่ออกแบบไว้ตั้งแต่ปี ค.ศ.1945 (พ.ศ. 2488) ซึ่งมีการกำหนดหน้าที่การทำงานที่สำคัญของโครงสร้างไว้ และถือว่าเป็นโครงสร้างพื้นฐานในการพัฒนาคอมพิวเตอร์จากอดีตจนถึงปัจจุบัน ซึ่งการออกแบบนี้มักจะเรียกกันว่า สถาปัตยกรรม von Neumann และมีแนวคิดพื้นฐานที่สำคัญอยู่ 3 ประการ คือ

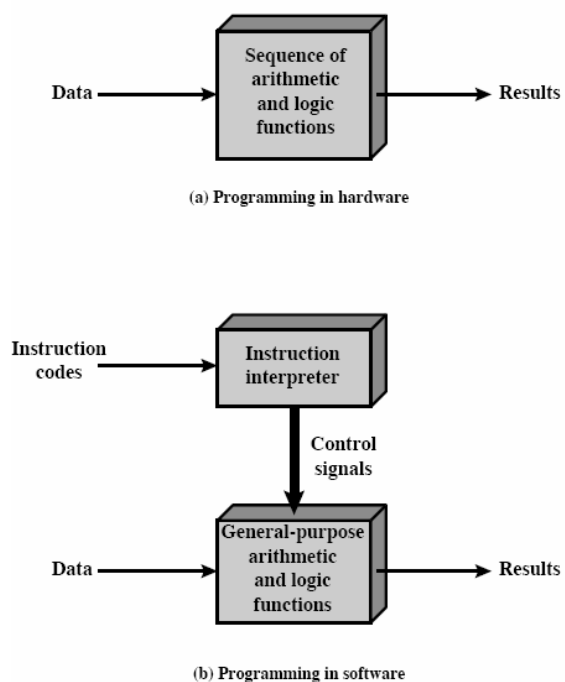
- ข้อมูลและคำสั่งจะต้องเก็บอยู่ในหน่วยความจำที่สามารถอ่านเขียน
- สิ่งที่เก็บอยู่ในหน่วยความจำนี้จะสามารถเข้าถึงได้โดยการระบุตำแหน่ง โดยไม่คำนึงถึงชนิดของข้อมูลที่บรรจุอยู่ภายใน
- การทำงานตามคำสั่งจะเป็นการทำแบบเรียงตามลำดับจากคำสั่งหนึ่งไปยังคำสั่งถัดไป (ยกเว้นบางกรณีที่มีการเปลี่ยนลำดับคำสั่ง)

เหตุผลของแนวคิดนี้ก็สืบเนื่องมาจากการข้อเสียของการออกแบบในเครื่องในยุคแรกๆ ที่สร้างขึ้นจากส่วนประกอบทางตรรกะขนาดเล็กหลายๆ ตัวนำมาเชื่อมต่อกันเพื่อให้สามารถเก็บข้อมูลและทำการคำนวณทางคณิตศาสตร์และตรรกะกับข้อมูลนั้นๆ ได้ เมื่อต้องการคำนวณแบบใดก็ตาม ต้องมีการออกแบบและกำหนดค่า (Configuration) ให้กับส่วนประกอบทางตรรกะแต่ละตัวเหล่านี้ เพื่อใช้สำหรับการคำนวณนั้นๆ ขึ้นมาโดยเฉพาะ

กระบวนการเชื่อมต่อส่วนประกอบเหล่านี้ตามที่แบบที่ได้กำหนดไว้ อาจมองได้ว่าเป็นรูปแบบหนึ่งของการเขียนโปรแกรม ซึ่งเรียกว่า *hardwired program* ผลที่ได้จากการเขียนโปรแกรมแบบนี้จะอยู่ในรูปของฮาร์ดแวร์

เรามามองอีกวิธีหนึ่งดูบ้าง ถ้าหากเราต้องการสร้าง configuration สำหรับใช้ในการคำนวณทางคณิตศาสตร์และตรรกะทั่วไป ชุดของฮาร์ดแวร์ที่จะทำหน้าที่เหล่านี้กับข้อมูลก็จะขึ้นอยู่กับสัญญาณควบคุมที่ส่งผ่านไปยังฮาร์ดแวร์ ในกรณีที่ที่ใช้การปรับเปลี่ยนฮาร์ดแวร์ ระบบจะรับข้อมูลเข้าและส่งผลลัพธ์ที่ได้ออกมาดังรูป 3.1a แต่ถ้าเป็นฮาร์ดแวร์สำหรับงานคำนวณทั่วไป ระบบจะรับข้อมูลและสัญญาณควบคุมเข้ามาและส่งผลลัพธ์ที่ได้ออกไป ดังนั้น แทนที่เราจะต้องไปปรับแก้สายสัญญาณของฮาร์ดแวร์ใหม่ทุกครั้งที่มีการเขียนโปรแกรมใหม่ โปรแกรมเมอร์ก็เพียงแค่ให้สัญญาณควบคุมชุดใหม่เท่านั้น

การให้สัญญาณควบคุมนี้ทำได้อย่างไร โปรแกรมทั้งหมดที่จริงแล้วก็เป็นขั้นตอนที่เรียงลำดับ ในแต่ละขั้นตอน อาจจะเป็นการทำงานทางคณิตศาสตร์หรือตรรกะกับข้อมูลบางส่วน และจะต้องมีชุดสัญญาณควบคุมสำหรับแต่ละขั้นตอนโดยเฉพาะโดยกำหนดเป็นรหัสเฉพาะ (Unique code) ไว้ แล้วเพิ่มอุปกรณ์เพื่อใช้ในการรับรหัสนี้แล้วสร้างเป็นชุดสัญญาณควบคุม ซึ่งก็คือ ตัวแปลคำสั่ง (Instruction interpreter) ดังแสดงในรูป 3.1b เพียงเท่านี้ก็ทำให้การเขียนโปรแกรมเป็นเรื่องง่าย เมื่อเราต้องการเขียนโปรแกรมใหม่ก็ทำได้โดยการระบุลำดับของรหัสที่จะใช้ขึ้นใหม่ รหัสที่ว่านี้ก็คือคำสั่ง (instruction) นั่นเอง แล้วส่งต่อไปให้ตัวแปลคำสั่งทำหน้าที่แปลคำสั่งเพื่อสร้างเป็นชุดสัญญาณควบคุมออกมา ลำดับของคำสั่งที่ได้จากการเขียนโปรแกรมวิธีนี้ เรียกว่า *ซอฟต์แวร์* (software) นั่นเอง



