



## CONTENTS

# 학습목표

- 컴퓨터 시스템의 기본 구조를 설명할 수 있다.
- 기억장치에 대해 설명할 수 있다.
- 중앙처리장치에 대해 설명할 수 있다.
- 마이크로프로세서에 대해 설명할 수 있다.

▶▶▶▶▶  
들어가기

## LEARNING

# 지난 주차 복습

▶▶▶▶▶  
복습하기

### 02주차 학습내용. 정보의 표현

- 1 컴퓨터의 자료 표현
- 2 진수와 수의 표현
- 3 컴퓨터의 정보 종류
- 4 컴퓨터의 연산

## 지난 주차 **복습**

LEARNING  
복습하기

### 컴퓨터의 자료 표현

- 컴퓨터의 정보 처리 단위 중에서 가장 작은 정보 단위는 Bit

### 진수와 수의 표현

- 컴퓨터와 관련된 진수
  - 10진수, 2진수, 8진수, 16진수

## 지난 주차 **복습**

LEARNING  
복습하기

### 컴퓨터의 정보 종류

- 정수, 부동소수, 문자와 논리 등

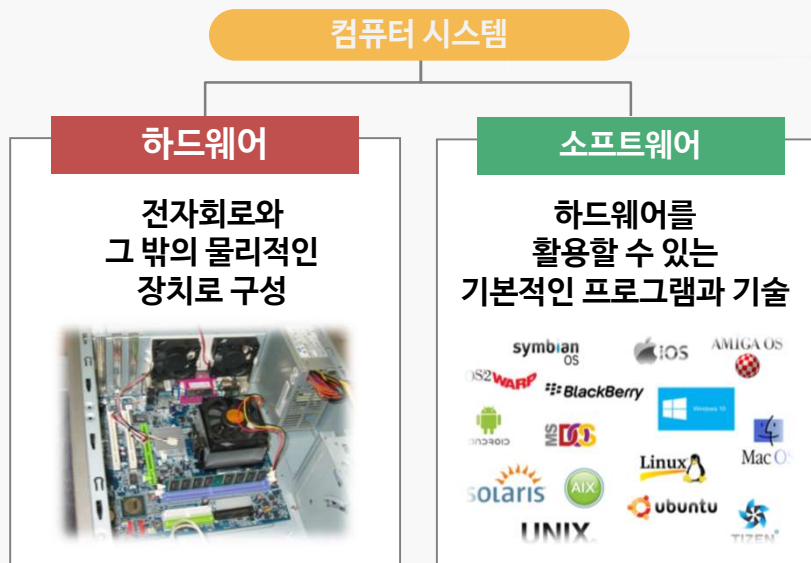
### 컴퓨터의 연산

- 정수의 뺄셈에 보수를 이용함
- 부동소수 연산은 지수를 같게 맞추고 가수부로 연산함
- 논리연산자들은 논리 게이트로 논리회로를 설계할 수 있음

## 1

컴퓨터 시스템의  
기본 구조

## 1] 컴퓨터 시스템의 기본 구조


 학습하기


## 2] 컴퓨터 시스템의 구성 요소

학습하기

### 컴퓨터 시스템의 구성요소

펌웨어

하드웨어

소프트웨어

시스템 소프트웨어

컴퓨터를 효율적으로 운영하여 제어하기 위한 프로그램(운영체제)

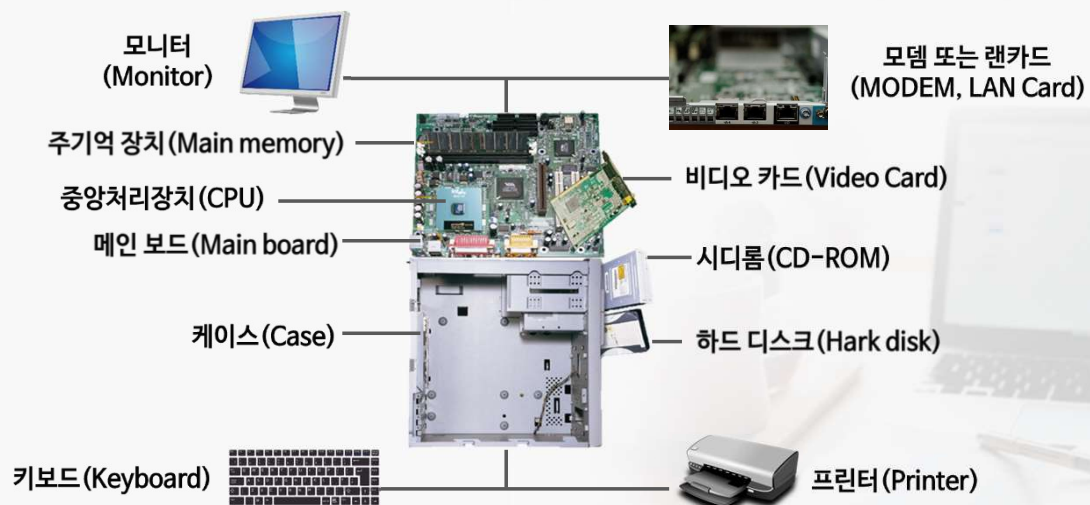
응용 소프트웨어

특정 분야의 업무를 처리하기 위한 프로그램 (한글, 엑셀 등)

