บทที่ 2

สถาปัตยกรรมซีพียู 8 บิต

พื้นฐานแนวความคิดของซีพียู 8 บิตโดยทั่วไป

1) สถาปัตยกรรมทางฮาร์ดแวร์

2) สถาปัตยกรรมทางซอฟต์แวร์

<u>สถาปัตยกรรมทางฮาร์ดแวร์</u>

1. โครงสร้าง
2. การทำงาน
3. รีจิสเตอร์

1. <u>โครงสร้างของซีพียู 8 บิต</u>

ซีพียู คือหน่วยที่ทำหน้าที่ควบคุมการทำงาน และจัดการข้อมูลให้กับระบบคอมพิวเตอร์ โดยนำคำสั่งจากหน่วยความจำเข้ามาถอดรหัส แล้วทำงานตามที่คำสั่งแต่ละคำสั่งระบุ

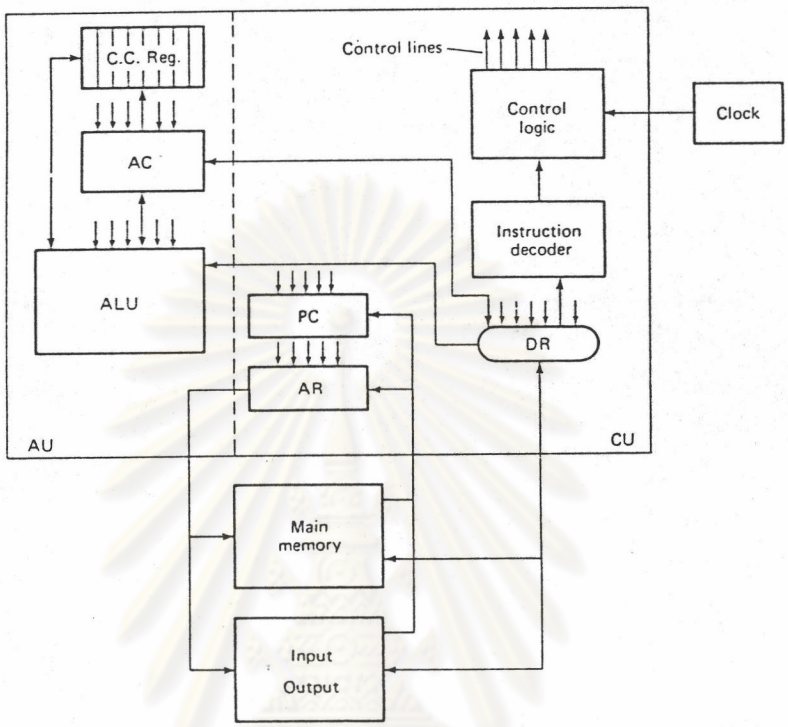
ซีพียู ประกอบด้วย 2 ส่วนหลักคือ

-หน่วยควบคุม (Control Unit)

-หน่วยทำงานคณิตศาสตร์และลอจิก (Arithmetic Logic Unit)

1.1 หน่วยควบคุม รับคำสั่งจากภายนอกเข้ามาทำให้เกิดการทำงานตามคำสั่งโดยการสร้างสัญญาณเวลา และสัญญาณควบคุมที่เหมาะสมแก่ส่วนประกอบอื่น

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 2.1 แผนภูมิซีพียู

หน่วยควบคุม ประกอบด้วย

-Address Register ชี้แอดเดรสที่ซีพียูกำลังแอ็กเซส

-Control Logic สร้างชุดสัญญาณไฟฟ้าที่จะ Enable ให้มีการถ่ายเทข้อมูลจากรีจิสเตอร์ หรือ หน่วยหนึ่งไปยังอีกหน่วยหนึ่ง สัญญาณไฟฟ้าเหล่านี้เรียกว่า Data Paths คำสั่งแต่ละคำสั่งจะสร้างชุดของสัญญาณควบคุมที่ไม่เหมือนกัน

