ระบบคอมพิวเตอร์และสถาปัตยกรรม

(Computer System and Architecture)

Chapter 5

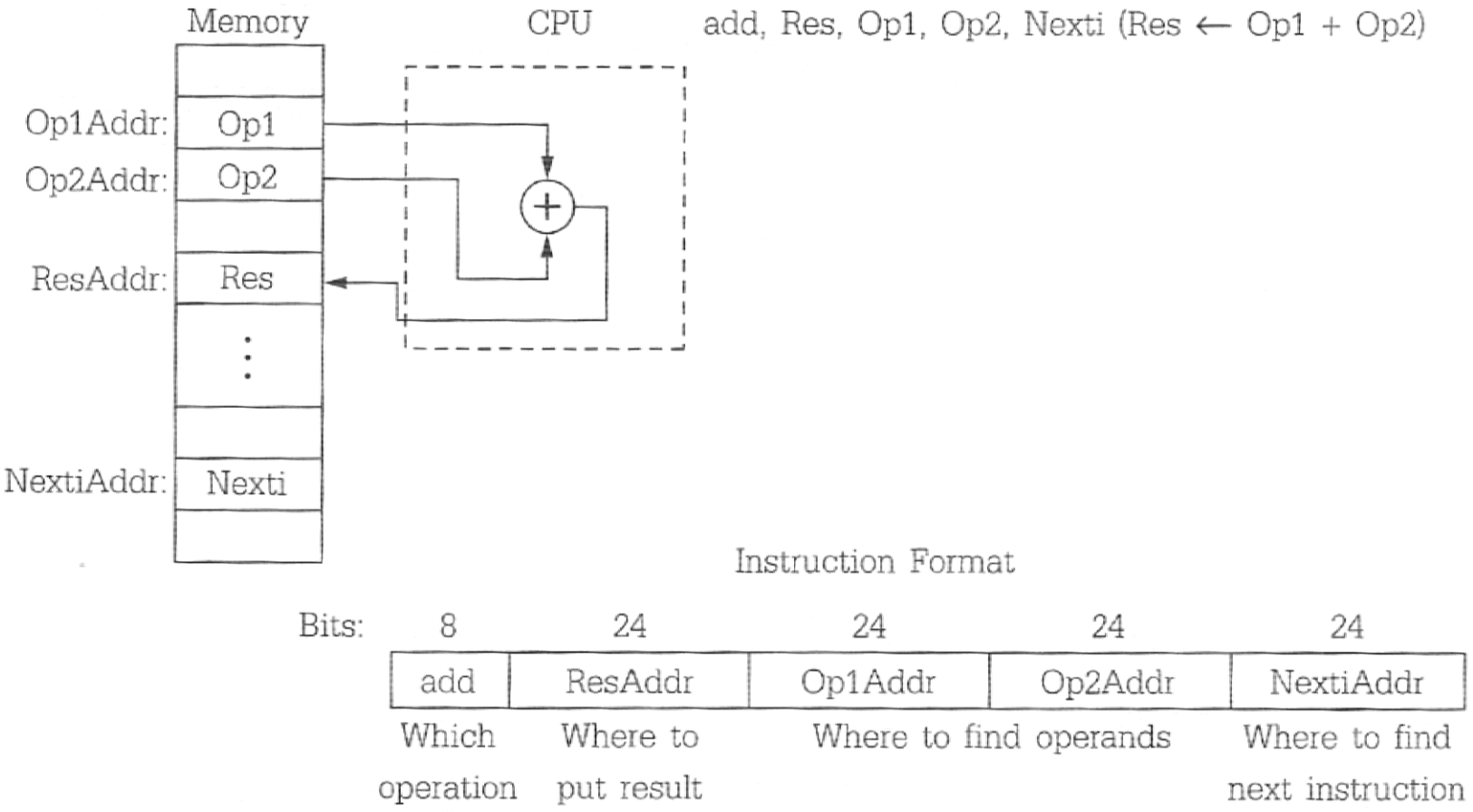
ไมโครโพรเซสเซอร์เบื้องต้น

โดย ผู้ช่วยศาสตราจารย์ภานุวัฒน์ เมฆะ

สาขาวิชาวิทยาการคอมพิวเตอร์

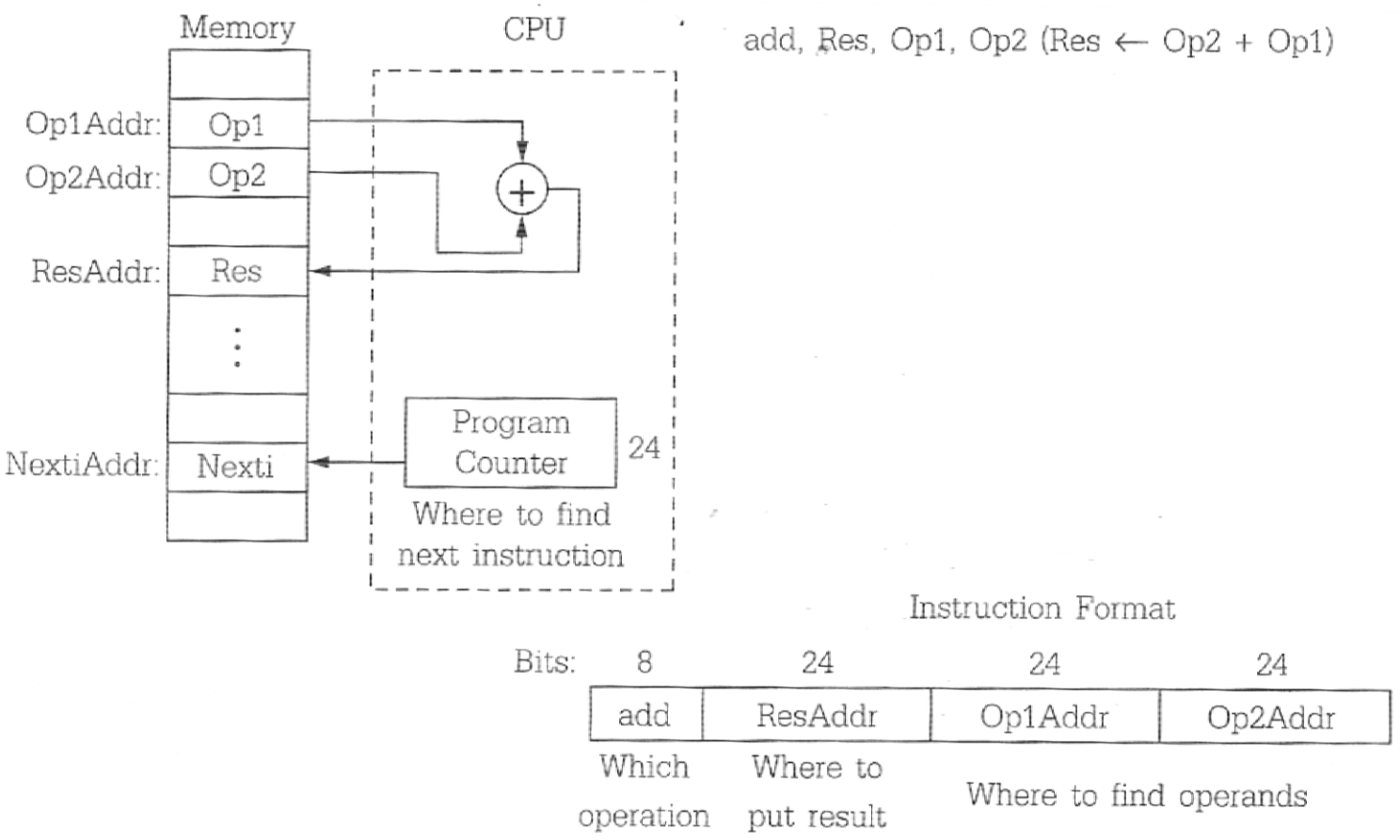
คณะวิทยาศาสตร์ มหาวิทยาลัยแม่โจ้

แนวคิดการออกแบบไมโครโพรเซสเซอร์



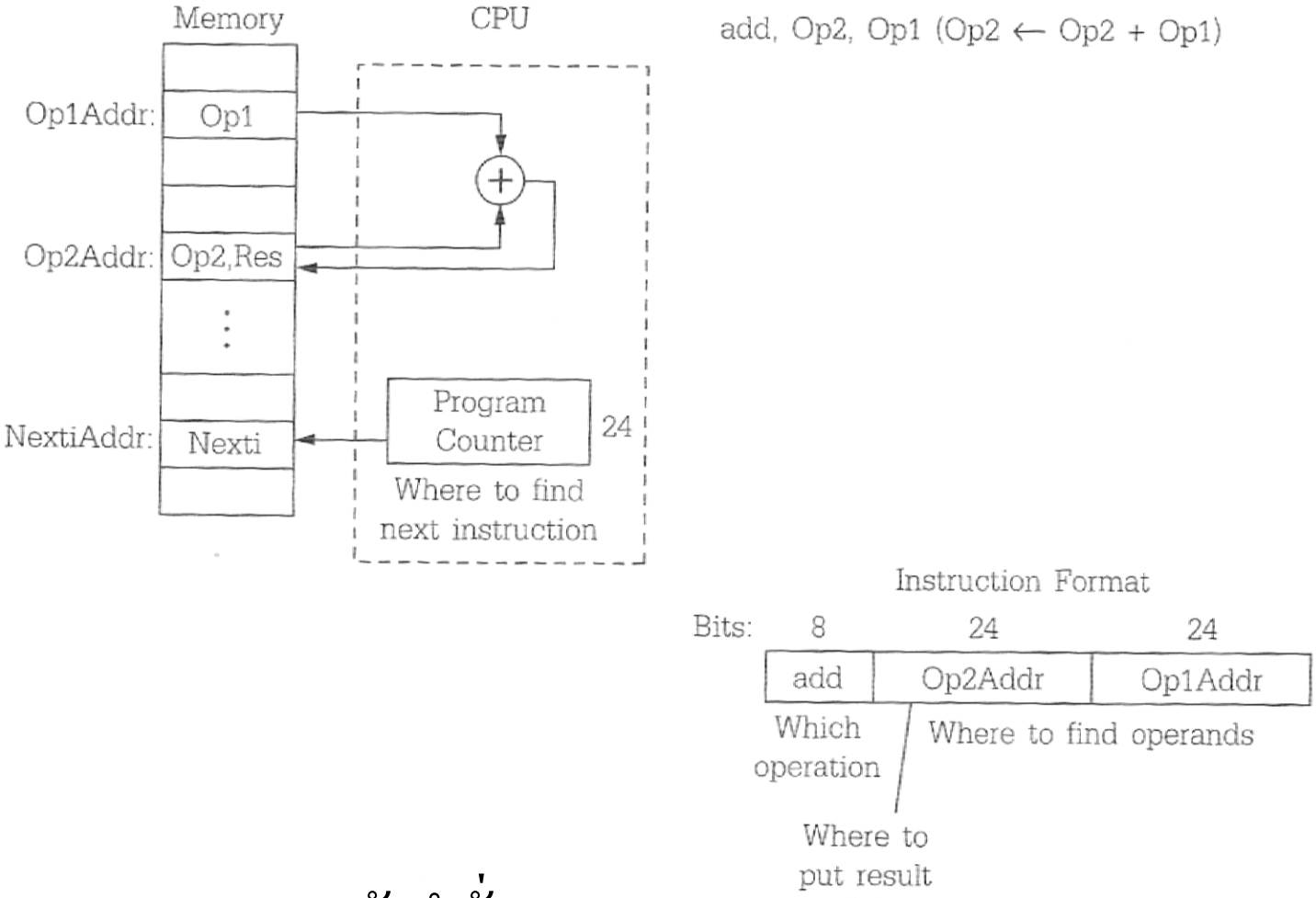
รหัสคำสั่งแบบ 4-Address Instruction