## 2] 컴퓨터 시스템의 구성 요소

학습하기

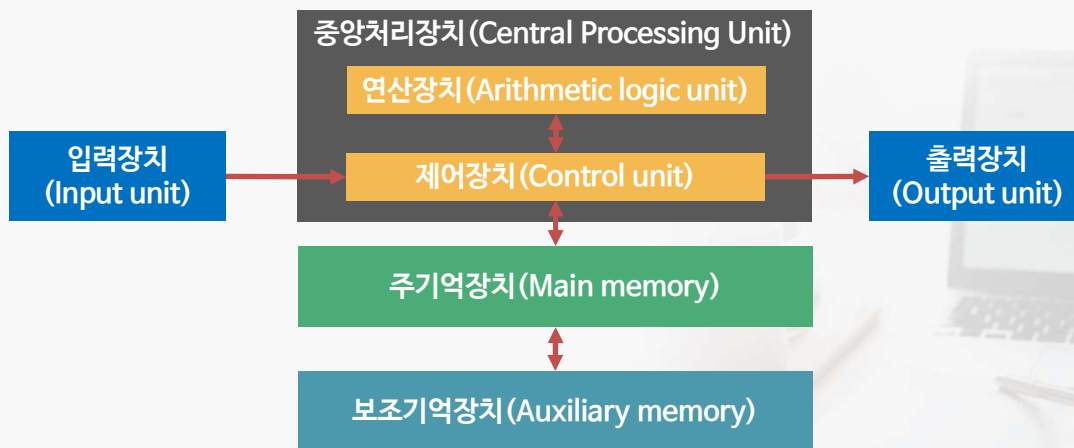
### ◇ 하드웨어



### 3] 컴퓨터의 내부 구조

학습하기

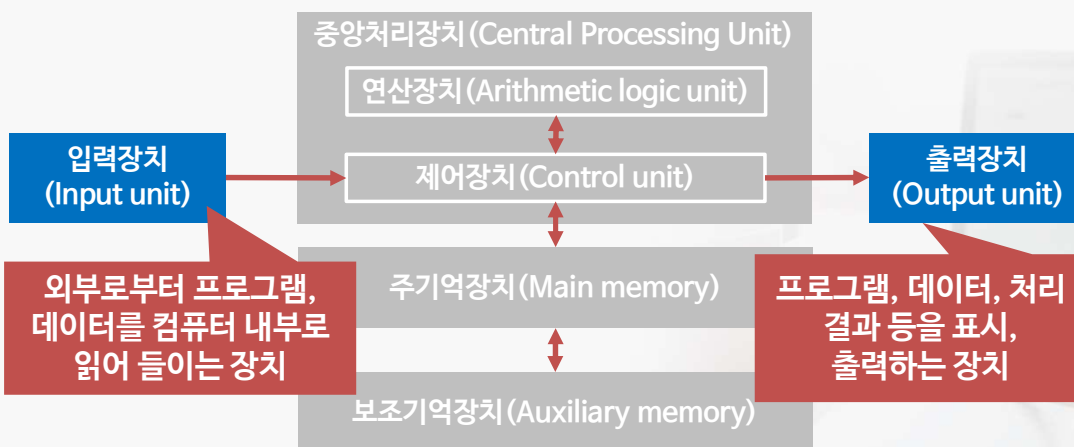
중앙처리장치를 중심으로 한 자료 신호와 명령 신호를 나타냄



### 3] 컴퓨터의 내부 구조

학습하기

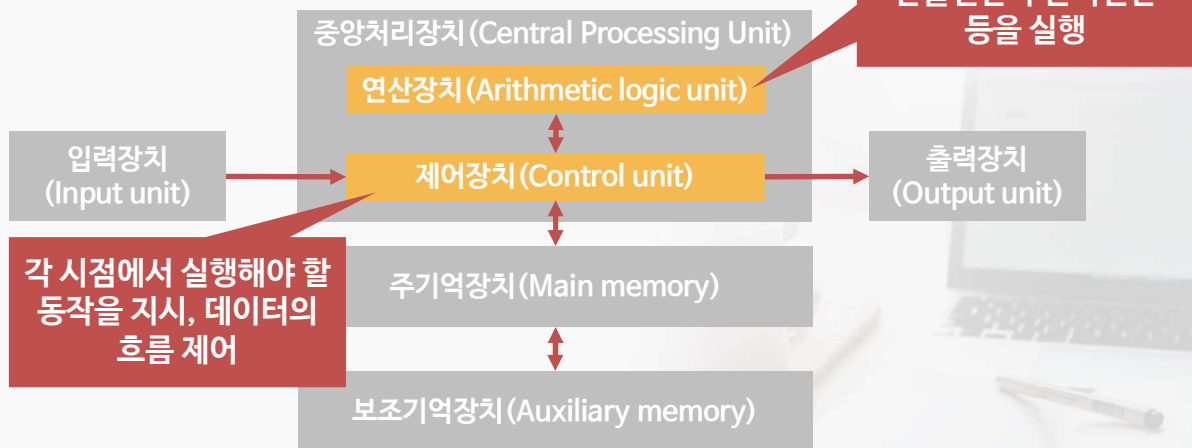
중앙처리장치를 중심으로 한 자료 신호와 명령 신호를 나타냄



### 3] 컴퓨터의 내부 구조

학습하기

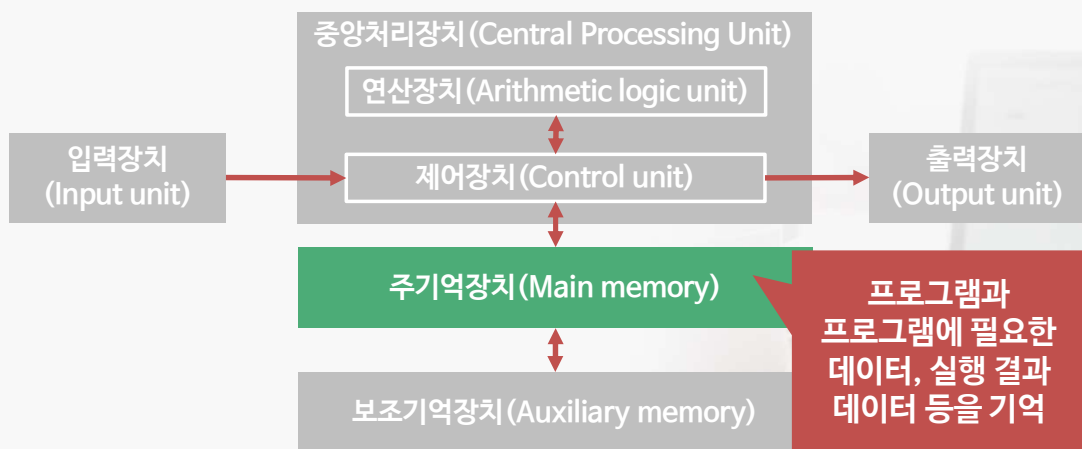
중앙처리장치를 중심으로 한 자료 신호와 명령 신호를 나타냄



### 3] 컴퓨터의 내부 구조

학습하기

중앙처리장치를 중심으로 한 자료 신호와 명령 신호를 나타냄



## 4] 프로그램 내장 방식

학습하기

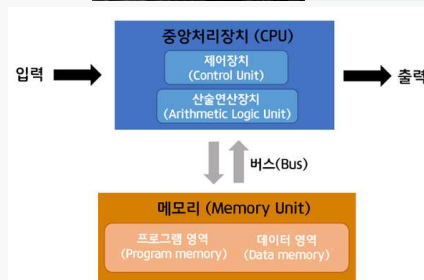
### ◆ 고안

Stored Program



폰 노이만

메모리에 자료와 프로그램이 함께 저장

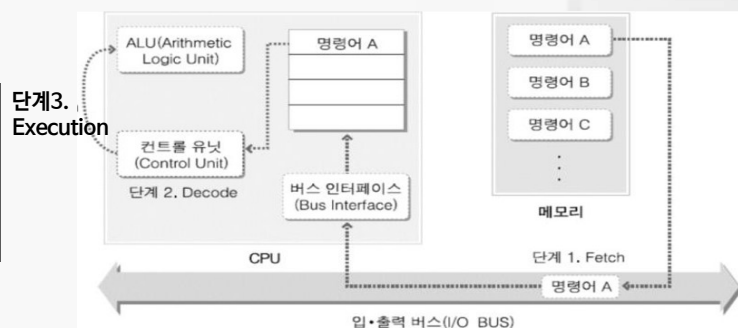
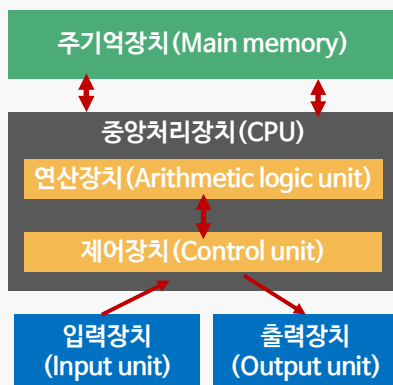


