

3주차 2차시 프로세서와 메모리 장치

【학습목표】

1. 프로세서의 역할을 살펴보고 설명할 수 있다.
2. 메모리 장치의 역할과 종류를 알고 설명할 수 있다.

학습내용1 : 프로세서

* 컴퓨터 CPU는 우리 두뇌와 같은 역할을 수행한다는 점이 중요하다.

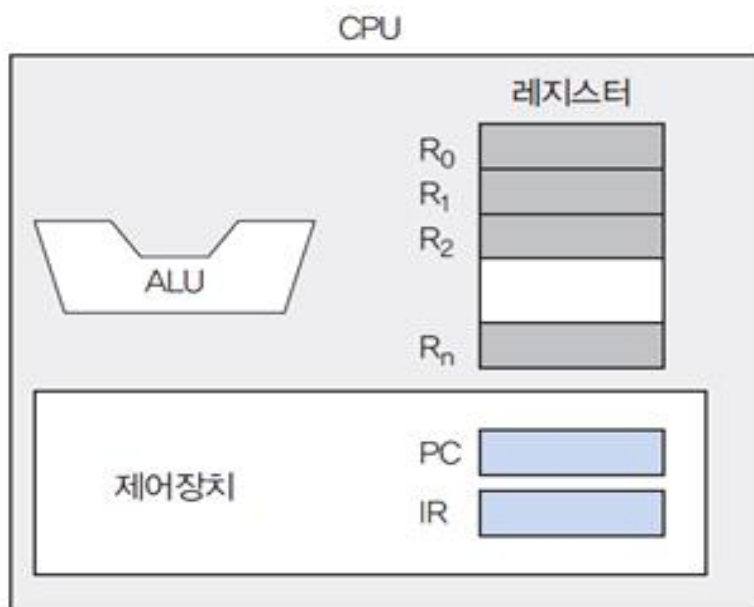
1. 프로세서의 구성 요소

가) ALU(Arithmetic & Logical Unit)

- 프로세서의 가장 핵심 요소로 연산을 담당
- 산술연산과 논리연산을 수행하는 디지털 논리회로

나) 레지스터(Register)

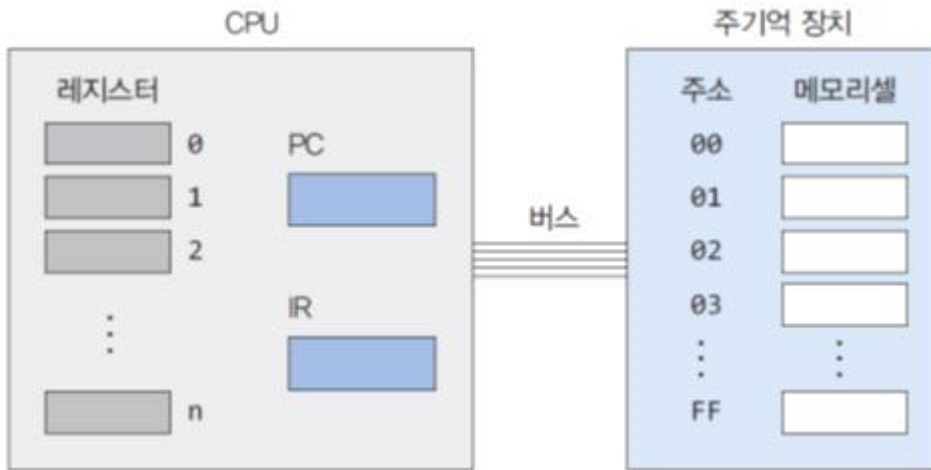
- 데이터 접근 속도가 매우 빠른 적은 수의 메모리 회로
- 범용 레지스터($R_0 \sim R_n$)와 특수 레지스터(PC, IR)로 구성되며, 컴퓨터의 워드 크기와 일치하므로 32비트 컴퓨터의 레지스터의 크기도 32비트



[CPU의 구성]

