

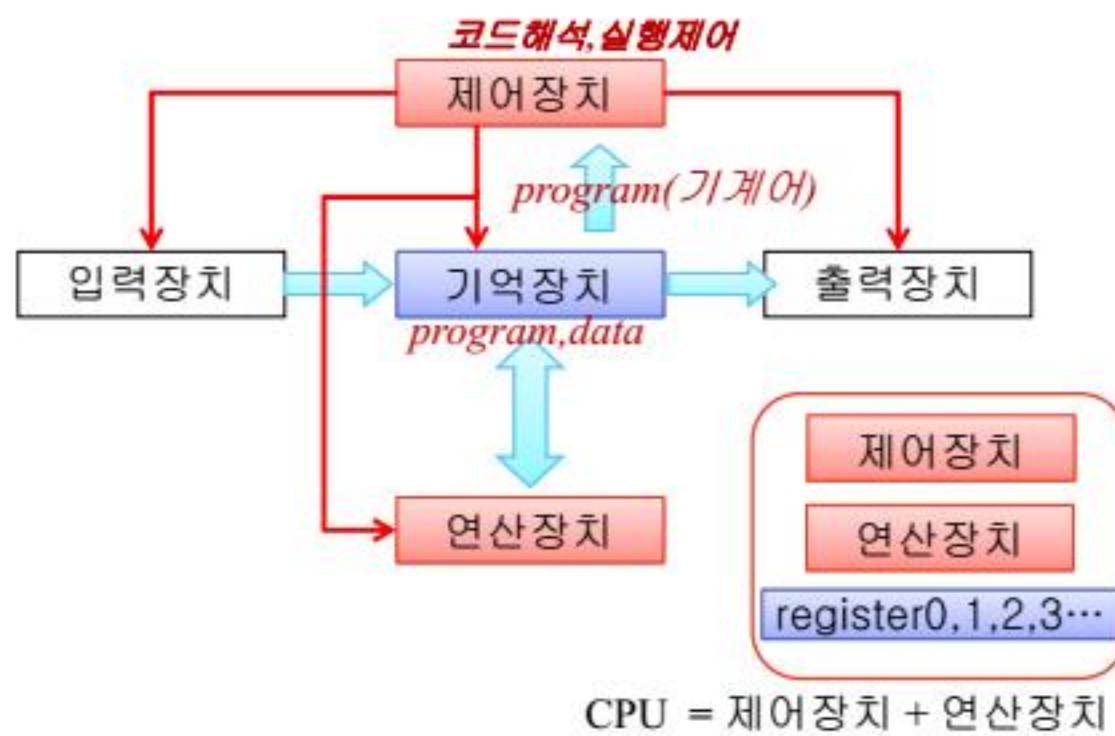
UWIR

# 컴퓨터 HW 관련 OVERVIEW



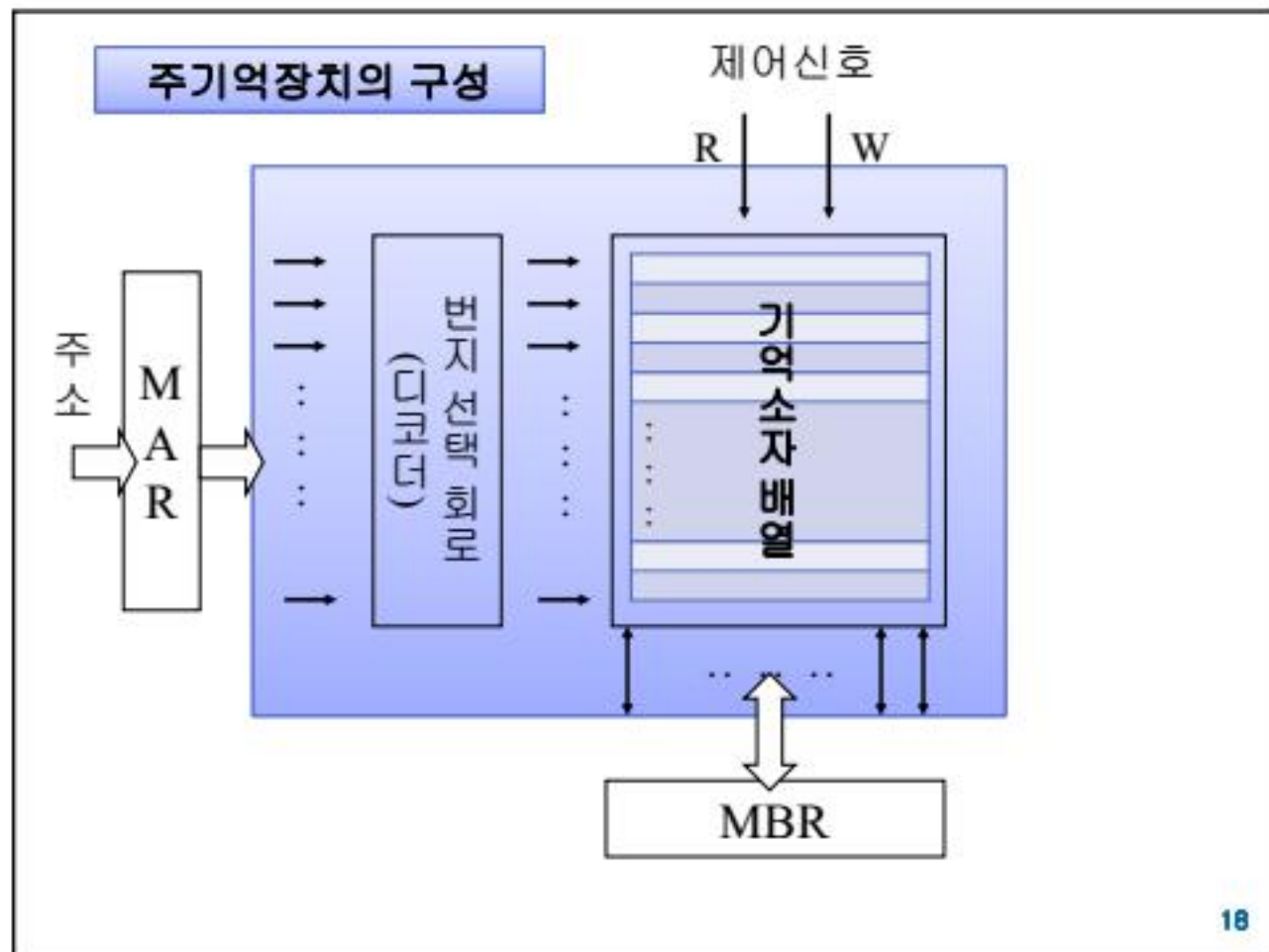
14

## 컴퓨터의 데이터 흐름/ 제어 흐름



17

17



18

## 설명

- MAR : memory address register

기억 소자 배열의  $n$ 번지에 데이터를 R/W할 경우, 먼저  $n$ (주소)을 MAR에 저장해야 한다.

- MBR : memory buffer register

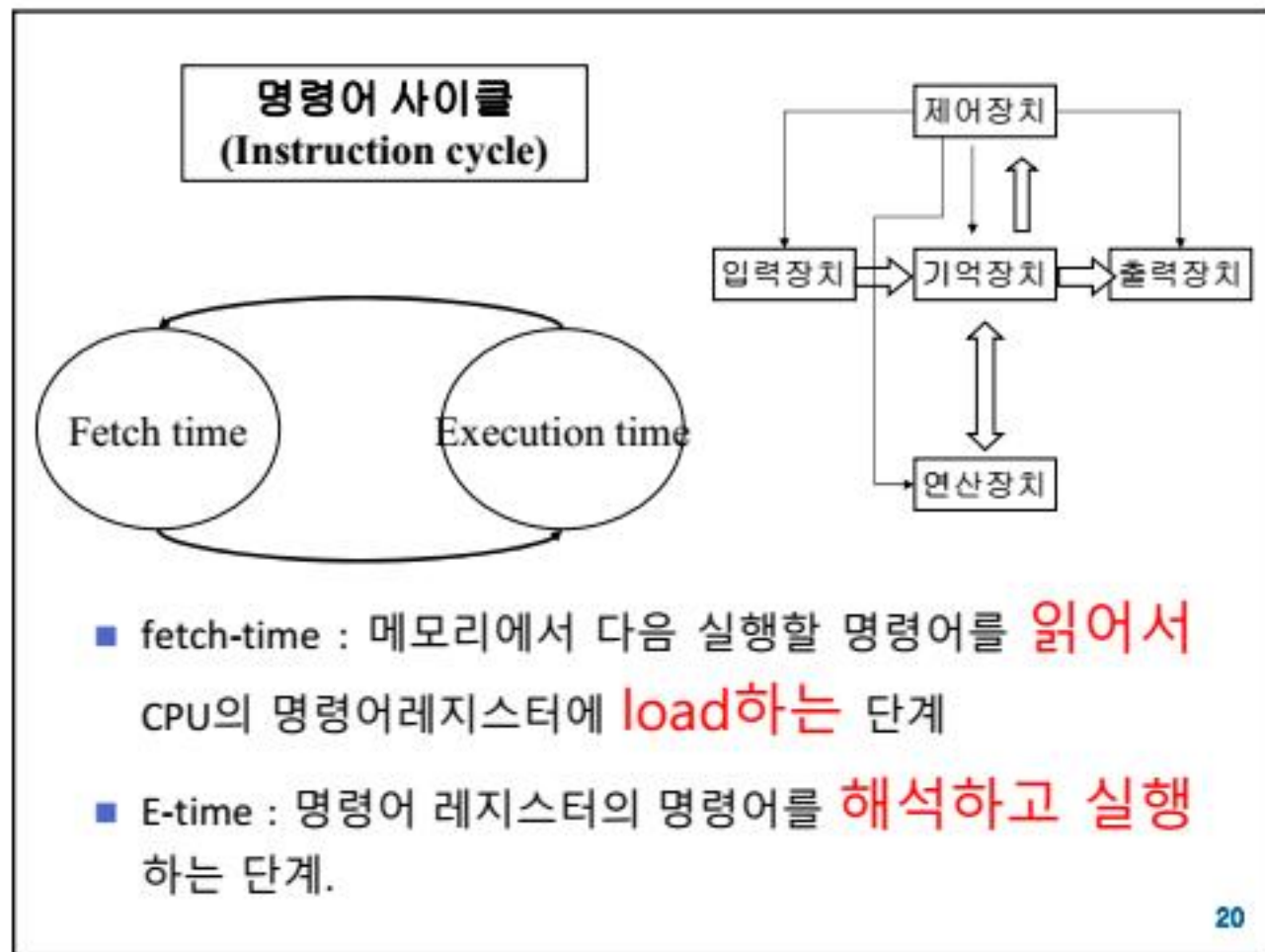
- $n$  번지에 **W**할 데이터를 임시 저장
- $n$  번지에 R한 후 데이터를 임시 저장하는 레지스터.

