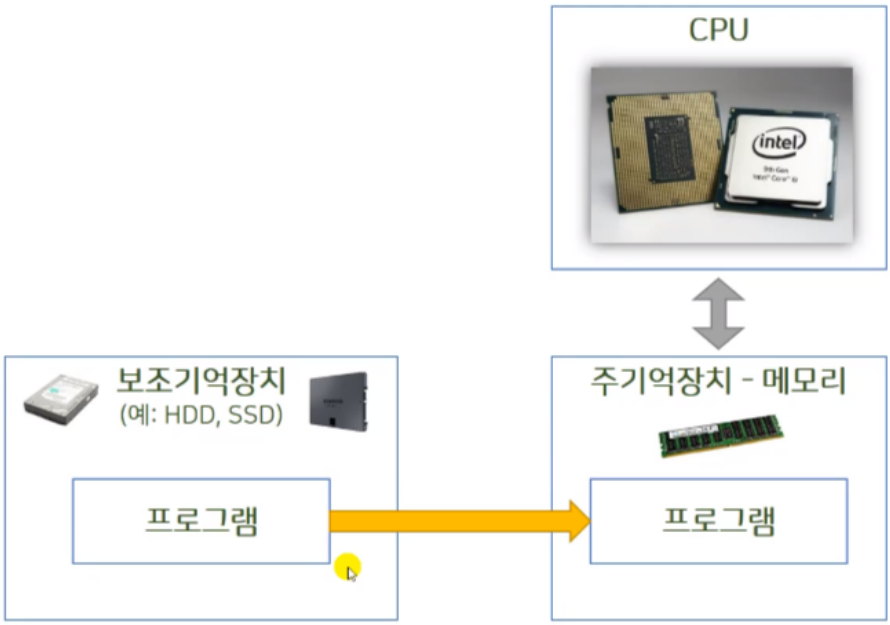
프로그램이 시작되는 과정



프로그램 -> CPU에게 일련의 행동들을 지시하는 것

고수준 프로그래밍 언어 High level programming language	A = 3 B = 4 C = A + B
어셈블리 언어 Assembly language	니모닉 Mnemonic LOAD [10] ADD [11] STORE [12]
기계어 Machine code	100110 0000001010 110011 0000001011 111010 0000001100

기계어 - CPU가 이해하는 방식의 언어(사람이 이해하기 힘듬)

어셈블리 언어 - 기계어로 되어 있는 명령들을 인간이 인식할 수 있는 단어로 바꾼 것

고수준 프로그래밍 언어 - 일반적으로 사용되는 프로그래밍 언어

CPU 명령어 집합
Instruction Set

Functional Group	Example Mnemonics
Move Instructions	MOV
Math Instructions	MUL DIV ADD SUB
Logic Instructions	AND IOR XOR NEG
Rotate/Shift Instructions	ASR LSR SL
Bit Instructions	BSET BCLR BTG BTST
Compare/Skip/Branch	BTSC BTSS CPBEQ CPBGT
Flow Control Instructions	BRA CALL RCALL REPEAT
Shadow/Stack Instructions	LNK POP PUSH ULNK
Control Instructions	NOP CLRWDT PWRSAV RESET
DSP Instructions	MAC LAC SAC SFTAC

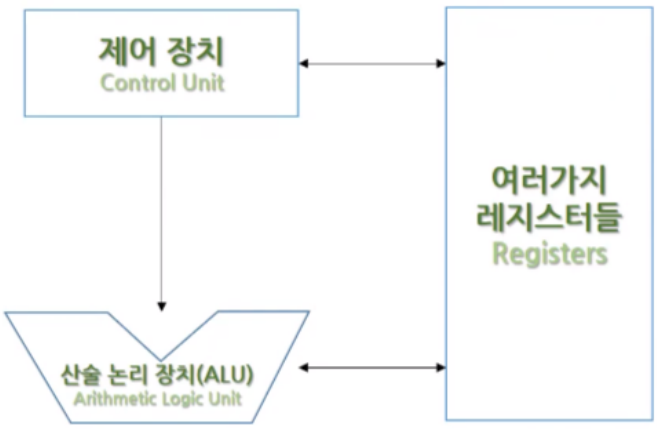
[출처] Instruction Set Architecture Overview
"Microchip Developer Help" webpage
http://microchipdeveloper.com/16bit:instruction-set-architecture

- **ADD** - 숫자 두 개를 더한다
- **COMPARE** - 숫자들끼리 비교한다
- **IN** - 키보드 같은 입력장치로부터 정보를 입력받는다
- **JUMP** - 지정된 메모리 주소로 점프한다
- **JUMP IF** - 조건에 따라 점프한다
- **LOAD** - 메모리에서 CPU로 정보를 가져온다
- **OUT** - 출력 장치로 정보를 출력한다
- **STORE** - 메모리에 정보를 저장한다

[출처] Instruction set
'Computer Hope' webpage
https://www.computerhope.com/jargon/i/instset.htm

CPU에서 수행하는 최소단위의 명령어

CPU의 구성 요소들



레지스터 종류	기능
주소 레지스터	읽거나 쓸 메모리 주소 저장
프로그램 카운터	다음 명령어의 메모리 주소 저장
데이터 레지스터	메모리에서 읽어온 데이터 저장
명령어 레지스터	메모리에서 읽어온 명령어 저장
어큐뮬레이터	연산에 사용되는 데이터 저장

"초보 프로그래머가 꼭 알아야 할 컴퓨터 동작 원리", 김종훈, 한빛미디어

산술 논리 장치(ALU)

- 실제 연산 작업을 하는 곳

제어장치

- 일을 할 때 관리의 역할

레지스터(중요)

- ALU가 일을 하기 위해 세팅을 해주는 역할
- 기능에 따라 다양한 레지스터들이 존재한다.
- 기능은 비슷하나 구조 상 메모리와 많은 차이를 보임