-Data Register เป็นรีจิสเตอร์ชั่วคราว สำหรับข้อมูลที่เข้าหรือออกจากซีพียู ในบางซีพียู Data Register อาจหมายถึง Buffer Register กับ Instruction Register รวมกัน

-Instruction Decoder แปลงคำสั่งจาก Data Register ออกมาเป็นรูปแบบที่ Control Logic สามารถเข้าใจ

-Program Counter (PC) เป็นรีจิสเตอร์ที่เก็บแอดเดรสของคำสั่งถัดไปที่จะถูกนำมาใช้งาน ในการทำงานในปกติ PC จะเพิ่มค่าหนึ่งทุกครั้งหลังจาก แอ็กเซสหน่วยความจำ ยกเว้นกรณี Branch หรือ Jump

1.2 หน่วยทำงานคณิตศาสตร์ลอจิก จัดการข้อมูลทางคณิตศาสตร์และลอจิก ประกอบด้วย

-ALC (Arithmetic & Logic Circuit) ตัวจัดการข้อมูลทางคณิตศาสตร์และลอจิก

-Accumulator เป็นรีจิสเตอร์ชั่วคราว ที่ใช้ในระหว่างการคำนวณโดยใช้เก็บค่าหนึ่งค่าที่จะใช้ในการคำนวน และ เก็บผลหลังจากการคำนวณหรือการกระทำทางลอจิก

-Condition Code Register แต่ละบิตหมายถึงสภาวะของการทำงานบางอย่างที่เกิดขึ้น หรือสถานะของซีพียู เรียกว่า แฟล็ก

แฟล็ก เป็นพื้นฐานการทำงานของซีพียู ในเรื่องการตัดสินใจ แฟล็กที่พบบ่อยในซีพียู ได้แก่

CARRY - การทดจาก MSB (บิต 7)

ZERO - ผลของการทำงานเป็นศูนย์

OVERFLOW - การทดเข้าสู่ MSB

SIGN หรือ NEGATIVE - สะท้อนค่าบิต MSB ซึ่งในระบบ Two's Complement number '0' หมายถึงค่าบวก และ '1' หมายถึงค่าลบ

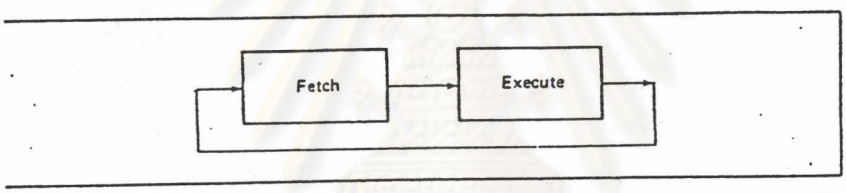
HALF CARRY - การทดจาก บิต 3 สู่ บิต 4

PARITY - แสดงถึงจำนวนบิตที่เป็น 1 ของผลจากการทำงานว่าเป็นจำนวนคู่ (Even parity) หรือจำนวนคี่ (Odd parity)

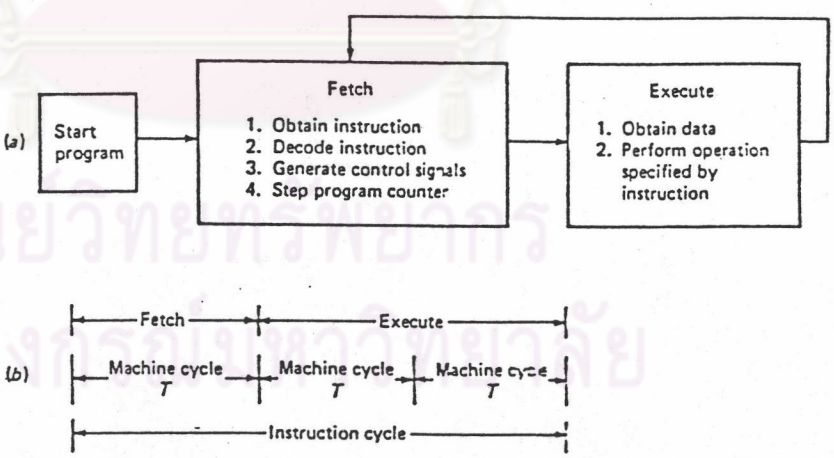
INTERRUPT ENABLE - ใช้ควบคุมการ Interrupt คืออนุญาตหรือไม่อนุญาตให้มีการ Interrupt

