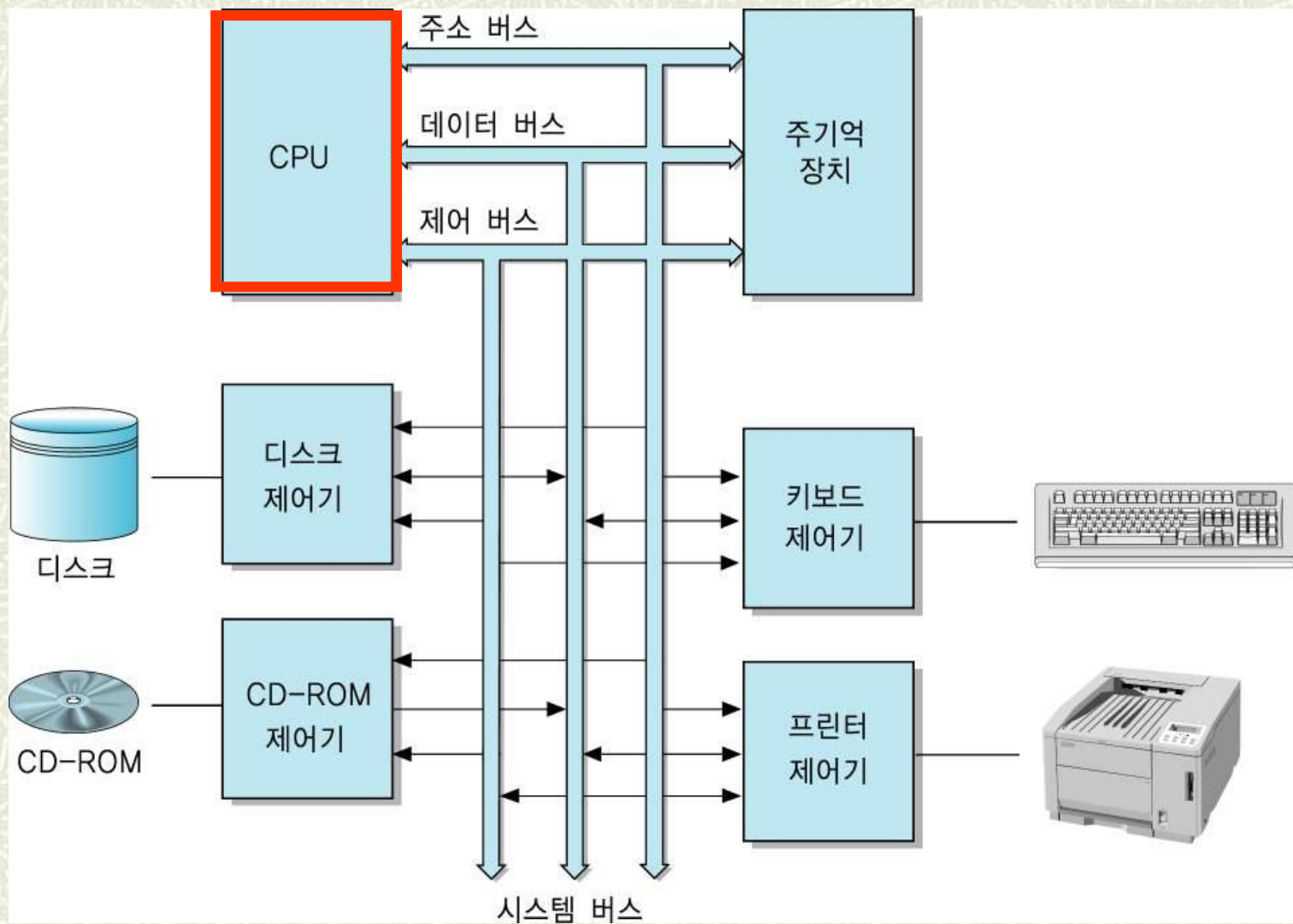


제2장 CPU의 구조와 기능 (1)