다) 제어장치

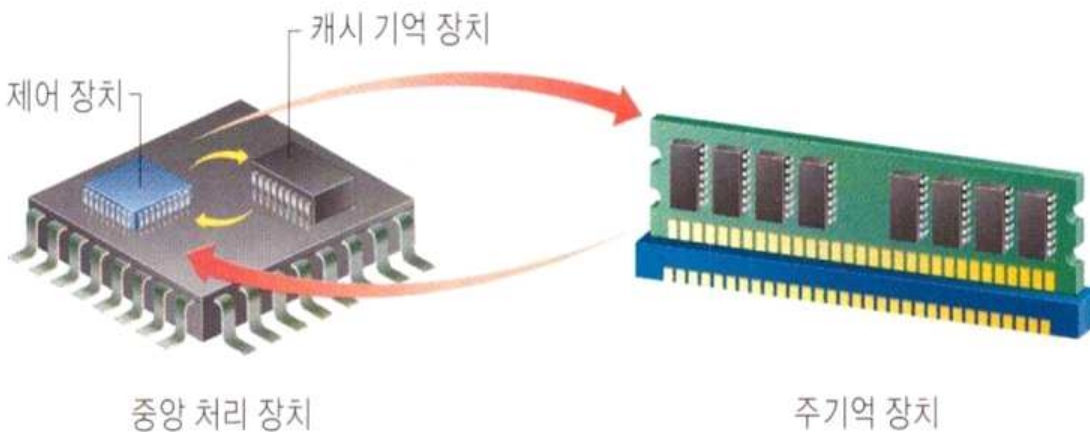
- PC(Program Counter): 주기억장치에 신호를 보내 어떤 주소에 있는 명령문을 가져올지를 지정
- IR(Instruction Register): 가져온 명령문을 저장, IR 레지스터에 있는 코드를 해석하여 ALU와 범용 레지스터에 적절한 전기 신호를 보내 어떤 연산을 해야 하는지 알려준다.
- 주기억장치의 주소(Address)와 내용(Contents)은 다름 (주소에 데이터 내용이 저장)



[주기억장치의 주소와 메모리셀]

라) 캐시 메모리(Cache Memory)

- 레지스터의 속도와 주기억장치의 속도 간의 차이를 극복
 - 캐시 메모리의 용량은 주기억장치보다 훨씬 작으나(수백 KB 수준) 데이터 접근 속도는 주기억장치보다 훨씬 빠르다.
- 주기억장치 내용을 → 캐시로 가져온 후 → 레지스터로 옮겨 처리

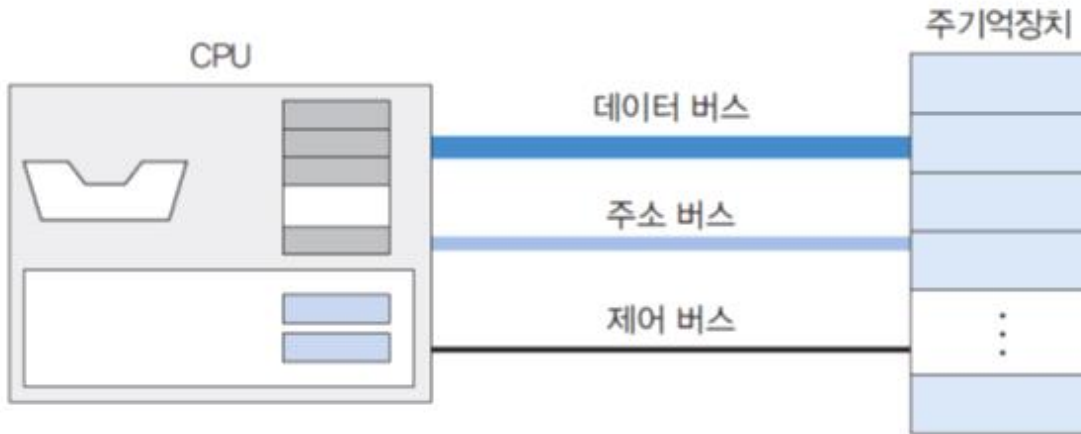


[캐시 기억장치와 주기억장치]

마) 버스(Bus)

* 프로세서와 주기억장치를 연결하는 도선의 묶음

- 데이터 버스
- 주소 버스
- 제어 버스: 예) “데이터를 읽어오라(Read)”. “데이터를 저장하라(Write)”



바) 저장 프로그램 개념(폰 노이만 구조)

* ‘저장 프로그램(Stored Program)’ 개념

- 초기에 컴퓨터는 새로운 응용을 처리하기 위하여 프로세서 하드웨어의 회로를 다시 구성(외부 프로그래밍 방식)
- 오늘날은 프로그램을 주기억장치에 저장했다가 프로세서가 명령어를 해석하여 처리 → 유연성이 증가

* 폰 노이만 구조(Von Neumann Architecture)