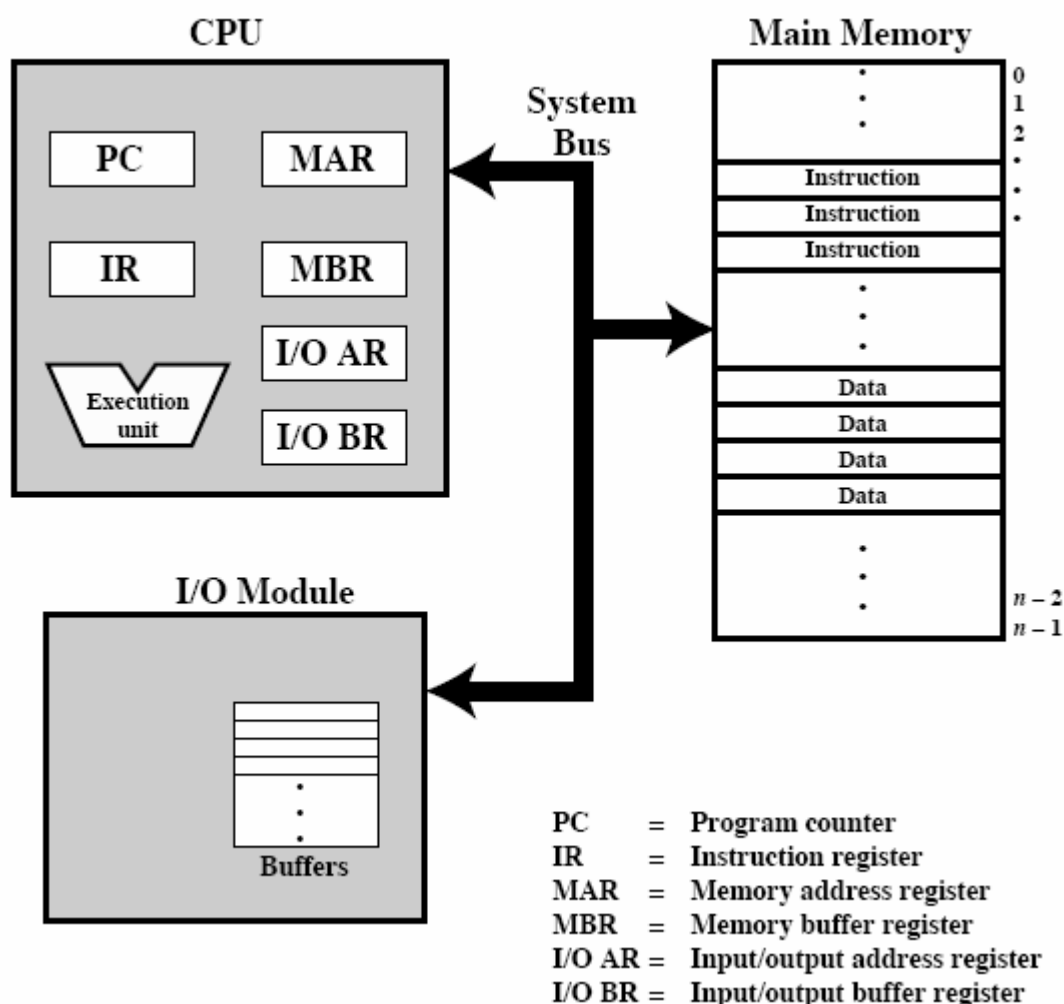
รูป 3.1 วิธีการโปรแกรมแบบใช้ฮาร์ดแวร์และซอฟต์แวร์

รูปที่ 3.1b แสดงส่วนประกอบหลัก 2 ส่วนในระบบ คือตัวแปลคำสั่ง และโมดูลสำหรับการคำนวณทางคณิตศาสตร์และตรรกะทั่วไป ทั้งสองส่วนนี้เป็นส่วนประกอบของซีพียู นอกจากนี้ยังมีส่วนประกอบอื่นๆ ที่ต้องใช้ในเพื่อสร้างคอมพิวเตอร์ให้ทำหน้าที่ได้สมบูรณ์อีก เช่น การรับข้อมูลและคำสั่งเข้าสู่ระบบจะกระทำได้อีกต้องมีโมดูลที่ใช้ในการรับ (input module) ซึ่งประกอบด้วยส่วนประกอบพื้นฐานที่ใช้ในการรับข้อมูลและคำสั่งในรูปแบบหนึ่ง แล้วแปลงให้อยู่ในรูปแบบของสัญญาณที่ใช้ภายในระบบ หรือการแสดงผลที่ได้ก็จำเป็นต้องมีโมดูลที่ใช้ในการแสดงผล (output module) เราเรียกทั้งสองโมดูลนี้รวมกันว่า หน่วยรับและแสดงผล (I/O component) ยังเหลืออีกหนึ่งส่วนประกอบที่

จำเป็น อุปกรณ์รับเข้าจะทำหน้าที่รับข้อมูลและคำสั่งเข้ามาตามลำดับ แต่โปรแกรมไม่จำเป็นจะต้องทำงานตามลำดับนั้น อาจจะมีคำสั่งให้ทำงานกระโดดข้ามไปข้ามมาได้ (ดังเช่นคำสั่ง Jump ในเครื่อง IAS) นอกจากนี้การทำงานกับข้อมูลอาจจะมีการเข้าถึงข้อมูลมากกว่าหนึ่งตัวในเวลาหนึ่งๆ ตามลำดับที่กำหนดไว้ก่อนแล้ว

ดังนั้น จึงจำเป็นต้องมีที่เก็บข้อมูลและคำสั่งไว้ชั่วคราว นั่นก็คือโมดูลที่เรียกว่า หน่วยความจำ (memory) หรืออาจเรียกว่า หน่วยความจำหลัก (main memory) เพื่อให้แตกต่างจากหน่วยบันทึกข้อมูลภายนอก (external storage) หรืออุปกรณ์รอบข้าง (peripheral device) ซึ่ง von Neumann กล่าวไว้ว่าหน่วยความจำแบบอย่างหลังนี้ก็สามารถใช้เก็บคำสั่งและข้อมูลได้เช่นกัน

รูปที่ 3.2 แสดงให้เห็นส่วนประกอบในระดับบนสุดและการติดต่อที่จำเป็นระหว่างกัน ซีพียูต้องมีการแลกเปลี่ยนข้อมูลกับหน่วยความจำ ซึ่งจำเป็นต้องใช้รีจิสเตอร์ภายในซีพียูสองตัวคือ memory address register (MAR) ที่ใช้ในการเก็บตำแหน่งหรือ address ในหน่วยความจำเพื่อทำการอ่านหรือเขียนต่อไป และอีกตัวคือ memory buffer register (MBR) ที่ใช้ในการเก็บข้อมูลที่จะเขียนลงในหน่วยความจำ หรือรับข้อมูลอ่านได้จากหน่วยความจำ ในส่วนของรีจิสเตอร์ที่ใช้ติดต่อกับหน่วยรับและแสดงผลก็มีลักษณะคล้ายกันคือมี 2 รีจิสเตอร์ในซีพียู ได้แก่ I/O address register (I/O AR) ที่เป็นตัวระบุอุปกรณ์รับและแสดงผลที่ติดต่อด้วย และ I/O buffer register (I/O BR) ที่ใช้ในการแลกเปลี่ยนข้อมูลระหว่างหน่วยรับและแสดงผลกับซีพียู



