

컴퓨터 구조

1강. 컴퓨터시스템 개요

권오상 교수



제1강 컴퓨터시스템 개요

1.1 컴퓨터의 기본 구조

1.2 정보의 표현과 저장

1.3 시스템의 구성

학습 목표

1. 컴퓨터의 기본적인 구조에 대해 학습한다.
2. 정보를 어떻게 표현하고 저장하는지 알아본다.
3. 컴퓨터시스템이 어떻게 구성되는지 살펴본다.

하드웨어와 소프트웨어

□ 하드웨어(hardware)

- 컴퓨터 정보들의 전송 통로를 제공해 주고, 그 정보에 대한 처리가 실제 일어나게 해주는 물리적인 실체들

□ 소프트웨어(software)

- 정보들이 이동하는 방향과 정보 처리의 종류를 지정해주고, 그러한 동작들이 일어나는 시간을 지정해주는 명령(command)들의 집합
- 시스템 소프트웨어(system software) : OS(WinXP, Unix, Linux 등)
- 응용 소프트웨어(application software) : 워드프로세서, 웹 브라우저, MS-Excel 등

1.1 컴퓨터의 기본 구조

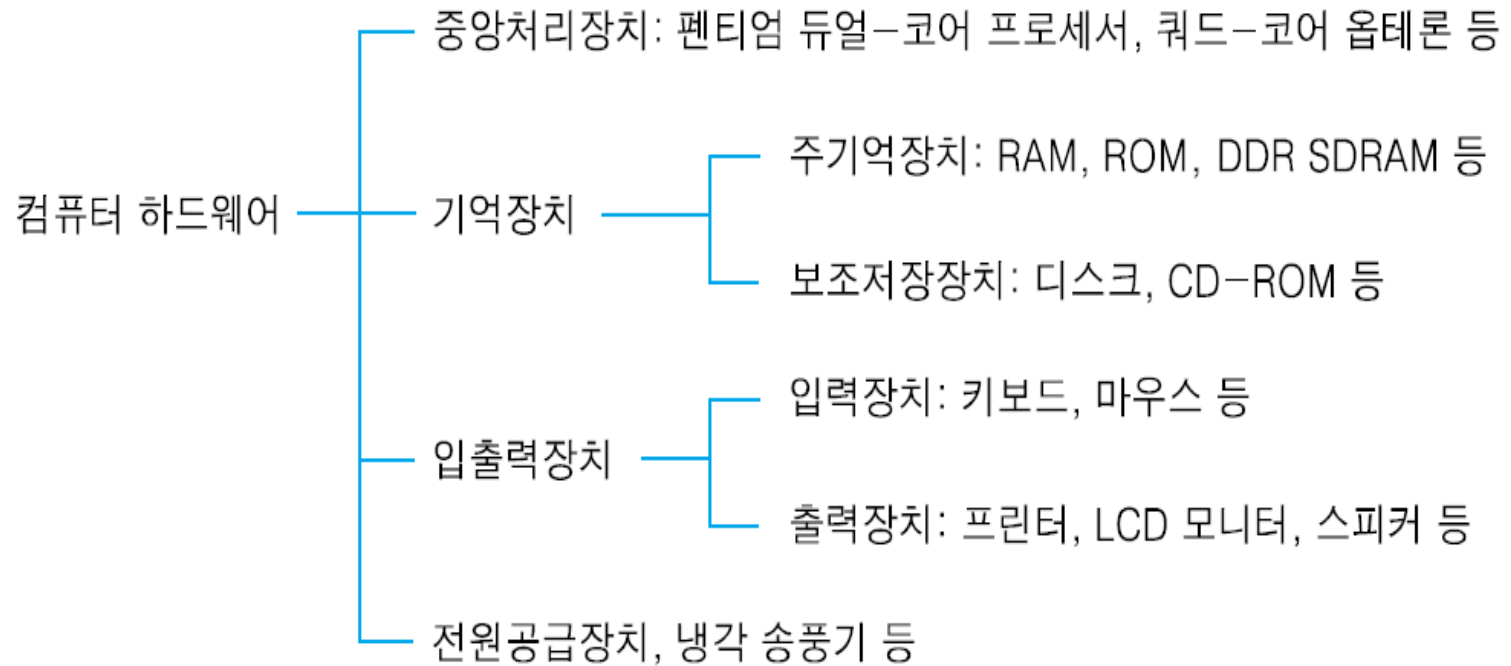
컴퓨터시스템의 구성

응용 소프트웨어
(application software)

시스템 소프트웨어(system software)