2. 기계어와 어셈블리어

1) 컴퓨터를 실제 작동할 수 있는 명령어는 기계어와 어셈블리어로 구성됨을 강조

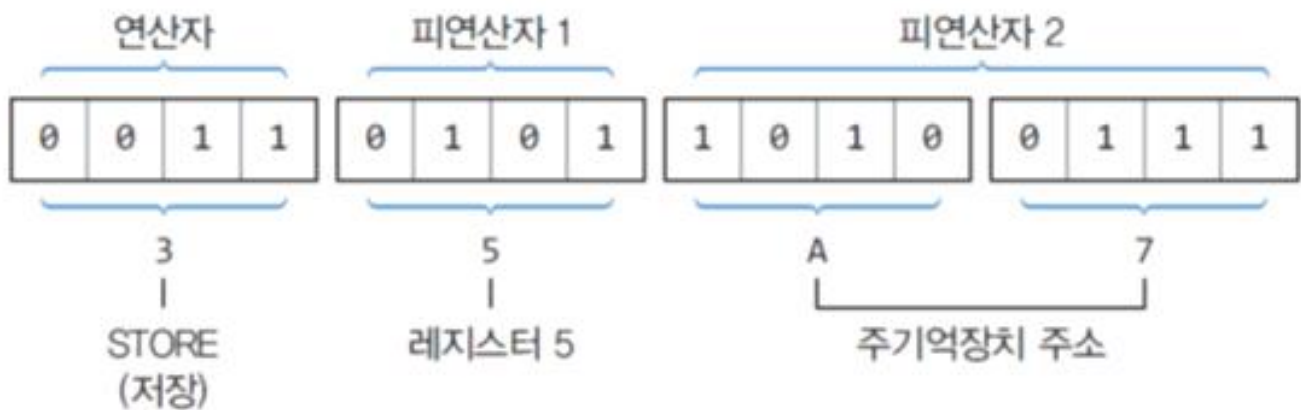
가) 기계어(Machine Instruction) - 연산자와 피연산자

- 기계어: 연산자(Opcode)와 피연산자(Operand)로 구성

- 기계어의 길이는 일반적으로 워드 크기와 일치하며, 기종에 따라 기계어가 다름(IBM, HP, CDC)

연산자는 프로세서가 처리할 작업을 지정, 피연산자는 처리할 대상(목적물=자료)을 의미

- 예) 가상의 프로세서 : 워드 크기 16비트, 연산자 4비트, 피연산자 12비트



| 그림 3-10 가상 프로세서의 기계어 명령어 예

[가상 프로세서의 기계어 명령어 예]

나) 어셈블리어

- 기계어 명령어를 문자화, 기호화

- 프로그램의 작성이 용이

- 기계어와 어셈블리어는 1:1 매핑 관계

- 예) 앞의 예 '0011 0101 1010 0111'에서

- 연산자 '0011' → "STORE"

- 레지스터 주소 '0101' → "R5(Register 5)"

- 주기억장치 주소 '1010 0111' → "MemAdd"

비트열 '0011 0101 1010 0111' ⇒ STORE R5 MemAdd

다) 기계어의 종류

- 데이터 이동(STORE, LOAD), 산술연산(ADD, SUBTRACT, MULTIPLY, DIVIDE), 논리연산(AND, OR, NOT, XOR), Shift/Rotate 연산(BIT 이동), 제어 기능(JUMP, HALT)

① 데이터 이동(Data transfer)

- 데이터를 한 장소에서 다른 장소로 이동
- 예) STORE, LOAD

② 산술연산(Arithmetic)

- 두 수 사이의 산술적 연산
- 예) ADD, SUBTRACT, MULTIPLY, DIVIDE 등

③ 논리연산(Logic)

- AND, OR, XOR, NOT과 같은 논리적 연산

	10101010		10101010		10101010
AND	11110000	OR	11110000	XOR	11110000
	<hr/> 10100000		<hr/> 11111010		<hr/> 01011010

④ Shift/Rotete

- Shift to Left 11001011 → 10010110
- 비트열의 제일 왼쪽 비트는 삭제
- Rotate to Left 11001011 → 10010111
- 비트열의 제일 왼쪽 비트는 오른쪽 비트로

Shift to Left	11001011	\Rightarrow	10010110
Rotate to Left	11001011	\Rightarrow	10010111

⑤ 제어(Control)

- 예) 'JUMP', 'HALT' 등

라) 가상 프로세서의 사례

* 예) 산술연산 “A + B = C”

- 데이터 A, B, C가 각각 주기억장치 주소 0A, 0B, 0C에 있다고 가정
- 먼저 주소 A의 값을 레지스터 1로(LOAD), 주소 B의 값을 레지스터 2로(LOAD), ALU는 레지스터 1과 레지스터 2에 덧셈을 수행(ADD), 그 결과를 레지스터 3에, 마지막으로 레지스터 3의 값을 주소 C에 저장(STORE)

