

컴퓨터 구조

1

컴퓨터시스템의 구조 (제 2주 차)

서울사이버대학교

오 창 환

학습목표

2

- 컴퓨터의 기본 구조를 설명할 수 있다.
- 정보의 표현과 저장을 설명할 수 있다.
- 시스템의 구성을 설명할 수 있다.

학습 내용

3

- 컴퓨터의 기본 구조
- 정보의 표현과 저장
- 시스템의 구성

1 교시

컴퓨터의 기본 구조 (1)

5

- 컴퓨터는 프로그램 코드들을 정해진 순서대로 실행하는데, 그 과정에서 (1) 필요한 데이터를 읽어서(read), (2) 처리(processing)하고, (3) 저장(store)함.
- 중앙처리장치(Central Processing Unit : CPU)는 프로세서(processor)라고 불리며 컴퓨터의 특성과 성능을 대부분 결정하게 됨.
- CPU가 한 번에 처리하는 데이터의 길이에 따라 16비트 컴퓨터 혹은 32비트 컴퓨터 등으로 분류되며, 컴퓨터의 이론적인 처리 속도는 CPU의 속도에 의해 결정됨.
- CPU가 실행할 프로그램 코드들과 데이터는 기억장치(memory)에 저장되는데, 기억장치는 크기와 속도 및 용도에 따라 주기억장치(main memory)와 보조기억장치 (auxiliary storage device)로 나누어짐.
- 보조기억장치는 2차 기억장치(secondary memory)라고도 함.

컴퓨터의 기본 구조 (2)

6

- 주기억장치는 CPU 가까이 위치하며 반도체 기억장치 칩들로 이루지기 때문에 고속 액세스가 가능하지만,
가격이 높고 면적을 많이 차지하므로 저장 용량을 크게 할 수 없음.
또한, 영구 저장 능력이 없기 때문에 프로그램 실행 중에 일시적으로만 사용되는 기억장치임.
- 보조저장장치는 디스크(disk)나 CD-ROM과 같은
영구 저장능력을 가진 기억장치들을 말하며,
저장 밀도가 높고 비트당 가격이 낮지만,
기계적인 장치가 포함되기 때문에 속도가 느림.
보조저장장치는 입출력장치로 분류되기도 하는데,
그 이유는 CPU에 의해 직접 액세스되지 못하고 다른 입출력장치들과 마찬가지로 별도의 제어기(controller)를 통해 액세스되기 때문임.

컴퓨터의 기본 구조 (3)

7

- 입출력장치(I/O device)는 입력장치(input device)와 출력장치(output device)로 이루어짐.

이 장치들은 사용자와 컴퓨터 간의 대화를 위한 도구라고 할 수 있는데,

키보드와 모니터 화면 및 프린터 외에도 다양한 장치들이 계속 개발되고 있음.