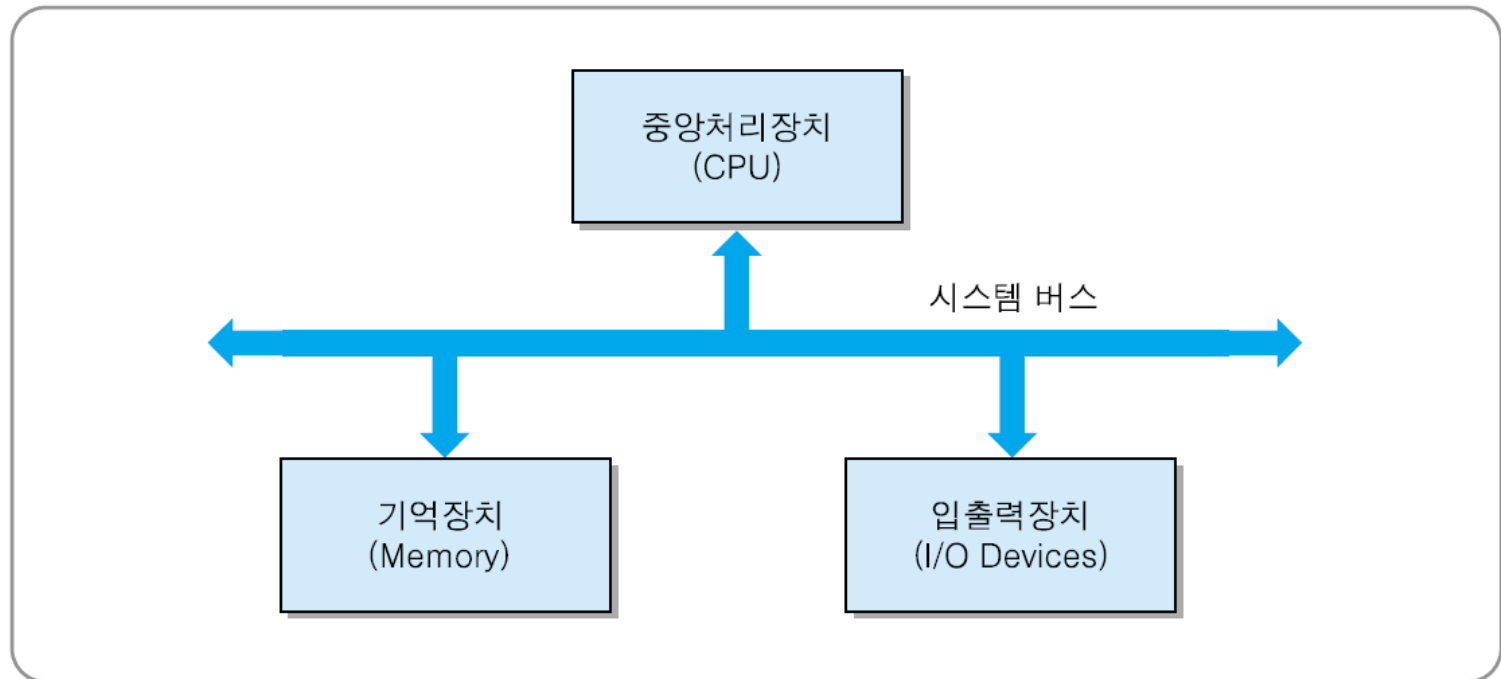
하드웨어(hardware)

컴퓨터 하드웨어의 주요 요소들



컴퓨터의 기본 구조

- 기본 기능: 프로그램 코드를 정해진 순서대로 실행하는 것
 - 필요한 데이터를 읽어서(read), 처리(processing)하고, 저장(store)한다



16p

컴퓨터의 주요 구성요소들

□ 중앙처리장치(Central Processing Unit: CPU)

- 프로세서(processor)
- '프로그램 실행'과 '데이터 처리'라는 중추적인 기능의 수행을 담당하는 요소

□ 기억장치

(1) 주기억장치(main memory)

- CPU 가까이 위치하며, 반도체 기억장치 칩들로 구성
- 고속 액세스
- 가격이 높고 면적을 많이 차지 → 저장 용량의 한계
- 영구 저장 능력이 없기 때문에, 일시적 저장장치로만 사용

컴퓨터의 주요 구성요소들 (계속)

(2) 보조저장장치(auxiliary storage device)

- 2차 기억장치(secondary memory)
- 저장 밀도가 높고, 비트 당 가격이 낮음
- 읽기/쓰기 속도가 느림
- 영구 저장 능력을 가진 저장장치 : 하드 디스크(hard disk), 플래시 메모리(flash memory), CD-ROM, 등

□ 입출력장치(I/O device)