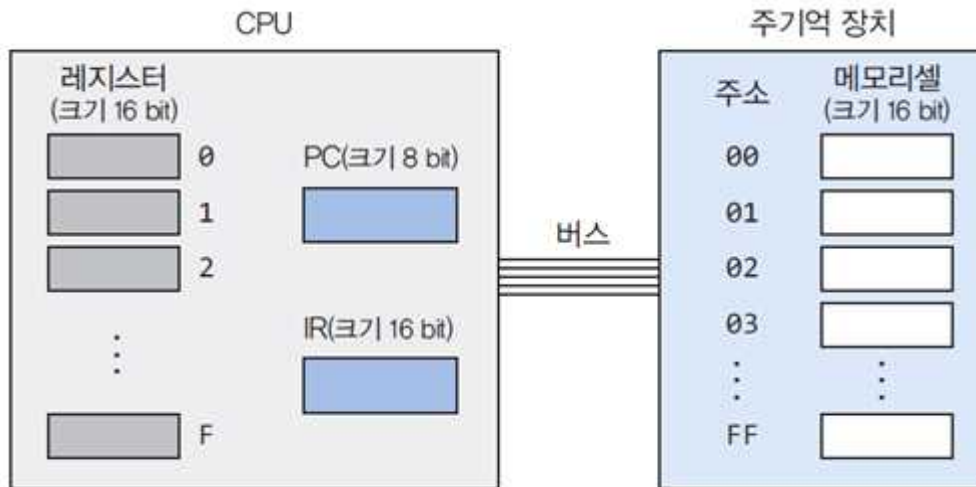
- 어셈블리 프로그램

```
LOAD  R1  A      ← ①
LOAD  R2  B      ← ②
ADD   R3  R1  R2  ← ③
STORE R3  C      ← ④
```

- 기계어 프로그램

```
0010 0001 0000 1010 ⇒ LOAD  R1  A
0010 0010 0001 1011 ⇒ LOAD  R2  B
0110 0011 0001 0010 ⇒ ADD   R3  R1  R2
0011 0011 0010 1100 ⇒ STORE R3  C
```

- 가상 프로세서의 구조



마) 프로세서 설계 방식

* 프로세서의 설계 철학에 따라 기계어 집합이 간단(CISC) 또는 복잡(RISC)

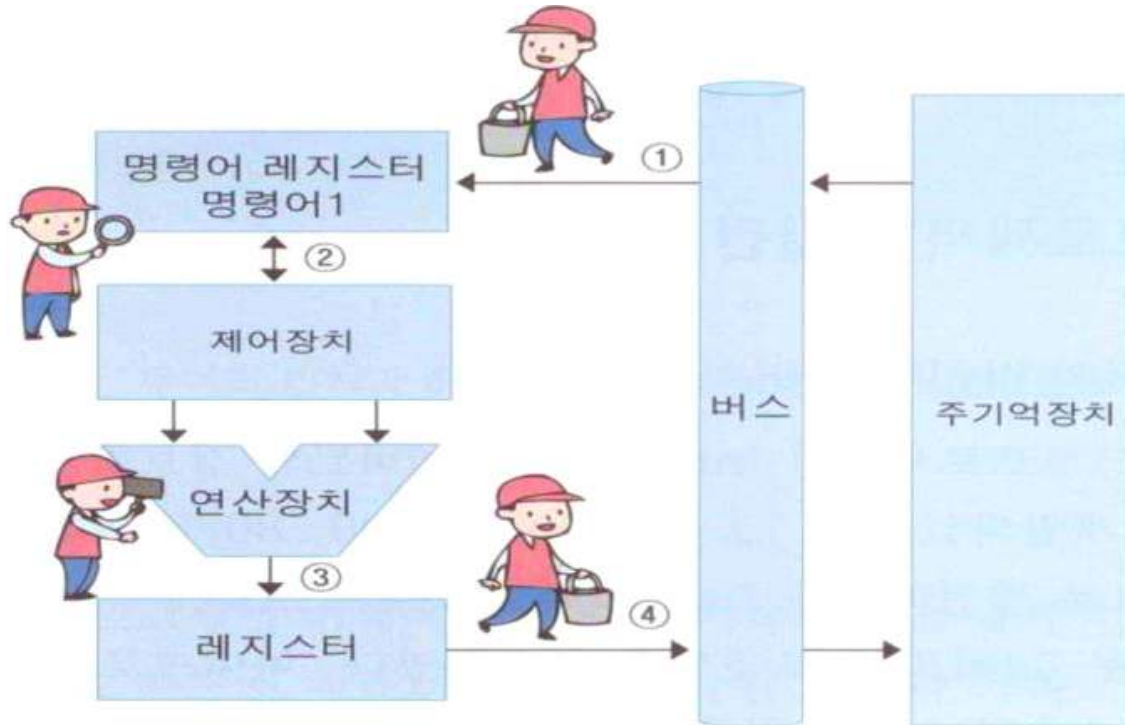
- CISC(Complex Instruction Set Computer) 머신
 - 하드웨어 강조 : 명령어가 복잡하고 종류도 많음
 - 예) INTEL 프로세서
- RISC(Reduced Instruction Set Computer) 머신
 - 소프트웨어 강조
 - 예) ARM 프로세서, MIPS 프로세서