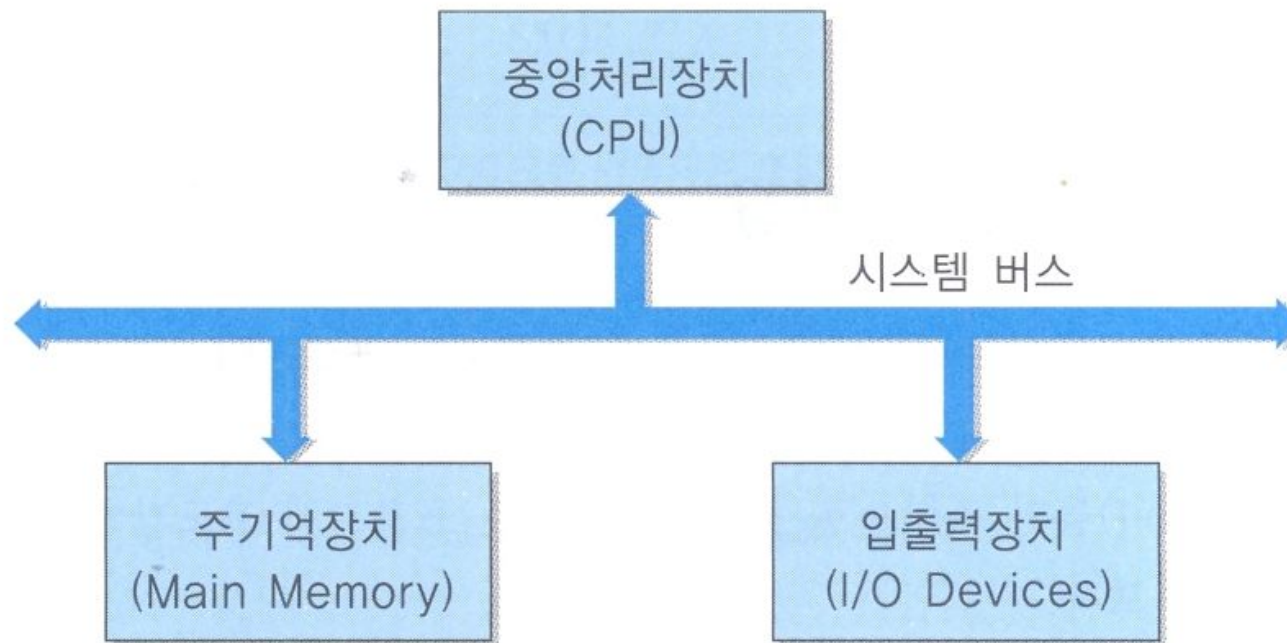
이들은 CPU와 직접 데이터 교환을 할 수 없으며,

각 장치마다 별도의 제어기가 있어서

CPU로부터의 명령을 받아서 장치의 동작을 제어하고 데이터를 이동시키는 일을 수행함.

컴퓨터의 기본 구조 (4)

8



- 컴퓨터의 기본 구조

2 교시

정보의 표현과 저장 (1)

10

- 컴퓨터가 받아들이고 처리하는 정보의 종류로는 프로그램 코드(program code)와 데이터(data)가 있음.
- 디지털컴퓨터에서는 이들이 모두 2진수(binary number)를 나타내는 비트(bit)들의 조합으로 표현됨.
- 컴퓨터 프로그램은 C, PASCAL, FORTRAN, COBOL , JAVA등과 같은 고급 언어를 이용하여 작성됨.

이렇게 작성된 프로그램은 영문자들과 숫자들로 이루어져 있어서
사람들이 이해하기 쉽지만,

디지털 회로 부품들로 이루어진 컴퓨터의 하드웨어 부품들은 전혀 이해하지 못함.
따라서 이 프로그램은 컴파일러(compiler)라고 하는 소프트웨어에 의해
하드웨어 부품들이 이해할 수 있는 언어, 즉 기계어 (machine language) 또는
기계코드(machine code)로 번역됨.

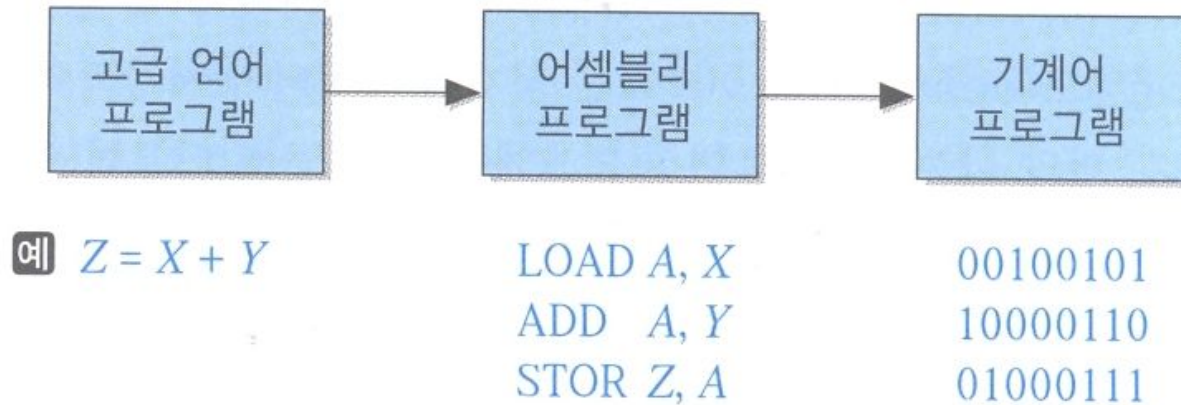
정보의 표현과 저장 (2)

11

- 고급 언어들은 어느 컴퓨터에서 사용되든 거의 동일하지만, 기계어는 CPU마다 서로 다름.
- 고급 언어와 기계어 사이에는 각 CPU 고유의 중간 언어가 존재하는데, 이 언어를 어셈블리 언어(assembly language) 또는 어셈블리 명령어(assembly instruction)라고 부름.
또한 이 언어로 작성된 프로그램을 어셈블리 프로그램(assembly program)이라고 함.
- 고급 언어 프로그램이 어떤 컴퓨터에서 수행되기 위해서는 일단 어셈블리 프로그램으로 번역되고, 그런 다음에 기계 코드로 번역되어야 함.

