

ระบบคอมพิวเตอร์และสถาปัตยกรรม (Computer System and Architecture)

Chapter 5

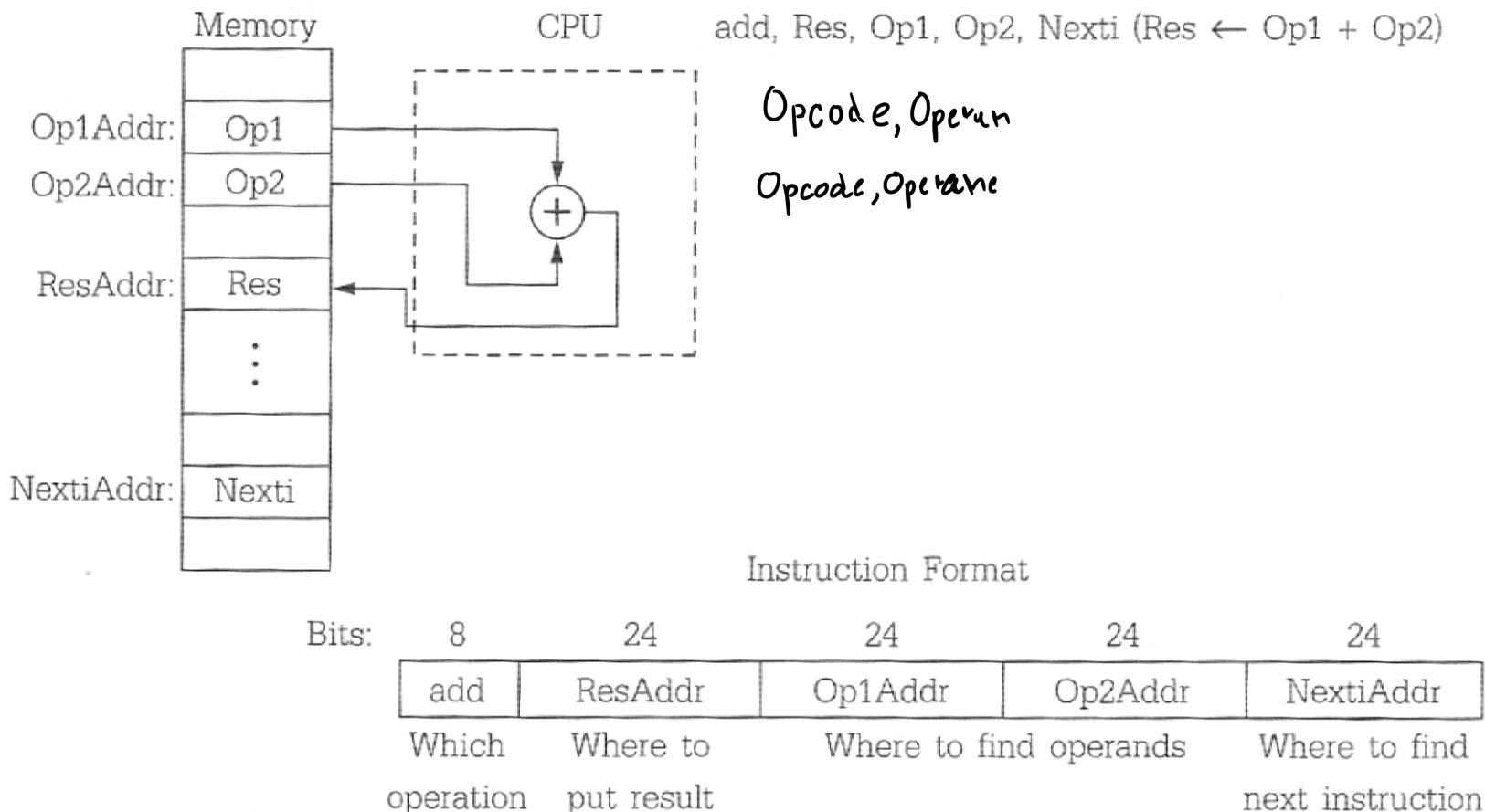
Chapter 5

ไมโครโปรเซสเซอร์เบื้องต้น

โดย ผู้ช่วยศาสตราจารย์ภาณุวัฒน์ เมฆะ

สาขาวิชาวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ มหาวิทยาลัยแม่โจ้

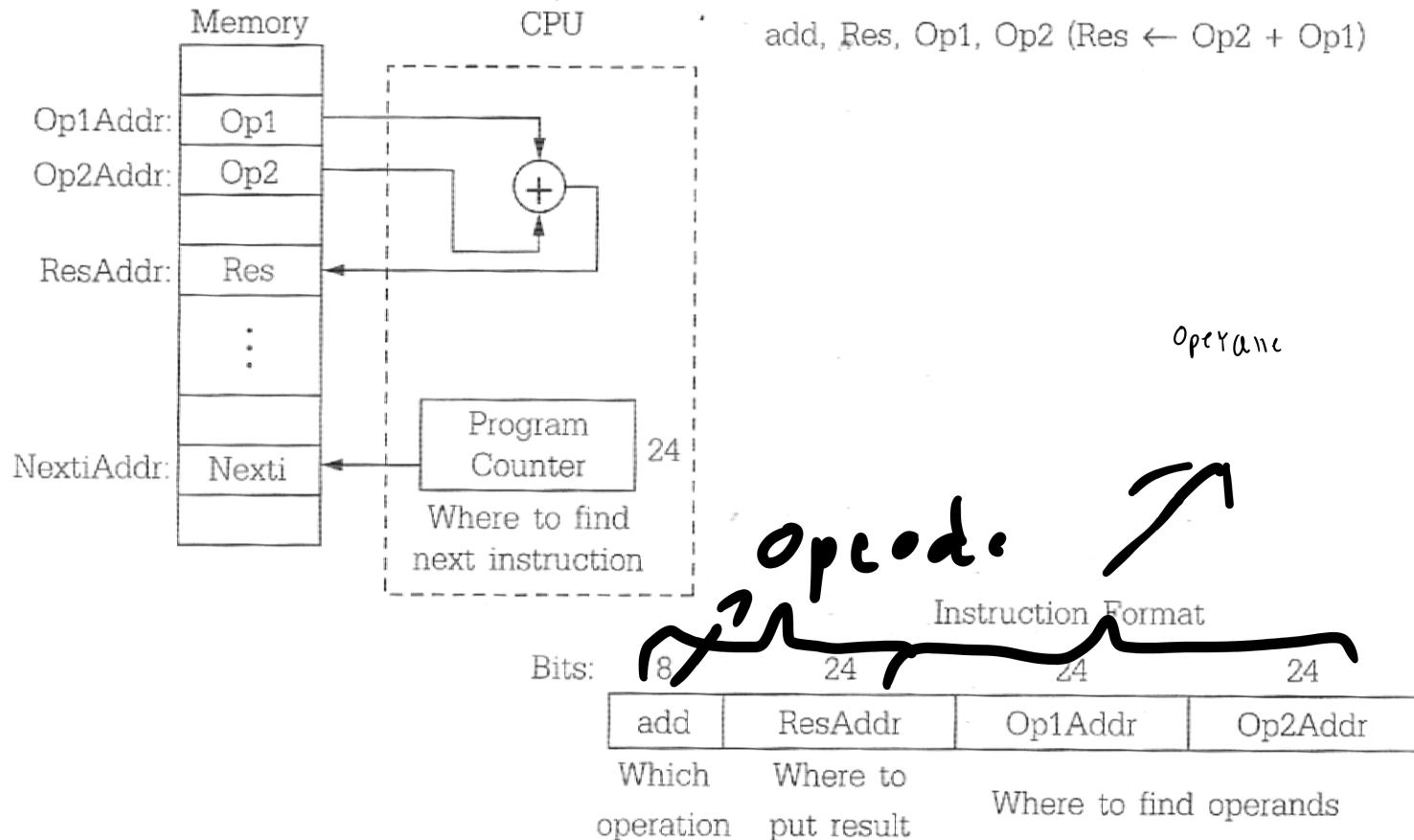
แนวคิดการออกแบบไมโครโปรเซสเซอร์



รหัสคำสั่งแบบ 4-Address Instruction

