

Chapter 2

หน้าที่ และ โครงสร้างการเชื่อมโยง
ภายในเครื่องคอมพิวเตอร์



หัวข้อการเรียนรู้

- ◎ ส่วนประกอบเครื่องคอมพิวเตอร์
- ◎ หน้าที่ของคอมพิวเตอร์
 - การดึงคำสั่ง และการประมวลผล
 - อินเทอร์เน็ต
 - ฟังก์ชันไอโอ
- ◎ โครงสร้างการเชื่อมโยงภายในเครื่องคอมพิวเตอร์
- ◎ การเชื่อมโยงโดยใช้บัส
 - โครงสร้างแบบบัส
 - โครงสร้างลำดับชั้นของบัสหลายระดับ
 - ประเด็นการพิจารณาในการออกแบบบัส



บทนำ

ที่ระดับบนสุดของโครงสร้างลำดับชั้น เครื่องคอมพิวเตอร์ประกอบด้วยซีพียู (Central Processing Unit: CPU) หน่วยความจำและอุปกรณ์ไอโอ หรืออุปกรณ์นำเข้าและรับข้อมูลอย่างละหนึ่งหน่วยเป็นอย่างต่ำ ส่วนประกอบเหล่านี้ถูกเชื่อมต่อเข้าด้วยกันด้วยวิธีการใดวิธีการหนึ่ง เพื่อให้สามารถทำหน้าที่พื้นฐานของเครื่องคอมพิวเตอร์ได้ นั่นคือการประมวลผลโปรแกรม ดังนั้นที่ระดับบนสุดนี้สามารถให้คำนิยามของคอมพิวเตอร์ได้โดย

1. อธิบายพฤติกรรมภายนอกของแต่ละส่วน นั่นคือการแลกเปลี่ยนข้อมูลและสัญญาณควบคุมกับอุปกรณ์ในส่วนอื่น
2. อธิบายโครงสร้างการเชื่อมต่อระหว่างอุปกรณ์และการควบคุมการจัดการโครงสร้างการเชื่อมต่อ



ส่วนประกอบเครื่องคอมพิวเตอร์

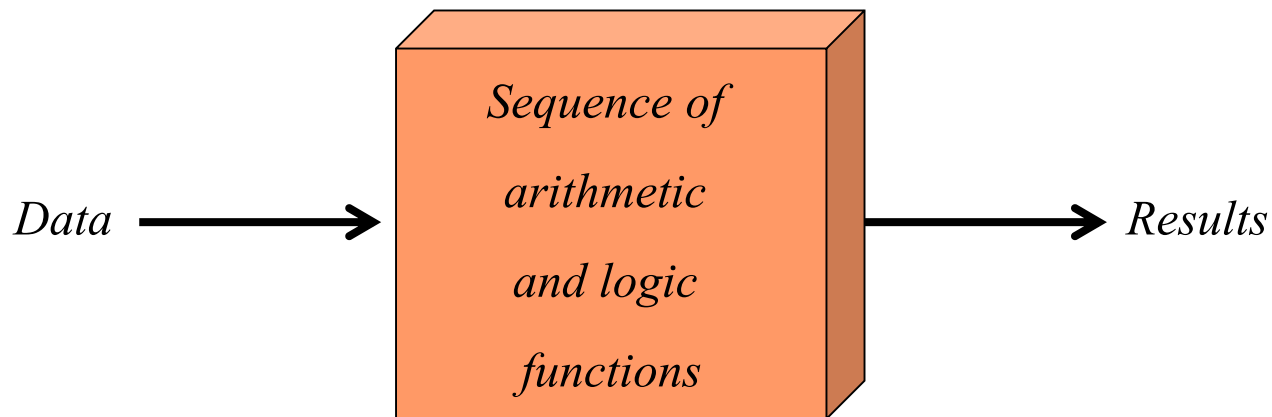
สถาปัตยกรรมแบบ *von Neumann* ประกอบด้วยนิยามพื้นฐานสามประการ คือ

- ◎ ข้อมูล และคำสั่งจะต้องถูกเก็บไว้ในหน่วยความจำที่สามารถเข้าถึงได้
- ◎ สิ่งที่เก็บอยู่ในหน่วยความจำนี้สามารถเข้าถึงได้ด้วยการใช้ตำแหน่งอ้างอิง โดยไม่ต้องสนใจว่าจะเป็นข้อมูลชนิดใด
- ◎ การประมวลผลเกิดขึ้นแบบตามลำดับจากคำสั่งหนึ่งไปยังอีกคำสั่งหนึ่งโดยอัตโนมัติ ยกเว้นมีการกำหนดลำดับให้เป็นไปอย่างอื่น



ส่วนประกอบเครื่องคอมพิวเตอร์

สำหรับฮาร์ดแวร์ที่สร้างขึ้นมาเพื่อใช้งานทั่วไป ระบบจะรับข้อมูลเข้ามาพร้อมกับ
สัญญาณควบคุมแล้วจึงสร้างผลลัพธ์

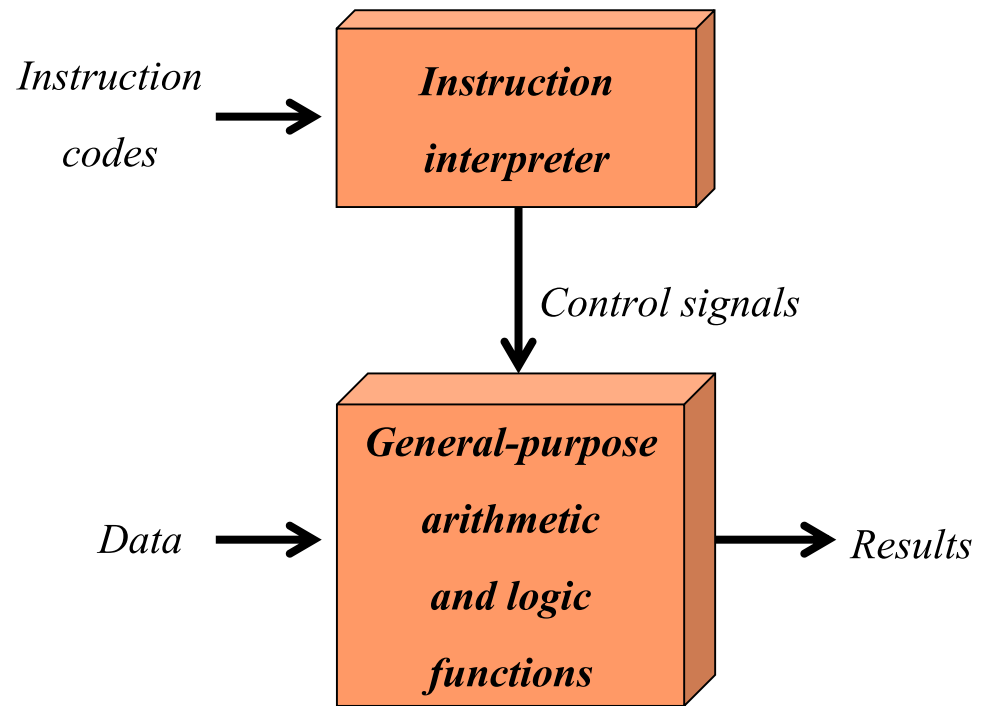


(a) Programming in hardware



ส่วนประกอบเครื่องคอมพิวเตอร์

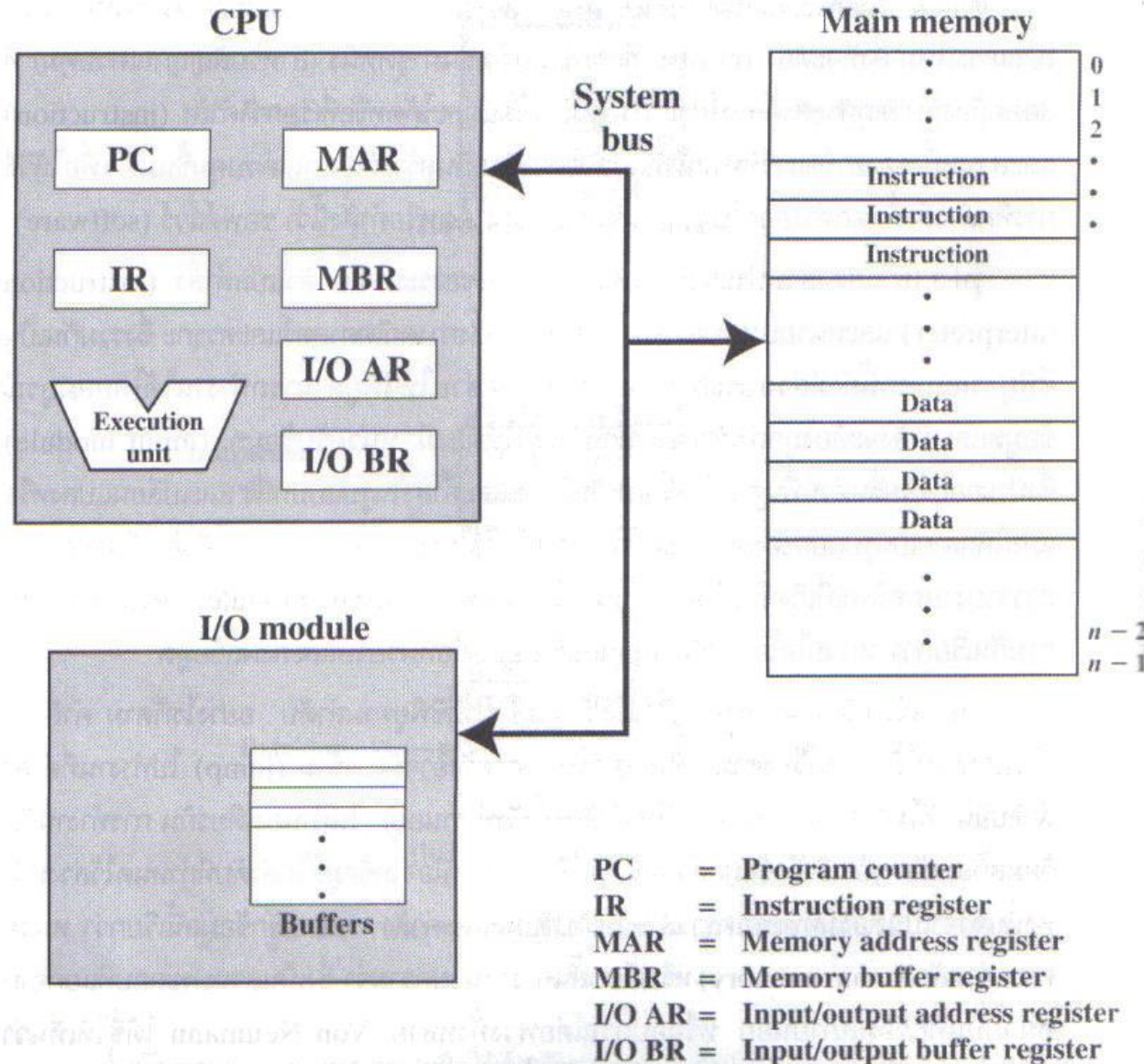
ขั้นตอนการคำนวณทางคณิตศาสตร์ หรือการคำนวณทางตรรกะกับข้อมูลกลุ่มหนึ่ง อาจต้องการสัญญาณควบคุมที่แตกต่างกันออกไป สัญญาณควบคุมแต่ละชนิดจะมีไค้เฉพาะตัวที่ไม่ซ้ำกัน ส่วนฮาร์ดแวร์ก็จะมีส่วนที่รับคำสั่ง และส่วนที่รับ ไค้คของสัญญาณควบคุม