แนวคิดการออกแบบไมโครโพรเซสเซอร์



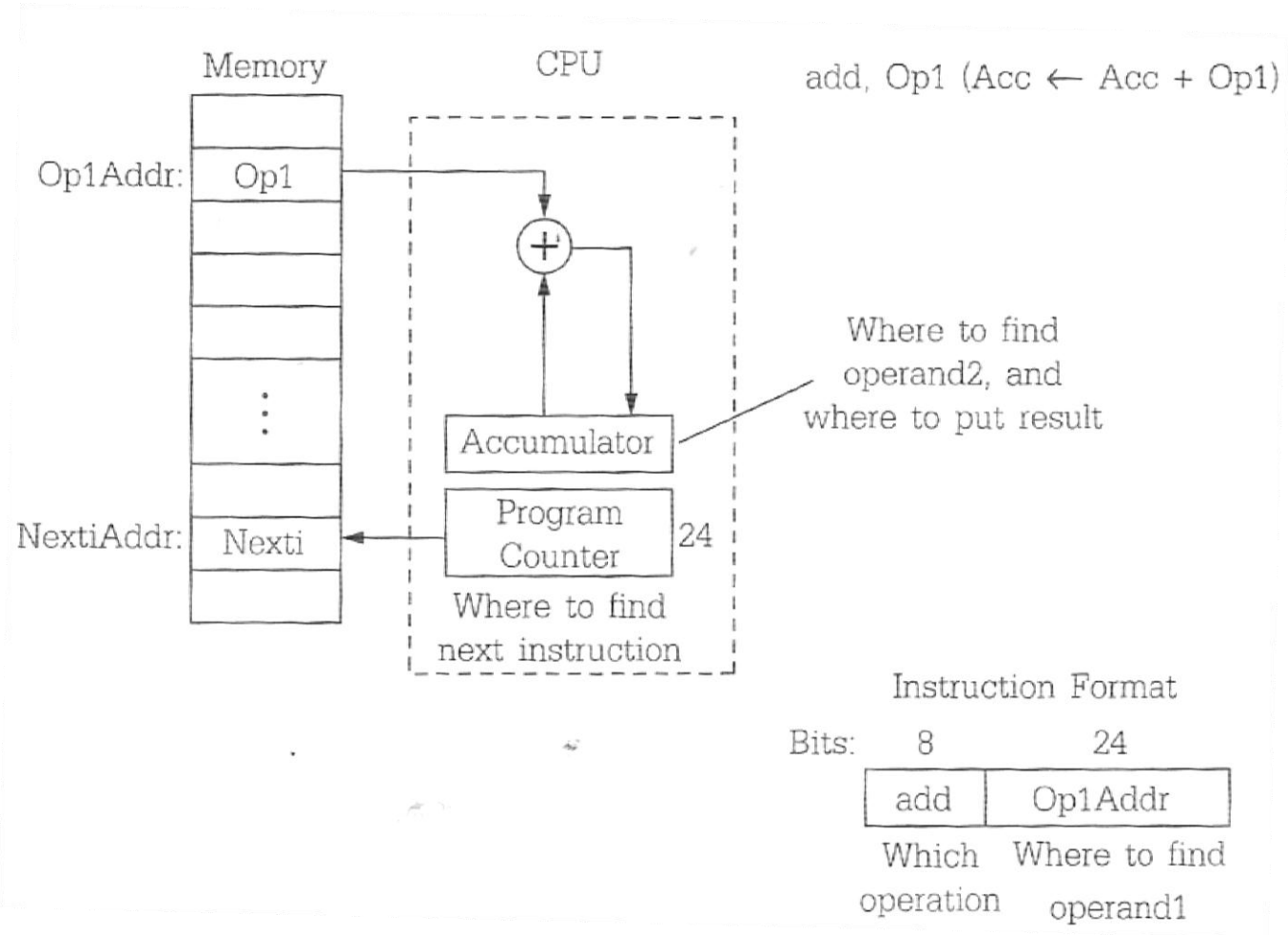
รหัสคำสั่งแบบ 3-Address Instruction

แนวคิดการออกแบบไมโครโพรเซสเซอร์



รหัสคำสั่งแบบ 2-Address Instruction

แนวคิดการออกแบบไมโครโพรเซสเซอร์



รหัสคำสั่งแบบ 1-Address Instruction

การเขียนคำสั่ง

รหัสช่วยจำ	ความหมาย
ADD	Addition
SUB	Subtraction
MPY	Multiply
DIV	Divide
LOAD	Load Data from Memory
