

## 명령(instruction)

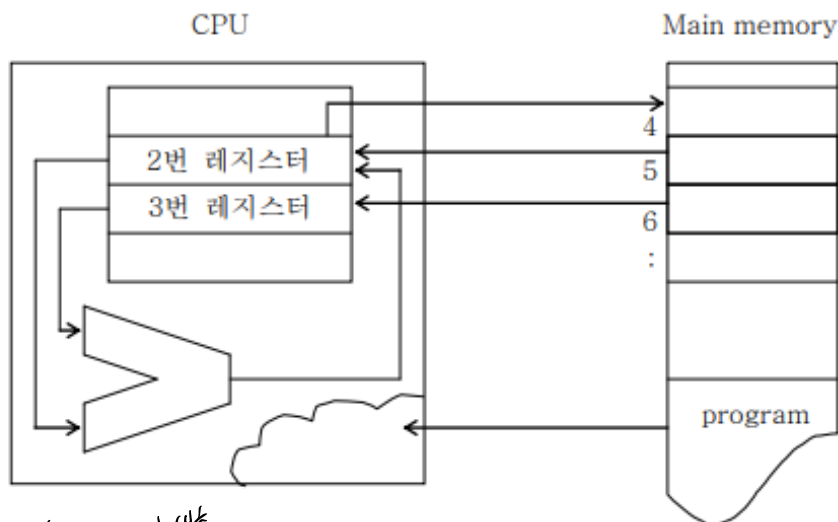
↑ CPU가 직접 수행하는 명령!

그럼 메모리의 특정 위치 5번지와 6번지에 있는 값을 더해서 4번지에 저장하는 프로그램을 작성해 보자. 일단 극히 제한된 세가지 명령만을 수행할 수 있는 가상의 CPU를 설정하고 이 CPU로 기계어와 어셈블리어 그리고 고급언어로 어떻게 프로그램이 작성되는지 간단히 살펴본다. 우리가 만든 가상의 CPU는 다음의 세가지 명령을 수행한다.

## 기계어 프로그램

명령어의 집합(프로그램)은 '기계어'로 작성될 수도 있고, '고급언어'로 작성될 수도 있다.

위의 세가지 명령을 이용해서 기계어 프로그램을 짜보면, 우선 메모리의 5번지와 6번지의 값을 두 개의 레지스터로 옮기는 명령과 이 레지스터의 값을 더하는 명령과 더해진 결과를 다시 레지스터에 저장하고, 최종적으로 이 더해진 레지스터의 값을 메모리의 4번지로 저장하는 순서로 진행된다.



load 레지스터 번호 메모리 위치

010	010	101	=> load, 레지스터 번호, 메모리 위치
2	2	5	=> 메모리 5번지의 값을 2번 레지스터로 load해라
010	011	110	=> load, 레지스터 번호, 메모리 위치
2	3	6	=> 메모리 6번지의 값을 3번 레지스터로 load해라
001	010	011	=> 덧셈, 레지스터 번호, 레지스터 번호
			=> 2번, 3번 레지스터의 값을 더해서 2번 레지스터에 저장해라
100	010	100	=> store, 레지스터 번호, 메모리 위치
			=> 2번 레지스터의 값을 메모리 주소 4번지에 저장해라

→ 짜 코드를

근접언어로 -  
정확하다면, 이진수 형태의  
영문상으로 번역된다.