



Instruction Set



What is an Instruction Set?

- เก็บในลักษณะของ Binary Word ใช้บอกร่างการทำงานที่ต้องการให้ Microprocessor ทำงาน
- เมื่อพูดถึง Instruction Set จะหมายถึงคำสั่งที่ Microprocessor รู้จักและสามารถทำงานตามคำสั่งนั้นได้
- ขนาดของคำสั่งใน Microprocessor มีขนาดเท่ากับ Data หรือมีขนาด เป็น 2 หรือ 3 เท่าของขนาด Data



What is an Instruction Set?

- ส่วนของ คำสั่ง สามารถแบ่งเป็น 2 ส่วน คือ
 - Microprocessor ต้องทำงาน อะไร (Op Code (Operation Code))
 - ทำกับข้อมูลใด หรือหน่วยความจำตำแหน่งใด (Data / Address)



Mnemonics Code

- เป็นคำย่อของคำสั่งใน Microprocessor เพื่อให้สามารถจดจำคำสั่งได้ง่ายขึ้น
 - Load --> LD
 - Increment --> INC
 - Halt --> HLT
- Mnemonics Code ไม่สามารถนำไปส่งงาน Microprocessor ได้โดยตรง ต้องทำการเปลี่ยนให้เป็น Binary Word เสียก่อน
- Binary Word ที่ Microprocessor เข้าใจเรียกว่า Machine Language



Basic Instruction Type

- Data transfer
- Exchanges, block transfer, and searches
- Arithmetic and logic
- Rotates and shifts
- General-purpose and CPU control
- Bit set, bit reset, and bit test
- Jump
- Call, returns, and restart
- Input and Output

An Introduction to Microprocessor
Instructions

5



Data Transfer

- คำสั่งที่อยู่ในกลุ่ม Data transfers เช่น load, move, store
- เป็นการย้ายข้อมูลระหว่าง Register กับ Register, Register กับ หน่วยความจำ , หน่วยความจำกับ หน่วยความจำ
- ตัวอย่างเช่น
 - LD r,r' : r <-- r')
 - LD r,(HL) : r <-- (HL)
 - LD r,n : r <-- n

An Introduction to Microprocessor
Instructions

6

Exchanges, block transfer, and searches

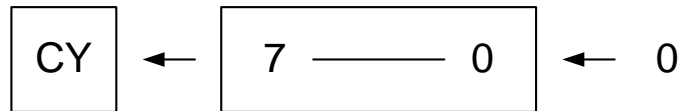
- Exchange เป็นคำสั่งพิเศษสำหรับการ โอนย้ายข้อมูล บางครั้งเรียกว่า Swap ตัวอย่างเช่น
 - EX DE,HL : DE <--> HL
- block transfer เป็นการโอนย้ายข้อมูล ในลักษณะเป็นกลุ่ม จากตำแหน่งในหน่วยความจำ จากตำแหน่งหนึ่งไปอีกตำแหน่งหนึ่ง
- Search instruction เป็นการค้นหาข้อมูล จากกลุ่มของข้อมูล

Arithmetic and Logic

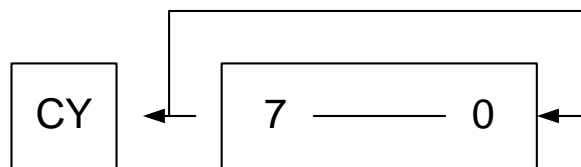
- ใช้ในกรณีที่ ต้องการจะเปลี่ยนแปลงข้อมูลใน Register หรือ ในหน่วยความจำ
- ในกรณีที่เป็นการกระทำระหว่างค่า 2 ค่า ค่าแรก เป็นข้อมูลที่อยู่ใน Accumulator ส่วนค่าที่ 2 เป็นค่าใน Register หรือข้อมูลในหน่วยความจำ
- ตัวอย่างเช่น
 - ADC (HL) : A <-- A + (HL) + CY
 - DEC r : r <-- r - 1
 - AND n : A <-- A.n

■ Rotate and shift instruction

■ SLA r



■ RLC r



An Introduction to Microprocessor
Instructions

9

General-purpose and CPU control

- CPL : $A \leftarrow \sim A$ (A 1'S Complement)
 - DI : Interrupt Disable
 - EI : Interrupt Enable
 - NOP : No Operation
- general-purpose instruction เป็นคำสั่งที่เกิดขึ้นกับ Register เพียงตัวเดียว ไม่ได้เกิดขึ้นจากการกระทำระหว่าง Register 2 ตัว

An Introduction to Microprocessor
Instructions

10

Bit set, Reset and Test Instruction

- ใช้ในกรณีที่ต้องการตรวจสอบบาง bit ของข้อมูล หรือต้องการ Set หรือ Reset บาง bit ของข้อมูล
- ตัวอย่างเช่น
 - SET b,r : $r_b \leftarrow 1$

Jump Instruction

- เป็นคำสั่งที่ใช้ส่งให้ Microprocessor กระโดดไปทำงาน ณ.ตำแหน่งอื่นของหน่วยความจำ
- คำสั่ง Jump มี 2 ลักษณะคือ
 - Absolute Jump เช่น
JMP nn : $PC \leftarrow nn$
 - Condition Jump เช่น
JP P nn : $PC \leftarrow nn$ (if sign bit is positive)