정보의 표현과 저장 (3)

12



- 고급언어 프로그램의 번역 과정
- 상기의 고급언어 프로그램은 X와 Y를 더하고,
결과 값을 기억장치의 Z번지에 저장하라는 의미인데
어셈블리 프로그램으로 번역하면 다음과 같음.
LOAD A, X : 기억장치 X번지의 내용을 읽어서 레지스터 A에 적재(load)하라.
ADD A, Y : 기억장치 Y번지 내용을 읽어서 레지스터 A에 적재된 값과 더하고,
결과를 다시 A에 적재하라.
STOR Z, A : 레지스터 A에 저장된 값을 Z번지에 저장(store)하라.

정보의 표현과 저장 (4)

13

- 컴파일러의 마지막 결과인 기계어 프로그램은 2진수인 0과 1들의 조합으로 이루어지는데, 이 예에서는 각 어셈블리 명령어가 8비트 기계어로 번역되었음.
- LOAD A, X 명령어에 대한 기계어의 예를 들어보면 다음과 같음.

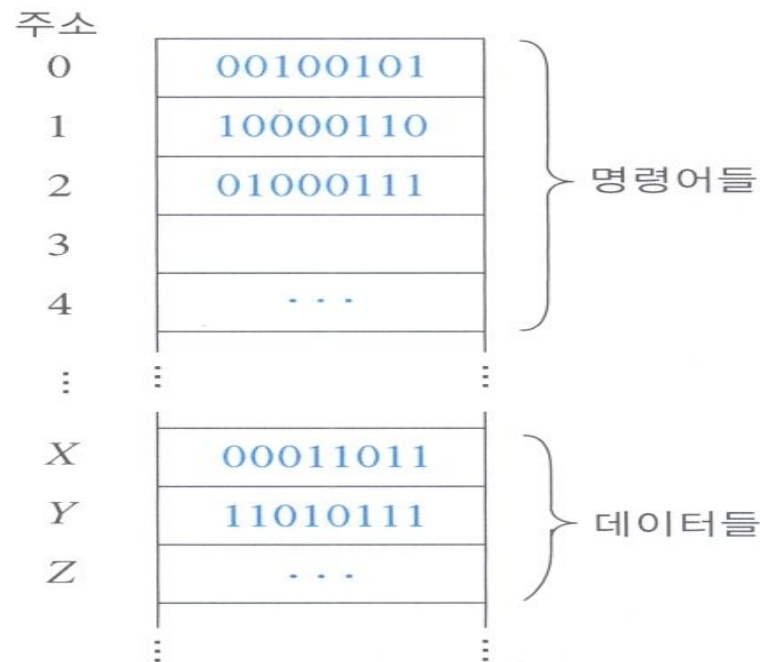
연산코드	오퍼랜드
0 0 1	0 0 1 0 1

- 즉, 연산 코드(op code)인 001은 ‘레지스터 A로 적재하라’라는 연산을 지정해주는 비트들이며, 오퍼랜드(operand)인 00101은 적재될 데이터가 저장되어 있는 기억장치 주소를 나타냄.
- 이 예에서는 연산 코드가 세 비트들로 이루어지므로 2의 3승 = 8가지의 연산들이 지정될 수 있음.
- 또한 오퍼랜드 비트들의 수가 다섯 개이므로 주소를 지정할 수 있는 기억장소들의 최대 수가 2의 5승 = 32개가 됨.

정보의 표현과 저장 (5)

14

- 번역된 기계 코드들은 순서대로 기억장치에 저장되는데
아래 그림은 이 코드들이 데이터들과 함께 기억장치에 저장되어 있는 모습을 보여주고 있음.
- CPU에 의해 한 번에 처리될 수 있는 비트들의 그룹을 단어(word)라고 하는데,
이 예에서는 한 단어의 크기가 8비트, 즉 한 바이트(byte)인 것으로 가정하고 있음.