## 4] 프로그램 내장 방식

학습하기

### ◆ 중앙처리장치 (CPU)

- 메모리에서 필요한 자료를 이용
- 저장된 명령어를 순차적 (Sequential)으로 실행





## 5] 명령어 (Instruction)

학습하기

### ◆ 명령어 형식

#### 연산 부분 (Operation part)

- 부분은 명령어가 수행해야 할 기능을 의미하는 코드

#### 피연산 부분 (Operand part)

- 연산에 참여하는 자료를 의미하는 코드

### ◆ 명령어가 16비트로 구성



## 5] 명령어 (Instruction)

학습하기

### ◆ 명령어 종류

명령어	구문형식	기능
ADD	ADD A	피연산자의 자료 A와 레지스터의 자료를 더하는 명령어
LDA	LDA B	피연산자의 자료 B를 레지스터에 가져오는 명령어
STA	STA C	레지스터의 내용을 피연산자 C에 저장하는 명령어
HLT	HLT	컴퓨터를 종료하는 명령어

### 예 16비트의 명령어 형식



## 2

## 기억장치



## 1] 주기억장치의 구조


 학습하기

## ◇ 주기억장치

## 주기억장치

메모리라고도 하며, 작업 내용인 프로그램 명령어와 프로그램에서 이용할 자료를 저장하는 장치

## ◇ 주소



메모리의 저장소는 주소(address)를 이용하여 각각 바이트 단위로 고유하게 식별



## 1] 주기억장치의 구조

학습하기

### ◇ 주소

컴퓨터가 한 번에 작업할 수 있는 데이터의 단위는?

자료는 주로 **워드 단위**로 주기억장치와 중앙처리장치 사이를 이동

워드는 32비트 또는 64비트

## 1] 주기억장치의 구조

학습하기

### ◇ 버스

#### ● 관련 자료 전달 경로

주 소	실제 메모리
00000000	
00000004	
00000008	
0000000C	
00000010	
00000014	
00000018	
0000001C	
3FFFFFF8	
3FFFFFFC	

주소버스

자료버스

제어버스

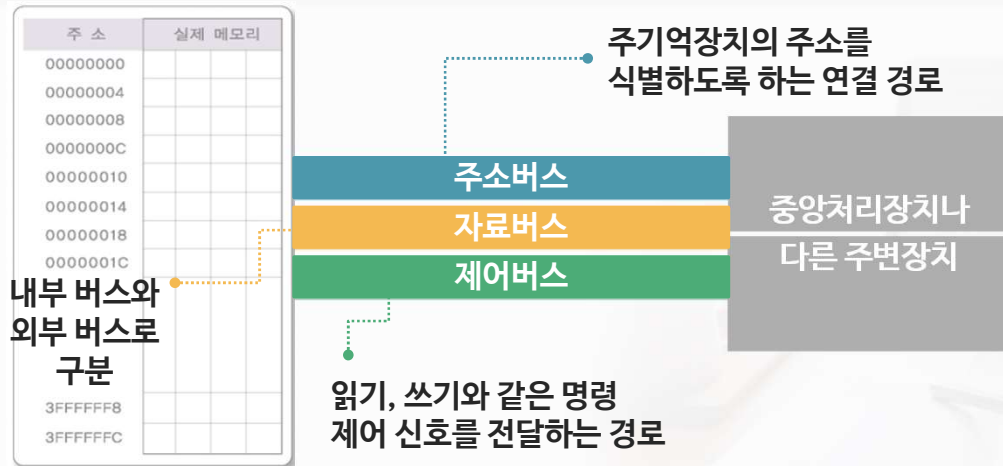
중앙처리장치나  
다른 주변장치

## 1] 주기억장치의 구조

학습하기

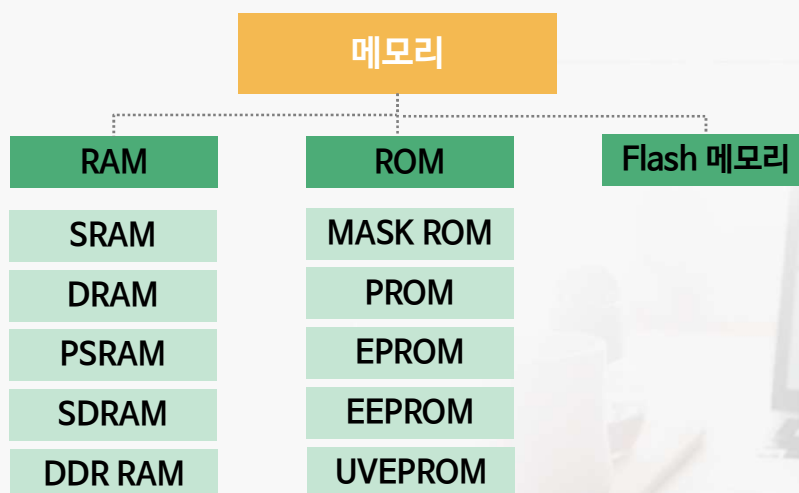
### ◇ 버스

#### ◆ 관련 자료 전달 경로



## 2] 기억장치의 종류

학습하기



## 2] 기억장치의 종류

학습하기

### ◇ RAM

#### RAM

- Random Access Memory, 임의 접근 메모리
- 소멸성(volatile) 기억장치

→ 쓰기과 읽기의 두 회로가 있어서 정보의 쓰기와 읽기가 가능

#### DRAM

- 전원이 연결된 상태에서 일정한 주기마다 전기적으로 재충전 필요
- 주기억장치로 주로 사용 (SDRAM)

#### SRAM

- 전원만 연결되어 있으면 정보가 지워지지 않는 기억장치
- 캐쉬 메모리(Cache memory)에 주로 사용

## 2] 기억장치의 종류

학습하기

### ◇ ROM

#### ROM

- 읽기 전용 메모리, Read Only Memory
- 비소멸성(nonvolatile) 메모리

→ 읽기만 가능한 메모리

Mask ROM

PROM

EPROM

EEPROM

기계 특성 정보와 컴퓨터가 전원공급을 받아 처음 수행해야 할  
**바이오스 프로그램(BIOS : Basic Input/ Output System) 저장에 사용**

## 2] 기억장치의 종류

학습하기

### ◆ Flash 메모리

#### Flash 메모리

메모리 셀들의 한 부분이 섬광(flash)처럼 단 한 번의 동작으로 지워질 수 있는 메모리



## 2] 기억장치의 종류

학습하기

### ◆ Flash 메모리

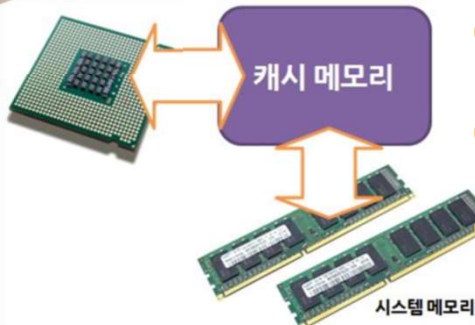
- 1 RAM과 ROM의 장점을 가진 메모리
- 2 소비전력이 작은 비소멸성 메모리
  - 정보의 입출력도 자유로워 디지털텔레비전, 디지털캠코더, 휴대전화, 디지털카메라, 개인휴대단말기(PDA), 게임기, MP3플레이어 등에 널리 이용되는 기억장치
- 3 바이트 단위가 아닌 블록 단위의 주소 지정이 가능하며 쓰기 시간도 오래 걸림

### 3] 캐시 메모리

학습하기



캐시는 왜 사용할까요?



