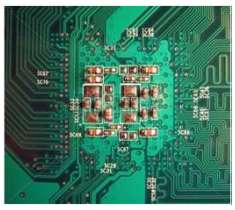
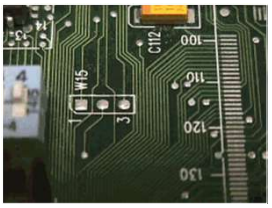
บัสและสถาปัตยกรรมคอมพิวเตอร์
(Bus and Computer Architecture)

- ระบบบัส
- สถาปัตยกรรมคอมพิวเตอร์เบื้องต้น
- สถาปัตยกรรมไมโครโปรเซสเซอร์
- สถาปัตยกรรมคอมพิวเตอร์แบบมัลติโปรเซสเซอร์
- สถาปัตยกรรมคอมพิวเตอร์แบบขนาน

ผู้ช่วยศาสตราจารย์พัชรินทร์ บัวเย็น
(ประกอบตำราไมโครคอมพิวเตอร์และพีชคณิตบูลีน)

Bus System

- ระบบบัสเป็นลายวงจรอิเล็กทรอนิกส์บนเมนบอร์ดและบนการ์ดต่างๆ
- บัสเปรียบได้กับถนนในการขนส่งคำสั่งหรือข้อมูลภายในเครื่องไมโครคอมพิวเตอร์
- บัสมีหน้าที่ในการรับส่งคำสั่งและข้อมูลระหว่างไมโครโปรเซสเซอร์กับส่วนอื่นๆ ของเครื่องคอมพิวเตอร์ เช่นระหว่างไมโครโปรเซสเซอร์กับแรม เพื่อรับส่งข้อมูลไปประมวลผล
- บัสจะต้องมีความกว้างของแบนด์วิดท์สอดคล้องกันกัน เพื่อให้รับ-ส่งข้อมูลเป็นไปด้วยความเร็วคงที่ แต่หากใช้อุปกรณ์และบัสที่มีแบนด์วิดท์ไม่สอดคล้องกัน จะทำให้เกิดปัญหาคอขวดซึ่งส่งผลให้การรับส่งข้อมูลช้าลง

2

Bus System

ส่วนประกอบของระบบบัส

1. เส้นทาง คือเส้นทางต่างๆ ที่ใช้ในการโอนถ่ายข้อมูลบนบัสจากจุดหนึ่งไปยังอีกจุดหนึ่ง ส่วนใหญ่จะสังเกตได้จากเป็นเส้นบนเมนบอร์ดและการ์ดต่างๆ
2. ชิปควบคุม (Control Chip) ทำหน้าที่บริหารการเข้าใช้บัสของชิ้นส่วนต่างๆ และทำหน้าที่ป้องกันปัญหาการขัดแย้ง เนื่องจากการแย่งใช้บัสในเวลาเดียวกัน
3. สล๊อตต่อขยาย (Expansion Slot) เป็นส่วนที่ใช้สำหรับการติดต่อกับการ์ดเสริมต่างๆ ซึ่งสล๊อตต่อขยายจะถูกออกแบบให้ตรงกับระบบบัสนั้นๆ เช่น ระบบพีซีไอบัส (PCI Bus) ก็มีจะสล๊อต พีซีไอ (PCI Slot) ซึ่งใช้ติดตั้งการ์ดแบบพีซีไอ (PCI Card) เป็นต้น

3

Bus System

ความเร็วของบัส

- ความเร็วของบัสจึงมีผลกับความเร็วในการประมวลผลโดยรวมของเครื่องคอมพิวเตอร์ นั่นคือ ยิ่งบัสมีความเร็วสูงและมีจำนวนบิตมากเท่าใด ก็จะทำให้เครื่องคอมพิวเตอร์มีความเร็วมากตามไปด้วย เพราะบัสเป็นสื่อกลางที่ใช้ในการติดต่อสื่อสารระหว่างชิ้นส่วนต่างๆ ในเครื่องคอมพิวเตอร์ เช่น หากบัสระหว่างซีพียูกับแคช แคชกับแรม และแรมกับฮาร์ดดิสก์เร็วอย่างเป็นระบบก็จะทำให้การรับส่งข้อมูลเพื่อประมวลผลเร็วตามไปด้วย
- ข้อจำกัดของความเร็วบัสขึ้นอยู่กับสัญญาณรบกวน (Noise) เพราะยิ่งบัสใช้ความเร็ว (ความถี่ของสัญญาณนาฬิกาของบัส) สูงขึ้นเท่าใดสัญญาณรบกวนก็จะเพิ่มขึ้นเท่านั้น หากสัญญาณรบกวนมากขึ้นก็จะทำให้โอกาสของข้อมูลที่ผ่านบัสมีความผิดพลาดมากขึ้น
- ความเร็วของสัญญาณนาฬิกาของระบบคอมพิวเตอร์จะมีความเร็วสูงกว่าระบบบัสมาก เนื่องจากเส้นทางบัสบนเมนบอร์ดมีระยะทางยาวมาก เมื่อเทียบกับระยะทางระหว่างแต่ละหน่วยในซีพียู ดังนั้นระบบบัสจึงต้องมีชิปควบคุมเพื่อควบคุมความเร็วและทิศทางในการรับส่งข้อมูลภายในระบบคอมพิวเตอร์

4

Bus System

อุปกรณ์ที่ได้รับผลจากการเปลี่ยนแปลงความเร็วบัส

การรับส่งข้อมูลและคำสั่งระหว่างอุปกรณ์ต่างๆ ในเครื่องคอมพิวเตอร์มีผลต่อความเร็วในการประมวลผล นั่นคือ หากความเร็วของบัสมีการเปลี่ยนแปลงจะส่งผลต่อความเร็วในการประมวลผล ซึ่งอุปกรณ์ที่ได้รับผลจากการเปลี่ยนแปลงความเร็วของบัส ได้แก่

1. แรม
2. แคชบนเมนบอร์ด
3. การ์ดควบคุมการแสดงผล
4. การ์ดเพิ่มขยาย
5. ฮาร์ดดิสก์
6. ชิปเซต

5

Bus System

ชนิดของบัส

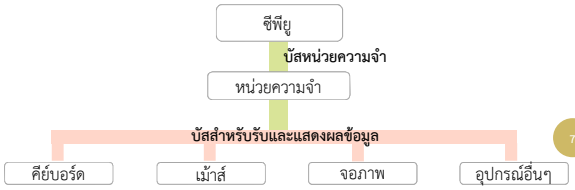
1. ชนิดของบัสที่แบ่งตามสัญญาณที่ส่ง
2. ชนิดของบัสที่แบ่งตามหน่วยการเชื่อมต่อ
3. ชนิดของบัสที่แบ่งตามข้อมูลที่ส่ง

1. ชนิดของบัสที่แบ่งตามสัญญาณที่ส่งไปบนบัส สามารถแบ่งออกได้ 3 ชนิด ดังนี้
   - เพาเวอร์บัส (Power Bus) เป็นเส้นทางบัสสำหรับจ่ายไฟฟ้า
   - บัสข้อมูล (Data Bus) เป็นเส้นทางสำหรับระบบข้อมูล
   - กราวด์บัส (Ground Bus) เป็นเส้นทางบัสสำหรับสายดิน

6

Bus System

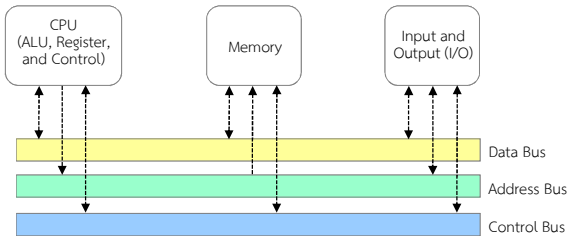
2. ชนิดของบัสที่แบ่งตามหน่วยการเชื่อมต่อในการติดสื่อสาร สามารถแบ่งเป็น 2 ชนิด ดังนี้
   - บัสหน่วยความจำ (Memory Bus) เป็นบัสความเร็วสูงที่ใช้เชื่อมต่อระหว่างซีพียูกับหน่วยความจำ
   - บัสรับและแสดงผลข้อมูล (I/O Bus) เป็นบัสที่เชื่อมต่อระหว่างซีพียูกับอุปกรณ์รับและแสดงผล ซึ่งมีอัตราการส่งข้อมูลช้ากว่าบัสหน่วยความจำแต่ด้วยเป็นการติดต่อกับอุปกรณ์รับส่งข้อมูลจึงถูกออกแบบและนำไปใช้อย่างกว้างขวาง โดยข้อมูลที่ส่งในบัสแบบนี้จะถูกส่งไปยังอุปกรณ์ที่ได้ระบุไว้ตามรูปแบบของแต่ละอุปกรณ์



Bus System

3. ชนิดของบัสที่แบ่งตามรูปแบบข้อมูลที่ส่งบนบัส สามารถแบ่งได้เป็น 3 ชนิด ดังนี้
   - บัสข้อมูล (Data Bus) เป็นบัสที่รับส่งข้อมูลในรูปสัญญาณไฟฟ้า (0/1) มี 24 เส้น บัสนี้จะถูกต่อเข้ากับอุปกรณ์ต่างๆ เช่น หน่วยความจำ และการ์ด เป็นต้น
   - บัสแอดเดรส (Address Bus) เป็นบัสที่รับส่งตำแหน่งที่อยู่ (Address) ของข้อมูลหรือคำสั่งในหน่วยความจำ ซึ่งจำนวนเส้นจะมีความแตกต่างไปตามซีพียูแต่ละรุ่น เช่น ไมโครโปรเซสเซอร์รุ่น 8088 มีบัสแอดเดรส จำนวน 20 เส้น ดังนั้นจึงจะอ้างถึงหน่วยความจำได้ 2^{20} หรือ 1 MB ส่วนไมโครโปรเซสเซอร์รุ่น 80286 มีบัสแอดเดรสจำนวน 24 เส้น นั่นคือสามารถอ้างถึงหน่วยความจำได้ 2^{24} หรือ 16 MB เป็นต้น
   - บัสควบคุม (Control Bus) เป็นบัสที่รับส่งสัญญาณควบคุม ว่าจะให้อุปกรณ์ที่ได้รับข้อมูลนั้นจัดการกับข้อมูลอย่างไร

Bus System



Bus System

<u>การติดต่อสื่อสารระหว่างบัสกับฮาร์ดแวร์ภายในเครื่องคอมพิวเตอร์</u>

การติดต่อสื่อสารระหว่างบัสกับอุปกรณ์และชิ้นส่วนต่างๆ ภายในระบบคอมพิวเตอร์ สามารถแบ่งออกได้ 3 รูปแบบ ดังนี้

1. การติดต่อสื่อสารระหว่างไมโครโปรเซสเซอร์กับหน่วยความจำ
2. การติดต่อสื่อสารระหว่างไมโครโปรเซสเซอร์หรือหน่วยความจำกับหน่วยรับข้อมูล
3. การติดต่อสื่อสารระหว่างไมโครโปรเซสเซอร์หรือหน่วยความจำกับหน่วยแสดงผล

1. การติดต่อสื่อสารระหว่างไมโครโปรเซสเซอร์กับหน่วยความจำ มีจุดประสงค์ดังนี้
   - เพื่อการดึงชุดคำสั่ง
   - เพื่อการดึงเครื่องหมายทางคณิตศาสตร์เพื่อการคำนวณ
   - เพื่อเก็บผลลัพธ์

Bus System

2. การติดต่อสื่อสารระหว่างไมโครโปรเซสเซอร์หรือหน่วยความจำกับหน่วยรับข้อมูล มีจุดประสงค์ดังนี้
   - เพื่อรับข้อมูลจากหน่วยรับข้อมูล
   - เพื่อส่งคำสั่งให้กับหน่วยรับข้อมูล
   - เพื่ออ่านสถานะของหน่วยรับข้อมูล

3. การติดต่อสื่อสารระหว่างไมโครโปรเซสเซอร์หรือหน่วยความจำกับหน่วยแสดงผล มีจุดประสงค์ดังนี้
   - เพื่อรับผลลัพธ์มาจากไมโครโปรเซสเซอร์หรือหน่วยความจำส่งไปยังหน่วยแสดงผล
   - เพื่อส่งคำสั่งจากไมโครโปรเซสเซอร์ไปยังหน่วยแสดงผล
   - เพื่ออ่านสถานะของหน่วยแสดงผลแล้วส่งสถานะไปยังโปรเซสเซอร์