STOR	Store Data to Memory

Y = (A - B) % (C + D * E)

Instruction		Comment
SUB	Y, A, B	Y ← A - B
MPY	T, D, E	T ← D × E
ADD	T, T, C	T ← T + C
DIV	Y, Y, T	Y ← Y ÷ T

(1) รหัสคำสั่งแบบ 3-Address Instruction

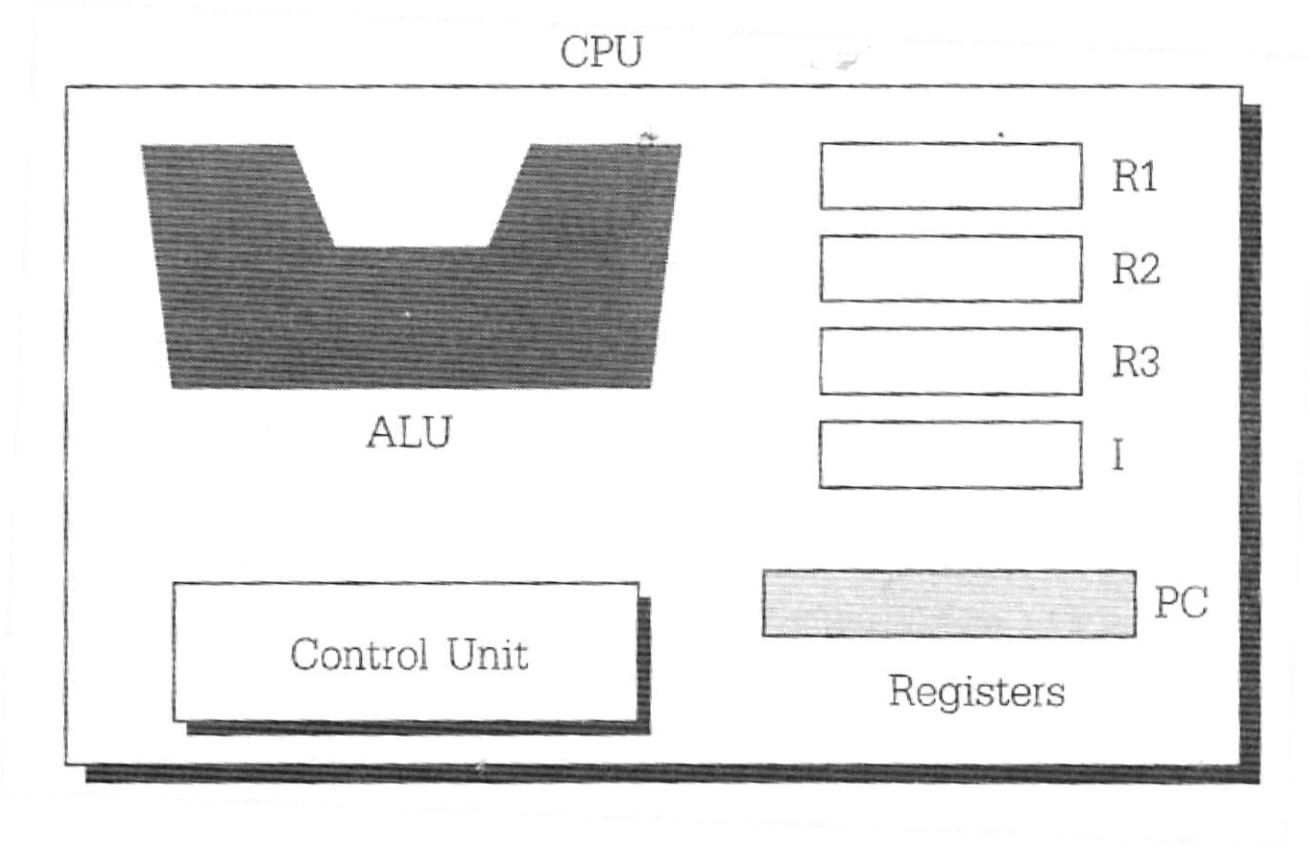
Instruction		Comment
MOVE	Y, A	Y ← A
SUB	Y, B	Y ← Y - B
MOVE	T, D	T ← D
MPY	T, E	T ← T × E
ADD	T, C	T ← T + C
DIV	Y, T	Y ← Y ÷ T