- 사용자와 컴퓨터간의 대화를 위한 입력 및 출력장치
- 유무선 통신 네트워크 인터페이스 장치

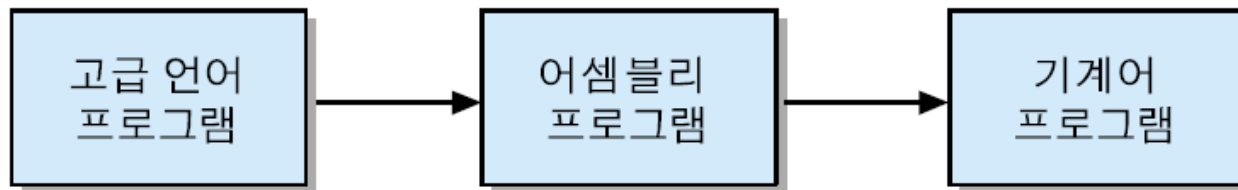
1.2 정보의 표현과 저장

- ❑ **컴퓨터 정보**: 2진수 비트들로 표현된 프로그램 코드와 데이터
- ❑ **프로그램 코드**
 - **기계어(machine language)**
 - 기계 코드(machine code)라고도 함
 - 컴퓨터 하드웨어 부품들이 이해할 수 있는 언어로서, 2진 비트들로 구성
 - **어셈블리 언어(assembly language)**
 - 고급 언어와 기계어 사이의 중간 언어
 - 어셈블러(assembler)로 번역시, 기계어와 일대일 대응
 - **고급 언어(high-level language)**
 - 영문자와 숫자로 구성되어 사람이 이해하기 쉬운 언어
 - C, C++, PASCAL, FORTRAN, COBOL 등
 - 컴파일러(compiler)를 이용하여 기계어로 번역

프로그램 언어의 변환 과정

[예] $Z = X + Y$

- **LOAD A, X**: 기억장치 X번지의 내용을 읽어서, 레지스터 A에 적재(load)
- **ADD A, Y**: 기억장치 Y번지 내용을 읽어서, 레지스터 A에 적재된 값과 더하고, 결과를 다시 A에 적재
- **STOR Z, A**: 그 값을 기억장치 Z 번지에 저장(store)



[예] $Z = X + Y$

LOAD A, X
ADD A, Y
STOR Z, A

00100101
10000110
01000111

18p

프로그래밍 언어 번역 소프트웨어

□ 컴파일러(compiler)

- 고급언어 프로그램을 기계어 프로그램으로 번역하는 소프트웨어

□ 어셈블러(assembler)

- 어셈블리 프로그램을 기계어 프로그램으로 번역하는 소프트웨어
- 니모닉스(mnemonics)
 - 어셈블리 명령어가 지정하는 연산을 가리키는 알파벳 기호
 - ‘LOAD’, ‘ADD’, ‘STOR’ 등

기계 명령어의 형식

연산코드	오퍼랜드
0 0 1	0 0 1 0 1

19p

□ 연산 코드(op code)

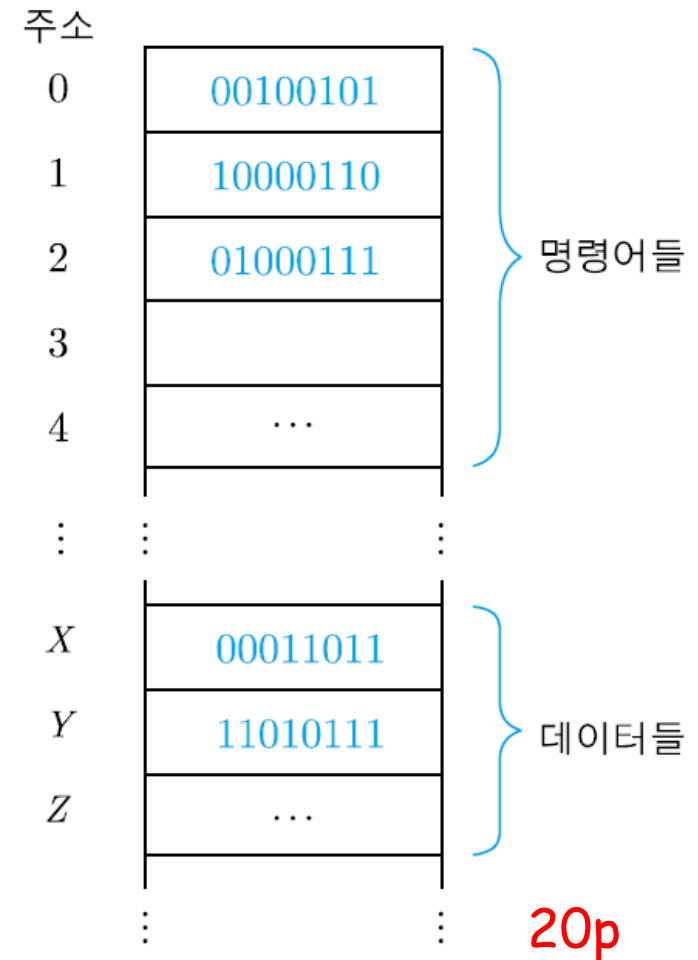
- CPU가 수행할 연산을 지정해 주는 비트들
- 비트 수 = 3이라면, 지정될 수 있는 연산의 최대 수: $2^3 = 8$ 개

□ 오퍼랜드(operand)

- 연산에 사용될 데이터 혹은 그것이 저장되어 있는 기억장치 주소 (memory address)
- 비트 수 = 5 라면, 주소지정(addressing) 할 수 있는 기억 장소의 최대 수: $2^5 = 32$ 개

프로그램 코드와 데이터의 기억장치 저장

- ❑ 프로그램 코드(명령어)와 데이터는 지정된 기억 장소에 저장
- ❑ 단어(word) 단위로 저장
 - 단어: 각 기억 장소에 저장되는 정보의 기본 단위로서, CPU에 의해 한 번에 처리될 수 있는 비트들의 그룹
 - 단어 길이의 예: 8비트, 32비트, 64비트, 128비트
 - 주소지정 단위: 단어 단위 혹은 바이트 단위