Bus System

<u>ประเภทของบัส</u>

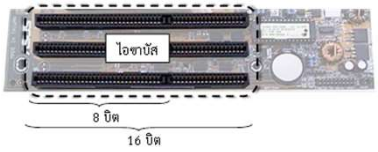
การติดต่อสื่อสารระหว่างบัสกับอุปกรณ์และชิ้นส่วนต่างๆ ภายในระบบคอมพิวเตอร์ สามารถแบ่งออกได้ 3 รูปแบบ ดังนี้

1. ไอซ่าบัส (ISA Bus)
2. ไมโครชาแนลบัส (Micro Chanel Bus)
3. อีซ่าบัส (ISA Bus)
4. วีซ่าบัส (VESA Bus)
5. พีซีไอบัส (PCI Bus)
6. เอจีพีบัส (AGP Bus)
7. พีซีเอ็มซีไอเอบัส (PCMCIA)
8. เอสซีเอสไอ (SCSI)
9. อาร์เอส-232 (RS-232)
10. ยูเอสบี (USB)
11. ไฟร์ไวร์ (FireWire)

Bus System

1. ไอซ่า (ISA: Industry Standard Architecture)

- ไอซ่าบัส เป็นเทคโนโลยีที่เป็นมาตรฐานเกี่ยวกับบัสที่ใช้ในเครื่องคอมพิวเตอร์ของไอบีเอ็ม ที่พัฒนาขึ้นในปี ค.ศ. 1984 ซึ่งมีความเร็ว 8 MHz
- ไอซ่าบัสสนับสนุนต่อการใช้การ์ดขยาย (Expansion Card) ชนิด 8 และ 16 บิต ซึ่งข้อมูลจะถูกส่งไปบนสายนำสัญญาณ 8 เส้น หรือ 16 เส้น (ขึ้นอยู่กับชนิดของการ์ด)
- ไอซ่าบัสมีความเร็วตั้งแต่ 8 -16 MHz เช่น ใช้กับซีพียูรุ่น 286 ความเร็ว 12 MHz หรือซีพียูรุ่น 386 SX ความเร็ว 16 MHz
- แต่หลังจากการพัฒนาซีพียูขนาด 32 บิต ทำให้มาตรฐานไอซ่าบัสเริ่มไม่เหมาะสม เพราะต้องส่งข้อมูลถึง 2 ครั้ง เพื่อให้ได้ข้อมูล 32 บิต



Bus System

2. ไมโครชาแนลบัส (Micro Chanel Bus)

- ไมโครชาแนลบัส หรือ เอ็มซีเอบัส (MCA Bus: Micro Channel Architecture) คือ เทคโนโลยีที่รองรับการใช้การ์ดขยาย ชนิด 32 บิต คิดค้นโดยไอบีเอ็ม เปิดตัวครั้งแรกในเดือนเมษายน ค.ศ. 1987
- ความเร็วในการถ่ายโอนข้อมูลภายในระบบคอมพิวเตอร์ของไมโครชาแนลบัสเร็วกว่าไอซ่าบัส และมีระดับการกวนสัญญาณน้อยกว่า
- นอกจากนี้ไมโครชาแนลการ์ด (MCA Card) สามารถกำหนดค่าได้ด้วยซอฟต์แวร์ ซึ่งสะดวกมากกว่าการ์ดไอซ่า
- ผู้ผลิตรายอื่นหากต้องการใช้การ์ดและสล๊อตแบบ ไมโครชาแนลบัสจะต้องซื้อสิทธิบัตรจากไอบีเอ็ม ทำให้ไครชาแนลบัสไม่ได้รับความนิยม
- ข้อดี: ทำให้อุปกรณ์ต่างๆ หรือการ์ดเสริมต่างๆ สามารถติดต่อสื่อสารได้โดยตรง โดยไม่ต้องผ่านซีพียู ดังนั้น ในขณะที่ซีพียูกำลังทำงานอยู่ อุปกรณ์ที่ใช้ไมโครชาแนลบัสจะสามารถติดต่อสื่อสารกันโดยตรงได้ โดยไม่ต้องผ่านซีพียู

Bus System

3. อีซ่าบัส (ISA Bus)

- อีซ่า (EISA: Extended Industry Standard Architecture) คือเทคโนโลยีเกี่ยวกับบัสที่ใช้กับเครื่องไอบีเอ็มคอมแพททิเบิล (IBM Compatible)
- พัฒนาขึ้นโดยกลุ่มผู้ผลิต AST Research, Compaq Computer Corporation, Epson, Hewlett Packard, NEC, Olivetii, Tandy, Wyse Technology เพื่อแข่งกับไมโครชาแนลบัสของไอบีเอ็มที่ออกกฎว่า ผู้ผลิตรายอื่นจะใช้ต้องเสียค่าสิทธิบัตร นั่นคืออีซ่าบัสจะสนับสนุนต่อการใช้งานโดยไม่ต้องเสียค่าสิทธิบัตร
- การ์ดขยายแบบ 32 บิตของอีซ่าบัสเป็นบัสแบบ 32 บิตเหมือนไมโครชาแนลบัส อีกทั้งอีซ่าบัสมีความเร็วในการรับส่งข้อมูลที่ 20 MHz และอนุญาตให้อุปกรณ์ต่อพ่วงสามารถควบคุมบัสได้
- ข้อดี: สามารถใช้ได้กับการ์ดขยายชนิด 8 บิต และ 16 บิต
- ข้อเสีย: มีความเร็วเพียง 20 MHz ทำให้ซีพียูที่มีความเร็วสูงกว่าต้องเข้าถึงหน่วยความจำด้วยความเร็วต่ำไปด้วย จึงทำให้อีซ่าบัสไม่ค่อยเป็นที่นิยม

Bus System

4. วีซ่าบัส (VESA Bus)

- วีซ่าบัส มีหลายชื่อ ได้แก่ วีซ่าโลคอลบัส (VESA Local Bus) วีแอลบัส (VL Bus) และโลคอลบัส (Local Bus)
- วีซ่า (VESA: Video Electronics Standards Association) ถูกตั้งขึ้นตามชื่อกลุ่มผู้สร้างซึ่งเป็นชมรมอุตสาหกรรมกลุ่มหนึ่งที่มีสมาชิกเป็นผู้ผลิตการ์ดและชิปแสดงผล
- ข้อดี คือ รับส่งข้อมูลได้เร็ว
- ข้อจำกัด คือ บัสชนิดนี้ไม่มีมาตรฐานที่แน่นอน ทำให้มีปัญหาความไม่เข้ากัน (Non-Compatible) เมื่อนำการ์ดอื่นมาใช้บนสล๊อตแบบวีซ่าบัส
- วีซ่าบัสมีขนาดของบัสข้อมูล 32 บิตและ 64 บิตซึ่งจะติดต่อระหว่างอุปกรณ์ต่างๆ กับซีพียูโดยตรง ไม่ต้องผ่านบัสคอนโทรลเลอร์ ดังนั้น จึงมีความเร็วค่อนข้างสูงในการติดต่อสื่อสารที่ความถี่ 33 - 40 MHz ทำให้ได้ความเร็ว 133-148 Mbps
- แม้ว่าในช่วงแรกบัสชนิดนี้จะถูกกำหนดมาตรฐานเน้นที่การแสดงผล แต่บัสชนิดนี้ยังสามารถใช้ได้กับการ์ดเพิ่มขยายอื่นที่มีมาตรฐานตรงกันได้ เช่น การ์ดแลน หรือ การ์ดควบคุมดิสก์ (Disk Controller)

Bus System

- สล๊อตของวีซ่าบัสแบ่งเป็น 3 ส่วน โดยมีท่อนหนึ่งเป็นสีน้ำตาล ท่อนที่เหลือมีสีเดียวกับอีซ่าสล๊อต

ที่มา (Pelajarteknik, 2013)

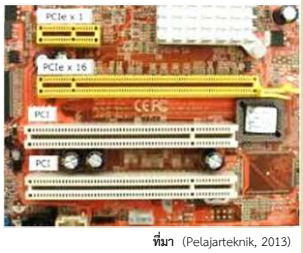
Bus System

5. พีซีไอบัส (PCI Bus)

- พีซีไอบัส (PCI: Peripheral Components Interconnect) เป็นมาตรฐานบัสที่กำหนดโดยบริษัทอินเทล (Intel) เพื่อแก้ปัญหาของวีซ่าบัส ที่มีสล๊อต 3 ส่วน ซึ่งเป็นสาเหตุให้ใส่การ์ดเพิ่มขยายได้จำกัด และไม่สามารถรองรับซีพียูแบบ Pentium ซึ่งต้องการบัสแบบ 64 บิตได้
- พีซีไอบัส เป็นบัสชนิด 64 บิตที่สามารถรับส่งข้อมูลขนาด 32 บิตได้เพื่อรองรับการ์ดรุ่นเก่า และชิปควบคุมของพีซีไอบัสยังสามารถใช้กับชิปควบคุมแบบเก่าได้อีกด้วย
- พีซีไอบัสมีความเร็วที่ 66 MHz และสนับสนุนการทำงานแบบ Plug And Play ได้

Bus System

ตัวอย่างสล๊อตสำหรับใส่การ์ดสำหรับพีซีไอบัส



ที่มา (Pelajarteknik, 2013)

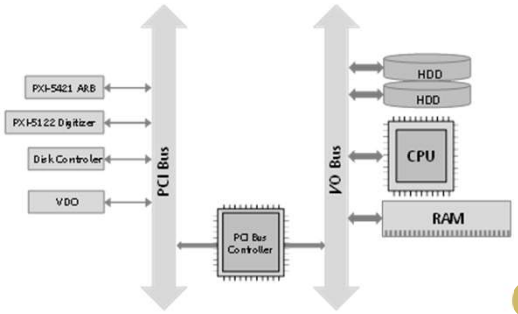
ตัวอย่างการ์ดที่ใช้กับพีซีไอบัส

ที่มา (Karbo, M., & ELI Aps., n.d.)

19

Bus System

ผังแสดงการเชื่อมต่อของพีซีไอบัสกับอุปกรณ์ต่างๆ



ที่มา (National Instruments Corporation, 2014)

20

Bus System

6. เอจีพีบัส (AGP Bus)

- เอพีจี (APG: Accelerated Graphics Port) เป็นสถาปัตยกรรมบนบัสที่เสริมประสิทธิภาพการแสดงผล ใช้กับอินเทลเพนเทียมทู (Intel Pentium II) ซึ่งเปิดตัวเมื่อกลางปี ค.ศ. 1999
- บัสชนิดนี้มีการเชื่อมต่อกับชิปเซตของระบบแบบ Point-To-Point ซึ่งทำให้การส่งผ่านข้อมูลระหว่างการ์ดกับชิปเซตของระบบรวดเร็วขึ้น
- AGP Bus ทำงานที่ความเร็ว 66 MHz (ขึ้นอยู่กับชนิดของหน่วยความจำหลัก นั่นคือถ้าหน่วยความจำหลักเป็นชนิดที่เร็วก็จะยิ่งช่วยเพิ่มอัตราเร็วในการส่งถ่ายข้อมูลได้เร็วมากขึ้นตามไปด้วย)
- AGP Bus มีประสิทธิภาพดีกว่า PCI Bus เพราะมีสล๊อตแบบเอกเทศ นั่นคือไม่ต้องใช้แบนด์วิดท์ร่วมกับอุปกรณ์ใดๆ เพราะเครื่องคอมพิวเตอร์หนึ่งจะมีการ์ดแสดงผลเพียงตัวเดียวเท่านั้น ดังนั้นจึงมีเอจีพีสล๊อต (AGP Slot) บนเมนบอร์ดเพียงสล๊อตเดียว

21

Bus System

- บัสนี้มีเส้นทางเฉพาะของตนเองสำหรับติดต่อกับหน่วยความจำหลักของระบบเพื่อใช้เรนเดอร์ (Render) ภาพแบบสามมิติ (3D) ได้อย่างรวดเร็ว
- เอจีพีการ์ดมีโหมดสำหรับเรนเดอร์ภาพ 2 แบบ คือ โลคอลเท็กซ์เจอร์ (Local Texturing) และเอจีพีเท็กซ์เจอร์ (AGP Texturing)
- การเรนเดอร์ภาพแบบ Local Texturing จะทำการคัดลอกหน่วยความจำของระบบไปเก็บไว้ที่ Buffer บนการ์ด จากนั้นจะดึงข้อมูลจากบัฟเฟอร์ไปประมวลผลต่อไป (วิธีการนี้ได้ถูกใช้บนระบบของ PCI Card ด้วย ซึ่งทำให้ต้องอาศัยขนาดของหน่วยความจำบนการ์ดจำนวนมาก
- AGP Texturing เป็นเทคนิคใหม่ที่ช่วยลดปริมาณของ Buffer บน Display Card ลง โดยให้สามารถใช้หน่วยความจำของระบบแทน Buffer โดยไม่ต้องดึงข้อมูลมาพักไว้ที่ Buffer อีก

22

Bus System

ตัวอย่างการ์ดแบบเอจีพีบัส

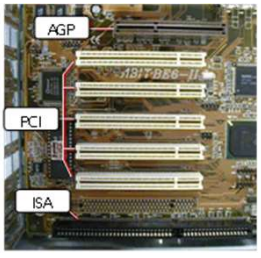
23

ที่มา (Scheffe, U., 2005.)

