

## บทที่ 5

### ระบบอินพุตเอาต์พุต

หน้าที่หลักของไมโครโพรเซสเซอร์คือทำการประมวลผล ซึ่งข้อมูลที่ประมวลผลนั้นรับมาจากหน่วยความจำหรืออุปกรณ์อินพุต (Input device) เมื่อทำการประมวลผลเสร็จแล้วก็นำผลลัพธ์ที่ได้ไปแสดงออกที่อุปกรณ์เอาต์พุต (Output device) ตัวอย่างของอุปกรณ์อินพุตที่เห็นได้ชัดเจนที่สุดคือคีย์บอร์ดเมาส์ สแกนเนอร์ เป็นต้น ตัวอย่างของอุปกรณ์เอาต์พุต ได้แก่จอแสดงผล เครื่องพิมพ์ และอื่นๆ

สไลด์หน้า 4 กล่าวถึงรูปแบบการจัดการอินพุตของไมโครโพรเซสเซอร์ในมุมมองของชุดคำสั่ง โดยแบ่งออกเป็น 2 รูปแบบ ได้แก่ วิธีแรกเรียกว่า Isolated I/O เป็นรูปแบบที่ไมโครโพรเซสเซอร์มีคำสั่งที่ออกแบบมาเพื่อใช้ในการจัดการอุปกรณ์อินพุตเอาต์พุตโดยเฉพาะ ยกตัวอย่างเช่นในซีพียู Z80 จะมี คำสั่ง IN ใช้ในกรณีต้องการอ่านข้อมูลจากอุปกรณ์อินพุต ส่วนคำสั่ง OUT ใช้ในการเขียนข้อมูลไปให้กับอุปกรณ์เอาต์พุต นอกเหนือจากซีพียู z80 แล้ว ซีพียูสถาปัตยกรรม 80x86 ก็เป็นอีกตัวหนึ่งที่ใช้รูปแบบการอินพุตแบบ Isolated I/O วิธีการจัดการอุปกรณ์อินพุตเอาต์พุตแบบที่สอง เรียกว่าแบบ Memory-mapped I/O ซึ่งวิธีนี้ตัวไมโครโพรเซสเซอร์จะไม่มีคำสั่งที่อ้างถึงอุปกรณ์ I/O โดยเฉพาะ แต่จะสงวนตำแหน่งหน่วยความจำไว้บางตำแหน่งสำหรับอ้างถึงอุปกรณ์แทน ตัวอย่างซีพียูที่ใช้รูปแบบนี้ได้แก่ซีพียู MCS-51 และ 680x0 เป็นต้น

สไลด์หน้า 5 แสดงให้เป็นรูปแบบของการเชื่อมต่ออุปกรณ์ I/O เข้ากับซีพียู สืบเนื่องจากอุปกรณ์ I/O แต่ละตัวจะต้องมีตำแหน่งแอดเดรสของตัวเอง ไม่ว่าจะเป็นตำแหน่ง I/O address สำหรับซีพียูแบบ Isolated I/O หรือตำแหน่งหน่วยความจำสำหรับอุปกรณ์ I/O ในกรณีใช้สถาปัตยกรรมแบบ memory mapped I/O ก็ตาม ในการติดต่ออุปกรณ์ I/O ซีพียูจะส่งหมายเลขแอดเดรสของอุปกรณ์ I/O ตำแหน่งที่ซีพียูต้องการติดต่อออกสู่แอดเดรสบัส และส่งค่าที่ต้องการส่งให้กับอุปกรณ์ I/O ผ่านทางบัสข้อมูล อุปกรณ์ I/O ทุกตัวบนบัสจะมองเห็นค่า data นี้ได้ แต่จะมีอุปกรณ์ I/O ตัวที่มีค่าแอดเดรสเท่ากับค่าที่ซีพียูส่งมาเท่านั้นที่จะรับเอาข้อมูลนี้ไปใช้งาน ดังนั้นในแต่ละอุปกรณ์ I/O จะต้องมียังวงจรถอดรหัสแอดเดรส (Address decoder) และเนื่องจากการติดต่ออุปกรณ์อินพุตเอาต์พุตซีพียูจะต้องส่งค่าควบคุมเข้ามาบอกอุปกรณ์ I/O ด้วย ในกรณีนี้เราจะใช้บัสควบคุมในการส่งค่าดังกล่าว อุปกรณ์ I/O จะมีรีจิสเตอร์อยู่ภายในซึ่งประกอบด้วย data register ทำหน้าที่รับส่งข้อมูลกับไมโครโพรเซสเซอร์และ status register สำหรับบอกค่าสถานะของอุปกรณ์ให้ซีพียูได้รับทราบ