CISC	RISC
하드웨어 강조	소프트웨어 강조
명령어가 복잡하고 종류가 많음	명령어가 간단하고 종류가 적음
적은 수의 레지스터 이용	많은 수의 레지스터 이용
다양한 주소 모드	적은 주소 모드
명령어 처리를 위해 몇 클럭사이클 필요	명령어를 한 클럭사이클에 처리
파이프라이닝 어려움	파이프라이닝 용이함

[CISC 머신과 RISC 머신의 비교]

3. 중앙처리장치의 기능

* 프로그램 내장방식에서 명령어의 처리 과정은 CPU의 주요 역할



[프로그램 내장방식 컴퓨터의 명령문 처리 과정]

* CPU가 명령어를 처리하기 위해 4개의 사이클을 반복

- 1단계 호출사이클 : 명령문을 주기억장치에서 CPU로 호출(fetch)한다.
- 2단계 해석 사이클 : 호출된 명령문의 명령어가 무엇인지 해석(decode)한다.
- 3단계 실행 사이클 : 해석된 명령문을 실행(execution)한다.
- 4단계 인터럽트 처리 : 인터럽트가 발생되었는지 확인(interrupt)한다.

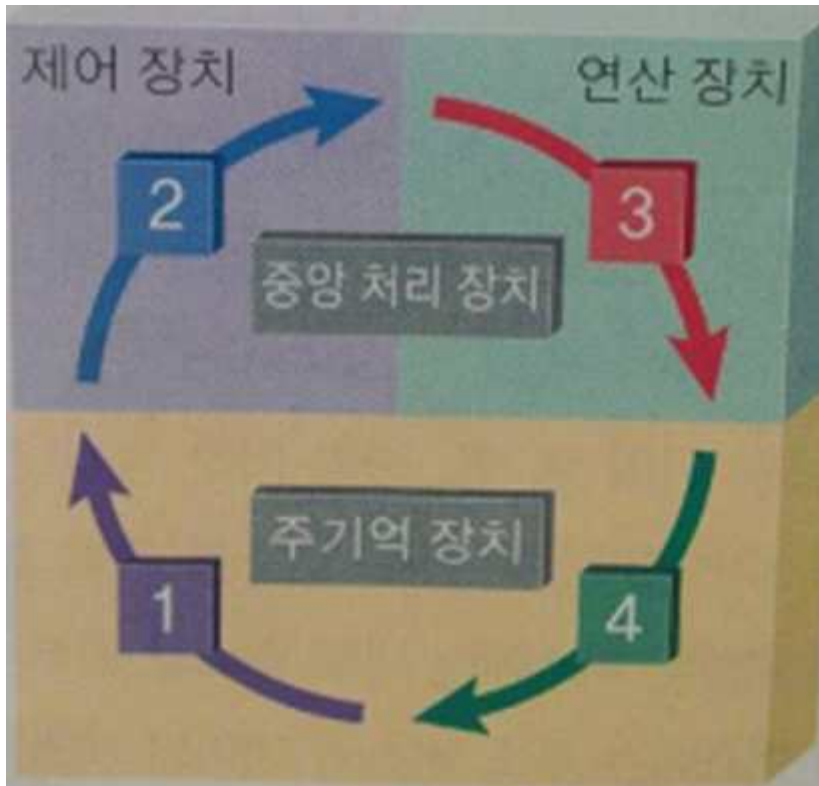


[명령어의 처리 과정]