Call Instruction

- Call ทำงานเหมือนกับคำสั่ง Jump แต่คำสั่ง Call จะมีการเก็บค่าของ PC ณ.ตำแหน่งปัจจุบันลงสู่ Stack ด้วย
- ใช้ในกรณีที่ ต้องการเรียกใช้งาน โปรแกรมย่อย (Subroutine)
- มี 2 ประเภทคือ
 - Unconditional Call Instruction
 - Conditional Call Instruction



Call Instruction

- มี 2 ประเภทคือ
 - Unconditional Call Instruction
CALL nn : (SP-1) <-- PC_H; (SP-2) <-- PC_L;
PC <-- nn
 - Conditional Call Instruction
CALL NZ, nn : (SP-1) <-- PC_H; (SP-2) <-- PC_L;
PC <-- nn
- มีการ Call เกิดขึ้นเมื่อ Zero bit ของ Status Register บ่งบอกว่าไม่เท่ากับ ศูนย์



Return Instruction

- Return Instruction ใช้ในกรณีที่ทำงานใน Subroutine เสร็จเรียบร้อยแล้ว แล้วต้องการจะกลับไปทำในส่วนของ โปรแกรมหลัก
- ตัวอย่างคำสั่ง
 - RET : $PC_L \leftarrow (SP); PC_H \leftarrow (SP+1)$
- ค่าที่นำออกจาก Stack เป็นค่าตำแหน่งของ Program Counter ก่อนเข้ามาทำงานใน Subroutine



Input and Output

- เป็นคำสั่งที่ใช้ในกรณีที่ต้องการติดต่อระหว่าง Microprocessor กับ Hardware Device ภายนอก
- ตัวอย่างของคำสั่ง
 - IN A, (n) : $A \leftarrow (n)$
 - OUT (n),A : $(n) \leftarrow A$
- Microprocessor บางตัวไม่มีคำสั่ง Input / Output สามารถติดต่อกับ Hardware Device ภายนอก โดยใช้หลักการของ Memory Mapped I/O

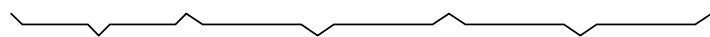
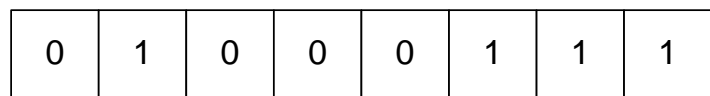
Microprocessor's Addressing Mode

■ Inherent OR Implied Addressing

- เป็นการโอนย้ายข้อมูลใน Register , Register Pair หรือตำแหน่งของหน่วยความจำที่อ้างโดย Register Pair
- คำสั่งเป็นขนาด 8-Bit
- ตัวอย่างเช่น

LD A,B

LD A,(HL)



LD

Register B

Register A

An Introduction to Microprocessor
Instructions

17

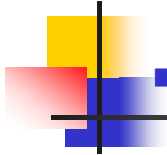
■ Immediate Addressing

- เป็นคำสั่งการอ้างตำแหน่งในลักษณะที่ ข้อมูล Byte แรก เป็น Op-code ส่วน Byte ที่ 2 เป็นข้อมูล
- ข้อมูลที่ต้องอยู่ในตำแหน่งของหน่วยความจำที่ต่อจาก ส่วนของ Op-code
- ข้อมูลมีขนาด 1 หรือ 2 Byte
- Immediate Addressing Mode ใช้ 2 Microprocessor Cycle ; Cycle แรกเป็นการ Fetched คำสั่ง หลังจากทำการ Execute คำสั่ง , Cycle ที่ 2 เป็นการอ่านข้อมูล
- ตัวอย่างคำสั่งเช่น

LD A,n

An Introduction to Microprocessor
Instructions

18

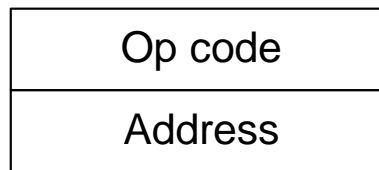


Direct Addressing

- เป็นคำสั่งในการอ้างตำแหน่งในลักษณะที่ ข้อมูล Byte แรกเป็น Op-code ส่วนของมูลถัดไป เป็นตำแหน่งของ หน่วยความจำ หรือตำแหน่งของ I/O
- ขนาดของ Op-code รวมกับคำสั่งมีขนาด 2 - 3 Byte ใช้ 3 - 4 Microprocessor Cycle
- ตัวอย่างคำสั่งเช่น