1.3 시스템의 구성

1.3.1 CPU와 기억장치의 접속

□ 시스템 버스(system bus)

- CPU와 시스템 내의 다른 요소들 사이에 정보를 교환하는 통로
- 기본 구성
 - 주소 버스(address bus)
 - 데이터 버스(data bus)
 - 제어 버스(control bus)

시스템 버스

□ 주소 버스(address bus)

- CPU가 외부로 발생하는 주소 정보를 전송하는 신호 선들의 집합
- 주소 선의 수는 CPU와 접속될 수 있는 최대 기억장치 용량을 결정

[예] 주소 버스의 비트 수 = 16 비트라면,
최대 $2^{16} = 64K$ 개의 기억 장소들의 주소지정 가능

□ 데이터 버스(data bus)

- CPU가 기억장치 혹은 I/O 장치와의 사이에 데이터를 전송하기 위한 신호 선들의 집합
- 데이터 선의 수는 CPU가 한 번에 전송할 수 있는 비트 수를 결정

[예] 데이터 버스 폭 = 32 비트라면,
CPU와 기억장치 간의 데이터 전송은 한 번에 32 비트씩 가능

시스템 버스 (계속)

- 제어 버스(control bus): CPU가 시스템 내의 각종 요소들의 동작을 제어하기 위한 신호 선들의 집합

[예]

- 기억장치 읽기/쓰기(Memory Read/Write) 신호
- I/O 읽기/쓰기(I/O Read/Write) 신호
- 인터럽트(Interrupt) 신호
- 버스 제어(Bus Control) 신호

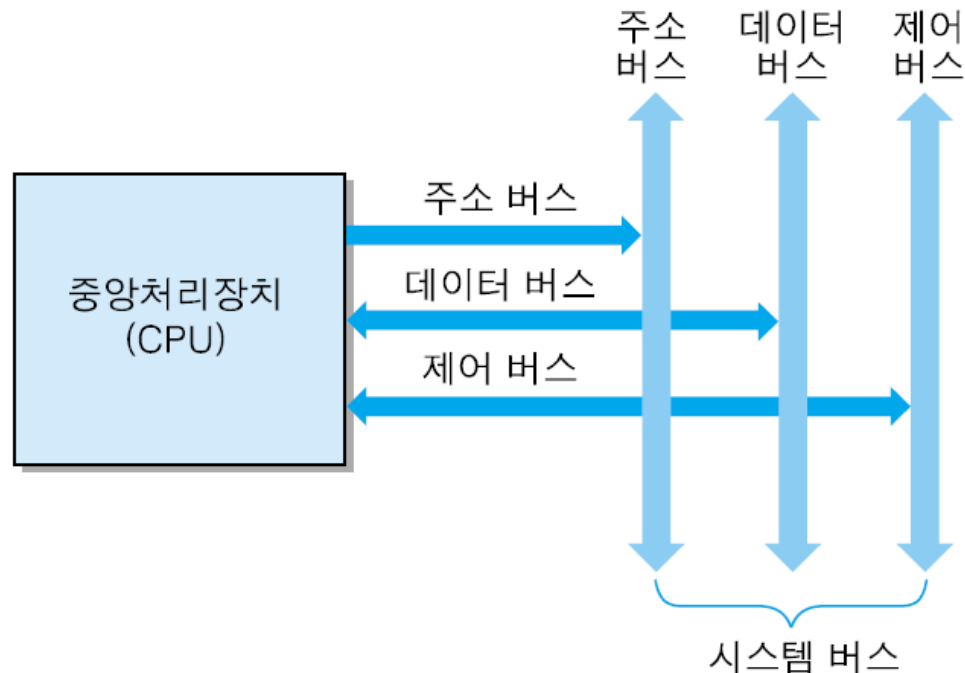
CPU와 시스템 버스 간의 접속

□ 주소 버스 : 단방향성(unidirectional)

- 주소는 CPU로부터 기억장치 혹은 I/O 장치들로 보내지는 정보

□ 데이터 버스, 제어 버스 : 양방향성(bidirectional)