- R/W제어 : 제어장치에서 발생하는 제어 신호

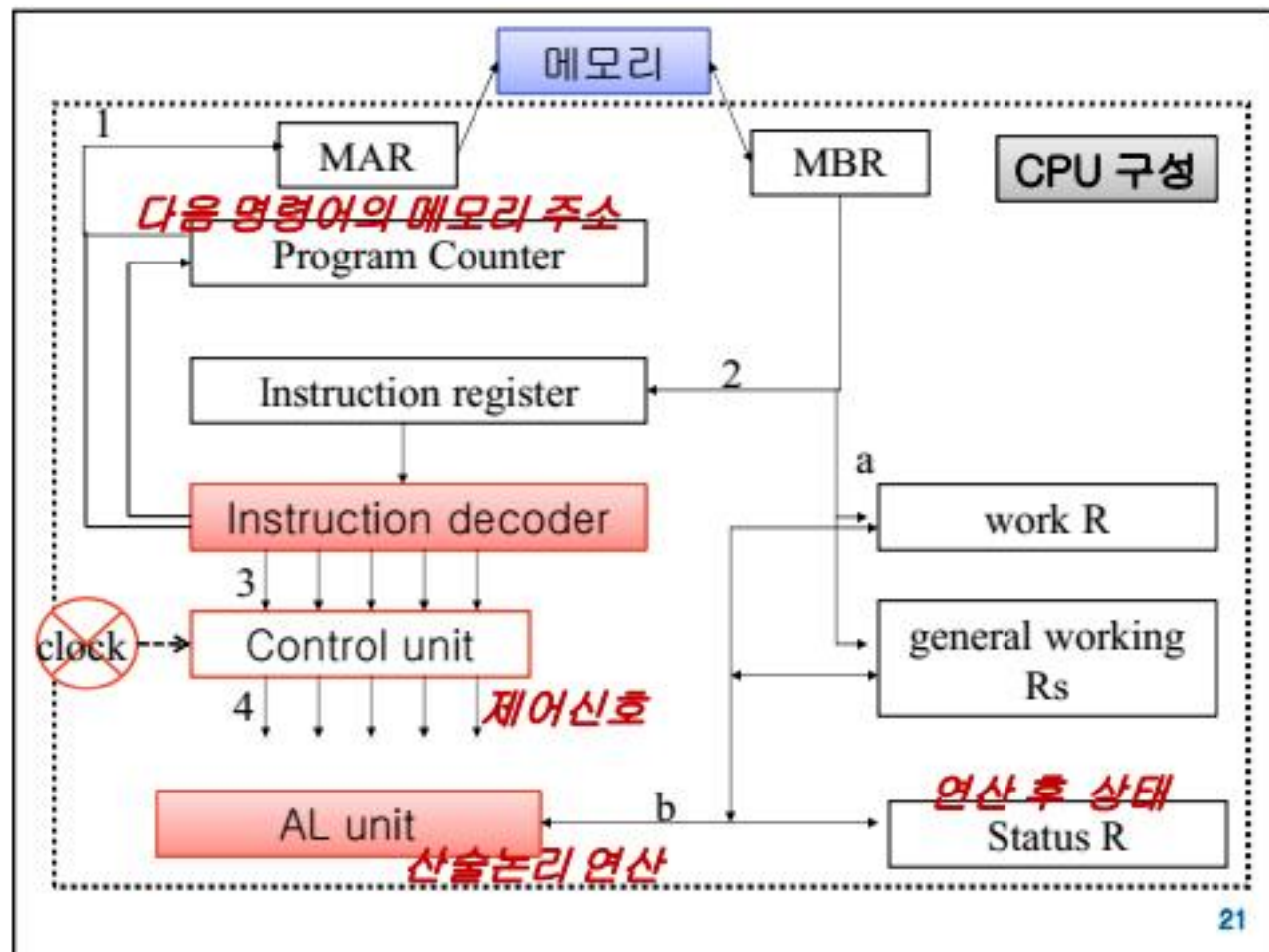
Register : CPU에 있는 소규모 데이터 기억장치 (64, 32, 128 bit 등)

19

19



20



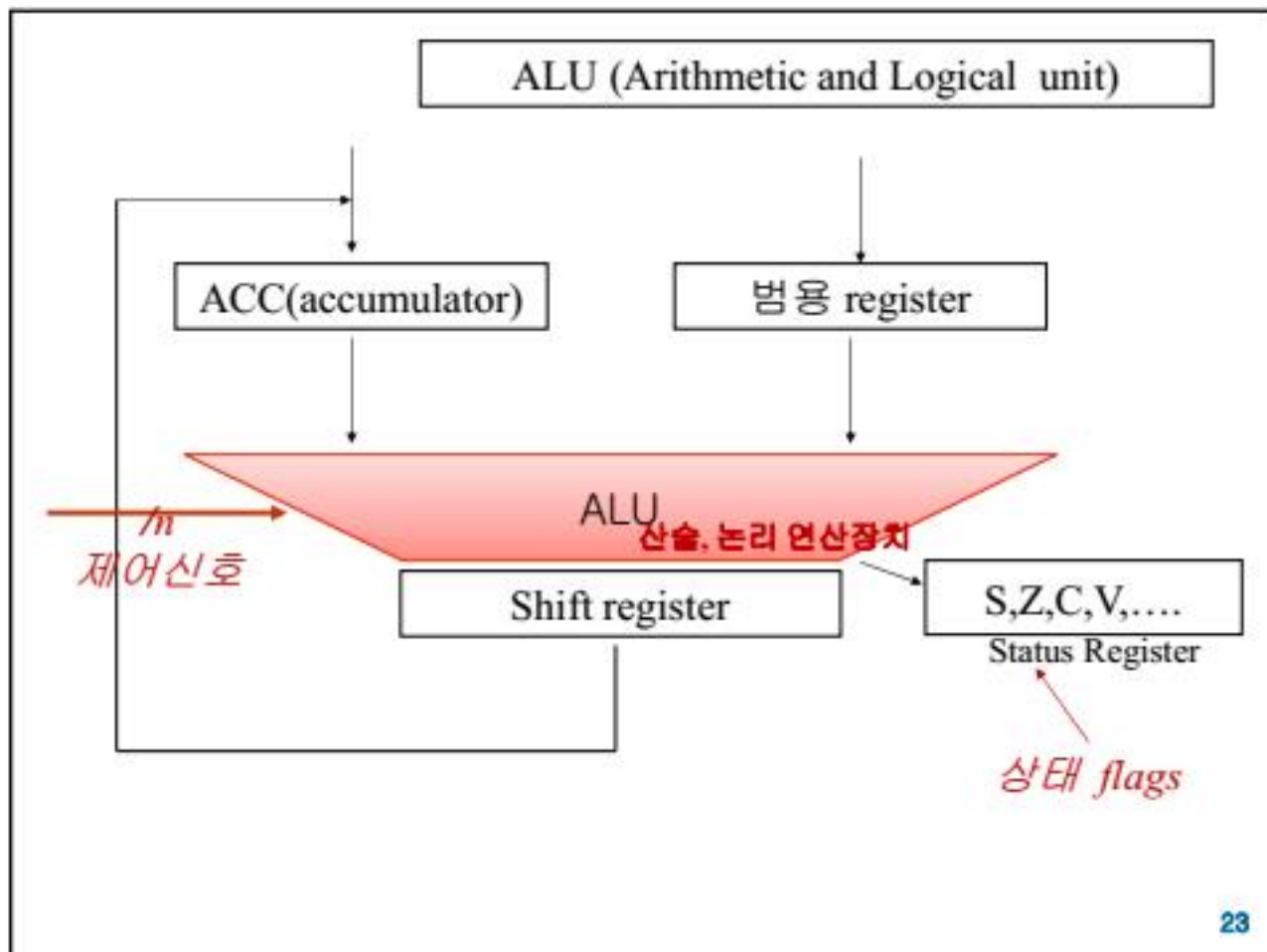
21

## 시스템

프로세서: Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz (3.60 GHz)  
설치된 메모리(RAM): 4.00GB(3.88GB 사용 가능)  
시스템 종류: 64비트 운영 체제, x64 기반 프로세서  
펜 및 터치: 이 디스플레이에 사용할 수 있는 펜 또는 터치식 입력이 없습니다.