(2) รหัสคำสั่งแบบ 2-Address Instruction

Instruction		Comment
LOAD	D	AC ← D
MPY	E	AC ← AC × E
ADD	C	AC ← AC + C
STOR	Y	Y ← AC
LOAD	A	AC ← A
SUB	B	AC ← AC - B
DIV	Y	AC ← AC ÷ Y
STOR	Y	Y ← AC

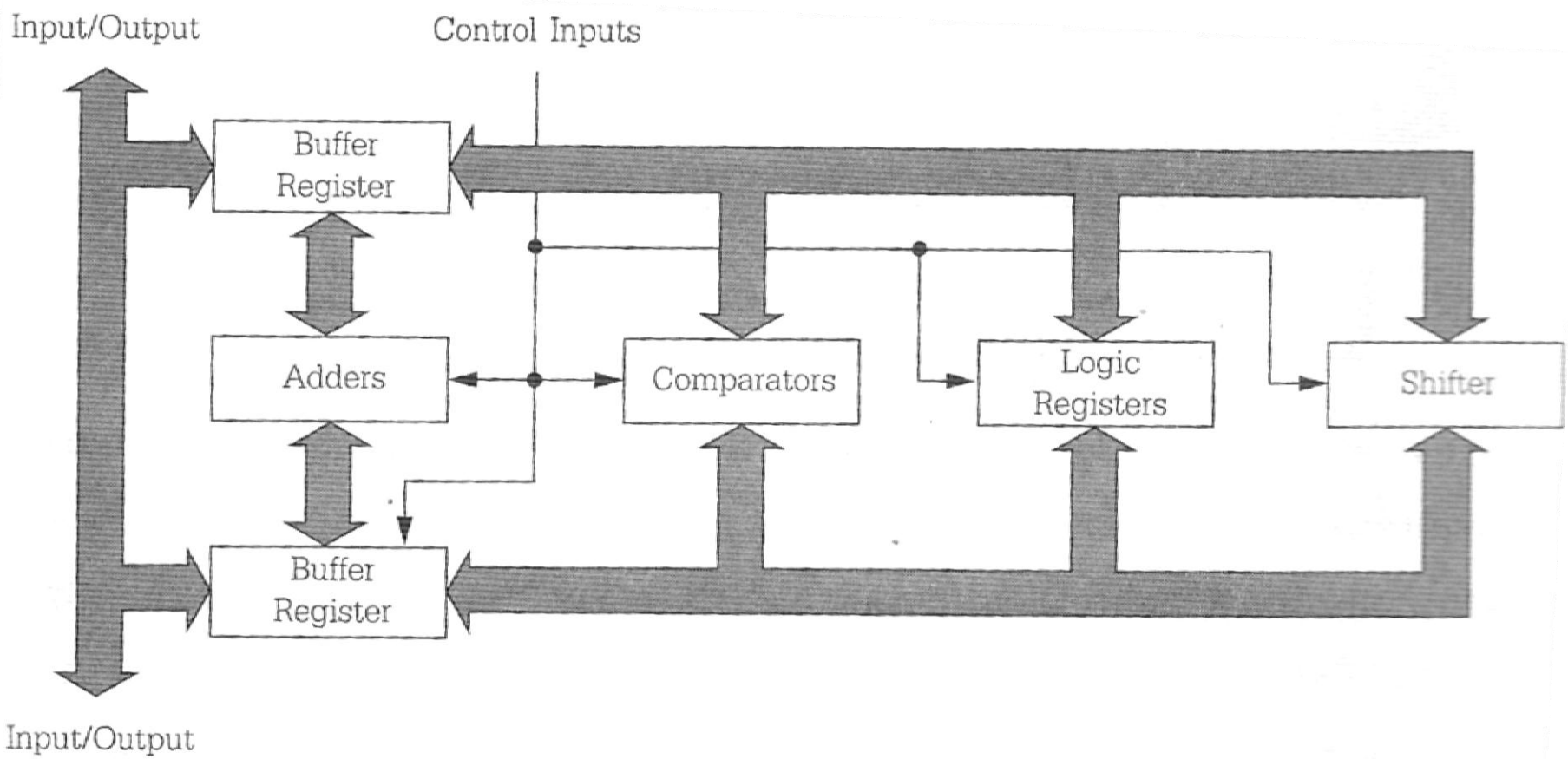
(3) รหัสคำสั่งแบบ 1-Address Instructior

โครงสร้างพื้นฐานของไมโครโพรเซสเซอร์



โครงสร้างอย่างง่ายภายในซีพียู

โครงสร้างพื้นฐานของไมโครโพรเซสเซอร์



โครงสร้างอย่างง่ายของ ALU

รูปแบบคำสั่งและโหมดของแอดเดรส

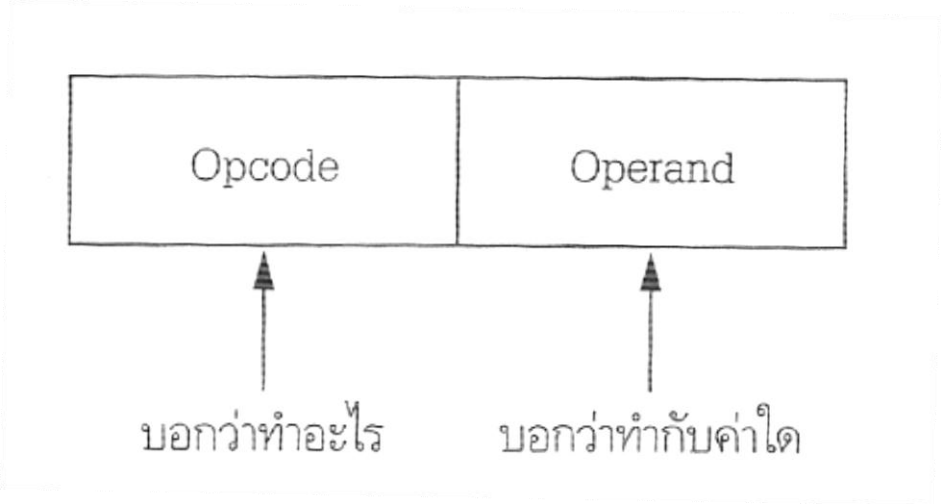
Ex.