Bus System

ตัวอย่างสล๊อตแบบเอจีพีบัส

ที่มา (Scheffe, U., 2005.)



24

ที่มา (ITWissen, n.d.)

Bus System

7. พีซีเอ็มซีไอเอบัส (PCMCIA Bus)

- พีซีเอ็มซีไอเอ (PCMCIA: Personal Computer Memory Card International Association) เป็นระบบบัสที่ใช้กับเครื่องคอมพิวเตอร์แบบพกพา หรือโน้ตบุ๊ค
- บัสชนิดนี้รองรับบัสข้อมูลชนิด 16 บิต ความเร็วบัสไม่เกิน 33 MHz และบัสแอดเดรสสำหรับอ้างอิงตำแหน่งหน่วยความจำทั้งหมดเป็นแบบ 26 บิต ดังนั้นจึงอ้างอิงหน่วยความจำสูงสุดได้เพียง 64 MB (2^{26})
- ข้อดีของการ์ดที่ใช้กับบัสชนิดนี้คือมีขนาดเล็กเท่ากับบัตรเครดิต ใช้กับการ์ดแบบต่างๆ เช่น การ์ดเครือข่าย การ์ดบางรุ่นจะติดตั้งระบบการทำงานให้โดยอัตโนมัติเมื่อมีการเสียบการ์ดเข้าหรือถอดออก

25

Bus System

ช่องและการ์ดแบบพีซีเอ็มซีไอเอบัส

26

Bus System

8. เอสซีเอสไอ (SCSI Bus: Small Computer System Interface)

- เอสซีเอสไอ หรือเรียกโดยทั่วไปว่า สกัสซี่ เป็นบัสแบบขนานที่ออกแบบมาเพื่อรองรับอุปกรณ์รับและแสดงผล เช่น ฮาร์ดดิสก์ เครื่องพิมพ์ ซีดีรอม และสแกนเนอร์ เป็นต้น
- สกัสซี่บัส มีความแตกต่างกับบัสแบบขนานทั่วไป คือจะกำหนดตำแหน่งที่แน่นอนให้แต่ละอุปกรณ์
- สกัสซี่บัสถูกออกแบบให้แต่ละอุปกรณ์เชื่อมต่อกับอุปกรณ์อื่นๆ ได้ เรียกว่า เดซี่เชนด์ (Daisy Chained) คืออุปกรณ์จะถูกเชื่อมต่อกันไปเรื่อยๆ คล้ายห่วงโซ่

- อุปกรณ์สุดท้ายจะมีเทอร์มิเนเตอร์ (Terminator) เพื่อป้องกันการสะท้อนกลับของสัญญาณที่ส่งเข้ามา และมีตัวควบคุมสกัสซี่บัส (SCSI Bus Controller) เพื่อควบคุมการส่งสัญญาณไปยังอุปกรณ์ที่ต้องการเพื่อความถูกต้องในการส่งข้อมูล

27

Bus System

9. อาร์เอส-232 (RS-232)

- อาร์เอส-232 เป็นการติดต่อสื่อสารระหว่างคอมพิวเตอร์กับเทอร์มินอลโดยผ่านโมเด็ม (MODEM) โดยอาศัยสายโทรศัพท์ ด้วยวิธีการโมดูเลชัน (Modulation)
- อาร์เอส-232 บัสเป็นมาตรฐานที่ใช้ในการรับส่งข้อมูลด้วยโมเด็ม ซึ่งถูกกำหนดโดยสมาคมอุตสาหกรรมอิเล็กทรอนิกส์ หรืออีไอเอ (EIA: Electronics Industries Association)
- สายเคเบิลที่เชื่อมต่อระหว่างโมเด็มจะต้องมีความยาวไม่เกิน 50 ฟุต ไม่เช่นนั้นคุณภาพของสัญญาณจะลดลง เนื่องจากสัญญาณที่ใช้จะอยู่ในรูปกระแสไฟฟ้าซึ่งความแรงของสัญญาณจะขึ้นอยู่กับระยะทาง

28

Bus System

10. ยูเอสบีบัส (USB: Universal Serial Bus)

ยูเอสบีบัส เป็นบัสที่มีประสิทธิภาพดีกว่าอาร์เอส-232 ในหลายด้าน อาทิ เชื่อมต่อได้ง่าย รับส่งข้อมูลได้เร็ว หรือสามารถเชื่อมต่ออุปกรณ์ได้หลายชิ้น เป็นต้น

โดยคุณสมบัติที่สำคัญของยูเอสบีบัส มีดังนี้

- สามารถเชื่อมต่ออุปกรณ์ได้มากกว่า 127 ตัวบนยูเอสบีบัสเดียวกัน
- สามารถสนับสนุนอุปกรณ์ได้หลากหลายชนิด ไม่ว่าจะเป็นอุปกรณ์ที่ต้องรับส่งข้อมูลด้วยความเร็วสูง เช่น ฮาร์ดดิสก์ และแฟลชไดร์ฟ หรืออุปกรณ์ที่รับส่งข้อมูลด้วยความเร็วต่ำ เช่น คีย์บอร์ด เม้าส์ จอยสติ๊ก หรือเครื่องพิมพ์ เป็นต้น
- การต่ออุปกรณ์แบบยูเอสบีกับเครื่องคอมพิวเตอร์ ไม่จำเป็นต้องปิดและเปิดเครื่องใหม่และไม่จำเป็นต้องลงไดร์ฟเวอร์ อุปกรณ์ที่สามารถใช้งานได้ทันทีเมื่อต่อเชื่อมกับเครื่องคอมพิวเตอร์ เรียกความสามารถเช่นนี้ว่าพลั๊กแอนด์เพล์ (Plug and Play)
- ตัวควบคุมยูเอสบีจะสามารถตรวจสอบอุปกรณ์ที่เชื่อมต่อและจ่ายกำลังไฟในระดับที่เหมาะสมให้กับอุปกรณ์นั้นๆ ได้

29

Bus System

- สามารถรับส่งข้อมูลได้ด้วยความเร็วสูง ซึ่งการใช้งานยูเอสบีในช่วงแรกจะมีความเร็วประมาณ 1.5 Mbps ซึ่งค่อนข้างช้า ต่อมาจึงได้มีการพัฒนาความเร็วเพิ่มขึ้นเป็น 12 Mbps โดยเรียกมาตรฐานการเชื่อมต่อนี้ว่า ยูเอสบี 1.1 (USB 1.1) และพัฒนาต่อมาเป็น ยูเอสบี 2.0 (USB 2.0) ซึ่งมีความเร็วในการรับส่งข้อมูลที่ 480 Mbps นอกจากนี้ปัจจุบันได้มีการพัฒนาให้ยูเอสบีบัสมีความเร็วถึง 2.5-4.8 Gbps. เรียกว่ามาตรฐานยูเอสบี 3.0 (USB 3.0) ซึ่งเริ่มมีการนำไปใช้เพิ่มมากขึ้น

30

Bus System

11. ไฟร์ไวร์ (FireWire)

- ไฟร์ไวร์ (FireWire) หรือ ไอทริปเปิ้ลอี 1394 (IEEE 1394) เป็นมาตรฐานในการเชื่อมต่ออุปกรณ์ต่างๆ โดยสามารถพ่วงต่ออุปกรณ์ได้มากกว่า 60 ชิ้น และสามารถรับส่งข้อมูลได้ถึง 400 Mbps
- ปัจจุบันได้พัฒนาให้มีความเร็วในการรับส่งข้อมูลได้มากถึง 3.2 Gbps ทำให้สามารถถ่ายโอนข้อมูลประเภทภาพเคลื่อนไหว เสียง รวมถึงการประชุมระยะไกลเรียกว่า วีดีโอคอนเฟอร์เรนต์ (VDO Conference)

(รูป: พอร์ตไฟร์ไวร์แบบ 4 พิน 6 พิน และ 9 พิน)

ที่มา (GIMComputer, 2008)

Bus System

คุณสมบัติที่เหมือนกันของไฟร์ไวร์และยูเอสบี

- รับส่งข้อมูลด้วยความเร็วสูง
- สามารถนำสัญญาณไฟฟ้าและสัญญาณข้อมูลได้พร้อมกัน ทำให้อุปกรณ์ที่ต่อกับบัสทั้งสองแบบสามารถใช้ไฟเลี้ยงจากบัสได้เลย
- อุปกรณ์ที่ใช้กับทั้งสองบัสเป็นอุปกรณ์ที่พกพาได้สะดวก
- มีการทำงานแบบพลั๊กแอนด์เพล พร้อมทั้งมีการตรวจสอบคุณสมบัติของอุปกรณ์ และปรับแต่งองค์ประกอบ (Configuration) ของคอมพิวเตอร์และอุปกรณ์ให้สามารถทำงานร่วมกันได้โดยอัตโนมัติ
- สามารถถอดอุปกรณ์ออกจากเครื่องคอมพิวเตอร์ในขณะที่ระบบกำลังทำงานอยู่ได้
- เป็นบัสที่ใช้ได้กับหลากหลายอุปกรณ์ เช่น คีย์บอร์ด เม้าส์ ฮาร์ดดิสก์ และเครื่องพิมพ์

ความแตกต่างระหว่างไฟร์ไวร์กับยูเอสบี

ความเร็วในการรับส่งข้อมูลของไฟร์ไวร์เร็วกว่าของยูเอสบี ดังนั้นไฟร์ไวร์จึงถูกนำไปใช้กับอุปกรณ์ที่ต้องการความเร็วในการรับส่งข้อมูลสูง เช่น กล้องวีดีโอ ในขณะที่ยูเอสบีถูกนำไปใช้กับอุปกรณ์ที่ไม่ต้องอาศัยความเร็วสูง เช่น คีย์บอร์ด เม้าส์ แฟลชไดรฟ์และฮาร์ดดิสก์ เป็นต้น

32

Bus System

สถาปัตยกรรมของบัสแบบบริดจ์

บริดจ์ (Bridge) เป็นรูปแบบการเชื่อมต่อบัสหลายๆ บัสไว้ด้วยกัน โดยจะทำหน้าที่จัดการการติดต่อสื่อสารทั้งหมดของระบบบัสให้มีประสิทธิภาพ

ทางกายภาพของบัส บัสประกอบด้วยสายไฟและตัวเชื่อมต่อ (Connector) ซึ่งทำหน้าที่เชื่อมต่ออุปกรณ์แต่ละตัว โดยสายไฟที่อยู่ในระบบบัสสามารถแบ่งการทำงานได้ 4 ชนิด ได้แก่

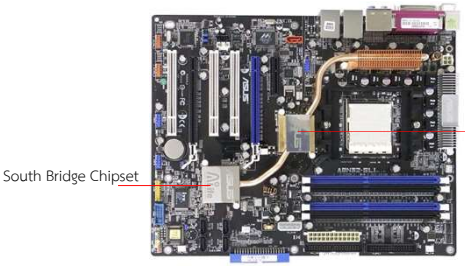
- สายไฟสำหรับรับส่งสัญญาณควบคุม (Control)
- สายไฟสำหรับรับส่งข้อมูล (Data)
- สายไฟสำหรับรับส่งตำแหน่ง (Address)
- สายไฟสำหรับเป็นจ่ายไฟ (Power)