22

22



23

23



## 설명

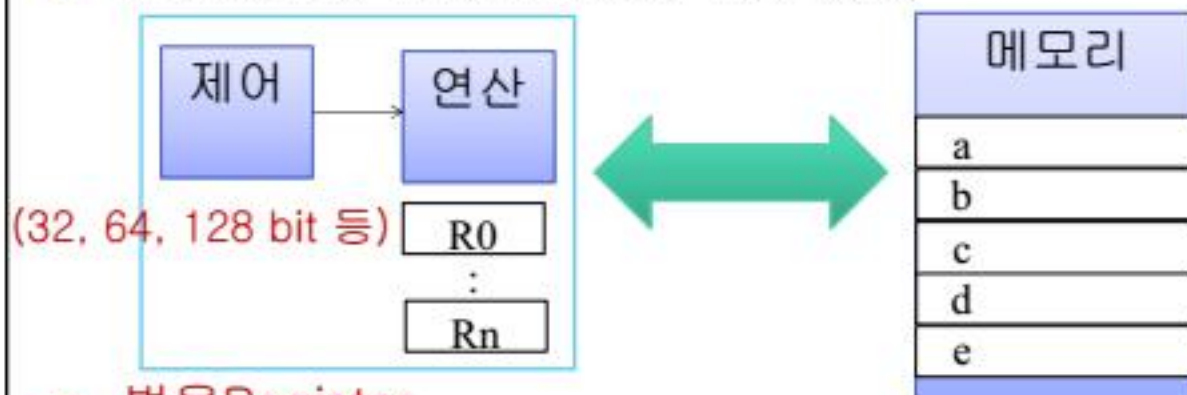
- program counter(PC): 다음 실행할 명령어의 메모리주소를 보관
- instruction register(IR) : PC가 지정한 메모리번지에서 읽어온 명령어를 해석하기전에 임시 보관 (**다음 실행할 명령어**)
- instruction decoder : 명령어 해독기(해석), instruction set에 대한 해독회로가 내장 ( 기계어 명령어 → 해석 → 순차적인 제어 신호set )
- control unit : clock에 맞춰 해독된 제어신호를 전송
- status register/flag 상태 레지스터 : 명령실행(연산,비교)후 결과의 상태를 저장하는 레지스터
  - 각 bit마다 carry,overflow,zero,sign 등의 결과를 표현한다.
- working register 작업 레지스터 : 명령의 실행에 쓰이는 레지스터 ( programmer가 지정하여 사용할 수 없다)
- general purpose register 범용 레지스터 : 여러 목적에 사용될 수 있는 레지스터. 레지스터번호(0,1,2..)를 두어 사용한다.
- clock : MHz, GHz

24

24

## 레지스터

- 고속 처리를 위해 n bits를 임시 저장



- 범용Register
- MAR, MBR
- 명령어Register
- Accumulator
- program counter
- 상태 Register
- Base Register
- index Register

25

25

## register

$$2^{32} = 4,294,967,296 \text{ (4GB)}$$

$$2^{64} = 18,446,744,073,709,551,616 \text{ (16 ExaByte)}$$

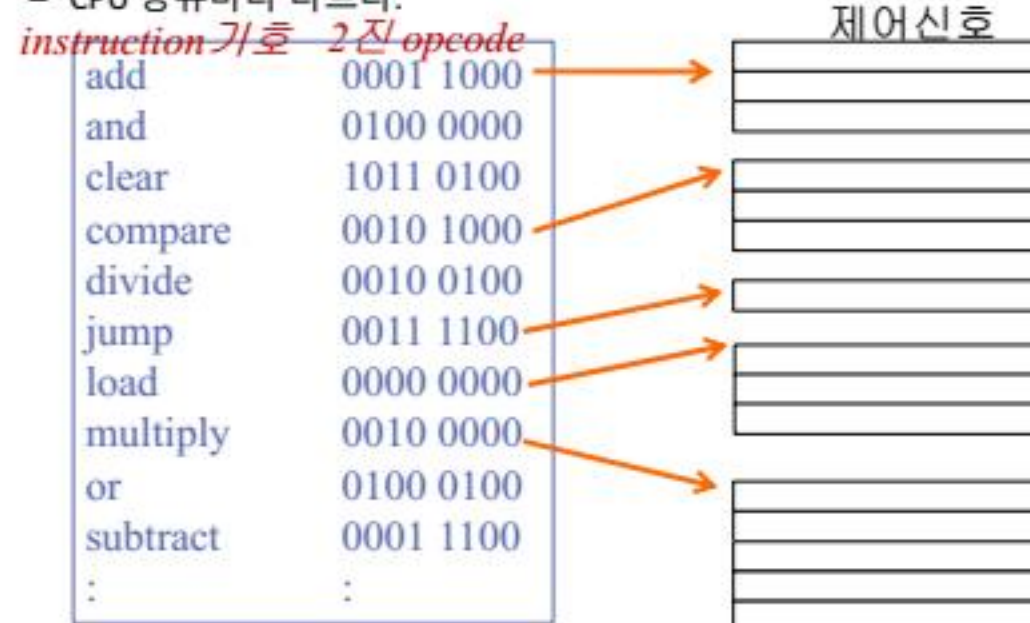
- 32bit-computer와 64bit-computer란?
  - CPU가 한 번에 처리할 수 있는 데이터의 크기(bits)
  - 32bit 단위의 처리능력 vs 64bit 단위의 처리능력
  - 32bit register                      64bit register
- MAR 의 크기 64bit
  - 실제 64bit 모두 사용하지 않는다.
  - OS에 따라 최대 지원 메모리 용량이 다르기 때문이다.
    - 예) Window 11 Education 2TB
    - Window 11 Enterprise 6TB

26

26

## Instruction set

- 한 컴퓨터가 사용할 수 있는 instruction code들의 집합
  - CPU 종류마다 다르다.



27

27

## CPU의 처리(fetch time)

- 메모리 → CPU 로의 데이터 흐름은 두 가지.

### ① 명령어

PC → MAR로 번지 전달

RAM의 해당 번지로 부터 데이터 → IR

### ② 데이터

IR → instruction Decoder

IR의 번지 부분 → MAR

RAM의 해당 번지로 부터 데이터

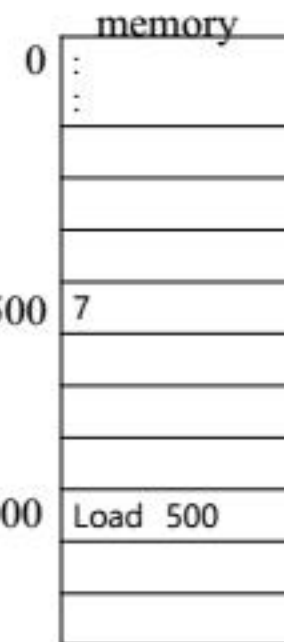
→ 작업 R 또는 범용 R

PC 1000

MAR

IR

general R



28

28

## CPU의 처리(fetch time)

- 메모리 → CPU 로의 데이터 흐름은 두가지.

### ① 명령어

PC → MAR로 번지 전달

RAM의 해당 번지로 부터 데이터 → IR

### ② 데이터

IR → instruction Decoder

IR의 번지 부분 → MAR

RAM의 해당 번지로 부터 데이터

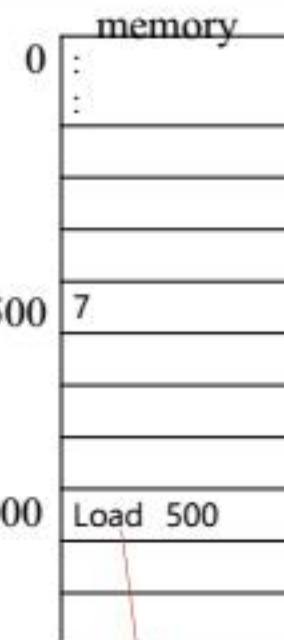
→ 작업 R 또는 범용 R

PC 1000

MAR 1000

IR

general R<sub>5</sub>



29

29



## CPU의 처리(execution time)

- 메모리 → CPU 로의 데이터 흐름은 두가지.

### ① 명령어

PC → MAR로 번지 전달

RAM의 해당 번지로 부터 데이터 → IR

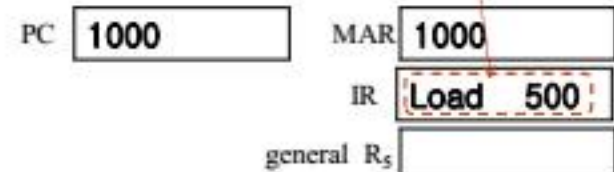
### ② 데이터

IR → instruction Decoder

IR의 번지 부분 → MAR

RAM의 해당 번지로 부터 데이터

→ 작업 R 또는 범용 R



30

30

## CPU의 처리(execution time)

- 메모리 → CPU 로의 데이터 흐름은 두가지.