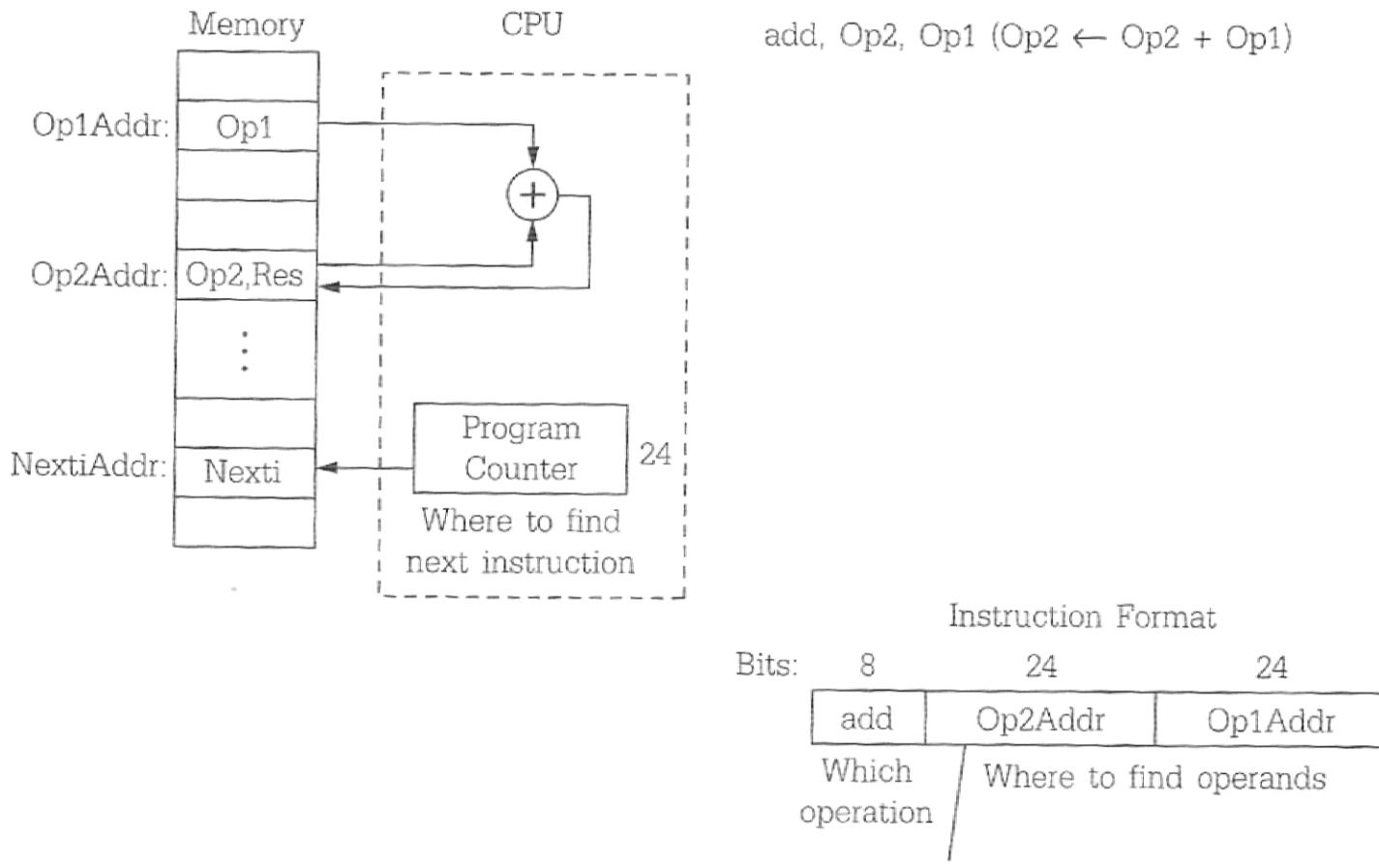
Exam

แนวคิดการออกแบบไมโครโปรเซสเซอร์



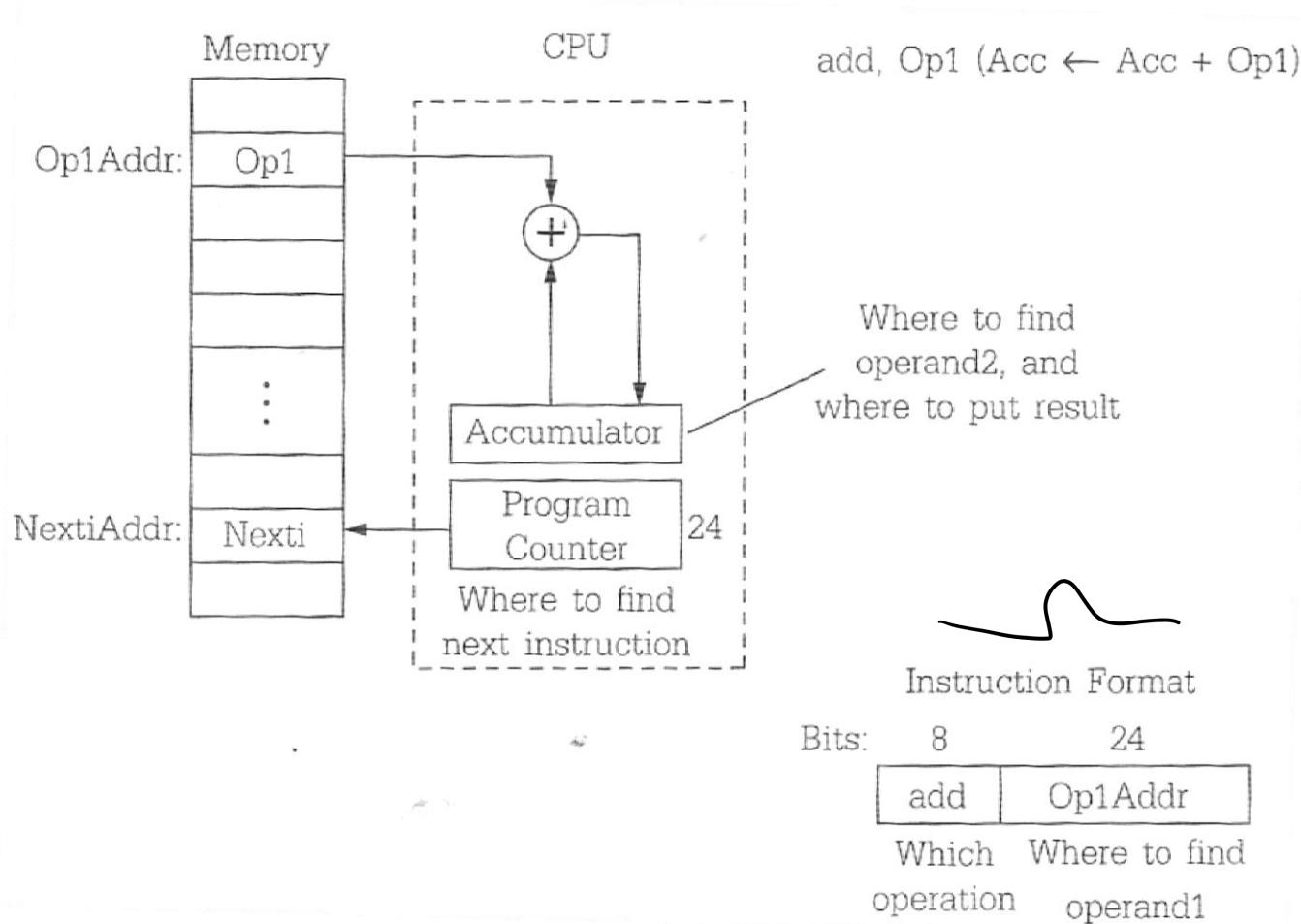
รหัสคำสั่งแบบ 3-Address Instruction

แนวคิดการออกแบบไมโครโปรเซสเซอร์



รหัสคำสั่งแบบ 2-Address Instruction
指令碼之類型 2-Address Instruction
Computer System and Architecture

แนวคิดการออกแบบไมโครโปรเซสเซอร์



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	$Y \leftarrow A - B$