- 프로그램과 데이터의 저장

3 교시

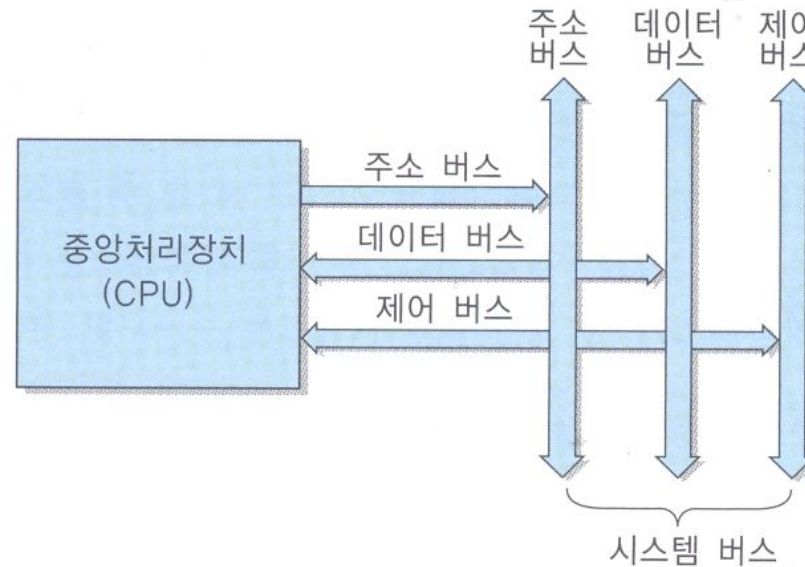
시스템의 구성 (1)

16

- 시스템 버스(system bus)는 주소 버스(address bus), 데이터 버스(data bus), 제어 버스(control bus) 등으로 구성됨.
- 주소 버스는 CPU가 외부로 발생하는 주소 정보를 전송하는 신호 선들의 집합이며, 이 주소 선들의 개수는 CPU와 접속될 수 있는 최대 기억장치 용량을 결정해 줌.
예를 들어서, 주소 비트 수가 16비트라면 2^{16} = 65,536개, 즉 64K개의 기억 장소들을 주소 지정할 수 있음.
- 데이터버스는 CPU가 기억장치 혹은 I/O 장치와의 사이에 데이터를 전송하기 위한 신호 선들의 집합이며,
데이터 선들의 수는 CPU가 한 번에 전송할 수 있는 비트 수를 결정해 줌.
예를 들어서 데이터 버스가 32비트인 시스템에서는
CPU가 기억장치로부터 한 번에 32비트씩 읽어올 수 있음.
- 제어 버스는 CPU가 시스템 내의 각종 요소들의 동작을 제어하기 위한 신호선들의 집합이며, 가장 기본적인 제어 신호들로는 기억장치 읽기/쓰기(memory read/write) 신호와 I/O 읽기/쓰기(I/O read/write) 신호가 있음.

시스템의 구성 (2)

17

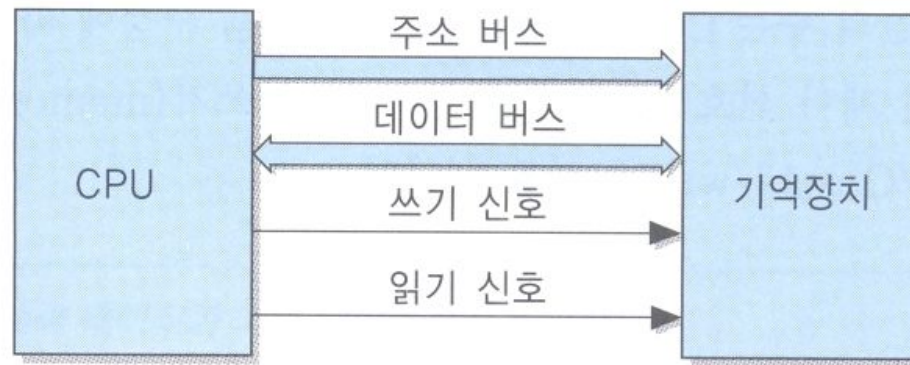


• CPU와 시스템 버스

- 주소는 CPU에 의해 발생되어 기억장치와 I/O 장치로 보내지는 정보이기 때문에 주소 버스는 단방향성이며, 데이터 버스는 읽기와 쓰기 동작을 모두 지원해야 하므로 양방향 전송이 가능해야 함.
- CPU가 기억장치의 특정 장소에 데이터를 저장하거나 저장된 내용을 읽는 동작을 액세스(access)한다고 함.

시스템의 구성 (3)