- 읽기와 쓰기 동작을 모두 지원

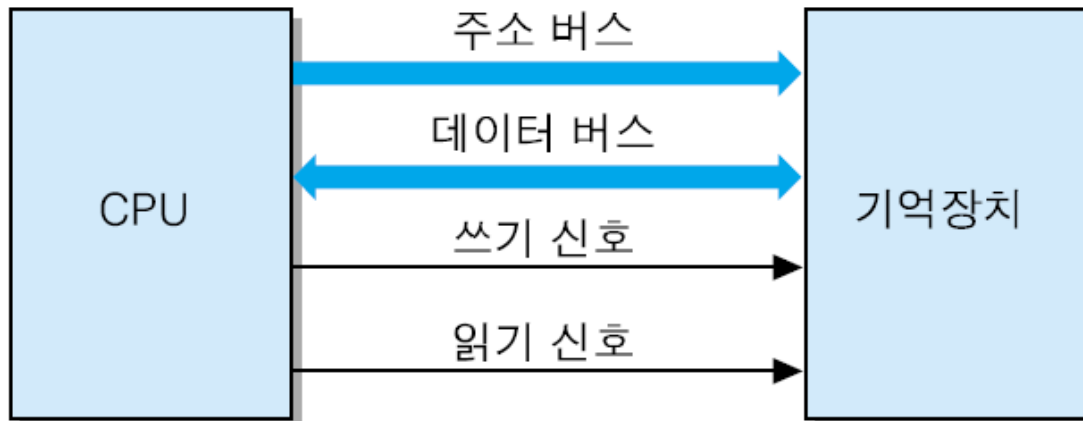


22p

CPU와 기억장치 간의 접속

□ 필요한 버스 및 제어신호

- 주소 버스
- 데이터 버스
- 제어 신호: 기억장치 읽기(memory read) 신호, 기억장치 쓰기(memory write) 신호



22p

CPU와 기억장치 간의 접속 (계속)

- 기억장치 쓰기 동작 : CPU가 데이터를 저장할 기억 장소의 주소와 저장할 데이터를 각각 주소 버스와 데이터 버스를 통하여 보내는 동시에, 쓰기 신호를 활성화
- 기억장치 쓰기 시간(memory write time) : CPU가 주소와 데이터를 보낸 순간부터 저장이 완료될 때까지의 시간

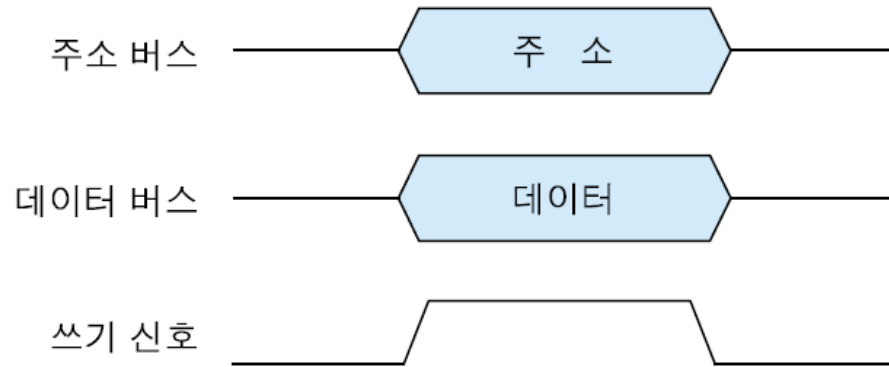
CPU와 기억장치의 접속 (계속)

□ 기억장치 읽기 동작

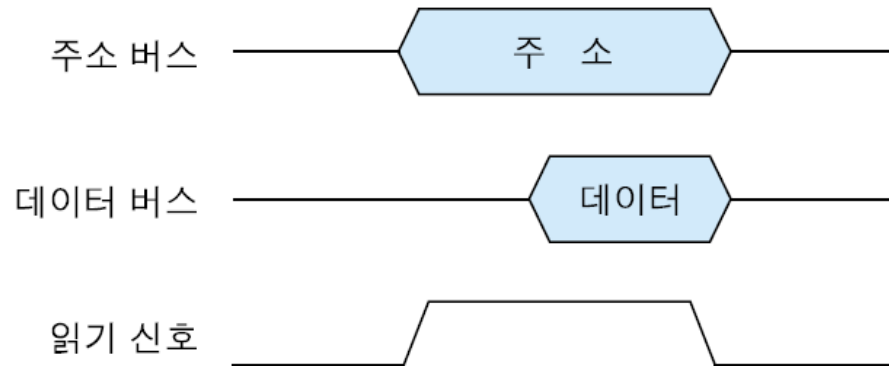
- CPU가 기억장치 주소를 주소 버스를 통하여 보내는 동시에, 읽기 신호를 활성화
- 일정 지연 시간이 경과한 후에 기억장치로부터 읽혀진 데이터가 데이터 버스 상에 실리고, CPU는 그 데이터를 버스 인터페이스 회로를 통하여 읽음