จากรูปในสไลด์หน้า 5 จะเห็นว่าในการที่จะติดต่อกับอุปกรณ์ I/O device ได้นั้นเราจะต้องมีวงจรอินเทอร์เฟสซึ่งประกอบไปด้วย address decoder, data/status register และวงจร control ซึ่งทั้งสามส่วนนี้จะเป็นตัวกลางระหว่างไมโครโพรเซสเซอร์กับอุปกรณ์ I/O device โดยการติดต่อระหว่างอุปกรณ์อินเทอร์เฟสและอุปกรณ์ I/O นี้จะสามารถทำได้สองวิธีดังแสดงในสไลด์หน้า 6 ซึ่งเราจะเห็นได้ว่ามีอยู่สองวิธีด้วยกันคือวิธีแรกเป็นการติดต่อสื่อสารแบบอนุกรม ซึ่งวิธีนี้จะเป็นการส่งข้อมูลจากต้นทางไปปลายทางทีละ 1 บิต ซึ่งมีข้อดีคือประหยัดจำนวนสายสัญญาณในการส่ง และสามารถส่งได้ระยะทางไกลได้ดีกว่าการส่งแบบอื่น ส่วนการส่งแบบขนาน จะเป็นการส่งข้อมูลจากต้นทางไปปลายทางทีละหลายๆ บิตพร้อมๆ กัน ซึ่งมีข้อดีคือมี

ความเร็วในการส่งสูงกว่า แต่ก็มีข้อเสียคือค่าใช้จ่ายสูงกว่าแบบอนุกรมเนื่องจากจำนวนสายที่ใช้เชื่อมต่อมีมากกว่า และระยะทางที่ส่งก็ได้ไม่ไกลเท่ากับแบบอนุกรม

สไลด์หน้าที่ 7 แสดงให้เราเห็นถึงรูปแบบวิธีการในการติดต่อกับอินพุตเอาต์พุต ซึ่งมีอยู่ด้วยกัน สามวิธีด้วยกัน ได้แก่

- Polling
- Interrupt
- DMA

สไลด์หน้าที่ 8-14 แสดงวิธีการ Polling ซึ่งทำงานโดยการที่ไมโครโพรเซสเซอร์จะต้องมีการวนลูปสอบถามสถานะความพร้อมในการติดต่อของอุปกรณ์ I/O ตลอดเวลา ยกตัวอย่างเช่น สมมุติว่ามีอุปกรณ์ I/O อยู่ในระบบ 2 ตัวคือ คีย์บอร์ด และเมาส์ การทำงานเริ่มจากซีพียูจะต้องสอบถามไปยังคีย์บอร์ดว่ามีการกดคีย์ใดๆ หรือไม่ หากมีการกดคีย์ ซีพียูก็จะรับค่าคีย์นั้นไปใช้งาน แต่ถ้าไม่มีการกดคีย์ ตัวคีย์บอร์ดก็จะแจ้งกลับไปยังซีพียูว่ายังไม่มีมีการกดคีย์ใดๆ เมื่อสอบถามและรับส่งค่ากับคีย์บอร์ดเสร็จซีพียูก็จะไปสอบถามสถานะการทำงานของเมาส์เป็นลำดับถัดไป ว่ามีการเลื่อนเมาส์หรือกดปุ่มใดๆ ของเมาส์หรือไม่ หากมีซีพียูก็จะรับค่าตำแหน่งของเมาส์และสถานะการกดปุ่มของเมาส์ไปใช้งาน แต่ถ้าไม่มีเมาส์ก็จะแจ้งให้ซีพียูได้รับทราบ เมื่อสอบถามและรับส่งค่ากับเมาส์เสร็จซีพียูก็จะกลับไปสอบถามคีย์บอร์ดอีกครั้งเป็นลำดับถัดไป ซึ่งเราจะเห็นได้ว่าการติดต่ออุปกรณ์ I/O แบบนี้ซีพียูจะสูญเสียเวลาในการทำงานส่วนหนึ่งไปในการสอบถามสถานะของอุปกรณ์ I/O ทำให้ซีพียูเหลือเวลาที่จะมาทำงานให้กับโปรแกรมหลักลดลง วิธีการติดต่ออุปกรณ์ I/O แบบการใช้อินเทอร์รัปต์ เป็นวิธีที่ออกแบบมาเพื่อแก้ไขปัญหาดังกล่าว