2. <u>การทำงานของซีพียู 8 บิต</u>

ซีพียูทำงานต่างๆให้แก่ระบบคอมพิวเตอร์ตามชุดคำสั่งที่เขียนไว้ในหน่วยความจำ ซีพียูใดๆ มีกลไกการทำงานที่เหมือนกันคือ รับคำสั่งและทำงานตามคำสั่ง การทำงานตามคำสั่งใดๆ ก็ตามประกอบด้วยวงจรของการทำงานที่เหมือนกันเพียง 2 อย่าง คือ เฟ็ตช์(Fetch) และ เอ็กซีคิวต์(Execute) ซึ่งรวมเรียกว่า Instruction Cycle



Two basic cycles of a microprocessor.



Processing an instruction. (*a*) Fetch-execute sequence; (*b*) time relationship.

<u>รูปที่ 2.2</u> ไซเคิลพื้นฐานของการทำงานของไมโครโปรเซสเซอร์: เฟ็ตช์, เอ็กซีคิวต์

2.1 เฟ็ตช์ คือการนำคำสั่งหรือข้อมูลจากภายนอกเข้ามาในซีพียู ขั้นตอนของการเฟ็ตช์ ได้แก่

-ซีพียู ส่งแอดเดรสจาก Program Counter ไปยัง Memory Address Register

-ซีพียู ส่งสัญญาณไปยังหน่วยความจำ เพื่อทำการอ่านข้อมูลจากหน่วยความจำ

-ข้อมูลจากหน่วยความจำ เข้าสู่ Memory Buffer Register

-ข้อมูลเข้ามาใน Instruction Register เพื่อที่จะถูกถอดรหัสคำสั่งเป็นการทำงาน

-Program Counter เพิ่มค่าหนึ่ง

2.2 เอ็กซีคิวต์ คือการทำงานของซีพียู ตามคำสั่งที่ถอดรหัสแล้ว

3. <u>รีจิสเตอร์</u>

รีจิสเตอร์ เป็นส่วนสำคัญอันหนึ่งของโครงสร้างซีพียู รีจิสเตอร์ ก็คือหน่วยความจำที่อยู่ในซีพียู ในส่วนของส่วนควบคุม

ซีพียูส่วนใหญ่ประกอบด้วยรีจิสเตอร์ต่างๆ ดังนี้

-Program Counter ขนาด 16 บิต ซีพียู 8 บิต โดยทั่วไปมีหน่วยความจำปกติได้สูงสุดขนาด 64 กิโลไบต์

-Accumulator ขนาด 8 บิต ซีพียูส่วนใหญ่มี Accumulator หนึ่งตัว แต่บางซีพียูก็มี Accumulator มากกว่าหนึ่งตัว

-Condition Code Register หรือ Status Register โดยมากมีขนาด 8 บิต ประกอบด้วยแฟล็กต่างๆ ดังได้กล่าวมาแล้ว แต่อาจไม่ได้ใช้งานครบทุกบิต

แอดเดรสรีจิสเตอร์ อื่นๆ เช่น

-Stack Pointer ไมโครโปรเซสเซอร์ 8 บิตส่วนใหญ่มีโครงสร้าง Stack อยู่ภายนอกซีพียู และมี Stack Pointer ซึ่งบอกตำแหน่งบนสุดของ Stack อยู่ภายในซีพียู

-Index Register ใช้ประโยชน์ในเรื่องการแอดเดรส

-รีจิสเตอร์ใช้งานทั่วไป ซึ่งอาจใช้งานเป็นได้ทั้ง Address Register และ Data Register