(b) Programming in software



ส่วนประกอบเครื่องคอมพิวเตอร์



ภาพในระดับบนสุดของ
โครงสร้างส่วนประกอบ
เครื่องคอมพิวเตอร์



ส่วนประกอบเครื่องคอมพิวเตอร์

MAR (Memory Address Register) ทำหน้าที่กำหนดตำแหน่งข้อมูลในหน่วยความจำที่ต้องการอ่านหรือบันทึกข้อมูล

MBR (Memory Buffer Register) ทำหน้าที่ในการเก็บข้อมูลที่อ่านมาจากหน่วยความจำ หรือเก็บข้อมูลที่เตรียมไว้สำหรับการบันทึกลงในหน่วยความจำ

รีจิสเตอร์ *I/O-AR (I/O Address Register)* ทำหน้าที่เก็บชื่อ (หมายเลข) ของหน่วยไอโอที่ต้องการอ่านหรือบันทึกข้อมูล

รีจิสเตอร์ *I/O-BR (I/O Buffer Register)* ทำหน้าที่เก็บข้อมูลที่อ่านมาจากไอโอหรือเตรียมไว้สำหรับส่งไปยังไอโอที่ต้องการ



หน้าที่ของคอมพิวเตอร์

ประกอบด้วยสองขั้นตอนคือ ซีพียูจะอ่านคำสั่งเข้ามาเรียกว่า การดึงคำสั่ง (*Instruction fetch*) จากหน่วยความจำเข้ามาทีละคำสั่ง และประมวลผลคำสั่งนั้น

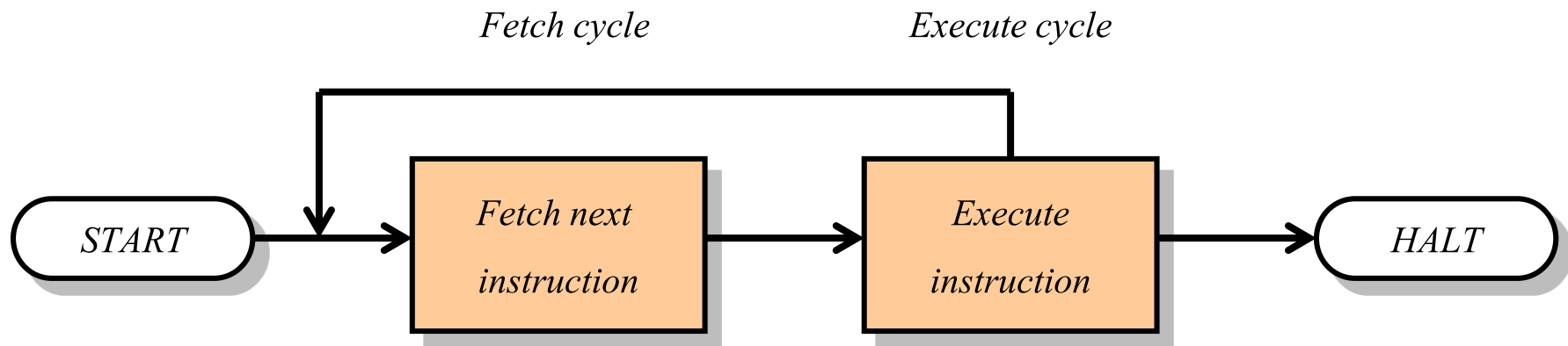
กระบวนการประมวลผลคำสั่งหนึ่งเรียกว่า วงรอบคำสั่ง (*Instruction cycle*) ซึ่งการประมวลผลโปรแกรมจะหยุดลงก็ต่อเมื่อเครื่องคอมพิวเตอร์ถูกปิด หรือเกิดข้อผิดพลาดอย่างรุนแรงที่เกิดขึ้นภายในโปรแกรม หรือพบคำสั่งในโปรแกรมที่สั่งให้เครื่องหยุดทำงาน



หน้าที่ของคอมพิวเตอร์

การดึงคำสั่ง และการประมวลผล

แสดงขั้นตอนการประมวลผลที่ประกอบด้วยสองขั้นตอนคือ วงรอบการดึงคำสั่ง (*fetch cycle*) และวงรอบการประมวลผล



หน้าที่ของคอมพิวเตอร์

การดึงคำสั่ง และการประมวลผล

คำสั่งที่ถูกดึงเข้ามาจะถูกนำไปไว้ที่รีจิสเตอร์ไออาร์ (*Instruction register: IR*) คำสั่งจะประกอบด้วยบิตต่าง ๆ ที่กำหนดให้โปรเซสเซอร์ทำงาน โปรเซสเซอร์จะนำบิตคำสั่งนั้นไปแปลความหมายและทำงานตามนั้น สิ่งที่โปรเซสเซอร์ทำนั้นแบ่งออกได้เป็น 4 กลุ่มดังนี้



หน้าที่ของคอมพิวเตอร์

การดึงคำสั่ง และการประมวลผล

1. *Processor-memory* : ข้อมูลอาจจะถูกถ่ายทอดจากหน่วยความจำมาสู่โปรเซสเซอร์ หรือจากโปรเซสเซอร์ไปสู่หน่วยความจำ
2. *Processor-I/O* : ข้อมูลอาจจะถูกถ่ายทอดมาจาก หรือส่งไปยังอุปกรณ์ต่อพ่วงผ่านส่วนติดต่อระหว่างโปรเซสเซอร์กับหน่วยไอโอ
3. *Data Processing* : โปรเซสเซอร์อาจทำการคำนวณทางคณิตศาสตร์ หรือทางตรรกะกับข้อมูล
4. *Control* : คำสั่งบางคำสั่งถูกออกแบบมาเพื่อเปลี่ยนแปลงลำดับในการประมวลผล



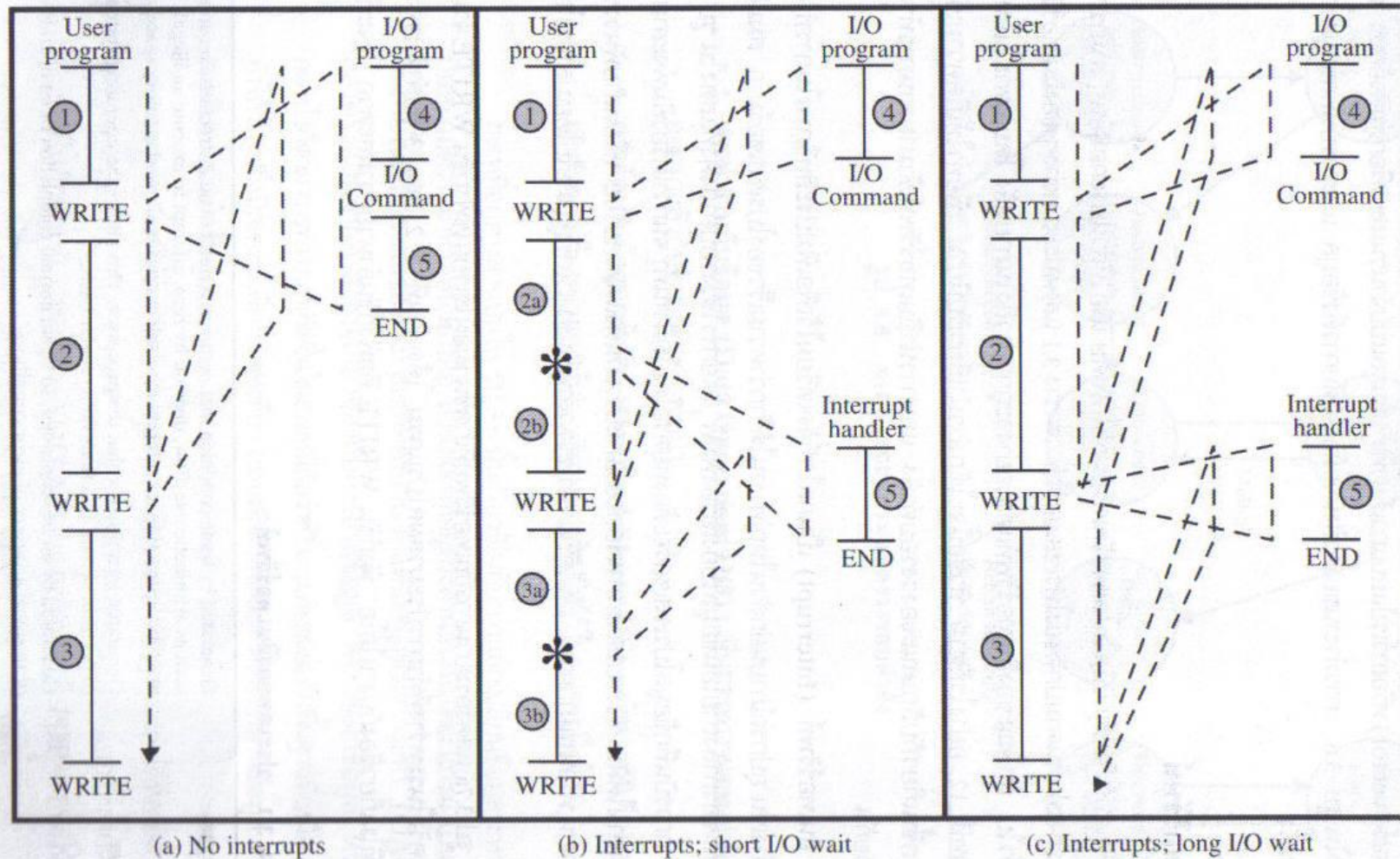
หน้าที่ของคอมพิวเตอร์

อินเทอร์รัพท์

อินเทอร์รัพท์ (Interrupt) เป็นกลไกที่จัดเตรียมไว้เพื่อเพิ่มประสิทธิภาพในการทำงาน เช่น อุปกรณ์ภายนอกส่วนใหญ่ทำงานได้ช้ากว่าความเร็วของโปรเซสเซอร์มาก สมมติว่าโปรเซสเซอร์กำลังถ่ายโอนข้อมูลไปยังเครื่องพิมพ์ โดยการไต่ถามวนคำสั่ง หลังจากการบันทึกข้อมูลโปรเซสเซอร์จะต้องหยุดรอโดยไม่มีการทำงาน จนกว่าเครื่องพิมพ์จะสามารถทำงานตามได้ทัน ช่วงระยะเวลาการรอกอนี้อาจยาวนานตั้งแต่หลายร้อย หรือหลายพันวงรอบการทำงาน หรือนานมากกว่านี้ เห็นได้ชัดเจนว่าช่วงเวลานี้เป็นช่วงเวลาที่สูญเสียไปโดยเปล่าประโยชน์จริง ๆ