สไลด์หน้าที่ 15-20 แสดงการทำงานวิธีการติดต่อ I/O แบบ Interrupt ซึ่งในวิธีนี้ซีพียูจะไม่ต้องคอยวนลูปสอบถามความพร้อมในการรับส่งข้อมูลของอุปกรณ์ I/O device ที่ละตัวอีกต่อไป ซีพียูจะทำงานปกติไปเรื่อยๆ หากมีอุปกรณ์ I/O ตัวใดต้องการรับส่งข้อมูลกับซีพียู อุปกรณ์ I/O นั้นจะส่งสัญญาณอินเทอร์รัปต์ไปให้กับซีพียูเพื่อให้ซีพียูหยุดการทำงานปัจจุบันไว้ชั่วคราวก่อน และหันมาให้บริการกับอุปกรณ์ I/O เมื่อซีพียูติดต่อกับอุปกรณ์ I/O เสร็จแล้วมันจะกลับไปทำงานเดิมที่ค้างค้างอยู่ต่อไป

วิธีการอินเทอร์รัปต์นี้ ซีพียูต้องมีขารับสัญญาณอินเทอร์รัปต์จากอุปกรณ์ I/O เพิ่มขึ้นมา และเมื่อเกิดอินเทอร์รัปต์ ซีพียูจะเรียกโปรแกรมบริการอินเทอร์รัปต์ (Interrupt service routine: ISR) ซึ่งเป็นโปรแกรมที่ออกแบบไว้สำหรับการติดต่อกับอุปกรณ์ I/O แต่ละตัวโดยเฉพาะ ในระบบอาจจะมีโปรแกรมบริการอินเทอร์รัปต์หลายๆ ตัว เช่นโปรแกรมบริการอินเทอร์รัปต์ของคีย์บอร์ด โปรแกรมบริการอินเทอร์รัปต์ของเมาส์ เป็นต้น ซึ่งโปรแกรมบริการอินเทอร์รัปต์นี้จะต้องเก็บไว้ในหน่วยความจำหลักของระบบ

ในการอินเทอร์รัปต์ซีพียูนั่น มีปัจจัยที่ต้องคำนึงถึงอยู่ นั่นคือหากมีอุปกรณ์หลายตัวส่งสัญญาณอินเทอร์รัปต์มาให้กับซีพียูแล้วซีพียูจะต้องมีการตัดสินใจว่าจะให้บริการกับอุปกรณ์ตัวใด ซึ่งมีอยู่สองวิธีด้วยกัน คือ Centralized arbitration และแบบ Daisy chain connection ซึ่งในวิธีแรก ซีพียูจะต้องมีขาสัญญาณอินเทอร์รัปต์หลายๆ เส้น และแต่ละเส้นจะมีการจัดลำดับความสำคัญเอาไว้และสัญญาณอินเทอร์รัปต์ทุกเส้นจะต่อเข้ากับวงจร Interrupt priority arbitration ซึ่งเป็นวงจรที่ทำหน้าที่ตัดสินใจลำดับ