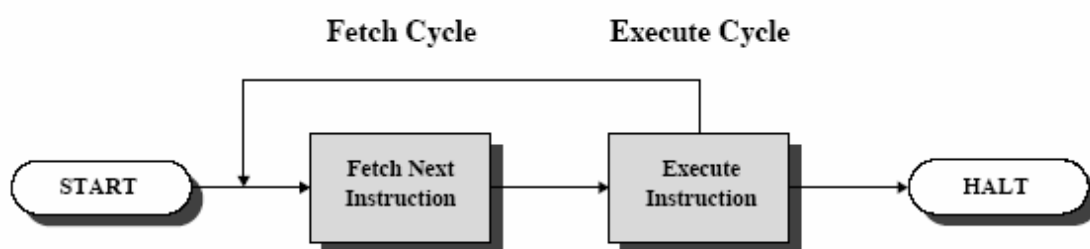
รูป 3.2 ส่วนประกอบของคอมพิวเตอร์ในระดับบนสุด

หน่วยความจำจะประกอบด้วยชุดของตำแหน่งที่ระบุโดยตัวเลขตำแหน่งเรียงตามลำดับ (sequentially numbered addresses) แต่ละตำแหน่งเก็บเลขฐานสองที่สามารถแปลงเป็นคำสั่งหรือข้อมูลก็ได้ หน่วยรับและแสดงผลจะรับข้อมูลจากอุปกรณ์ภายนอกเข้าสู่ซีพียูและหน่วยความจำ หรือกลับกัน คือ ส่งข้อมูลออกจากซีพียูและหน่วยความจำออกไปยังอุปกรณ์ภายนอก และจะมีบัฟเฟอร์ (buffer) ภายในที่ใช้เก็บข้อมูลเหล่านี้ไว้ชั่วคราว จนกว่าจะสามารถส่งออกไปได้

เมื่อมองเข้าไปในส่วนประกอบหลักเหล่านี้ เราก็จะเห็นภาพรวมว่าอุปกรณ์เหล่านี้ทำหน้าที่ร่วมกันเพื่อทำงานตามคำสั่งได้อย่างไร

3.2 หน้าที่ของคอมพิวเตอร์

หน้าที่พื้นฐานของคอมพิวเตอร์ก็คือ การประมวลผลโปรแกรม ซีพียูจะเป็นตัวหลักในการทำหน้าที่นี้โดยจะประมวลผลทีละคำสั่งตามลำดับที่โปรแกรมระบุไว้ กระบวนการในการประมวลผลหนึ่งคำสั่ง เรียกว่า วงรอบคำสั่ง (instruction cycle) ซึ่งประกอบด้วยขั้นตอนย่อย 2 ขั้นตอน เรียกว่า วงรอบดึงคำสั่ง (fetch cycle) และ วงรอบประมวลผล (execute cycle) ดังแสดงในรูป 3.3

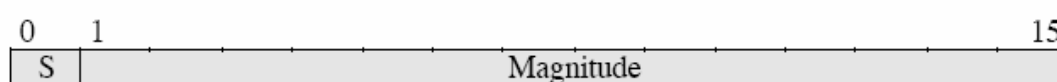


รูป 3.3 วงรอบคำสั่งพื้นฐาน

การทำงานในแต่ละวงรอบคำสั่ง จะเป็นการทำงานร่วมกันระหว่างโปรเซสเซอร์และหน่วยความจำ โดยโปรเซสเซอร์จะใช้รีจิสเตอร์ภายในตัวเองในการเก็บข้อมูลหรือคำสั่งชั่วคราว โครงสร้างของคำสั่งจะประกอบด้วยคำสั่ง (opcode) และตำแหน่งในหน่วยความจำ (address) ซึ่งจะใช้เนื้อที่ขนาดเท่ากับที่ใช้ในการเก็บข้อมูล ดังตัวอย่างในรูป 3.4