<u>สถาปัตยกรรมทางซอฟต์แวร์</u>

1. รูปแบบคำสั่งของซีพียู
2. วิธีแอดเดรสชิง
3. ประเภทของคำสั่งซีพียู
4. ผลของคำสั่งต่อแฟล็ก
5. ชุดคำสั่งของซีพียู

1. <u>รูปแบบคำสั่งของซีพียู</u> สิ่งที่ซีพียูนำเข้าอาจเป็นคำสั่งหรือข้อมูล ถ้าเป็นคำสั่ง ซีพียูจะส่งเข้าใน Instruction Register เพื่อถอดรหัส ถ้าเป็นข้อมูล ซีพียูจะส่งเข้าใน Data Register หรือ Arithmetic Unit

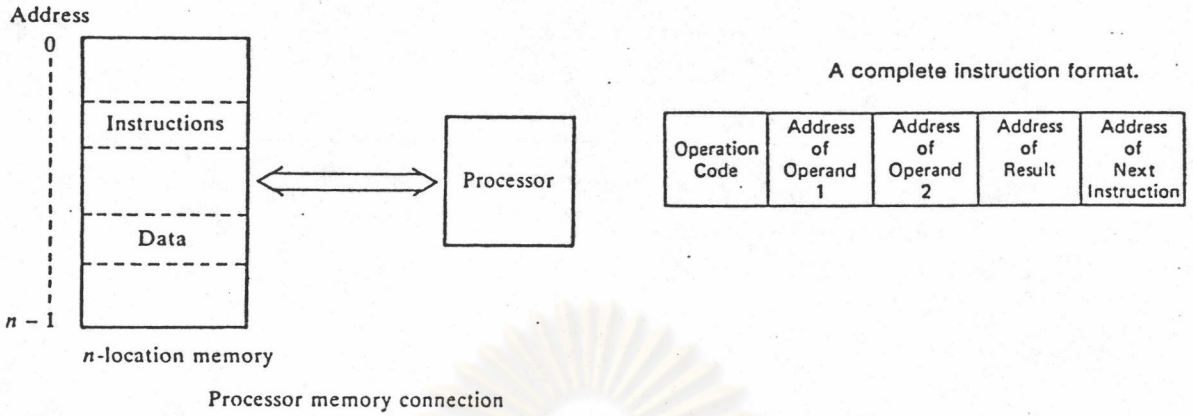
ในคำสั่งหนึ่งๆ จะต้องมีการระบุข้อมูลที่เพียงพอจะทำให้ซีพียูทำงานได้และอย่างต่อเนื่อง ได้แก่

ระบุการทำงาน เช่น บวก ลบ ย้ายข้อมูล

ระบุแหล่งของข้อมูลที่จะนำมาจัดการ จำนวนข้อมูลอาจเป็นหนึ่งหรือสองขึ้นกับชนิดของการทำงาน

ระบุจุดหมายปลายทางที่จะเป็นที่เก็บผลของการทำงาน

ระบุแอดเดรสของหน่วยความจำ ที่เก็บคำสั่งถัดไปที่จะได้รับการจัดการเพื่อให้เกิดความต่อเนื่องของการทำงานตามชุดคำสั่ง



รูปที่ 2.3 Complete Instruction Format

ในไมโครโปรเซสเซอร์ ข้อมูลเหล่านี้ส่วนมากจะบอกเป็นนัยอยู่ในคำสั่ง ทำให้คำสั่งมีขนาดสั้นลง และมีรูปแบบที่เรียกว่า One-Address Instruction

