

5주차 3차시 컴퓨터 기능과 동작

【학습목표】

1. 컴퓨터의 기본적인 특징과 기능을 설명할 수 있다.
2. 프로그램내의 명령어가 수행되는 과정을 설명할 수 있다.

학습내용1 : 기능과 특징

1. 컴퓨터의 기능과 동작

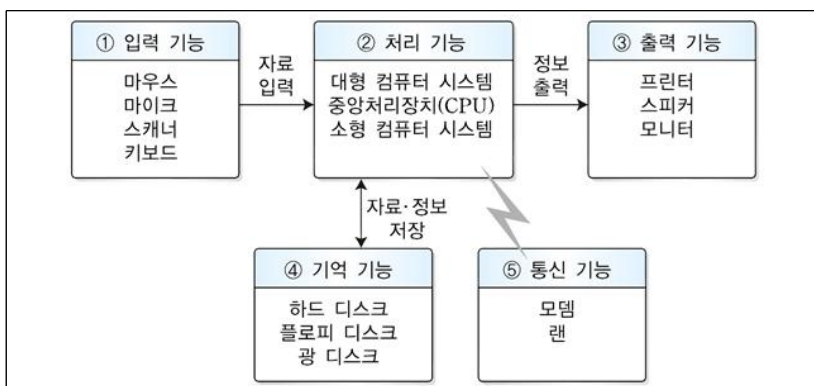
1) 컴퓨터의 기본적인 특징

- ① 신속성 : 컴퓨터의 처리 속도는 고속이어서 1초에 수억 번의 작업을 수행함
- ② 신뢰성 : 입력된 자료를 오류 없이 정확하게 처리해줘서 그 결과를 믿을 수 있음
- ③ 정확성 : 입력된 자료가 정확하고 사용 방법이 올바르면 처리 결과는 정확함
- ④ 대용량성 : 작은 기억장치 하나에 방대한 양의 자료를 저장할 수 있음
- ⑤ 공유성 : 인터넷에 연결된 컴퓨터는 정보를 많은 사람이 공유하게 함

2) 컴퓨터의 기본적인 기능

- ① 입력 기능 : 데이터나 프로그램을 컴퓨터 내부로 읽어오는 기능
- ② 기억 기능 : 입력된 데이터나 프로그램, 중간 결과 및 처리된 결과를 기억함
- ③ 처리 기능 : 산술이나 논리 연산 등을 수행하며, 프로그램을 읽고 해석하여 각 장치에 필요한 지시를 함
- ④ 출력 기능 : 처리 결과를 기호, 문자, 그림, 음성 등의 형태로 외부로 냄
- ⑤ 통신 기능 : 다른 컴퓨터와 연결하여 자료를 입력 또는 처리하거나 출력함

3) 컴퓨터의 기능



학습내용2 : 시스템의 동작

1. 컴퓨터 시스템의 동작

1) 컴퓨터 시스템의 작동 과정



* 펌웨어(Firmware)

- 컴퓨팅과 공학 분야에서 특정 하드웨어 장치에 포함된 소프트웨어로, 소프트웨어를 읽어 실행하거나, 수정되는 것도 가능한 장치를 뜻함

2) 펌웨어의 예

- PC에서 흔히 볼 수 있는 바이오스
- 애플 매킨토시에 사용되는 오픈 펌웨어
- 롬 집적 회로의 컴퓨터 프로그램
- EPROM 칩은 특정한 하드웨어로 수정이 가능한 프로그램이지만, 응용 프로그램으로는 수정이 되지 않음
- 산업용 프로그램 각종 계측기, 오실로스코프 등의 프로그램
- 각종 로봇의 프로그램
- 휴대전화의 프로그램
- PDA의 프로그램
- 내비게이션의 프로그램
- 홈 네트워크
- MP3 플레이어의 프로그램
- PMP
- DVR
- DMB
- 디지털 카메라의 프로그램
- USB 기억 장치, USB 허브 등 각종 USB 장치
- 가전 전자제품의 프로그램 전기밥솥, 냉장고

3) 컴퓨터에서 프로그램이 실행 시, 구성 장치들이 수행하는 동작

① 실행

- CPU가 주기억장치에서 프로그램 코드를 읽어서 실행함

② 데이터 저장

- 프로그램 실행 결과로서 얻어진 데이터를 주기억장치에 저장함

③ 데이터 이동

- 보조기억장치에 저장되어 있는 프로그램과 데이터 블록을 주기억장치로 이동시키는 기능

④ 데이터 입력

- 사용자가 키보드를 통해 보내는 명령이나 데이터를 읽어 들임

⑤ 데이터 출력

- CPU가 처리한 결과 값이나 기억장치의 내용을 출력함

⑥ 제어

- 프로그램이 순서대로 실행되도록 또는 필요에 따라 실행 순서를 변경하도록 조정하며, 각종 제어 신호들을 발생시킴

2. 컴퓨터에서 프로그램의 역할

* 컴퓨터의 동작이란?