(a) Instruction format

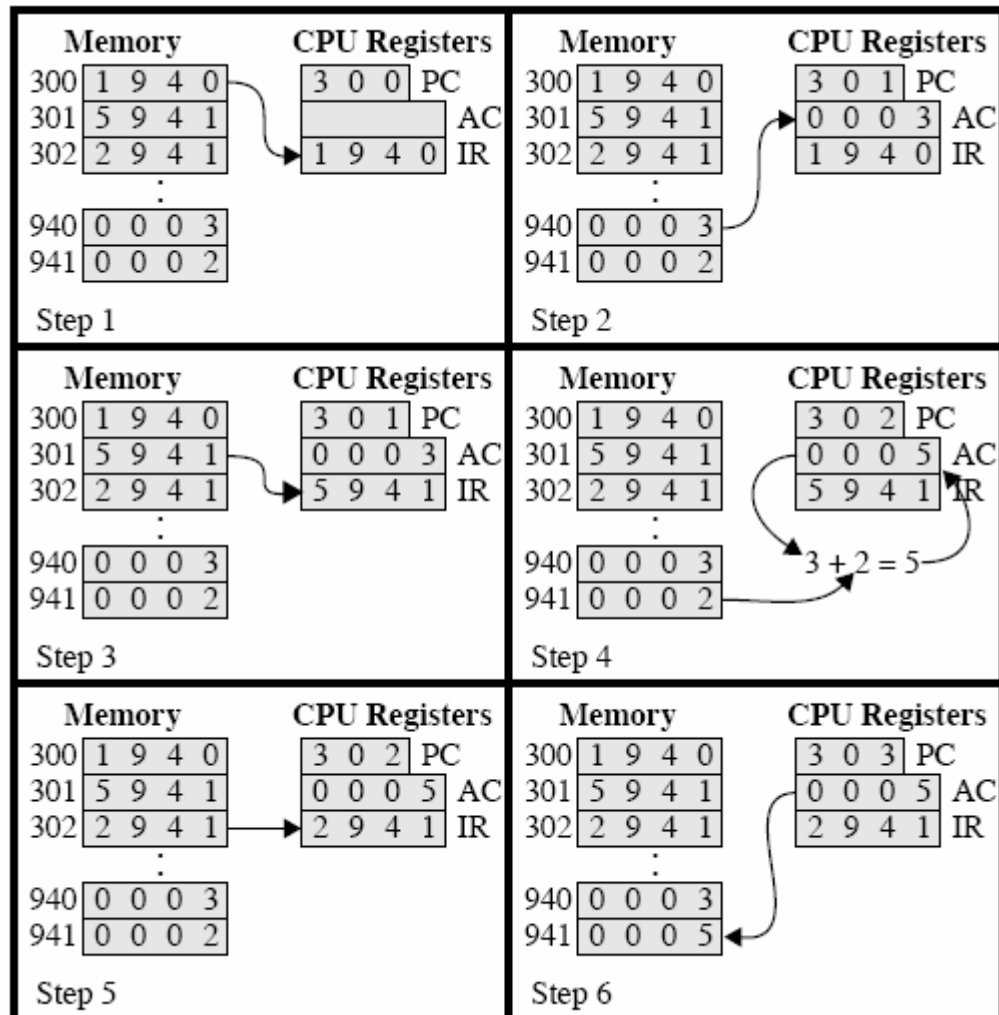


(b) Integer format

รูป 3.4 ตัวอย่างรูปแบบคำสั่งและข้อมูลที่เก็บอยู่ในรีจิสเตอร์

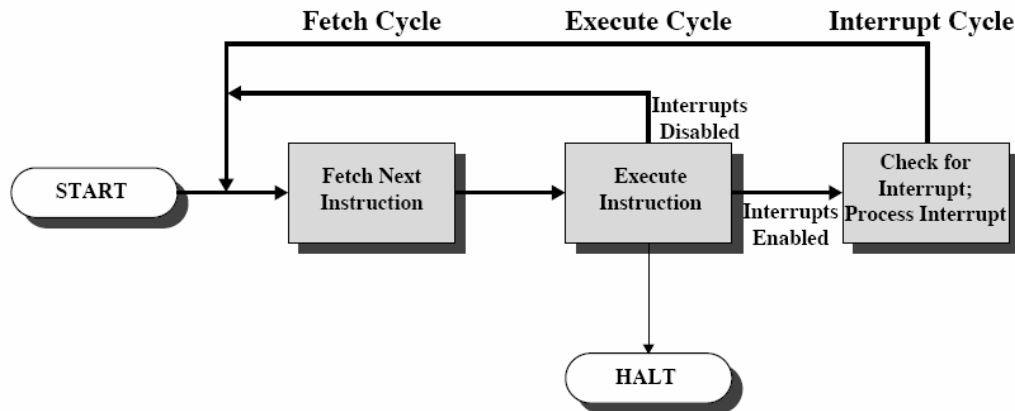
วงรอบคำสั่งเริ่มต้นเมื่อมีการดึงคำสั่งจากหน่วยความจำเข้ามาเก็บไว้ที่รีจิสเตอร์คำสั่ง (instruction register - IR) โดยตำแหน่งของคำสั่งลำดับถัดไปที่จะถูกดึงเข้ามา จะถูกเก็บไว้ที่รีจิสเตอร์โปรแกรมเคาน์เตอร์ (program counter - PC) ซึ่งค่าในรีจิสเตอร์นี้จะเพิ่มขึ้นเป็นตำแหน่งถัดไปหลังจากมีการดึงคำสั่งไปแล้ว รีจิสเตอร์อีกตัวหนึ่งที่ใช้เก็บข้อมูลชั่วคราวระหว่างการประมวลผล เรียกว่า รีจิสเตอร์สะสม (accumulator - AC)

รูป 3.5 แสดงตัวอย่างการประมวลผลโปรแกรม โดยประมวลผลตามวงรอบคำสั่งสามวงรอบ

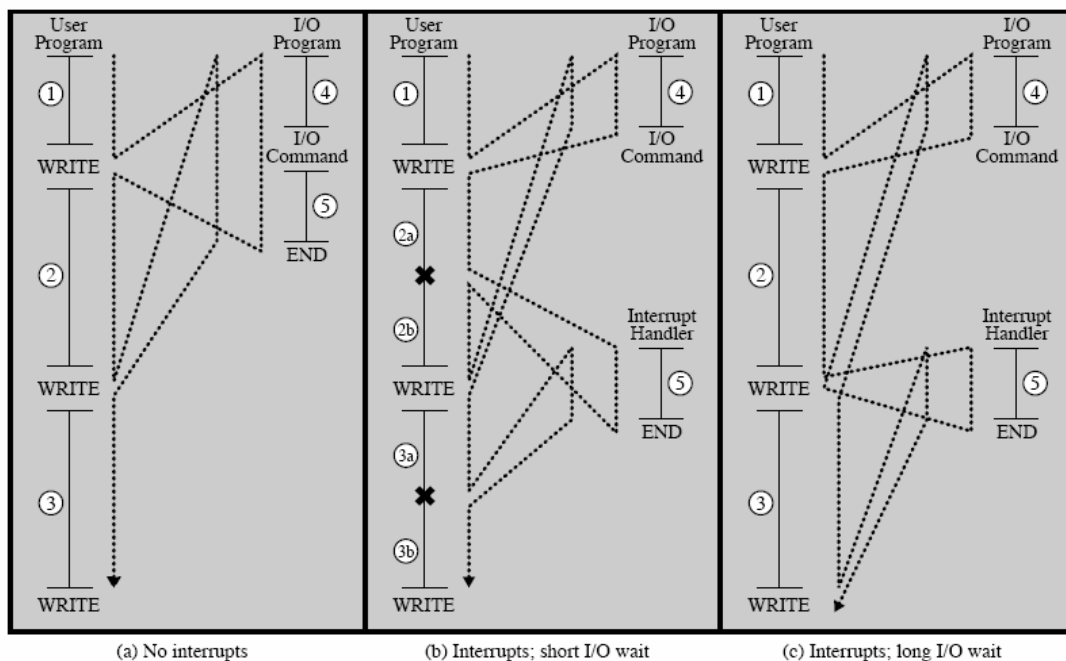


รูป 3.5 ตัวอย่างการประมวลผลโปรแกรม

การทำงานของโปรเซสเซอร์อาจถูกขัดจังหวะได้ เมื่อส่วนประกอบอื่นๆ หรืออุปกรณ์ภายนอกต้องการติดต่อกับโปรเซสเซอร์ ดังนั้นจึงต้องมีกลไกที่ช่วยให้โปรเซสเซอร์สามารถประมวลผลคำสั่งของโปรแกรมได้ โดยไม่เสียเวลารอคอยให้อุปกรณ์ที่ติดต่อมาทำงานจนเสร็จก่อน ก็คือ อินเทอร์รัพท์ (interrupt) ซึ่งจะช่วยให้การประมวลผลคำสั่งเร็วขึ้น และโปรเซสเซอร์ทำงานได้เต็มประสิทธิภาพ กระบวนการอินเทอร์รัพท์นี้จะเป็นการทำงานร่วมกันระหว่างโปรเซสเซอร์และระบบปฏิบัติการ โปรแกรมของผู้ใช้ทั่วไปไม่จำเป็นต้องมีคำสั่งในการจัดการอินเทอร์รัพท์ ดังนั้นในวงรอบคำสั่งจึงอาจมีวงรอบอินเทอร์รัพท์เพิ่มขึ้นได้ ดังแสดงในรูป 3.6 และรูป 3.7 แสดงการทำงานของโปรแกรมแบบมีและไม่มีอินเทอร์รัพท์