MPY T, D, E	$T \leftarrow D \times E$
ADD T, T, C	$T \leftarrow T + C$
DIV Y, Y, T	$Y \leftarrow Y \div T$

(1) รหัสคำสั่งแบบ 3-Address Instruction

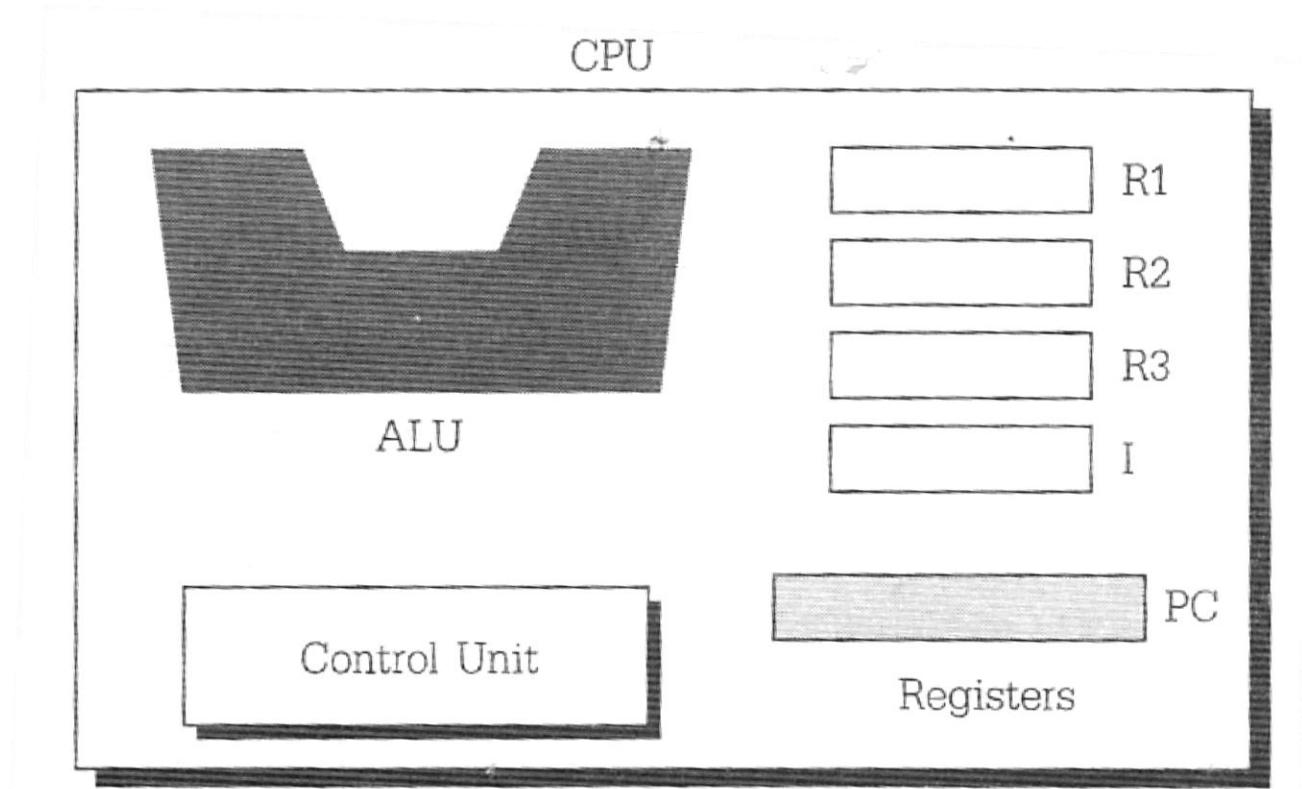
Instruction	Comment
MOVE Y, A	$Y \leftarrow A$
SUB Y, B	$Y \leftarrow Y - B$
MOVE T, D	$T \leftarrow D$
MPY T, E	$T \leftarrow T \times E$
ADD T, C	$T \leftarrow T + C$
DIV Y, T	$Y \leftarrow Y \div T$