ความสำคัญของขาสัญญาณอินเทอร์รัพต์เส้นต่างๆ ภายในซีพียูดังแสดงให้เห็นในสไลด์หน้าที่ 19 วิธีดังกล่าวมีข้อเสียคือซีพียูจะต้องมีขาอินเทอร์รัพต์หลายๆ เส้น ซึ่งบางระบบคอมพิวเตอร์บางแบบได้ออกแบบให้ซีพียูมีขาสัญญาณอินเทอร์รัพต์เพียงเส้นเดียวและมีไอซีทำหน้าที่จัดการอินเทอร์รัพต์หลายๆ เส้นข้างนอก ซึ่งเราเรียกไอซีนี้ว่า Interrupt Controller ตัวอย่างเช่นไอซี 8259A ของบริษัทอินเทล เป็นต้น ส่วนวิธีการเชื่อมต่ออุปกรณ์แบบ Daisy chain นั้นจะเป็นอีกวิธีหนึ่งที่ใช้จัดการลำดับความสำคัญของการอินเทอร์รัพต์ได้ ซึ่งแสดงให้เห็นในสไลด์หน้าที่ 20 ซึ่งในวิธีนี้ซีพียูจะรับสัญญาณอินเทอร์รัพต์จากอุปกรณ์ใดๆ แค่เส้นเดียวเท่านั้น ซีพียูเมื่อรับสัญญาณอินเทอร์รัพต์แล้วจะตอบรับการอินเทอร์รัพต์โดยการส่งสัญญาณ Interrupt acknowledge (INTA) กลับไปให้กับอุปกรณ์ I/O โดยในวิธีนี้อุปกรณ์ตัวที่ต่อขาสัญญาณ INTA กับซีพียูโดยตรงจะมีความสำคัญในการได้รับบริการอินเทอร์รัพต์สูงสุด และอุปกรณ์ที่อยู่ไกลจากซีพียูมากที่สุดจะมีลำดับความสำคัญในการอินเทอร์รัพต์ต่ำสุด อุปกรณ์ตัวใดๆ เมื่อต้องการส่งอินเทอร์รัพต์จะต้องรู้ว่าอุปกรณ์ตัวที่มีลำดับความสำคัญสูงกว่ากำลังขออินเทอร์รัพต์อยู่หรือไม่ หากไม่มีมันจะส่งสัญญาณอินเทอร์รัพต์และซีพียูจะส่งสัญญาณตอบรับมาให้กับอุปกรณ์ที่มีลำดับความสำคัญสูงสุดซึ่งจะส่งสัญญาณตอบรับมาให้กับอุปกรณ์ที่ขออินเทอร์รัพต์ และอุปกรณ์ที่ขออินเทอร์รัพต์นั้นจะต้องส่งสัญญาณไปบอกให้อุปกรณ์ที่มีลำดับความสำคัญต่ำกว่าไม่ให้ส่งอินเทอร์รัพต์ไปให้ซีพียู

สไลด์หน้าที่ 21-22 กล่าวถึงวิธีการส่งข้อมูลระหว่างอุปกรณ์ I/O กับหน่วยความจำโดยตรงโดยไม่ผ่านซีพียู หรือการเข้าถึงเอ็มเม (Direct memory access : DMA) โดยวิธีการนี้เหมาะกับส่งข้อมูลปริมาณมากๆ และมีความต่อเนื่องระหว่างหน่วยความจำและอุปกรณ์ I/O ในการที่จะทำดีเอ็มเอได้ จะต้องมีส่วนควบคุมดีเอ็มเอหรือ DMA controller ซึ่งทำหน้าที่ส่งสัญญาณไปบอกซีพียูเพื่อขอการใช้บัส ซีพียูจะตอบรับด้วยการยกตัวเองลอยตัวจากบัสทั้งสามจากนั้นดีเอ็มเอคอนโทรลเลอร์จะควบคุมการส่งข้อมูลระหว่างอุปกรณ์ I/O กับหน่วยความจำโดยตรง

สไลด์หน้าที่ 23 แสดงให้เห็นถึงการเชื่อมต่อซีพียู หน่วยความจำ และอุปกรณ์ I/O ด้วยกันผ่านระบบบัสแบบบัสเดียว (single bus) ซึ่งเป็นระบบที่ง่ายที่สุด และมีใช้ในระบบไมโครโพรเซสเซอร์ควบคุมโดยทั่วไป แต่อย่างไรก็ตามวิธีนี้มีข้อเสียคืออุปกรณ์ I/O มักจะเป็นคอขวดในการรับส่งข้อมูลของระบบเนื่องจากอุปกรณ์ I/O มักมีคุณลักษณะที่เหมือนกันคือการรับส่งข้อมูลช้าเมื่อเทียบกับหน่วยความจำ ดังนั้นระบบคอมพิวเตอร์สมัยใหม่มักนิยมใช้ระบบบัสแบบ Mezzanine bus ดังแสดงให้เห็นในสไลด์หน้าที่ 24 ซึ่งระบบนี้จะต่อซีพียูกับหน่วยความจำผ่านบัสระบบ (system bus) และต่ออุปกรณ์ I/O กับระบบผ่าน expansion bus โดยจะมีวงจร Expansion bus interface เพื่อเชื่อมต่อระบบบัสทั้งสองเข้าด้วยกัน ตัวอย่างของ Expansion bus ที่เห็นได้ง่ายที่สุดคือบัสพีซีไอ (PCI bus)

