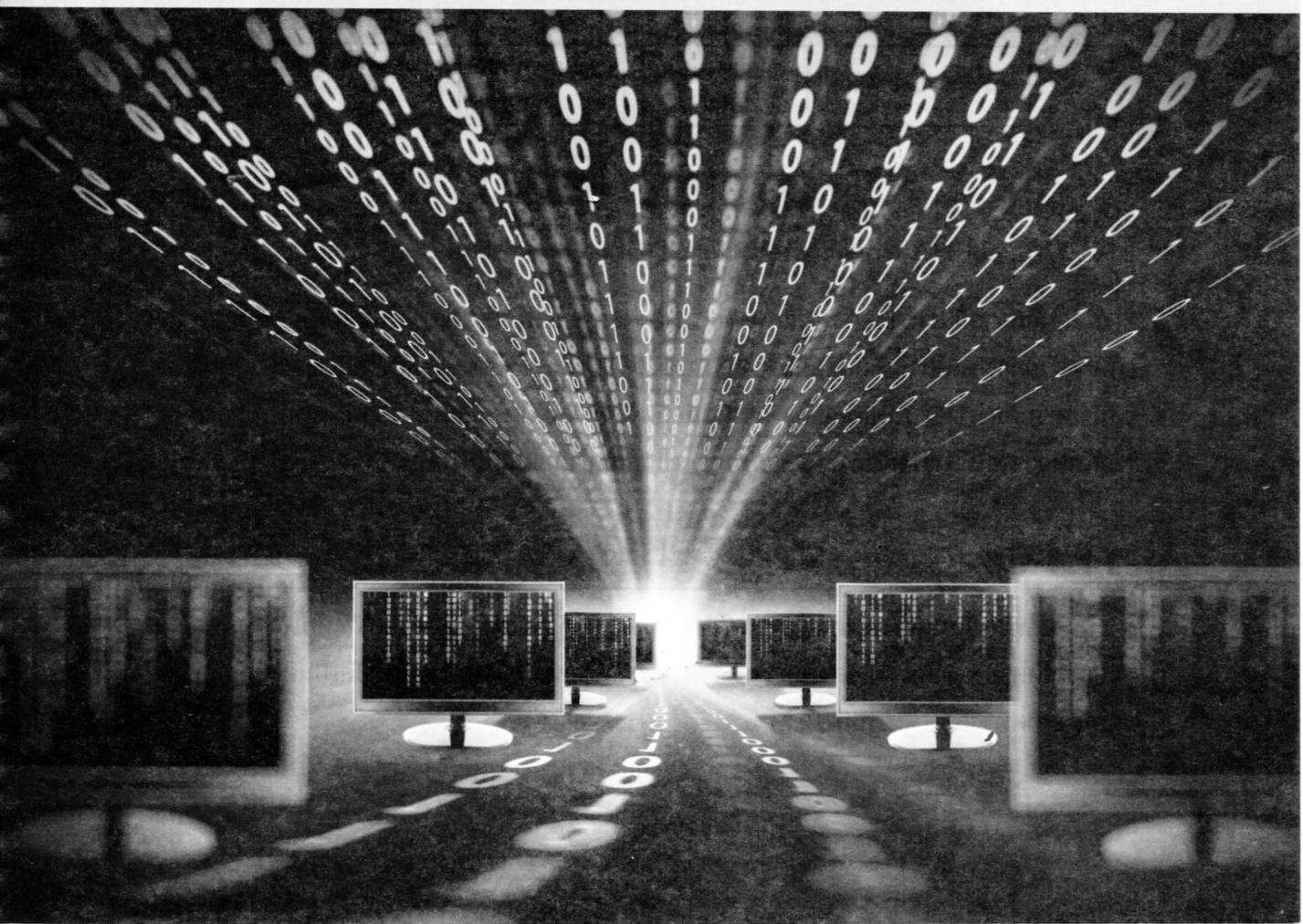


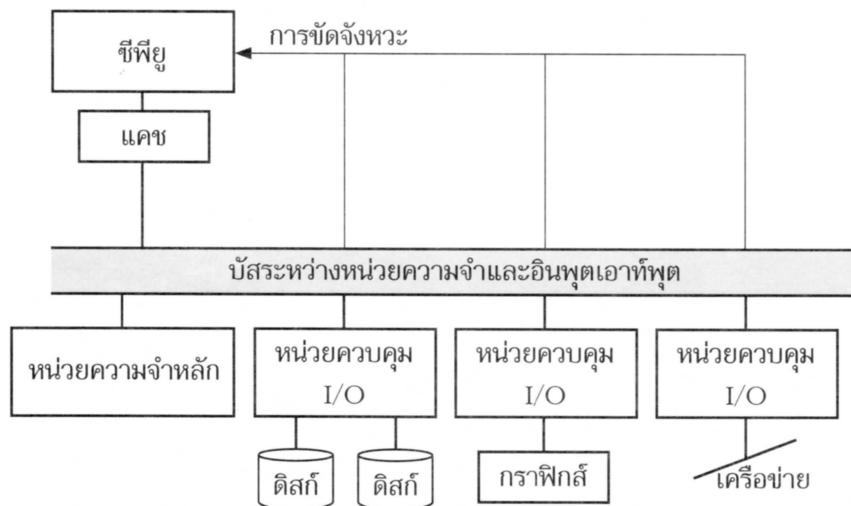
11

ระบบอันพุต/เอาร์พุต



นอกเหนือจากระบบหน่วยความจำแล้ว เครื่องคอมพิวเตอร์ยังต้องเข้ามาร่วมกับอุปกรณ์ภายนอกอื่นๆ อีก ในบทนี้จะกล่าวถึงระบบอินพุต/เอาต์พุต ซึ่งรวมไปถึงการทำงานของบัสรูปแบบต่างๆ

ระบบอินพุต/เอาต์พุตนั้นแบ่งเป็นขั้นตอนกับประสิทธิภาพของระบบโดยรวมทั้งทางตรงและทางอ้อม เช่น เวลาในการเข้าถึงข้อมูลและ Throughput ของการรับส่งข้อมูล รูปที่ 11.1 แสดงการเข้ามาร่วมกับหน่วยความจำ อุปกรณ์อินพุตและเอาต์พุต