Operation Code	Address of Operand

รูปที่ 2.4 One-Address Instruction Format

ซีพียู จะได้รับข้อมูลหรือ โอเปอแรนด์ อันหนึ่งมาจาก Accumulator และผลจากการทำงานก็จะคงอยู่ที่ Accumulator ด้วย ส่วนแอดเดรสของคำสั่งถัดไป มาจากการเพิ่มค่าหนึ่ง Program Counter ทุกครั้งหลังการเฟ็ตช์ ดังนั้นข้อมูลที่ระบุในคำสั่งจะเหลือเพียงชนิดของการทำงาน และแหล่งของโอเปอแรนด์อีกอันหนึ่งเท่านั้น

2. วิธีแอดเดรสซิง การระบุแอดเดรสของโอเปอแรนด์ ที่เป็นข้อมูลหนึ่งในคำสั่ง One-Address Instruction ทำได้หลายวิธี เทคนิคในการระบุแอดเดรส ที่นิยมใช้ในไมโครโปรเซสเซอร์ 8 บิต ได้แก่

- Direct หรือ Absolute
- Indirect
- Indexed
- Immediate
- Relative
- Register-direct
- Register-indirect

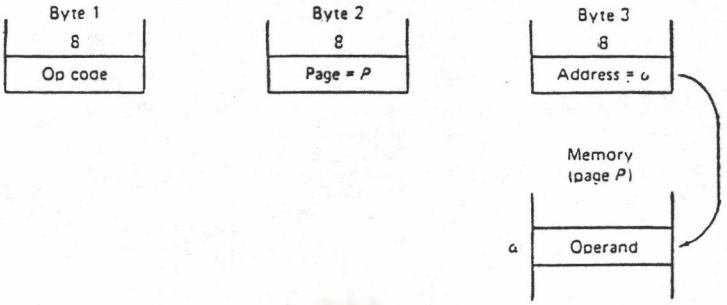
2.1 Direct Addressing หรือ Absolute Addressing เป็นการระบุแอดเดรสตรงๆ ในคำสั่ง สำหรับซีพียู 8 บิต ที่มีแอดเดรสขนาด 16 บิตก็จะต้องใช้ข้อมูล 2 ไบต์ในการระบุโอเปอแรนด์ แบบ Direct Addressing ทำให้คำสั่งค่อนข้างจะยาว

Direct Address อาจจะทำให้สั้นลงได้ โดยการแบ่งหน่วยความจำออกเป็นส่วนๆ เรียกว่า เพจ สำหรับแอดเดรส 16 บิตโดยทั่วไป จะแบ่งเป็น 256 เพจๆ ละ 256 ตำแหน่ง ส่วนของแอดเดรส ที่ระบุเพจ คือไบต์สูงจะเก็บอยู่ใน Page Register จึงเหลือที่ระบุอยู่ในคำสั่งอยู่ไบต์เดียว คือไบต์ต่ำของแอดเดรส เป็น Offset ของเพจนั้น

ถ้ามีการจำกัด ให้ซีพียูสามารถแอ็กเซส เพียงเพจใดเพจหนึ่งในเวลาใดเวลาหนึ่ง ก็อาจจะไม่จำเป็นต้องมี Page Register เช่นซีพียูอาจจะสามารถแอดเดรส หน่วยความจำโดยตรง เฉพาะเพจศูนย์เท่านั้น เทคนิคนี้เรียกว่า Zero-Page

เทคนิคอีกอย่างหนึ่ง โดยการจำกัดให้ ซีพียูสามารถที่จะแอดเดรสหน่วยความจำที่อยู่ในเพจเดียวกันกับคำสั่งได้โดยตรง เรียกว่า Current-Page Addressing

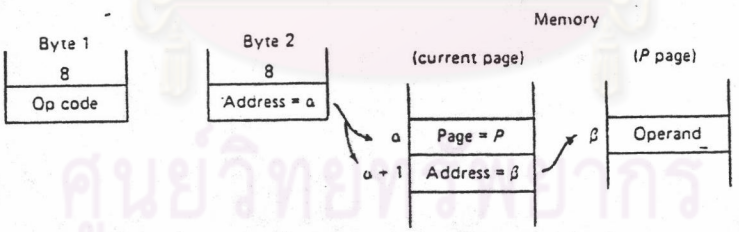
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 2.5 Direct Addressing

