

3주차 1차시

컴퓨터 구조(1)

컴퓨터 시스템의 구성과 동작 원리_컴퓨터 시스템에 대한 기본적 이해

1 컴퓨터 시스템의 구성

2 중앙처리장치



컴퓨터 구조(1)

◆ 학습목표

- 컴퓨터 시스템의 구성에 대해 알아본다.
- 중앙처리장치의 구성과 명령어 처리 과정을 알아본다.

1. 컴퓨터 시스템의 구성

- ◆ **하드웨어** : 컴퓨터를 구성하는 기계적 장치
- ◆ **소프트웨어** : 하드웨어의 동작을 지시하고 제어하는 명령어의 집합



1. 컴퓨터 시스템의 구성

1) 하드웨어

◆ 하드웨어의 구성

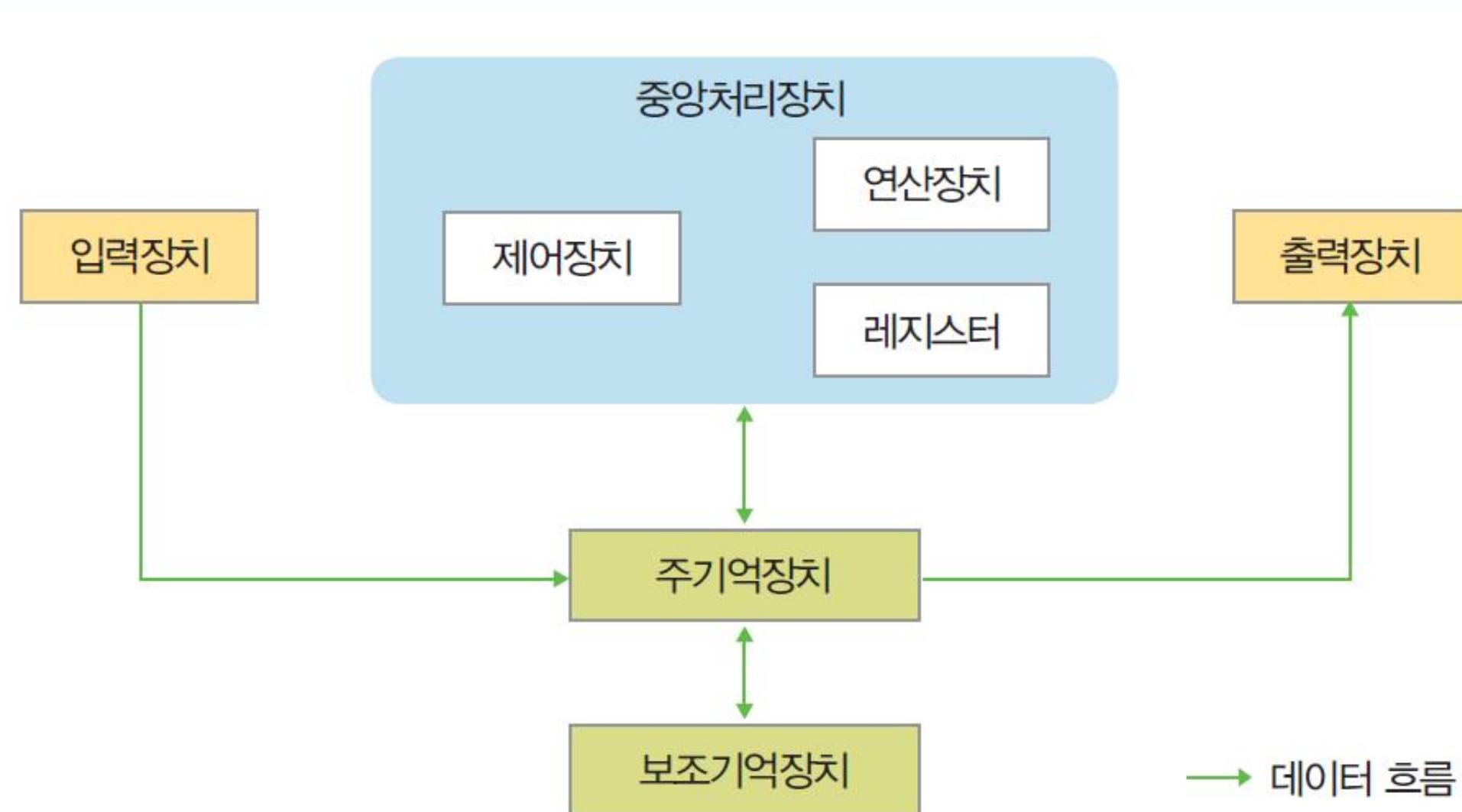


그림 3-2 하드웨어의 구성

1. 컴퓨터 시스템의 구성

1) 하드웨어

◆ 하드웨어의 구성

- 중앙처리장치
 - 주기억장치로부터 프로그램 명령어와 데이터를 읽어온 뒤 처리
 - 산술논리연산장치, 제어장치, 레지스터로 구성
- 기억장치
 - 프로그램과 데이터, 연산의 중간 결과 등을 저장
 - 주기억장치와 보조기억장치로 구분
 - 주기억장치는 프로그램과 데이터를 일시적으로 저장
 - 보조기억장치는 데이터를 영구적으로 보관
- 입출력장치
 - 입력장치는 각종 자료를 컴퓨터 내부로 입력하는 장치
 - 출력장치는 컴퓨터에서 처리한 결과를 외부로 표현하는 장치