รูปที่ 11.1 การเข้ามาร่วมกับอุปกรณ์อินพุต/เอาต์พุต (I/O) แบบต่างๆ

นอกจากนี้ ยาร์ดดิสก์ การแสดงผล เครื่อข่าย ฯลฯ ก็ยังจัดเป็นอุปกรณ์อินพุต/เอาต์พุตเช่นกัน หน่วยต่างๆ เหล่านี้เข้ามาร่วมกันด้วยบัส อุปกรณ์อินพุต/เอาต์พุตหลายๆ ชิ้นอาจจะควบคุมด้วย หน่วยควบคุม (I/O Controller) การติดต่อกับชีพีซีของอุปกรณ์อินพุต/เอาต์พุตเหล่านี้ต้องผ่านบัส แต่อาจจะใช้การขัดจังหวะ (Interrupt) เป็นสัญญาณที่ส่งไปยังชีพีซี

■ 11.1 บัส

สำหรับบัสนั้นเป็นสายสัญญาณที่ต้องมีการแบ่งกันใช้งาน (Shared) โดยใช้เข้ามาร่วมกับอุปกรณ์หลายๆ อย่างที่มีเวลาในการทำงาน และอัตราการรับส่งที่ต่างกัน บัสเป็นสายสัญญาณแบบ 2 ทิศทางคือ (ดูในรูปที่ 10.2)

- สายสัญญาณควบคุม (Control Lines)** สำหรับการส่งสัญญาณร้องขอ (Request) และสัญญาณการตอบรับ (Acknowledge) จะเป็นการบอกประเภทของข้อมูลที่ส่ง โดยผู้ส่ง (Master) จะเป็นผู้เริ่มต้นร้องขอข้อมูลเสมอ
- สายสัญญาณข้อมูล (Data Lines)** ใช้ส่งข้อมูล ส่งแอดเดรส (Address) ของข้อมูลที่ส่งไป และส่งคำสั่งที่ขับข้อนไปยังหน่วยต่างๆ ที่ต้องการ



รูปที่ 11.2 การเชื่อมต่อบัส

ในการรับ-ส่งข้อมูลผ่านบัสต้องทำเป็นขั้นตอนหรือ Transaction โดยประกอบด้วยขั้นตอนระหว่างผู้ส่ง (Master) และผู้รับ (Slave) ดังนี้

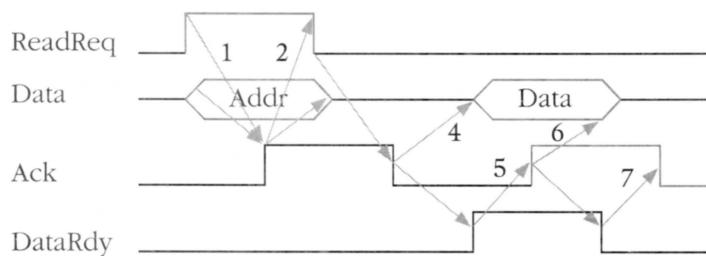
- ผู้ส่ง (Master) ทำการส่งคำสั่งและแอดเดรสของข้อมูลที่จะส่ง เรียกว่าเริ่มการร้องขอ
- ผู้รับ (Slave) ทำการรับ (หรือส่ง) ข้อมูล เรียกว่าการกระทำ (Action)

โดยทั่วไปแล้ว การออกแบบบัสต้องคำนึงถึงการใช้งานร่วมกัน ความยาวของบัส จำนวนอุปกรณ์ รวมทั้งข้อดี-ข้อเสียของวงจรที่ต้องใช้ รวมถึงการต้องการให้เวลาทำงานเร็วที่สุด การรองรับอุปกรณ์หลายประเภท และราคาที่ถูก

ประเภทของบัสมีหลายประเภท ได้แก่ บัสที่ใช้มาระหว่างชิปปิ้งและหน่วยความจำ บัสแบบนี้ต้องการการเข้าถึงข้อมูลที่เร็วและมักใช้กับแอดเดรส Backplane เป็นบัสที่ใช้มอยู่ตรงกลางระหว่างบัสอินพุต/เอาต์พุต (I/O Bus) และบัสระหว่างโปรเซสเซอร์และหน่วยความจำ (ซึ่งมักต้องการให้ส่งข้อมูลได้เร็วและต้องใช้ตามมาตรฐานที่มีอยู่ เช่น PCI, ATA) สำหรับหน่วยอินพุต/เอาต์พุตนั้นก็ต้องการบัสที่ยาว เขื่อมต่อกับอุปกรณ์ได้หลากหลายและตามมาตรฐาน เช่น SCSI, USB, Firewire เป็นต้น

นอกจากนี้ยังมีบสประเภท Synchronous และแบบ Asynchronous ถ้าเป็นแบบ Synchronous ก็จะใช้สัญญาณนาฬิกาและใช้โปรโตคอลแบบ Synchronous โดยอุปกรณ์ต่างๆ ที่เข้มต่ออยู่ด้วยกันต้องทำงานในอัตราที่เท่ากัน และเนื่องจากมีการใช้สัญญาณนาฬิกา ดังนั้น จะต้องให้บสที่ลับเพื่อหลีกเลี่ยง Clock Skew ซึ่งเป็นความคลาดเคลื่อนที่เกิดจากสัญญาณนาฬิกา ถ้าเป็นรูปแบบ Asynchronous จะไม่ใช้สัญญาณนาฬิกาและใช้การโต้ตอบที่เรียกว่ามีการทำ Handshaking ตามขั้นตอนดังต่อไปนี้