หน้าที่ของคอมพิวเตอร์ (อินเทอร์รัพท์)



ภาพแสดงการไหลเวียนการควบคุมแบบที่มีและไม่มีอินเทอร์รัพท์

หน้าที่ของคอมพิวเตอร์

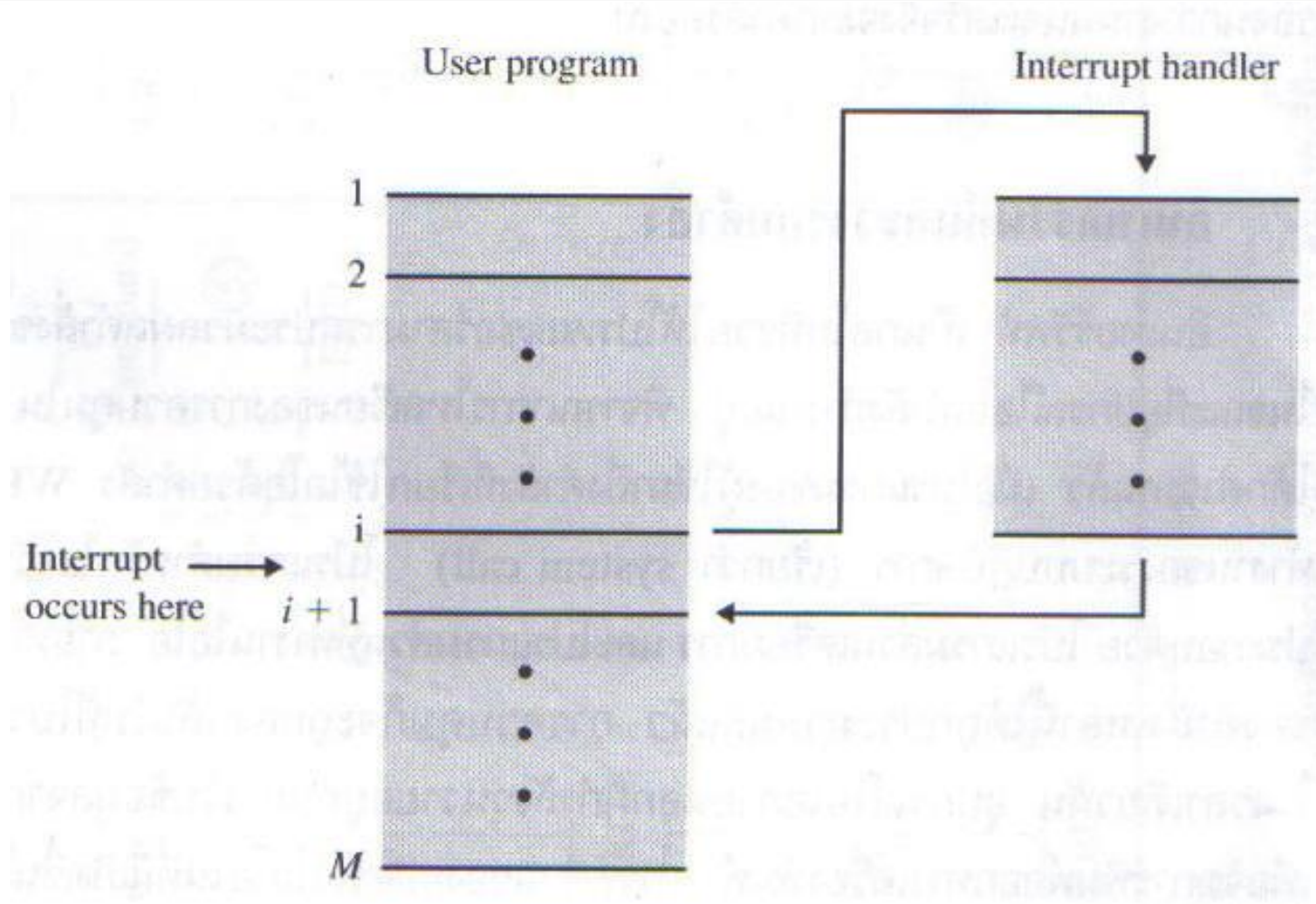
อินเทอร์เน็ต และวงรอบคำสั่ง

จากมุมมองของโปรแกรมผู้ใช้ กระบวนการอินเทอร์เน็ตคือ การขัดจังหวะ การประมวลผลตามปกติของโปรแกรม เมื่อกระบวนการอินเทอร์เน็ตเสร็จสิ้นลง การประมวลผลของโปรแกรมเดิมก็จะดำเนินต่อไป

ดังนั้น โปรแกรมของผู้ใช้จึงไม่จำเป็นจะต้องมีคำสั่งสำหรับการจัดการอินเทอร์เน็ตแต่อย่างใด โปรเซสเซอร์และระบบปฏิบัติการจะเป็นส่วนที่รับผิดชอบในการหยุดการประมวลผล และทำการประมวลผลโปรแกรมผู้ใช้



หน้าที่ของคอมพิวเตอร์ อินเทอร์เน็ต และวงรอบคำสั่ง



การส่งมอบการควบคุมผ่านอินเทอร์เน็ต

หน้าที่ของคอมพิวเตอร์

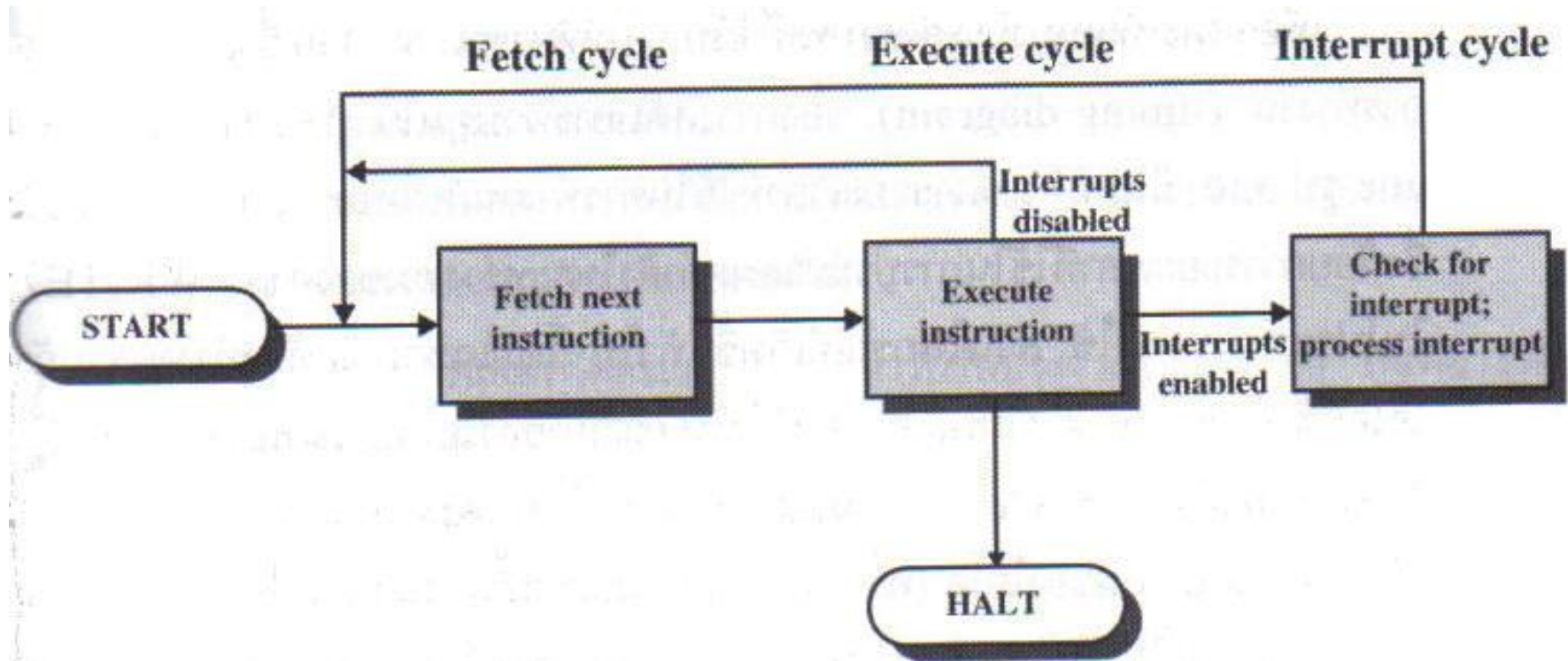
อินเทอร์รัพท์ และวงรอบคำสั่ง

ในการสนับสนุนการทำงานของอินเทอร์รัพท์ ระบบคอมพิวเตอร์ได้เพิ่มวงรอบอินเทอร์รัพท์ (*Interrupt cycle*) เข้าไปในวงรอบอินเทอร์รัพท์ โปรเซสเซอร์จะตรวจสอบว่ามีสัญญาณอินเทอร์รัพท์เกิดขึ้นหรือไม่ ถ้าไม่มีก็จะดำเนินวงรอบการดึงคำสั่งต่อไปเข้ามาประมวลผล แต่ถ้ามีสัญญาณอินเทอร์รัพท์เกิดขึ้น โปรเซสเซอร์ก็จะทำงาน



