



컴퓨터 구조

8. 입출력장치 (보조기억장치 포함)



01 장치 컨트롤러, 장치 드라이버

KEY WORD 장치 컨트롤러, 장치 드라이버

장치 컨트롤러 device controller

(또는 입출력 제어기 I/O controller, 입출력 모듈 I/O module)

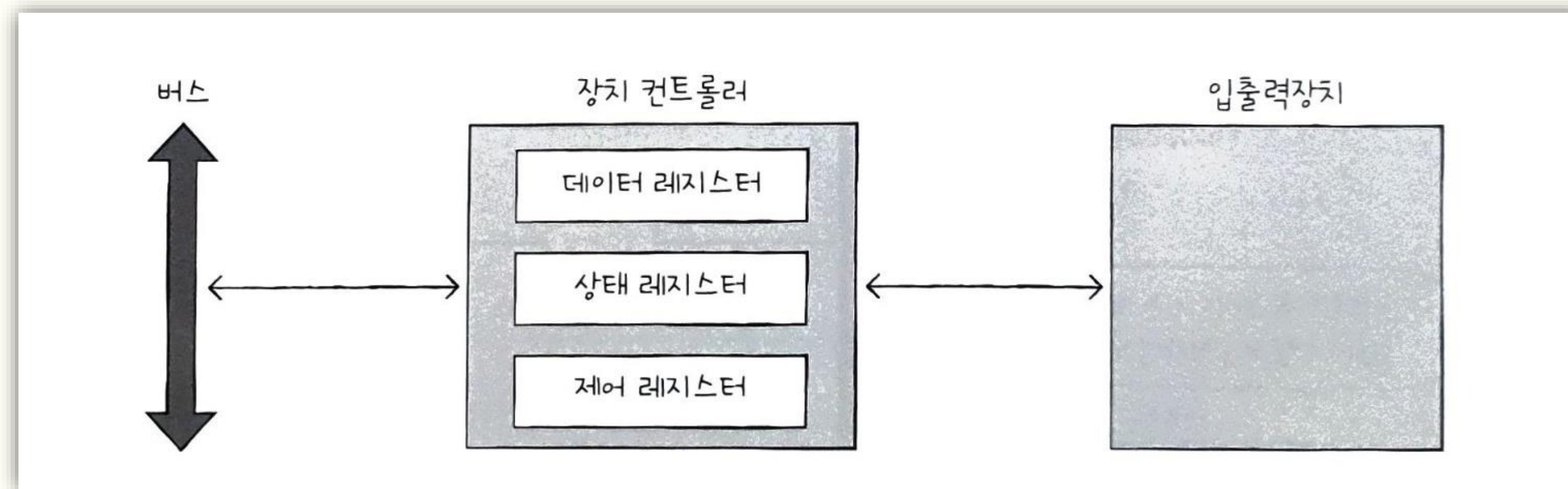
- 입출력장치 종류가 많아서, 정보 주고받는 방식의 규격화 어려워!
- CPU, 메모리의 데이터 전송률은 높지만 입출력장치는 낮아서, 같은 시간에 주고받는 데이터 양이 달라!
- 즉, 중간에 **장치 컨트롤러**가 필요하다.
 - ✓ CPU와 입출력장치 간의 통신 중개 : 정보 규격화 해결!
 - ✓ 오류 검출 : 입출력장치 오류 검출
 - ✓ **데이터 버퍼링** : 임시 공간인 **버퍼** buffer 에 데이터를 모았다가 내보내는 방법.



조금 받아서 한꺼번에 보내거나, 많이 받아서 조금씩 보내는 방법 (전송률 높은 장치 \leftrightarrow 낮은 장치)