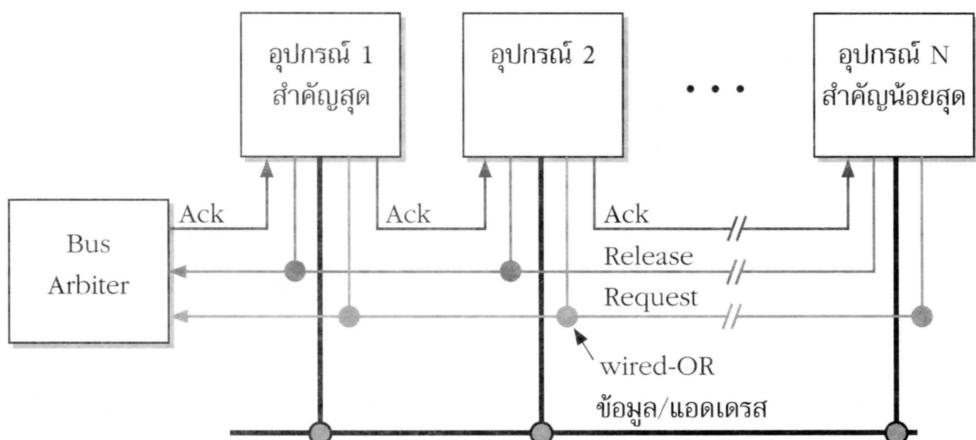
1. เริ่มต้นที่พิจารณาสัญญาณ ReadReq และส่งแอดเดรสก่อน เมื่อหน่วยความจำได้สัญญาณ ReadReq สำหรับการอ่าน จะทำการอ่านแอดเดรสและต้องตอบสนองด้วยสัญญาณ Ack
2. เมื่ออุปกรณ์อินพุต/เอาต์พุตเห็นสัญญาณ Ack จะทำการปล่อยสัญญาณ ReadReq
3. เมื่อหน่วยความจำได้ข้อมูลที่ต้องการ จะทำการส่งสัญญาณ DataRdy เมื่อหน่วยความจำได้ข้อมูลที่ต้องการ
4. ส่งข้อมูลที่ต้องการและส่งสัญญาณว่าข้อมูลนั้นพร้อมอ่าน จนนั้นอุปกรณ์อินพุต/เอาต์พุต จะสามารถนำข้อมูลไปใช้ได้ และอุปกรณ์จะอ่านข้อมูลเมื่อสัญญาณ Data เกิดขึ้นและส่งสัญญาณ Ack ตอบกลับ
5. เมื่อหน่วยความจำเห็นสัญญาณ Ack จะลดสัญญาณ DataRdy เป็น 0
6. เมื่ออุปกรณ์อินพุต/เอาต์พุตเห็นสัญญาณ DataRdy ลดลง ก็จะลดสัญญาณ Ack เช่นกัน ลำดับการทำงานข้างต้นเป็นดังตัวเลขที่ระบุในรูปที่ 11.3 การใช้บัสจะต้องมีการจัดลำดับการเข้าใช้ ในการนี้ที่มีหลายอุปกรณ์ทำการร้องขอใช้บัสเข้ามาพร้อมกัน การจัดลำดับเรียกว่าการทำ Bus Arbitration ซึ่งมีรูปแบบ Bus Arbitration หลายแบบ Bus Arbitration เป็นการควบคุมการใช้บัสสำหรับกรณีมีหลายอุปกรณ์ จะมีการใช้หน่วยควบคุมที่พิชัยหรือหน่วยควบคุม DMA และอาจจะมีการอนุญาตให้เข้าใช้ได้ทีละอุปกรณ์ในแต่ละครั้ง วิธีการควบคุมอาจจะเป็นแบบศูนย์กลางหรือแบบกระจายก็ได้ วิธีการควบคุมนี้ต้องคำนึงถึงความยุติธรรมและลำดับความสำคัญของแต่ละอุปกรณ์ การควบคุมแบบศูนย์กลางจะใช้ชาร์ดแวร์ตัวเดียวในการควบคุมการเข้าถึงบัส โดยจะต้องใช้หน่วยควบคุมบัสและ Arbiter ซึ่งชาร์ดแวร์อาจจะเป็นส่วนหนึ่งของชิปปี้ หรือแยกออกจากชิปปี้ ก็ได้ สำหรับการควบคุมแบบกระจายศูนย์นั้น แต่ละอุปกรณ์อาจจะต้องการการใช้บัสพร้อมกัน และมีวงจรลอกิจใช้ในการควบคุมการใช้บัสสำหรับทุกอุปกรณ์ที่เชื่อมกัน



รูปที่ 11.3 ตัวอย่างการให้สัญญาณแบบ Asynchronous

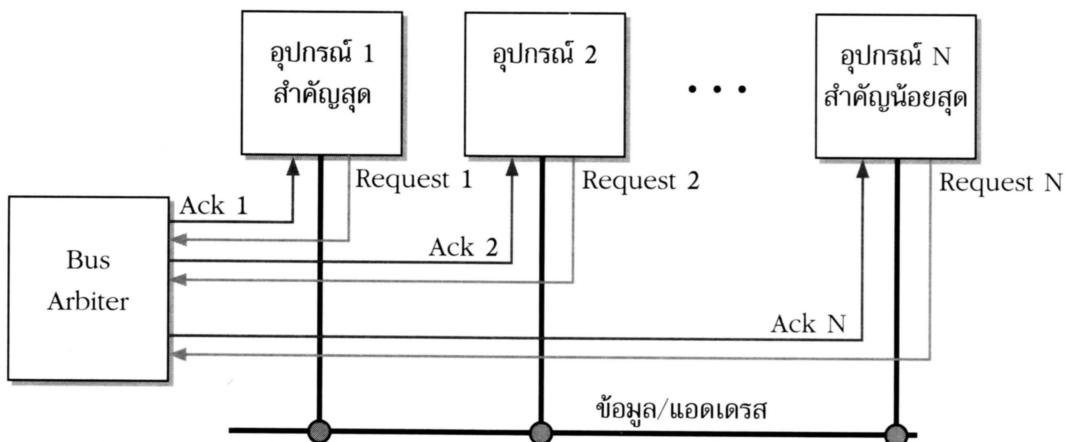
ตัวอย่างเช่น

1. Daisy Chain Arbitration เป็นรูปแบบที่ง่ายแต่ไม่ค่อยยุติธรรม อุปกรณ์ที่มีความสำคัญสูงนั้นอยู่ติดกับ Bus Arbiter มากที่สุด จะมีลิทีชีดใหญ่ที่สุด และอุปกรณ์ที่มีลำดับความสำคัญต่ำอาจรอเข้าใช้โดยไม่มีกำหนดเวลา ดังแสดงในรูปที่ 11.4



รูปที่ 11.4 การใช้ Daisy Chain

2. Centralized Arbitration (ต้องการผู้ไก่เลี้ยหรือ Arbiter) ดังแสดงในรูปที่ 11.5 วิธีนี้ง่ายและยุติธรรม แต่ละอุปกรณ์จะต้องส่งสัญญาณร้องขอของตนไปยัง Arbiter และ Arbiter ก็จะตอบกลับด้วยการตอบรับ (Ack) วิธีนี้ต้องการ Arbiter ที่ซับซ้อนในการไก่เลี้ยการใช้ในบัสระหว่างซีพียูและหน่วยความจำ เป็นวิธีการที่ใช้ในบัสแบบ PCI

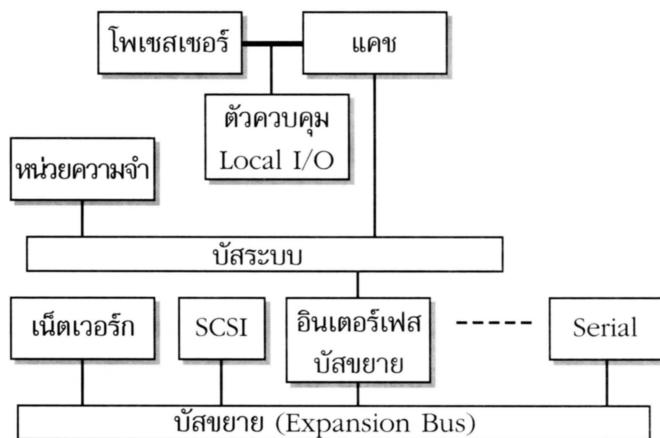


รูปที่ 11.5 การใช้ Centralized Bus

3. **Self** ในวิธีนี้แต่ละอุปกรณ์ที่ต้องการใช้บัสจะส่งรหัสออกมานะเพื่อบอกว่าตนต้องการใช้บัสได้แก่ สัญญาณ NuBus ที่ใช้ในเครื่องแมคอินโทห์
4. **Collision Detection** เป็นวิธีการตรวจสอบการชนกันของการใช้บัส ใช้ใน Ethernet CDMA ดังจะได้อธิบายเพิ่มเติมต่อไป

