

บทที่ 6

สถาปัตยกรรมของโปรเซสเซอร์แบบต่างๆ

สไลด์ในหน้าที่ 4 แสดงการแบ่งประเภทของสถาปัตยกรรมคอมพิวเตอร์ ซึ่งสามารถมองได้หลายแง่มุม หากมองในแง่ของการจัด memory organization ก็แบ่งสถาปัตยกรรมคอมพิวเตอร์ได้เป็นสองแบบคือสถาปัตยกรรมแบบ Von-Neumann และสถาปัตยกรรมแบบ Harvard แต่ถ้าเราแบ่งประเภทของสถาปัตยกรรมคอมพิวเตอร์ตามชนิดของการจัดการชุดคำสั่งจะแบ่งออกเป็น 3 ประเภทคือ CISC, RISC และ VLIW

สถาปัตยกรรมแบบ Harvard จะแยกหน่วยความจำออกเป็นสองส่วนอย่างชัดเจน ได้แก่ หน่วยความจำโปรแกรมและหน่วยความจำข้อมูล โดยข้อกำหนดของสถาปัตยกรรมนี้จะกำหนดให้เก็บแอปพลิเคชันโปรแกรมและระบบปฏิบัติการไว้ในหน่วยความจำโปรแกรมเท่านั้น การแก้ไขข้อมูลในหน่วยความจำโปรแกรมขณะรันไทม์ไม่สามารถทำได้ และข้อมูลที่แอปพลิเคชันโปรแกรมประมวลผลจะเก็บอยู่ในหน่วยความจำข้อมูลเท่านั้น สถาปัตยกรรมซีพียูที่ใช้วิธีนี้ได้แก่ ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ของบริษัทอินเทล ซึ่งเป็นซีพียูขนาด 8 บิต ซึ่งมีจำนวนบัสแอดเดรสขนาด 16 เส้น สามารถมีหน่วยความจำโปรแกรมได้สูงสุดจำนวน 64 กิโลไบต์ และหน่วยความจำข้อมูลสูงสุดได้ 64 กิโลไบต์เช่นกัน ดังนั้นสถาปัตยกรรมของ MCS-51 จึงสามารถต่อหน่วยความจำได้รวมเท่ากับ 128 กิโลไบต์ ตัวอย่างของซีพียูอีกสถาปัตยกรรมหนึ่งที่จัดหน่วยความจำแบบ Harvard ก็คือซีพียูตระกูล MIPS ของบริษัทซิลิคอนกราฟิกส์ สไลด์หน้า 6-7 แสดงให้เห็นถึงโครงสร้างของสถาปัตยกรรมดังกล่าว

สถาปัตยกรรมอีกรูปแบบหนึ่งคือสถาปัตยกรรมแบบ Von-Neumann ซึ่งไม่ได้แบ่งหน่วยความจำของข้อมูลและโปรแกรมออกจากกัน ซึ่งในสถาปัตยกรรมแบบนี้อนุญาตให้สามารถเปลี่ยนแปลงโค้ดโปรแกรมในขณะรันแอปพลิเคชันได้ โดยในสถาปัตยกรรมนี้เราจะเก็บแอปพลิเคชันโปรแกรม ระบบปฏิบัติการและข้อมูลที่ประมวลผลไว้ในหน่วยความจำหลักไว้ที่ใดก็ได้ดังสไลด์หน้า 9 แต่ในทางปฏิบัติ แล้วการจัดการข้อมูลว่าเก็บโปรแกรมและข้อมูลไว้ที่ตำแหน่งใดในหน่วยความจำนั้นจะขึ้นอยู่กับการตัดสินใจของระบบปฏิบัติการเป็นหลัก ตัวอย่างซีพียูที่ใช้สถาปัตยกรรมนี้ได้แก่ซีพียู 80x86 ของบริษัทอินเทลและสถาปัตยกรรม z80 ของบริษัท ไชล๊อค เป็นต้น

นอกจากการแบ่งสถาปัตยกรรมของคอมพิวเตอร์ตามลักษณะการต่อหน่วยความจำแล้วยังมองถึงรูปแบบของการออกแบบชุดคำสั่งด้วย สถาปัตยกรรมแบบ CISC ย่อมาจาก Complex Instruction Set Computer ซึ่งจะมีชุดคำสั่งที่มีความซับซ้อนในการทำงานสูง ยกตัวอย่างเช่นในบางซีพียู เราสามารถใช้คำสั่งในการค้นหาข้อมูลที่ต้องการในบล็อกของหน่วยความจำขนาดใหญ่ได้ด้วยคำสั่งเพียงคำสั่งเดียว และมีจำนวนคำสั่งใช้งานเป็นจำนวนมาก ประมาณ 200-300 คำสั่งหรือมากกว่านั้นซีพียูมีแอดเดรสซึ่งโหมดให้เลือกใช้เป็นจำนวนมาก ซีพียูแบบนี้มีข้อดีคือเขียนโปรแกรมได้ง่าย ตัวอย่างสถาปัตยกรรมนี้ได้แก่สถาปัตยกรรม 680x0 ของโมโตโรล่า และสถาปัตยกรรม 80x86 ของอินเทล ซึ่งสาเหตุที่มีชุดคำสั่งจำนวน