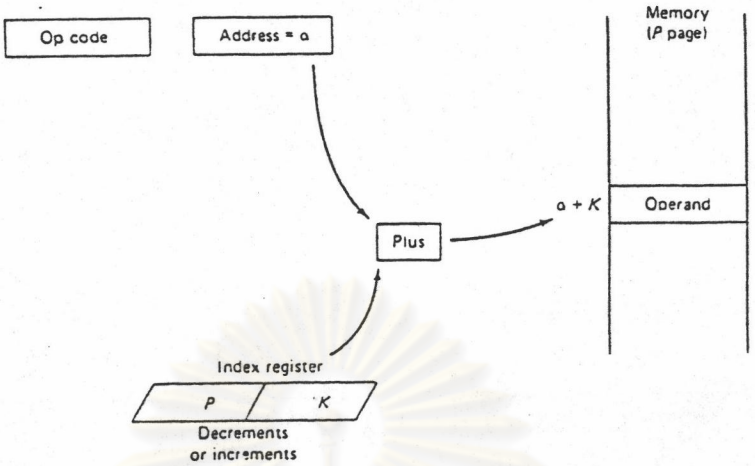
2.2 Indirect Addressing แอดเดรสที่ระบุอยู่ในคำสั่งไม่ใช่แอดเดรสของโอเปอแรนด์โดยตรง แต่เป็นแอดเดรสของแอดเดรสของโอเปอแรนด์ อีกต่อหนึ่ง การทำงานของคำสั่งที่ระบุแอดเดรสโดยวิธีนี้ จึงย่อมช้ากว่าแบบ Direct เพราะจะต้อง เฟ็ตช์แอดเดรสถึง 2 ครั้ง

ขนาดของ Indirect Address ที่ระบุอยู่ในคำสั่งก็เช่นเดียวกับ Direct Addressing คือสามารถทำให้สั้นลงจาก Absolute Address เป็น Page-Offset Address ได้เช่น Zero-Page Indirect Addressing