ระบบบัสพีซีไอเป็นมาตรฐานบัสที่กำหนดขึ้นโดยบริษัทอินเทล ซึ่งกลายเป็นระบบบัสมาตรฐานอุตสาหกรรม และสามารถเข้ากับซีพียูตระกูลใดก็ได้ ซึ่งในท้องตลาดได้มีคอมพิวเตอร์หลายแพลตฟอร์มด้วยกันที่นำระบบบัสพีซีไอไปใช้ เช่นสถาปัตยกรรม 80x86 ของอินเทล หรือเครื่องคอมพิวเตอร์ McIntosh ของบริษัทแอปเปิ้ล หรือเครื่องคอมพิวเตอร์ตระกูล Sparc ของบริษัท ซันไมโครซิสเต็ม เป็นต้น ความเร็วของระบบบัสพีซีไอมีอยู่สองความเร็วคือ 33 MHz และ 66 MHz เนื่องจากมาตรฐานพีซีไอบัสสนับสนุนการส่ง

ข้อมูลทั้งแบบ 32 บิตและ 64 บิต ดังนั้นจึงสามารถรับส่งข้อมูลได้ด้วยความเร็วสูงสุด 133 เมกะไบต์ต่อวินาทีที่การส่งข้อมูลครั้งละ 32 บิตที่ความเร็ว 33 MHz สไลด์หน้าที่ 26 แสดงให้เห็นสล็อตพีซีไชนา 32 บิต ซึ่งมีใช้งานบนเมนบอร์ดของเครื่องคอมพิวเตอร์ส่วนบุคคลโดยทั่วไป

สไลด์หน้าที่ 27 แสดงให้เห็นถึงการเชื่อมต่อกับระบบอุปกรณ์ต่างๆในระบบผ่านระบบบัสพีซีไช ซึ่งจะเห็นว่าเราจะมีอุปกรณ์ควบคุมบัสพีซีไชอยู่ในส่วนที่เรียกว่า bridge/memory controller ซึ่งอุปกรณ์นี้จะเป็นตัวควบคุมการทำงานของอุปกรณ์บนบัสทั้งหมด ระบบบัสพีซีไชยังอนุญาตให้ระบบบัสอื่นบนบัสพีซีไชได้ผ่านทางอุปกรณ์ที่เรียกว่า bus bridge เช่นระบบบัส ISA เป็นต้น จากรูปจะเห็นว่ามีการต่ออุปกรณ์ I/O ทั้งหมดลงบนบัสพีซีไช เช่น การ์ดแลน การ์ดเสียง การ์ดจอ เป็นต้น

สไลด์หน้าที่ 28-36 แสดงให้เห็นรูปคลื่นสัญญาณควบคุมของการรับส่งข้อมูลในระบบบัสพีซีไช โดยจะเป็นการอ่านข้อมูล 1 ครั้ง เนื่องจากบัสพีซีไชเป็นบัสแบบซิงโครนัส จึงต้องมีสัญญาณนาฬิกาไว้เข้าจังหวะการทำงาน โดยเริ่มจากส่งสัญญาณแอดเดรสออกที่ขา AD และสัญญาณ Initiator ready : IRDY เพื่อบอกว่า initiator พร้อมรับข้อมูลแล้ว ตัว initiator จะส่งคำสั่งผ่านทางขาสัญญาณ C/BE เมื่อ Target พร้อมที่จะส่งข้อมูลแล้วมันจะแอกติฟสัญญาณ Target ready : TRDY พร้อมทั้งส่งข้อมูลออกที่ขา AD จะเห็นได้ว่าระบบบัสพีซีไชจะทำการมัลติเพล็กซ์ค่าสัญญาณแอดเดรสและข้อมูลส่งไปบนสายกลุ่มเดียวกัน ทั้งนี้เพื่อประหยัดจำนวนสายสัญญาณบนระบบบัสนั่นเอง