มากเพราะผู้พัฒนาต้องการให้ซีพียูรุ่นใหม่สามารถใช้ซอฟต์แวร์ของเก่าได้ ดังนั้นเพื่อเพิ่มประสิทธิภาพในการทำงานผู้ผลิตจึงเพิ่มคำสั่งใหม่ขึ้นมาให้ใช้งาน ส่งผลให้ซีพียูรุ่นใหม่ ๆ มีคำสั่งใช้งานเพิ่มมากขึ้นเรื่อย ๆ

อีกแนวทางในการพัฒนาซีพียูคือพยายามลดจำนวนคำสั่งให้เหลือน้อยที่สุด เท่าที่จำเป็นเท่านั้น ทั้งนี้เนื่องจากมีงานวิจัยหลายชิ้นพบว่าแม้ว่าผู้ผลิตซีพียูออกแบบคำสั่งมาให้จำนวนมากแต่คอมพิวเตอร์ทั่วไปไม่ได้ใช้ชุดคำสั่งบางคำสั่งของซีพียูเลย ดังนั้นจึงมีแนวคิดว่าจะลดความซับซ้อนของซีพียูให้ได้มากที่สุด โดยคงไว้เฉพาะคำสั่งที่จำเป็น ซึ่งจะทำให้ง่ายต่อการออกแบบซีพียูและเพิ่มสมรรถนะของซีพียูในภายหลังจึงเกิดแนวคิดของ RISC (Reduced Instruction Set Computer) ซึ่งแสดงให้เห็นในสไลด์หน้าที่ 12 แนวคิดของ RISC นี้พยายามลดความซับซ้อนของคำสั่งลงโดยกำหนดให้คำสั่งที่สามารถอ่านหน่วยความจำได้คือคำสั่ง Load และคำสั่ง Store เท่านั้น ส่วนคำสั่งอื่นๆ จะทำกับข้อมูลในรีจิสเตอร์เท่านั้น และเพื่อให้ง่ายต่อการถอดรหัสคำสั่ง ซีพียูจึงกำหนดให้มีความยาวของคำสั่งคงที่ ซึ่งแตกต่างจาก CISC ที่มีความยาวของคำสั่งแตกต่างกันไปตามชนิดของคำสั่ง ส่งผลให้วงจร Instruction decoder ในซีพียูมีความซับซ้อนขึ้นเป็นเงาตามตัว ในซีพียูแบบนี้เนื่องจากมีชุดคำสั่งน้อย ดังนั้นในการทำโอเปอเรชั่นที่มีความซับซ้อนทำได้โดยเอาคำสั่งพื้นฐานที่มีให้มาเขียนโปรแกรมให้ทำงานซับซ้อนให้ได้ตามต้องการ ซีพียูแบบ RISC ส่วนใหญ่มีรีจิสเตอร์ให้ใช้งานเป็นจำนวนมากเพื่อลดการสลับข้อมูลระหว่างหน่วยความจำหลักและรีจิสเตอร์ให้ได้มากที่สุด และเนื่องจากคำสั่งมีรูปแบบที่ง่าย จึงสามารถสร้างวงจร control unit แบบ Hardwired ขึ้นมาควบคุมซึ่งมีความเร็วสูงกว่าแบบใช้ไมโครโปรแกรมที่ใช้ในแบบ CISC ตัวอย่างซีพียูที่ใช้สถาปัตยกรรมแบบนี้ได้แก่ซีพียู MIPS ของบริษัทซิลิคอนกราฟิกส์ ซีพียู Alpha ของบริษัท Compaq-HP เป็นต้น

สไลด์หน้าที่ 14 เป็นการเปรียบเทียบคุณสมบัติของสถาปัตยกรรมแบบ CISC และแบบ RISC ซึ่งสถาปัตยกรรมแบบ CISC เนื่องจากคำสั่งมีความซับซ้อนจึงสามารถใช้แทนคำสั่งของ RISC ได้หลายคำสั่ง ส่งผลให้แอปพลิเคชันโปรแกรมของซีพียู CISC มักจะมีขนาดเล็กกว่าของแบบ RISC อย่างไรก็ตามเนื่องจากสถาปัตยกรรมแบบ RISC มีความซับซ้อนน้อยกว่า ส่งผลให้ต้องการจำนวนทรานซิสเตอร์ในการสร้างวงจรมีน้อยกว่า ส่งผลให้การกินไฟของซีพียูแบบ RISC ค่อนข้างต่ำมากเมื่อเทียบกับซีพียูแบบ CISC ดังนั้นซีพียู RISC มักจะพบในงานที่ต้องการความสามารถในงานประหยัดพลังงาน เช่น ซีพียูในโทรศัพท์มือถือมักจะใช้ซีพียูแบบ RISC ทั้งสิ้นยกตัวอย่างซีพียู RISC ที่นิยมใช้ในโทรศัพท์มือถือเช่น ARM, MIPS เป็นต้น