(2) รหัสคำสั่งแบบ 2-Address Instruction

Instruction	Comment
LOAD D	$AC \leftarrow D$
MPY E	$AC \leftarrow AC \times E$
ADD C	$AC \leftarrow AC + C$
STOR Y	$Y \leftarrow AC$
LOAD A	$AC \leftarrow A$
SUB B	$AC \leftarrow AC - B$
DIV Y	$AC \leftarrow AC \div Y$
STOR Y	$Y \leftarrow AC$

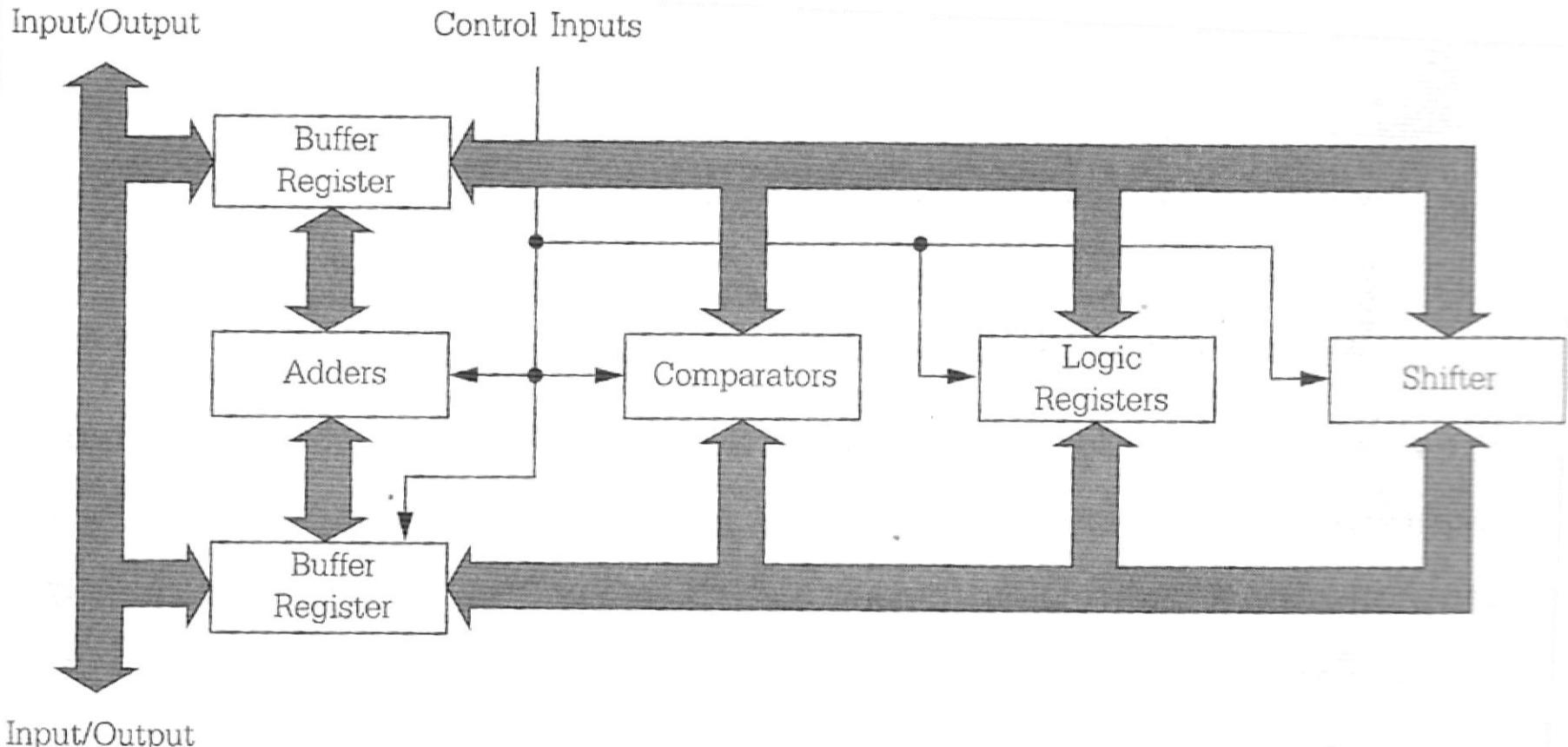
(3) รหัสคำสั่งแบบ 1-Address Instruction