33

Bus System

บริดจ์ แบ่งออกเป็น 2 ประเภทคือ

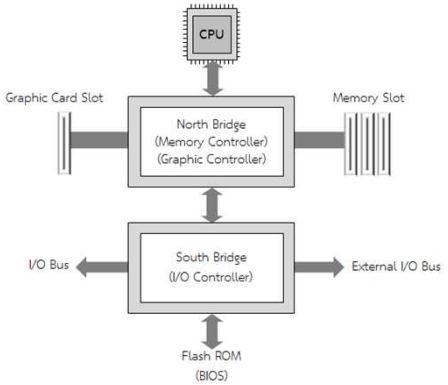
- นอร์ทบริดจ์ (North Bridge) ทำหน้าที่ควบคุมการรับส่งข้อมูลของอุปกรณ์ที่ต้องการความเร็วสูง ได้แก่ หน่วยความจำหลักและการ์ดแสดงผล
- เซาท์บริดจ์ (South Bridge) ทำหน้าที่ควบคุมการรับส่งข้อมูลของอุปกรณ์ที่มีความเร็วต่ำ อาทิ ฮาร์ดดิสก์ การ์ดเสียง หรือการ์ดแลน (LAN Card) เป็นต้น



34

Bus System

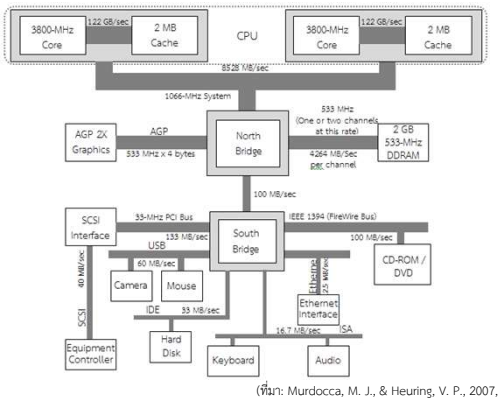
(รูป: โครงสร้างพื้นฐานของระบบบัสบนเมนบอร์ดของเครื่องไมโครคอมพิวเตอร์)



35

Bus System

(รูป: ตัวอย่างโครงสร้างระบบบัสตามสถาปัตยกรรมของบัสแบบบริดจ์)



(ที่มา: Murdocca, M. J., & Heuring, V. P., 2007, p. 310)

36

Bus System

จากภาพสไลด์ที่ 36 เป็นการแสดงการทำงานของบัสที่ใช้แนวคิดของบริดจ์ (Front Side Bus: FSB) ซึ่งจะรับข้อมูลจากส่วนประกอบต่างๆ ในระบบมาประมวลผลพร้อมกัน ดังนั้น ระบบบัสจึงต้องมีแบนด์วิดท์ขนาดใหญ่และความเร็วในการรับส่งข้อมูลสูง

จากภาพเป็นสถาปัตยกรรมบัสของไมโครโปรเซสเซอร์ แบบ 2 แกน (Core II) โดยไมโครโปรเซสเซอร์จะมีบัสที่ใช้ติดต่อกับแคช (เรียกว่า แบคไซด์บัส (Backside Bus)) ที่มีความเร็วในการรับส่งข้อมูลได้ถึง 122 GB/sec ไมโครโปรเซสเซอร์ทั้งสองตัวจะเชื่อมต่อกันด้วยบัสที่มีความกว้าง 8 ไบต์ และมีความเร็ว 1066 MHz ดังนั้นแบนด์วิดท์ของระบบบัสนี้จึงมีค่าเท่ากับ 8,528 MB/sec (8 ไบต์ * 1066 MHz)

นอร์ทบริดจ์ เป็นตัวกลางในการติดต่อสื่อสารระหว่างหน่วยความจำดีดีอาร์แรมที่สามารถรับส่งข้อมูลได้ 4,264 MB/sec กับหน่วยประมวลผลกราฟิก (Graphic Processor) ที่เชื่อมต่อบนบัส เรียกว่า เอจีพี (AGP: Advanced Graphics Port) รวมทั้งเชื่อมต่อกับเซาท์บริดจ์ที่อยู่ด้านล่าง โดยเซาท์บริดจ์จะเชื่อมต่อกับอุปกรณ์ต่อพ่วงต่างๆ ที่รับส่งข้อมูลได้ช้า

Computer Architecture

สถาปัตยกรรมคอมพิวเตอร์ คือ ทฤษฎีที่มีรูปแบบตามโมเดลของวอน นิวแมน (Von Neumann) และเป็นทฤษฎีที่ใช้ในการออกแบบโครงสร้างหน่วยประมวลผลกลาง ซึ่งประกอบด้วย รีจิสเตอร์ (Register) ที่จำเป็นและหน้าที่ของหน่วยควบคุม (Control Unit) หน่วยคำนวณและตรรกะ (ALU) และชุดของคำสั่งเครื่อง (Instruction Set)

โมเดลของวอน นิวแมน

ในช่วงต้นศตวรรษ 1950 จอห์น วอน นิวแมน (John Von Neumann) ได้เสนอแนวคิดและหลักการของสถาปัตยกรรมคอมพิวเตอร์แบบดิจิตอลที่ได้กลายเป็นจุดเริ่มต้นของโปรเซสเซอร์และการพัฒนาระบบคอมพิวเตอร์ในทุกวันนี้

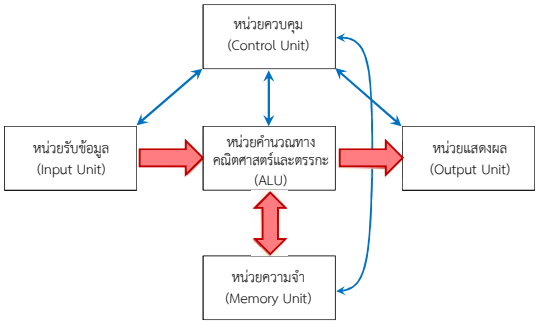
เครื่องคอมพิวเตอร์ตามแนวคิดของวอน นิวแมน ส่วนโปรแกรมและข้อมูลจะใช้หน่วยความจำเดียวกัน โดยมีโปรแกรมเคาเตอร์ (PC) ชี้คำสั่งปัจจุบันในหน่วยความจำ แล้วจะดึงคำสั่งจากหน่วยความจำมาประมวลผลเรียงลำดับไปเรื่อยๆ ไปจนกว่าจะหมดคำสั่ง

Computer Architecture

สถาปัตยกรรมตามแนวคิดของวอน นิวแมน มีองค์ประกอบ โดยสรุป ดังนี้

1. **หน่วยรับข้อมูล (Input Unit)** เป็นส่วนที่ใช้ในการรับคำสั่งหรือข้อมูลเข้าสู่ระบบ
2. **หน่วยความจำ (Memory Unit)** เป็นส่วนที่ใช้ในการเก็บคำสั่งหรือข้อมูลที่ใช้ในการประมวลผล
3. **หน่วยคำนวณและเปรียบเทียบทางตรรกะ (ALU)** เป็นส่วนที่ใช้ในการคำนวณและเปรียบเทียบทางตรรกะเพื่อการประมวลผลข้อมูล
4. **หน่วยควบคุม (Control Unit)** เป็นส่วนที่ควบคุมการทำงานของระบบ
5. **หน่วยแสดงผล (Output Unit)** เป็นส่วนแสดงผลลัพธ์ที่ได้จากการประมวลผล

Computer Architecture



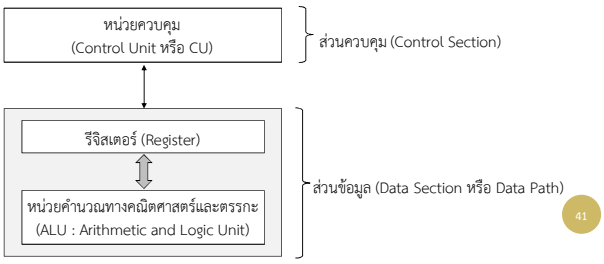
(**ที่มา**: Murdocca, M. J., & Heuring, V. P., 2007, p. 10)

Computer Architecture

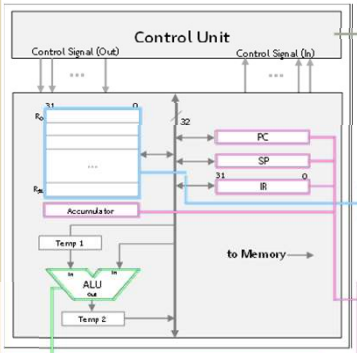
ส่วนประกอบภายในไมโครโปรเซสเซอร์

จากโมเดลของวอน นิวแมน มีหน่วยหลักที่ใช้ในการประมวลผล 2 หน่วย ที่ถูกบรรจุอยู่ในหน่วยประมวลผลกลาง คือ (หน่วยประมวลผลกลาง เรียกว่าไมโครโปรเซสเซอร์ หรือซีพียู)

- ส่วนของควบคุม (CU: Control Section)
- ส่วนของข้อมูล (Data Section หรือ Data Path)



Computer Architecture



ควบคุมการทำงานทั้งหมดภายในไมโครโปรเซสเซอร์ โดยส่งสัญญาณไปควบคุมส่วนข้อมูล ส่วนคำนวณและประมวลผล และรับสัญญาณเข้าเพื่อตอบสนองการประมวลผลเพื่อใช้ร่วมกับคำสั่งควบคุมถัดไป อีกทั้งยังมีการตรวจสอบว่าผลทำงานสมบูรณ์หรือไม่

รีจิสเตอร์ทั่วไป (General Purpose Register)

รีจิสเตอร์สำหรับหน้าที่พิเศษ (Special Purpose Register) ได้แก่
- แอคคิวมูเลเตอร์ (Accumulator: Acc)
- อินสตรักชันรีจิสเตอร์ (Instruction Register: IR)
- โปรแกรมเคาน์เตอร์ (Program Counter: PC)
- สแต็กพอย์เตอร์ (Stack Pointer: SP)

ALU มีฟังก์ชันและเครื่องหมายดำเนินการ ได้แก่
- เครื่องหมายทางการคำนวณ (Arithmetic Operations)
- เครื่องหมายทางตรรกะ (Logic Operations)

COMPUTER ARCHITECTURE

- **แอคคิวมูเลเตอร์ (Acc)** เป็นรีจิสเตอร์สำหรับเก็บผลการคำนวณ
- **อินสตรักชันรีจิสเตอร์ (IR)** เป็นรีจิสเตอร์ที่ใช้สำหรับเก็บคำสั่งที่ไมโครโปรเซสเซอร์กำลังประมวลผลอยู่ (Currently Executing Instruction)
- **โปรแกรมเคาน์เตอร์ (PC)** เป็นรีจิสเตอร์ที่ใช้สำหรับเก็บตำแหน่งที่อยู่ของคำสั่งถัดไป (Next Instruction Address) ในหน่วยความจำ ในขณะที่อินสตรักชันรีจิสเตอร์เก็บคำสั่งที่กำลังประมวลผลอยู่ หน่วยควบคุมจะกำหนดให้นำเอาค่าแอดเดรสของคำสั่งที่จะประมวลผลถัดไปมาเก็บไว้ในโปรแกรมเคาน์เตอร์เพื่อให้ทราบตำแหน่งของคำสั่งที่จะนำไปประมวลผลต่อไป
- **สแต็กพอยน์เตอร์ (SP)** เป็นรีจิสเตอร์ที่ใช้สำหรับเก็บค่าแอดเดรสที่ต้องกลับไปประมวลผล (Return Address) ซึ่งจะระบุโดยผู้เขียนโปรแกรม โดยจะใช้เมื่อมีการเรียกใช้ฟังก์ชัน (Function) ที่มีการวนรอบ (Looping) หรือกระโดด (Jump) ค่าแอดเดรสที่เก็บในสแตกพอยต์เตอร์ คือตำแหน่งในหน่วยความจำที่ผู้เขียนโปรแกรมจะใช้ในการอ้างอิงเพื่อนำคำสั่งในตำแหน่งนั้นไปประมวลผลฟังก์ชันต่อไป ซึ่งค่าแอดเดรสที่ต้องกลับไปประมวลผลจะถูกนำไปเก็บชั่วคราวในสแต็ก (Stack) โดยภายในสแต็กมีการทำงานแบบมาที่หลังออกก่อน (LIFO – Last In First Out)