MOV Oper1, Oper2

: เป็นการใช้คำสั่งโอนย้ายข้อมูลจากโอเปอร์แรนด์ตัวที่ 2 ไปยังโอเปอร์แรนด์ตัวที่ 1

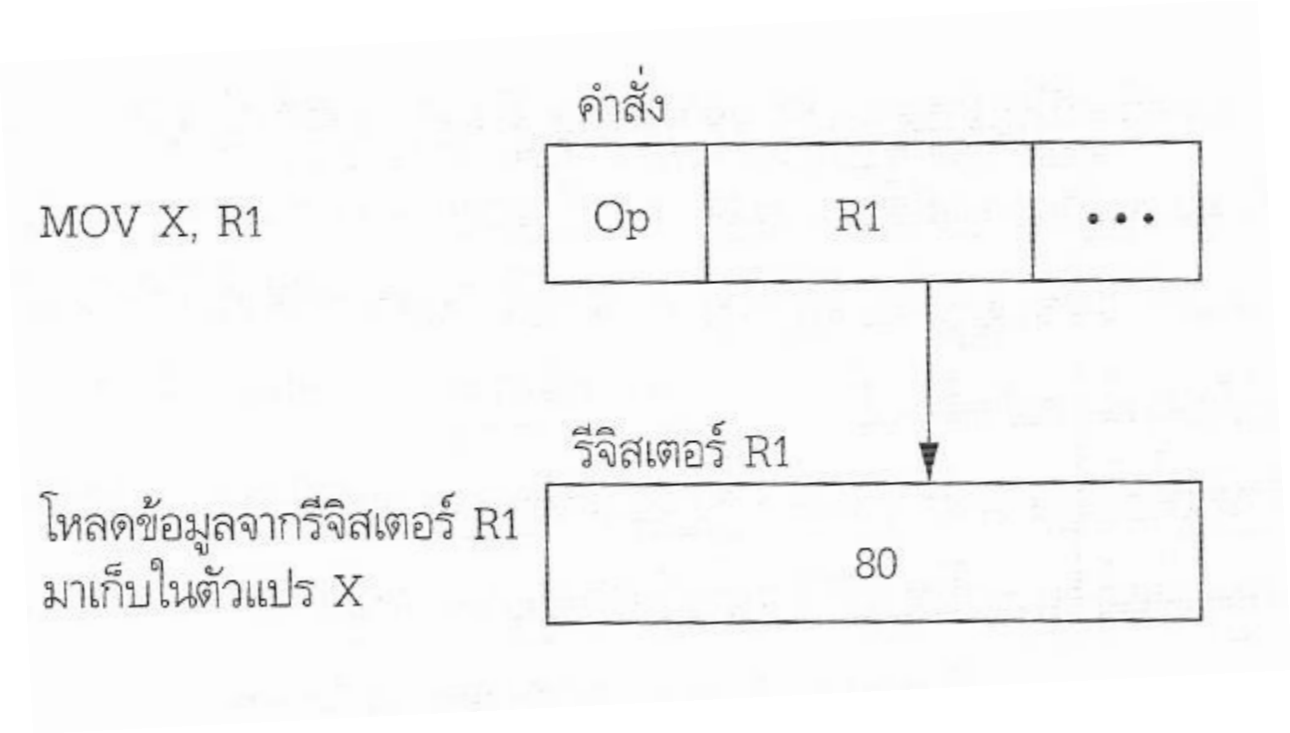
การกำหนดแอดเดรสแบบให้ค่าตรง

(Immediate Address Mode)



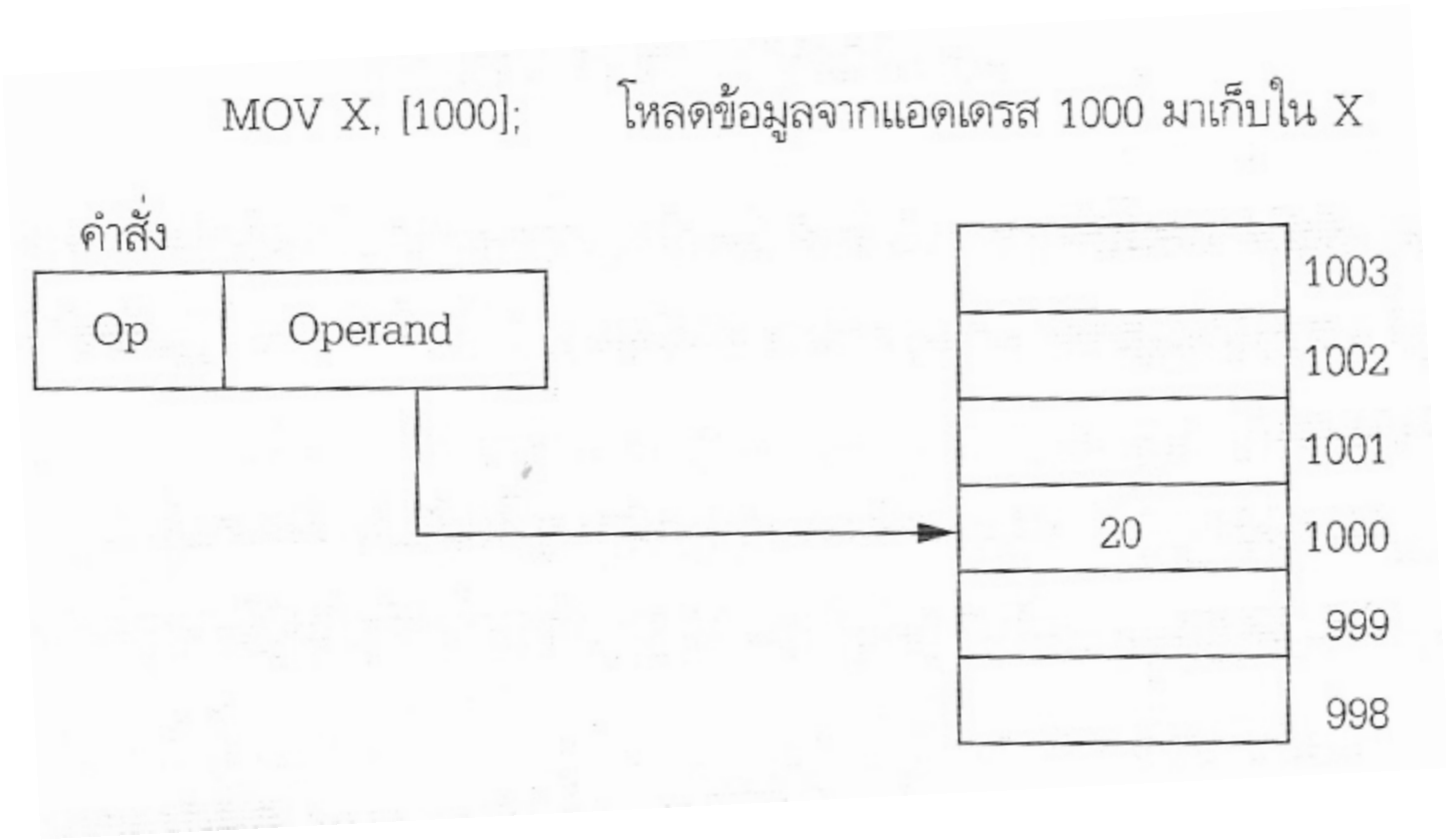
การกำหนดแอดเดรสโดยใช้รีจิสเตอร์

(Register Addressing Mode)