โครงสร้างพื้นฐานของไมโครโปรเซสเซอร์



โครงสร้างอย่างง่ายภายในซีพียู

โครงสร้างพื้นฐานของไมโครโปรเซสเซอร์



โครงสร้างอย่างง่ายของ ALU

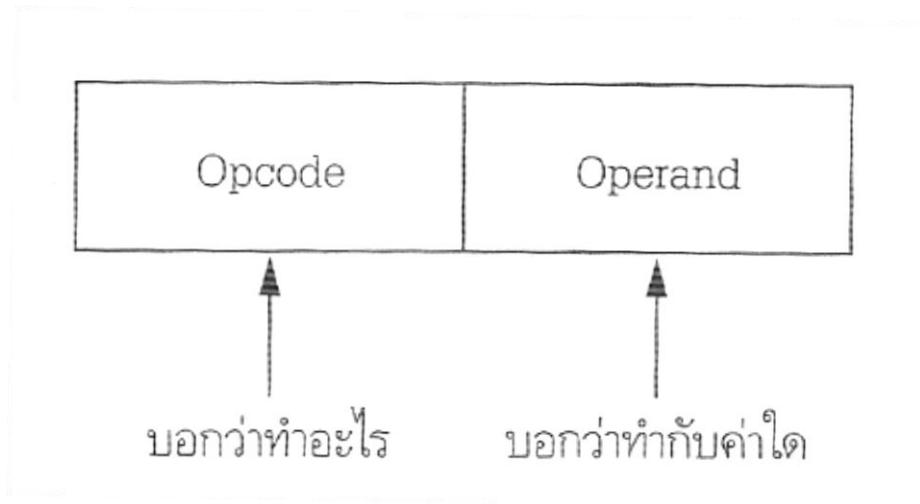
รูปแบบคำสั่งและโหมดของแอดเดรส

Ex.

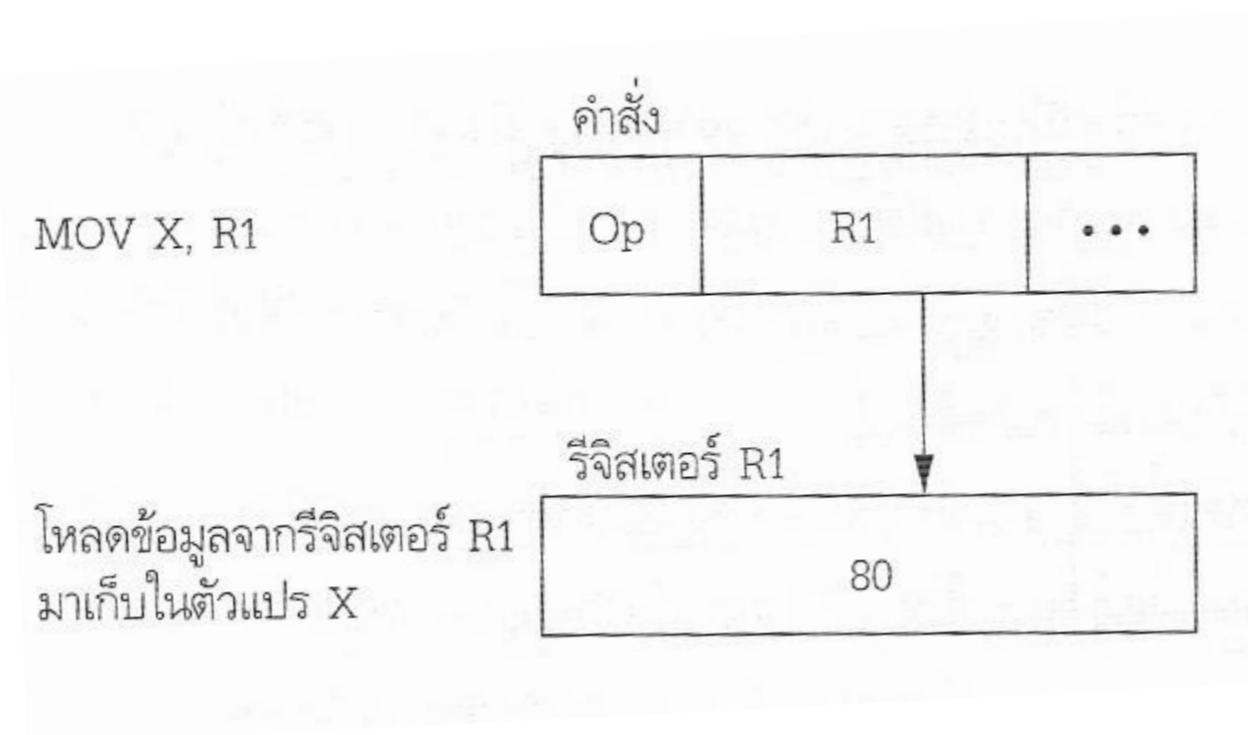
MOV Oper1, Oper2

: เป็นการใช้คำสั่ง โอนข้อมูลจาก โอเปอร์เรนด์ตัวที่ 2 ไปยัง
โอเปอร์เรนด์ตัวที่ 1

การกำหนดแอดเดรสแบบให้ค่าตรง (Immediate Address Mode)