### ① 명령어

PC → MAR로 번지 전달

RAM의 해당 번지로 부터 데이터 → IR

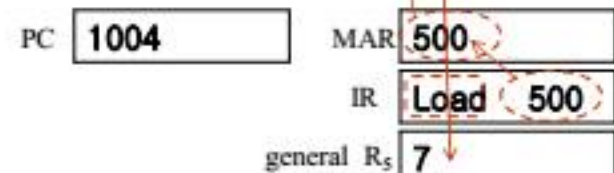
### ② 데이터

IR → instruction Decoder

IR의 번지 부분 → MAR

RAM의 해당 번지로 부터 데이터

→ 작업 R 또는 범용 R



31

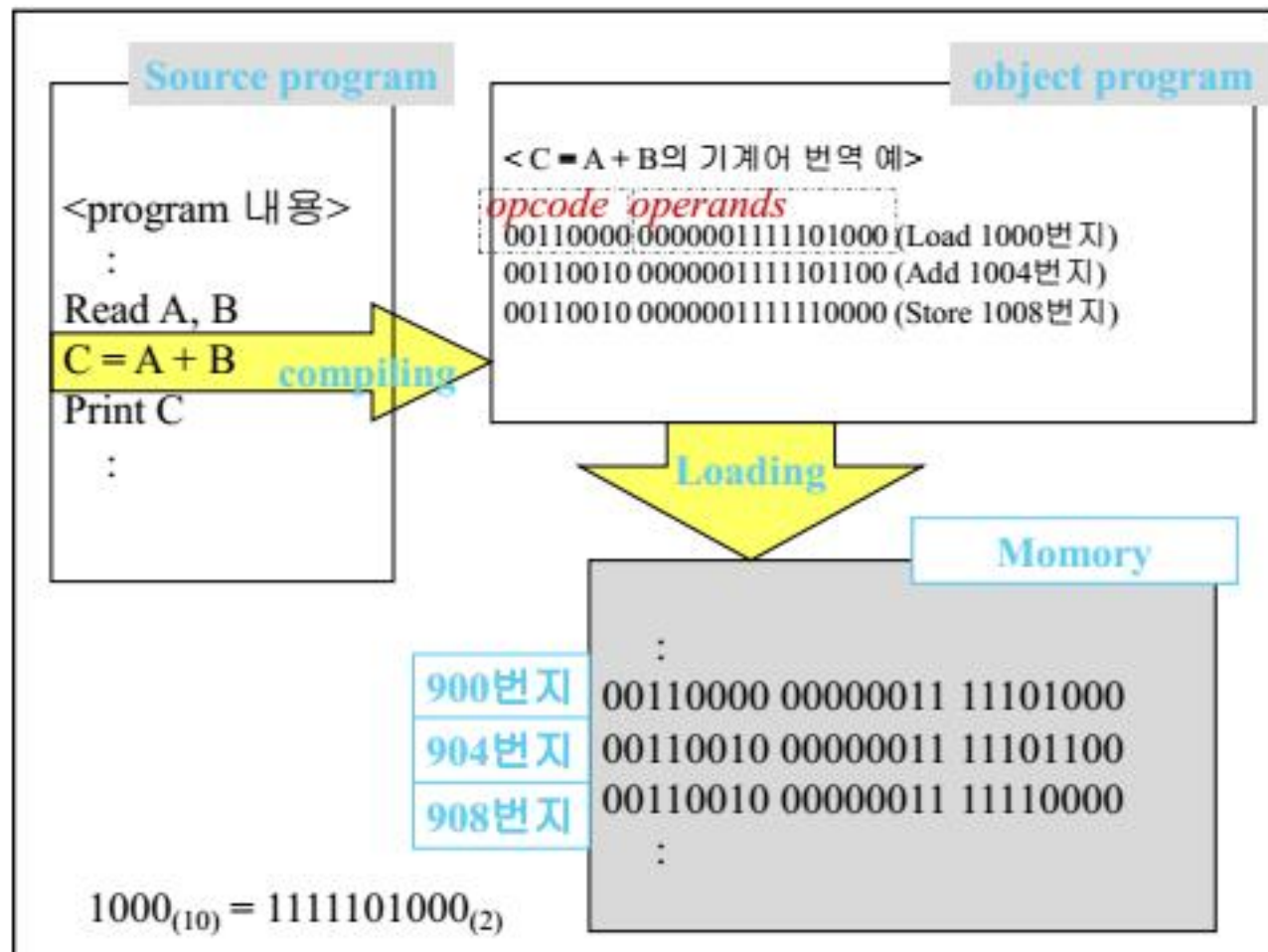
31



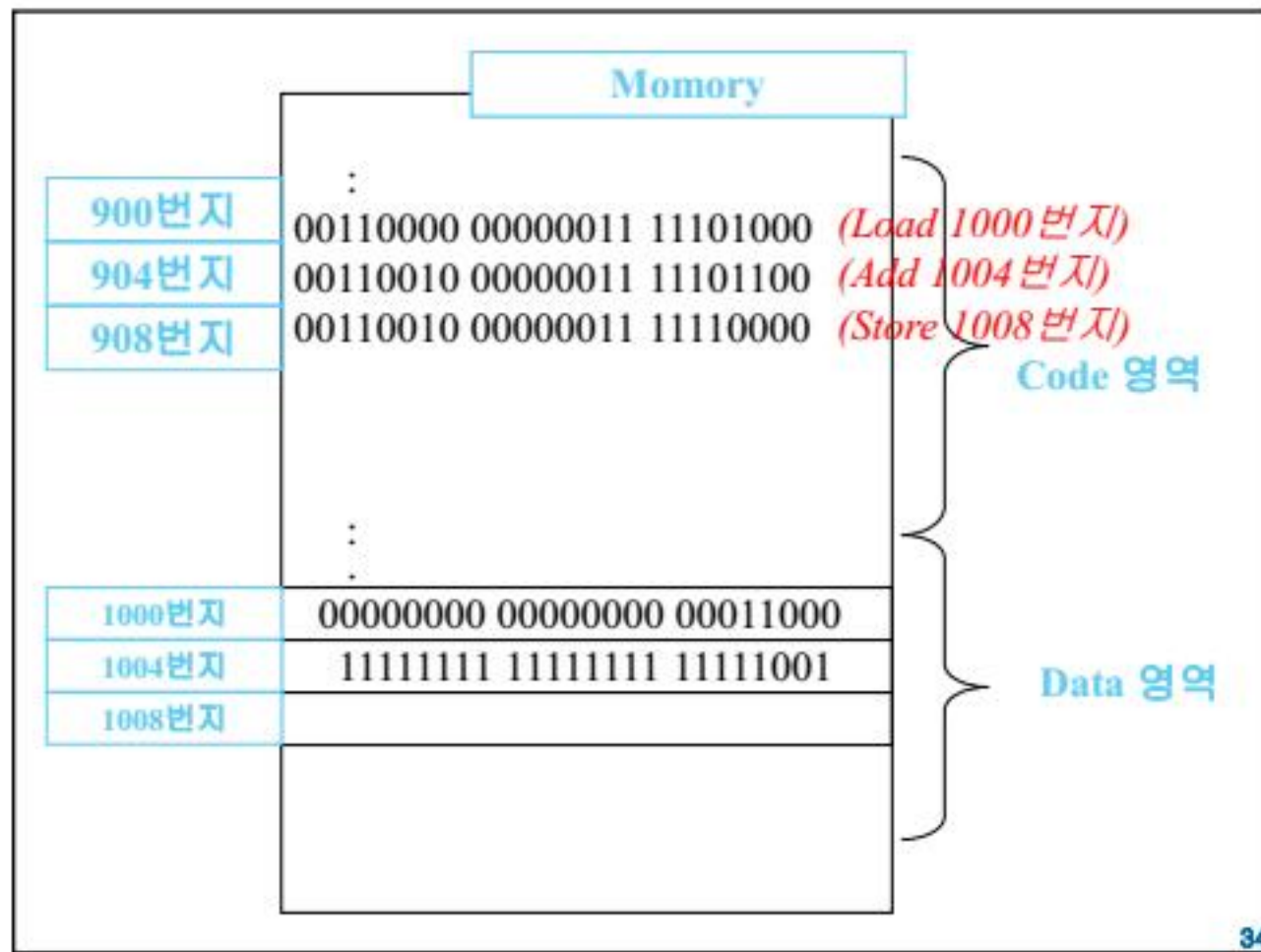
그림으로 이해

32

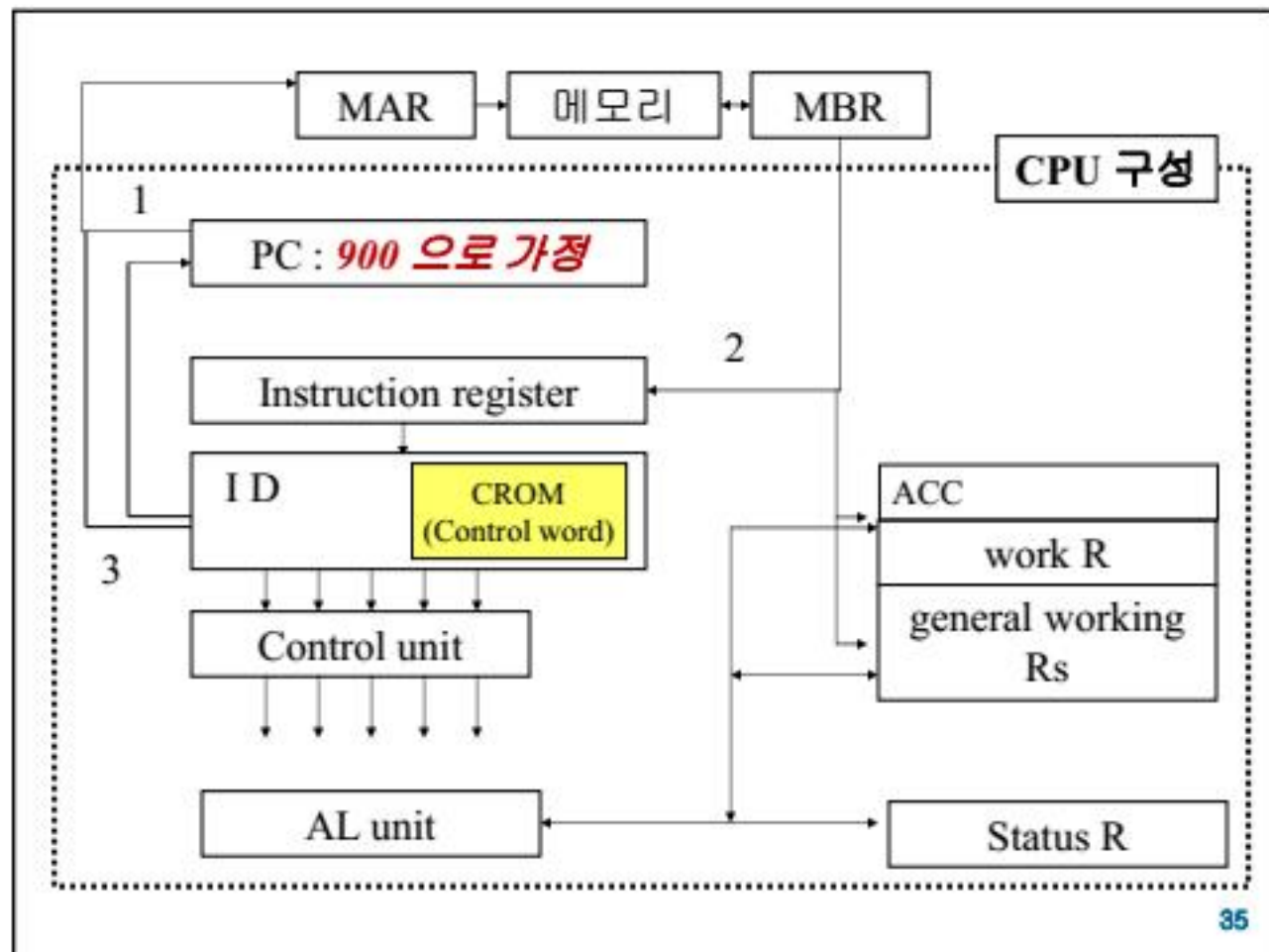
32



33



34



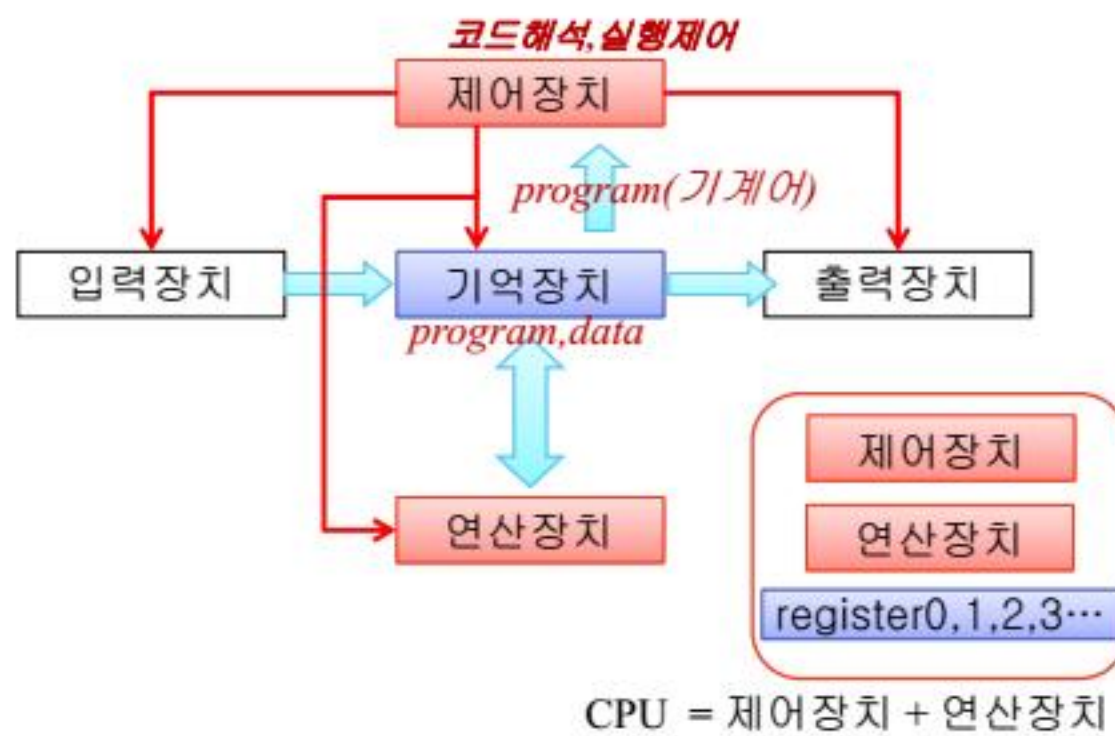