18

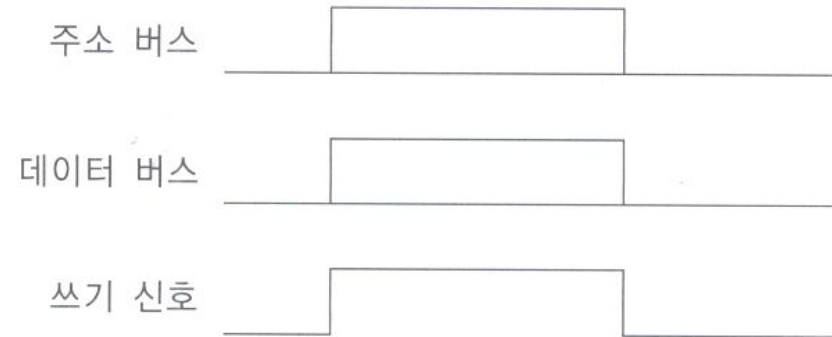


• CPU와 기억장치의 접속

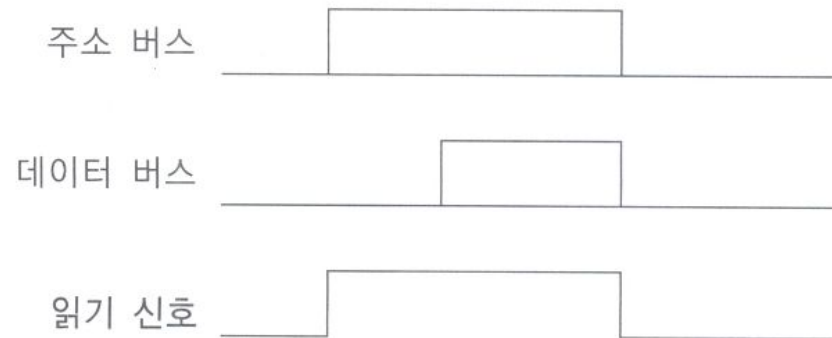
- CPU가 특정 기억 장소로부터 데이터를 읽고자 할 때에 기억장치로 보내주는 정보는 주소와 읽기 신호임.
- 기억 장소에 데이터를 저장하는 경우에는 주소와 데이터 그리고 쓰기 신호를 보내주어야 함.
- 따라서 CPU와 기억장치 사이에는 그러한 정보들의 전송 통로인 주소버스, 데이터버스, 제어버스 등이 필요함.

시스템의 구성 (4)

19



(a) 기억장치 쓰기 동작의 시간 흐름도



(b) 기억장치 읽기 동작의 시간 흐름도

- 기억장치 액세스 동작의 타이밍 다이어그램

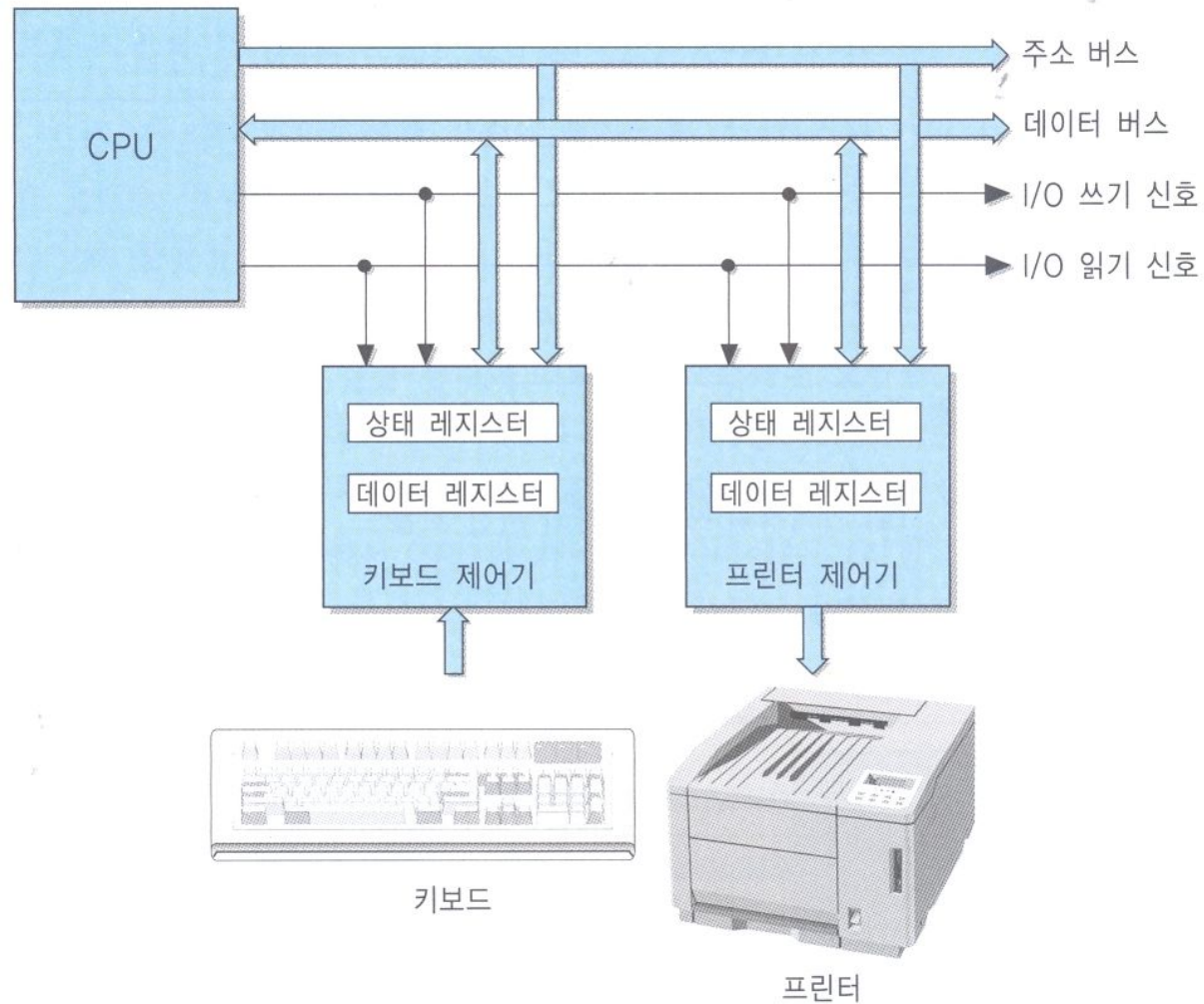
시스템의 구성 (5)

