



동주 - Computer Architecture

📅 날짜	@2023년 4월 17일
🏷 태그	

컴퓨터 구성

하드웨어

- 중앙처리장치(CPU), 기억장치(RAM, SSD), 입출력 장치(마우스, 프린터 등...)로 구성되어 있다.
- 이 세가지는 **시스템 버스**로 연결되어 있으며, 시스템 버스는 데이터와 명령 제어 신호를 각 장치로 실어나르는 역할을 한다.
- 중앙처리장치(CPU)
 - 인간으로 따지면 두뇌에 해당하는 부분
 - 주기억장치에서 프로그램 명령어와 데이터를 읽어와 처리하고 명령어의 수행 순서를 제어한다.
 - 비교와 연산을 담당하는 **산술논리연산장치(ALU)**와 명령어의 해석과 실행을 담당하는 제어장치, 속도가 빠른 데이터 기억장소인 **레지스터**로 구성되어 있음
- 기억장치
 - 프로그램, 데이터, 연산의 중간 결과를 저장하는 장치
 - 주기억장치
 - ROM : 비휘발성 read-only 특성을 가진 장치로, 제조 단계에서 시스템에 기억시키고 변화시키면 안 되는 BIOS와 같은 데이터가 이 장치에 저장된다.
 - RAM : 읽기, 쓰기가 모두 가능한 장치로 응용 프로그램, 운영체제 등을 불러와 CPU가 작업할 수 있도록 하는 공간이다. 그러나 휘발성이므로 전원이 꺼지면 가지고 있던 데이터가 전부 사라지게 되므로 실행하고 있는 파일을 보조기억장치에 수시로 저장하게 된다. (휘발성인 만큼 대신 속도는 매우 빠른 편이다.)

- 보조기억장치
 - 하드디스크나 SSD 등과 같은 장치가 이에 해당된다.
 - 주기억장치에 비해 속도는 느리지만 많은 자료를 영구적으로 보관할 수 있는 장점이 있다.
- 캐시 메모리
 - 고속의 처리 장치인 CPU와 주기억장치 사이의 속도 차이를 완화시키기 위한 고속 버퍼(임시) 메모리로, CPU와 주기억장치 사이에 존재한다.

소프트웨어

- 시스템 소프트웨어: 운영체제, 컴파일러
- 응용 소프트웨어: 워드프로세서, 스프레드시트와 같은 프로그램들

시스템 버스

- 컴퓨터는 기본적으로 **읽고 처리한 뒤 저장**하는 과정으로 이루어진다. (READ -> PROCESS -> WRITE)
 - 이 과정을 진행하면서 끊임없이 주기억장치(RAM) 과 소통한다. 이때 운영체제가 64bit라면, CPU는 RAM으로부터 데이터를 한번에 64bit씩 읽어온다.
 - 이러한 과정에서 사용되는 통로가 바로 시스템 버스
- **중앙처리장치**와 하드웨어 구성 요소를 물리적으로 연결하는 선
- 각 구성요소가 다른 구성요소로 데이터를 보낼 수 있도록 통로가 되어줌
- 용도에 따라 데이터 버스, 주소 버스, 제어 버스로 나누어짐

데이터 버스

- 중앙처리장치와 기타 장치 사이에서 데이터를 전달하는 통로
- 기억장치와 입출력장치의 명령어와 데이터를 중앙처리장치로 보내거나, 중앙처리장치의 연산 결과를 기억장치와 입출력장치로 보내는 '**양방향**' 버스

주소 버스

- 데이터를 정확히 실어나르기 위해서는 기억장치 '주소'를 정해주어야 함 (메모리 주소?)
- 중앙처리장치가 주기억장치나 입출력장치로 기억장치 주소를 전달하는 '**단방향**' 버스

제어 버스

- 주소 버스와 데이터 버스는 모든 장치에 공유되기 때문에 이를 제어할 수단이 필요함
- 중앙처리장치가 기억장치나 입출력장치에 제어 신호를 전달하며, 읽기와 쓰기 동작을 모두 수행하는 '**양방향**' 버스 (데이터 버스는 데이터를 전달! 헷갈리지 말기)
- 제어 신호 종류 : 기억장치 읽기 및 쓰기, 버스 요청 및 승인, 인터럽트 요청 및 승인, 클락, 리셋 등

중앙처리장치(CPU) 구성 요소

- CPU는 컴퓨터에서 가장 핵심적인 역할을 수행하는 부분으로, 인간의 두뇌에 해당한다.
- 연산장치, 제어장치, 레지스터 3가지로 구성되어있다.