35

# 컴퓨터 HW 관련 OVERVIEW



14

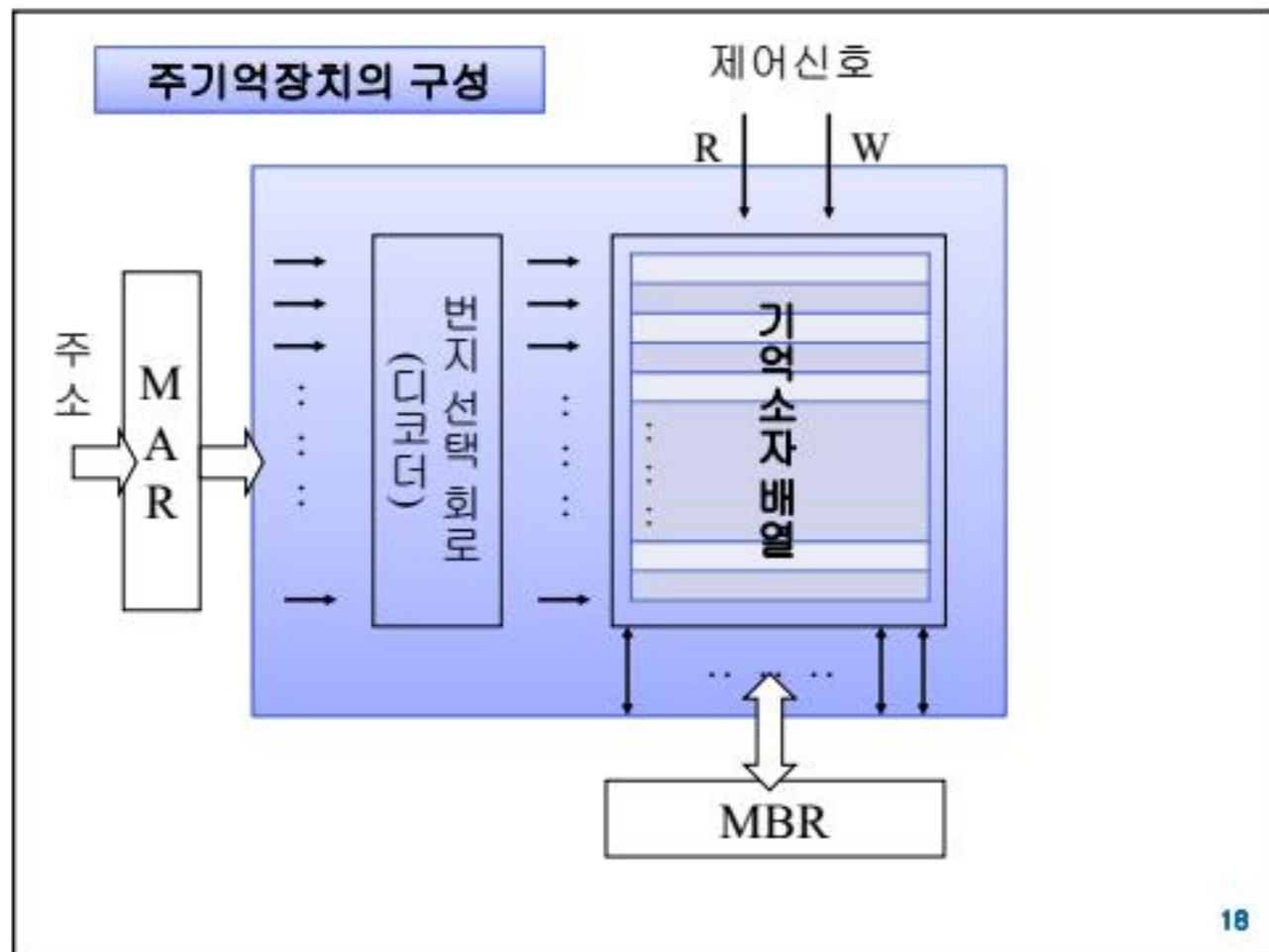
## 컴퓨터의 데이터 흐름/ 제어 흐름



17

17





18

## 설명

- MAR : memory address register

기억 소자 배열의  $n$ 번지에 데이터를 R/W할 경우, 먼저  $n$ (주소)을 MAR에 저장해야 한다.

- MBR : memory buffer register

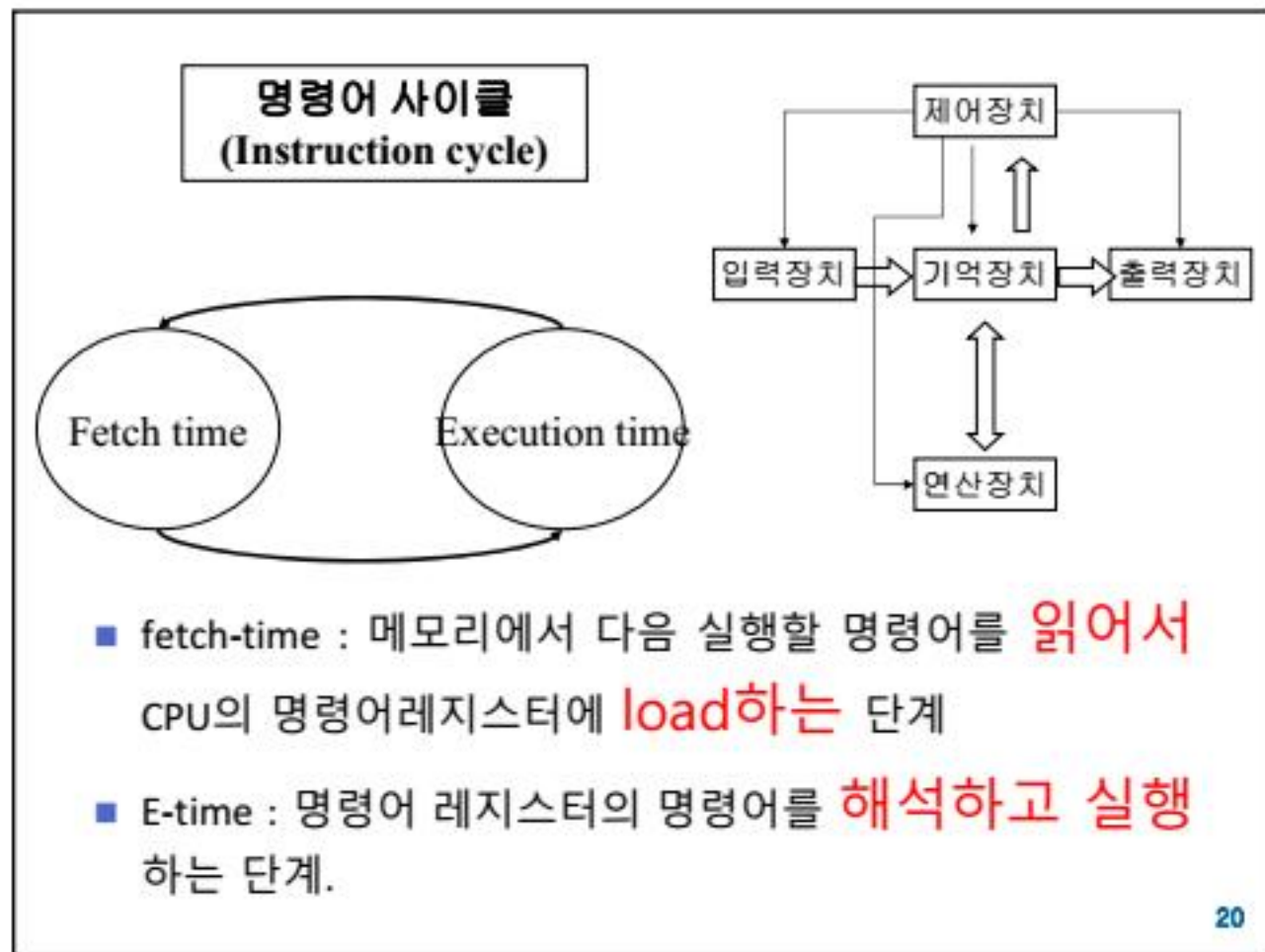
- $n$  번지에 **W**할 데이터를 임시 저장
- $n$  번지에 R한 후 데이터를 임시 저장하는 레지스터.