20

- CPU가 키보드로부터 데이터를 읽어들이는 과정은 다음과 같음.
 - * 키보드의 어느 한 키(key)가 눌러지면
그에 대응하는 8비트(ASCII코드 7비트+패리티 1비트) 데이터가
키보드 제어기로 전송되어 데이터 레지스터(data register)에 저장되며,
동시에 상태 레지스터(status register)의 지정된 비트가 세트됨.
 - * 이 비트는 입력 데이터가 준비되어있다는 것을 가리키므로 여기에서는 In_RDY비트
라고 부르기로 함.
 - * CPU는 데이터 입력을 원하는 시점에서 먼저 상태 레지스터를 읽어들이어서 In_RDY
비트가 1로 세트 되어 있는지를 검사하는데,
만약 세트 되어 있으면, 즉시 데이터 레지스터의 내용을 CPU로 읽어 들임으로써
데이터 입력 과정이 완료됨.
 - * 그러나 만약 In_RDY 비트가 세트 되어 있지 않으면, CPU는 상태 레지스터의
내용을 읽어와서 검사하는 동작을 반복하면서 기다림.

시스템의 구성 (6)

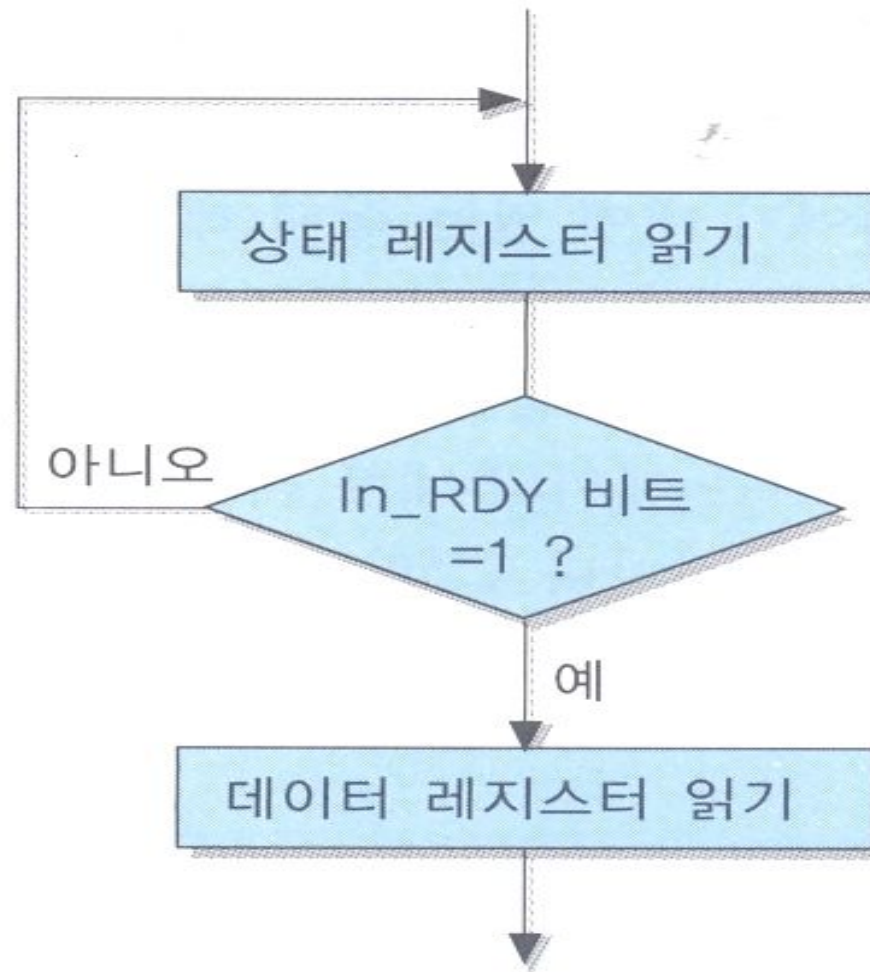
21



- CPU와 I/O 장치들 간의 접속