□ 기억장치 읽기 시간(memory read time) : 주소를 발생한 시간부터 기억 장치의 데이터가 CPU에 도착할 때까지의 시간

기억장치 액세스(읽기/쓰기) 동작의 시간 흐름도



(a) 기억장치 쓰기 동작의 시간 흐름도



(b) 기억장치 읽기 동작의 시간 흐름도

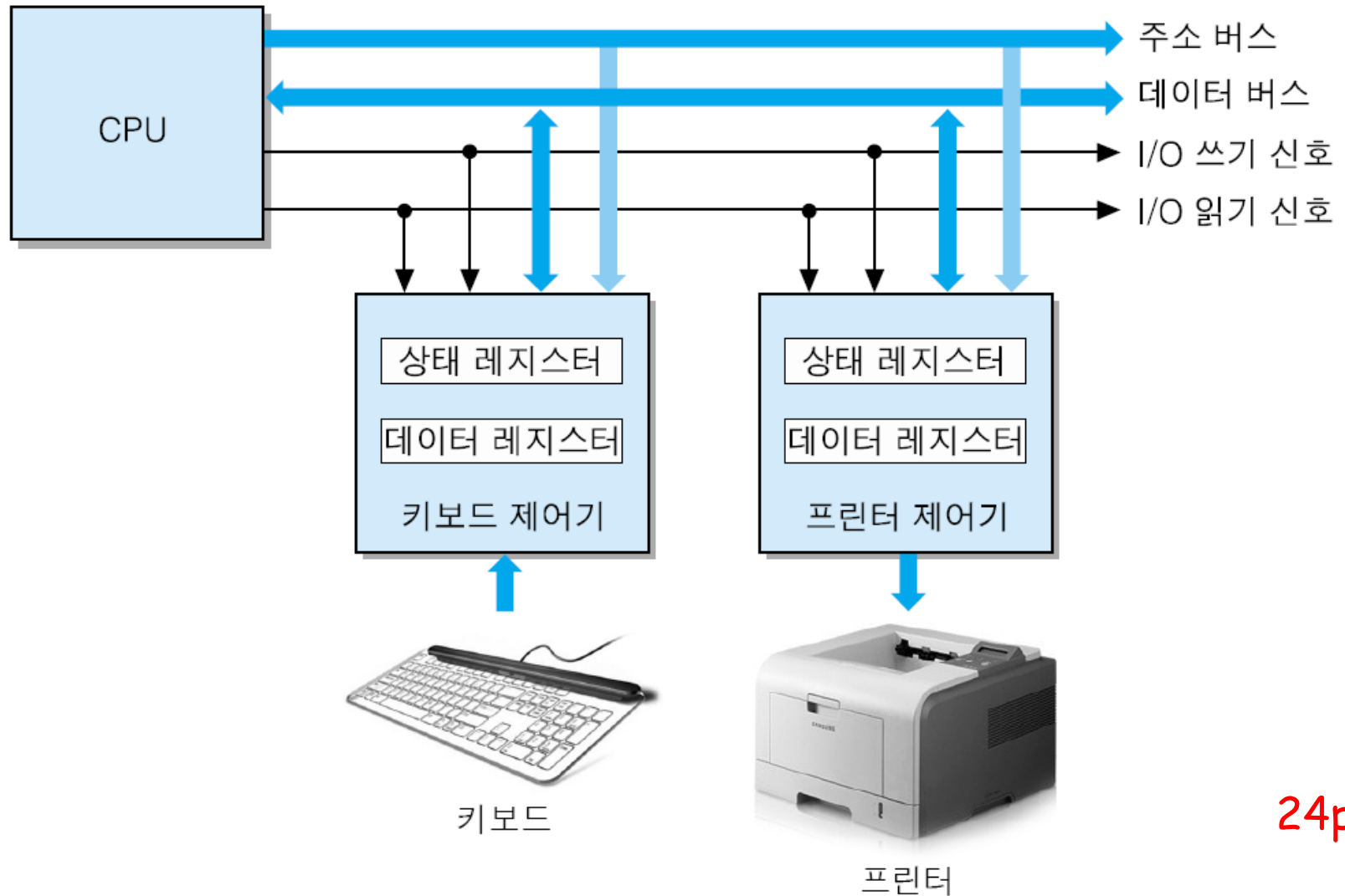
1.3.2 CPU와 I/O 장치의 접속

□ 필요한 버스 및 제어 신호

- 주소 버스
- 데이터 버스
- 제어 신호 : I/O 읽기 신호, I/O 쓰기 신호

□ 접속 경로: CPU – 시스템 버스 – I/O 장치 제어기 – I/O 장치

I/O 장치 접속 사례 : CPU - 키보드 & 프린터



24p

I/O 장치 제어기(I/O device controller)

❑ **기능:** CPU로부터 I/O 명령을 받아서, 해당 I/O 장치를 제어하고, 데이터를 이동함으로써 명령을 수행하는 전자회로 장치 (예: 키보드 제어기, 프린터 제어기 등)