- R/W제어 : 제어장치에서 발생하는 제어 신호

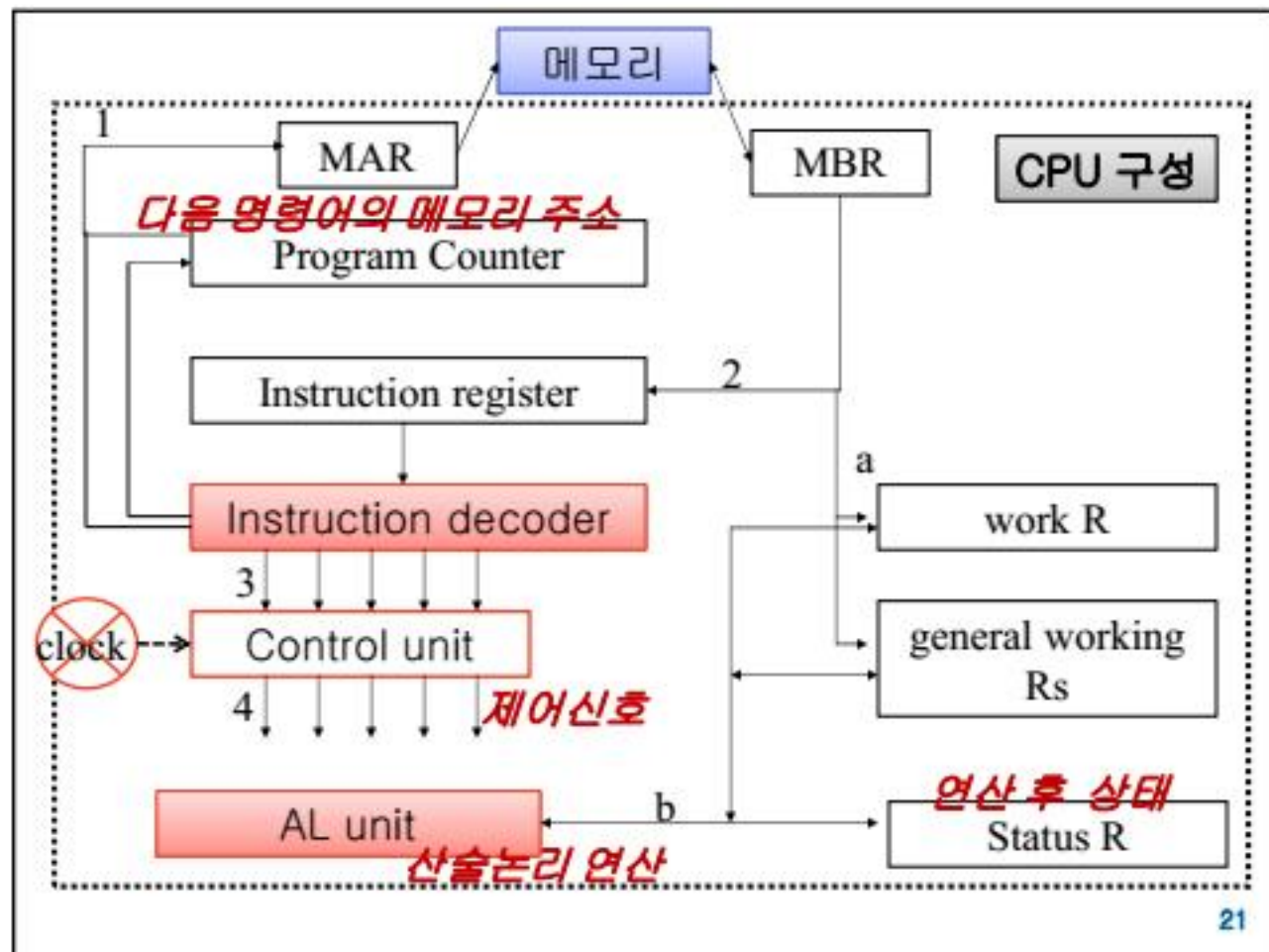
Register : CPU에 있는 소규모 데이터 기억장치 (64, 32, 128 bit 등)

19

19



20



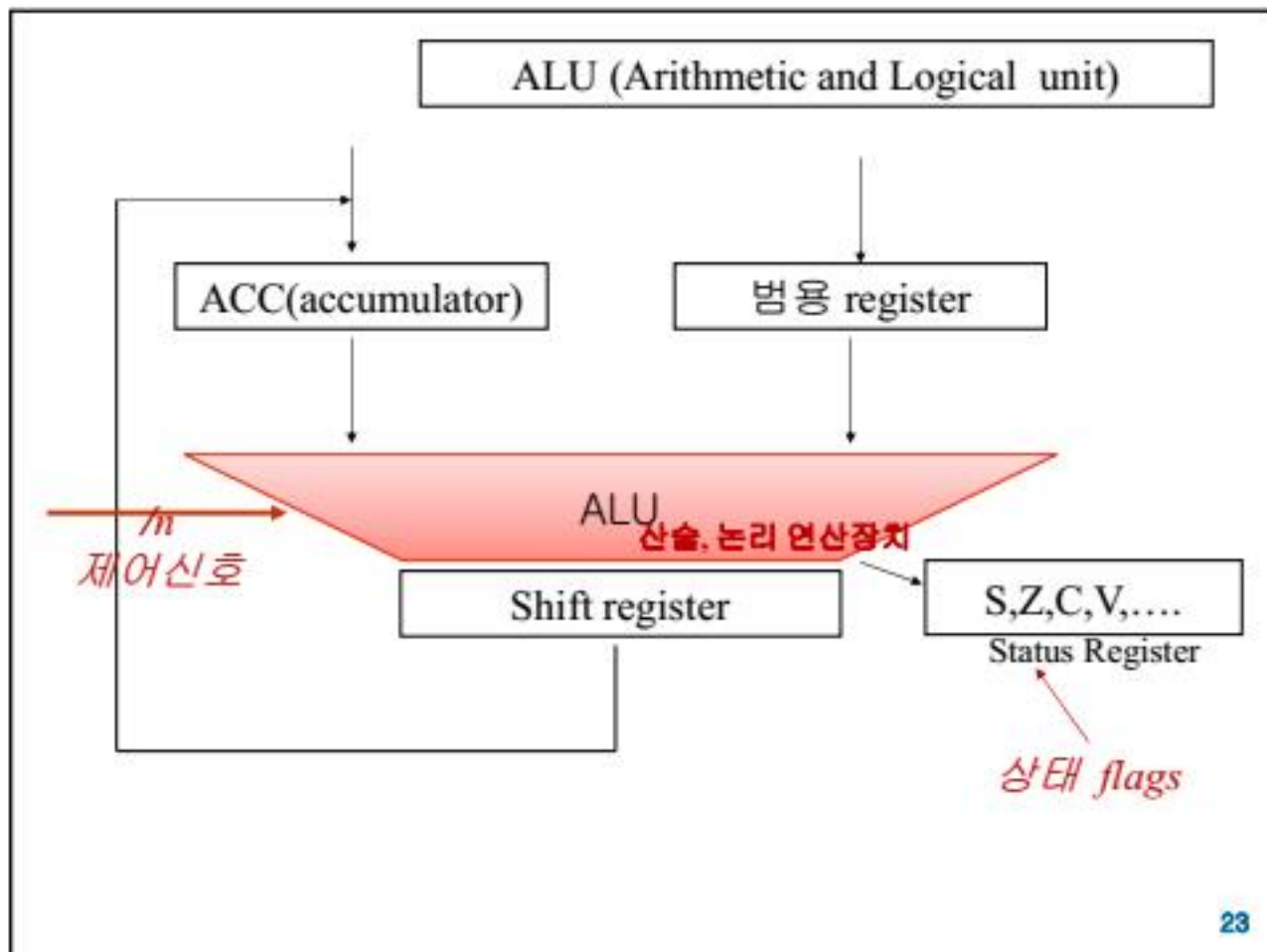
21

## 시스템

프로세서: Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz (3.60 GHz)  
설치된 메모리(RAM): 4.00GB(3.88GB 사용 가능)  
시스템 종류: 64비트 운영 체제, x64 기반 프로세서  
펜 및 터치: 이 디스플레이에 사용할 수 있는 펜 또는 터치식 입력이 없습니다.