* 머신 사이클 = 명령사이클 + 실행사이클

- 명령사이클: 명령을 인출하고 해독하는 단계

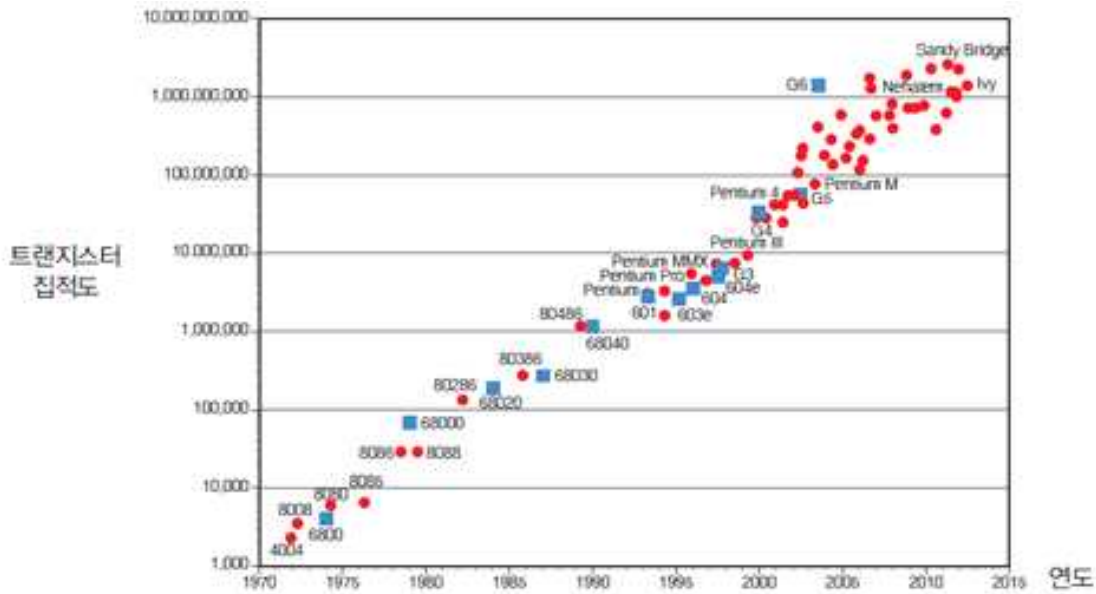
- 실행사이클: 해독된 명령을 실행하고, 그 결과를 저장하는 단계



[CPU 내부에서의 명령어 처리]

* 무어의 법칙

- 1965년 인텔사 회장 무어(Gordon Moore)는 마이크로프로세서 집적도의 발전에 관한 무어의 법칙(Moore's Law)
- 마이크로프로세서 칩의 성능은 18개월마다 두 배씩 증가

출처: <http://electroiq.com>

[무어의 법칙 적용 - INTEL 마이크로프로세서(파란점)의 경우]

* 다양한 기능의 프로세서 칩

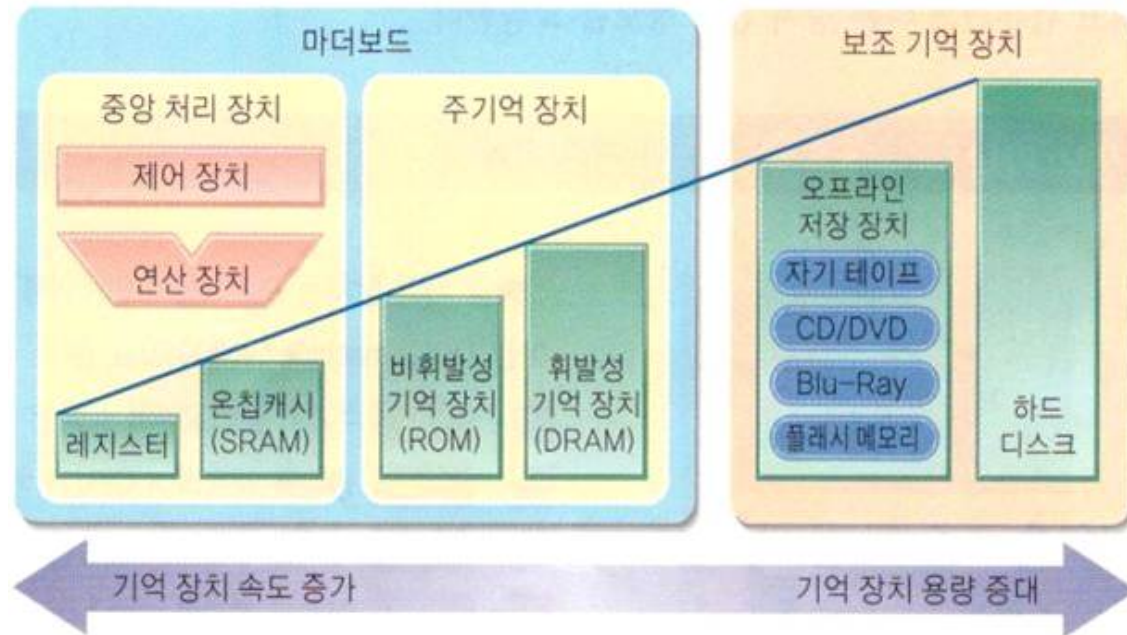
- 3차원 그래픽 가속기 프로세서
- 비디오 코덱 프로세서
- 미디어 프로세서 - 음악 플레이
- 통신용 프로세서 - 모뎀에서 신호처리
- 일반제품용 칩과 주문형 반도체(ASIC)(ASIC: Application Specific Integrated Circuit)