รูป 3.6 วงรอบคำสั่งแบบมีวงรอบอินเทอร์รัพท์

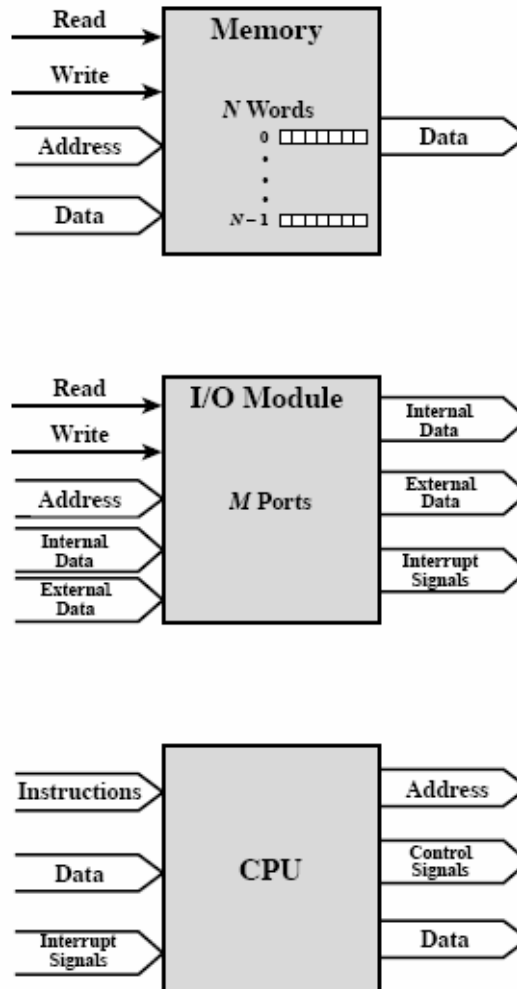


รูป 3.7 การทำงานของโปรแกรมแบบมีและไม่มีกรขัดจังหวะ

3.3 โครงสร้างการเชื่อมต่อภายใน

ส่วนประกอบพื้นฐานของคอมพิวเตอร์จำเป็นต้องมีการสื่อสารแลกเปลี่ยนข้อมูลระหว่างกัน ข้อมูลที่จำเป็นต้องมีการแลกเปลี่ยนระหว่างกันของส่วนประกอบทั้งสามแสดงไว้ดังรูป 3.8 ดังนั้นจึงจำเป็นต้องมีเส้นทางการเชื่อมส่วนประกอบเหล่านี้เข้าด้วยกัน เรียกว่า โครงสร้างการเชื่อมต่อภายใน (interconnection structure) โดยโครงสร้างนี้จะมีการสนับสนุนการถ่ายโอนข้อมูลดังนี้

- หน่วยความจำ -> โปรเซสเซอร์ เมื่อโปรเซสเซอร์อ่านคำสั่งหรือข้อมูลจากหน่วยความจำ
- โปรเซสเซอร์ -> หน่วยความจำ เมื่อโปรเซสเซอร์บันทึกผลลัพธ์จากการประมวลผล
- หน่วยรับและแสดงผล -> โปรเซสเซอร์ เมื่อโปรเซสเซอร์อ่านข้อมูลจากหน่วยรับข้อมูลเข้า
- โปรเซสเซอร์ -> หน่วยรับและแสดงผล เมื่อโปรเซสเซอร์ส่งข้อมูลไปยังหน่วยแสดงผล
- หน่วยรับและแสดงผล -> หน่วยความจำ เมื่อมีการแลกเปลี่ยนข้อมูลโดยตรงระหว่างสองหน่วยนี้

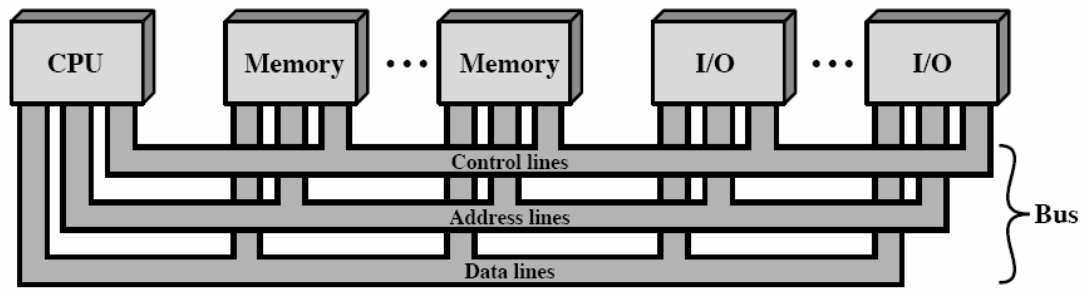


รูป 3.8 ส่วนประกอบพื้นฐานและข้อมูลที่มีการแลกเปลี่ยนระหว่างกัน

3.4 การเชื่อมต่อแบบบัส

เส้นทางในการเชื่อมโยงระหว่างส่วนประกอบหรืออุปกรณ์ต่างๆ ทำให้ติดต่อสื่อสารกันได้ เรียกว่า บัส (bus) ซึ่งถือเป็นการใช้สายสื่อสารร่วมกัน บัสจะประกอบด้วยสายสื่อสารหลายเส้น แต่ละเส้นสามารถใช้ส่งข้อมูลได้ 1 บิต (ส่งสัญญาณที่แทนค่า 0 หรือ 1) โดยทั่วไปจะสามารถส่งสัญญาณได้หลายบิตพร้อมกัน โดยความกว้างหรือขนาดของบัสก็คือจำนวนสายสัญญาณนั่นเอง โดยระบุเป็นจำนวนบิต เช่น บัสขนาด 8 บิตจะมีสายสัญญาณ 8 เส้น สามารถส่งข้อมูลได้ครั้งละ 8 บิต ความกว้างของบัสนี้เป็นองค์ประกอบหนึ่งที่ใช้อธิบายประสิทธิภาพของคอมพิวเตอร์ จำนวนบิตที่หน่วยประมวลผลสามารถจัดการได้ในแต่ละครั้งจะเรียกว่า ขนาดของคำ (word size) เพื่อให้การทำงานของหน่วยประมวลผลสอดคล้องกับเส้นทางในการรับส่งข้อมูล ขนาดของบัสและขนาดของคำควรมีขนาดเท่ากัน

ในทางกายภาพแล้ว บัสอาจเป็นตัวนำไฟฟ้าที่ฝังอยู่บนเมนบอร์ด สายไฟ สายเคเบิล ช่องที่ใช้เสียบแผงวงจรเข้ากับเมนบอร์ด เรียกว่า สล็อต (slot) สายสัญญาณที่ประกอบขึ้นเป็นบัสสามารถแบ่งตามหน้าที่ออกเป็น 3 ประเภท คือ บัสข้อมูล (data bus) บัสตำแหน่ง (address bus) และบัสควบคุม (control bus) ดังแสดงในรูป 3.9



รูป 3.9 การเชื่อมต่อแบบบัส

ความกว้างของบัสข้อมูลมักใช้เป็นปัจจัยหนึ่งที่บ่งบอกถึงประสิทธิภาพของเครื่องคอมพิวเตอร์ ส่วนความกว้างของบัสตำแหน่งจะเป็นตัวบ่งบอกถึงขนาดของหน่วยความจำสูงสุดที่สามารถเข้าถึงได้

นอกจากจะแบ่งบัสตามหน้าที่ของสายสัญญาณแล้ว บัสยังสามารถแบ่งเป็นหลายประเภทตามการใช้งานในการเชื่อมต่อกับอุปกรณ์ที่มีลักษณะเป็นโครงสร้างลำดับชั้น โดยทั่วไปจะแบ่งเป็น 2 ประเภท คือ