1. 컴퓨터 시스템의 구성

2) 시스템 버스

◆ 시스템 버스

- 데이터 버스 : 중앙처리장치와 기타 장치(기억장치, 입출력장치 등) 사이에서 데이터를 전달하는 통로
- 주소 버스 : 중앙처리장치가 주기억장치나 입출력장치로 기억장치 주소를 전달하는 통로
- 제어 버스 : 중앙처리장치가 기억장치나 입출력장치에 제어 신호를 전달하는 통로



2. 중앙처리장치

1) 중앙처리장치의 구성

◆ 중앙처리장치의 구성

- 연산장치 : 덧셈, 뺄셈, 곱셈, 나눗셈 등의 산술연산과 논리곱, 논리합, 부정 등의 논리연산을 수행
- 제어장치 : 명령어를 순서대로 실행할 수 있도록 제어하는 장치
- 레지스터 : 중앙처리장치의 속도와 비슷한 고속의 기억장치로 명령어 주소, 명령어 코드, 연산에 필요한 데이터, 연산 결과 등을 임시로 저장함

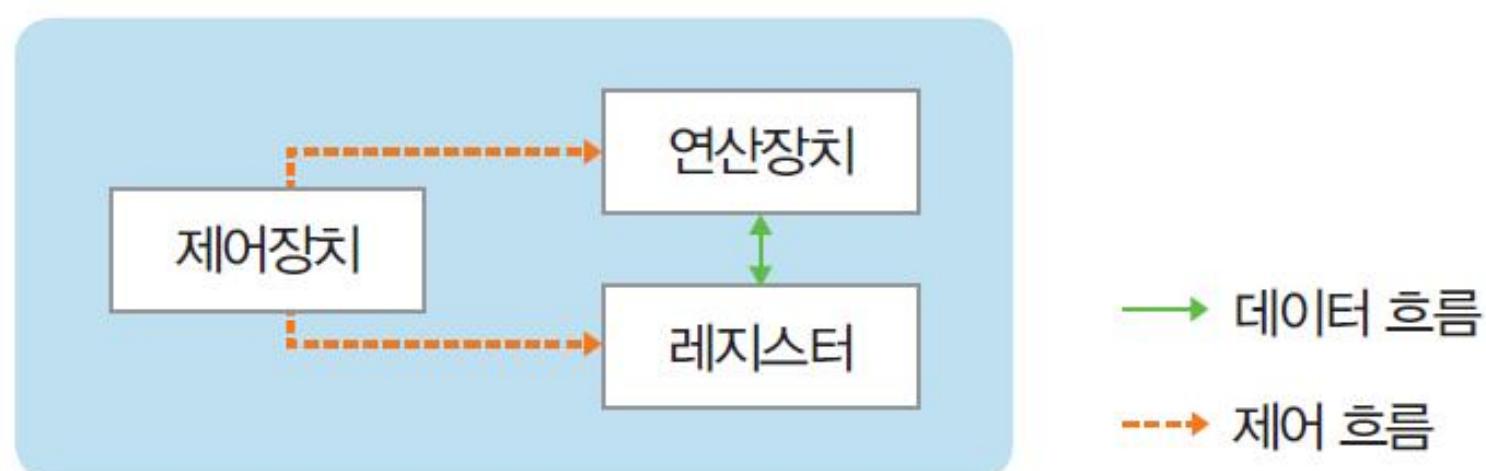


그림 3-4 중앙처리장치의 구성

2. 중앙처리장치

1) 중앙처리장치의 구성

표 3-1 특수 목적 레지스터의 용도와 기능

용도	명칭	기능
주소 저장	메모리 주소 레지스터 (MAR, Memory Address Register)	읽기와 쓰기 연산을 수행할 주기억장치의 주소를 저장한다.
	프로그램 카운터 (PC, Program Counter)	다음에 수행할 명령어의 주소를 저장한다.
	스택 포인터(SP, Stack Pointer)	스택의 최상위 주소를 저장한다.
	인덱스 레지스터(IX, IndeX register)	인덱스 주소 지정 방식에서 인덱스를 저장한다.
명령어 저장	명령어 레지스터 (IR, Instruction Register)	현재 실행 중인 명령어를 저장한다.
데이터 저장	메모리 버퍼 레지스터 (MBR, Memory Buffer Register)	주기억장치에서 읽어온 데이터나 주기억장치에 저장할 데이터를 임시로 저장한다.
	누산기(AC, ACCumulator)	연산 결과를 임시로 저장한다.
CPU 상태 저장	프로그램 상태 레지스터 (PSR, Program Status Register)	CPU의 현재 상태 정보를 저장한다.