학습내용2 : 메모리 장치

- 메모리는 우리 두뇌의 뇌세포의 역할을 수행한다는 점을 강조

1. 메모리 장치 기능과 역할

- * 기억장치 계층
- 기억장치의 탑재 위치와 크기, 속도와 기능에 따라 구분



[기억장치의 계층구조]

- 데이터와 프로그램의 실행 여부에 따른 계층구조

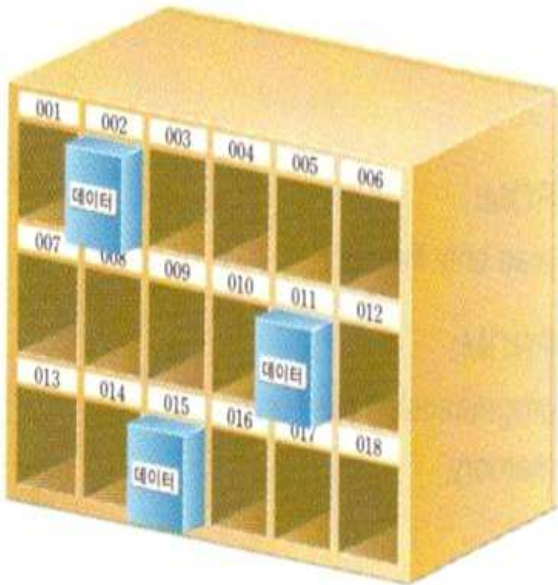


[기억장치의 계층적 구조도]

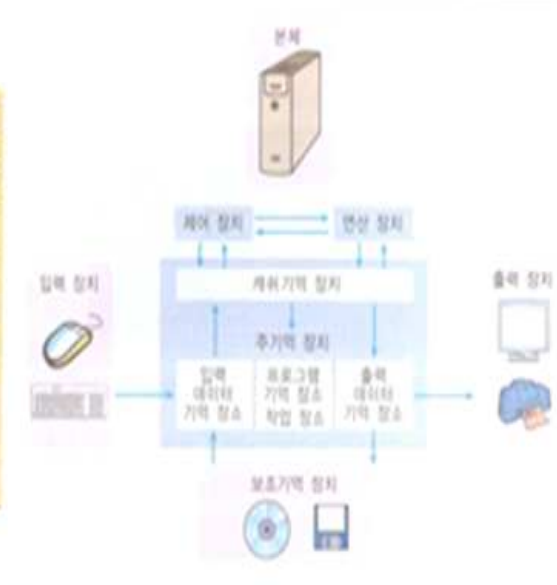
- 주기억장치 = Memory = 1차 기억장치
- 보조기억장치 = Storage = 2차 기억장치

* 기억장치의 역할

- 두뇌의 세포와 같이 체계적으로 주소화되어 구성
- 기억된 프로그램이나 데이터는 몇 번 읽거나 복사해도 변하지 않지만, 새로운 내용을 쓰면 기억된 내용이 소멸



[기억장치의 주소 체계]

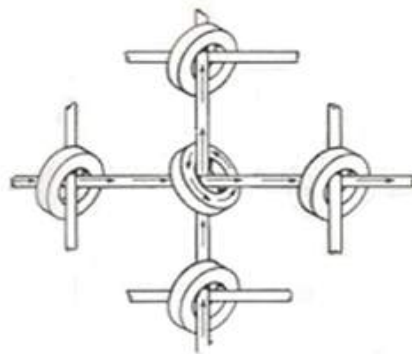
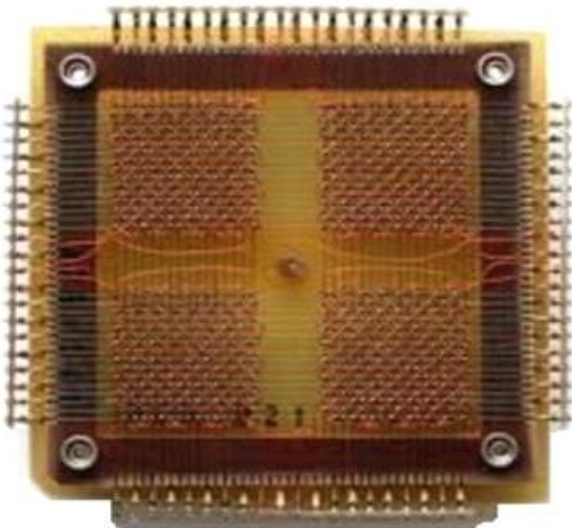


[주기억장치 구성과 역할]

2. 주기억장치

* 과거의 주기억장치

- 1세대 컴퓨터: 진공관
- 2세대 컴퓨터: 자기코어와 트랜지스터



[자기코어와 선택]