CPU에 비해 상대적으로 주변기기의 속도가 매우 느림

주기억장치와 CPU의 속도 차이를 해결

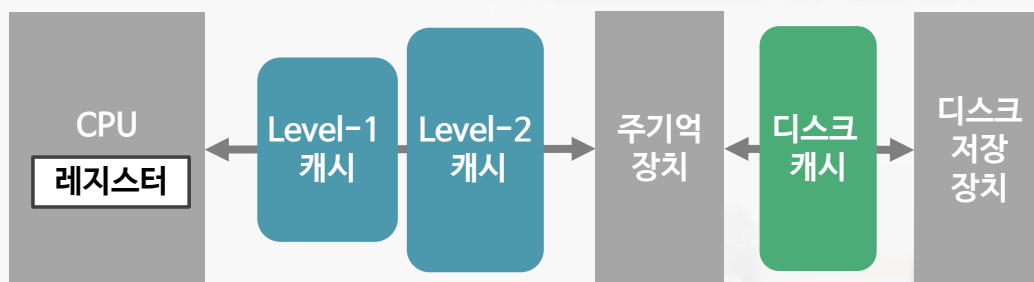
캐시 메모리는 메인 메모리보다 대개 약 10배쯤 더 빠름

저장 속도가 빠르고 고가인 SRAM을 이용

### 3] 캐시 메모리

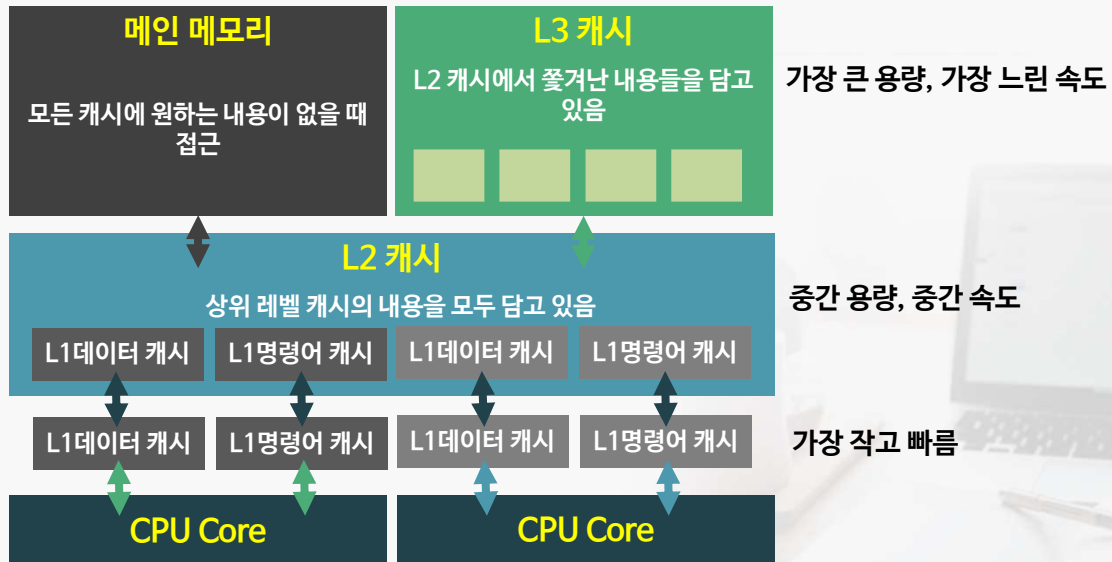
학습하기

#### ◇ 캐시의 종류



### 3] 캐시 메모리

학습하기



### 4] 기억장치의 계층

학습하기

저장장치의 속도와 용량, 가격과 그 쓰임새를 고려해야 함





## 3

## 중앙처리장치



## 1] 개요

 학습하기

## 중앙처리장치

메모리에 저장된 프로그램과 자료를 이용하여 실제 작업을 수행하는 회로 장치

## 연산장치

- 자료의 연산을 수행

## 레지스터

- 연산에 필요한 자료를 임시로 저장

## 제어장치

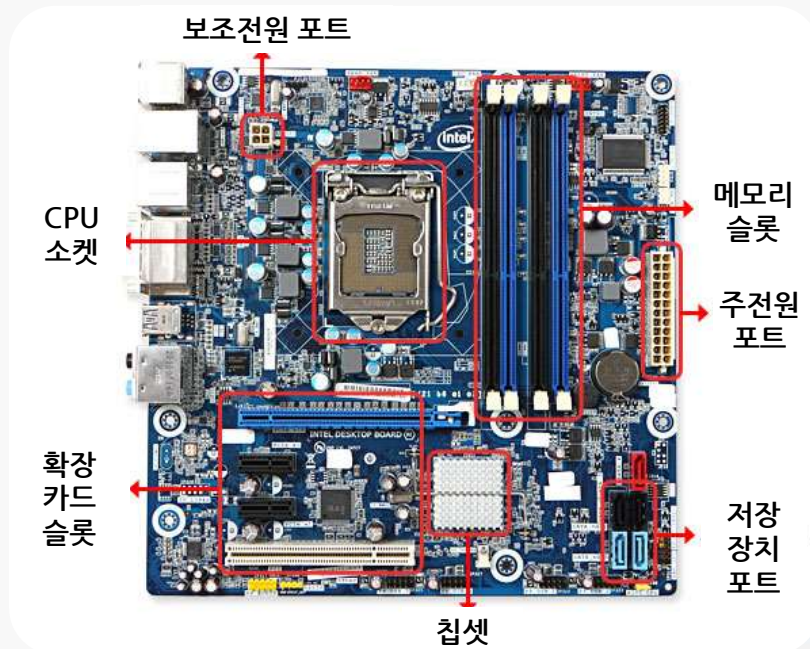
- 컴퓨터의 작동을 제어

## 버스

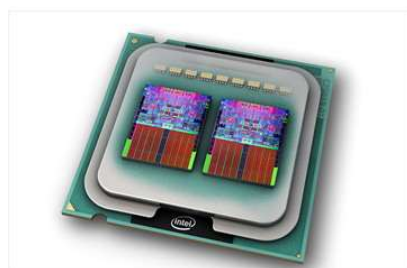
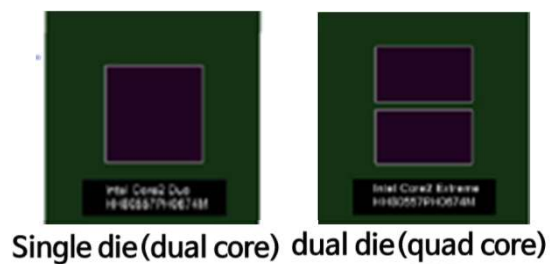
- 자료버스, 제어버스