22

22



23

23

## 설명

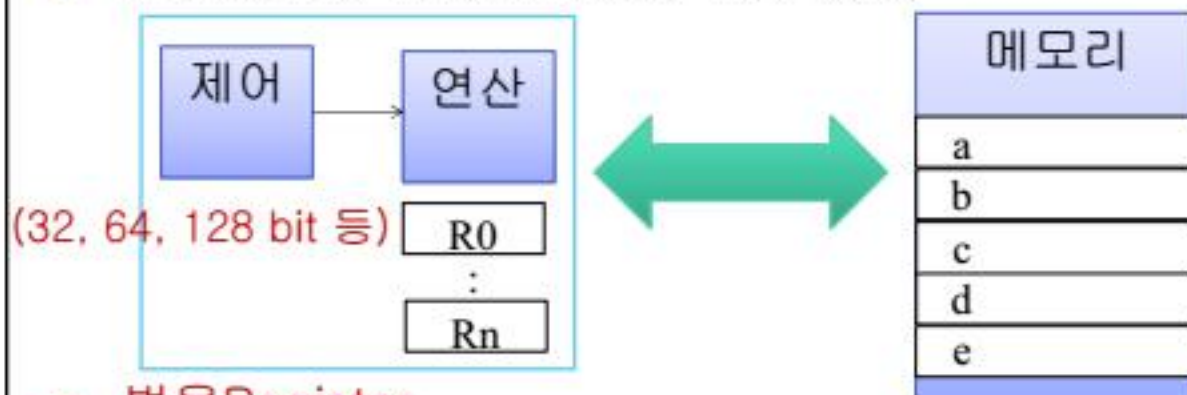
- program counter(PC): 다음 실행할 명령어의 메모리주소를 보관
- instruction register(IR) : PC가 지정한 메모리번지에서 읽어온 명령어를 해석하기전에 임시 보관 (**다음 실행할 명령어**)
- instruction decoder : 명령어 해독기(해석), instruction set에 대한 해독회로가 내장 ( 기계어 명령어 → 해석 → 순차적인 제어 신호set )
- control unit : clock에 맞춰 해독된 제어신호를 전송
- status register/flag 상태 레지스터 : 명령실행(연산,비교)후 결과의 상태를 저장하는 레지스터
  - 각 bit마다 carry,overflow,zero,sign 등의 결과를 표현한다.
- working register 작업 레지스터 : 명령의 실행에 쓰이는 레지스터 ( programmer가 지정하여 사용할 수 없다)
- general purpose register 범용 레지스터 : 여러 목적에 사용될 수 있는 레지스터. 레지스터번호(0,1,2..)를 두어 사용한다.
- clock : MHz, GHz

24

24

## 레지스터

- 고속 처리를 위해 n bits를 임시 저장



- 범용Register
- MAR, MBR
- 명령어Register
- Accumulator
- program counter
- 상태 Register
- Base Register
- index Register

25

25



## register

$$2^{32} = 4,294,967,296 \text{ (4GB)}$$

$$2^{64} = 18,446,744,073,709,551,616 \text{ (16 ExaByte)}$$

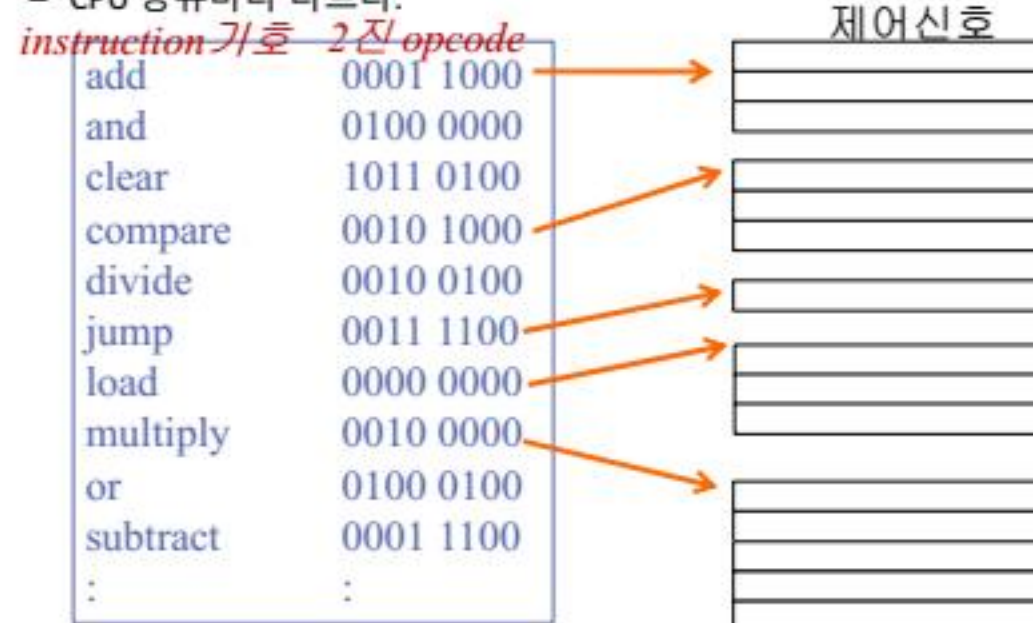
- 32bit-computer와 64bit-computer란?
  - CPU가 한 번에 처리할 수 있는 데이터의 크기(bits)
  - 32bit 단위의 처리능력 vs 64bit 단위의 처리능력
  - 32bit register                      64bit register
- MAR 의 크기 64bit
  - 실제 64bit 모두 사용하지 않는다.
  - OS에 따라 최대 지원 메모리 용량이 다르기 때문이다.
    - 예) Window 11 Education 2TB
    - Window 11 Enterprise 6TB

26

26

## Instruction set

- 한 컴퓨터가 사용할 수 있는 instruction code들의 집합
  - CPU 종류마다 다르다.



27

27

## CPU의 처리(fetch time)

- 메모리 → CPU 로의 데이터 흐름은 두 가지.

### ① 명령어

PC → MAR로 번지 전달

RAM의 해당 번지로 부터 데이터 → IR

### ② 데이터

IR → instruction Decoder

IR의 번지 부분 → MAR

RAM의 해당 번지로 부터 데이터

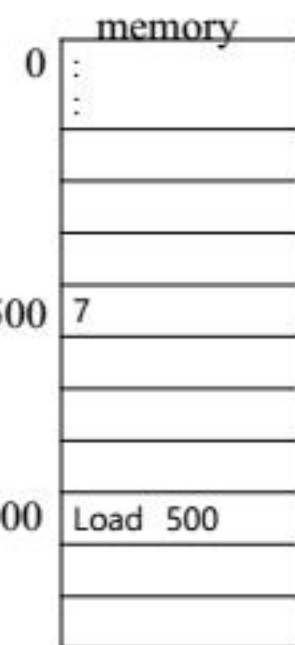
→ 작업 R 또는 범용 R

PC 1000

MAR

IR

general R



28

28

## CPU의 처리(fetch time)

- 메모리 → CPU 로의 데이터 흐름은 두가지.

### ① 명령어

PC → MAR로 번지 전달

RAM의 해당 번지로 부터 데이터 → IR

### ② 데이터

IR → instruction Decoder

IR의 번지 부분 → MAR

RAM의 해당 번지로 부터 데이터

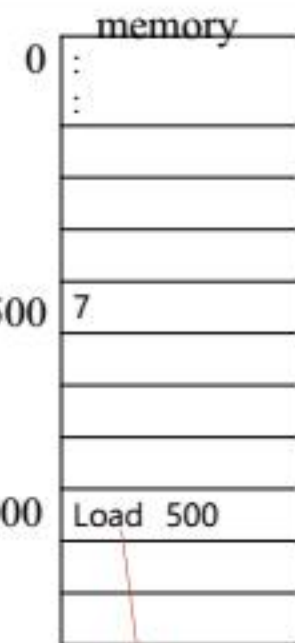
→ 작업 R 또는 범용 R

PC 1000

MAR 1000

IR Load 500

general R<sub>5</sub>



29

29

CPU의 처리(execution time)<sup>0</sup>

- 메모리 → CPU 로의 데이터 흐름은 두가지.

### ① 명령어

PC → MAR로 번지 전달

RAM의 해당 번지로 부터 데이터 → IR

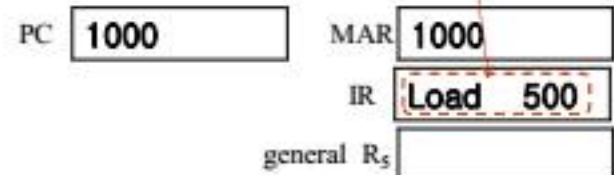
## ② 데이터

IR → instruction Decoder

IR의 변지 부분 → MAR

RAM의 해당 번지로 부터 데이터

→ 작업 R 또는 범용 R



30

30

CPU의 처리(execution time)<sup>0</sup>

- 메모리 → CPU 로의 데이터 흐름은 두가지.