สไลด์หน้าที่ 37 แสดงการอ่านข้อมูลบนบัสในแบบ burst mode ซึ่งจะเห็นว่าการส่งแอดเดรสเริ่มต้นที่ต้องการอ่านข้อมูลแค่ครั้งเดียว หลังจากนั้นจะเป็นการรับส่งข้อมูลชุดที่ 1, 2, 3, ..... ไปจนถึงข้อมูลชุดสุดท้าย ส่วนสไลด์หน้าที่ 38 แสดงการอ่านข้อมูลในแบบ burst mode แบบมีการส่งสัญญาณรอคอย ซึ่งในกรณีที่อุปกรณ์ target ไม่พร้อมที่จะส่งสัญญาณมันจะบอกให้ initiator รู้ด้วยการดึงสัญญาณ TRDY ให้เป็นลอจิกหนึ่ง สไลด์หน้าที่ 39-48 แสดงการเขียนข้อมูลในบัสพีซีไช เริ่มตั้งแต่แบบเขียนข้อมูลแคชชุดเดียว และตามด้วยการเขียนแบบ burst write

สไลด์ในหน้าที่ 49 แสดงให้เห็นถึงโครงสร้างขาสัญญาณต่างๆ ของพอร์ตอนุกรม ซึ่งใช้ในเครื่อง PC โดยทั่วไปโดยสร้างตามมาตรฐาน RS-232C ซึ่งกำหนดโดยหน่วยงาน EIA โดยสนับสนุนการส่งข้อมูลแบบฟูลดูเพลกซ์ ซึ่งสามารถรับและส่งข้อมูลได้ในเวลาเดียวกัน ในการรับส่งข้อมูลอนุกรมนี้จะต้องมีไอซีที่ทำหน้าที่แปลงข้อมูลระหว่างแบบอนุกรมและขนาน ซึ่งไอซีนี้คือไอซี UART: Universal Asynchronous Receiver/Transmitter ซึ่งไอซีนี้จะฝังตัวอยู่ใน chipset บนเมนบอร์ด สไลด์หน้าที่ 52 แสดงฟอร์มेटข้อมูลที่ส่งผ่าน serial port ซึ่งจะมี start bit ใช้บอกว่าเริ่มมีการส่งข้อมูลแล้วตามด้วยบิตของข้อมูลที่ต้องการตั้งแต่บิตที่ 0, 1, .... จนถึงบิตสุดท้าย และตามด้วยบิตพาริตีไว้ตรวจสอบความผิดพลาด และตามด้วย stop bit เพื่อบอกจุดสิ้นสุดของข้อมูล

การอินเตอร์เฟสแบบอนุกรมมีข้อดีคือง่ายในการเชื่อมต่อ ประหยัดสาย แต่จะมีข้อเสียคือ ช้า ดังนั้นในงานที่ต้องการความเร็วสูงจะเหมาะสมกว่า สไลด์หน้าที่ 53 แสดงตัวอย่างการอินเตอร์เฟสคีย์บอร์ดผ่านพอร์ตขนานและสไลด์หน้าที่ 54 แสดงตัวอย่างการอินเตอร์เฟสเครื่องพิมพ์ผ่านพอร์ตขนาน

สไลด์หน้าที่ 55 แสดงโครงสร้างระบบอินพุตเอาต์พุตของเครื่อง PC ที่เราใช้กันโดยทั่วไป ซึ่งเราจะเป็นว่าบนเมนบอร์ดส่วนใหญ่จะรวบรวมวงจรรีโมตอร์เฟส PCI, ฮาร์ดดิสก์, ฟลอปปีดิสก์, คีย์บอร์ด, พอร์ตอนุกรม, พอร์ตนานาเอไอไว้ในไอซีเพียงตัวเดียวซึ่งเราเรียกไอซีนี้ว่า South bridge โดยสไลด์หน้าที่ 55 แสดงชิปเซ็ตของบริษัท VIA ส่วนสไลด์หน้าที่ 56 แสดงชิปเซ็ตของบริษัท SiS