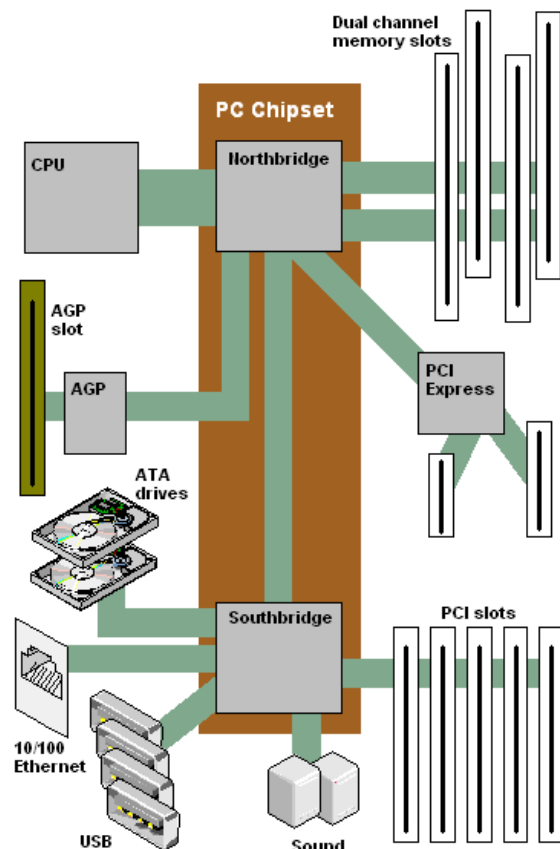
- บัสระบบ (system bus) ซึ่งเป็นส่วนหนึ่งของเมนบอร์ด เป็นเส้นทางที่เชื่อมต่อระหว่างอุปกรณ์หลักเข้าด้วยกัน
- บัสขยาย (expansion bus) เป็นบัสที่ทำให้หน่วยประมวลผลติดต่อกับอุปกรณ์อื่นๆ ได้ โดยอุปกรณ์เหล่านี้จะต่อเข้ากับสล롯 หรือพอร์ต (port)¹ ซึ่งเป็นจุดที่ใช้เชื่อมต่ออุปกรณ์เข้ากับบัสขยาย

From Computer Desktop Encyclopedia
© 2005 The Computer Language Co. Inc.

รูป 3.10 แสดงการเชื่อมต่อระหว่างหน่วยประมวลผล หน่วยความจำ อุปกรณ์รอบข้างต่างๆ ผ่านทางบัสระบบและบัสขยาย

การเชื่อมต่ออุปกรณ์ในลักษณะนี้ทำให้เกิดความยืดหยุ่นในการปรับเปลี่ยนอุปกรณ์และเพิ่มขีดความสามารถให้กับระบบคอมพิวเตอร์ในภายหลังได้

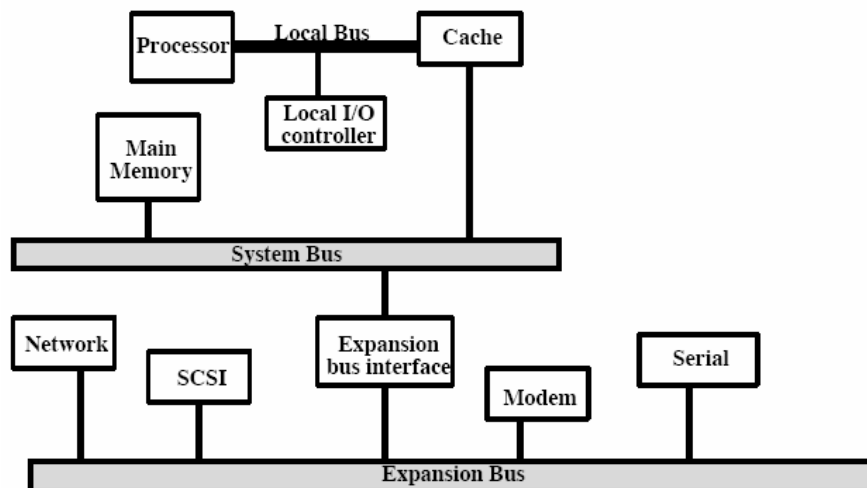
บัสขยายบนเมนบอร์ดมีหลายชนิด แต่ละชนิดบ่งบอกถึงชนิดของการ์ตที่ต่างๆ ที่ต่อกับคอมพิวเตอร์ เช่น บัส ISA บัส PCI บัส AGP บัส USB และบัส Firewire



รูป 3.10 บัสระบบและบัสขยาย

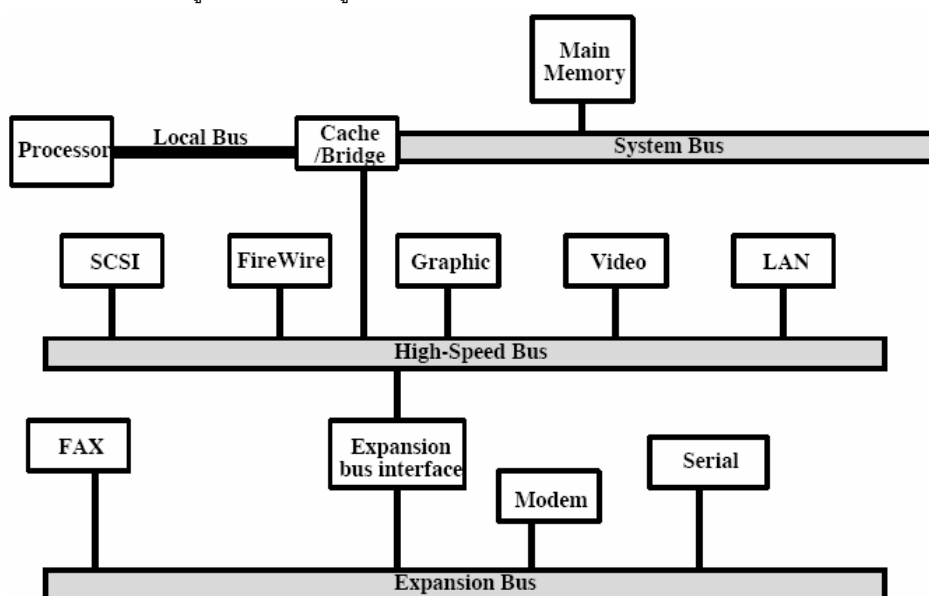
¹ พอร์ต อาจถือได้ว่าเป็นบัสรูปแบบหนึ่งที่ใช้ในการเชื่อมต่อระหว่างสองอุปกรณ์เข้าด้วยกัน ซึ่งแตกต่างจากคำว่าบัส ที่โดยทั่วไปใช้ในการเชื่อมต่ออุปกรณ์ตั้งแต่สองชนิดขึ้นไปและเป็นการใช้สายสัญญาณร่วมกัน (share)

บางระบบยังอาจแบ่งบัสออกเป็น 3 ระบบ โดยมีบัสท้องถิ่น (local bus) ที่เชื่อมต่อระหว่างโปรเซสเซอร์กับหน่วยความจำแคชเพิ่มเข้ามา ดังแสดงในรูป 3.11 โดยหน่วยความจำแคช (cache) จะเป็นตัวเชื่อมต่อเข้ากับบัสหลักอีกทีหนึ่ง การใช้หน่วยความจำแคชจะช่วยลดความถี่ในการติดต่อระหว่างโปรเซสเซอร์และหน่วยความจำได้ ซึ่งจะขอล่าวถึงรายละเอียดในบทถัดไป การเชื่อมต่อระหว่างบัสหลักกับบัสขยายจะทำโดยผ่านอินเทอร์เฟซของบัสขยาย (expansion bus interface) โดยมีบัฟเฟอร์ (buffer) ขนาดเล็กสำหรับเก็บข้อมูลชั่วคราวที่ส่งผ่านกัน จะเห็นว่าบัสขยายนี้นี้เป็นตัวให้บริการแก่อุปกรณ์อื่นๆ ในคอมพิวเตอร์ ทำให้ไม่ไปรบกวนการทำงานของบัสระบบ