รูปที่ 2.6 Indirect Addressing

2.3 Indexed Addressing ค่า Effective Address ได้จากการบวกค่าแอดเดรสที่ระบุในคำสั่งกับค่าใน Index Register

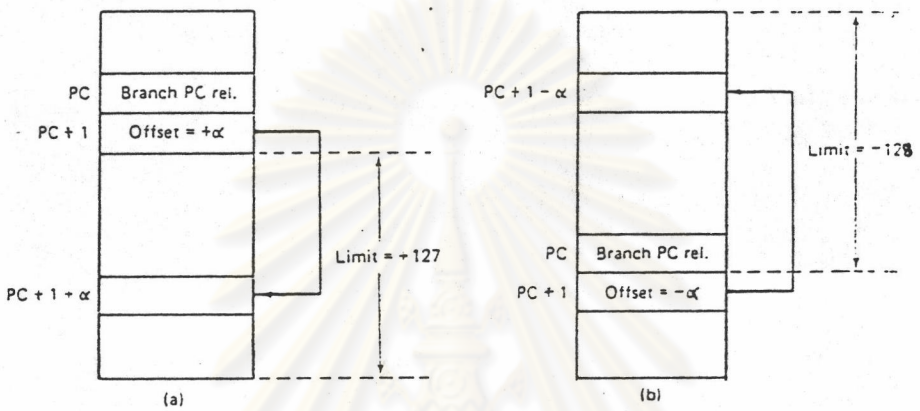


รูปที่ 2.7 Indexed Addressing

2.4 Immediate Addressing ข้อมูลเป็นส่วนหนึ่งในคำสั่งนั้น

รูปที่ 2.8 Immediate Addressing

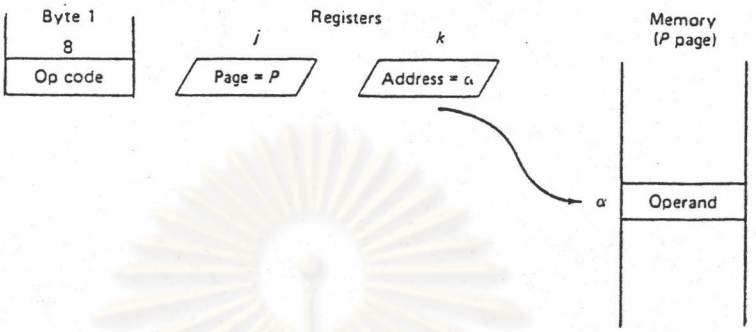
2.5 Relative Addressing ค่าแอดเดรสที่ระบุในคำสั่งเป็นตำแหน่งเปรียบเทียบกับ Program Counter ดังนั้นซีพียูจะต้องเพ็ตช์ค่านั้นมาบวกหรือลบกับ PC ก่อน จึงจะได้เป็น Effective Address



รูปที่ 2.9 Relative Addressing

2.6 Register-Direct Addressing คล้ายกับ Direct Addressing แต่จะระบุแอดเดรสของรีจิสเตอร์ หรือระบุรีจิสเตอร์ แทนการระบุแอดเดรสของหน่วยความจำ และไม่มีการเฟ็ตช์ข้อมูล เพราะอยู่ในซีพียูอยู่แล้ว

2.7 Register-Indirect Addressing คล้ายกับ Indirect Addressing แต่เป็นการระบุ Address Register ที่เก็บค่า Effective Address แทนที่จะเป็น Indirect Address ของหน่วยความจำ จึงไม่มีการเฟ็ตช์แอดเดรสเลย มีแต่การเฟ็ตช์ข้อมูล



รูปที่ 2.10 Register-Indirect Addressing

วิธีแอดเดรสชิงแต่ละวิธี มีประโยชน์และข้อดีแตกต่างกันไป ในการสร้างโปรแกรม จำเป็นจะต้องใช้หลายวิธีร่วมกัน

3. ประเภทของคำสั่งซีพียู คำสั่งในการทำงานของซีพียู อาจแบ่งออกได้เป็นประเภทต่างๆ

- Data Manipulation Instructions
- Data Transfer Instructions
- Program Manipulation Instructions
- Status Management Instructions
- Combined Instructions

3.1 Data Manipulation Instructions เป็นคำสั่งกระทำการเปลี่ยนแปลงข้อมูลด้วยวิธีการต่างๆซึ่งมักจะเป็นการทำงานของหน่วย ALU ได้แก่

Arithmetic Operations

Logical Operations

Shift Operations

Comparison Operations

Special-Purpose Operations เช่น Decimal Arithmetic

3.2 Data Transfer Instructions ทำการเคลื่อนย้ายข้อมูลจากที่หนึ่งไปยังอีกที่หนึ่ง โดยไม่มีการเปลี่ยนแปลงข้อมูลนั้น ได้แก่

Memory Transfer

Input/Output

Internal Transfer

Stack

3.3 Program Manipulation Instructions ย้าย Program Control จากจุดหนึ่งไปยังอีกจุดหนึ่ง คือการเปลี่ยนแปลงค่าของ Program Counter ซึ่งทำให้การทำงานตามชุดของคำสั่งไม่เป็นไปตามลำดับต่อเนื่องตามปกติ ได้แก่