การกำหนดแอดเดรสโดยตรง

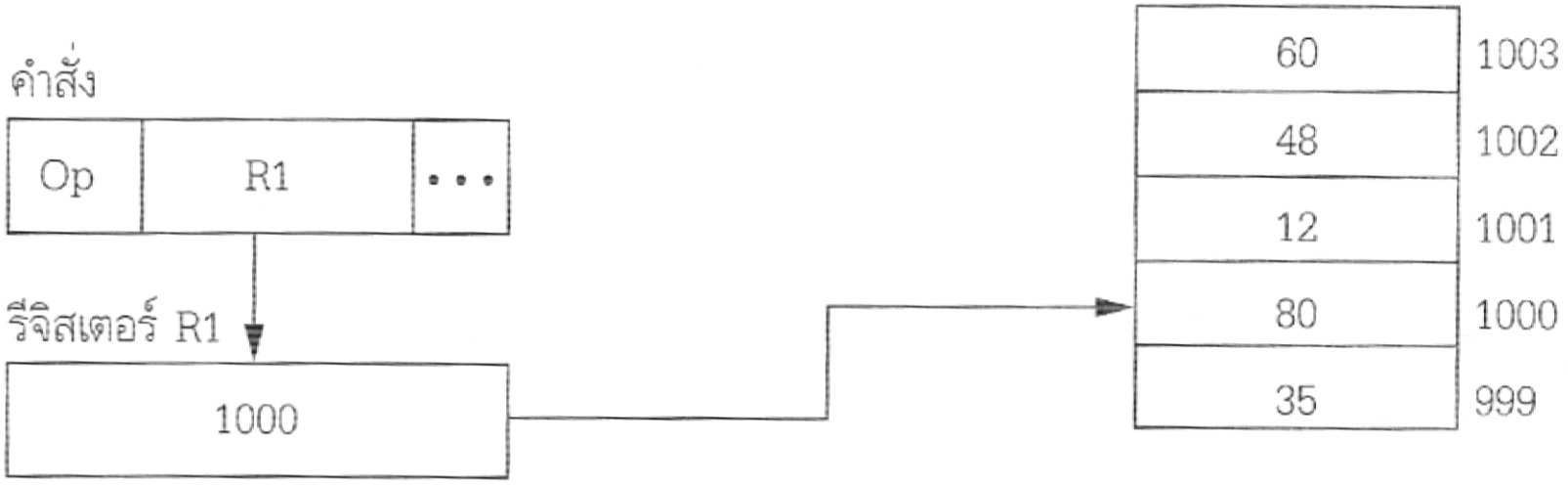
(Direct Addressing Mode)



การกำหนดแอดเดรสผ่านรีจิสเตอร์โดยอ้อม

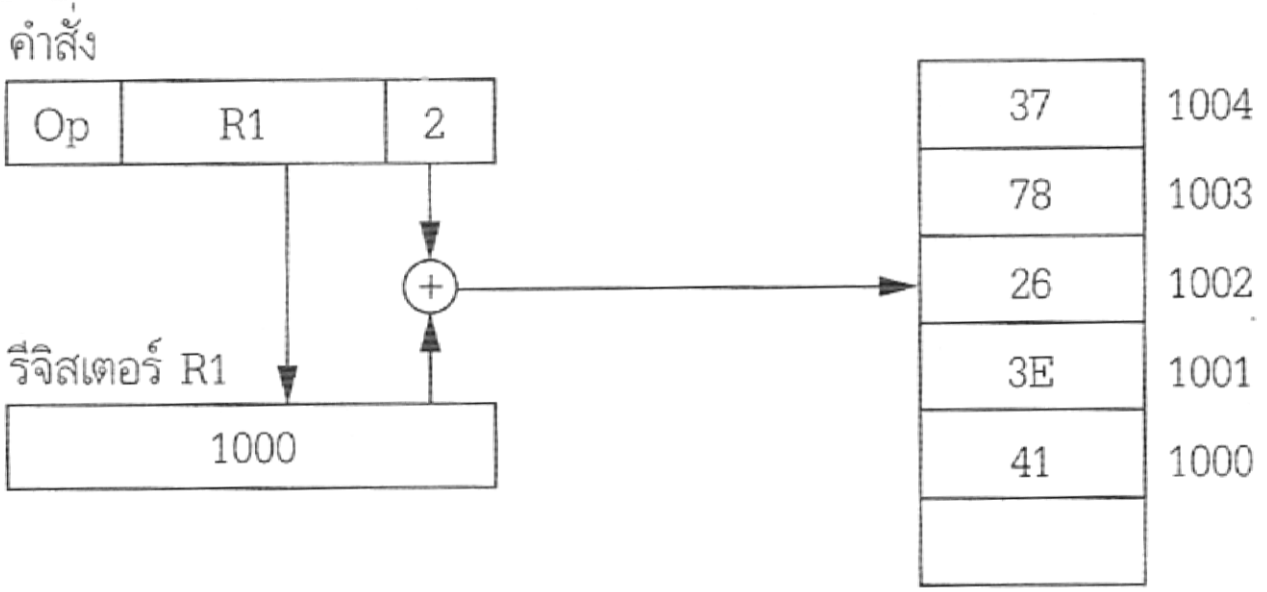
(Register Indirect Addressing Mode)

MOV X, [R1] ; โหลดข้อมูลจากแอดเดรสที่เก็บอยู่ในรีจิสเตอร์ R1 มาเก็บในตัวแปร X
ถ้า R1 เก็บ 1000 จะทำให้ค่าในตัวแปร X มีค่าเป็น 80



การกำหนดแอดเดรสแบบแทนที่

(Indexed Addressing Mode)