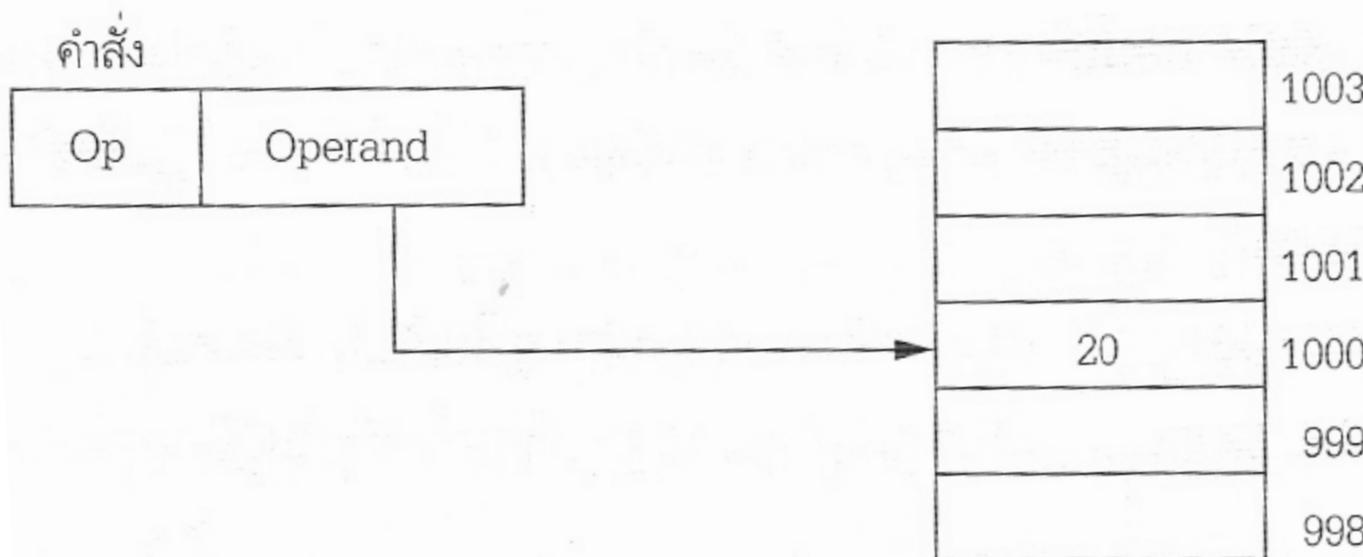


การกำหนดแอดเดรสโดยใช้รีจิสเตอร์ (Register Addressing Mode)



การกำหนดแอดเดรสโดยตรง (Direct Addressing Mode)

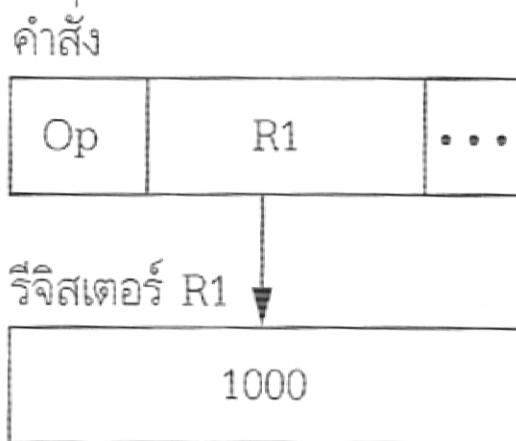
MOV X, [1000]; โหลดข้อมูลจากแอดเดรส 1000 มาเก็บใน X



การกำหนดแอดเดรสผ่านรีจิสเตอร์โดยอ้อม (Register Indirect Addressing Mode)

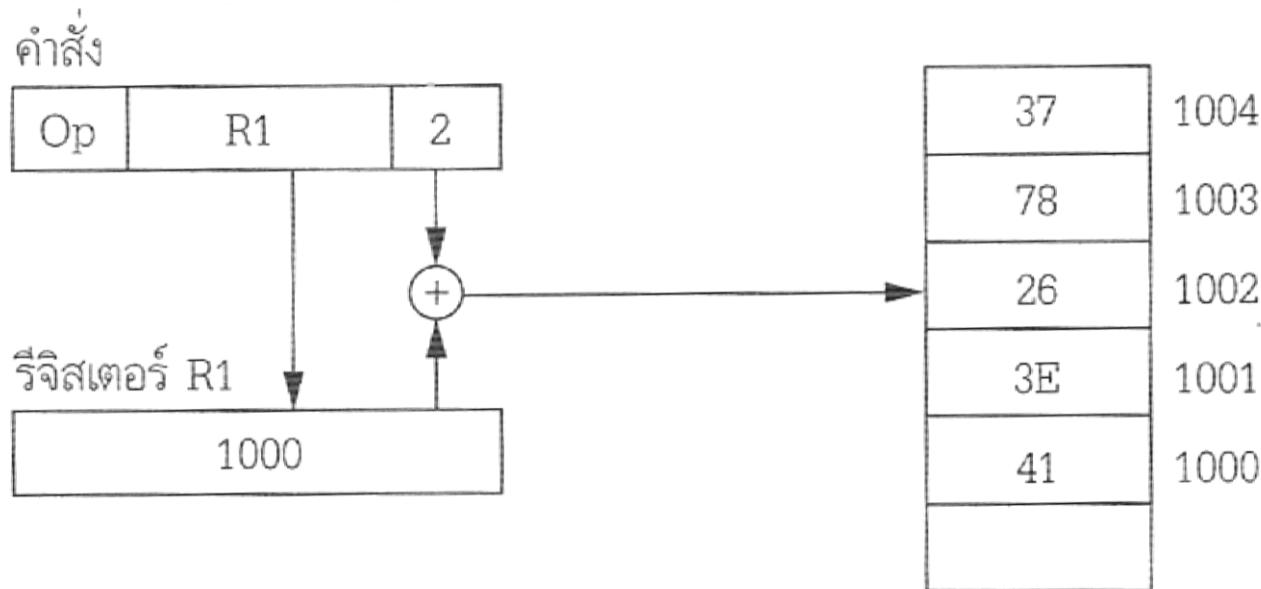
MOV X, [R1] ;

โหลดข้อมูลจากแอดเดรสที่เก็บอยู่ในรีจิสเตอร์ R1 มาเก็บในตัวแปร X
ถ้า R1 เก็บ 1000 จะทำให้ค่าในตัวแปร X มีค่าเป็น 80



60	1003
48	1002
12	1001
80	1000
35	999

การกำหนดแอดเดรสแบบแทนที่ (Indexed Addressing Mode)