หน้าที่ของคอมพิวเตอร์

อินเทอร์รัพท์ และวงรอบคำสั่ง

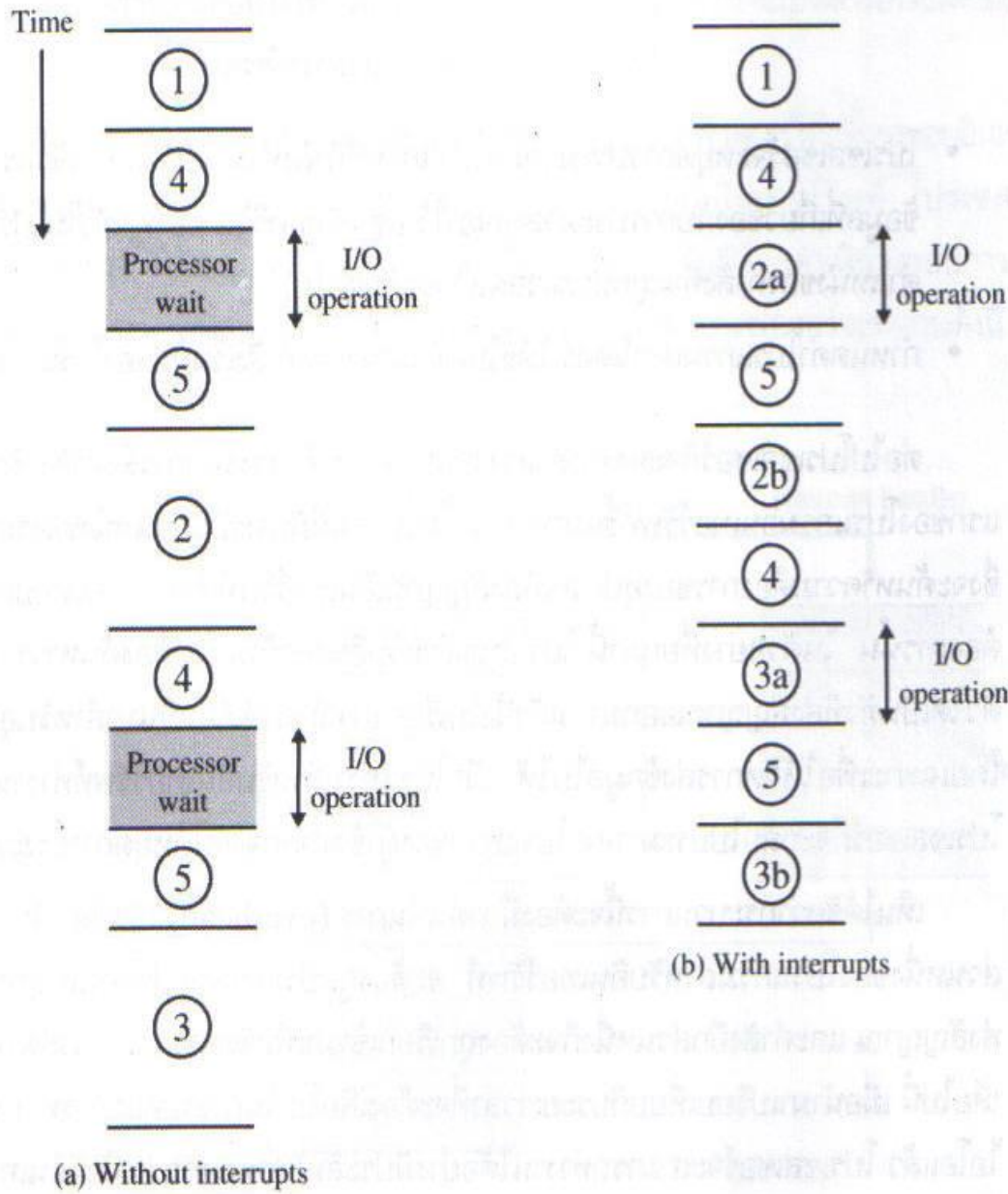


วงรอบคำสั่งพร้อมด้วยอินเทอร์รัพท์

หน้าที่ของคอมพิวเตอร์

อินเทอร์รัพท์ และวงรอบคำสั่ง

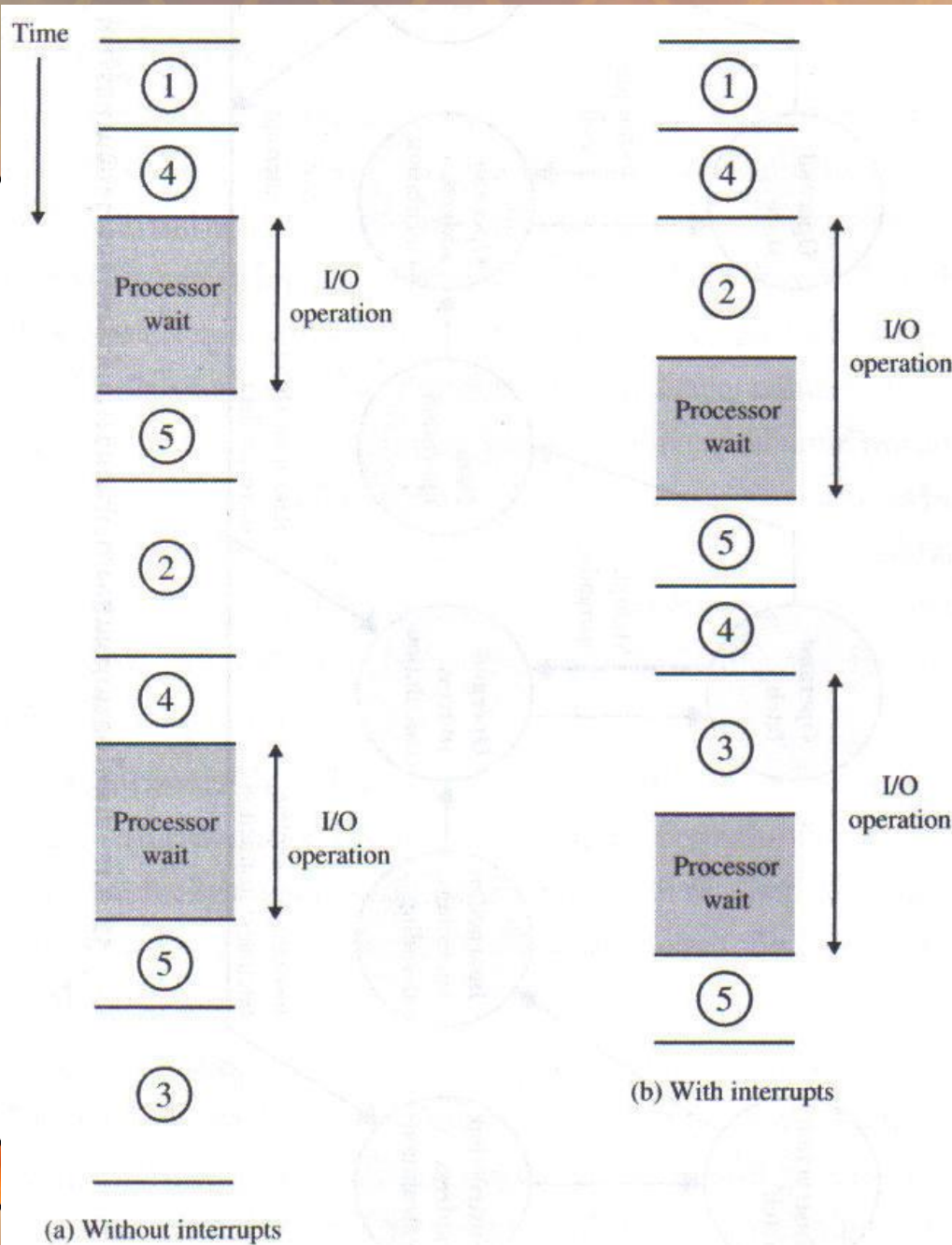
ตารางเวลาการทำงานของโปรแกรม
(มีการรอคอยไอโอเล็กน้อย)



หน้าที่ของคอมพิวเตอร์

อินเทอร์เน็ต และวงรอบคำสั่ง

ตารางเวลาการทำงานของโปรแกรม
(มีการรอคอยไอโอนาน)



หน้าที่ของคอมพิวเตอรื

การใช้อินเทอร์รัพท์ซ้อน

การแก้ปัญหาอินเทอร์รัพท์ซ้อนนี้มีสองแนวทาง แนวทางแรกจะใช้วิธีการยกเลิกการใช้สัญญาณอินเทอร์รัพท์เป็นการชั่วคราว (*disable interrupt*) ในขณะที่โปรเซสเซอร์กำลังประมวลผลโปรแกรมสำหรับอินเทอร์รัพท์ตัวใดตัวหนึ่งอยู่

การยกเลิกเป็นการชั่วคราวนี้ หมายถึง การทำให้โปรเซสเซอร์เพิกเฉยต่อสัญญาณอินเทอร์รัพท์ที่เกิดขึ้น สัญญาณอินเทอร์รัพท์ที่เกิดขึ้นในช่วงเวลานี้จึงถูกเก็บรักษาไว้ และจะถูกตรวจสอบเมื่อโปรเซสเซอร์กลับมาทำงานตามปกติ



หน้าที่ของคอมพิวเตอร์

การใช้อินเทอร์เน็ตที่ซ้อน

ข้อเสียของแนวทางแรก คือ การที่ไม่สามารถจัดการกับอินเทอร์เน็ตที่อาจมีลำดับความสำคัญมากกว่า หรือมีความเร่งด่วนมากกว่าได้เลย

ตัวอย่าง เมื่อข้อมูลเข้ามาถึงทางสายสื่อสารอาจมีความจำเป็นจะต้องดึงข้อมูลนั้นเข้ามาเก็บในหน่วยความจำทันที เพื่อเปิดทางให้กับข้อมูลอื่น ๆ ที่ตามมา ซึ่งถ้าไม่สามารถเก็บข้อมูลชุดแรกไว้ก่อนที่ข้อมูลชุดที่สองจะมาถึงแล้ว ข้อมูลชุดแรกก็จะถูกลบทิ้งไปโดยอัตโนมัติ