2. 중앙처리장치

2) 중앙처리장치의 동작

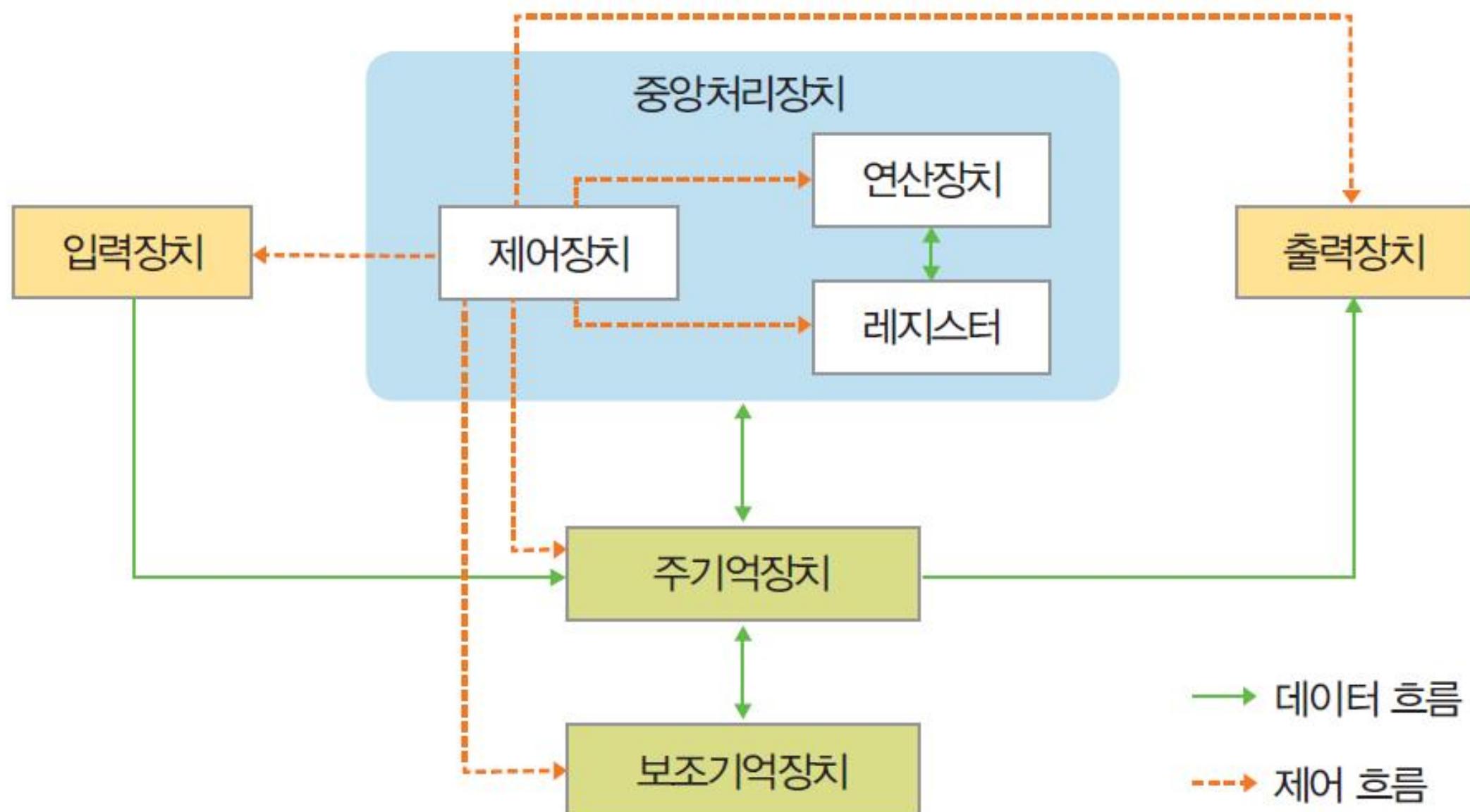


그림 3-5 중앙처리장치의 동작

2. 중앙처리장치

3) 명령어 세트

◆ 명령어 형식

연산 코드

피연산자

그림 3-6 명령어의 기본 형식

- 연산 코드 : 실행하는 연산의 종류에 따라 연산 기능, 제어 기능, 데이터 전달 기능, 입출력 기능으로 나뉨
- 피연산자 : 주소, 숫자, 문자, 논리 데이터 등을 저장