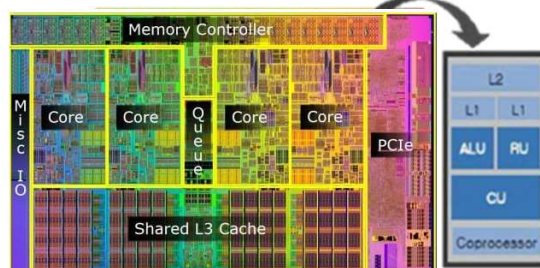
## 1] 개요


 학습하기


## CPU 구조와 기능

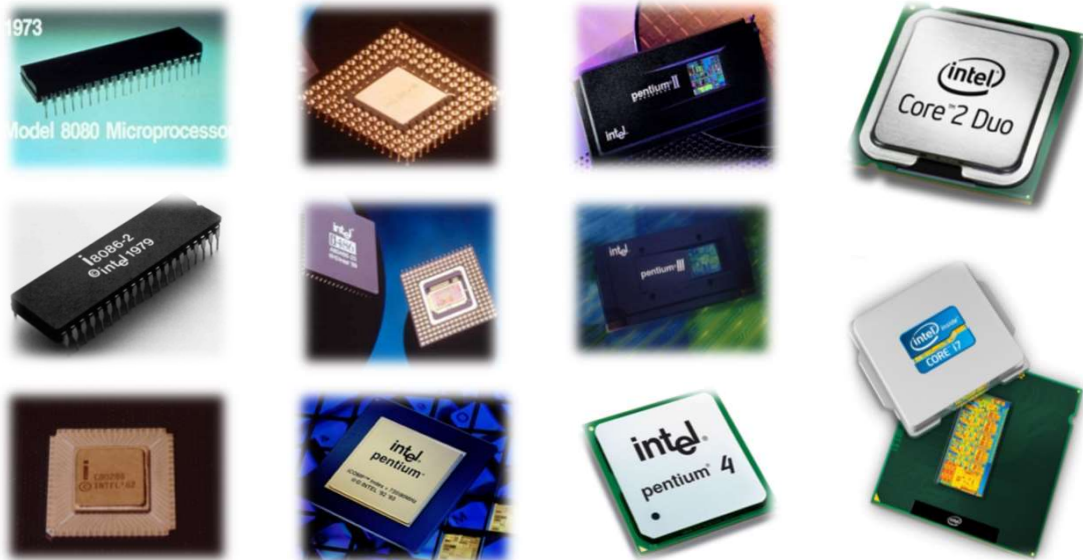


Dual core



Intel Core i7 Lynnfield

## CPU의 종류



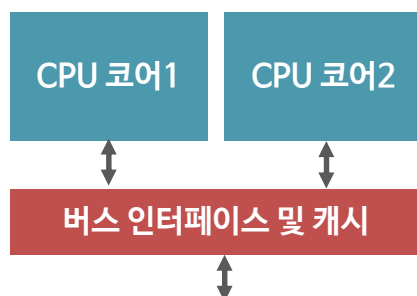
## 1] 개요

학습하기

### ◇ CPU 코어

#### CPU 코어(Core)

명령어 실행에 필요한 CPU 내부의 핵심 하드웨어(슈퍼스칼라) 모듈



● 각 CPU 코어는 시스템 버스와 캐시 공유

● 각 코어는 독립적으로 프로그램 실행

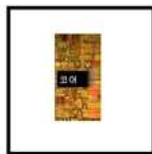
## 1] 개요

학습하기

### ◆ CPU 코어(core)

#### 멀티 코어 프로세서

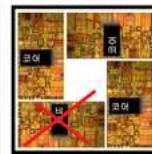
여러 개의 CPU 코어들을 하나의 칩에 포함시킨 프로세서



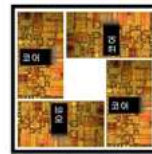
싱글코어



듀얼코어



트리플코어



쿼드코어

두 개의 CPU 코어 포함

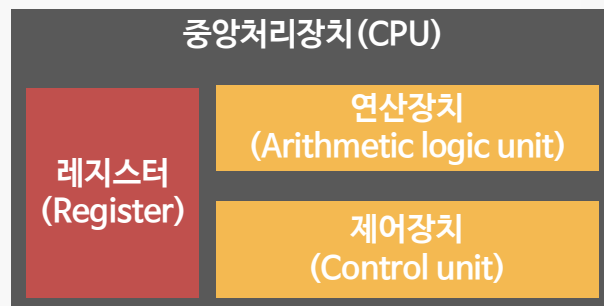
네 개의 CPU 코어 포함

## 2] 연산장치

학습하기

#### 연산장치 (Arithmetic and Logic Unit)

- 더하기, 빼기, 나누기, 곱하기 등의 산술연산
- NOT, AND, OR, XOR 등의 논리 연산을 수행하는 회로



## 2] 연산장치

학습하기

### ◆ 레지스터의 이용

- 1 중앙처리장치의 임시기억장소인 누산 레지스터 (Accumulator)와 자료 레지스터 (Data Register)에 저장된 자료를 연산에 참여할 피연산자로 이용
- 2 결과는 다시 누산 레지스터에 저장되어 필요하면 주기억장치에 저장되거나 다른 연산에 이용

예  $AC \leftarrow AC + DR$

## 3] 제어장치

학습하기

### 제어장치

산술 및 논리 연산에 요구되는 작업을 연속적으로 수행하는 신호를 보냄으로써 연산장치와 레지스터가 명령을 수행하게 하는 장치

제어장치 = 인간의 뇌



## 4] 레지스터

학습하기

### 레지스터

중앙처리장치 내부에서 연산에 필요한 자료를 잠시 저장하기 위한 임시 기억장소

중앙처리장치는 컴퓨터가 명령을 수행하는 과정을 처리하기 위해 여러 개의 레지스터를 가짐

## 4] 레지스터

학습하기

### ◇ 주요 종류와 기능

레지스터 심볼	레지스터 이름	기능
DR	자료 레지스터 (Data Register)	▪ 연산에 필요한 피연산자를 저장하는 레지스터
AR	주소 레지스터 (Address Register)	▪ 현재 접근할 기억장소의 주소를 기억하는 레지스터
AC	누산 레지스터 (Accumulator Register)	▪ 연산장치의 입출력 데이터를 임시적으로 기억하는 레지스터
IR	명령어 레지스터 (Instruction Register)	▪ 현재 수행중인 명령어를 저장하고 있는 레지스터
PC	프로그램 카운터 (Program Counter)	▪ 다음에 실행할 명령어의 메모리 주소가 저장된 레지스터
TR	임시 레지스터 (Temporary Register)	▪ 임시로 자료를 저장하는 레지스터로 범용 레지스터라고도 부름