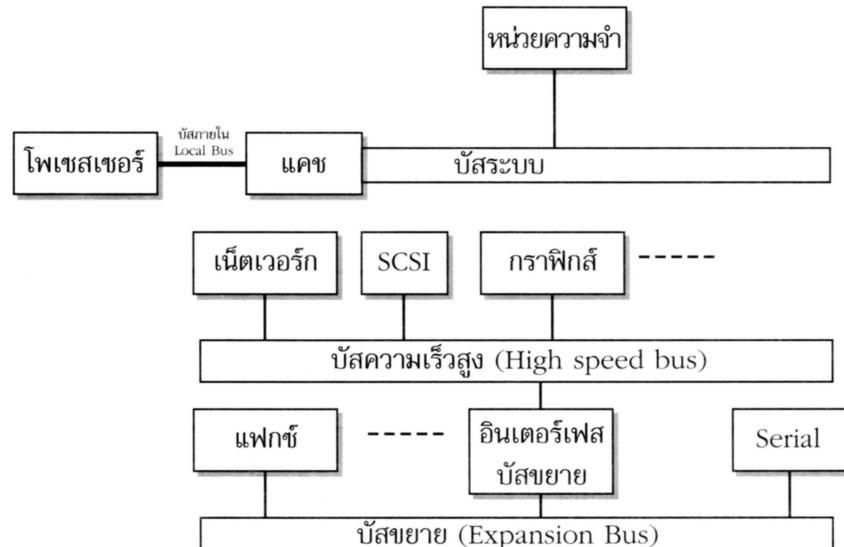
สำหรับการใช้บัสเดียว (Single Bus) นั้นยังมีปัญหาเมื่อใช้กับหลายๆ อุปกรณ์ เช่น มีการหน่วงเวลามากขึ้น (Delay) เมื่อมีบัสที่ยว การประสานงานการใช้บัสจะทำได้ช้าลง และอาจจะทำให้อัตราการส่งข้อมูลเกินกำลังของบัสที่จะรองรับได้ ดังนั้น ระบบส่วนใหญ่ในปัจจุบันจะใช้บัสหลายชุด

ในการใช้หลายบัสนั้นอาจจะทำการเชื่อมต่อเป็นลำดับชั้นดังแสดงในรูปที่ 11.6 โดยมีบัสภายใน (Local Bus) เชื่อมต่อระหว่างชิปปิญและแครช มีบัสระบบ (System Bus) เชื่อมต่อระหว่างชิปปิญและหน่วยความจำหลัก และมีบัสอื่นๆ เชื่อมต่อกับอุปกรณ์ภายนอกที่เรียกว่าบัสขยาย



รูปที่ 11.6 โครงสร้างระบบแบบหลายบัส

ในรูปที่ 11.7 แสดงการเชื่อมต่ออีกรูปแบบที่ใช้บัสระบบเชื่อมแครชกับหน่วยความจำหลักเข้าด้วยกัน ใช้บัสแบบเร็วเชื่อมกับอุปกรณ์ที่ต้องการการเข้าถึงแบบความเร็วสูง และมีบัสส่วนขยาย (Expansion Bus) เชื่อมต่อกับภายนอก



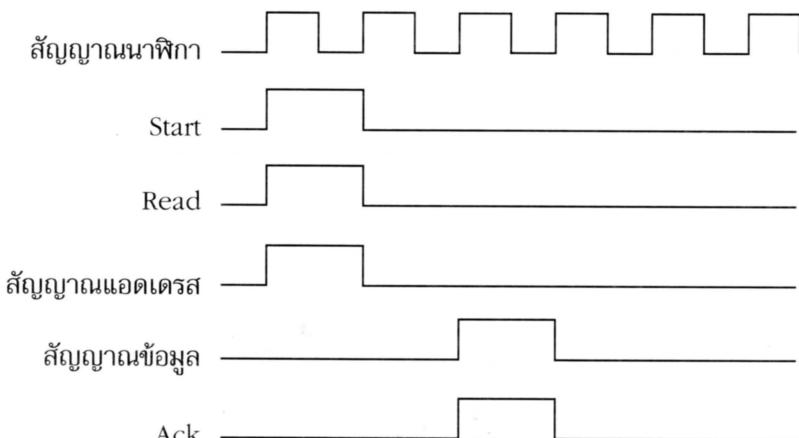
รูปที่ 11.7 ตัวอย่างระบบบัสแบบที่ 2

รูปแบบการทำงานของบัสมีหลายประเภท ได้แก่

1. Dedicated เป็นการแยกระหว่างบัสสำหรับส่งข้อมูลและบัสส่งแอดเดรส
2. Multiplexed เป็นการใช้ร่วมกัน โดยส่งได้ทั้งข้อมูลและแอดเดรส รูปแบบนี้จะได้เปรียบ เพราะใช้จำนวนบัสน้อยกว่า แต่ต้องการการควบคุมแบบสลับการใช้งาน แต่จะให้ประสิทธิภาพการใช้งานสูงสุดได้

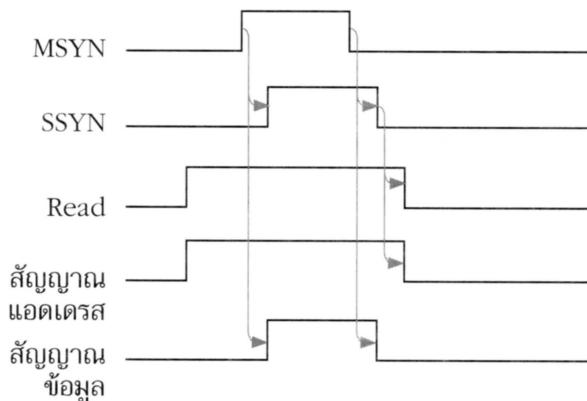
ในการจัดการด้านเวลาหนึ่งจำเป็นต้องประสานงานเมื่อเกิดเหตุการณ์ต่างๆ ระหว่างการใช้งานบัส โดยการจัดการอาจจะเป็นแบบ Synchronous หรือแบบ Asynchronous ในแบบ Synchronous นั้นมีการอ่านการเกิดเหตุการณ์ โดยจะต้องอ่านตามลำญาณนาฬิกา สัญญาณควบคุมต้องมีสายสัญญาณนาฬิกาด้วย มีการทำงานใน 1 ไซเคิล (Single Cycle) อุปกรณ์จะต้องสามารถอ่านสัญญาณนาฬิกาได้ ต้องทำงานพร้อมกันที่ขอบสัญญาณนาฬิกาที่ขาเข้า และใน 1 ไซเคิลอ่านได้เพียง 1 เหตุการณ์