สไลด์หน้าที่ 15 แสดงให้เห็นภาพรวมในการพัฒนาซีพียูในท้องตลาดซึ่งจะเห็นว่าส่วนใหญ่จะใช้สถาปัตยกรรมแบบ RISC เป็นส่วนใหญ่ แต่อย่างไรก็ตามซีพียูที่ประสบความสำเร็จสูงสุดในท้องตลาดกลับเป็นสถาปัตยกรรมแบบ CISC ซึ่งก็คือสถาปัตยกรรม 80x86 ของบริษัทอินเทล สาเหตุหลักเนื่องจากมีแอปพลิเคชันสนับสนุนเป็นจำนวนมากและราคาที่ถูกนั่นเอง ส่วนซีพียู RISC มักจะใช้ในเครื่องระดับเวิร์กสเตชันขึ้นไป ซึ่งมีตลาดที่แคบกว่า

สถาปัตยกรรมชุดคำสั่งแบบสุดท้ายที่พบที่นี่จะพูดถึงคือสถาปัตยกรรมชุดคำสั่งแบบ VLIW (Very long Instruction Word) ซึ่งแสดงให้เห็นในสไลด์หน้าที่ 16 จุดที่แตกต่างจาก RISC และ CISC คือสถาปัตยกรรมแบบนี้จะมีคำสั่งที่ยาวมาก ในหนึ่งคำสั่งจะประกอบไปด้วยโอเปอเรชั่นหลายๆ โอเปอเรชั่น ซึ่งจากรูปในสไลด์ยกตัวอย่าง คำสั่ง 1 คำสั่งของซีพียูแบบ VLIW จะแทนคำสั่งของแบบ CISC/RISC ได้ถึง 4

คำสั่ง ด้วยกัน ในซีพียูแบบ VLIW จะมีหน่วย execution unit หลายตัว ซึ่งจะมีจำนวนเท่ากับจำนวนโอเปอเรชั่นที่สามารถทำได้ใน 1 คำสั่ง ซึ่งโอเปอเรชั่นดังกล่าวสามารถทำได้พร้อมๆ กัน ส่งผลให้ในขณะใดขณะหนึ่ง ซีพียูแบบ VLIW สามารถทำงานได้มากกว่าซีพียูแบบ RISC/CISC เพื่อที่จะให้ได้การทำงานดังกล่าว ซีพียูแบบนี้ต้องอาศัยความสามารถของคอมไพเลอร์ในการจัดหาโอเปอเรชั่นที่สามารถทำงานพร้อมกันได้บรรจุลงไปในการคำสั่งเดียวกันหรือ Instruction Level parallelism (ILP) หากคอมไพเลอร์ไม่สามารถหาโอเปอเรชั่นที่สามารถทำพร้อมกันได้ครบเท่ากับจำนวนโอเปอเรชั่นที่คำสั่งต้องการ ตัวคอมไพเลอร์จะใส่โอเปอเรชั่น NOP เข้าไปในคำสั่งแทน เนื่องจากการทำงานขนานกันของโอเปอเรชั่นถูกจัดการมาตั้งแต่ขั้นตอนการคอมไพล์แล้ว ดังนั้นหน่วยเฟตช์ข้อมูลในซีพียูแบบนี้จะมีความซับซ้อนน้อยกว่าการใช้เทคนิคซูเปอร์สเกลาร์ค่อนข้างมาก อย่างไรก็ตาม ซีพียูแบบ VLIW มีข้อเสียในแง่ของการที่มีการระบุการทำงานขนานลงไปอย่างชัดเจนในคำสั่งว่าโอเปอเรชั่นใดทำงานกับ execution unit ตัวใด ส่งผลให้โปรแกรมที่เขียนขึ้นมาเพื่อใช้กับซีพียูรุ่นใดรุ่นหนึ่งไม่ได้รับประโยชน์จาก execution unit ที่เพิ่มขึ้นมาในซีพียูรุ่นหลังเลย หากต้องการใช้ความสามารถของ execution unit ที่เพิ่มขึ้นมาใหม่จะต้องนำซอร์สโค้ดของแอปพลิเคชันนั้นมาคอมไพล์ใหม่ ดังนั้นซีพียูที่ใช้เทคนิค VLIW นี้จึงไม่ค่อยประสบความสำเร็จมากนักในตลาดคอมพิวเตอร์ทั่วไป สืบเนื่องจากผู้ใช้ส่วนใหญ่ต้องการความสามารถทางด้านการนำเอาแอปพลิเคชันเดิมมาใช้กับซีพียูรุ่นใหม่ได้ อย่างไรก็ตามสืบเนื่องจากคุณสมบัติของความเร็วในการทำงานและกินพลังงานต่ำของสถาปัตยกรรมแบบ VLIW ส่งผลให้สถาปัตยกรรมดังกล่าวนิยมมากในงานที่ไม่ต้องการการเปลี่ยนแปลงซอฟต์แวร์โดยผู้ใช้ ดังนั้นสถาปัตยกรรมแบบนี้จึงนิยมใช้งานมากในงานการประมวลผลสัญญาณดิจิทัล (Digital signal processing : DSP) เช่น ชิป TriMedia ของบริษัท Philips เป็นต้น