- 프로그램내의 명령어를 읽고, 처리하고, 그리고 저장하는 과정을 통해서 정해진 순서대로 프로그램을 실행

1) 프로그램내의 명령어가 수행되는 과정

① 명령어 인출

- 명령어를 읽어가지고 오는 단계

② 명령어 해독

- 명령어의 내용을 결정하는 단계

③ 명령어 실행

- 해독 명령어의 내용을 수행하는 과정으로 처리 결과는 저장함

④ 프로그램 카운터 증가 단계

- 다음 명령어를 수행하기 위한 단계

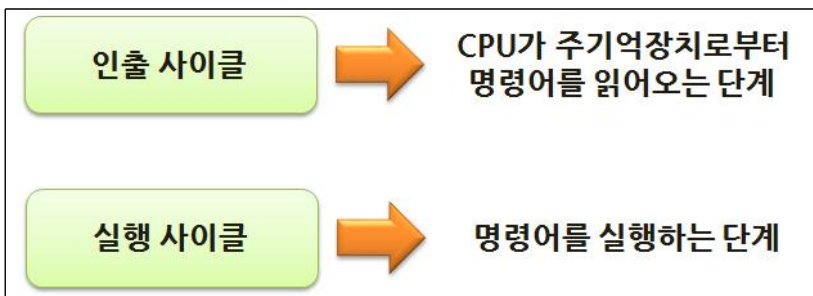
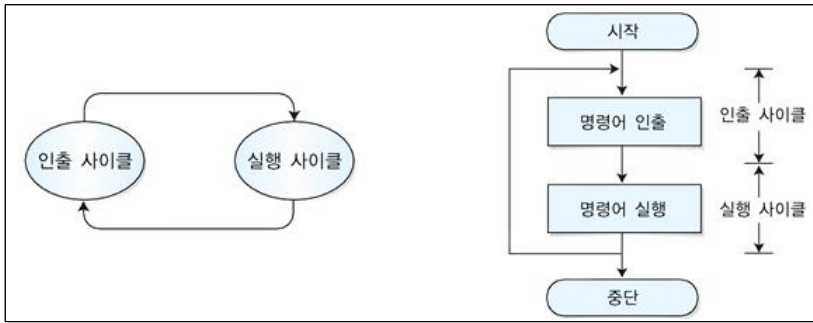
학습내용3 : 명령어사이클

1. 명령어 사이클(Instruction Cycle)이란?

* 명령어 사이클 정의

- 중앙처리장치가 하나의 명령어를 실행하는데 필요한 전체 처리 과정

<명령어 사이클은 기본적으로 인출 사이클(fetch cycle)과 실행 사이클(execution cycle)의 부 사이클(sub-cycle) 두 개로 구성되어 있음>



1) 실행되는 동작은 다음과 같이 분류함

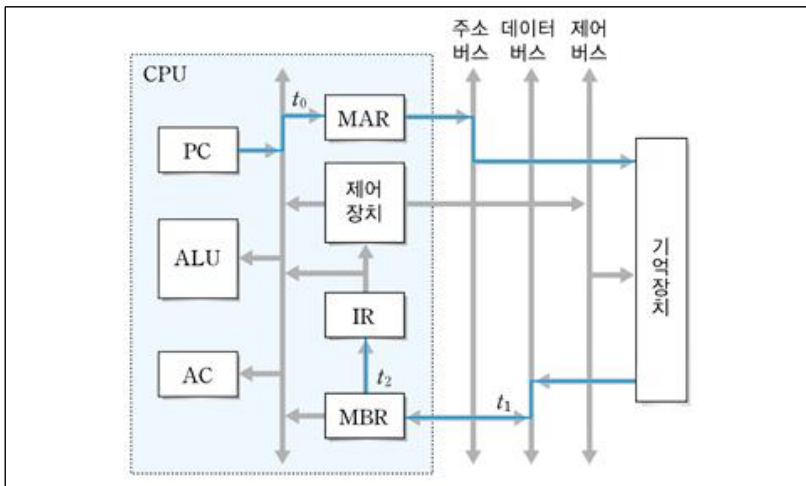
- ① 프로세서와 기억장치 간에 데이터가 전송됨
- ② 프로세서와 I/O 모듈 간에 데이터가 전송됨
- ③ 데이터에 대하여 지정된 산술 혹은 논리 연산이 수행됨
- ④ 제어(Control)동작
 - 점프(Jump) 같이 실행 명령어의 순서가 변경될 때 사용됨