ในรูปที่ 11.8 แสดงการใช้สัญญาณแบบ Synchronous จะมีสัญญาณ ได้แก่ สัญญาณนาฬิกา สัญญาณ Start และสัญญาณ Read จะทำงานตามสัญญาณนาฬิกาที่ขอบขาเข้า จากนั้นจะมีการส่งแอดเดรสที่สัญญาณแอดเดรส และจากนั้นจะได้ข้อมูลที่สัญญาณข้อมูล และมีการตอบกลับด้วยสัญญาณ Ack ซึ่งก็จะให้สัญญาณตามขอบขาเข้าทั้งสิ้น



รูปที่ 11.8 การใช้สัญญาณแบบ Synchronous

รูปที่ 11.9 นี้จะให้สัญญาณแบบ Asynchronous ซึ่งไม่ใช้สัญญาณนาฬิกามาเกี่ยวข้อง เมื่อ มีสัญญาณต้องการการอ่าน (Read) ส่งเข้ามาพร้อมกับแอดเดรสของข้อมูลที่สัญญาณแอดเดรส จะเกิดสัญญาณ MSYN และ SSYN จากนั้นก็จะเกิดการส่งข้อมูลที่สัญญาณข้อมูล



รูปที่ 11.9 การให้สัญญาณแบบ Asynchronous

รูปแบบการติดต่อระหว่างอินพุต/เอาต์พุตกับชีพียูทำได้ด้วยวิธีต่างๆ ได้แก่

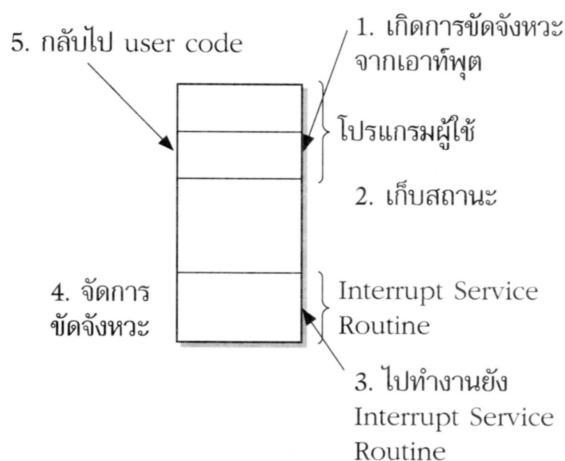
- การใช้คำสั่งพิเศษ (Special I/O Instructions) โดยจะต้องระบุคำสั่งทั้งของอุปกรณ์ และของคำสั่ง
- Memory-Mapped อินพุต/เอาต์พุต วิธีนี้แต่ละอุปกรณ์อินพุต/เอาต์พุตจะถูกกำหนดให้ไปอยู่ที่แอดเดรสต่างกัน ในการเข้าถึงอุปกรณ์เหล่านี้ โปรแกรมจะไปอ่านหรือเขียนค่าในแอดเดรสที่ระบุถึงอุปกรณ์นั้น ซึ่งจะเป็นการอ่านหรือเขียนไปยังอุปกรณ์นั้นเอง

■ 11.2 การสื่อสารระหว่างอุปกรณ์อินพุต/เอาต์พุตกับชีพียู

วิธีการที่อุปกรณ์อินพุต/เอาต์พุตจะสื่อสารกับชีพียู ได้แก่ วิธีการ Polling การขัดจังหวะ (Interrupt Polling) และวิธีการใช้ DMA

การ Polling เป็นวิธีที่จะทำการถาม (Poll) แต่ละอุปกรณ์ว่าต้องการใช้บัสอินพุต/เอาต์พุต หรือไม่ โดยชีพียูจะควบคุมหน้าที่นี้ ซึ่งวิธีนี้จะทำให้ชีพียูเสียเวลาทำการถามแต่ละอุปกรณ์ ทำให้อุปกรณ์ที่ร้องขอจะต้องรอตามลำดับการ Poll ดังนั้น การใช้การขัดจังหวะควบคู่กันจะมีประสิทธิภาพมากกว่า

ในวิธีการขัดจังหวะจะใช้สัญญาณ 1 เส้นต่อ 1 อุปกรณ์ และอาจจะใช้ร่วมกับการวิธี Polling เพื่อสอบถามแต่ละอุปกรณ์ว่ามีการขัดจังหวะเกิดขึ้นหรือไม่ ถ้ามี ก็จะต้องมีการตอบรับหรือ Acknowledge ในกรณีการจัดการการขัดจังหวะนั้น จะต้องมีตารางเก็บตำแหน่งของโปรแกรมอยู่ที่จัดการการขัดจังหวะที่เรียกว่าอินเทอร์รัปต์เวกเตอร์ (Interrupt Vector) เมื่อเกิดการขัดจังหวะ จะต้องมีการระบุประเภทของการขัดจังหวะ ซึ่งค่านี้จะถูกส่งไปในบัสผ่านไปให้ชีพิญ เพื่อให้ชีพิญใช้ระบุโปรแกรมอยู่ที่จะใช้จัดการการขัดจังหวะ หรือที่เรียกว่า Handler Routine เพื่อจัดการกับการขัดจังหวะนั้น แต่ในการเกิดการขัดจังหวะนั้น อุปกรณ์นั้นๆ จะต้องร้องขอการใช้บัสก่อนจะเกิดการขัดจังหวะ ตัวอย่างบัสที่ใช้วิธีนี้ ได้แก่ PCI, SCSI บัส เป็นต้น



รูปที่ 11.10 การจัดการการเกิดการขัดจังหวะ

ในรูปที่ 11.10 แสดงการเกิดการขัดจังหวะในรูปแบบตั้งแต่มีการขัดจังหวะชีพิญมาจากหน่วยเอาต์พุต จนกลับไปยังโปรแกรมผู้ใช้งาน