COMPUTER ARCHITECTURE

การทำงานภายในไมโครโปรเซสเซอร์

การทำงานภายในไมโครโปรเซสเซอร์ในแต่ละรอบ เรียกว่า **การวนรอบเฟ็ทช แอนด์เอ็กซ์คิวท์ (Fetch-and-Execute Cycle)** ซึ่งประกอบด้วยขั้นตอนดังนี้

- การเฟ็ทช (Fetch) คือการนำเอาคำสั่งจากหน่วยความจำ หรือรีจิสเตอร์เข้าสู่บัส

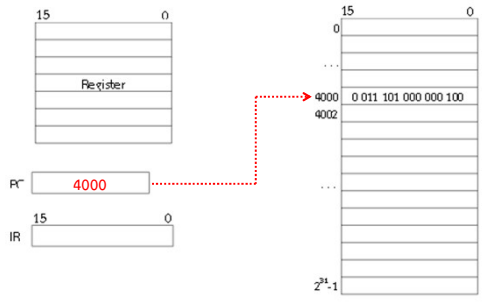
เป็นขั้นตอนที่ข้อมูลหรือคำสั่งที่อยู่ในหน่วยความจำจะถูกส่งผ่านบัสข้อมูลเข้าสู่อินสตรักชันรีจิสเตอร์ (IR) ซึ่งในขณะเดียวกันนั้นโปรแกรมเคาท์เตอร์ (PC) จะเก็บตำแหน่งของคำสั่งถัดไปที่จะประมวลผล

- การเอ็กซ์คิวท์ (Execute) มีขั้นตอนย่อยดังนี้
  - ดีโค้ดเดอร์ (Decoder) เป็นการอ่านข้อมูลหรือคำสั่งในอินสตรักชันรีจิสเตอร์ (IR) ที่ได้เฟ็ทชมา เพื่อแยกส่วนหรือระบุประเภทว่าเป็นคำสั่ง หรือข้อมูล
  - เอ็กซ์คิวท์ เป็นการประมวลผลตามคำสั่งที่ดีโค้ดได้

COMPUTER ARCHITECTURE

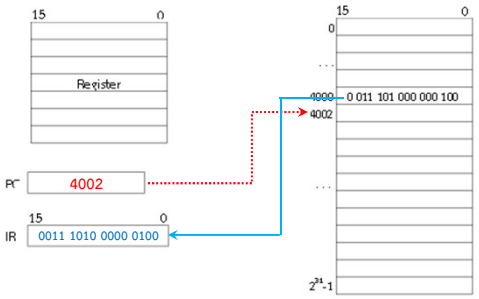
ตัวอย่าง สมมุติ ไมโครโปรเซสเซอร์รุ่นหนึ่งมีรีจิสเตอร์ ขนาด 2 Bytes ซึ่งขณะนี้มีคำสั่งที่กำลังจะประมวลผลคือคำสั่ง “0011 101 000 000 100” ซึ่งอยู่ในหน่วยความจำตำแหน่งที่ 4000

ขั้นที่ 1 โปรแกรมเคาเตอร์ (PC) เก็บหมายเลขแอดเดรสของคำสั่งที่กำลังจะประมวลผล



COMPUTER ARCHITECTURE

ขั้นที่ 2 คอนโทรยูนิตเฟ็ทช์คำสั่งในตำแหน่งที่ 4000 เข้าสู่อินสตรักชันรีจิสเตอร์ (IR)



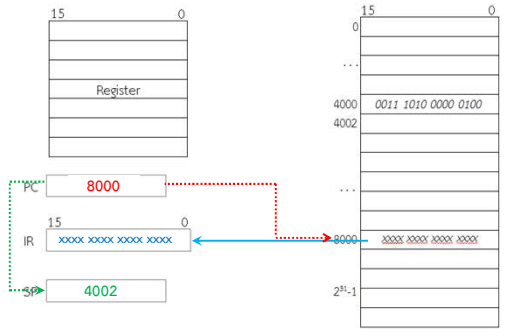
ขั้นที่ 3 ดีโค้ด และเอ็กซ์คิวท์คำสั่ง

ขั้นที่ 4 ไมโครโปรเซสเซอร์เพิ่มค่าในโปรแกรมเคาท์เตอร์ให้ชี้ไปยังตำแหน่ง 4002 (PC=4002) ซึ่งเป็นคำสั่งถัดไปที่จะต้องประมวลผลต่อไป

COMPUTER ARCHITECTURE

กรณีที่มีการขัดจังหวะการทำงาน (Interrupt) ซึ่งมีการเรียกใช้ฟังก์ชันในหน่วยความจำตำแหน่งอื่น ไมโครโปรเซสเซอร์จะต้องทำการเก็บตำแหน่งของฟังก์ชันเดิมที่ประมวลผลค้างไว้ใน SP

สมมุติให้ในขณะที่โปรเซสเซอร์กำลังประมวลผลคำสั่งในตำแหน่งที่ 4000 (PC=4002) มีการขัดจังหวะให้ไปดึงคำสั่งในหน่วยความจำตำแหน่งที่ 8000 มาประมวลผลทันที



COMPUTER ARCHITECTURE

ชุดคำสั่ง

ชุดคำสั่ง คือ ชุดของคำสั่งที่โปรเซสเซอร์กำลังประมวลผลเพื่อดำเนินการตามที่ผู้เขียนโปรแกรมได้สร้างไว้ เรียกว่า คำสั่งเครื่อง (Machine Language) หรือคำสั่งคอมพิวเตอร์ (Computer Instruction) ซึ่งในแต่ละชุดคำสั่งจะประกอบด้วยคำสั่งย่อยๆ หลายคำสั่ง

โดยไมโครโปรเซสเซอร์จะมีคำสั่งในการนำข้อมูลจากรีจิสเตอร์มาไว้ในหน่วยความจำ แล้วไมโครโปรเซสเซอร์จะเรียกใช้คำสั่งสำหรับการคำนวณนั้นมาประมวลผล และเก็บผลลัพธ์ที่ได้ไปเก็บไว้ในหน่วยความจำ

ชุดคำสั่งประกอบด้วยคำสั่งย่อยที่โปรเซสเซอร์จะนำไปประมวลผลเพื่อให้ครบตามการวนรอบเฟ็ทช์แอนด์เอ็กซ์คิวท์ ซึ่งในแต่ละชุดคำสั่งจะประกอบด้วยคำสั่งเครื่องดังนี้

- รหัสการดำเนินการ (Operation Code : Op-code)
- ส่วนอ้างอิงต้นทางของตัวดำเนินการ (Source Operand Reference)
- ส่วนอ้างอิงผลลัพธ์ของตัวดำเนินการ (Result Operand Reference)
- ส่วนอ้างอิงชุดคำสั่งถัดไป (Next Instruction Reference)

COMPUTER ARCHITECTURE

ประเภทของชุดคำสั่ง

ชุดคำสั่งที่ไมโครโปรเซสเซอร์จะนำไปประมวลผล แบ่งออกเป็น 4 ประเภท ดังนี้

- **ชุดคำสั่งทางคณิตศาสตร์และตรรกะ (Data Processing)** เป็นชุดคำสั่งที่ประกอบด้วยคำสั่งย่อยๆ ที่ต้องใช้ตัวดำเนินการทางคณิตศาสตร์และตรรกะ
- **ชุดคำสั่งเพื่อการจัดการหน่วยความจำ (Data Storage)** เป็นชุดคำสั่งที่กำหนดว่าจะนำคำสั่งหรือข้อมูลมาจากหน่วยความจำตำแหน่งใด และนำผลลัพธ์ที่ได้ไปเก็บในตำแหน่งใดของหน่วยความจำ
- **ชุดคำสั่งสำหรับรับและแสดงผล (Data Movement)** เป็นชุดคำสั่งที่ไมโครโปรเซสเซอร์จะประมวลผลว่าข้อมูลมาจากอุปกรณ์ใดในรูปแบบใด และจะนำออกไปยังอุปกรณ์ใด
- **ชุดคำสั่งเพื่อตรวจสอบเงื่อนไขและกระโดดการทำงาน (Control)** เป็นชุดคำสั่งที่จะใช้ตัวดำเนินการทางตรรกะร่วมเพื่อตรวจสอบว่าจะต้องกระโดดไปประมวลผลยังชุดคำสั่งใด

49

COMPUTER ARCHITECTURE

X = A x B + C x C

จำนวนแอดเดรสของชุดคำสั่ง

➢ **แบบ 3-แอดเดรส (Three-Address)**

คำสั่ง	สิ่งที่กระทำ
MULTIPLY T, A, B	T := A x B
MULTIPLY X, C, C	X := C x C
ADD X, X, T	X := X + T

➢ **แบบ 2-แอดเดรส (Two-Address)**

คำสั่ง	สิ่งที่กระทำ
MOVE T, A	T := A
MULTIPLY T, B	T := T x B
MOVE X, C	X := C
MULTIPLY X, C	X := X x C
ADD X, T	X := X + T

50

ที่มา (Hayes, J. P., 1998, p. 192)

COMPUTER ARCHITECTURE

X = A x B + C x C

➢ **แบบ 1-แอดเดรส (One-Address)**

คำสั่ง	สิ่งที่กระทำ
LOAD A	นำค่าในตำแหน่ง A ไปไว้ในแอ็คคิวมูเลเตอร์ (AC)
MULTIPLY B	AC := AC x B
STORE T	นำค่าในแอ็คคิวมูเลเตอร์ไปเก็บไว้ในหน่วยความจำตำแหน่ง T
LOAD C	นำค่าในตำแหน่ง C ไปไว้ในแอ็คคิวมูเลเตอร์ (AC)
MULTIPLY C	AC := AC x C
ADD T	AC := AC + T
STORE X	นำค่าในแอ็คคิวมูเลเตอร์ไปเก็บไว้ในหน่วยความจำตำแหน่ง X

51

ที่มา (Hayes, J. P., 1998, p. 192)

COMPUTER ARCHITECTURE

X = A x B + C x C

➢ **แบบ 0-แอดเดรส (Zero-Address)**

คำสั่ง	สิ่งที่กระทำ
PUSH A	ย้ายค่าในตำแหน่ง A ไปไว้บนสุดของสแต็ก
PUSH B	ย้ายค่าในตำแหน่ง B ไปไว้บนสุดของสแต็ก
MULTIPLY	นำค่า A และ B ออกจากสแต็ก และแทนที่ด้วยค่า AxB
PUSH C	ย้ายค่าในตำแหน่ง C ไปไว้บนสุดของสแต็ก
PUSH C	คัดลอกค่าในตำแหน่ง C ไปไว้บนสุดของสแต็กอีกครั้ง
MULTIPLY	นำค่า C ทั้งสองค่าออกจากสแต็ก และแทนที่ด้วยค่า CxC
ADD	นำค่า CxC และ AxB ที่อยู่บนสุดของสแต็กมาบวกกัน แล้วแทนที่ด้วยค่า (CxC)+(AxB)
POP X	นำค่าบนสุดของสแต็กไปไว้ในหน่วยความจำตำแหน่ง X

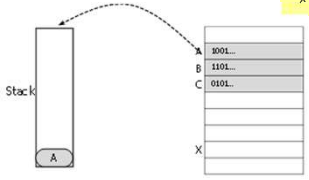
52

ที่มา (Hayes, J. P., 1998, p. 193)