หน้าที่ของคอมพิวเตอร์

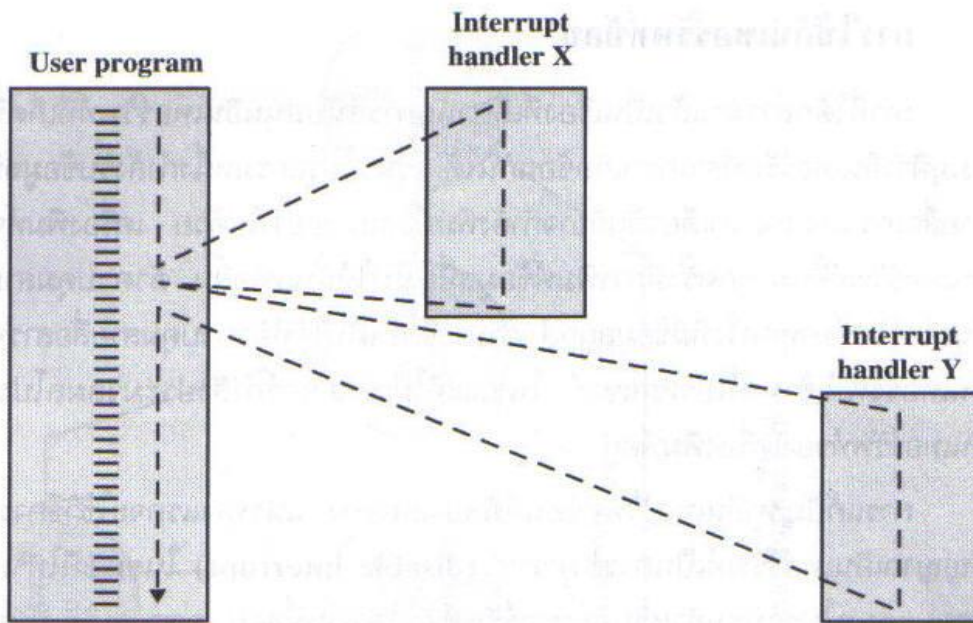
การใช้อินเทอร์เน็ตรัพท์ซ้อน

แนวทางที่สองคือ การกำหนดลำดับความสำคัญให้กับอินเทอร์เน็ตรัพท์ทุกตัว และยินยอมให้อินเทอร์เน็ตรัพท์ที่มีลำดับความสำคัญสูงกว่า บังคับให้อินเทอร์เน็ตรัพท์ตัวที่มีลำดับความสำคัญน้อยกว่าถูกขัดจังหวะการทำงานได้เช่นเดียวกับโปรแกรมผู้ใช้ทั่วไป

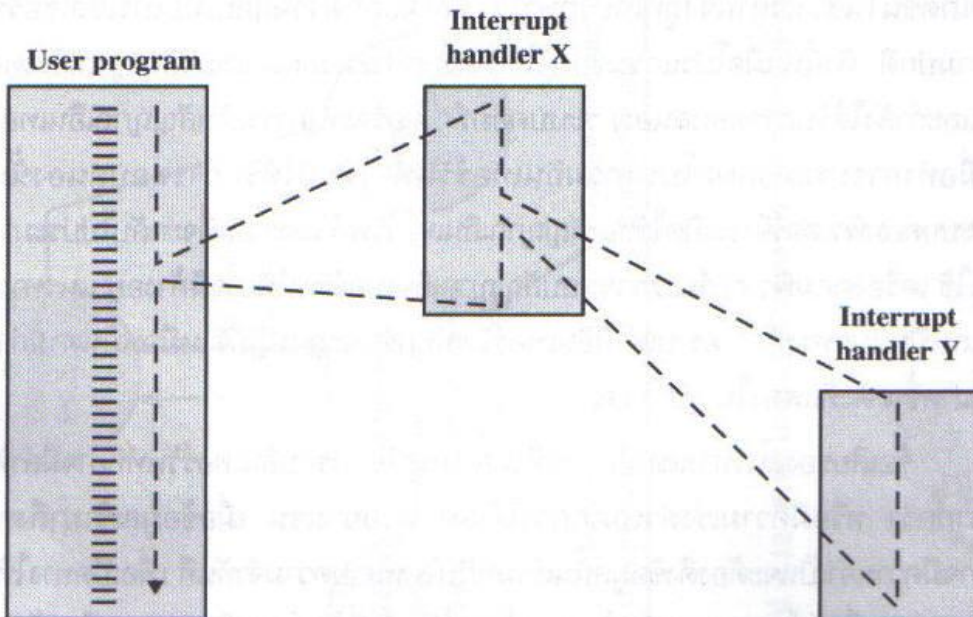


หน้าที่ของคอมพิวเตอร์

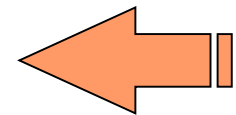
การใช้อินเทอร์รัพท์ซ้อน



(a) Sequential interrupt processing



(b) Nested interrupt processing

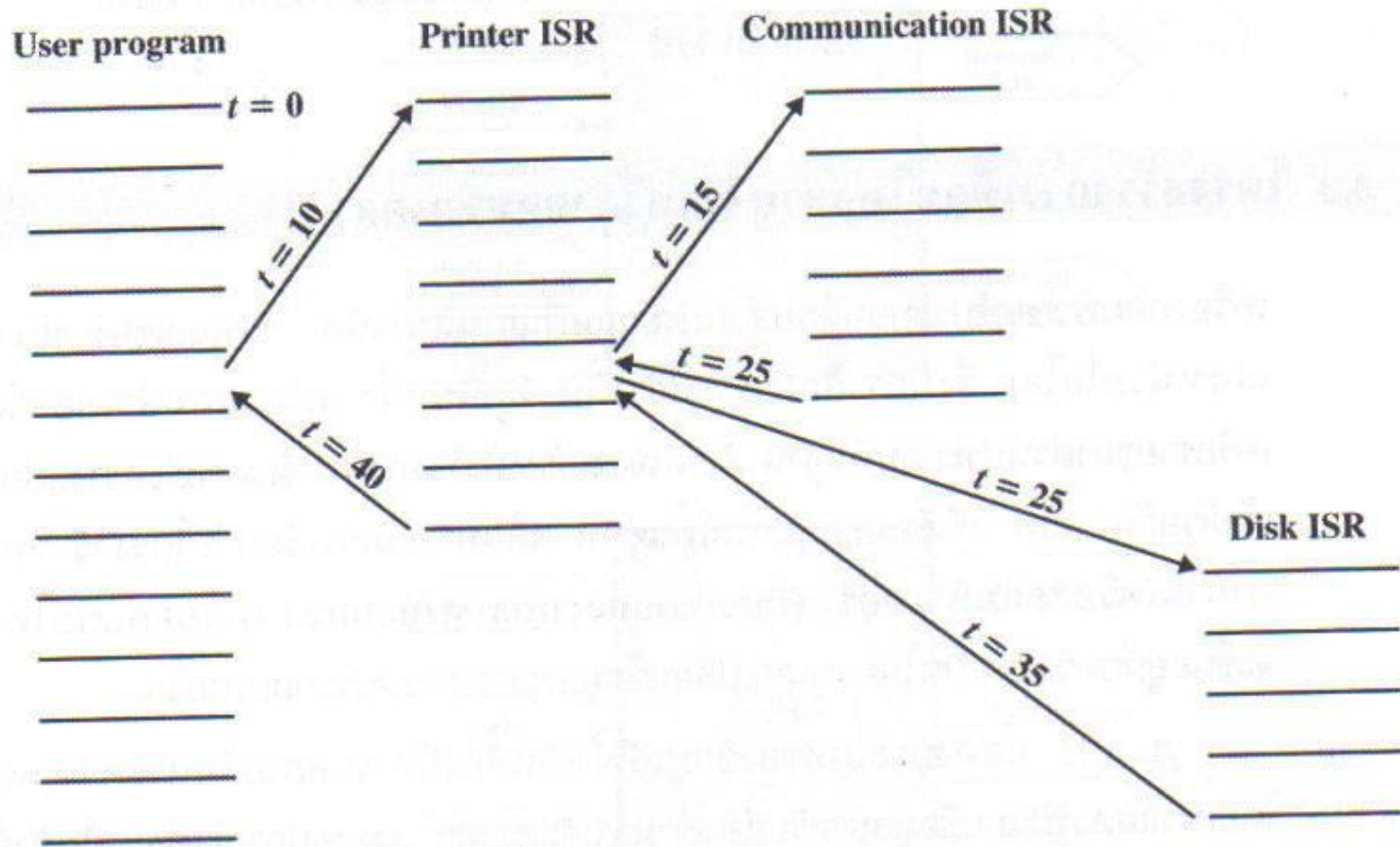


การส่งผ่านการควบคุมที่มี
การอินเทอร์รัพท์ซ้อน



หน้าที่ของคอมพิวเตอร์

การใช้อินเทอร์รัพท์ซ้อน



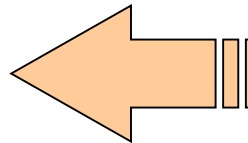
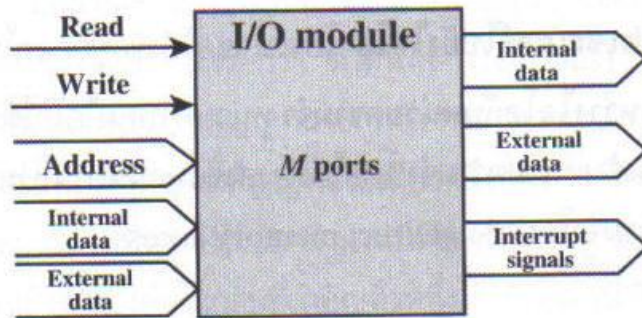
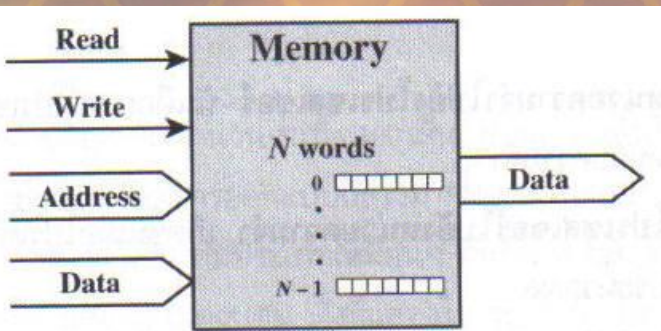
สมมติให้คอมพิวเตอร์เครื่องหนึ่งมีอุปกรณ์ไอโอสามอย่าง คือ เครื่องพิมพ์ ดิสก์ และสายสื่อสาร ซึ่งมีลำดับความสำคัญเป็น 2, 4 และ 5 ตามลำดับ

โครงสร้างการเชื่อมโยงภายในเครื่องคอมพิวเตอร์

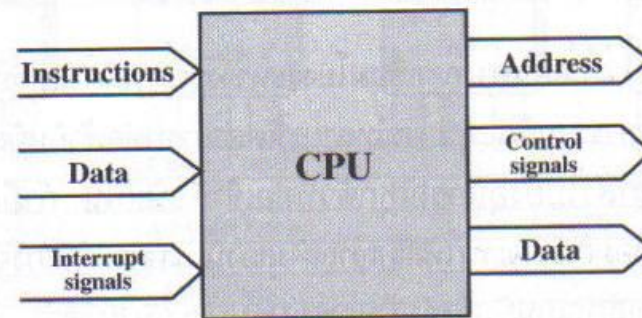
โครงสร้างการเชื่อมโยงภายในเครื่องคอมพิวเตอร์ (*interconnection structure*) คือ เส้นทางที่เชื่อมต่ออุปกรณ์พื้นฐานทุกเส้นทางรวมกัน การออกแบบโครงสร้างจะขึ้นอยู่กับความจำเป็นในการแลกเปลี่ยนข้อมูลระหว่างส่วนประกอบภายใน