รูป 3.11 สถาปัตยกรรมบัสแบบเก่า

เมื่ออุปกรณ์ต่างๆ ที่เชื่อมต่อเข้ากับบัสขยายมีการพัฒนาให้มีความเร็วในการรับส่งข้อมูลสูงขึ้น การเชื่อมต่อผ่านบัสขยายร่วมกับอุปกรณ์ที่มีความเร็วต่ำจะทำให้ประสิทธิภาพของอุปกรณ์ได้ไม่เต็มที่ จึงมีการออกแบบบัสความเร็วสูง (high speed bus) เพื่อมารองรับอุปกรณ์เหล่านี้ โดยใช้หน่วยความจำแคชเป็นตัวเชื่อมต่อในระบบเข้ากับบัสความเร็วสูง ดังแสดงในรูป 3.12



รูป 3.11 สถาปัตยกรรมบัสประสิทธิภาพสูง

ตัวอย่างของอุปกรณ์ที่มีความเร็วสูง เช่น อุปกรณ์สื่อสารระหว่างเครือข่ายความเร็วสูง (high-speed LAN เช่น 100 Mbps Fast Ethernet) หน่วยควบคุมการแสดงผลภาพ 3 มิติหรือภาพวิดีโอ (เช่น การ์ด AGP - Accelerated Graphics Port) หน่วยควบคุมการติดต่ออุปกรณ์รอบข้างที่มีความเร็วสูง (เช่น SCSI และ Firewire)

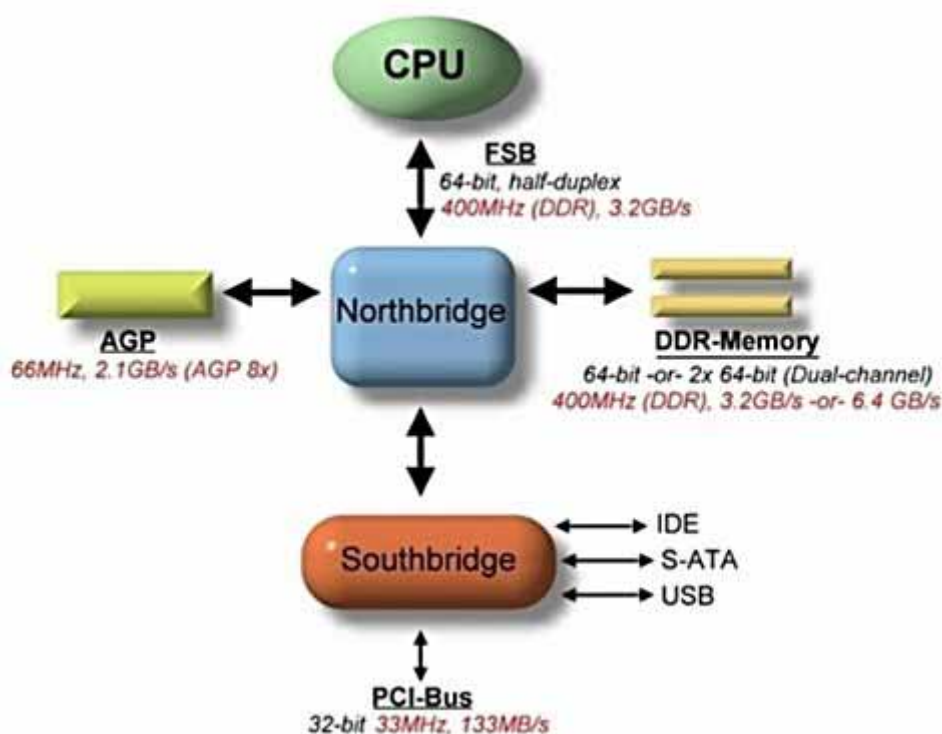
3.5 บัสแบบต่างๆ

บัสแต่ละชนิดก็เหมาะกับการเชื่อมต่ออุปกรณ์ที่แตกต่างกัน ดังจะกล่าวถึงต่อไปนี้

- ISA (Industry Standard Architecture bus) เป็นบัสที่มีการส่งข้อมูลช้าที่สุด ตัวอย่างอุปกรณ์ ได้แก่ เมาส์ โมเด็ม การ์ดเสียง และการ์ดเครือข่ายที่มีความเร็วต่ำ มีความกว้าง 8 และ 16 บิต ปัจจุบันเมนบอร์ดส่วนใหญ่ไม่มี ISA Slot สำหรับใช้งานแล้ว เนื่องจากมีบัส PCI เข้ามาแทนที่
- EISA (Extended ISA) เป็นบัสที่พัฒนามาจาก ISA ให้มีความกว้างเพิ่มขึ้นเป็น 32 บิต แต่ยังสามารถรองรับการ์ดแบบ ISA ได้ด้วย
- VL-BUS หรือ VESA Local bus (Video Electronics Standards Association local bus) เป็นบัสที่ออกแบบมาสำหรับการควบคุมการแสดงผลของจอภาพโดยเฉพาะ ส่วนใหญ่จึงจะใช้กับการวิดีโอเท่านั้น ใช้ในคอมพิวเตอร์ 80486 มีความกว้าง 32 บิต
- AGP (Accelerated Graphics Port) เป็นบัสที่ออกแบบโดยบริษัทอินเทล ในลักษณะเดียวกับ VL-BUS คือ เพื่อใช้งานกับการ์ดควบคุมการแสดงผลของคอมพิวเตอร์โดยเฉพาะ เพื่อเพิ่มประสิทธิภาพในการแสดงผลภาพ 3 มิติหรือภาพวิดีโอ มีความกว้าง 32 บิต
- USB (Universal Serial Bus) เป็นบัสที่ไม่ต้องการติดตั้งการ์ดลงบนเมนบอร์ด โดยอุปกรณ์จะถูกต่อเข้ากับพอร์ต USB ที่ต่อกับบัส USB ซึ่งเชื่อมต่อกับบัส PCI อีกที
- FireWire เป็นบัสความเร็วสูงที่ออกมาสำหรับอุปกรณ์รับและแสดงผลความเร็วสูงบางชนิด เช่น กล้องวิดีโอ กล้องดิจิทัล มีลักษณะและการทำงานคล้ายกับ USB
- PCI (Peripheral Component Interconnection) เป็นบัสที่บริษัทอินเทลออกแบบใหม่ เพื่อให้สนับสนุนการทำงานของเพนเทียมได้อย่างมีประสิทธิภาพ ออกแบบให้ใช้ได้กับอุปกรณ์ต่างๆ ที่หลากหลายกว่า VESA local bus มาก และมีความเร็วในการส่งข้อมูลเป็น 4 เท่าของบัส ISA มีความกว้าง 32 และ 64 บิต ปัจจุบันเป็นที่เห็นพ้องต้องกันว่า PCI เป็นบัสที่มีประสิทธิภาพสำหรับการประมวลผลของคอมพิวเตอร์จึงได้รับการพัฒนาออกมาเป็นมาตรฐานหลายรุ่นด้วยกันให้มีความสามารถที่เพิ่มขึ้นเรื่อยๆ
- PCI-X (PCI Express) เป็นมาตรฐานใหม่ที่จะมาแทน PCI และ AGP ในไม่ช้า ซึ่งมีความเร็วในการรับส่งข้อมูลให้เลือกใช้ได้ตามความต้องการของอุปกรณ์แต่ละชนิด มีความกว้าง 64 บิต อีกทั้งอัตราการรับส่งข้อมูลก็เพิ่มขึ้นเป็นหลายเท่าเมื่อเทียบกับ PCI และ AGP ปัจจุบันนิยมใช้ในเครื่องคอมพิวเตอร์ระดับ Server หรือ Workstation ที่ต้องการประสิทธิภาพในการรับส่งข้อมูลมากกว่าเครื่องพีซีทั่วไป