Unconditional Jump/Branch

Conditional Jump/Branch

Subroutine

Halt and No Operation

3.4 Status Management Instructions เปลี่ยนแปลงค่าของ Status ต่างๆ โดยไม่มีผลกระทบต่อข้อมูล หรือลำดับการทำงานของคำสั่ง

3.5 Combined Instructions ซีพียูบางตัว มีคำสั่งที่สามารถทำงานหลายๆ อย่างได้ด้วยคำสั่งเดียว เช่น นอกจากจะกระทำทางคณิตศาสตร์ หรือลอจิกแล้ว ยังสามารถทราบค่าของแฟล็ก Carry เลื่อนค่า และทำคำสั่งอื่นต่อไปภายใต้เงื่อนไขบางประการ คำสั่งเช่นนี้ อาจเพิ่มประสิทธิภาพในการทำงานและลดขนาดของโปรแกรม แต่อย่างไรก็ตามก็มีความยากในการใช้งานสำหรับโปรแกรมเมอร์ เทคนิคในเรื่องของไมโครโปรแกรม ทำให้การอิมพลีเมนต์ Combined Instruction ทำได้ง่ายขึ้นแต่ยังคงมีความยากอยู่ที่การเลือกคำสั่งที่จะนำมารวมกันเป็นคำสั่งเดียวให้เกิดประโยชน์ได้อย่างแท้จริง

4. ผลของคำสั่งต่อแฟล็ก ผลของคำสั่งที่มีต่อแฟล็ก แตกต่างกันบ้างในแต่ละซีพียู แต่ในคำสั่งทั่วไป เช่น ADD, SUBTRACT มีผลต่อแฟล็กเสมอโดยเฉพาะอย่างยิ่งแฟล็ก Carry คำสั่ง INCREMENT, DECREMENT มักไม่มีผลต่อแฟล็ก Carry จึงอาจใช้ในการควบคุมลูปที่ทำ Multiple-Precision Arithmetic

คำสั่งลอจิกเช่น AND, OR, EXCLUSIVE-OR มักมีผลต่อแฟล็ก Sign และแฟล็ก Zero

คำสั่ง SHIFT อาจมีหรือไม่มีผลต่อแฟล็กใดๆ ยกเว้นแฟล็ก Carry

คำสั่งเปรียบเทียบมีหน้าที่โดยตรงกับการมีผลต่อ Status Flags

คำสั่งทำงานพิเศษ เช่น MULTIPLY, DIVIDE, COMPLEMENT มีผลต่อแฟล็กแตกต่างกันในแต่ละซีพียู

คำสั่งที่มีผลต่อ StatusFlags ส่วนใหญ่เป็นคำสั่งประเภท Data Manipulation

5. ชุดคำสั่งของซีพียู ไม่มีซีพียูใดที่บรรจุคำสั่งทั้งหมดที่เป็นไปได้ไว้ โดยมากจะมีคำสั่งแตกต่างอยู่ระหว่าง 20 ถึง 200 คำสั่ง ซึ่งบางตัวแตกต่างกันเฉพาะแต่ในเรื่องของวิธีแอดเดรสซิง เท่านั้น

ชุดคำสั่งขนาดใหญ่อาจมีผลทำให้โปรแกรมสั้นลงและทำงานเร็วขึ้น แต่ขณะเดียวกัน คำสั่งที่มีขนาดยาวขึ้น การถอดรหัสคำสั่งก็ยุ่งยากขึ้น การเรียนรู้และใช้งานให้มีประสิทธิภาพก็ยากขึ้นด้วย

โปรแกรมเมอร์มักไม่ค่อยได้ใช้คำสั่งจำนวนมากๆจากชุดคำสั่งที่ใหญ่ โดยความเป็นจริงแล้วมีโปรแกรมเมอร์จำนวนน้อยมากที่จะสามารถใช้งานชุดคำสั่งที่มีมากกว่า 100 คำสั่ง ได้อย่างมีประสิทธิภาพ แต่ขณะเดียวกันชุดคำสั่งขนาดเล็กก็มักทำให้เกิดความงุ่มง่ามในการใช้คำสั่ง ชุดคำสั่งที่ค่อนข้างเหมาะสมคือ ชุดคำสั่งที่มีคำสั่งแตกต่าง อยู่ระหว่าง 40 ถึง 80 คำสั่ง ชุดคำสั่งที่ใหญ่เป็นพิเศษมักจะได้รับการออกแบบมาให้เหมาะสมกับงานเฉพาะอย่างเท่านั้น