โครงสร้างการเชื่อมโยงภายในเครื่องคอมพิวเตอร์



ส่วนประกอบของเครื่องคอมพิวเตอร์



โครงสร้างการเชื่อมโยงภายในเครื่องคอมพิวเตอร์

◎ หน่วยความจำ (*memory*) : หน่วยความจำประกอบด้วยเซลล์ความจำขนาด N หน่วย (*words*) ซึ่งแต่ละหน่วยจะต้องมีขนาดเท่ากันทั้งหมด แต่ละหน่วยจะถูกตั้งชื่อสำหรับการอ้างอิงโดยการใช้หมายเลข ($0, 1, 2, \dots$) ข้อมูลจะถูกอ่านหรือบันทึกลงในหน่วยความจำครั้งละหนึ่งหน่วย ลักษณะการทำงานของหน่วยความจำจะถูกควบคุมด้วยสัญญาณ *READ* หรือ *WRITE* ตำแหน่งของข้อมูลที่ถูกอ่านหรือถูกบันทึกจะกำหนดโดย “*Address*” และข้อมูลจะถูกส่งเข้ามาหรือส่งออกไปผ่าน “*Data*”



โครงสร้างการเชื่อมโยงภายในเครื่องคอมพิวเตอร์

◎ หน่วยไอโอ (I/O module) : การทำงานสองแบบ คือ การอ่านข้อมูล (read) และการบันทึกข้อมูล (write) และอาจเชื่อมต่อกับอุปกรณ์ไอโอได้มากกว่าหนึ่งอย่าง ซึ่งอุปกรณ์แต่ละอย่างจะถูกอ้างอิงถึงด้วย พอร์ต (port) โดยแต่ละพอร์ตจะมีหมายเลขเฉพาะเป็นของตนเอง ยังมีเส้นทางเชื่อมโยงระหว่างหน่วยควบคุมไอโอกับอุปกรณ์ต่าง ๆ และหน่วยควบคุมไอโอสามารถส่งสัญญาณอินเทอร์รัพท์ไปยังโปรเซสเซอร์ได้

◎ โปรเซสเซอร์ (Processor) : โปรเซสเซอร์อ่านคำสั่ง และข้อมูล บันทึกผลลัพธ์ภายหลังการประมวลผล และใช้สัญญาณควบคุมในการควบคุมการทำงานของระบบคอมพิวเตอร์ทั้งระบบ และโปรเซสเซอร์ยังสามารถรับสัญญาณอินเทอร์รัพท์ได้



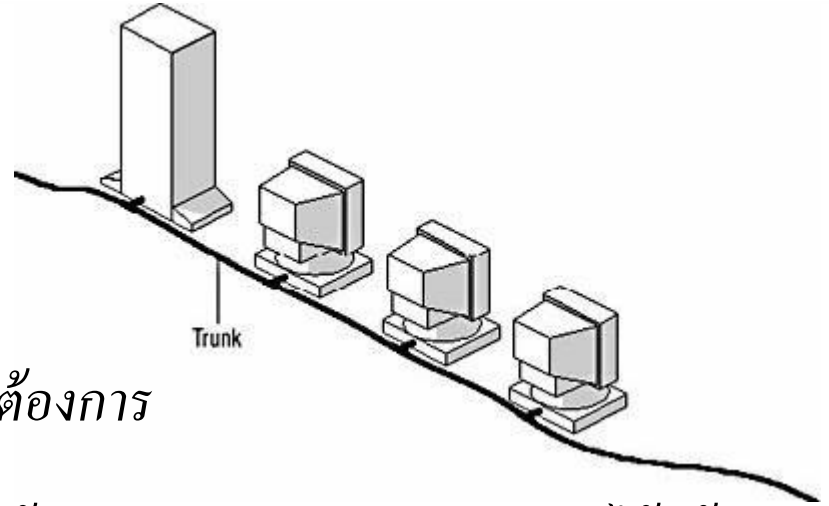
การเชื่อมโยงโดยใช้บัส

บัส (bus) เป็นเส้นทางการเชื่อมโยงระหว่างอุปกรณ์ตั้งแต่สองชนิดขึ้นไป ที่มีลักษณะเด่นตรงที่ใช้สายสื่อสารร่วมกัน ทำให้การส่งสัญญาณจากอุปกรณ์ตัวหนึ่งถูกส่งไปยังอุปกรณ์ทุกตัวที่ต่อเข้ากับบัส ถ้ามีอุปกรณ์สองตัวขึ้นไปส่งสัญญาณออกมาพร้อมกัน ก็จะทำให้สัญญาณทั้งสองนั้นรบกวนกันเองจนไม่สามารถนำไปใช้งานได้ ดังนั้น การส่งสัญญาณบนบัสจึงจำเป็นต้องรับประกันให้ได้ว่า มีอุปกรณ์เพียงตัวเดียวเท่านั้นที่จะได้รับอนุญาตให้ส่งสัญญาณ ณ เวลาใดเวลาหนึ่ง



การเชื่อมโยงโดยใช้บัส

1. บัสประกอบด้วยสายสื่อสารจำนวนหลายเส้น
2. ส่งสัญญาณที่ใช้แทนข้อมูล 0 หรือ 1
3. สามารถส่งข้อมูลติดต่อกันได้ตามระยะเวลาที่ต้องการ
4. เมื่อนำสายสัญญาณหลายเส้นมารวมกัน ก็ทำให้สามารถส่งสัญญาณหลายบิตได้พร้อมกัน



*** บัสที่เชื่อมโยงอุปกรณ์หลักของเครื่องคอมพิวเตอร์เข้าด้วยกันนั้นเรียกว่า
บัสหลัก (system bus)



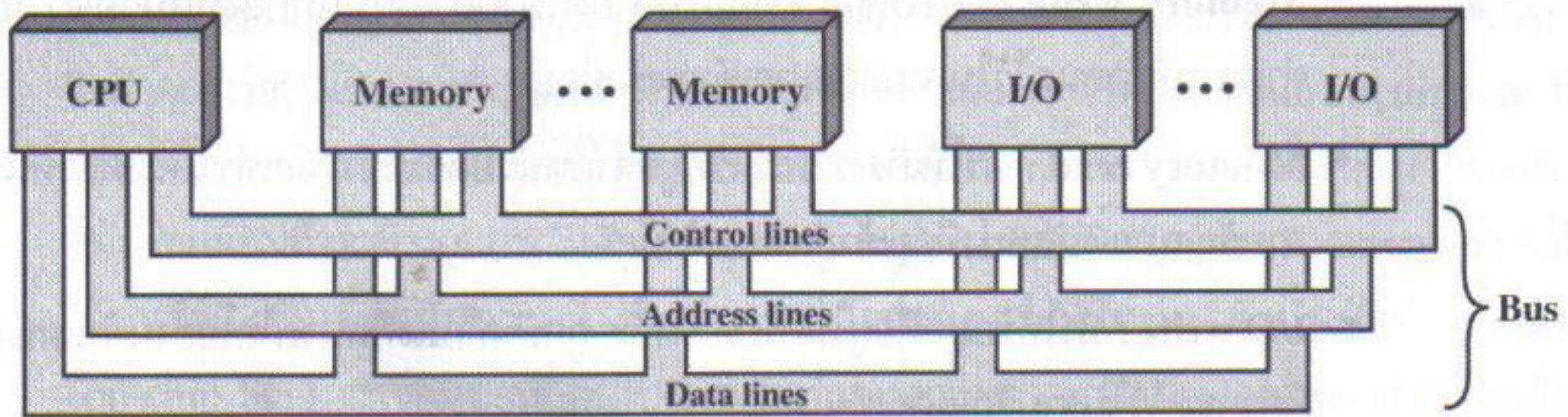
โครงสร้างแบบบัส

ระบบบัสหลักประกอบด้วยสายสัญญาณจำนวน 50 ถึง 100 เส้น สายแต่ละเส้นได้รับการกำหนดความหมายหรือหน้าที่เฉพาะ ไว้สามประเภทเหมือน ๆ กัน คือ สายสัญญาณข้อมูล สายสัญญาณตำแหน่งที่อยู่ และสายสัญญาณควบคุม

สายสัญญาณข้อมูล (*data line*) เป็นเส้นทางที่ใช้ในการเคลื่อนย้ายข้อมูลระหว่างอุปกรณ์ สายประเภทนี้มักจะมีอยู่หลายเส้น และรวมเรียกว่า คาต้าบัส (*data bus*) จำนวนสายที่มีนั้นอาจมีตั้งแต่ 32 ถึง 100 เส้น จำนวนสายสัญญาณจะถูกเรียกว่า ความกว้างของช่องสัญญาณ (*width*)



โครงสร้างแบบบัส



แบบแผนการเชื่อมต่อแบบบัส



โครงสร้างแบบบัส

สายสัญญาณตำแหน่งที่อยู่ (*address line*) ใช้ในการกำหนดตำแหน่งของแหล่งที่มาของข้อมูล (*source*) หรือแหล่งรับข้อมูล (*destination*) สำหรับข้อมูลที่อยู่ในสายดาต้าบัส

นอกจากนี้ สายสัญญาณตำแหน่งที่อยู่ยังทำหน้าที่ในการบอกชื่อของอุปกรณ์ไอโอหรือพอร์ตที่ต้องการ โดยแบ่งออกเป็นสองส่วน ส่วนหัวหรือ *Higher-order bits* มักจะใช้ในการเลือกอุปกรณ์ที่ต้องการใช้ในบัส และส่วนหาง หรือ *lower-order bits* จะใช้เลือกตำแหน่งในหน่วยความจำ หรือหมายเลขพอร์ตในหน่วยควบคุมนั้น ๆ



โครงสร้างแบบบัส

สายสัญญาณควบคุม (control line) ใช้ในการควบคุมการใช้งานสายสัญญาณข้อมูล และสายสัญญาณตำแหน่งที่อยู่ ประกอบด้วย คำสั่งและช่วงเวลาที่เกี่ยวข้องระหว่างอุปกรณ์ในระบบ สัญญาณช่วงเวลา (timing signal) บอกให้ทราบว่า ข้อมูลในสายสัญญาณทั้งหมดนั้นเป็นสัญญาณที่กำลังใช้งานอยู่หรือไม่ สัญญาณคำสั่ง (command signal) บอกชนิดของงานที่จะต้องทำโดยทั่วไปสายสัญญาณควบคุมเป็นดังนี้