LD 000EH,A

LD A,000EH



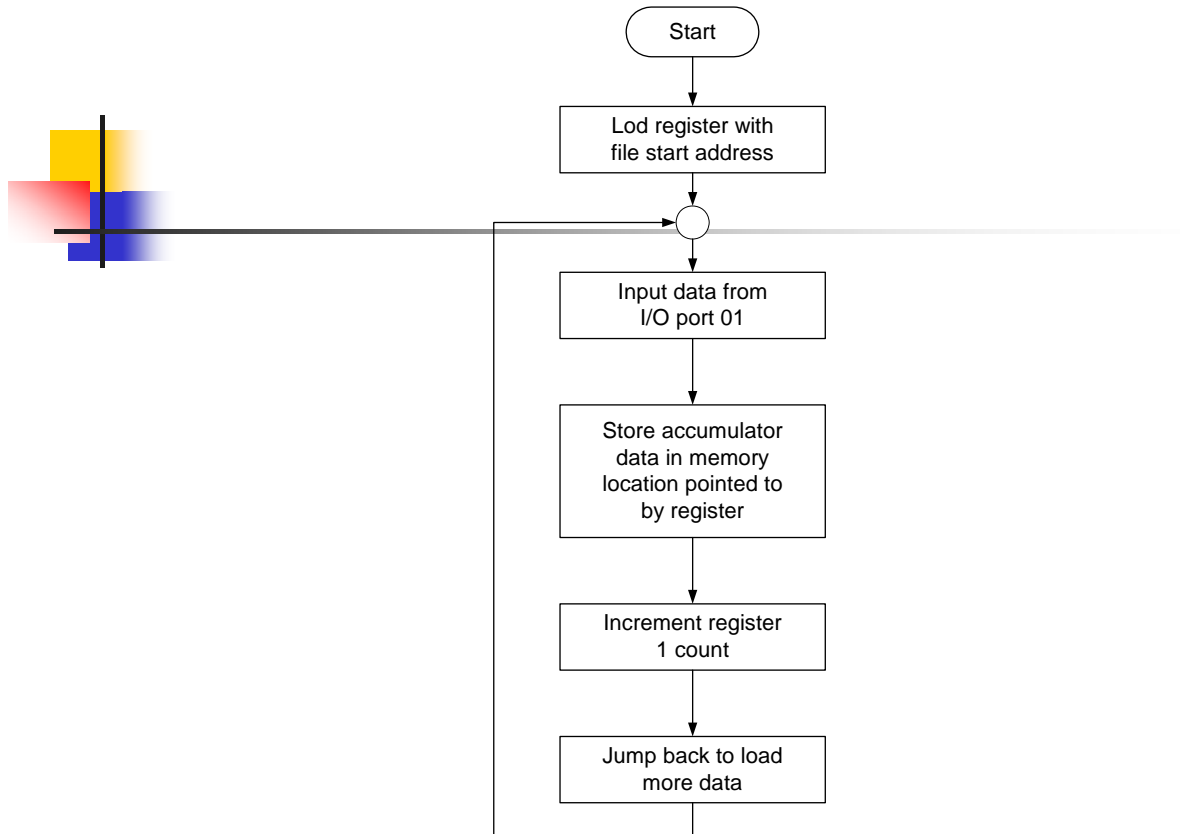
1st Byte

2nd Byte



Register Indirect Addressing

- คำสั่งใช้เพียง Byte เดียว โดยในคำสั่งจะประกอบไปด้วย Op-code และ Register ที่ใช้เป็นตัวชี้ยังตำแหน่งต่าง ๆ
- อาจใช้เพียง Register ตัวเดียว หรือ Register Pair
- เหมาะสำหรับการโอนย้ายข้อมูลหรือการอ้างถึงตำแหน่งของหน่วยความจำ ที่ใช้งานบ่อย
- ช่วยในการต้องการอ่านหรือเขียนข้อมูลในกลุ่มของหน่วยความจำ ที่ติดต่อกัน



■ Index Addressing

- Microprocessor 8-bit บางตัว มี Register พิเศษ ขนาด 16 bit ที่เรียกว่า Indexed register
- ใช้ได้กับ Microprocessor ที่มี Index register เท่านั้น
- ผลรวมระหว่าง Index register กับค่าตัวเลข ใช้ในการชี้ตำแหน่งของหน่วยความจำ
- ตัวเลขขนาด 8-Bit ที่บวกเข้ากับ Index Register เรียกว่า ค่า offset
- ตัวอย่างคำสั่งเช่น
ADD A,(IX + n)

Relative Addressing

- เป็นการอ้างตำแหน่งที่สัมพันธ์ กับตำแหน่งของ PC เดิม
- ใช้หลักการของ การทำ 2'S Complement ในการบอกตำแหน่งที่สัมพันธ์กัน
- กรณีที่ตัวเลขที่ใช้บอกตำแหน่ง MSB เป็น 1 แสดงว่า เป็น Backward ถ้าเป็น 0 แสดงว่า Forward
- ตัวอย่างเช่นการทำคำสั่ง Jump แบบ Relative
JR Z,0FAH

Memory Address	Content
0000	IN
0001	00
0002	AND A
0003	02
0004	JR Z
0005	FA
0006	IN
0007	01
0008	RET

- Address ที่ต้องการกระโดด จาก 0006 กลับไป 0000
- นำมาลบกัน $0006 - 0000 = 6$
- นำมาทำ 2'S

0000	0110	
1111	1001	1'S
0000	0001	
<u>1111</u>	<u>1010</u>	2'S
F	A	