❑ 상태 레지스터

- I/O 장치의 현재 상태를 나타내는 비트들을 저장한 레지스터
- 준비 상태(RDY) 비트, 데이터 전송확인(ACK) 비트 등

❑ 데이터 레지스터

- CPU와 I/O 장치 간에 이동되는 데이터를 일시적으로 저장하는 레지스터

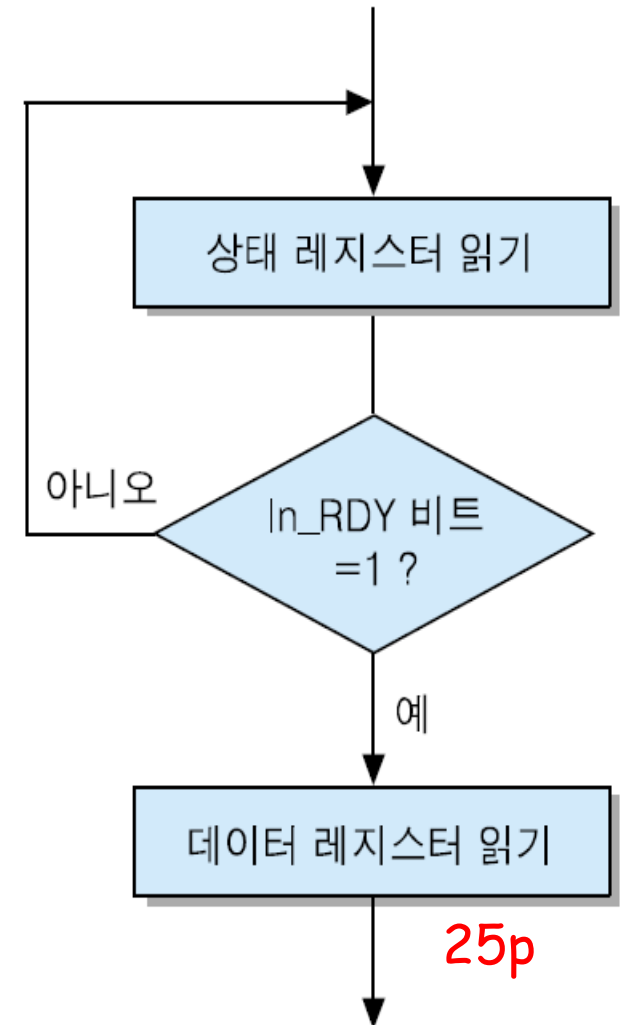
키보드의 데이터 입력 과정

□ 키보드 제어기 :

- 키보드의 어떤 한 키(key)를 누르면,
그 키에 대응되는 ASCII 코드가 키보드 제어
기의 데이터 레지스터에 저장되고, 동시에
상태 레지스터의 In_RDY 비트가 1로 세트

□ CPU :

- 키보드 제어기로부터 상태 레지스터의 내용을 읽어서 In_RDY 비트가 세트 되었는지 검사(In_RDY 비트는 데이터 레지스터에 외부로부터 데이터가 적재되었는지를 표시)
- 만약 세트 되지 않았으면, 1번을 반복하며 대기. 만약 세트 되었다면, 데이터 레지스터의 내용을 읽음



프린터의 데이터 출력 과정

□ CPU :

- ① 프린터 제어기의 상태 레지스터 내용을 읽어서 Out_RDY 비트 검사 (Out_RDY 비트: 프린터가 출력할 준비가 되었는지를 표시)
- ② 만약 세트 되지 않았으면, ① 번을 반복하며 대기
만약 세트 되었다면, 프린트할 데이터를 프린터 제어기의 데이터 레지스터에 저장

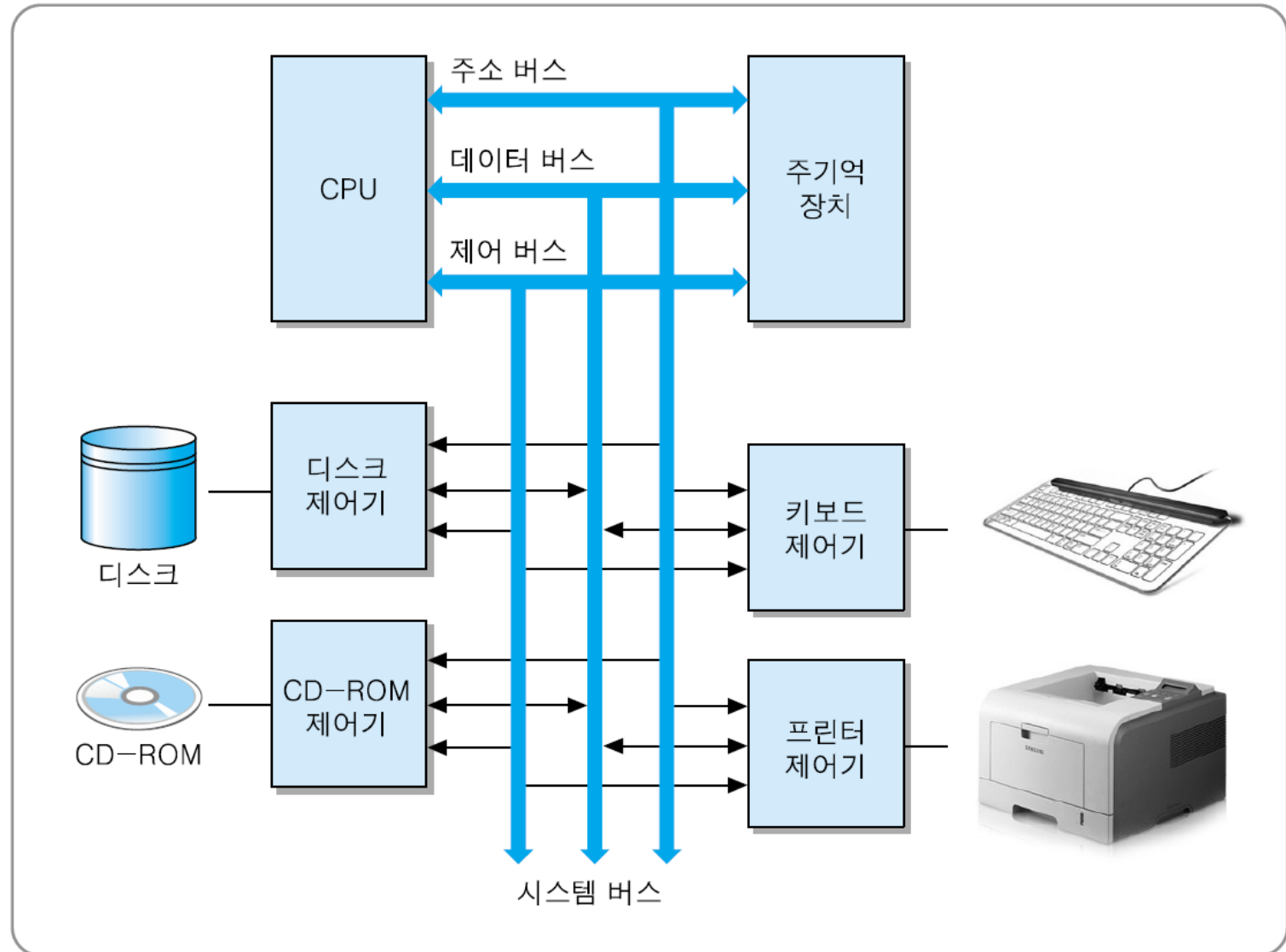
□ 프린터 제어기 :