2. 중앙처리장치

3) 명령어 세트

◆ 피연산자 수에 따른 명령어 분류

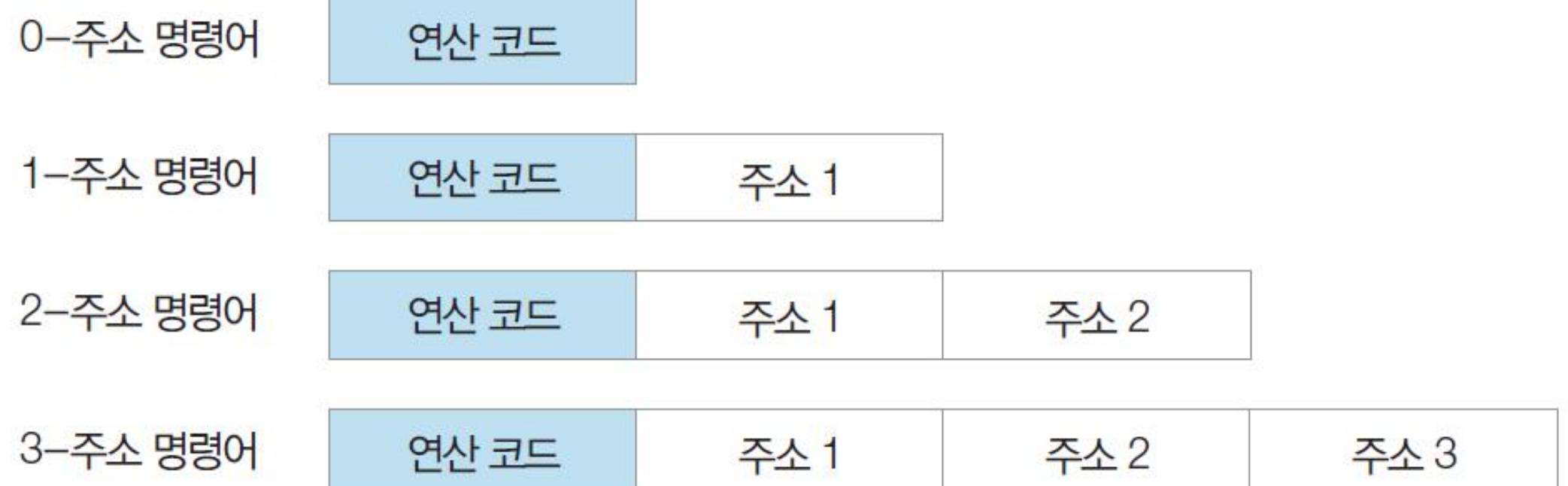


그림 3-7 피연산자 수에 따른 명령어 분류

2. 중앙처리장치

3) 명령어 세트

◆ 주소 필드 수에 따른 명령어 분류

- 0-주소 명령어 : 연산 코드만 존재하며 스택 구조 컴퓨터에서 사용

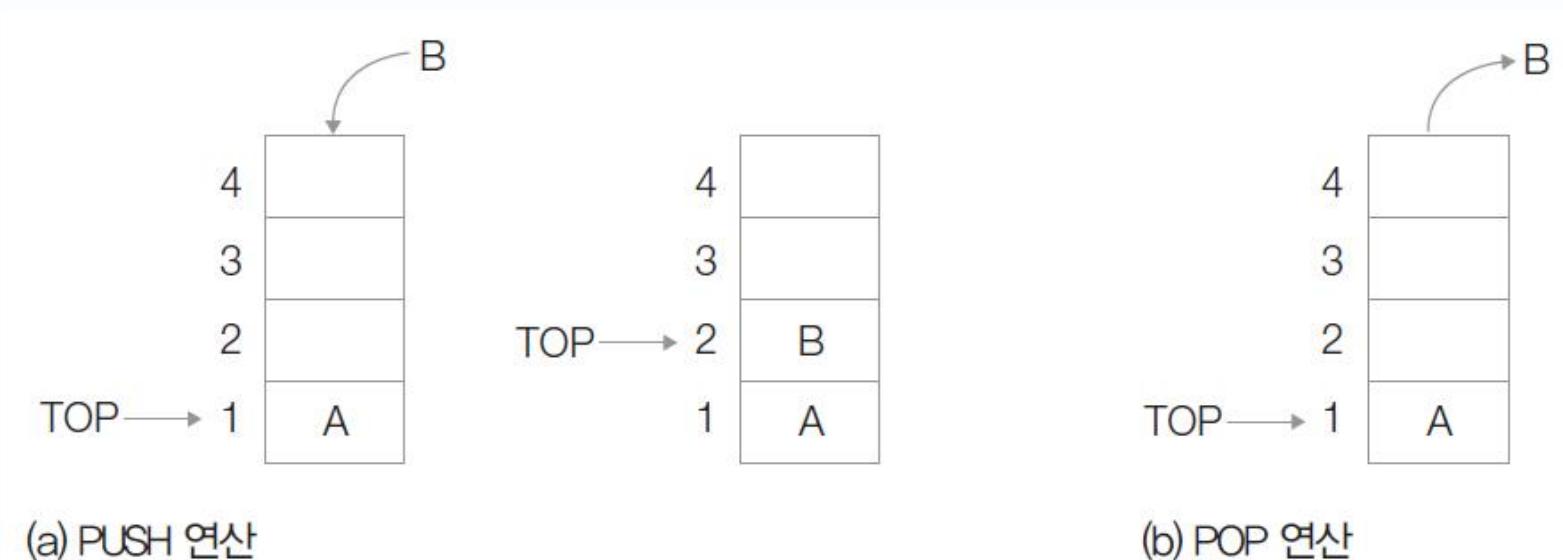


그림 3-8 PUSH 연산과 POP 연산

X=(A+B)의 처리 과정

PUSH A // 스택의 TOP \leftarrow A

PUSH B // 스택의 TOP \leftarrow B

ADD // 스택의 TOP \leftarrow (A+B)