การขัดจังหวะของอินพุต/เอาต์พุตนั้นเป็นการทำงานแบบ Asynchronous เนื่องจากการทำงานไม่ได้สัมพันธ์กับคำสั่งใดๆ หรือสัญญาณนาฬิกา จึงไม่ได้ขัดขวางการทำงานของคำสั่งอื่นๆ ให้เลื่อนด้วย

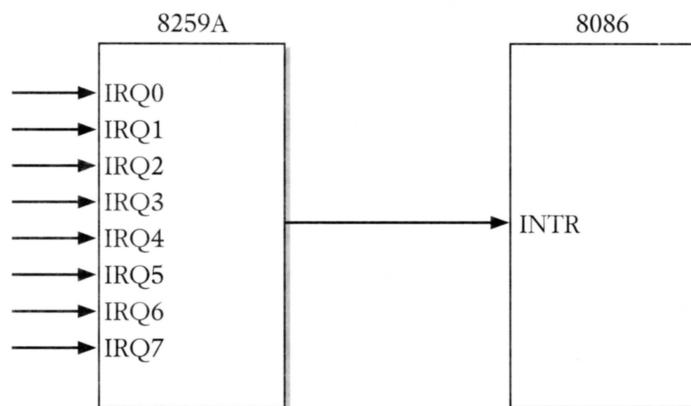
ประโยชน์ของการใช้การขัดจังหวะนั้นจะเป็นการลดเวลาการสอบถามหรือการ Poll ของชีพิญ และทำให้ประยุตเวลาการทำงานของชีพิญนั้น โปรแกรมผู้ใช้จะถูกหยุดชั่วคราวระหว่างการ

โอนย้ายการทำงานไปยัง Interrupt Routine และต้องมีการจัดเก็บสถานะของชีพิญก่อนจะไปทำงาน Interrupt Routine ตามที่ระบุไว้ใน Interrupt Vector นั้น

กรณีมีการขัดจังหวะหลายๆ อันเกิดขึ้นพร้อมกัน แต่ละสายของการขัดจังหวะจะมีลำดับความสำคัญต่างกัน สายที่มีความสำคัญมากจะสามารถขัดจังหวะสายที่สำคัญน้อยกว่าได้ และอุปกรณ์ที่เป็นเจ้าของบัสตอนนั้นเท่านั้นจะสามารถทำการขัดจังหวะได้

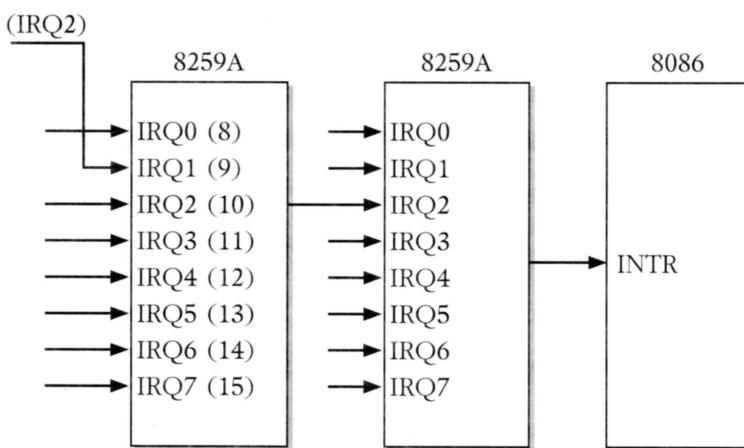
ตัวอย่างของ PC Bus ใน 8086 ใช้ 8259A Interrupt Controller ตัวนี้มีสายการขัดจังหวะจำนวน 8 เส้น ในการจัดการการขัดจังหวะจะมีลำดับการทำงานดังนี้

เริ่มต้น 8259A จะยอมรับการขัดจังหวะที่เกิดขึ้น และจะพิจารณาลำดับความสำคัญและจะส่งสัญญาณไปยังชีพิญ 8086 (ผ่านสาย INTR) ชีพิญจะตอบรับโดยการให้สัญญาณ Ack กลับมา จากนั้น 8259A จะส่งตำแหน่งของ Interrupt Vector ไปยังบัสและให้ชีพิญดำเนินการจัดการการขัดจังหวะนั้น ตัวอย่างการเชื่อมต่อแสดงดังรูปที่ 11.11



รูปที่ 11.11 การเชื่อมต่อ 8259A กับ 8086

สำหรับบัสแบบ ISA นั้นจะใช้ 8259A จำนวน 2 ตัวเชื่อมกันด้วยสายการขัดจังหวะ ทำให้มีสายการขัดจังหวะทั้งหมด 15 เส้น โดยใช้ IRQ9 สำหรับการส่งข้อมูลอีกครั้ง martyr IRQ2 รูปที่ 11.12 แสดงการเชื่อมต่อแบบนี้



รูปที่ 11.12 การเชื่อมต่อ 8259A จำนวน 2 ตัวใน ISA

สำหรับการใช้การขัดจังหวะและแบบ Program I/O นั้นจะเป็นการลดประสิทธิภาพการทำงานของชีพียู และอาจจะทำให้อัตราการรับ-ส่งข้อมูลลดลง ชีพียูต้องรอการจัดการการขัดจังหวะให้เสร็จลินก์ก่อน ดังนั้น การใช้หน่วยควบคุม DMA (Direct Memory Access) จะทำให้ชีพียูไม่ต้องรอให้การจัดการการขัดจังหวะเสร็จลินก์ก่อน และชีพียูสามารถไปทำอย่างอื่นได้ระหว่างนั้น

ในการทำงานเมื่อใช้ DMA ช่วงนั้น เริ่มต้นชีพียูจะส่งสัญญาณมาบอกหน่วยควบคุม DMA ว่าต้องการเขียนหรืออ่านข้อมูลพร้อมแจ้งตำแหน่งข้อมูลที่ต้องการ จำนวนข้อมูลทั้งหมดที่ต้องการจากนั้นชีพียูจะไปทำงานอื่นก่อน และหน่วยควบคุม DMA จะจัดการเรื่องการโอนย้ายข้อมูลเอง เมื่อได้รับข้อมูลครบแล้วจะทำการส่งการขัดจังหวะไปบอกยังชีพียู

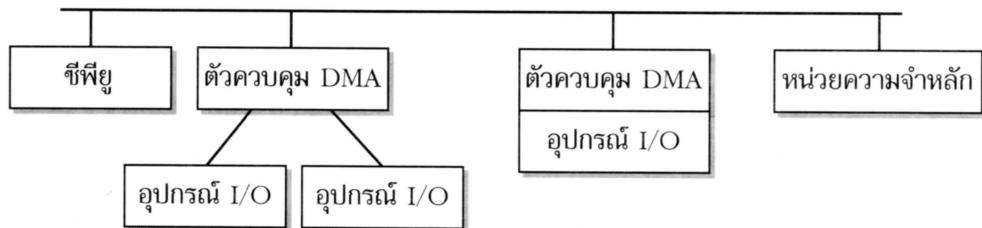
หน่วยควบคุม DMA จะใช้บัสในการโอนย้ายข้อมูล วิธีนี้ชีพียูจะหยุดชั่วคราวก่อนจะทำการเข้าถึงบัส แต่จะหยุดไม่นานเท่ากับการให้ชีพียูทำการโอนข้อมูลเอง