- ① 데이터 레지스터의 내용을 프린터로 보내고,
- ② 프린터의 하드웨어를 제어하면서 인쇄 동작 수행

CPU와 보조저장장치의 접속

- 보조저장장치들(디스크, 플래시 메모리, CD-ROM 등)도 각 장치를 위한 제어기를 통하여 키보드나 프린터와 유사한 방법으로 접속
 - 차이점 : 데이터 전송 단위
 - 키보드, 프린터 : 바이트(8 비트) 단위로 전송
 - 보조저장장치 : 블록(512/1024/4096 바이트) 단위로 전송
- 제어기 내에 한 블록 이상을 임시 저장할 수 있는 데이터 버퍼 필요
- 트랙 버퍼(track buffer): 하드 디스크상의 한 트랙 내용을 모두 저장할 수 있는 디스크 제어기 내의 데이터 버퍼

1.3.3 컴퓨터시스템의 전체 구성



27p

컴퓨터의 기본적인 기능들

- ❑ **프로그램 실행** : CPU가 주기억장치로부터 프로그램 코드를 읽어서 실행
- ❑ **데이터 저장** : 프로그램 실행 결과로서 얻어진 데이터를 주기억 장치에 저장
- ❑ **데이터 이동** : 디스크 혹은 CD-ROM에 저장되어 있는 프로그램과 데이터 블록을 기억장치로 이동
- ❑ **데이터 입력/출력** : 사용자가 키보드를 통하여 보내는 명령이나 데이터를 읽어 들인다. 또한 CPU가 처리한 결과값이나 기억장치의 내용을 프린터(혹은 모니터)로 출력
- ❑ **제어** : 프로그램이 순서대로 실행되도록 또는 필요에 따라 실행 순서를 변경하도록 조정하며, 각종 제어 신호들을 발생


학습 정리

- ❑ **컴퓨터시스템**은 하드웨어와 소프트웨어로 구성되어 있습니다.
- ❑ **하드웨어**는 중앙처리장치, 기억장치, 입출력장치, 그리고 기타 장치로 구성되어 있습니다.
- ❑ **소프트웨어**는 시스템 소프트웨어와 응용 소프트웨어로 나누어 집니다.
- ❑ **컴퓨터 정보**는 2진수로 표현된 프로그램 코드와 데이터입니다.
- ❑ **프로그램 코드**는 기계어, 어셈블리 언어, 고급언어로 나누어 집니다.
- ❑ **컴파일러**는 고급언어 프로그램을 기계어 프로그램으로 번역하는 소프트웨어이며, **어셈블러**는 어셈블리 프로그램을 기계어 프로그램으로 번역하는 소프트웨어입니다.


- ❑ **단어(word)**는 각 기억 장소에 저장되는 데이터의 기본 단위로서 CPU에 의해 한 번에 처리될 수 있는 비트들의 그룹을 말합니다.
- ❑ **시스템 버스**는 CPU와 시스템 내의 다른 요소들 사이에 정보를 교환하는 통로이며, 주소 버스, 데이터 버스, 제어 버스로 구성됩니다.
- ❑ 컴퓨터의 기본적인 기능으로서 프로그램 실행, 데이터 저장, 데이터 이동, 데이터 입력/출력, 제어 등이 있습니다.

참고 문헌

컴퓨터구조론, 김종현 저, 생능출판사, 2014.



이 강의록은 저작권법에 의해 보호받는 저작물로서 저작권자의 허락 없이 저작재산권 일체 (복제권, 배포권, 대여권, 공연권, 공중전송권, 전시권, 2차적 저작물 작성권)를 침해 시 저작권법에 의거 처벌받을 수 있습니다.



계동캠퍼스(03051)서울특별시 종로구 북촌로 106 **안암캠퍼스**(02841)서울특별시 성북구 안암로 145
고려대학교 안암캠퍼스내