* 스택에서 A와 B를 POP하고 더한 후 결과를 PUSH

STORE X // X \leftarrow 스택의 TOP

2. 중앙처리장치

3) 명령어 세트

◆ 주소 필드 수에 따른 명령어 분류

- 1-주소 명령어
 - 단일 누산기 구조 컴퓨터에서 사용
 - 누산기 레지스터를 이용하여 데이터 연산을 수행
 - 연산에는 주기억장치에서 읽힌 데이터와 누산기에 저장된 데이터가 사용
 - 연산결과는 다시 누산기에 저장

X=(A+B)의 처리 과정

```
LOAD A // AC ← A
```

```
ADD B // AC ← AC + B
```

* AC에 저장된 A와 주기억장치에서 읽어온 B를 더한 후 결과를 AC에 저장

```
STORE X // X ← AC
```

2. 중앙처리장치

3) 명령어 세트

◆ 주소 필드 수에 따른 명령어 분류

- 2-주소 명령어 : 2개의 주소필드를 가지며 가장 일반적인 형태

X=(A+B)의 처리 과정

```
MOV R1, A    // R1 ← A  
ADD R1, B    // R1 ← R1 + B  
* 레지스터 R1과 주기억장치에서 읽어온 B를 더한 후 결과를 R1에 저장  
MOV X, R1    // X ← R1
```

- 3-주소 명령어 : 3개의 주소 필드를 가지며, 주소 필드에 레지스터 번호나 주기억장치의 주소를 지정

X=(A+B)의 처리 과정

```
ADD X, A, B    // X ← A + B  
* 주기억장치에서 읽어온 A와 B를 더한 후 결과를 주기억장치 X에 저장
```

2. 중앙처리장치

4) 명령어 처리 과정

◆ 명령어 사이클

- 중앙처리장치가 주기억장치로부터 한 번에 하나의 명령어를 인출하여 실행하는데 필요한 일련의 활동

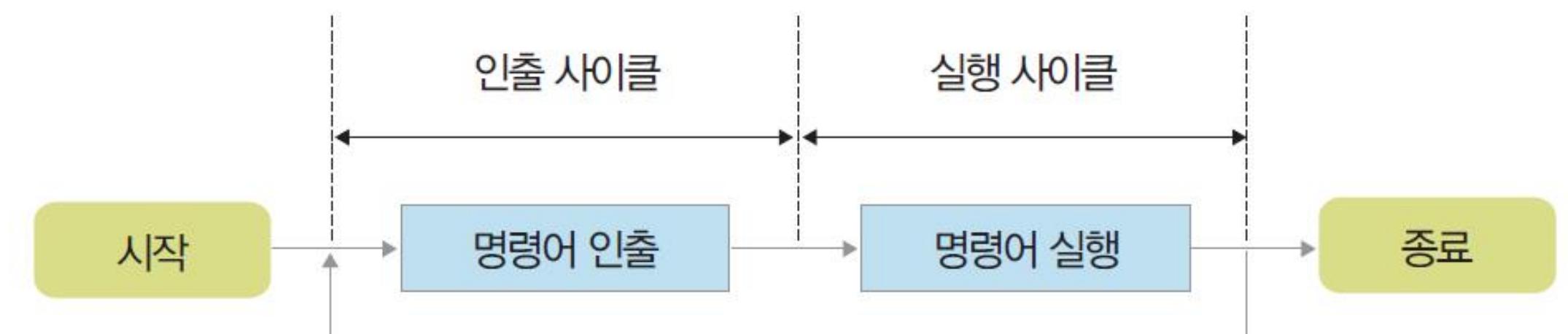


그림 3-9 명령어 사이클

2. 중앙처리장치

4) 명령어 처리 과정

- ◆ **인출 사이클과 실행 사이클은 항상 수행**
- ◆ **간접 사이클과 인터럽트 사이클은 주소지정 방식과 인터럽트 요구에 따라 필요할 때만 수행**