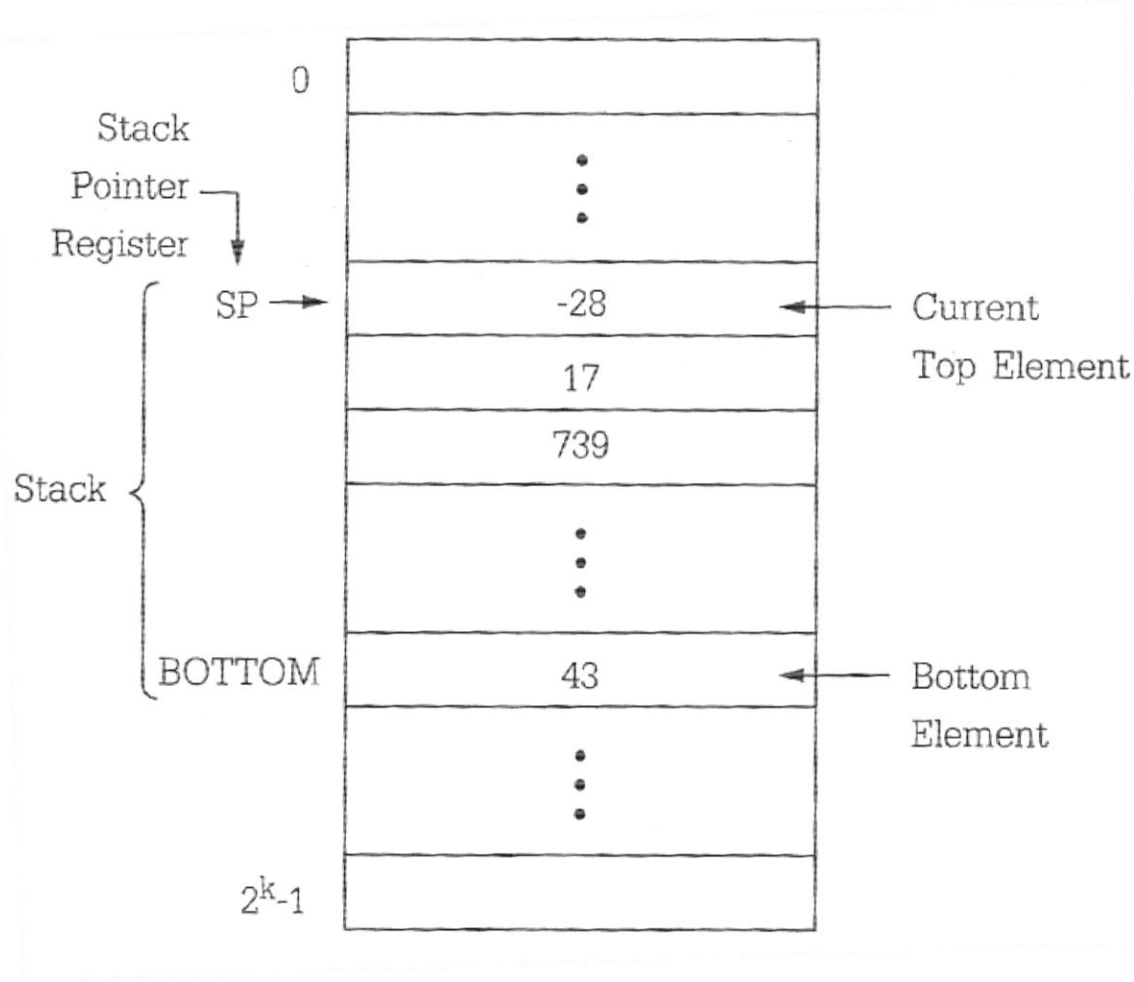
การกำหนดแอดเดรสแบบสัมพันธ์

(Relative Addressing Mode)

เป็นการอ้างถึงตำแหน่งหน่วยความจำที่อ้างอิงกับโปรแกรมเคาน์เตอร์ (PC) โดยจะนำค่าคงที่ไปบวกกับค่าของโปรแกรมเคาน์เตอร์ ค่าที่ได้จะเป็นแอดเดรสของหน่วยความจำที่ต้องการอ้างอิงถึง

การกำหนดแอดเดรสโดยใช้สแตก

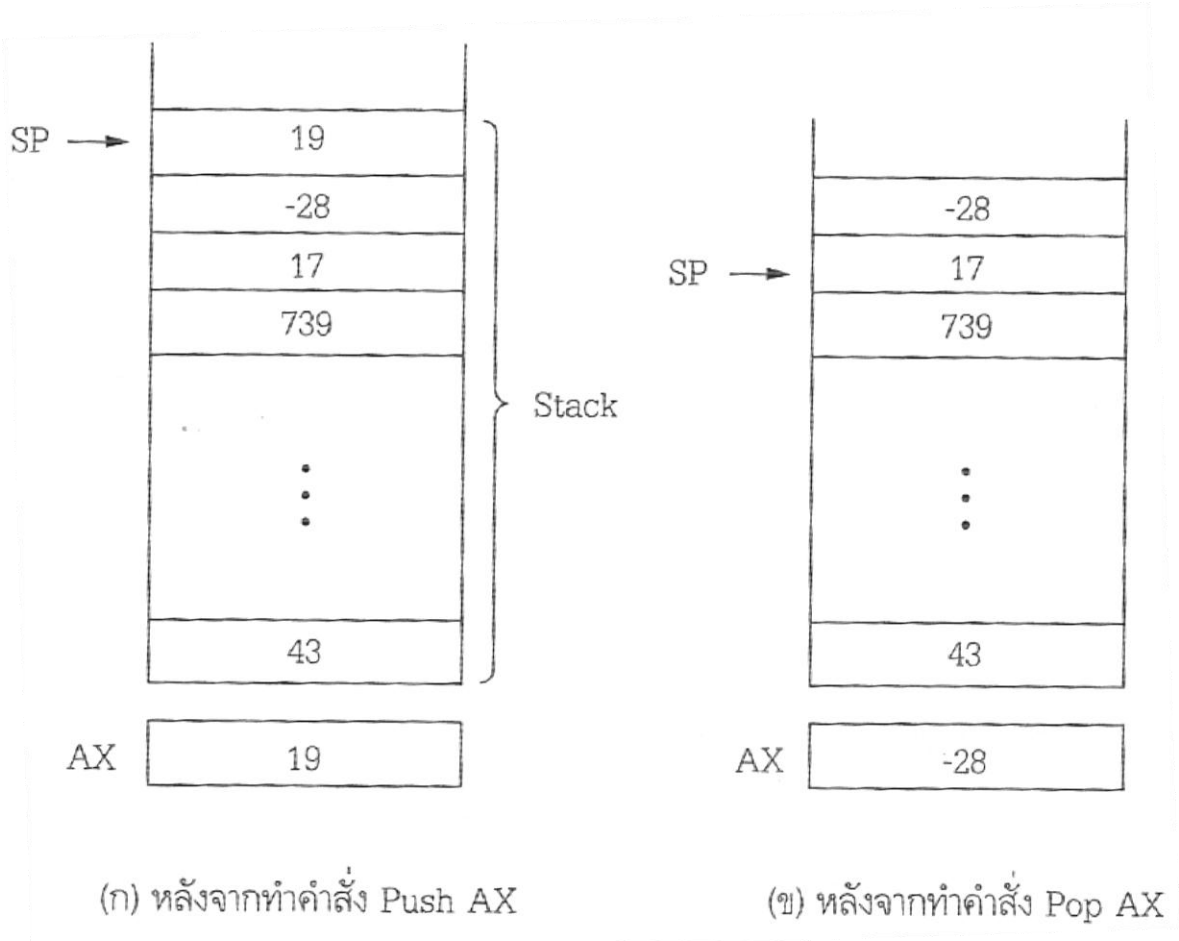
(Stack Addressing Mode)



ตัวอย่างสแตกในหน่วยความจำ

การกำหนดแอดเดรสโดยใช้สแตก

(Stack Addressing Mode)