장치 컨트롤러 내부 구조

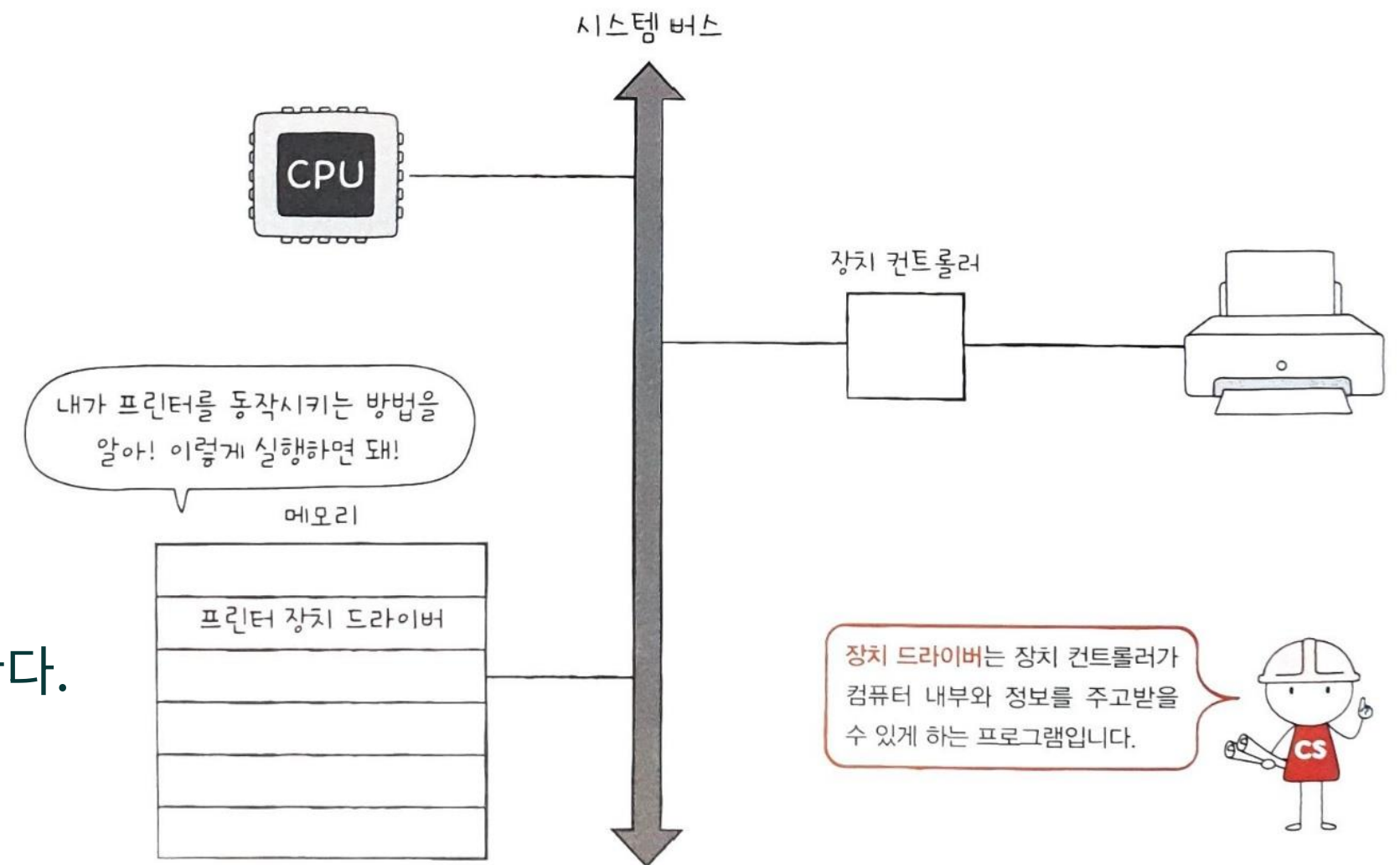
- **데이터 레지스터**: CPU와 입출력장치 사이에 주고받을 **데이터** 담음 (버퍼 역할)
- **상태 레지스터**: 입출력 준비 되었는지, 작업 완료 되었는지, 오류는 없는지 **상태 정보** 저장
- **제어 레지스터**: 입출력장치가 수행할 내용에 대한 제어 정보, **명령**을 저장



장치 드라이버 device driver

(또는 입출력 제어기 I/O controller, 입출력 모듈 I/O module)

- 입출력장치 연결을 위한 **소프트웨어적** 통로
- 장치 컨트롤러는 **하드웨어적** 통로이며,
드라이버가 컨트롤러의 동작을 감지하고 제어함.
- 운영체제가 기본으로 제공하는 경우도 있으나,
필요한 경우 장치 제작사가 제공하는 것을 직접 설치해야 한다.