시스템의 구성 (7)

22



• 키보드 데이터 입력 과정 흐름도

시스템의 구성 (8)

23

- CPU가 데이터를 프린트 하는 과정도 유사함.
 - * CPU는 데이터를 프린터로 보내기 전에 먼저 프린터 제어기 내의 상태 레지스터를 읽어서, 데이터 출력 준비(Out_RDY) 비트를 검사함.
 - * 이것은 프린터가 이전에 받은 데이터에 대한 프린트 동작을 완료하고 다음 프린트 동작을 시작할 준비가 되었는지를 검사하는 절차임.
 - * Out_RDY 비트가 1로 세트 된 것이 확인되면,
CPU는 프린트할 데이터를 프린터 제어기의 데이터 레지스터로 쓰게 되고,
제어기는 프린터를 적절히 구동하여 그 내용을 프린트 함.
 - * 이 과정에서도 만약 Out_RDY 비트가 세트 되지 않았다면 CPU는 그 비트에 대한 검사를 반복하면서 기다리게 됨.
 - * 여기에서 레지스터는 CPU와 입출력 장치 사이에서
임시 저장 장소 역할을 하기 때문에 데이터 버퍼(data buffer)라고 부름.

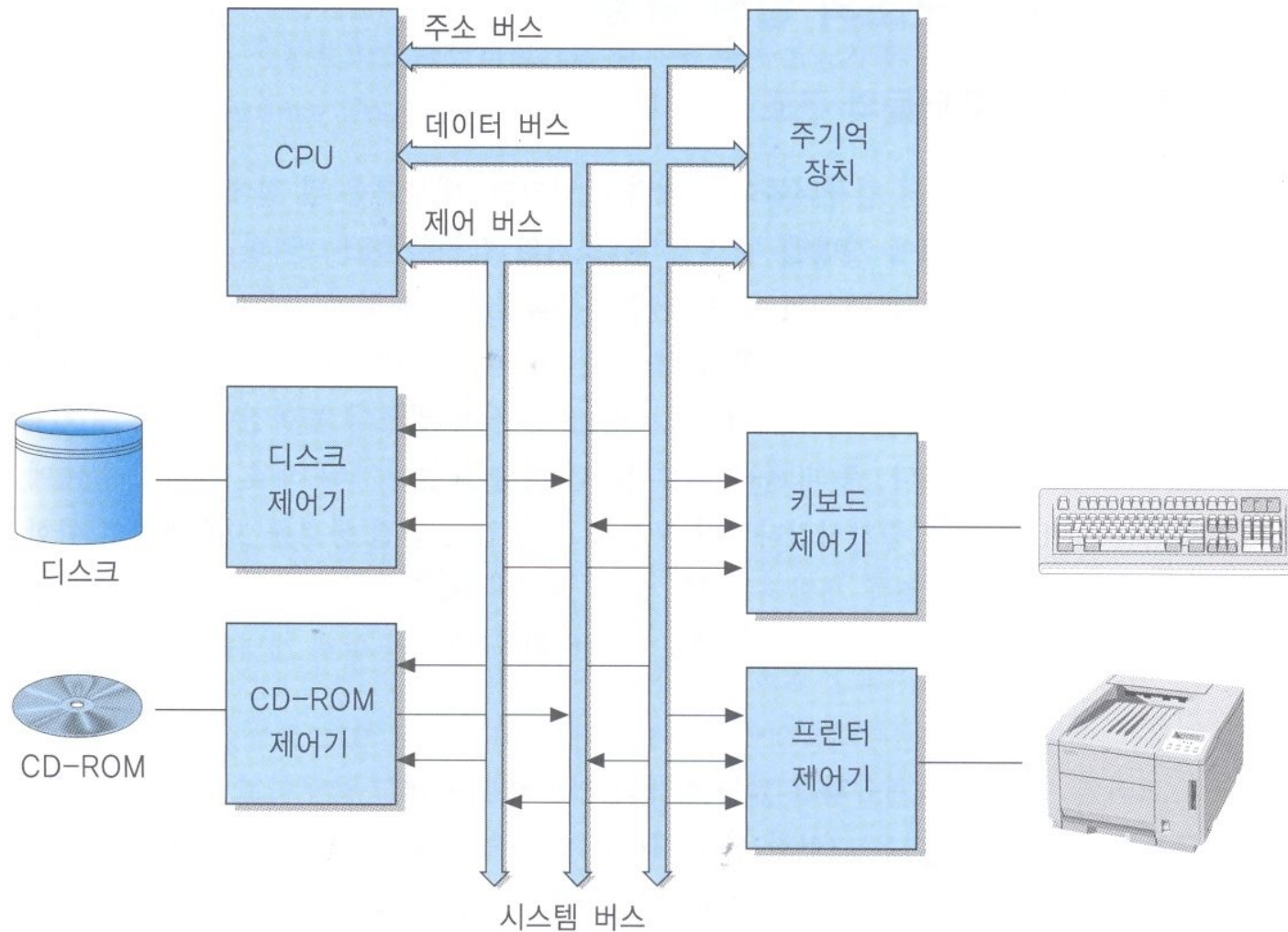
시스템의 구성 (9)