◎ Memory write : ทำให้ข้อมูลที่อยู่ในสายดาต้าบัส ถูกบันทึกลงในหน่วยความจำตามตำแหน่งที่กำหนดไว้ในสายสัญญาณตำแหน่งที่อยู่

◎ Memory read : ทำให้เกิดการอ่านข้อมูลจากหน่วยความจำ จากตำแหน่งที่กำหนดไว้ในสายสัญญาณตำแหน่งที่อยู่เข้าสู่สายดาต้าบัส



โครงสร้างแบบบัส (สายสัญญาณควบคุม)

- ◎ *I/O write* : ทำให้ข้อมูลที่อยู่ในสายดาต้าบัส ถูกบันทึกลงที่หน่วยไอโอตามตำแหน่งพอร์ตที่กำหนดไว้ในสายสัญญาณตำแหน่งที่อยู่
- ◎ *I/O read* : ทำให้เกิดการอ่านข้อมูลจากหน่วยไอโอตามตำแหน่งพอร์ตที่กำหนดไว้ในสายสัญญาณตำแหน่งที่อยู่เข้าสู่สายดาต้าบัส
- ◎ *Transfer ACK* : การอ่านหรือการบันทึกข้อมูลนั้น เสร็จเรียบร้อยแล้ว
- ◎ *Bus request* : อุปกรณ์ตัวที่ส่งสัญญาณออกมานั้นต้องการใช้บัส
- ◎ *Bus grant* : อุปกรณ์ตัวที่ส่งสัญญาณแสดงความต้องการใช้บัสนั้นได้รับอนุญาตให้ใช้บัสได้



โครงสร้างแบบบัส (สายสัญญาณควบคุม)

- ◎ *Interrupt request* : มีสัญญาณอินเทอร์รัพท์รออยู่
- ◎ *Interrupt ACK* : โปรเซสเซอร์บอกให้ทราบว่า อินเทอร์รัพท์ที่รอนั้นจะได้รับการประมวลผล
- ◎ *Clock* : ใช้ในการเทียบจังหวะสัญญาณนาฬิกากับอุปกรณ์ต่าง ๆ
- ◎ *Reset* : สัญญาณบอกให้อุปกรณ์ทุกตัว เริ่มกระบวนการเตรียมการเพื่อให้พร้อมใช้งาน



โครงสร้างแบบบัส

สรุปการทำงานของบัส (bus)

ถ้าอุปกรณ์ตัวหนึ่งต้องการส่งข้อมูลไปยังอุปกรณ์อีกตัวหนึ่งจะต้องทำสองอย่าง อย่างแรกจะต้องได้รับอนุญาตให้ใช้บัส อย่างที่สองจัดการส่งข้อมูลผ่านบัส ในกรณีที่อุปกรณ์ตัวหนึ่งต้องการอ่านข้อมูลจากอุปกรณ์อีกตัวหนึ่งจะต้องทำสองอย่าง อย่างแรกจะต้องได้รับอนุญาตให้ใช้บัส อย่างที่สองจะต้องส่งความต้องการไปยังอุปกรณ์ตัวนั้น โดยใช้สายสัญญาณควบคุม และสายสัญญาณบอกตำแหน่งที่ถูกต้อง จากนั้นก็ต้องรอให้อุปกรณ์ตัวนั้นส่งข้อมูลมาให้

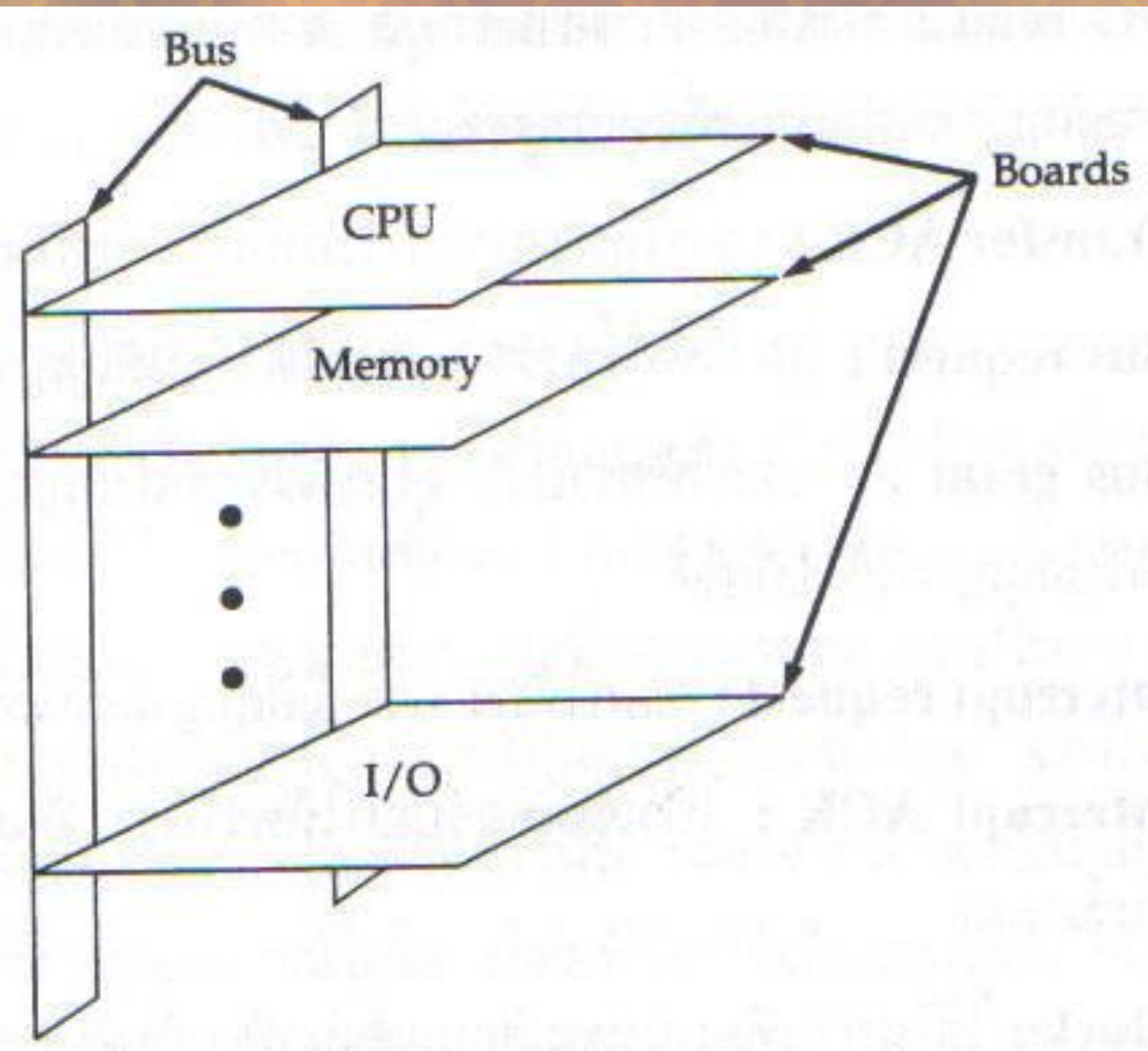


โครงสร้างแบบบัส

ในทางกายภาพ บัสหลัก (*system bus*) ก็คือตัวนำไฟฟ้าจำนวนหนึ่ง ซึ่งถูกห่อหุ้มหรือฝังลงไปบนบอร์ดเรียกว่า “*printed circuit board*” สายบัสจะถูกวางไว้ทั่วทั้งบอร์ด เพื่อเชื่อมต่อกับอุปกรณ์ต่าง ๆ ซึ่งอุปกรณ์แต่ละตัว อาจจะเชื่อมต่อเข้ากับสายสัญญาณทุกเส้นหรือบางเส้นก็ได้



โครงสร้างแบบบัส



ภาพแสดงโครงสร้าง
ของการเชื่อมต่อโดยใช้บัส



โครงสร้างลำดับชั้นของ巴士หลายระดับ

ถ้ามีอุปกรณ์จำนวนมากเชื่อมต่อเข้ากับ巴士 ก็จะทำให้ประสิทธิภาพการทำงานของ巴士 ลดต่ำลงเนื่องจาก

1. ถ้ามีอุปกรณ์จำนวนมากเชื่อมต่อเข้ากับ巴士 ก็จะทำให้巴士มีความยาวมากขึ้น ซึ่งก็จะทำให้การสื่อสารใน巴士ใช้ระยะเวลาหน่วงนานมากขึ้น
2. 巴士อาจกลายเป็นจุดคอขวดในการสื่อสาร เมื่อความต้องการใช้งาน巴士ของอุปกรณ์ต่าง ๆ เพิ่มมากขึ้นจนถึงจุดอิ่มตัวในการให้บริการ



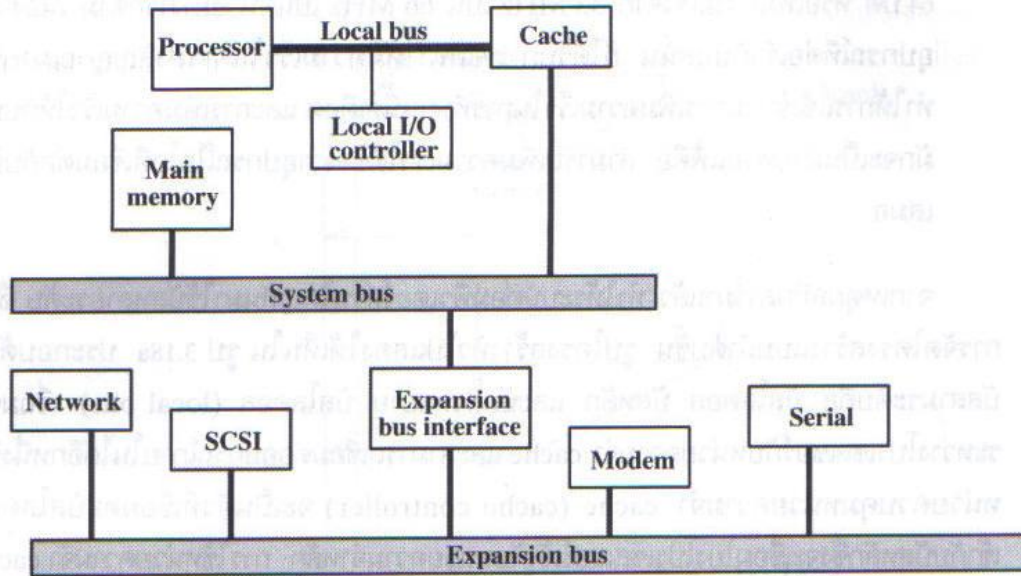
โครงสร้างลำดับชั้นของ巴士หลายระดับ

ประสิทธิภาพของหน่วยควบคุมการทำงานของอุปกรณ์ไอโอ (I/O controller) คือการใช้巴士ส่วนขยาย (expansion bus) ในการเชื่อมต่ออุปกรณ์ไอโอ โดยใช้บัฟเฟอร์ขนาดเล็กเป็นที่สำหรับพักข้อมูลที่มีการส่งผ่านระหว่างกัน