ตัวอย่างการเชื่อมต่อ DMA แสดงดังรูปที่ 11.13



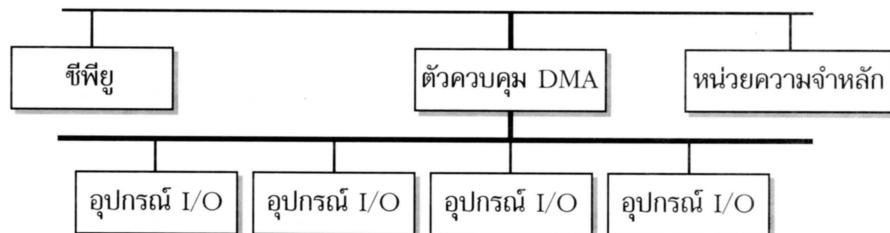
รูปที่ 11.13 การเชื่อมต่อ DMA แบบบัสเดียว

ในรูปที่ 11.13 เป็นการเชื่อมต่อแบบบัสเดียว แต่ละครั้งของการโอนข้อมูลจะต้องใช้บัส 2 ครั้ง ได้แก่ จากอินพุต/เอาต์พุตไปยัง DMA และจาก DMA ไปยังหน่วยความจำ นั่นคือชีพียุจะหยุดการทำงานชั่วคราว 2 ครั้ง



รูปที่ 11.14 การเชื่อมต่อ DMA แบบที่ 2

ในรูปที่ 11.14 เป็นการใช้บัสเดียวเข่นกัน แต่ตัวหน่วยความคุมรองรับหลายๆ อุปกรณ์ได้ แต่ละครั้งของการโอนข้อมูลต้องมีการใช้บัส 1 ครั้ง คือ จาก DMA ไปยังหน่วยความจำ ดังนั้น ชีพียุจะหยุดชั่วคราว 1 ครั้ง



รูปที่ 11.15 การเชื่อมต่อ DMA แบบที่ 3

ในรูปที่ 11.15 จะใช้บัสอินพุต/เอาต์พุตแยกกัน แต่ละบัสจะรองรับอุปกรณ์ที่ใช้ DMA ได้ แต่ละครั้งของการโอนจะใช้บัสครั้งเดียว คือ จาก DMA ไปหน่วยความจำ ดังนั้น ชีพียุจะหยุดชั่วคราว 1 ครั้งเข่นกัน

ในปัจจุบันการใช้อินพุต/เอาต์พุตมีความซับซ้อนมากขึ้น เนื่อง ภาพิกการ์ดแบบ 3D ทำให้ชีพียุสามารถสั่งหน่วยความคุม I/O ให้ทำการโอนข้อมูล และหน่วยความคุม I/O นั้นจะทำการโอนข้อมูลทั้งหมด ซึ่งเมื่อมีการโอนข้อมูลมากๆ อาจจะทำงานได้ช้า จึงทำให้มีการนำ Co-Processor มาใช้แทนรูปแบบของหน่วยความคุม I/O แทน เพื่อให้ทำงานได้เร็วขึ้น

ตัวอย่างบัสแบบต่างๆ ได้แก่

- บัส PCI (Peripheral Component Interconnection)** คิดคันโดยบริษัท Intel มีขนาด 32 บิต หรือ 64 บิต จำนวน 50 เส้น สำหรับสัญญาณระบบ (System Line) จะเป็นสัญญาณที่รวมสัญญาณนาฬิกาและสัญญาณ Reset เข้าด้วยกันสำหรับสัญญาณแอดเดรสและสัญญาณข้อมูลมีขนาด 32 เส้น จะใช้กับ MUX และมีสาย Interrupt และสาย Validate มีตัวควบคุมอินเตอร์เฟช และวิธีการไกล์เกลี่ยนั้นใช้ PCI Bus Arbiter โดยตรง และมีสัญญาณ Error รองรับ多彩และสามารถขยายได้ถึง 64 บิต

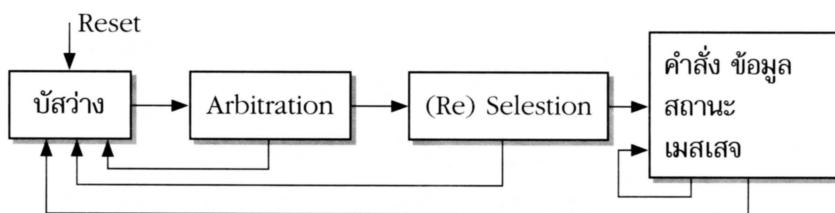
ในการทำงานเริ่มต้น Master จะร้องขอการใช้บัส จากนั้นจะพิจารณาประเภทของ การร้องขอว่าเป็นการอ่านหรือเขียน จากนั้นก็จะส่งตำแหน่งตามด้วยคำตัวของข้อมูลต่อไป

- SCSI (Small Computer Systems Interface)** เป็นรูปแบบการเชื่อมต่อแบบขนาน มีสายสัญญาณขนาด 8,16,32 บิตก็ได้ ใช้วิธี Daisy Chain ในการเชื่อมต่ออุปกรณ์แต่ละตัว ทำให้อุปกรณ์แต่ละตัวสามารถสื่อสารระหว่างกันได้ด้วย

SCSI-I เกิดขึ้นในปี ค.ศ. 1980 มีขนาด 8 บิต และทำงานที่ความถี่ 5 เมกะเฮิรตซ์ มีอัตราการส่งข้อมูลเท่ากับ 5 เมกะไบต์ต่อวินาที ใช้ได้ถึง 7 อุปกรณ์พร้อมกัน

SCSI-2 เกิดขึ้นในปี ค.ศ. 1991 มีขนาด 16 บิต และ 32 บิต ทำงานที่ความถี่ 10 เมกะเฮิรตซ์ มีอัตราการส่งข้อมูลที่ 20 หรือ 40 เมกะไบต์ต่อวินาที