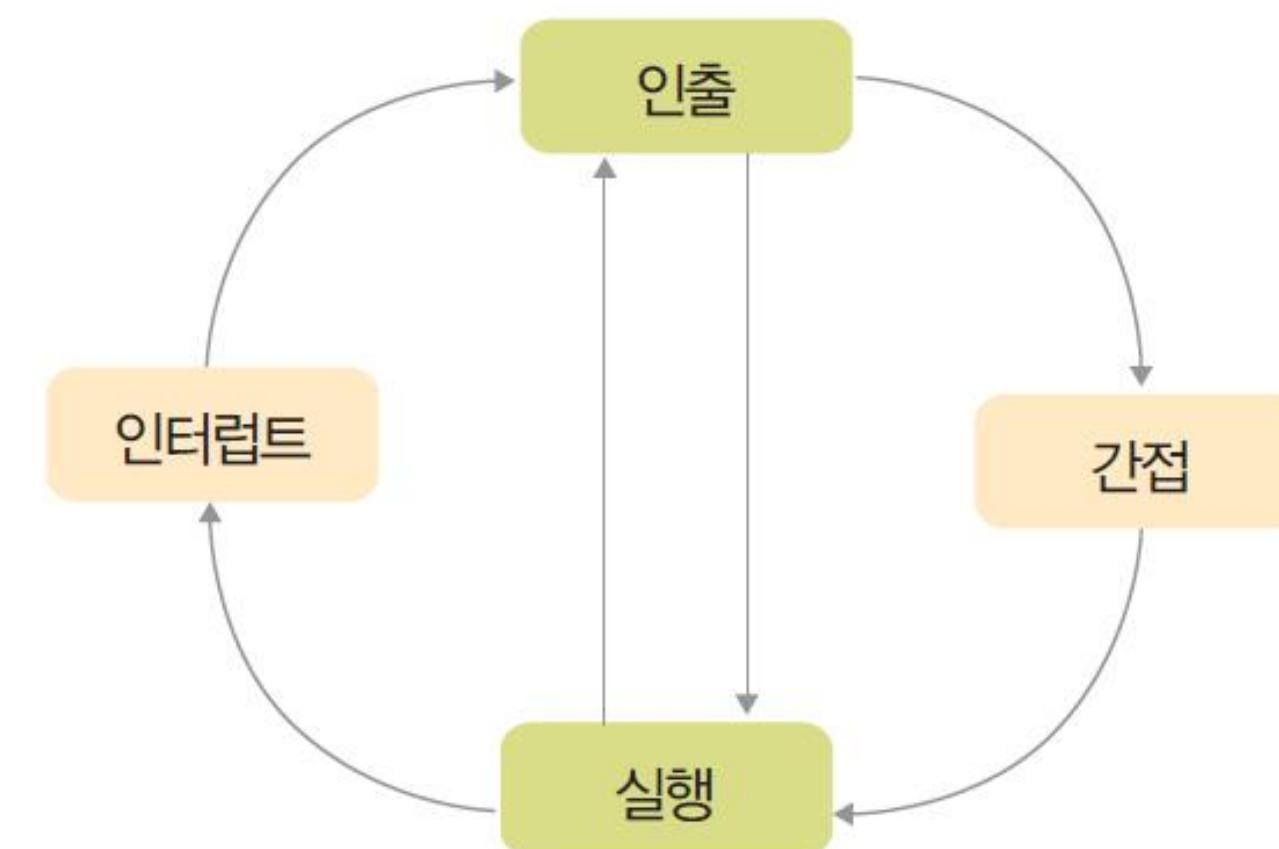


그림 3-10 세분화된 명령어 사이클

2. 중앙처리장치

4) 명령어 처리 과정

◆ 인출 사이클

- 주기억장치에서 명령어를 인출
- 다음 명령어를 인출하기 위해 PC(프로그램 카운터)값을 증가시킴

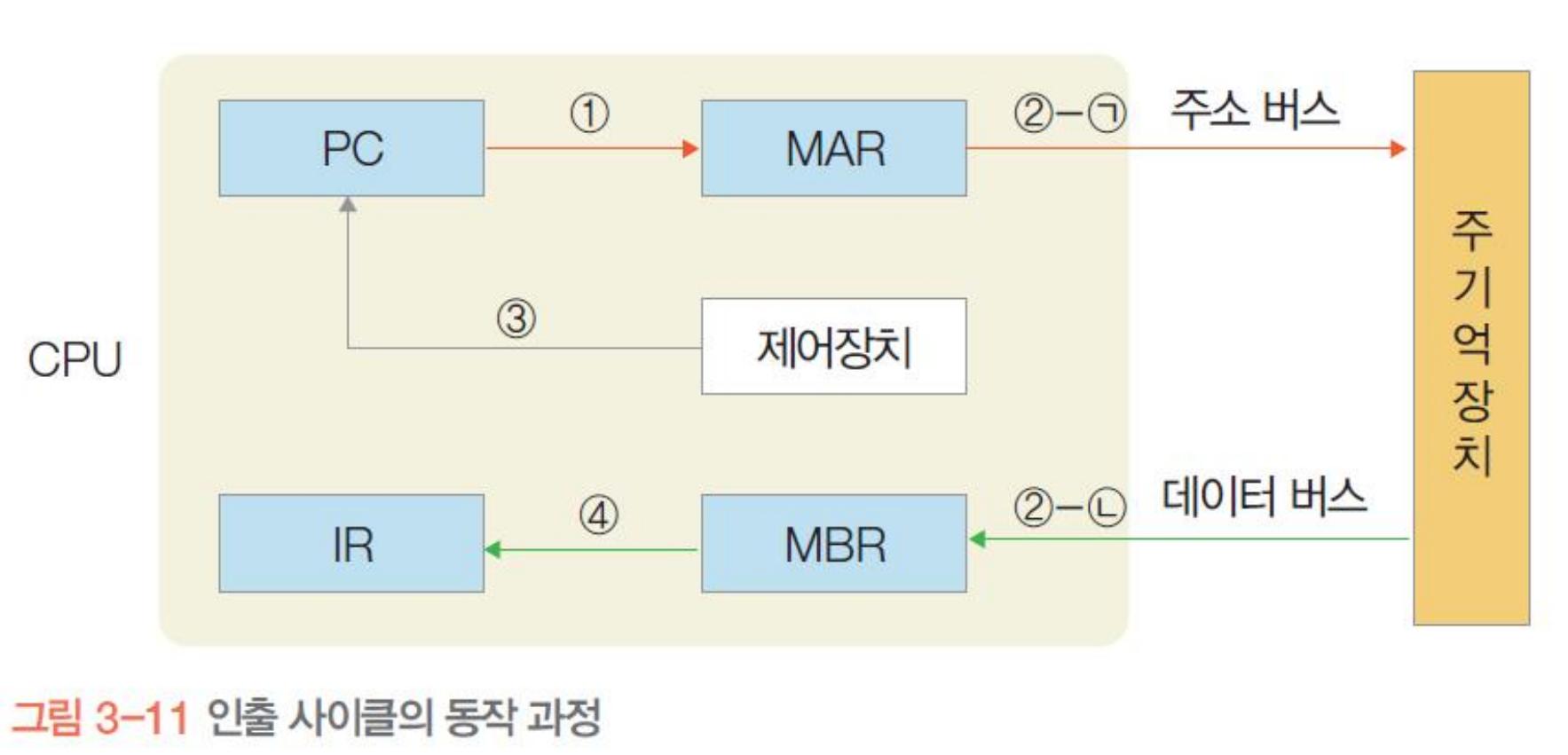


그림 3-11 인출 사이클의 동작 과정

2. 중앙처리장치

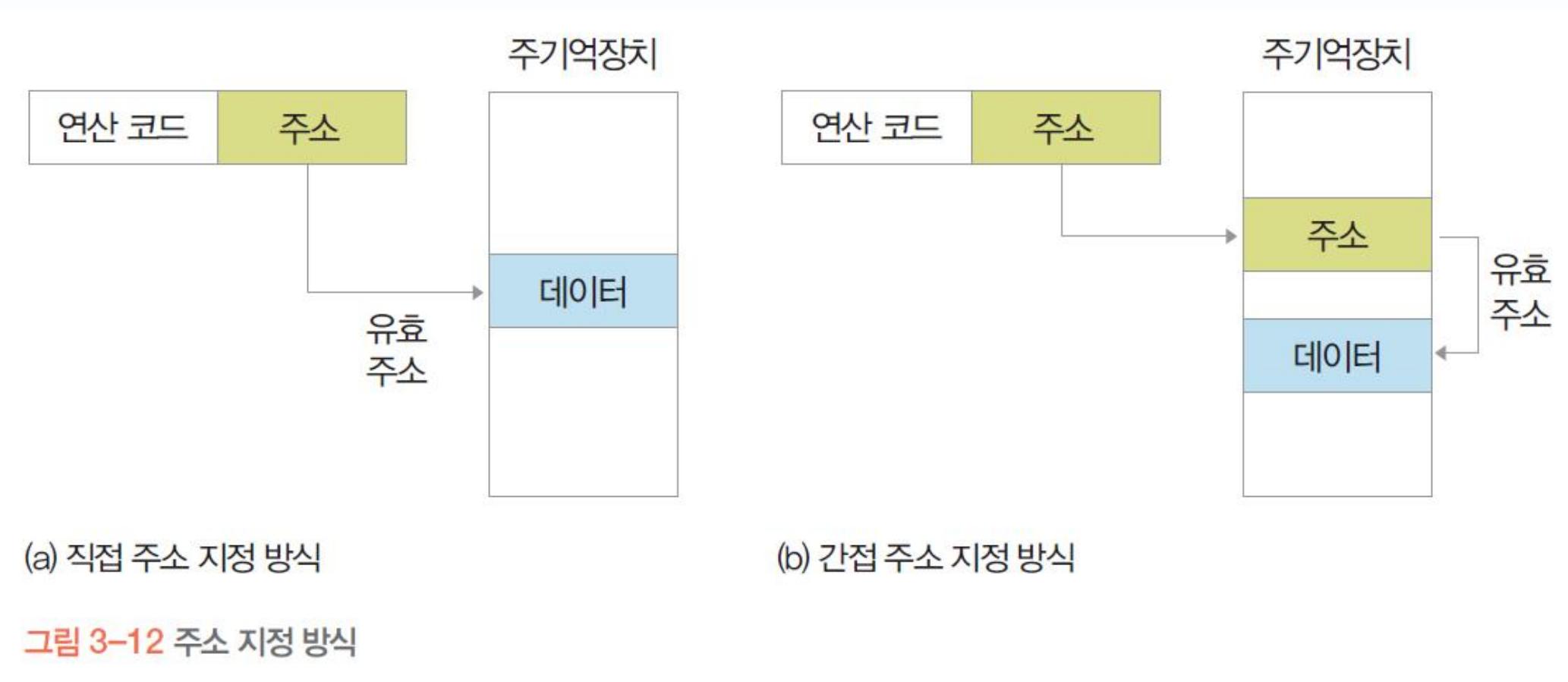
4) 명령어 처리 과정

◆ 실행 사이클