연산 장치

- 산술연산과 논리연산 수행 (따라서 **산술논리 연산장치**라고도 불림)
- 연산에 필요한 데이터를 레지스터에서 가져오고, 연산 결과를 다시 레지스터로 보냄

제어 장치

- 명령어를 순서대로 실행할 수 있도록 제어하는 장치
- 주기억장치에서 프로그램 명령어를 꺼내 해독하고, 그 결과에 따라 명령어 실행에 필요한 제어 신호를 기억장치, 연산장치, 입출력장치로 보냄
- 또한 이들 장치가 보낸 신호를 받아, 다음에 수행할 동작을 결정함
- 궁금한 점 : 제어 장치와 제어 버스와의 연관성? 제어 장치는 제어 버스와 연결되어 있는가?

레지스터

- 고속 기억장치
- 명령어 주소, 코드, 연산에 필요한 데이터, 연산 결과 등을 임시로 저장
- 용도에 따라 범용 레지스터와 특수목적 레지스터로 구분됨
- 중앙처리장치 종류에 따라 사용할 수 있는 레지스터 개수와 크기가 다름
 - 범용 레지스터 : 연산에 필요한 데이터나 연산 결과를 임시로 저장
 - 특수목적 레지스터 : 특별한 용도로 사용하는 레지스터
- 특수한 목적으로 사용되는 레지스터가 존재.
 - MAR (메모리 주소 레지스터, Memory Address Register) : 읽기와 쓰기 연산을 수행할 주기억장치 주소 저장
 - PC (프로그램 카운터, Program Counter) : 다음에 수행할 명령어 주소 저장
 - IR (명령어 레지스터, Instruction Register?) : 현재 실행 중인 명령어 저장
 - MBR (메모리 버퍼 레지스터, Memory Buffer Register) : 주기억장치에서 읽어온 데이터 or 저장할 데이터를 임시저장
 - AC (누산기, Accumulator?) : 연산 결과 임시 저장

중앙처리장치(CPU) 동작 과정

1. 주기억장치(RAM)는 입력장치에서 입력받은 데이터 또는 보조기억장치(하드, SSD)에 저장된 프로그램을 읽어온다.
2. **이때 CPU**는 프로그램을 실행하기 위해 주기억장치(RAM)에 저장된 프로그램 명령어와 데이터를 읽어와 처리하고 결과를 다시 주기억장치(RAM)에 저장
3. 주기억장치(RAM)은 처리 결과를 보조기억장치(하드, SSD)에 저장하거나 출력장치로 보냄
4. 제어장치는 위의 1~3의 과정에서 명령어가 순서대로 실행되도록 각 장치를 제어한다.

명령어 세트

- CPU가 실행할 명령어의 집합
 - 연산 코드(Operation Code) + 피연산자(Operand)로 이루어짐
 - 연산 코드 : 실행할 연산
 - 피연산자 : 필요한 데이터 or 저장 위치

- 연산 코드는 연산, 제어, 데이터 전달, 입출력 기능을 가짐
- 피연산자는 주소, 숫자/문자, 논리 데이터 등을 저장
- CPU는 프로그램을 실행하기 위해 주기억장치에서 명령어를 순차적으로 인출하여 해독하고 실행하는 과정을 반복함
- CPU가 주기억장치에서 한번에 하나의 명령어를 인출하여 실행하는데 필요한 일련의 활동을 '**명령어 사이클**'이라고 함.
 - 명령어 사이클은 **인출/실행/간접/인터럽트** 사이클로 나누어짐
 - 인출 : 주기억장치의 지정된 주소에서 하나의 명령어를 가져온다.
 1. PC (프로그램 카운터)에 저장된 주소를 MAR (메모리 주소 레지스터)로 전달
 2. 저장된 내용을 토대로 주기억장치의 해당 주소에서 명령어 인출
 3. 인출한 명령어를 MBR (메모리 버퍼 레지스터)에 저장
 4. 다음 명령어를 인출하기 위해 PC (프로그램 카운터) 값 증가시킴
 5. MBR (메모리 버퍼 레지스터)에 저장된 내용을 IR (명령어 레지스터)에 전달

```
T0 : MAR ← PC
T1 : MBR ← M[MAR], PC ← PC+1
T2 : IR ← MBR
```

- 실행 : 명령어를 실행한다.

```
T0 : MAR ← IR(Addr)
T1 : MBR ← M[MAR]
T2 : AC ← AC + MBR
```

- 이미 인출이 진행되고 명령어만 실행하면 되기 때문에 PC (프로그램 카운터)를 증가할 필요가 없다.
- IR (명령어 레지스터)에 MBR (메모리 버퍼 레지스터)의 값이 이미 저장된 상태를 의미함.
- 따라서 AC (누산기)에 MBR (메모리 버퍼 레지스터)를 더해주기만 하면 됨.

참고

- 슈퍼스칼라 (Superscalar)
 - CPU 내에 파이프라인을 여러개 두어 명령어를 동시에 실행하는 기술
- 파이프 라인 (Pipeline)
 - 시스템의 효율을 높이기 위해 명령문을 수행하면서 몇 가지의 특수한 작업들을 병렬 처리하도록 설계된 하드웨어 기술