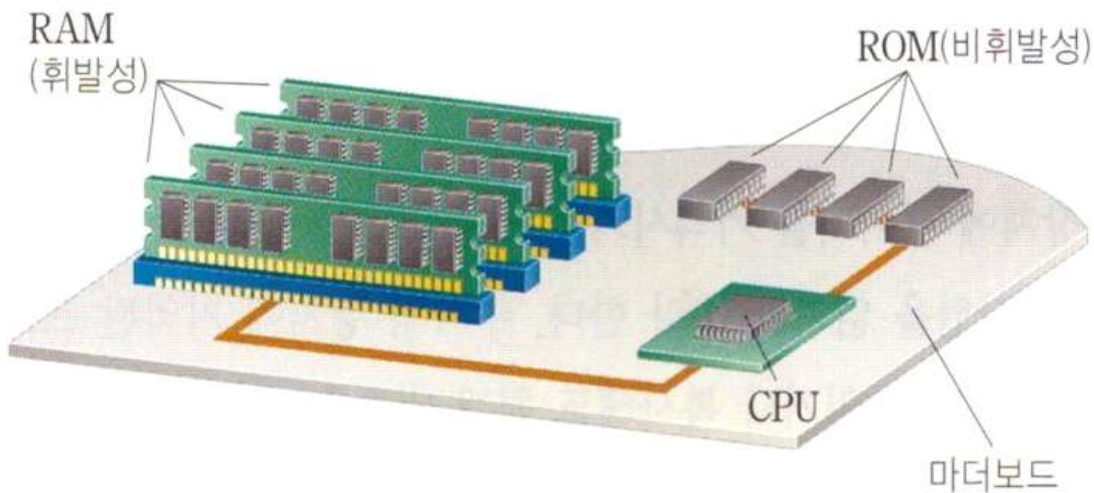
* 현재의 주기억장치

① 휘발성 기억장치 : RAM

- 전원이 중단되면 기억된 내용이 소멸되는 RAM
- SRAM : 캐시기억장치로 휘발성 기억 장치로 캐시기억장치에 사용
- DRAM : 주기억장치로 칩판과 같이 자유롭게 기록하고,수정,삭제할 수 있는 기억장치

② 비휘발성 기억장치: ROM

- 한번 기억된 내용을 다시 기록하는 특별한 과정을 거치지 않고는 기억된 내용을 영구히 기억
- PROM: 하드디스크나 프린터에 탑재되어 전원을 공급하기 위한 프로그램이 내장
- EROM

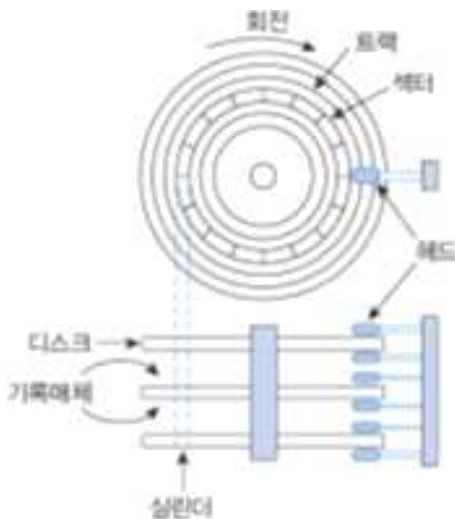


[주기억장치]

3. 보조기억장치

* 하드디스크 시스템

- 성능을 좌우하는 요인 : 회전 수, 데이터 접근속도, 단위면적당 자기밀도 등



(a) 하드디스크 구조 도해



(b) 하드디스크 내부 구조

* 이동식 하드디스크는 외장하드디스크

* SSD(Solid State Disk)는 플래시 메모리로 구성



* 기억장치의 종류와 특징

종류	장치	접근속도	가격
캐시기억장치	SRAM	아주 빠름	가장 높음
주기억장치	DRAM	빠름	높음
보조기억장치	하드디스크 CD-ROM 자기테이프 USB메모리	보통 느림 매우 느림 비교적 빠름	보통 낮음 매우 낮음 가장 낮음

【학습정리】

1. 중앙처리장치가 프로그램과 데이터를 직접 읽고 쓸 수 있는 기억장치는 레지스터, 캐시메모리, 주기억장치(RAM) 임
2. SRAM은 전원이 공급되는 동안에는 기억된 내용이 유지되나, DRAM은 전원이 공급되어도 주기적으로 충전(Refresh)을 해야 기억된 자료가 유지
3. 블루레이(Blu-ray)메모리는 고 선명 비디오 디지털데이터를 저장할 수 있도록 만들어진 제4세대 광 저장 장치