- 인출한 명령어를 해독하고 그 결과에 따라 제어 신호를 발생시켜 명령어를 실행

◆ 간접 사이클

- 직접 주소 지정 방식과 간접 주소 지정 방식으로 나뉨



2. 중앙처리장치

4) 명령어 처리 과정

◆ 인터럽트 사이클

- 인터럽트 : 중앙처리장치가 프로그램을 수행하는 동안 컴퓨터 시스템의 내부와 외부에서 발생하는 예기치 못한 사건

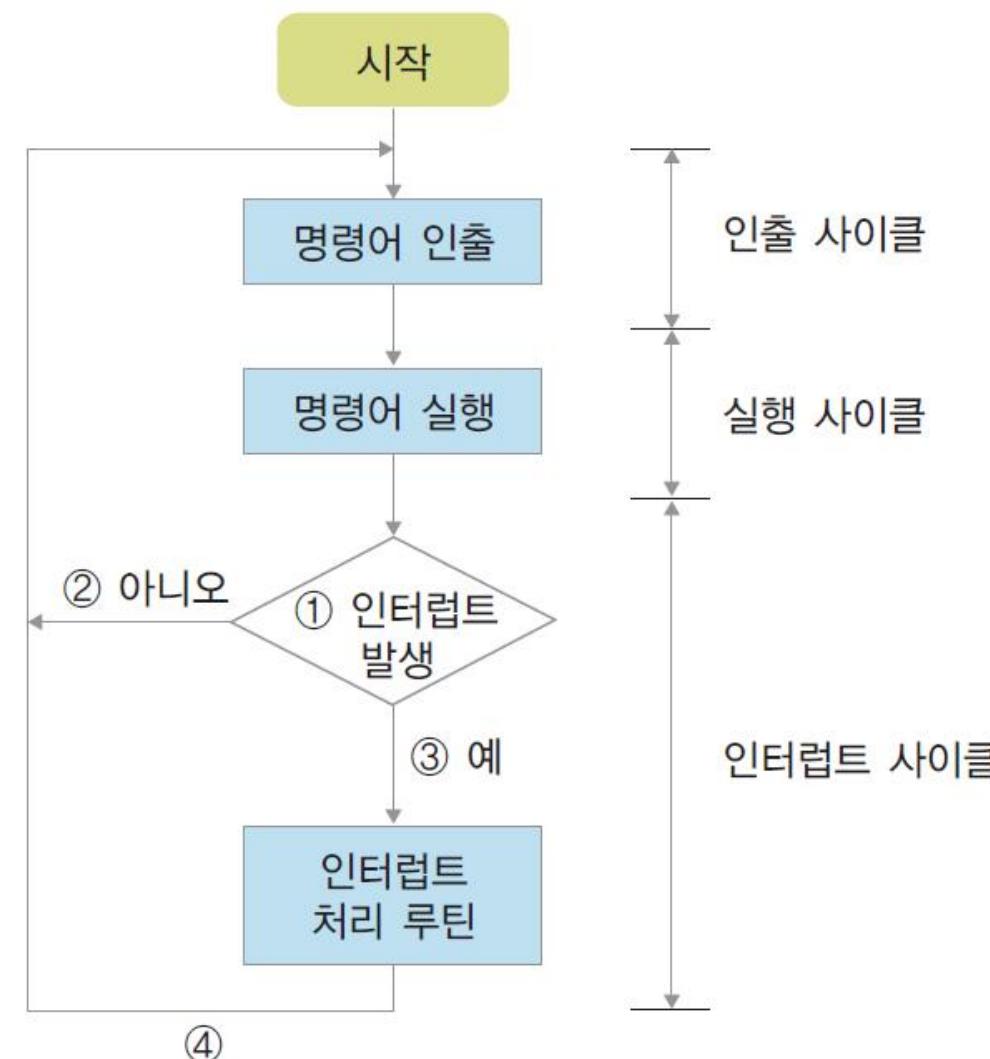


그림 3-13 인터럽트 사이클이 포함된 명령어 사이클



요약

◆ 하드웨어의 구성

- 중앙처리장치: 주기억장치로부터 프로그램 명령어와 데이터를 읽어와 처리하고 명령어의 수행순서를 제어한다.
- 기억장치: 프로그램, 데이터, 연산의 중간결과 등을 저장한다.
- 입출력장치: 문자, 숫자, 소리, 그림 등의 자료를 컴퓨터 내부로 입력하거나 컴퓨터 내부에서 처리한 결과를 인간이 인지 할 수 있는 형태로 변환하여 컴퓨터 외부로 출력한다.

◆ 시스템 버스

- 하드웨어 구성요소를 물리적으로 연결하여 구성 요소 사이의 데이터 이동 통로를 제공한다. 용도에 따라 주소버스, 데이터 버스, 제어버스로 구분된다.

◆ 중앙처리장치

- 주기억장치로부터 프로그램 명령어와 데이터를 읽어와 처리하고 명령어의 수행 순서를 제어한다.