## 5] 명령어 처리 과정

학습하기

### ◇ 기계 주기(Machine cycle)

메모리에서  
실행할  
명령어를 추출



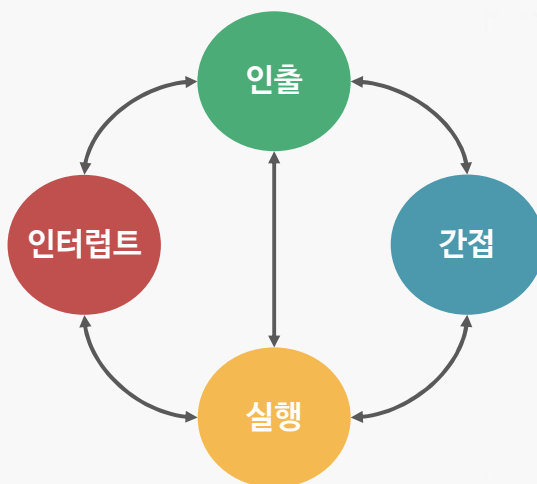
명령 레지스터의  
내용 해독

명령 레지스터의  
연산 실행

## 5] 명령어 처리 과정

학습하기

### ◇ 명령어 사이클



인출 사이클 (fetch cycle)

실행 사이클 (execution cycle)

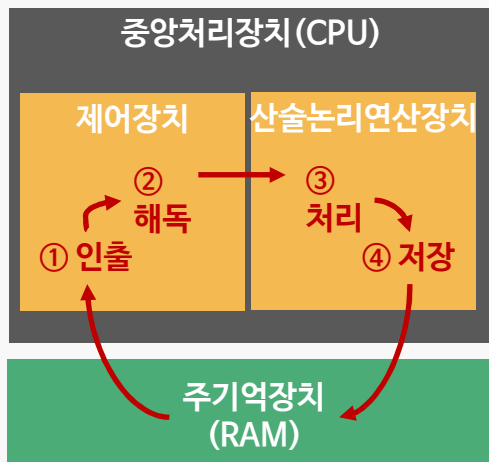
간접 사이클 (indirect cycle)

인터럽트 사이클 (interrupt cycle)

## 5] 명령어 처리 과정

학습하기

### ◆ 명령어 실행



명령어 인출 (Instruction fetch)

명령어 해독 (Instruction decode)

데이터 인출 (Data fetch)

데이터 처리 (Data processing)

데이터 저장 (Data store)

## 6] 두 수의 합: $C \leftarrow A + B$

학습하기

### ◆ 두 정수의 합을 위한 명령어 집합

순서	명령어	의미
명령어 1	LDA A	메모리 A의 내용을 누산 레지스터 (AC)에 저장
명령어 2	ADD B	메모리 B의 내용과 누산 레지스터 (AC)의 값을 더하여 누산 레지스터 (AC)에 다시 저장
명령어 3	STA C	누산 레지스터 (AC)의 값을 메모리 C에 저장
명령어 4	HLT	프로그램 종료



## 6] 두 수의 합 : $C \leftarrow A + B$

학습하기

### ◆ 명령어의 세부 수행 기능

명령어	세부 수행 기능 표시	세부 수행 기능 의미
LDA	$DR \leftarrow M[AR]$ $AC \leftarrow DR$	<ul style="list-style-type: none"> <li>주소 레지스터(AR)의 주소 값의 메모리 자료(<math>M[AR]</math>)를 자료 레지스터(DR)에 저장</li> <li>자료 레지스터(DR)를 다시 누산 레지스터(AC)에 저장</li> </ul>
ADD	$DR \leftarrow M[AR]$ $AC \leftarrow AC + DR$	<ul style="list-style-type: none"> <li>주소 레지스터(AR)의 주소 값의 메모리 자료(<math>M[AR]</math>)를 자료 레지스터(DR)에 저장한 후, 누산 레지스터(AC)와 자료 레지스터(DR)를 더하여 그 결과를 다시 누산 레지스터(AC)에 저장</li> </ul>
STA	$M[AR] \leftarrow AC$	<ul style="list-style-type: none"> <li>누산 레지스터(AC)의 값을 주소 레지스터(AR)의 주소 값 위치의 메모리(<math>M[AR]</math>)에 저장</li> </ul>

## 6] 두 수의 합 : $C \leftarrow A + B$

학습하기

### ◆ 명령어 실행 전과 실행 후의 메모리 내부

#### 명령어 실행 전

주소	실제 메모리 내용
00121FF30	LDA 0012FF40
00121FF34	ADD 0012FF44
00121FF38	STA 0012FF48
00121FF3C	HLT
00121FF40	32
00121FF44	-18
00121FF48	

#### 명령어 실행 후

주소	실제 메모리 내용
00121FF30	LDA 0012FF40
00121FF34	ADD 0012FF44
00121FF38	STA 0012FF48
00121FF3C	HLT
00121FF40	32
00121FF44	-18
00121FF48	14