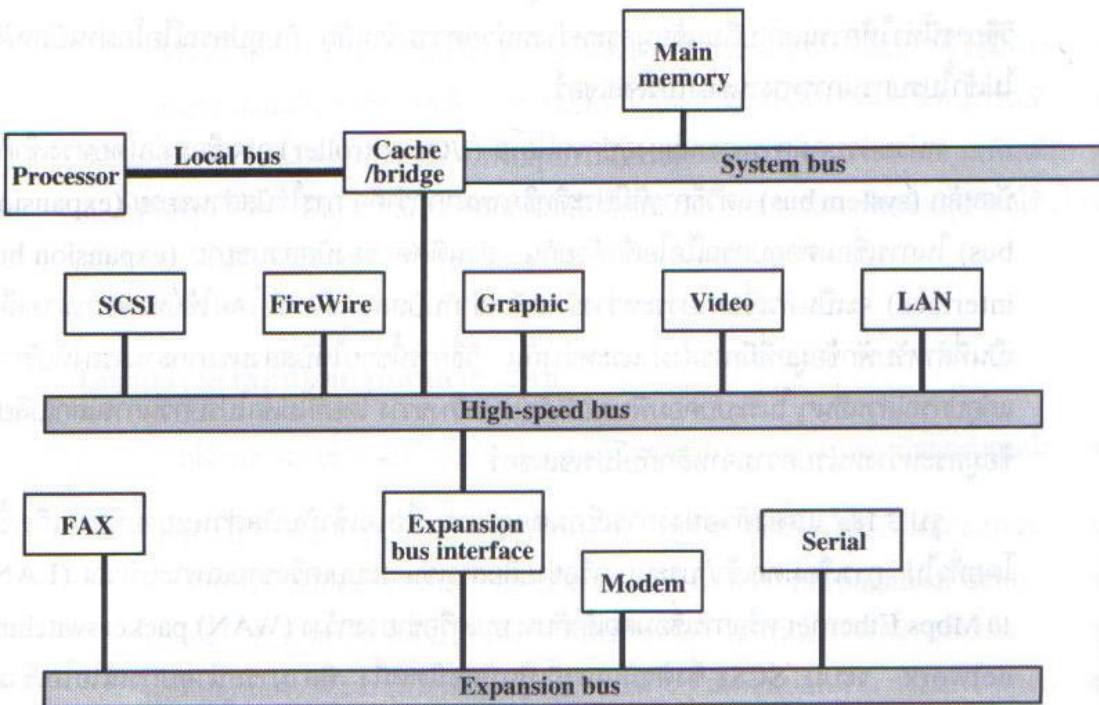
วิธีการนี้ช่วยให้巴士ส่วนขยายสามารถให้บริการแก่อุปกรณ์ส่วนอื่น ๆ ในระบบคอมพิวเตอร์ได้อย่างกว้างขวาง โดยที่ไม่เข้าไปรบกวนการแลกเปลี่ยนข้อมูลระหว่างหน่วยความจำหลักกับโปรเซสเซอร์



โครงสร้างลำดับชั้นของ บัสหลายระดับ



(a) Traditional bus architecture



(b) High-performance architecture

ตัวอย่างการนำโครงสร้าง
แบบบัสมาใช้งาน



โครงสร้างลำดับชั้นของ巴士หลายระดับ

ข้อดีของโครงสร้างแบบนี้คือ巴士ความเร็วสูงสนับสนุนการทำงานของอุปกรณ์ไอโอความเร็วสูงให้ใกล้ชิดกับโปรเซสเซอร์ แต่ในเวลาเดียวกันก็ยังคงเป็นอิสระจากกันและกัน ทำให้อุปกรณ์ทั้งสองชนิดยังคงสามารถทำงานของตัวเองในเวลาเดียวกันได้ นอกจากนี้ การเปลี่ยนแปลงที่เกิดขึ้นในโปรเซสเซอร์ หรือใน巴士ความเร็วสูงจะไม่มีผลกระทบซึ่งกันและกัน



ประเด็นการพิจารณาในการออกแบบบัส

ประเภทของบัส

สายบัสแยกออกได้เป็นสองประเภท คือ สายแบบ *dedicated* และสายแบบ *multiplexed* สายบัสประเภท *dedicated* นั้น ได้รับการกำหนดหน้าที่ไว้อย่างถาวรให้ทำงานอย่างใดอย่างหนึ่ง หรือถูกกำหนดให้ใช้งานในบางระบบย่อยของคอมพิวเตอร์

ข้อดีของการใช้สายสัญญาณแบบ *time multiplexing* คือมีความต้องการจำนวนสายสัญญาณน้อยกว่า ซึ่งสามารถช่วยลดพื้นที่แผงวงจรหลัก ก็คือการลดค่าใช้จ่ายนั่นเอง ข้อเสียก็คือการที่ต้องใช้แผงวงจรที่มีความซับซ้อนมากกว่าในหน่วยควบคุมอุปกรณ์

Physical dedication หมายถึง การใช้สายสัญญาณหลายประเภท โดยที่สายแต่ละประเภทจะเชื่อมต่อเข้ากับอุปกรณ์บางส่วนอย่างถาวร



ประเด็นการพิจารณาในการออกแบบบัส

วิธีการตัดสินใจขาด

แบ่งออกเป็นสองประเภท คือ การแก้ปัญหาแบบรวมศูนย์ และการแก้ปัญหาแบบกระจายศูนย์

ในแบบรวมศูนย์ (*centralized scheme*) จะมีอุปกรณ์ตัวหนึ่งเรียกว่า หน่วยควบคุมบัส (*bus controller or arbiter*) ทำหน้าที่ในการจัดตารางเวลาการใช้บัสให้แก่อุปกรณ์ทุกตัวที่เชื่อมต่อเข้าบัสนั้น อุปกรณ์ตัวนี้อาจเป็นหน่วยแยกต่างหาก หรือบางครั้งก็ถูกรวมเข้าไว้เป็นส่วนหนึ่งของโปรเซสเซอร์



ประเด็นการพิจารณาในการออกแบบบัส

วิธีการตัดสินใจขาด

ในแบบกระจาย (*distributed scheme*) จะไม่มีหน่วยควบคุมการใช้งานบัส หน่วยควบคุมอุปกรณ์ที่เชื่อมต่อเข้ากับบัส จะมีวงจรพิเศษในการควบคุมการใช้บัสอยู่ในตัวเอง และหน่วยควบคุมทุกตัวจะทำหน้าที่ร่วมกันในการใช้งานบัสร่วมกัน วิธีการควบคุมการใช้บัสทั้งสองแบบ มีวัตถุประสงค์เหมือนกัน คือการเลือกอุปกรณ์เพียงตัวเดียว คือ โปรเซสเซอร์หรือหน่วยไอโอให้ทำหน้าที่เป็นมาสเตอร์ (*master*) ซึ่งจะมีสิทธิในการใช้บัสในการถ่ายเทข้อมูล ซึ่งจะมีอุปกรณ์อีกอย่างน้อยหนึ่งตัวทำหน้าที่เป็นสเลฟ (*slave*) ทำหน้าที่รับหรือส่งข้อมูลไปยังมาสเตอร์



ประเด็นการพิจารณาในการออกแบบบัส

จังหวะเวลา

จังหวะเวลา (*timing*) หมายถึง วิธีการที่ควบคุมให้เหตุการณ์ต่าง ๆ สามารถทำงานร่วมกันได้บนบัสซึ่งมีอยู่สองวิธีคือ ซิงโครนัส และอะซิงโครนัส

การใช้จังหวะเวลา แบบซิงโครนัส (*Synchronous timing*) อาศัยการควบคุมจังหวะการทำงานด้วยสัญญาณนาฬิกา (*clock*) ซึ่งจะส่งสัญญาณนาฬิกา “0” และ “1” ออกมาอย่างสม่ำเสมอ การส่งสัญญาณ 1 – 0 หนึ่งครั้งเรียกว่า วงรอบนาฬิกา (*clock cycle*) หรือวงรอบบัส (*bus cycle*)

การใช้จังหวะเวลา แบบอะซิงโครนัส (*Asynchronous timing*) เหตุการณ์หนึ่งที่เกิดขึ้นบนบัสจะเกิดขึ้นตามหลังเหตุการณ์ที่เกิดขึ้นก่อนหน้านี้



ประเด็นการพิจารณาในการออกแบบ巴士

จังหวะเวลา

สรุปได้ว่า การส่งสัญญาณแบบชิงโครนัส มีความซับซ้อนน้อยกว่าในการสร้างขึ้นมำใช้งาน และการทดสอบ อย่างไรก็ตาม ก็มีความคล่องตัวน้อยกว่าแบบอะชิงโครนัส เนื่องจากอุปกรณ์ทุกชนิดบน巴士แบบชิงโครนัสจะต้องใช้สัญญาณนาฬิกา ร่วมกัน ทำให้ระบบไม่สามารถได้รับประโยชน์จากอุปกรณ์ที่ได้รับการพัฒนาประสิทธิภาพให้ก้าวหน้ามากขึ้น ในระบบอะชิงโครนัส อุปกรณ์ที่ทำงานช้าและเร็ว อุปกรณ์ที่ใช้เทคโนโลยีใหม่ล่าสุดเทคโนโลยีเก่า ล้วนแล้วแต่สามารถใช้งาน巴士 ร่วมกันได้



ประเด็นการพิจารณาในการออกแบบบัส

ความกว้างของบัส

ความกว้าง (*width*) ของดาต้าบัส มีผลโดยตรงต่อประสิทธิภาพของระบบ บัสยิ่งมีความกว้างเท่าใด ก็จะช่วยในการถ่ายทอดข้อมูลได้มากขึ้นต่อหน่วยเวลา ความกว้างของช่องสัญญาณบอกตำแหน่งที่อยู่ จะเป็นตัวบอกปริมาณหน่วยความจำสูงสุดเท่าที่ระบบจะสามารถมีได้ เพราะความกว้างของสายสัญญาณบอกตำแหน่ง เป็นตัวบอกขอบเขตของตำแหน่งที่อยู่ที่ระบบนั้นจะสามารถอ้างอิงได้





THE END