การให้สัญญาณสำหรับบัส SCSI นั้นจะทำระหว่างคอมพิวเตอร์กับอุปกรณ์ภายนอก โดยจะตรวจสอบว่าบัสว่างหรือไม่ จะใช้วิธีการไกล์เกลี่ยเข่นเดียวกับแบบ PCI ทำการเลือกผู้รับและมีหน่วย Reselection ใช้ในกรณีที่มีการหยุดการทำงานเพื่อซ่อมต่อช่วงเวลา และทำให้สามารถเชื่อมต่อใหม่ได้ มีสถานะสำหรับการทำงานดังแสดงในรูปที่ 11.16 รูปที่ 11.16 แสดงตัวอย่างสถานะบัสว่าง จากนั้นเมื่อความต้องการใช้ ก็จะเข้าไปขั้นตอน Arbitration จากนั้นจะเลือกผู้รับและทำการส่งคำสั่ง ข้อมูล สถานะ หรือแมสเซจต่างๆ จนกว่าจะเสร็จ และกลับอยู่ในสถานะว่างอีกครั้ง

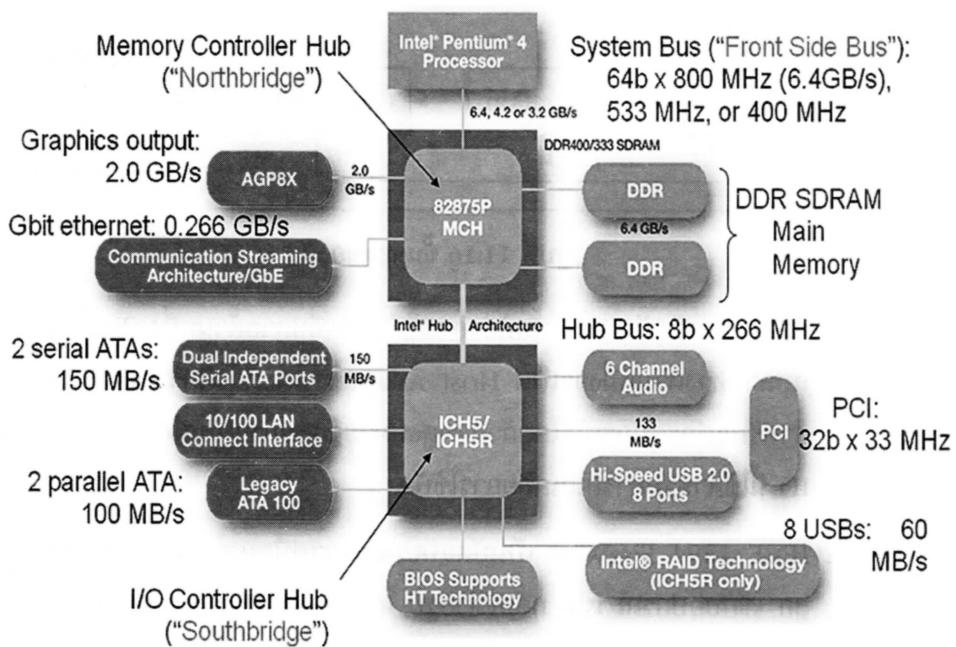


รูปที่ 11.16 ขั้นตอนของบัส SCSI

สำหรับบัส SCSI นั้นต้องการตัวปิดบัสที่ปลายด้านหนึ่ง (Bus Terminator) และที่ปลายทางด้านหนึ่งจะเป็น Host Adapter และต้องมีการเซ็ต SCSI ID ด้วยสวิตซ์สำหรับแต่ละอุปกรณ์ โดยถ้ามีค่าสูงหมายถึงอุปกรณ์นั้นมีลำดับความสำคัญสูง SCSI ID นี้ใช้สำหรับการไกล์เกลี่ยการใช้บัส

- IEEE 1394 FireWire เป็นบัสแบบอนุกรมแต่มีสมรรถนะสูง รับ-ส่งได้เร็ว และราคาถูก ใช้กับอุปกรณ์ เช่น กล้องถ่ายรูป เครื่องเล่นวิดีโอ โทรทัศน์ ใช้รูปแบบในการเชื่อมต่อ Daisy Chain สามารถรองรับมากสุดได้ถึง 63 อุปกรณ์ ใน 1 พอร์ตเชื่อมต่อได้ถึง 1022 บัส เมื่อใช้บริจ์สามารถปรับแต่งได้อย่างอัตโนมัติ และไม่ต้องใช้ Bus Terminator อาจจะใช้รูปแบบตันไม้ในการเชื่อมต่อ ก็ได้ จากการเปรียบเทียบความสามารถของบัสแบบ Firewire กับแบบ USB พบว่าโดยมาก FireWire จะมีประสิทธิภาพสูงกว่า โดยมีแบบเดิมที่เร็วกว่าโดยรวม มีการทำงานแบบ Asynchronous ขณะที่ USB ยังทำงานแบบ Synchronous ส่วนจำนวนอุปกรณ์ที่ใช้ได้สูงสุดยังน้อยกว่าของ USB

รูปที่ 11.17 แสดงตัวอย่างบัสของ Pentium 4 ซึ่งมีบัสหลายแบบ ได้แก่ ATA แบบขนาด ATA แบบอนุกรม AGP สำหรับกราฟิก DDR สำหรับหน่วยความจำ PCI USB Ethernet เป็นต้น เพื่อความสะดวกในการเชื่อมต่อกับอุปกรณ์หลายๆ แบบ



รูปที่ 11.17 ตัวอย่างบัสของ Pentium 4 (ที่มา: <http://www.intel.com>)

■ 11.3 สรุป

อินพุต/เอาต์พุตเป็นส่วนประกอบที่สำคัญของคอมพิวเตอร์ในการเชื่อมต่อกับโลกภายนอก ในบทนี้ได้กล่าวถึงพื้นฐานแนวคิดเกี่ยวกับการจัดการอินพุต/เอาต์พุต บัญชีแบบต่างๆ การเชื่อมต่อรูปแบบต่างๆ แนวคิดที่เกี่ยวข้องกับระบบปฏิบัติการ ได้แก่ DMA และการจัดการอินเทอร์รัปต์

คำถ้ามก้ายบท

1. จงยกตัวอย่างอุปกรณ์อินพุต/เอาต์พุตมาสัก 2 ชนิด
2. การใช้การขัดจังหวะเพื่อวัดถูประஸงค์อะไร
3. การทำ Interrupt Polling มีข้อเสียอย่างไร
4. การให้สัญญาณแบบ Synchronous และต่างจากแบบ Asynchronous อย่างไร
5. MTTF และ MTTR ต่างกันอย่างไร
6. ประสิทธิภาพของติสกรีบบุ้นได้ด้วยตัววัดอะไรบ้าง
7. DMA คืออะไร มีข้อดีอย่างไร
8. จงบอกข้อดี-ข้อเสียของการใช้บัสแบบ Daisy Chain และ Centralized Bus