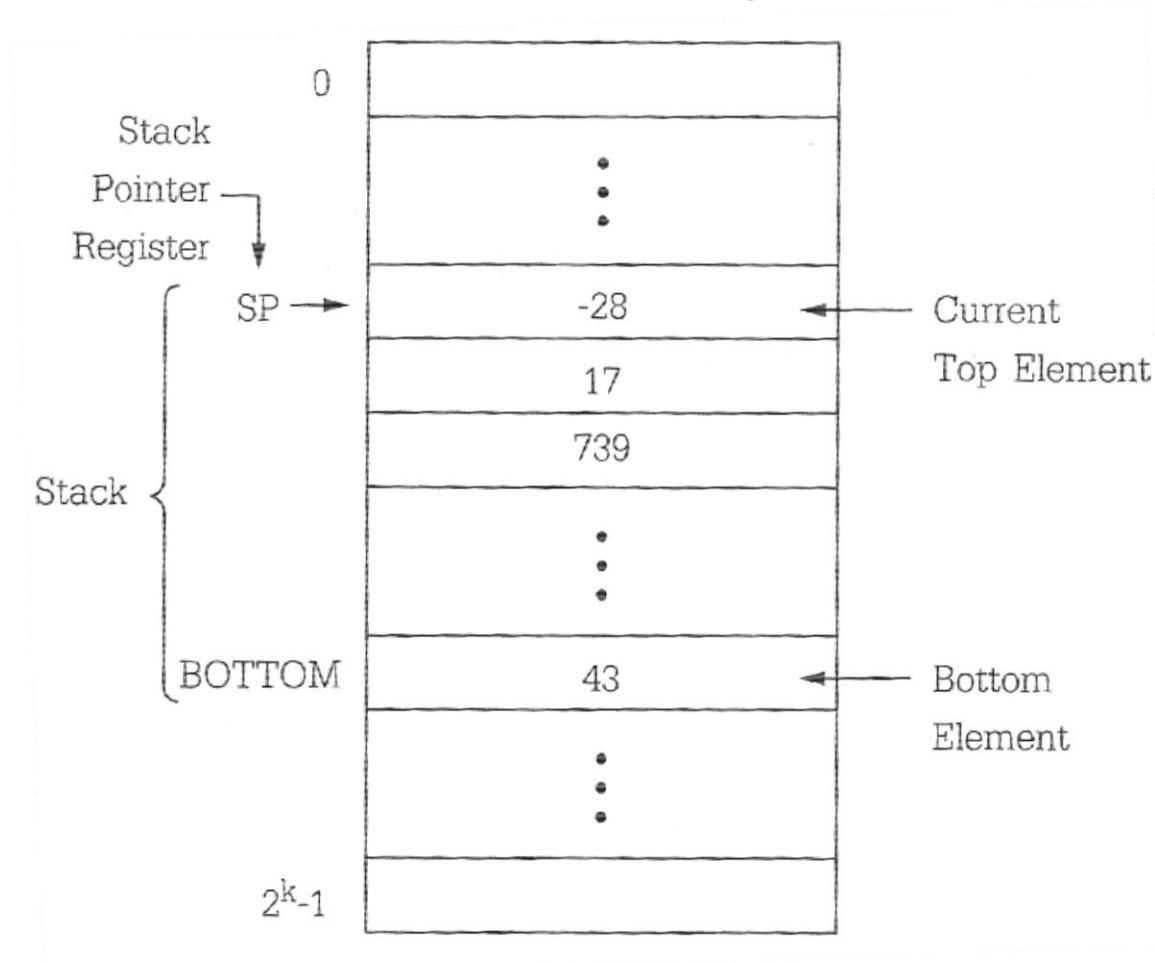


การกำหนดแอดเดรสแบบสัมพันธ์ (Relative Addressing Mode)

เป็นการอ้างถึงตำแหน่งหน่วยความจำที่อ้างอิงกับโปรแกรมคาน์เตอร์ (PC) โดยจะนำค่าคงที่ไปบวกกับค่าของโปรแกรมคาน์เตอร์ ค่าที่ได้จะเป็นแอดเดรสของหน่วยความจำที่ต้องการอ้างอิงถึง

การกำหนดแอดเดรสโดยใช้สแตก

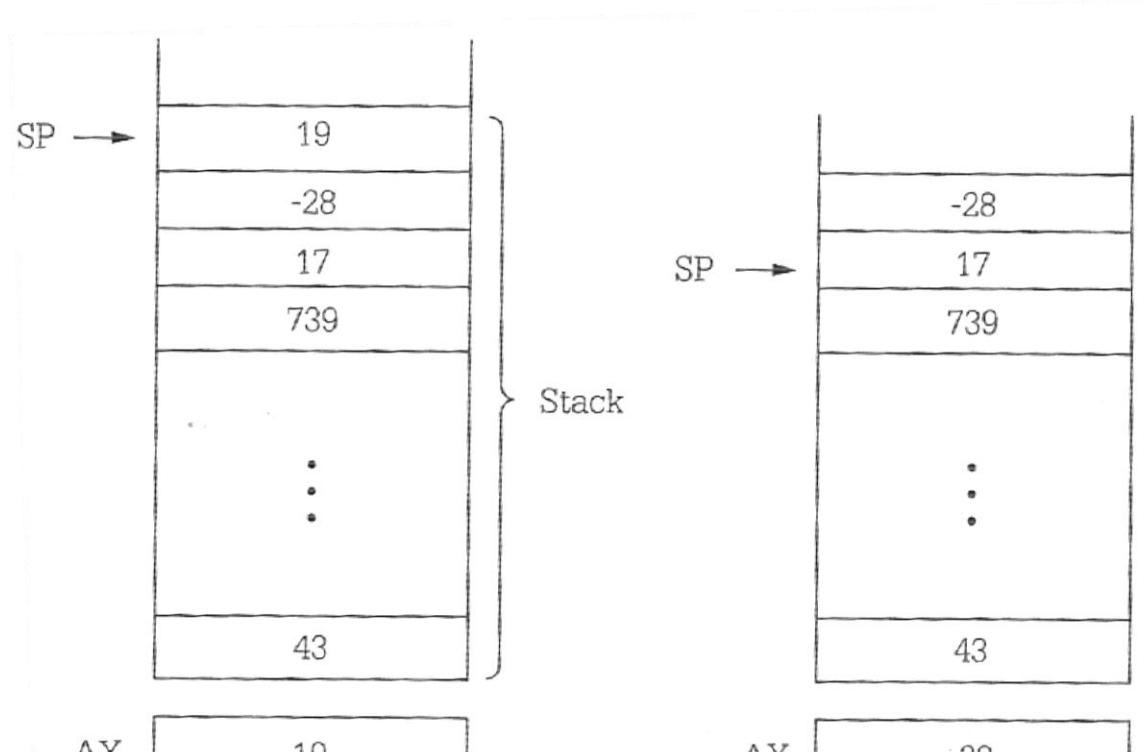
(Stack Addressing Mode)