COMPUTER ARCHITECTURE

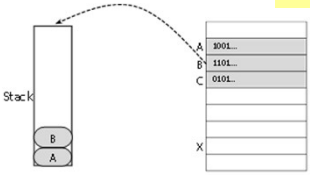
X = A x B + C x C

PUSH A



X = A x B + C x C

PUSH B

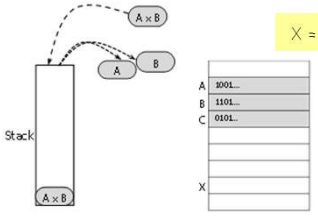


53

COMPUTER ARCHITECTURE

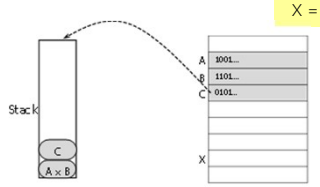
X = A x B + C x C

MULTIPLY



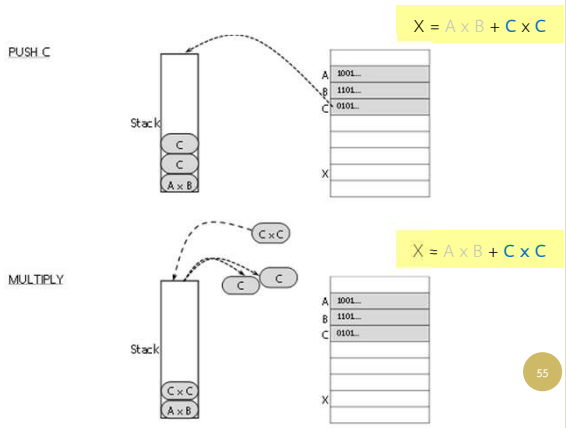
X = A x B + C x C

PUSH C



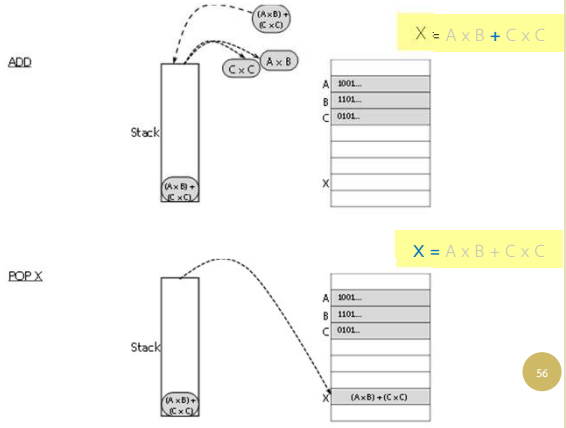
54

Computer Architecture



55

Computer Architecture



56

Microprocessor Architecture

สถาปัตยกรรมของไมโครโปรเซสเซอร์ที่ได้รับความนิยม แบ่งออกเป็น 2 กลุ่มใหญ่ๆ คือ สถาปัตยกรรมซีไอเอสซี (CISC) และสถาปัตยกรรมอาร์ไอเอสซี (RISC)

สถาปัตยกรรมซีไอเอสซี (CISC: Complex Instruction Set Computers)

เป็นสถาปัตยกรรมของโปรเซสเซอร์รุ่นเก่า เช่น สถาปัตยกรรมของเมนเฟรมไอบีเอ็ม (IBM Mainfraim) ที่มีชื่อว่า ซีซีรี่ (zSeries) และสถาปัตยกรรมของอินเทลเอ็กซ์ (Intel X86)

- การใช้หน่วยความจำ

ชุดคำสั่งของสถาปัตยกรรมแบบซีไอเอสซีมีจำนวนมาก หลากหลายรูปแบบและมีความซับซ้อน ดังนั้นจำนวนบิตที่ใช้จึงแตกต่างกันตามความซับซ้อน นั่นคือหากคำสั่งมีความซับซ้อนมากจำนวนบิตก็จะมากตามไปด้วย แต่หากชุดคำสั่งนั้นมีความซับซ้อนน้อยจำนวนบิตก็จะใช้น้อย ในการเก็บชุดคำสั่งของสถาปัตยกรรมแบบซีไอเอสซีจะเก็บเท่ากับจำนวนจริงของการใช้งานจึงทำให้ประหยัดเนื้อหาในหน่วยความจำ แต่ไม่ได้ช่วยให้การประมวลผลเร็วเพราะต้องใช้เวลาในการถอดรหัสที่มีความยาวไม่เท่ากันทำให้เกิดความยุ่งยากในการถอดรหัส

57

Microprocessor Architecture

- ประสิทธิภาพ
  - สถาปัตยกรรมแบบซีไอเอสซีมีชุดคำสั่งที่ซับซ้อนและมีชุดคำสั่งพิเศษ เช่นการทำโพลิโนเมียล (Polynomial) ดังนั้นในการประมวลผลคำสั่งหนึ่งๆ จะใช้จำนวนของวงรอบสัญญาณนาฬิกามากกว่าสถาปัตยกรรมแบบอาร์ไอเอสซี
  - ด้วยชุดคำสั่งของสถาปัตยกรรมแบบซีไอเอสซีมีความยาวไม่เท่ากัน ทำให้ต้องเสียเวลาในการถอดรหัส อีกทั้งมีวงจรที่สลับซับซ้อน และใช้วงรอบสัญญาณนาฬิกามากทำให้ต้องเสียค่าใช้จ่ายสูงและใช้เวลาในการประมวลผลนาน
- การสนับสนุนคอมไพเลอร์ (Compiler)

ภายในโปรเซสเซอร์ของสถาปัตยกรรมแบบซีไอเอสซีจะบรรจุชุดคำสั่งที่ซับซ้อน แต่เมื่อผู้เขียนโปรแกรมได้เขียนโปรแกรมแล้วตัวคอมไพเลอร์ที่อยู่ในไมโครโปรเซสเซอร์ (ทำหน้าที่แปลคำสั่งในชุดโปรแกรมให้เป็นภาษาเครื่อง) จะทำการแปลให้เป็นภาษาเครื่องได้ง่าย โดยไม่ต้องใช้ชุดคำสั่งที่มีในโปรเซสเซอร์ หรือในบางโปรเซสเซอร์มีคอมไพเลอร์ในไมโครโปรเซสเซอร์สำหรับชุดคำสั่งที่ซับซ้อนซึ่งไม่เหมาะกับการเขียนโปรแกรม นั่นคือระบบฮาร์ดแวร์กับซอฟต์แวร์ไม่สนับสนุนซึ่งกันและกัน

58

Microprocessor Architecture

สถาปัตยกรรมอาร์ไอเอสซี (RISC: Reduced Instruction Set Computers)

เป็นสถาปัตยกรรมที่ถูกออกแบบและพัฒนาเพื่อแก้ไขปัญหาของสถาปัตยกรรมซีไอเอสซี สถาปัตยกรรม อาร์ไอเอสซียังคงถูกใช้เป็นสถาปัตยกรรมของโปรเซสเซอร์จนถึงปัจจุบัน

- การใช้หน่วยความจำ

สถาปัตยกรรมอาร์ไอเอสซีเน้นหลักการนำเอาชุดคำสั่งง่ายๆ ไม่กี่คำสั่ง อาทิ การบวก การลบ การคูณ การหาร การโหลดข้อมูลเก็บไว้ในรีจิสเตอร์โดยตรง (LOAD) และการเก็บค่าไว้ในหน่วยความจำ (STORE) เป็นต้น ซึ่งมีไม่เกิน 128 คำสั่ง นั่นคือใช้หน่วยความจำในการเก็บชุดคำสั่งเพียง 6 บิต (2^6 = 128) ซึ่งเรียกการเก็บชุดคำสั่งที่จำกัดขนาดของบิตเช่นนี้ ว่า ฟิกซ์โค้ด (Fix Code) การเก็บชุดคำสั่งแบบนี้มีข้อเสียคือ หากคำสั่งนั้นใช้จำนวนบิตเพียง 1 บิต การเก็บข้อมูลก็ยังคงต้องใช้บิตจำนวน 6 บิตเช่นเดิม ซึ่งทำให้เกิดการสูญเสียหน่วยความจำ แต่การเก็บข้อมูลแบบฟิกซ์โค้ดนี้ มีข้อดีคือ การถอดรหัสจะทำได้อย่างรวดเร็วเพราะชุดคำสั่งมีขนาดเท่ากัน

59

Microprocessor Architecture

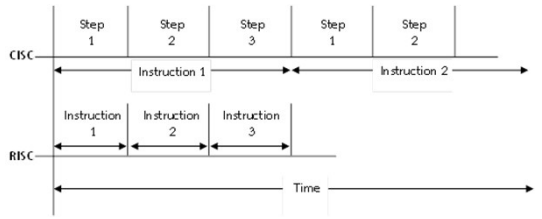
- ประสิทธิภาพ
  - สถาปัตยกรรมอาร์ไอเอสซีสามารถประมวลผลได้ความเร็วกว่าสถาปัตยกรรมซีไอเอสซี เพราะสถาปัตยกรรมอาร์ไอเอสซีจะประกอบด้วยคำสั่งง่ายๆ
  - การถอดรหัสชุดคำสั่งของสถาปัตยกรรมอาร์ไอเอสซีทำได้ง่ายกว่าสถาปัตยกรรมซีไอเอสซี เพราะมีลักษณะของชุดคำสั่งแบบฟิกซ์โค้ด
  - สถาปัตยกรรมอาร์ไอเอสซีมีรีจิสเตอร์จำนวนมากที่รองรับการเก็บข้อมูลระหว่างประมวลผล ทำให้การทำงานโดยรวมเป็นไปได้อย่างรวดเร็ว
  - การใช้งานคำสั่งง่ายๆ ของสถาปัตยกรรมอาร์ไอเอสซี บางคำสั่งจะใช้วงรอบสัญญาณนาฬิกาไม่ถึง 1 สัญญาณนาฬิกา ทำให้ประมวลผลได้เร็ว
- การสนับสนุนคอมไพเลอร์ (Compiler)

สถาปัตยกรรมอาร์ไอเอสซีมีการบรรจุคำสั่งประมาณ 128 คำสั่ง และอนุญาตให้ใช้งานคำสั่งประเภทโหลดและเก็บข้อมูล (LOAD/STORE) ที่ทำงานกับรีจิสเตอร์ได้โดยตรง ซึ่งทำให้การทำงานโดยรวมเร็วกว่าสถาปัตยกรรมซีไอเอสซี

60

MICROPROCESSOR ARCHITECTURE

ในแต่ละคำสั่งของสถาปัตยกรรมซีไอเอสซีจะมีขั้นตอนย่อยๆ ประกอบอยู่ ซึ่งขั้นตอนย่อยๆ นี้จะมีไม่เท่ากันขึ้นอยู่กับแต่ละคำสั่งว่าจะมี 2 ขั้นตอน 3 ขั้นตอน หรือมีมากกว่า ในขณะที่สถาปัตยกรรมอาร์ไอเอสซีเป็นการลดคำสั่งให้แต่ละคำสั่งมีเพียงหนึ่งขั้นตอนต่อหนึ่งชุดคำสั่ง



MULTIPROCESSOR ARCHITECTURE

สถาปัตยกรรมคอมพิวเตอร์แบบมัลติโปรเซสเซอร์ เป็นสถาปัตยกรรมในการออกแบบระบบคอมพิวเตอร์ที่มีโปรเซสเซอร์ที่ทำงานร่วมกันมากกว่าหนึ่งโปรเซสเซอร์ ซึ่งจะจำเป็นจะต้องใช้ทรัพยากรของระบบ อาทิ หน่วยความจำ และบัสร่วมกัน หากการออกแบบสถาปัตยกรรมออกมาไม่เหมาะสมจะทำให้เกิดความขัดแย้งในการใช้ทรัพยากรซึ่งจะส่งผลให้ระบบคอมพิวเตอร์ทำงานได้อย่างไม่มีประสิทธิภาพ

องค์ประกอบพื้นฐานในการสร้างระบบมัลติโปรเซสเซอร์

- ส่วนประมวลผล (Processing Element) หรือพีอี (PE)
- ส่วนสวิตช์ (Switch)
- ส่วนเส้นทางเชื่อม (Interconnection Path) หรือเรียกว่า เส้นทางการส่งผ่านข้อมูล (Transmission Line)

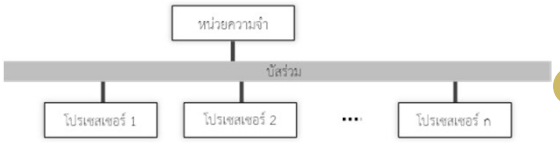
MULTIPROCESSOR ARCHITECTURE

รูปแบบการต่อโปรเซสเซอร์ให้เป็นมัลติโปรเซสเซอร์

1. รูปแบบบัสร่วม (Common Buses)
2. รูปแบบหน่วยความจำหลายพอร์ต (Multiport Memory)
3. รูปแบบการเชื่อมต่อบัสผ่านส่วนรับและแสดงผล (Connect Through I/O
4. รูปแบบบัสวินโดว์ (Bus Windows)

1. รูปแบบบัสร่วม (Common Buses)