02 다양한 입출력 방법

KEY WORD

프로그램 입출력, 메모리 앱 입출력, 고립형 입출력, 인터럽트 기반 입출력,
DMA 입출력, 입출력 버스

시작하기 전에...

1. 컴퓨터에 입출력 장치가 연결되었다.
2. 그리고 입출력 작업을 수행하려고 한다.
3. 그러려면 CPU와 장치 컨트롤러가 정보를 주고 받아야 한다.

➔ 이때 3가지 방법으로 정보를 주고 받는다.

- ① 프로그램 입출력
- ② 인터럽트 기반 입출력
- ③ DMA 입출력

① 프로그램 입출력 programmed I/O

- 프로그램 속 명령어로 입출력장치를 제어하는 방법
- P. 241

- 이때, CPU는 입출력장치의 주소를 어떻게 아는 걸까?
(= 입출력 컨트롤러의 레지스터가 몇번지인지 어떻게 아는 걸까?)

① 메모리 맵 입출력

- 메모리와 입출력장치가 같은 주소공간 사용

예) 1,024개의 주소공간을 가진 컴퓨터가 있을 때,
512개는 메모리 주소를, 512개는 장치 컨트롤러의 레지스터를 표현하기 위해 사용.

- 메모리와 입출력장치에 같은 명령어 사용

② 고립형 입출력

- 메모리와 입출력장치가 분리된 주소공간 사용

예) 1,024개의 주소공간을 가진 컴퓨터가 있을 때, 둘 다 각각의 1,024개의 주소공간 사용.
대신 같은 제어 버스를 사용해 접근한다. (P. 243)

- 메모리와 입출력장치에 다른 명령어 사용

② 인터럽트 기반 입출력 Interrupt-Driven I/O

- (복습) 인터럽트? 인터럽트 신호를 보내면 CPU는 하던 일을 멈추고 해당 인터럽트 처리
- 장치 컨트롤러가 인터럽트 요청 신호를 보내면, CPU는 하던 일을 백업하고 인터럽트 서비스 루틴 실행
- 다중 인터럽트 발생시, 순차적 또는 비순차적으로 처리 가능
- NMI Non-Maskable Interrupt가 발생한 경우 CPU는 우선순위가 높은 인터럽트부터 처리
- 대부분의 컴퓨터는 PIC (Programmable Interrupt Controller) 하드웨어를 사용하여 다중 인터럽트 처리. 우선순위를 CPU에게 알려줌.

③ DMA 입출력 Direct Memory Access

- 기존: 메모리 \leftrightarrow 입출력장치간에 데이터를 주고 받을 때 CPU를 거쳐야했음. (CPU는 가뜰이나 바쁜데!!)
- DMA 입출력 방법: CPU를 거치지 않고 직접 메모리에 접근할 수 있는 방법.
- DMA 컨트롤러: DMA 입출력을 위해 시스템 버스에 연결된 하드웨어. 입출력 과정은 다음과 같다.
 - ① CPU가 DMA에 입출력 작업 명령.
 - ② 장치 컨트롤러와 상호작용하며 입출력 작업 수행. 이때 필요한 경우 메모리에 직접 접근. (CPU 거치지 않음)
 - ③ 입출력 과정이 끝나면 DMA는 CPU에 인터럽트 요청 신호 보냄.
- 입출력 버스: DMA가 시스템 버스 대신 메모리에 접근하는 경로 (버스).
CPU와 동일한 버스 사용을 피하기 위함임



오늘 배운 것

- 01.** 장치 컨트롤러에 대해 이해한다.
- 02.** 장치 드라이버에 대해 이해한다.
- 03.** 프로그램 입출력 방식이 무엇인지 이해한다.
- 04.** 인터럽트 기반의 입출력 방식을 이해한다.
- 05.** DMA 입출력 방식을 이해한다.



사랑합니다 고객님

다음 발표 일정 : 6장 나머지, 7장