ตัวอย่างสแตกในหน่วยความจำ

การกำหนดแอดเดรสโดยใช้สแตก

(Stack Addressing Mode)

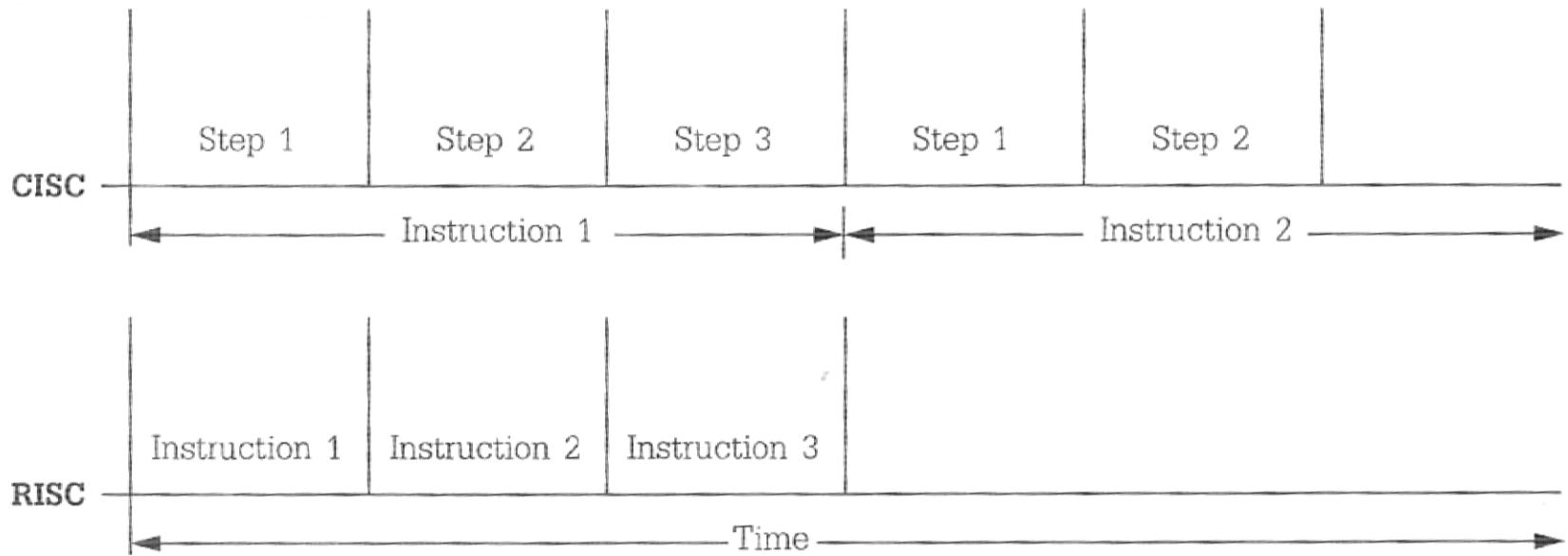


การดำเนินการหลังจากการ Push และการ Pop

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 ๆ ในการทำงานของคอมพิวเตอร์ต้องมองเป็นราคាដ้อยหน่วย
- สถาปัตยกรรมแบบ CISC มีจำนวน Addressing Mode หลายแบบ
- ลักษณะนี้ส่งผลให้ชุดคำสั่งที่ใช้มีขนาด และใช้เวลาในการประมวลผลแตกต่างกัน
- ผู้ใช้ยอมรับและพอใจ เนื่องจากผู้ใช้สามารถเพิ่มจำนวน operation ในโปรแกรมที่มีขนาดเท่าเดิมได้

คุณลักษณะของคอมพิวเตอร์แบบ RISC

- เดิมคอมพิวเตอร์ยุคแรกจะประมวลผลชุดคำสั่งทีละคำสั่งในหนึ่งช่วงเวลา
- นักออกแบบได้พยายามที่จะปรับปรุงอัตราการประมวลผลชุดคำสั่งให้เพิ่มมากขึ้น โดยใช้วิธีการ โอบเวอร์แลป (Overlap) ชุดคำสั่งให้มีมากกว่าชุดคำสั่งในการประมวลผลแต่ละครั้ง
- วิธีการ โอบเวอร์แลปดังกล่าว呢ีต่อมาเรียกกันในชื่อของ Pipelining และ Superscalar
- หลักการทั่ว ๆ ไป ของการประมวลผลชุดคำสั่งคือคอมพิวเตอร์จะโหลดชุดคำสั่งที่ต้องการประมวลผลมาจากหน่วยความจำ เราเรียกขั้นตอนนี้ว่าดึงหรือเฟชช์คำสั่ง (Fetch)

คุณลักษณะของคอมพิวเตอร์แบบ RISC

- หลังจากนั้นก็จะทำการประมวลผลหรืออี็กซิคิวต์ (Execute) และจึงจะเพ็ตช์ ชุดคำสั่งต่อไปเข้ามาหลังจากประมวลผลชุดคำสั่งแรกเสร็จสิ้น
- วิธีการเพิ่มประสิทธิภาพด้วยการ โอล์เวอร์แลป ก็คือการเพ็ตช์ชุดคำสั่งถัดไปเข้ามาก่อนที่ชุดคำสั่งแรกจะทำเสร็จ เราเรียกว่า พรีเพ็ตช์ (Prefetch)
- อีก เทคนิคที่คล้ายกันเรียกว่า ชูปเปอร์สเกลลาร์ (Superscalar) ซึ่งเป็นเทคนิคที่ใช้กับโปรเซสเซอร์ที่มีความสามารถรับชุดคำสั่งเข้าไปประมวลผลได้ทีละหลาย ๆ ชุดคำสั่งพร้อม ๆ กัน
- เทคนิคไปป์ไลน์และชูปเปอร์สเกลลาร์ไม่สามารถนำมาใช้กับ CISC ได้เนื่องจาก CISC มีความยาวของชุดคำสั่งที่ไม่แน่นอน และ Addressing mode ที่หลากหลายและซับซ้อนทำให้ลดประสิทธิภาพของ CISC ลง

End of Chapter 5