## 4

## 마이크로프로세서

## 1] 성능


 학습하기

여러 요인으로  
마이크로프로세서의 성능이 결정됨



## 1] 성능

학습하기

### ◆ 자료 버스의 폭

#### 자료 버스의 폭

연산장치와 레지스터 등과 같은 CPU의 내부 구성 요소 간에 자료를 전달하는 통로의 비트 수

→ 자료 버스 폭은 마이크로프로세서의 워드 크기와 비례함



## 1] 성능

학습하기

### ◆ 자료 버스의 종류

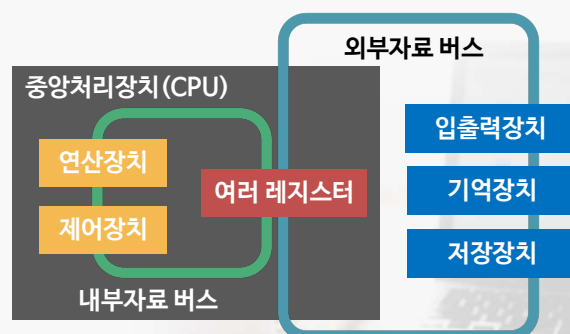
#### ● 내부자료 버스와 외부자료 버스

##### 내부 버스(시스템 버스)

- CPU 내부의 버스

##### 외부 버스(확장 버스)

- CPU를 벗어나 다른 입출력장치나 기억장치, 저장장치로 이동을 위한 버스



## 1] 성능

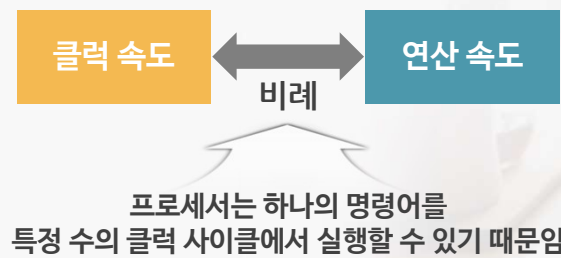
학습하기

### ◆ 클럭 속도

#### 클럭

1초당 진동의 반복 횟수를 재는 단위

→ 클럭 속도의 단위 = Hz



## 1] 성능

학습하기

### ◆ 컴퓨터의 성능

- CPU 성능, RAM의 용량, 외부 자료 버스의 크기

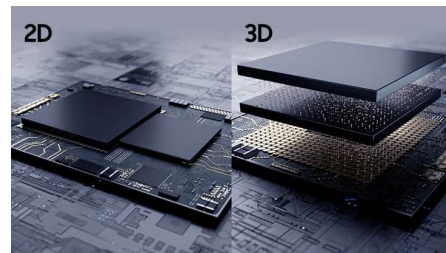
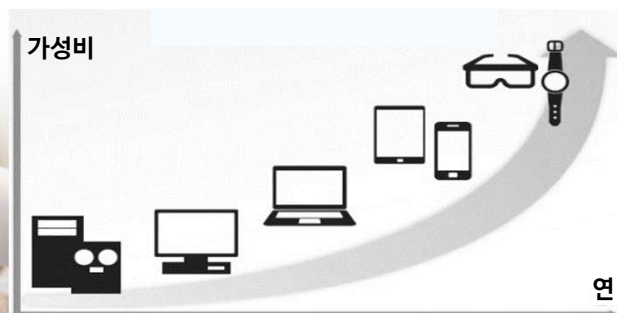


### 무어의 법칙

- 인텔의 공동 설립자인 고든 무어(Gorden Moore)가 1965년도에 한 연설에서 유래

“마이크로 칩의 처리 능력은 18개월마다 2배로 증대된다.”

“마이크로프로세서의 성능은 약 18~24개월마다 2배로 증가한다.”

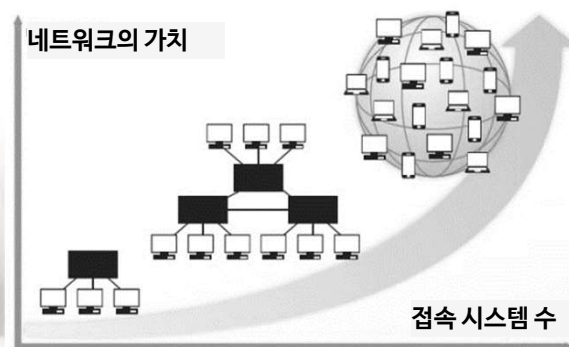


시스템반도체에 3차원 적층 패키지

### 메트칼프의 법칙

- 컴퓨터 네트워크의 규격 중 하나인 이더넷을 발명한 로버트 메트칼프가 1995년도에 한 연설에서 유래

“통신 네트워크의 가치는 참여자(접속 시스템) 수의 제곱에 비례한다.”



## 2] 명령어에 따른 프로세서 분류

학습하기



Complex Instruction Set Computing  
복합 명령어 집합 컴퓨팅 계열



Reduced Instruction Set Computing  
축소 명령어 집합 컴퓨팅 계열

## 2] 명령어에 따른 프로세서 분류

학습하기

### ◇ CISC

- ✓ 명령어의 구조가 복잡하고 100 ~ 250개의 다양한 명령어를 제공
- ✓ 인텔의 80x86 계열과 모토롤라의 680x0 계열의 프로세서
- ✓ 복잡한 연산을 하나의 명령어로 처리하려는 의도에서 시작
- ✓ CISC의 명령어는 복잡한 연산을 수행하기 위해 다양한 길이를 가지며 메모리의 자료를 직접 참조하는 연산도 많이 제공

## 2] 명령어에 따른 프로세서 분류

학습하기

### ◇ CISC

#### 장점

- 복잡한 프로그램을 적은 수의 명령어로 구성

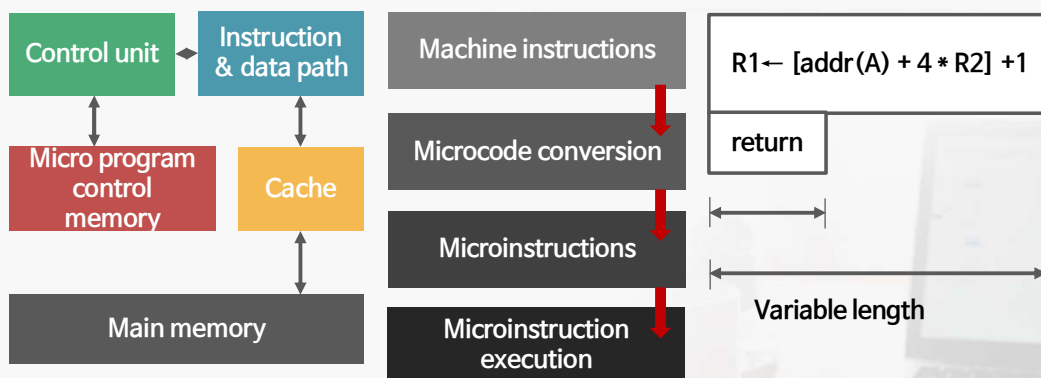
#### 단점

- 복잡한 명령어의 실행을 위한 복잡한 회로가 이용되므로 생산가가 비쌈
- 전력 소모가 많아 열이 많이 발생

## 2] 명령어에 따른 프로세서 분류

학습하기

### ◇ CISC



## 2] 명령어에 따른 프로세서 분류

학습하기

### ◇ RISC

- ✓ 명령어의 수가 적고 그 구조도 단순함
- ✓ 레지스터 내부에서 모든 연산이 수행되며 메모리의 참조는 제한적임
- ✓ 상대적으로 레지스터가 많음
- ✓ 명령어는 고정 길이이며 쉽게 해독이 되는 명령어 형식
- ✓ 1988년 중반 애플의 매킨토시에 장착된 모토로라의 PowerPC에서 처음 구현된 RISC 프로세서는 이후 Sun, HP, NEC의 워크스테이션 컴퓨터

## 2] 명령어에 따른 프로세서 분류

학습하기

### ◇ RISC

#### 장점

- 전체적으로는 RISC 프로세서는 CISC 프로세서보다 수행 속도가 빠름
- 하나의 명령어가 단순하여 그 처리 속도가 매우 빠름

#### 단점

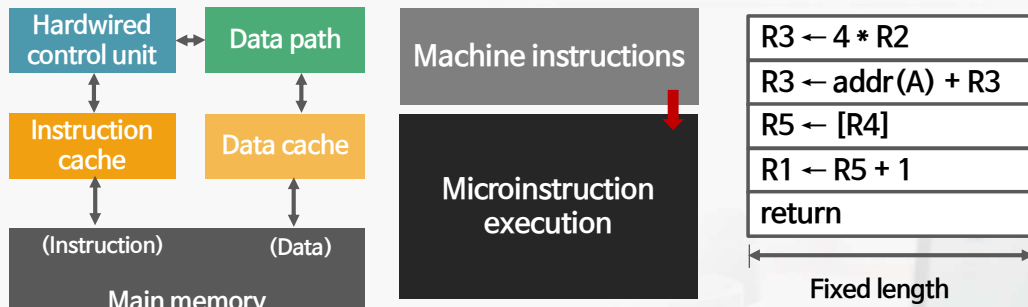
- 하나의 프로그램을 수행하려면 RISC 프로세서는 CISC보다 많은 명령어를 실행해야 함