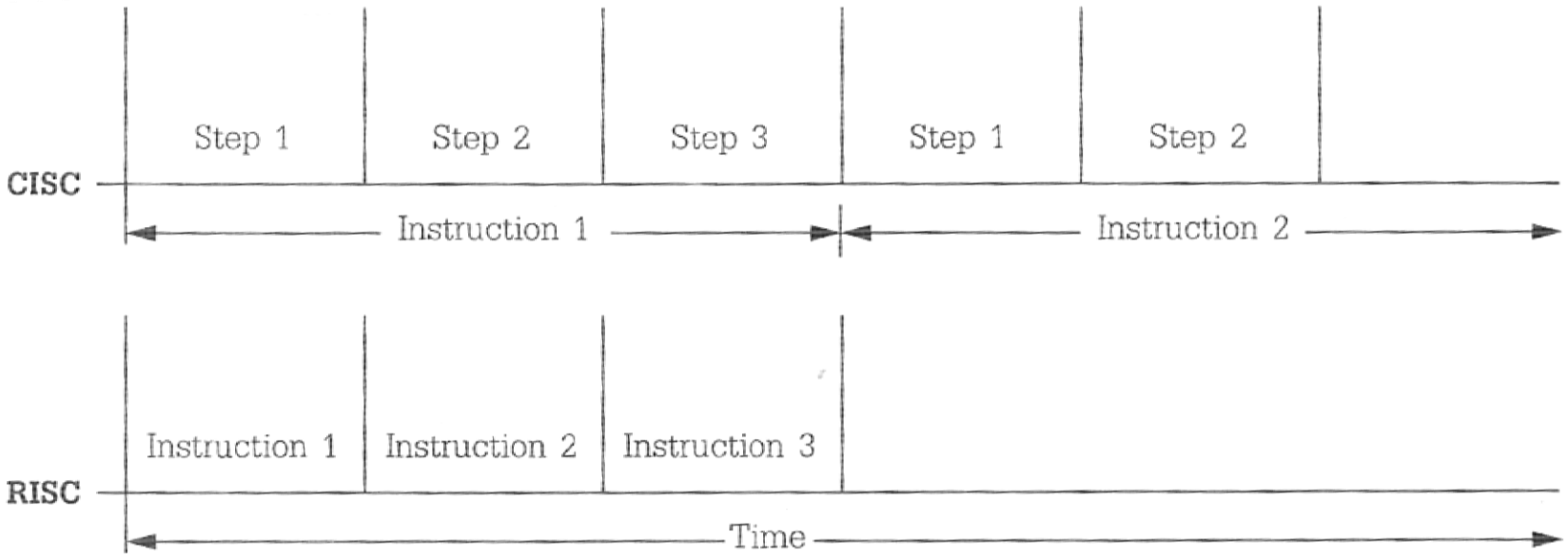


การดำเนินการหลังจากการ Push และการ Pop

สถาปัตยกรรมของ CPU

- สถาปัตยกรรมของซีพียูเป็นตัวบ่งบอกถึงลักษณะเฉพาะ และลักษณะการทำงานที่สำคัญของซีพียู
- อาจจะเรียกอีกแบบหนึ่งว่า “สถาปัตยกรรมชุดคำสั่ง” (ISA :Instruction Set Architecture)
- ลักษณะเฉพาะของซีพียูนี้กล่าวรวมไปถึง จำนวนและประเภทของรีจิสเตอร์, วิธีการกำหนดโหมดของ address ของหน่วยความจำและการออกแบบชุดคำสั่งต่าง ๆ (Instruction Sets)
- ปัจจุบันนี้แบ่งออกเป็น 2 กลุ่มคือ CISC (Complex Instruction Set Computers) และ RISC (Reduced Instruction Set Computers)

การทำคำสั่งของคำสั่งแบบ CISC และ RISC



คุณลักษณะของคอมพิวเตอร์แบบ CISC

- มีการรวมเอาคุณสมบัติด้านต่าง ๆ เช่น การกำหนดโหมดของ address (Addressing Mode) หรือประเภทคำสั่ง (Instruction Type) เข้าด้วยกันเพื่อปรับปรุงขีดความสามารถในการทำงาน
- นักออกแบบมองว่าปริมาณการใช้หน่วยความจำ และเวลาที่ใช้ในการเข้าถึงต่าง ๆในการทำงานของคอมพิวเตอร์ต้องมองเป็นราคาต่อหน่วย
- สถาปัตยกรรมแบบ CISC มีจำนวน Addressing Mode หลายแบบ
- ลักษณะนี้ส่งผลให้ชุดคำสั่งที่ใช้มีขนาด และใช้เวลาในการประมวลผลแตกต่างกัน
- ผู้ใช้ยอมรับและพอใจ เนื่องจากผู้ใช้สามารถเพิ่มจำนวน operation บนโปรแกรมที่มีขนาดเท่าเดิมได้

คุณลักษณะของคอมพิวเตอร์แบบ RISC

- เดิมคอมพิวเตอร์ยุคแรกจะประมวลผลชุดคำสั่งทีละคำสั่งในหนึ่งช่วงเวลา
- นักออกแบบได้พยายามที่จะปรับปรุงอัตราการประมวลผลชุดคำสั่งให้เพิ่มมากขึ้นโดยใช้วิธีการโอเวอร์แลป (Overlap) ชุดคำสั่งให้มีมากกว่าชุดคำสั่งในการประมวลผลแต่ละครั้ง
- วิธีการโอเวอร์แลปดังกล่าวนี้ต่อมารู้จักกันในชื่อของ Pipelining และ Superscalar
- หลักการทั่ว ๆ ไป ของการประมวลผลชุดคำสั่งคือคอมพิวเตอร์จะโหลดชุดคำสั่งที่ต้องการประมวลผลมาจากหน่วยความจำ เราเรียกขั้นตอนนี้ว่าดึงหรือเฟ็ตช์คำสั่ง (Fetch)

คุณลักษณะของคอมพิวเตอร์แบบ RISC

- หลังจากนั้นก็จะทำการประมวลผลหรือเอ็กซิคิวต์ (Execute) แล้วจึงจะเฟ็ตช์ชุดคำสั่งต่อไปเข้ามาหลังจากประมวลผลชุดคำสั่งแรกเสร็จสิ้น
- วิธีการเพิ่มประสิทธิภาพด้วยการโอเวอร์แลปก็คือการเฟ็ตช์ชุดคำสั่งถัดไปเข้ามาก่อนที่ชุดคำสั่งแรกจะทำเสร็จ เราเรียกว่า พรีเฟ็ตช์ (Prefetch)
- อีก เทคนิคที่คล้ายกันเรียกว่าซุปเปอร์สเกลลาร์ (Superscalar) ซึ่งเป็นเทคนิคที่ใช้กับโปรเซสเซอร์ที่มีความสามารถรับชุดคำสั่งเข้าไปประมวลผลได้ทีละหลาย ๆ ชุดคำสั่งพร้อม ๆ กัน
- เทคนิคไปป์ไลน์และซุปเปอร์สเกลลาร์ไม่สามารถนำมาใช้กับ CISC ได้ เนื่องจาก CISC มีความยาวของชุดคำสั่งที่ไม่แน่นอน และ Addressing mode ที่หลากหลายและซับซ้อนทำให้ลดประสิทธิภาพของ CISC ลง

End of Chapter 5