24

- 하드 디스크(hard disk), 플로피 디스켓(floppy diskette), CD-ROM 등과 같은 보조 기억장치들도 각 장치를 위한 제어기를 통해 이와 유사한 방법으로 접속됨.
- 그러나 키보드나 프린터의 경우에는 바이트 단위로 데이터 전송이 일어나지만, 보조저장장치의 경우에는 블록(512바이트, 1024바이트, 또는 4096바이트) 단위로 전송이 이루어지기 때문에
제어기 내에 적어도 한 블록 이상을 임시 저장할 수 있는 데이터 버퍼가 존재해야 함.
- CPU가 각 장치의 제어기를 구분하기 위해서 각 제어기 내에 있는 상태 레지스터와 데이터 레지스터를 각각 한 단어 길이의 기억장치로 간주하여
고유의 주소를 한 개씩 지정함으로써
기억장치와 동일한 방법으로 I/O장치의 읽기와 쓰기 동작을 수행할 수 있음.

시스템의 구성 (10)

25



• 컴퓨터시스템의 전체 구성도

셀프 테스트

26

- CPU가 실행할 프로그램 코드들과 데이터가 저장되는 곳은 무슨 장치인가?

- * 기억장치

해설) 기억장치에는 크게 두 종류, 즉 프로그램 코드와 데이터가 저장됨.

- 컴퓨터의 시스템 버스 중에서 단방향 버스는 어느 것인가?

- * 주소 버스

해설) 주소버스는 CPU가 기억장치의 위치를 지정하는 데에 사용되므로 CPU로부터 메모리와 I/O제어기로 연결되는 단방향 버스임.

- 키보드 제어기와 프린터 제어기가 데이터 읽기와 쓰기 준비가 되어있음을 나타내는 레지스터는 무엇인가?

- * 상태 레지스터

해설) 제어기에 데이터가 저장되어 있음을 알리기 위해서 필요한 레지스터가 상태 레지스터임.
상태 레지스터의 값이 변한 것을 감지하여 CPU는 데이터의 존재유무를 판단하는 것임.

요점 정리

27

- 컴퓨터는 프로그램 코드들을 정해진 순서대로 실행하는데, 그 과정에서 (1) 필요한 데이터를 읽어서(read), (2) 처리(processing)하고, (3) 저장(store)함.
- 컴퓨터가 받아들이고 처리하는 정보의 종류로는 프로그램 코드(program code)와 데이터(data)가 있음.
- 연산 코드(op code)인 001은 ‘레지스터 A로 적재하라’라는 연산을 지정해주는 비트들이며, 오퍼랜드(operand)인 00101은 적재될 데이터가 저장되어 있는 기억장치 주소를 나타냄.
- 시스템 버스(system bus)는 주소 버스(address bus), 데이터 버스(data bus), 제어 버스(control bus) 등으로 구성됨.
- CPU가 키보드 제어기와 프린터 제어기에 데이터를 읽기와 쓰기를 시작하려 할 때에 우선적으로 점검해야 할 레지스터가 바로 상태 레지스터임.
- CPU가 각 장치의 제어기를 구분하기 위해서 각 제어기 내에 있는 상태 레지스터와 데이터 레지스터를 각각 한 단어 길이의 기억장치로 간주하여 고유의 주소를 한 개씩 지정함.