## 2] 명령어에 따른 프로세서 분류

학습하기

### ◇ RISC



## 2] 명령어에 따른 프로세서 분류

학습하기

### ◇ CISC와 RISC 비교

CISC	구분	RISC
▪ 많고 복잡한 명령어 구조	명령어 구조	▪ 적고 단순한 명령어 구조
▪ 100~250개로 명령어 수가 많음	명령어 세트의 크기	▪ 레지스터 기반의 명령어들로서 명령어 수가 적음
▪ 가변 형식 : 명령어당 16~64 비트	명령어 형식	▪ 고정 형식 : 32비트
▪ 복잡함 : 12~24개	주소 지정 모드	▪ 간단함 : 3~5개
▪ 적은 레지스터 : 8~24개 정도 - 느린 처리 속도 - 높은 호환성	레지스터	▪ 많은 레지스터 : 32~192개 - 빠른 처리 속도 - 낮은 호환성
▪ 제어 메모리(ROM) 사용	CPU 제어 방식	▪ 하드 와이어드(Hard-wired) 방식 사용

### 3] 명령어 파이프라이닝



#### 명령어 파이프라이닝

CPU의 프로그램 처리속도를 높이기 위해 CPU 내부 하드웨어를 여러 단계로 나누어 병행 처리하도록 하는 기술

**예** 명령어 수행과 다음 명령어 인출을 동시에 수행

- 2단계 명령어 파이프라인
- 4단계 명령어 파이프라인
- 6단계 명령어 파이프라인
- 파이프라인에 의한 속도 향상

### 4] 병렬 처리 시스템



◇ 처리 능력이 뛰어난 컴퓨터 시스템의 구현 방법

고성능의 단일 프로세서를 이용하여 시스템을 구성

다수의 프로세서를 사용하여 많은 연산을 동시에 수행하는 시스템을 구성

다수의 프로세서가 동시에 정보를 처리하는 것을 병렬 처리

## 4] 병렬 처리 시스템



### ◇ 마이클폴린의 분류법

단일 명령어 흐름, 단일 데이터 흐름  
(Single Instruction Stream, Single Data Stream)

단일 명령어 흐름, 다중 데이터 흐름  
(Single Instruction Stream, Multiple Data Stream)

다중 명령어 흐름, 단일 데이터 흐름  
(Multiple Instruction Stream, Single Data Stream)

다중 명령어 흐름, 다중 데이터 흐름  
(Multiple Instruction Stream, Multiple Data Stream)

## 4] 병렬 처리 시스템

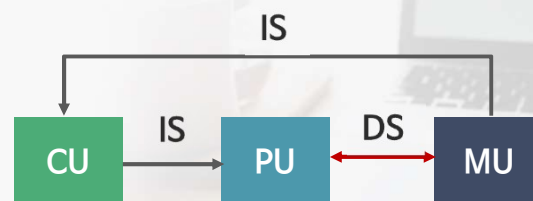


### ◇ 마이클폴린의 분류법

#### 1 단일 명령어 흐름, 단일 데이터 흐름

- 하나의 명령어로 하나의 데이터를 처리하는 구조로서, 폰 노이만 구조에 해당
- 단일 처리기 시스템

#### SISD의 동작 방식



## 4] 병렬 처리 시스템

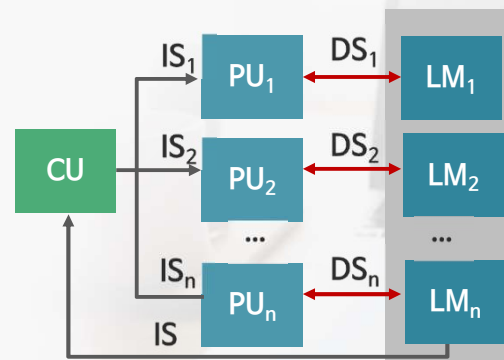
학습하기

### ◇ 마이클폴린의 분류법

#### 2 단일 명령어 흐름, 다중 데이터 흐름

- 하나의 명령어로 다수의 데이터를 처리하는 구조
- 다중처리기 시스템
- 벡터 프로세서나 배열 프로세서 시스템

#### SIMD의 동작 방식



## 4] 병렬 처리 시스템

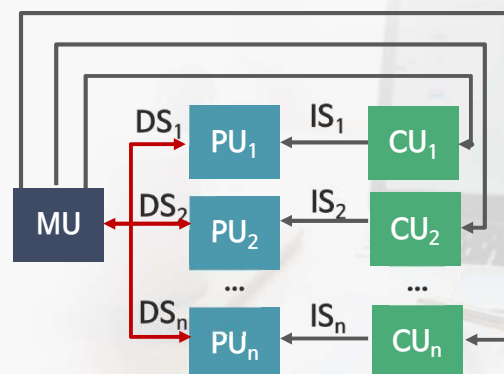
학습하기

### ◇ 마이클폴린의 분류법

#### 3 다중 명령어 흐름, 단일 데이터 흐름

- 다수의 처리장치들이 동일한 데이터에 대해 서로 다른 연산(명령어)을 실행하는 구조
- 파이프라인 구조와 결합 허용 시스템

#### MISD의 동작 방식



## 4] 병렬 처리 시스템

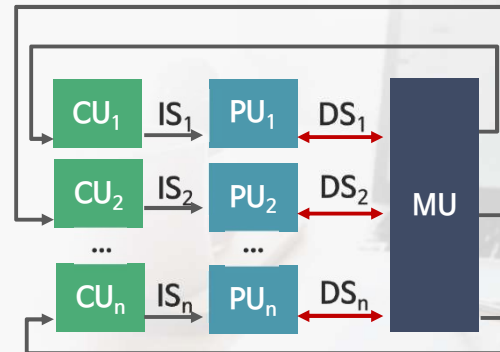
학습하기

### ◆ 마이클폴린의 분류법

#### 4 다중 명령어 흐름, 다중 데이터 흐름

- 독립적으로 동작하는 다수의 처리장치들이 서로 다른 데이터에 대해 상이한 명령어를 동시에 처리하는 구조
- 공유 메모리 시스템과 분산 메모리 시스템

#### MIMD의 동작 방식



## 정리하기

정리하기

### 컴퓨터 시스템의 기본 구조

#### ● 저장 프로그램 방식

- 메모리에 자료뿐만 아니라 프로그램도 저장되는 프로그램 내장 방식

### 기억장치

- 컴퓨터가 작동하는 동안 중앙처리장치가 해야 할 작업 내용인 프로그램 명령어와 프로그램에서 이용할 자료를 저장하고 있는 기억장치

## SUMMARY

# 정리하기

정리하기

### 중앙처리장치

- 메모리에 저장된 프로그램과 자료를 이용하여 실제 작업을 수행하는 회로 장치

### 마이크로프로세서

- 마이크로프로세서의 성능 결정 요소
  - 사이클 당 연산 수와 자료 버스의 폭
  - 레지스터의 수와 크기
  - 캐시 메모리의 크기 등

## ANNOUNCEMENT

# 차시예고

정리하기

1주차 2주차 3주차 **4주차** 5주차 6주차 중간고사

프로그래밍 언어

- 수고하셨습니다.