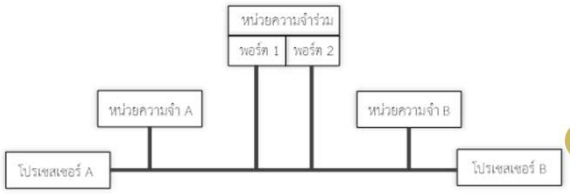
การต่อบัสหรือเส้นทางร่วมเป็นรูปแบบดั้งเดิมที่นิยมใช้ในการต่อโปรเซสเซอร์หลายๆ ตัวเข้าด้วยกันเพราะจะต้องเสียค่าใช้จ่ายน้อยที่สุด แต่ความน่าเชื่อถือและความทนทานต่อความผิดพลาดโดยรวมของระบบไม่สูงมาก เนื่องจากหากบัสที่ใช้ร่วมกันเกิดเสียหายจะทำให้การติดต่อระหว่างโปรเซสเซอร์ขาดการติดต่อทันที และจะทำให้เกิดความเสียหายทั้งระบบได้



MULTIPROCESSOR ARCHITECTURE

2. รูปแบบหน่วยความจำหลายพอร์ต (Multiport Memory)

หน่วยความจำหลายพอร์ต หรือหน่วยความจำที่มีหลายช่องทางจะช่วยให้โปรเซสเซอร์มากกว่าหนึ่งโปรเซสเซอร์สามารถเข้าถึงข้อมูลได้ในเวลาเดียวกันเพื่อการอ่านหรือเขียนข้อมูลลงในหน่วยความจำร่วมพร้อมกันได้ หลักสำคัญคือหน่วยความจำในระบบจะต้องมีความเร็วในการเข้าถึงสูงจึงทำให้มีราคาแพง ดังนั้นเพื่อไม่ให้ค่าใช้จ่ายสูงเกินจำเป็น จึงใช้หน่วยความจำพอร์ตเดียว ที่ขึ้นอยู่กับโปรเซสเซอร์ใดโปรเซสเซอร์หนึ่ง ร่วมกับหน่วยความจำหลายพอร์ต รูปแบบหน่วยความจำที่มีหลายพอร์ต

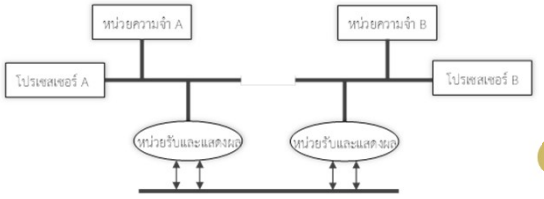


MULTIPROCESSOR ARCHITECTURE

3. รูปแบบการเชื่อมต่อบัสผ่านส่วนรับและแสดงผล (Connect Through I/O)

คอมพิวเตอร์รูปแบบการต่อโปรเซสเซอร์ให้เป็นมัลติโปรเซสเซอร์โดยผ่านส่วนรับและแสดงผลจะใช้ในคอมพิวเตอร์ยุคแรกๆ ที่ยังไม่มีระบบเครือข่าย โดยแต่ละโปรเซสเซอร์จะมีการเชื่อมต่อกับหน่วยความจำ หน่วยรับและแสดงผลที่มีความเป็นอิสระต่อกัน (ดังภาพ)

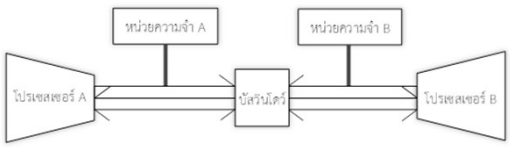
รูปแบบการเชื่อมต่อแบบนี้จะอยู่ในระบบโปรเซสเซอร์ขนาดใหญ่และมีความเร็วสูง ซึ่งส่วนรับและแสดงผลจะมีโปรเซสเซอร์เล็กๆ บรรจุอยู่เพื่อเพิ่มความสามารถในการทำงาน



MULTIPROCESSOR ARCHITECTURE

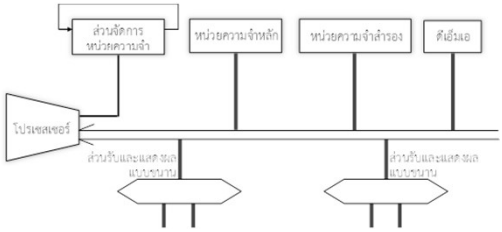
4. รูปแบบบัสวินโดว์ (Bus Windows)

การต่อโปรเซสเซอร์ให้เป็นมัลติโปรเซสเซอร์แบบบัสวินโดว์ เป็นรูปแบบที่โปรเซสเซอร์แต่ละตัวจะสามารถใช้หน่วยความจำของตนเองรวมทั้งโปรเซสเซอร์อื่นได้เสมือนว่าเป็นหน่วยความจำของตัวเองโดยผ่านบัสวินโดว์



MULTIPROCESSOR ARCHITECTURE

บัสวินโดว์ คือ ดีเอ็มเอ (DMA: Direct Access Memory ซึ่งเป็นการเข้าถึงหน่วยความจำได้โดยตรง โดยดีเอ็มเอจะทำหน้าที่แทนโปรเซสเซอร์ในกรณีที่มีการถ่ายโอนข้อมูลไปมาระหว่างอุปกรณ์ต่อพ่วงต่างๆ ในระบบไปยังหน่วยความจำอื่นๆ โดยไม่ต้องใช้โปรแกรมใดๆ ช่วย



MULTIPROCESSOR ARCHITECTURE

- บัสวินโดว์แบบดีเอ็มเอ ถูกออกแบบขึ้นเพื่อเพิ่มความน่าเชื่อถือได้ของระบบ ตอบสนองที่รวดเร็ว และมีอัตราการเคลื่อนย้ายข้อมูลที่สูงขึ้น
- ดังนั้นระบบไมโครโปรเซสเซอร์รุ่นใหม่จึงได้เพิ่มเติมดีเอ็มเอเข้าไปในระบบเพื่อควบคุมบัสในการถ่ายโอนข้อมูลไปมาระหว่างหน่วยรับและแสดงข้อมูล และหน่วยความจำ โดยไม่ต้องผ่านโปรเซสเซอร์
- ในการออกแบบการเชื่อมต่อเช่นนี้จึงเป็นการแบ่งเบาภาระของโปรเซสเซอร์ ทำให้ภาพรวมของระบบมีความเร็วมากขึ้น
- โดยในระบบไมโครโปรเซสเซอร์ที่มีสมรรถนะสูงจะใช้หน่วยความจำในปริมาณมาก ได้แก่ หน่วยความจำหลัก หน่วยความจำแบบแฟลช และหน่วยความจำสำรอง ดังนั้นในระบบจึงจำเป็นจะต้องมีส่วนการจัดการหน่วยความจำ (Memory Management Unit) เข้ามาประกอบด้วยเพื่อจัดการการเข้าถึงหน่วยความจำในแต่ละแบบตามความเหมาะสมและความต้องการของระบบ

MULTIPROCESSOR ARCHITECTURE

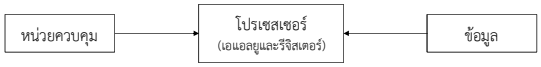
รูปแบบการประมวลผล

1. คำสั่งเดี่ยวและข้อมูลเดี่ยว (SISD)
   [Single Instruction Single Data Stream]
2. คำสั่งเดี่ยวและหลายชุดข้อมูล (SIMD)
   [Single Instruction Multiple Data Stream]
3. หลายชุดคำสั่งและข้อมูลเดี่ยว (MISD)
   [Multiple Instruction Single Data Stream]
4. หลายชุดคำสั่งและหลายชุดข้อมูล (MIMD)
   [Multiple Instruction Multiple Data Stream]

MULTIPROCESSOR ARCHITECTURE

1. คำสั่งเดี่ยวและข้อมูลเดี่ยว (SISD)

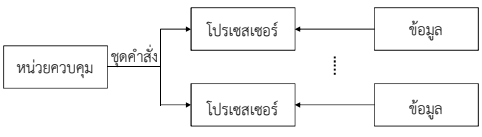
เป็นรูปแบบโปรเซสเซอร์ที่ใช้การประมวลผลด้วยชุดข้อมูลเพียงชุดเดียว และทำงานด้วยคำสั่งเดียวภายในหนึ่งสัญญาณนาฬิกา สถาปัตยกรรมการประมวลผลแบบคำสั่งเดี่ยวและข้อมูลเดี่ยวนี้ เป็นกระบวนการทำงานที่อยู่ในแบบอนุกรม (ยังไม่มีการทำงานแบบขนาน) โดยมีโปรเซสเซอร์เพียงตัวเดียว และยังคงรูปแบบแนวคิดของวอน นิวแมน ลักษณะการทำงานในแต่ละจังหวะสัญญาณนาฬิกา จะเป็นการประมวลผลของชุดคำสั่งเดียวต่อชุดข้อมูลเดียว



MULTIPROCESSOR ARCHITECTURE

2. คำสั่งเดี่ยวและหลายชุดข้อมูล (SIMD)

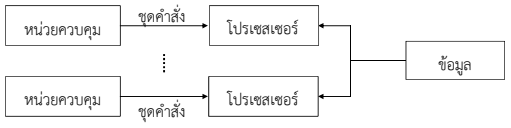
เป็นรูปแบบโปรเซสเซอร์ที่ใช้การประมวลผลด้วยชุดข้อมูลหลายชุด แต่ทำงานด้วยคำสั่งเดียวภายในหนึ่งสัญญาณนาฬิกาและได้ผลลัพธ์หลายชุด ซึ่งรูปแบบการประมวลผลนี้ถูกใช้ในโปรเซสเซอร์เพนเทียมเอ็มเอ็มเอ็กซ์ (Pentium MMX) ซึ่งรูปแบบการประมวลผลแบบคำสั่งเดี่ยวและหลายชุดข้อมูลยังคงเป็นสถาปัตยกรรมตามแนวคิดของวอน นิวแมน แต่มีขีดความสามารถในการคำนวณมากขึ้น โดยการประมวลผลคำสั่งหนึ่งครั้งจะสามารถประมวลผลกับชุดข้อมูลหลายๆ ชุดพร้อมๆ กันได้ เหมือนเป็นเว็กเตอร์โปรเซสเซอร์ (Vector Processor)



MULTIPROCESSOR ARCHITECTURE

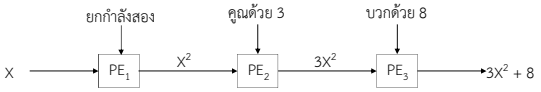
3. หลายชุดคำสั่งและข้อมูลเดี่ยว (MISD)

เป็นรูปแบบโปรเซสเซอร์ที่ใช้การประมวลผลด้วยชุดข้อมูลเพียงชุดเดียว แต่ทำงานหลายคำสั่งภายในหนึ่งสัญญาณนาฬิกา ซึ่งระบบการประมวลผลแบบหลายชุดคำสั่งและข้อมูลเดี่ยว จะมีโปรเซสเซอร์หลายโปรเซสเซอร์ โดยโปรเซสเซอร์แต่ละตัวจะรับคำสั่งจากส่วนควบคุมให้ดำเนินการทางคณิตศาสตร์หรือตรรกศาสตร์อย่างใดอย่างหนึ่งกับข้อมูลหนึ่งชุด (1 Data) และคำสั่งที่โปรเซสเซอร์ได้รับนั้นสามารถแตกต่างกันได้ (M Instruction)



Multiprocessor Architecture

รูปแบบการประมวลผลแบบหลายชุดคำสั่งและข้อมูลเดี่ยวนี้ โปรเซสเซอร์จะใช้ข้อมูลชุดเดียวกัน เมื่อโปรเซสเซอร์ตัวแรกทำงานเสร็จจะส่งผลลัพธ์ที่ได้ไปให้โปรเซสเซอร์ตัวถัดไปทำงานต่อ ดังตัวอย่างการทำงานของฟังก์ชันคณิตศาสตร์ $f(X) = 3X^2 + 8$ จะใช้โปรเซสเซอร์จำนวน 3 ตัว



PE หมายถึงหน่วยประมวลผล ซึ่งจะเป็นการรวมโปรเซสเซอร์กับหน่วยความจำเข้าด้วยกัน โดยในการคำนวณ $f(X) = 3X^2+8$ เริ่มจากหน่วยประมวลผลที่ 1 (PE_1) จะยกกำลังสองให้กับข้อมูลที่นำเข้ามา (ซึ่งคือค่า X) จากนั้นจะส่งผลที่ได้ไปให้หน่วยประมวลผลที่ 2 โดยนำผลลัพธ์ที่ได้จากหน่วยประมวลผลที่ 1 คูณด้วย 3 ซึ่งผลลัพธ์ที่ได้จากโปรเซสเซอร์ที่ 2 คือ $X^2 \times 3$ แล้วจะส่งผลลัพธ์ที่ได้ไปบวกกับ 8 ในหน่วยประมวลผลที่ 3