การทำงานของซีพียูจะมีการป้อนสัญญาณนาฬิกา เพื่อเป็นการให้จังหวะในการทำงานในแต่ละวงรอบ สัญญาณนาฬิกานี้จะมาจากวงจรบนเมนบอร์ด ซึ่งผู้ผลิตกำหนดให้สัญญาณนาฬิกามีความถี่เป็นเฮิร์ต (hertz หรือ Hz) เมื่อพูดถึงเมกะเฮิร์ต (MHz) จะหมายถึงสัญญาณนาฬิกา (ตึก) 1 ล้านครั้งใน 1 วินาที ความถี่ของซีพียูแต่ละรุ่นจะถูกกำหนดมาตั้งแต่ขั้นตอนการออกแบบ และถือเป็นความถี่หลักในการทำงานของทั้งเมนบอร์ด เรียกว่า ความถี่ Front Side Bus (FSB) ของซีพียู วงจรอื่นๆ ที่ติดต่อกับซีพียูไม่ว่าจะเป็นหน่วยความจำ หรือบัสด้านต่างๆ ก็ล้วนแล้วแต่ทำงานโดยอ้างอิงกับสัญญาณนาฬิกานี้ ดังแสดงตัวอย่างในรูป 3.12

ปัจจุบันความถี่ FSB อยู่ที่ช่วง 100-266 MHz แต่ซีพียูจะมีความถี่สูงสุดได้ถึงหลัก GHz เนื่องจากภายในซีพียูจะมีวงจรที่ทำหน้าที่คูณ FSB ที่ได้รับเข้ามาให้ทำงานได้เร็วขึ้น ดังเช่นตัวอย่างในรูป 3.12 เป็นตัวอย่างของซีพียู Athlon 64 รุ่นหนึ่ง มีสัญญาณนาฬิกาพื้นฐาน (FSB) ที่ 200 MHz (400 MHz DDR² ตามรูป) นำมาเพิ่มความถี่ด้วยวงจรคูณสัญญาณนาฬิกาซึ่งถูกตั้งค่าตัวคูณไว้ที่ x12 ก็จะได้ความเร็วเท่ากับ 2.4 GHz



รูป 3.12 โครงสร้างสถาปัตยกรรมที่แสดงความถี่ของสัญญาณนาฬิกาและความเร็วในการเชื่อมต่อ

ดังที่กล่าวไปแล้วว่าบัสดังกล่าวทำงานโดยอ้างอิงสัญญาณนาฬิกาเช่นเดียวกับหน่วยประมวลผล สัญญาณนาฬิกาของบัสดังกล่าวที่มีความถี่สูง ก็จะมีความเร็วในการส่งข้อมูลหรือแบนด์วิธ (Bandwidth) ของบัสดังกล่าวมากขึ้นตามไปด้วย บัสแต่ละชนิดก็จะมีค่าความถี่ของสัญญาณนาฬิกาและความเร็วในการรับส่งข้อมูลที่แตกต่างกันไป ดังแสดงตัวอย่างไว้ในตาราง 3.1

² การใช้เทคนิค Double Data Rate ทำให้สามารถรับส่งข้อมูลได้ 2 ครั้งในแต่ละลูกคลื่น ดังนั้นความถี่ที่แสดงจึงเป็นสองเท่าของความถี่ FSB ที่แท้จริง

บัส	ความกว้าง (bits)	ความถี่ (MHz)	อัตราการรับส่งข้อมูล (Mbytes/sec)
8-bit ISA	8	8.3	7.9
16-bit ISA	16	8.3	15.9
EISA	32	8.3	31.8
VL	32	33	127.2
PCI 2.3	32	66	254.3
PCI 2.3	64	66	508.6
AGP	32	66	254.3
AGP 2x	32	66x2	508.6
AGP 4x	32	66x4	1007.3
PCI-X	64	133	1014.7

ตาราง 2.1 ความถี่และความเร็วในการรับส่งข้อมูลของบัสแต่ละชนิด

แบนด์วิธของบัสที่แสดงในตารางนี้เป็นเพียงอัตราเร็วสูงสุดในการรับส่งในทางทฤษฎีเท่านั้น แต่ในทางปฏิบัติแล้วค่าที่ได้มักจะต่ำกว่านี้ขึ้นอยู่กับค่า overhead ของคำสั่งและปัจจัยอื่นๆ จากตารางนี้จะเห็นว่าค่าแบนด์วิธของบัสที่ได้อาจจะไม่ตรงตามที่บริษัทและหลายๆ คนมักจะเขียนระบุไว้ เช่น มาตรฐาน PCI-X จะอยู่ที่ $64/8 \text{ (bytes)} \times 133 = 1,064 \text{ MB/sec}$ ซึ่งเป็นการเขียนแบบประมาณค่า จึงไม่ใช่อัตราเร็วที่แท้จริง เนื่องจากการนิยามของ M ที่แตกต่างกัน ตัว M ใน MHz หมายถึง ค่า 1,000,000 (106) แต่ M ใน Mbytes/sec หมายถึง ค่า 1,048,576 (220) ดังนั้นหากจะเขียนให้ถูกต้องแล้วบัส PCI จะมีแบนด์วิธเป็น $64/8 \times 133 \times 1,000,000 / 1,048,576$ เท่ากับ 1014.7 MBytes/sec

3.6 เอกสารอ้างอิงและเว็บไซต์ที่ควรรู้

Chapter 3 William Stalling. Computer organization and architecture: designing for performance.

บทที่ 5 เมนบอร์ดและระบบบัส อนิรุทธิ์ รัชตะวราห์และคณะ. ฝากอมพิวเตอร์ 2005. สำนักพิมพ์ provision

<http://www.pcguides.com/ref/mbsys/buses/index.htm>

<http://www.ackadia.com/computer/system-architecture/system-architecture-motherboard.php>

http://www.pcmag.com/encyclopedia_term/0,2542,t=bus&i=39054,00.asp

[http://www3.ipst.ac.th/research/assets/web/mahidol/computer\(10\)/system/hardware.htm](http://www3.ipst.ac.th/research/assets/web/mahidol/computer(10)/system/hardware.htm)