① 명령어

PC → MAR로 번지 전달

RAM의 해당 번지로 부터 데이터 → IR

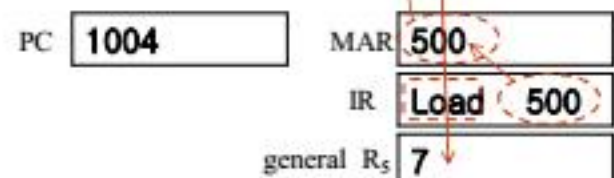
## ② 데이터

IR → instruction Decoder

IR의 번지 부분 → MAR

RAM의 해당 번지로 부터 데이터

→ 작업 R 또는 범용 R



31

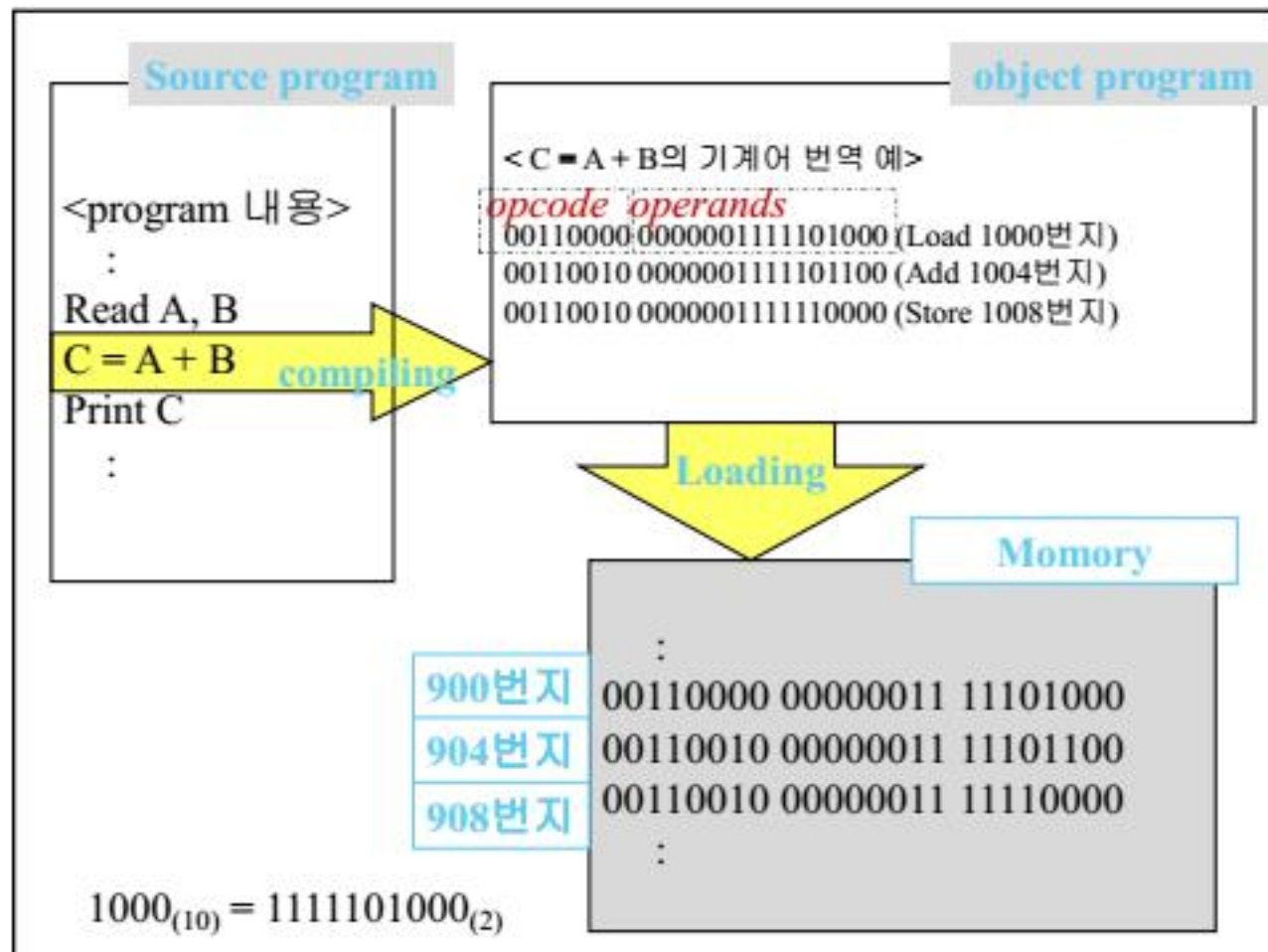
31



그림으로 이해

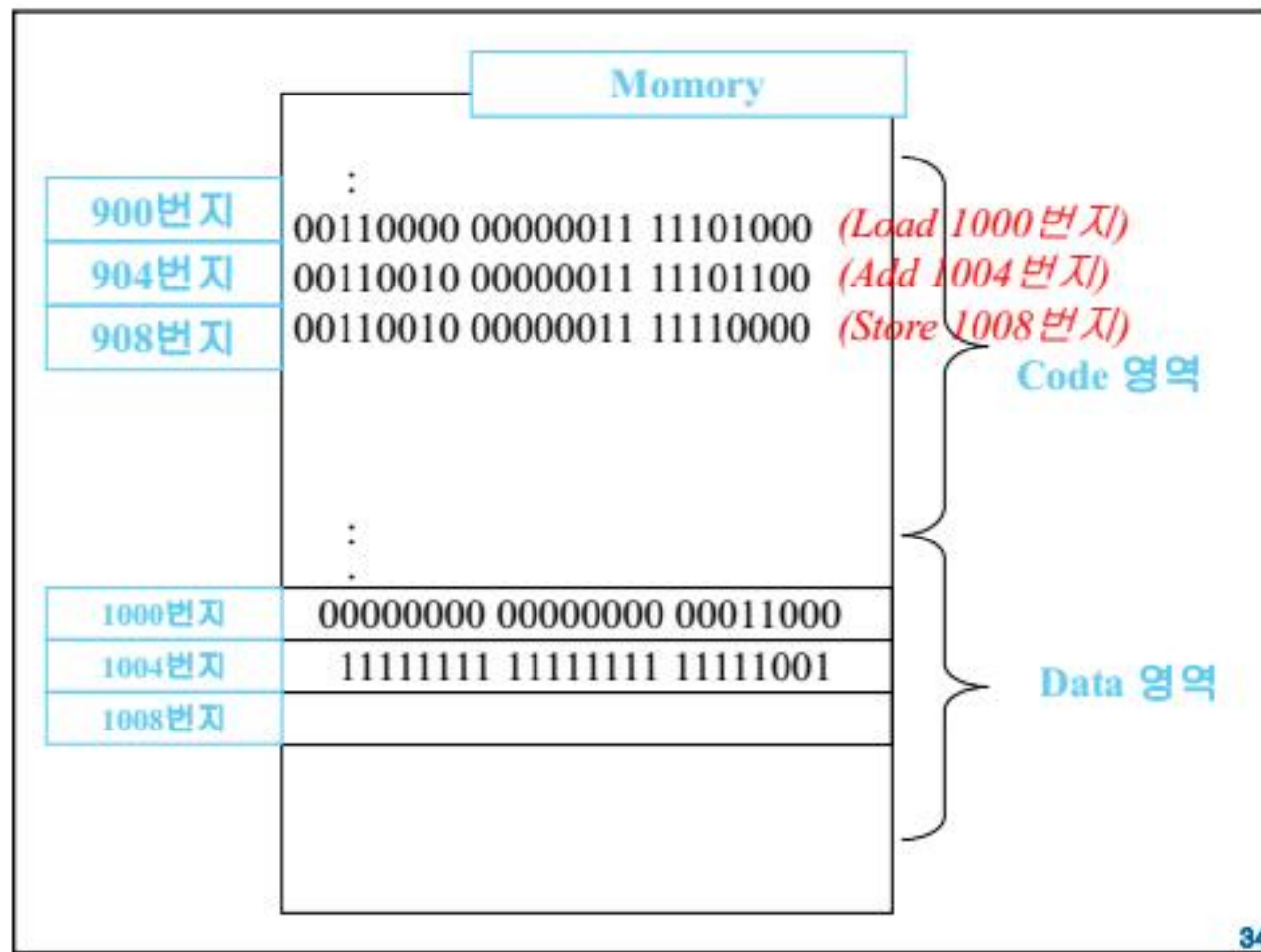
32

32

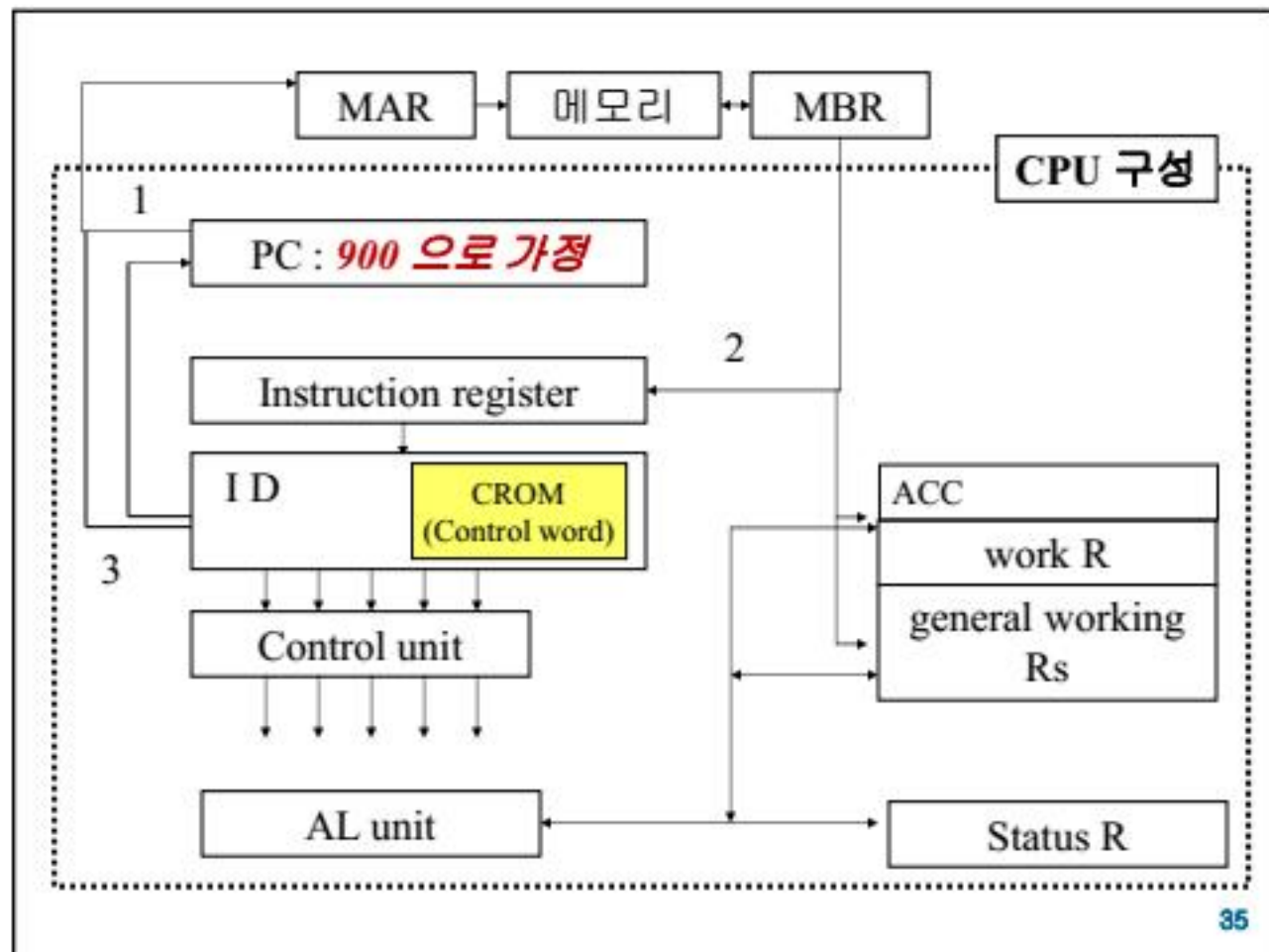


33





34



35