Multiprocessor Architecture

4. หลายชุดคำสั่งและหลายชุดข้อมูล (MIMD)

เป็นรูปแบบโปรเซสเซอร์ที่ใช้การประมวลผลด้วยชุดข้อมูลหลายชุด และทำงานด้วยได้หลายคำสั่งภายในหนึ่งสัญญาณนาฬิกา ซึ่งระบบการประมวลผลแบบหลายชุดคำสั่งและหลายชุดข้อมูลนี้เป็นระบบการประมวลผลแบบขนานอย่างสมบูรณ์ คือ โปรเซสเซอร์แต่ละตัวสามารถทำงานได้อย่างเป็นอิสระกับชุดคำสั่งที่ส่งมาจากส่วนควบคุมโดยตรง (M Instruction) ในขณะเดียวกันแต่ละโปรเซสเซอร์ก็ประมวลผลข้อมูลคนละชุดกัน (M Data)

Parallel Computer Architecture

สถาปัตยกรรมแบบไปป์ไลน์ (Pipeline Architecture)

เป็นเทคนิคในการประมวลผลคำสั่งหลายๆ คำสั่งแบบคาบเกี่ยวกัน (Overlap) เพื่อเพิ่มประสิทธิภาพให้กับหน่วยประมวลผลกลาง โดยแบ่งคำสั่งออกเป็นขั้นตอนย่อยๆ โดยในขณะที่หน่วยประมวลผลได้ประมวลผลขั้นตอนย่อยของคำสั่งหนึ่ง ไมโครโปรเซสเซอร์จะสามารถประมวลผลขั้นตอนย่อยของอีกคำสั่งหนึ่งซึ่งไม่ใช่ขั้นตอนเดียวกัน

การทำงานแบบไปป์ไลน์เป็นการทำงานที่ไม่มีการเว้นช่วงเวลาร่อ นั่นคือเมื่อทำงานเสร็จแล้วจะทำส่วนอื่นต่อทันที โดยในขณะเดียวกันคำสั่งถัดไปจะประมวลผลส่วนถัดมาตามลำดับ ดังนั้นจึงไม่มีการว่างงานเกิดขึ้นในสัญญาณนาฬิกา

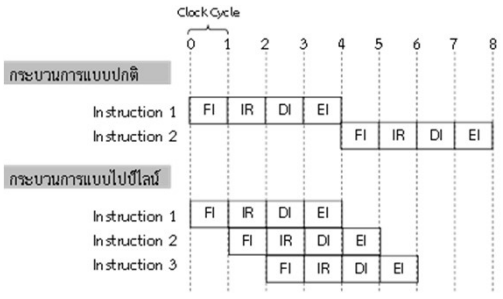
Parallel Computer Architecture

ขั้นตอนการประมวลผลในแต่ละชุดคำสั่งจะถูกแบ่งออกเป็น 5 ขั้นตอนย่อย:

1. ส่วนรับคำสั่ง (IF: Instruction Fetch) ทำหน้าที่ดึงคำสั่งทั้งจากหน่วยความจำหลักหรือจากในหน่วยความจำของคำสั่ง (Instruction Cache) เพื่อส่งต่อให้ส่วนต่อไปดำเนินการ
2. ส่วนแปลงคำสั่ง (ID: Instruction Decode) ทำหน้าที่แยกแยะคำสั่งต่างๆ ของสถาปัตยกรรมแบบซีไอเอสซี ซึ่งในแต่ละคำสั่งจะมีขนาดที่ไม่แน่นอน ตรงส่วนนี้ก็จะทำการแบ่งคำสั่งนั้นเป็นคำสั่งย่อยๆ ให้มีความยาวเท่าๆ กัน เช่นเดียวกับสถาปัตยกรรมแบบอาร์ไอเอสซี โดยเรียกคำสั่งย่อยๆ นี้ว่าไมโครโอเปอร์เรชั่น (Micro Operation)
3. ส่วนรับข้อมูล (IS: Instruction Issue) จะรับข้อมูลที่จะใช้ในการประมวลผลเข้ามาเก็บไว้
4. ส่วนประมวลผล (EX: Execute) จะประมวลผลตามคำสั่งและ ตัวดำเนินการ (Operand) ที่ได้รับมาจากส่วนที่ 2 และ 3 โดยหากในส่วนที่ 2 เป็นการถอดรหัสว่าเป็นการบวก ส่วนที่ 3 รับค่าที่จะบวก ส่วนนี้ (ส่วนที่ 4) ก็จะทำการบวกให้ได้ผลลัพธ์ออกมา
5. ส่วนการเขียนข้อมูลกลับ (WB: Write Back) เมื่อทำการประมวลผลเสร็จสิ้น ผลลัพธ์ที่ได้ก็จะนำไปเก็บไว้ในรีจิสเตอร์ หรือในดาต้าแคช (Data Cache)

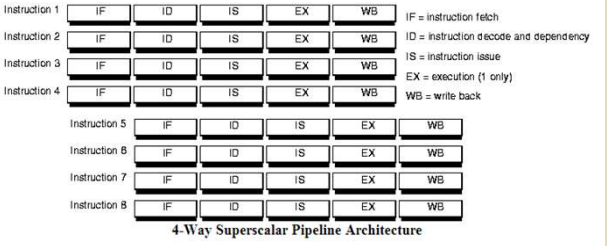
Parallel Computer Architecture

(รูป: โครงสร้างแสดงการประมวลผลแบบไปป์ไลน์ที่มีการแบ่งกระบวนการมากขึ้นเพื่อเพิ่มความเร็วในการประมวลผล)



Parallel Computer Architecture

(รูป: โครงสร้างแสดงการทำงานของไปป์ไลน์ตามทฤษฎี)

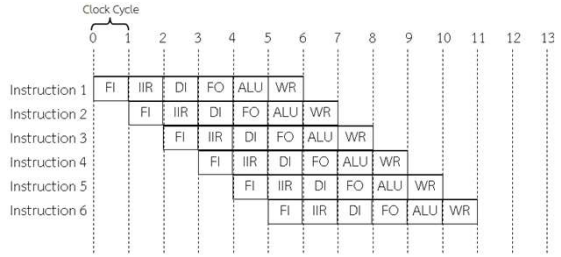


4-Way Superscalar Pipeline Architecture

ที่มา (MIPS Technologies, 1996)

Parallel Computer Architecture

จากการทำงานข้างต้น ได้มีแนวทางพัฒนาเพื่อเพิ่มประสิทธิภาพในการประมวลผลให้เร็วขึ้น โดยการแบ่งกระบวนการเดิมออกเป็นส่วนที่ย่อยกว่าเดิม ซึ่งจะทำให้สามารถดึงกระบวนการย่อยของคำสั่งถัดไปได้เร็วขึ้น ส่งผลให้มีการประมวลผลได้เร็ว เช่น แบ่งกระบวนการรวนรอบเฟ็ทช์แอนด์เอ็กซ์คิวท์ ออกเป็นส่วนย่อยๆ ซึ่งจะได้รูปแบบลำดับการประมวลผลด้วยสถาปัตยกรรมแบบไปป์ไลน์ ดังนี้



Parallel Computer Architecture

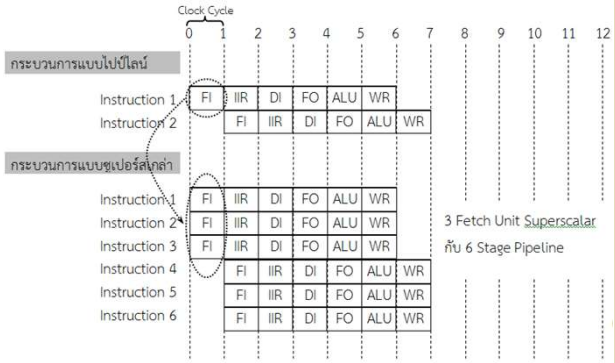
สถาปัตยกรรมแบบซูเปอร์สเกล่า (Superscalar Architecture)

สถาปัตยกรรมซูเปอร์สเกล่า เป็นการคำนวณแบบขนานที่ใช้ในการประมวลผลแบบอาร์ไอเอสซีหลายๆ ตัว ซึ่งจัดการกับคำสั่งแบบไปป์ไลน์ (Instruction Pipeline) ที่จะไปประมวลผล (Execute) หลายๆ คำสั่งในเวลาเดียวกันระหว่างรอบสัญญาณนาฬิกา

สถาปัตยกรรมซูเปอร์สเกล่า มีการทำงานพร้อมๆ กันหลายๆ คำสั่ง ซึ่งเป็นการนำเอาไปป์ไลน์หลายๆ ไปป์ไลน์มาทำงานขนานกันไป

Parallel Computer Architecture

(รูป: โครงสร้างเปรียบเทียบการประมวลผลแบบไปป์ไลน์ กับซูเปอร์สเกล่า)



References

หนังสือ/ตำราอ้างอิง

ธีรวัฒน์ ประกอบผล, และจันทนา ผ่องเพ็ญศรี. (2551). *สถาปัตยกรรมคอมพิวเตอร์*. กรุงเทพฯ: สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น).

พัชรินทร์ บัวเย็น. (2556). *ไมโครคอมพิวเตอร์และพีซคณิตบูลีน*: บทที่ 3 ระบบบัสและสถาปัตยกรรมคอมพิวเตอร์เบื้องต้น [เอกสารอัดสำเนา]. จันทบุรี: สาขาวิชาคอมพิวเตอร์ธุรกิจ คณะวิทยาการจัดการ มหาวิทยาลัยราชภัฏรำไพพรรณี.

ไพศาล โมลิสกุลมงคล, ประสงค์ ประณีตพลกรัง, อนุโชต วุฒิพรพงษ์, และศรายุธ คลังทอง. (2547). *สถาปัตยกรรมคอมพิวเตอร์*. กรุงเทพฯ: ดวงกมลสมัย.

Englander, I. (2003). *The Architecture of Computer Hardware and System Software (3th edition): An Information Technology Approach.* John Wiley & Sons.

_____. (2010). *The Architecture of Computer Hardware, System Software, and Networking (4th edition): An Information Technology Approach.* John Wiley & Sons (Asia) Pte.

Hayes, J. P. (1998). *Computer Architecture and Organization (3rd edition).* Singapore: McGraw-Hill.

Murdocca, M. J., & Heuring, V. P. (2007). *Computer Architecture and Organization: In Integrated Approach.* USA: John Wiley & Sons.

References

เว็บไซต์อ้างอิง (รูปภาพ)

Englander, I. (2003). *The Architecture of Computer Hardware and System Software (3th edition): An Information Technology Approach.* John Wiley & Sons.

ITWissen. (n.d.). *AGP (Accelerated Graphics Port).* Retrieved May 17, 2015, from http://www.itwissen.info/definition/lexikon/accelerated-graphics-port-AGP-AGP-Bus.html

Karbo, M., & ELI Aps. (n.d.). *From ISA to PCI Express.* Retrieved May 2, 2013, from http://karbosguide.com/books/pcarchitecture/chapter39.htm

MIPS Technologies. (1996). *Superscalar Architecture.* Retrieved Jan 2, 2013, from http://techpubs.sgi.com/library/tpl/cgi-bin/getdoc.cgi/hdwr/bks/SGI_Developer/books/R10K_UM/sgi_html/t5.Ver.2.0.book_10.html

Murdocca, M. J., & Heuring, V. P. (2007). *Computer Architecture and Organization: In Integrated Approach.* USA: John Wiley & Sons.

National Instruments Corporation. (2014). *Data Streaming Architectures in PXI Systems.* In National Instruments Corporation (p.1). Retrieved Jan 2, 2015, from http://www.ni.com/white-paper/3221/en

Pelajarteknik. (2013). *Slot EKSPANSI.* Retrieved May 17, 2015, from https://pelajarteknik.wordpress.com/2013/11/19/slot-ekspansi

Scheffe, U. (2005). *PC Interfaces 101.* Retrieved May 17, 2015, from http://www.tomshardware.com/reviews/ pc-interfaces-101,1177-16.html