2. 명령어 인출 사이클 과정

1) 명령어 인출 사이클의 3단계 과정

- ① t_0
 - 프로그램 카운터(PC)는 다음에 인출할 명령어의 주소를 가지고 있음 이것을 MAR에 저장함
- ② t_1
 - 프로세서는 PC가 지정하는 기억장소로부터 명령어를 인출함 그리고 PC 내용을 증가시킴
- ③ t_2
 - 인출된 명령어가 명령어 레지스터(IR)로 적재됨 그리고 프로세서는 명령어를 해석하고, 요구된 동작을 수행함

2) 인출 사이클에서 주소 및 명령어의 흐름



<CPU는 실행 사이클 동안에 명령어 코드를 해독(Decode)하고, 그 결과에 따라 필요한 연산들을 수행>

* 예시 : ADD 명령어

- 기억장치에 저장된 데이터를 AC(Accumulator, 누산기)의 내용과 더하고, 그 결과는 다시 AC에 저장하는 명령어임

3) ADD 명령어 실행 사이클 3단계 과정

① t₀

- MBR에 저장될 데이터의 기억장치 주소를 MAR로 전송함

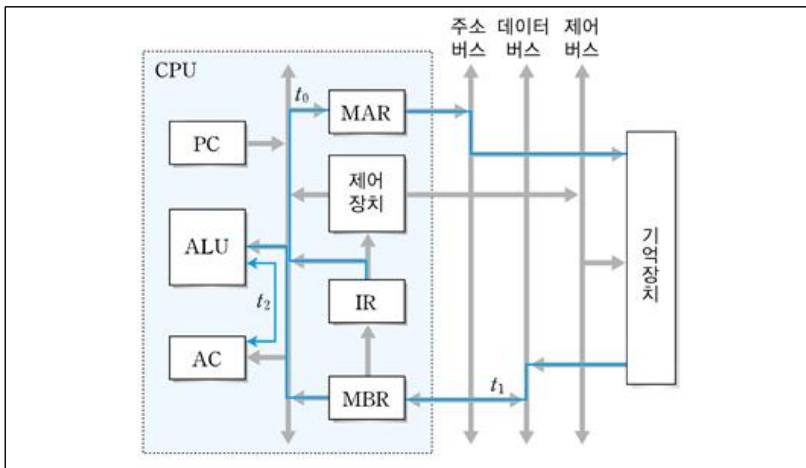
② t₁

- 저장할 데이터를 버퍼 레지스터인 MBR로 이동함

③ t₂

- MBR 데이터와 AC의 내용을 더하고 결과값을 다시 AC에 저장함

4) ADD 명령어 실행 사이클 동안의 정보 흐름



3. 컴퓨터 언어를 이용한 프로그램의 작성

- * 어셈블리 언어(assembly language)

- 어셈블리 코드들의 집합으로, 고급 언어와 기계어 사이의 중간 언어
- 저급 언어인 기계어와 일대일 대응 관계로 기계어로의 변환과정이 쉬움

* 어셈블러 : 어셈블리 프로그램을 2진수의 기계어로 번역하는 소프트웨어

* 니모닉스(mnemonics) : 어셈블리 명령어가 지정하는 동작을 개략적으로 짐작할 수 있도록 하기 위해 사용되는 기호

- LOAD : 기억장치에 저장된 데이터나 명령어를 명령어 레지스터 IR로 읽어 오는 어셈블리 명령어
- ADD : 두 레지스터에 저장된 데이터를 산술적으로 덧셈을 수행하고 결과를 레지스터에 저장하라는 명령어
- STOR : 처리되거나 계산된 데이터를 기억장치로 저장하는 명령어