ในระบบคอมพิวเตอร์สมัยใหม่จะมีพอร์ตอนุกรมความเร็วสูงให้ใช้งานเช่น พอร์ต USB และพอร์ต Firewire โดยมาตรฐาน USB (Universal Serial Bus) จะเป็นที่นิยมในพีซีต่างๆ ไปมากกว่าเนื่องจากมีราคาถูก โดยมาตรฐานยูเอสบีจะมีความเร็วที่ 1.5 Mbps ที่ Low-speed mode และ 12 Mbps ที่ Full-speed mode และ 480 Mbps ที่ High-speed mode คุณสมบัติเด่นของบัสยูเอสบีคือ Plug-and-play หรือเสียบแล้วเล่นได้ โดยที่ผู้ใช้ไม่ต้องกำหนดค่าแอดเดรสและค่า IRQ ของอุปกรณ์ด้วยตนเอง นอกจากนี้อุปกรณ์ยังมีความสามารถที่เรียกว่า Hot Swap นั่นก็คือสามารถเสียบเข้าออกในระบบได้โดยไม่ต้องปิดเครื่องก่อน บัสยูเอสบีสนับสนุนอุปกรณ์บนบัสได้มากที่สุดเท่ากับ 127 ตัว

สไลด์หน้าที่ 59 แสดงให้เห็น Topology การเชื่อมต่ออุปกรณ์ของระบบบัสยูเอสบีซึ่งจะเป็นแบบสตาร์ โดยอุปกรณ์ควบคุมหลักของบัสยูเอสบีจะถูกเรียกว่า USB host controller ซึ่งจะเชื่อมต่อกับระบบผ่านทางระบบบัสพีซีอีกทีหนึ่ง ที่โฮสต์คอนโทรลเลอร์จะมีจุดเชื่อมต่อหลักเรียกว่า Root hub ซึ่งเป็นศูนย์กลางของการเชื่อมต่อ โดยเราสามารถต่อ Hub ได้หลายชั้น และเสียบอุปกรณ์ USB ผ่านทางฮับอีกทีหนึ่ง สไลด์หน้าที่ 60 แสดงให้เห็นมาตรฐานคอนเนกเตอร์เชื่อมต่อของบัสยูเอสบีซึ่งมีอยู่ 2 แบบคือแบบ A ซึ่งใช้ในการต่อเข้ากับ Hub และแบบ B ใช้สำหรับต่อกับ อุปกรณ์ปลายทาง

มาตรฐานพอร์ตอนุกรมความเร็วสูงอีกตัวหนึ่งที่เริ่มเป็นที่นิยมในเครื่องคอมพิวเตอร์สมัยใหม่ก็คือมาตรฐาน IEEE1394 หรือ Firewire ซึ่งเป็นบัสอนุกรมสมรรถนะสูงกว่ายูเอสบี ซึ่งสนับสนุนอุปกรณ์บนบัสได้สูงสุด 63 ตัว และมีความเร็ว ตั้งแต่ 25Mbps จนถึง 400 Mbps ในเวอร์ชันแรก และในเวอร์ชันสองหรือ IEEE1394b มีความเร็วได้ถึง 3.2 Gbps มาตรฐาน Firewire มีความสามารถด้าน เสียบแล้วเล่น และ Hot plugging เช่นเดียวกับบัสยูเอสบี แต่มีราคาสูงกว่าพอสมควร จึงได้รับความนิยมน้อยกว่ายูเอสบี

สไลด์หน้าที่ 64 เป็นการเปรียบเทียบคุณสมบัติระหว่างบัสยูเอสบี พอร์ตอนุกรม และพอร์ตนานาเอไอต่างๆ ไปจะเห็นว่าพอร์ตอนุกรมมีความเร็วสูงสุดแค่ 115 กิโลบิตต่อวินาที ในขณะที่พอร์ตนานาเอไอมีความเร็วสูงถึง 115 กิโลบิตต่อวินาทีหรือ 920 กิโลบิตต่อวินาที ในขณะที่บัสยูเอสบีมีความเร็วที่ Full-speed mode สูงถึง 12 Mbps นอกจากนี้ในสไลด์ยังเปรียบเทียบคุณสมบัติในแง่ของความยาวของสายเคเบิลสูงสุด ความสามารถด้าน Plug and play นอกจากนี้ระบบบัสยูเอสบียังสามารถจ่ายไฟให้อุปกรณ์ I/O ได้อีกด้วย โดยจ่ายไฟให้สูงสุดผ่านบัสได้ไม่เกิน 500 มิลลิแอมป์ ซึ่งความสามารถนี้ไม่มีในพอร์ตอนุกรมและพอร์ตนานาเอไอ หากทำงานที่โหมด ECP จะสามารถส่งข้อมูลได้ด้วยความเร็วถึง 3MBps

---