4. 덧셈 연산에 대한 고급 언어에서 기계어로 변경되는 과정

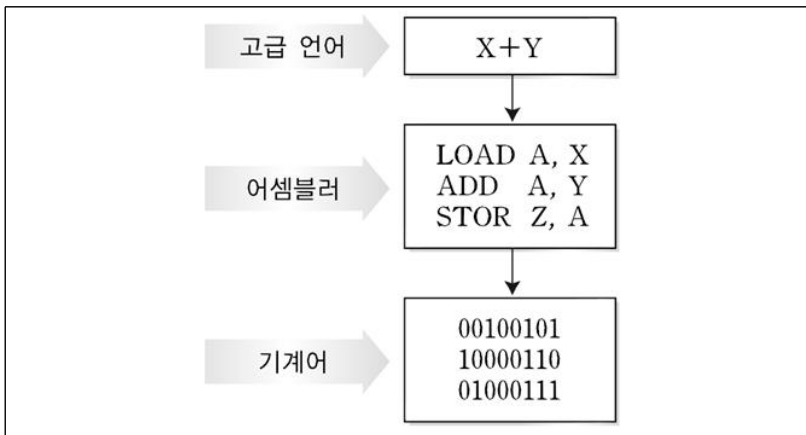
① 고급 언어로 두 수가 덧셈을 수행하도록 작성된 코드의 예

- $$-Z = X + Y$$

② 어셈블리로 변환

- 1 단계 : LOAD A, X
- 2 단계 : ADD A, Y
- 3 단계 : STOR Z, A

③ 기계어로의 변환



5. 기계어

1) 기계어는 두 개의 필드로 구분된 일정한 형식을 가짐

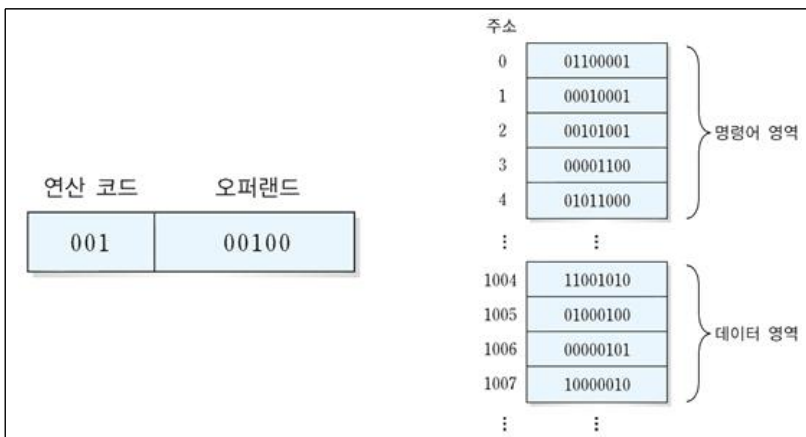
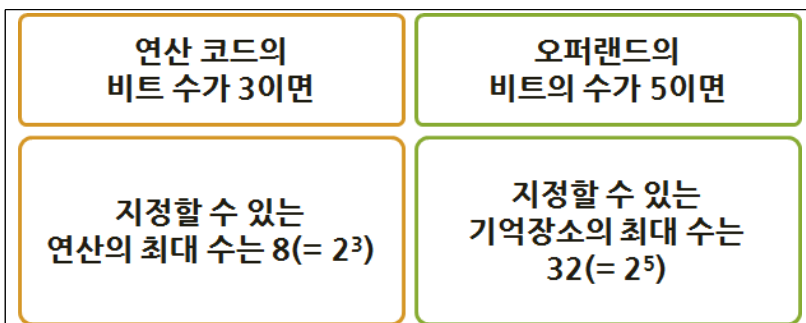
① 연산 코드(Operation Code) 필드

- CPU가 수행할 각종 연산을 지정해 주는 비트

② 오퍼랜드(Operand) 필드

- 적재될 데이터가 저장되어 있는 기억장치 주소 또는 연산에 바로 사용될 데이터 비트 값을 나타냄

2) 8비트 기계명령어 형식



* 기억장치에 저장되어 있는 단어 단위의 기계어

- 프로그램과 데이터가 기억장치에 저장되는 단위를 단어(word)라고 하는데 CPU가 한 번에 처리할 수 있는 비트들의 그룹임

【학습정리】

1. 컴퓨터의 기본 기능으로 입력, 기억, 처리, 출력, 통신 기능이 있다.
2. 명령어 사이클은 크게 인출사이클과 실행사이클로 구분한다.