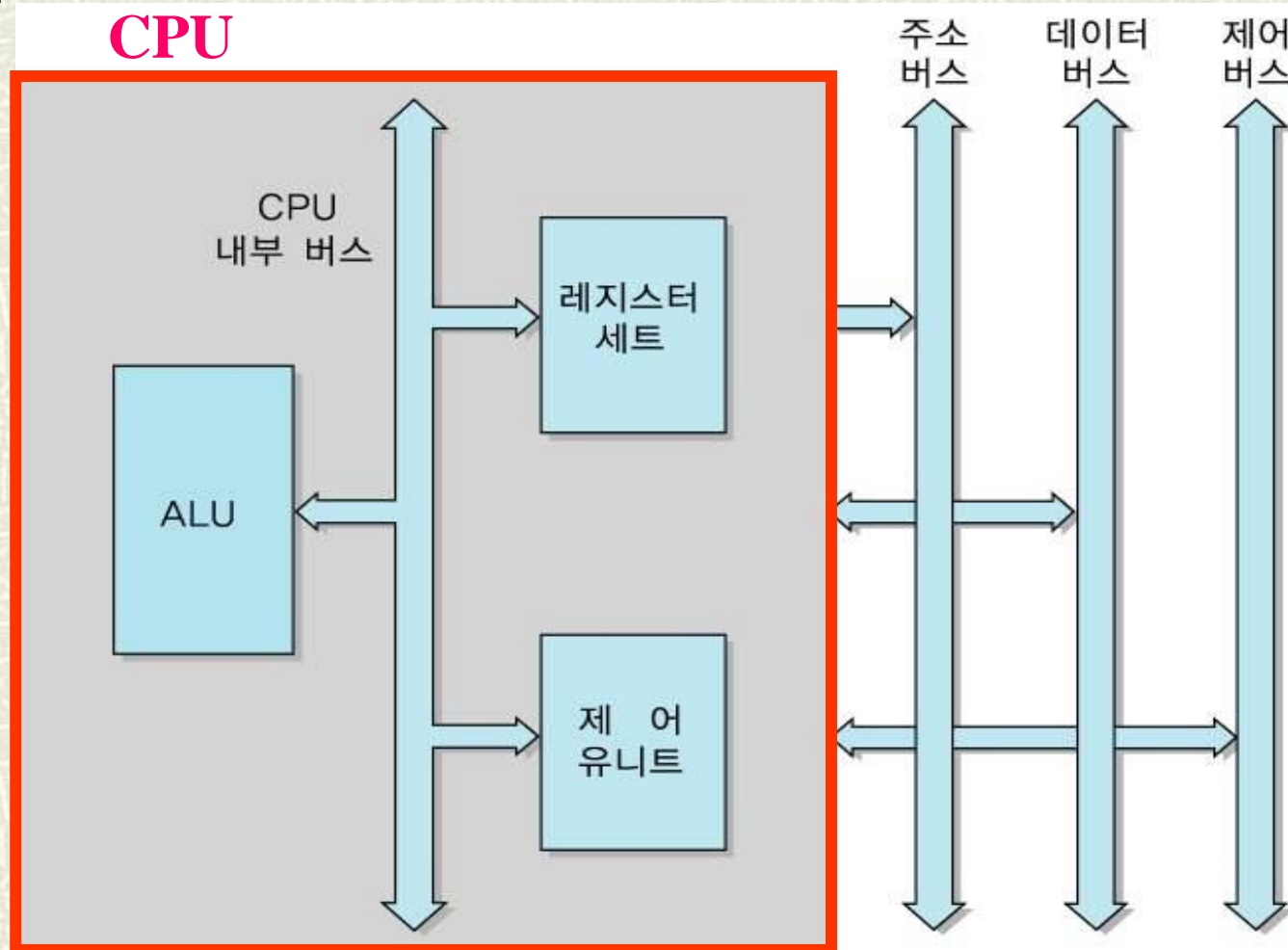
- CPU의 기본 구조
 - CPU 동작 원리
- CPU 내부 Register들의 이해
 - 레지스터와 메모리와의 관계
- 명령어 실행/처리 과정
 - Instruction Cycle

컴퓨터 시스템 구조



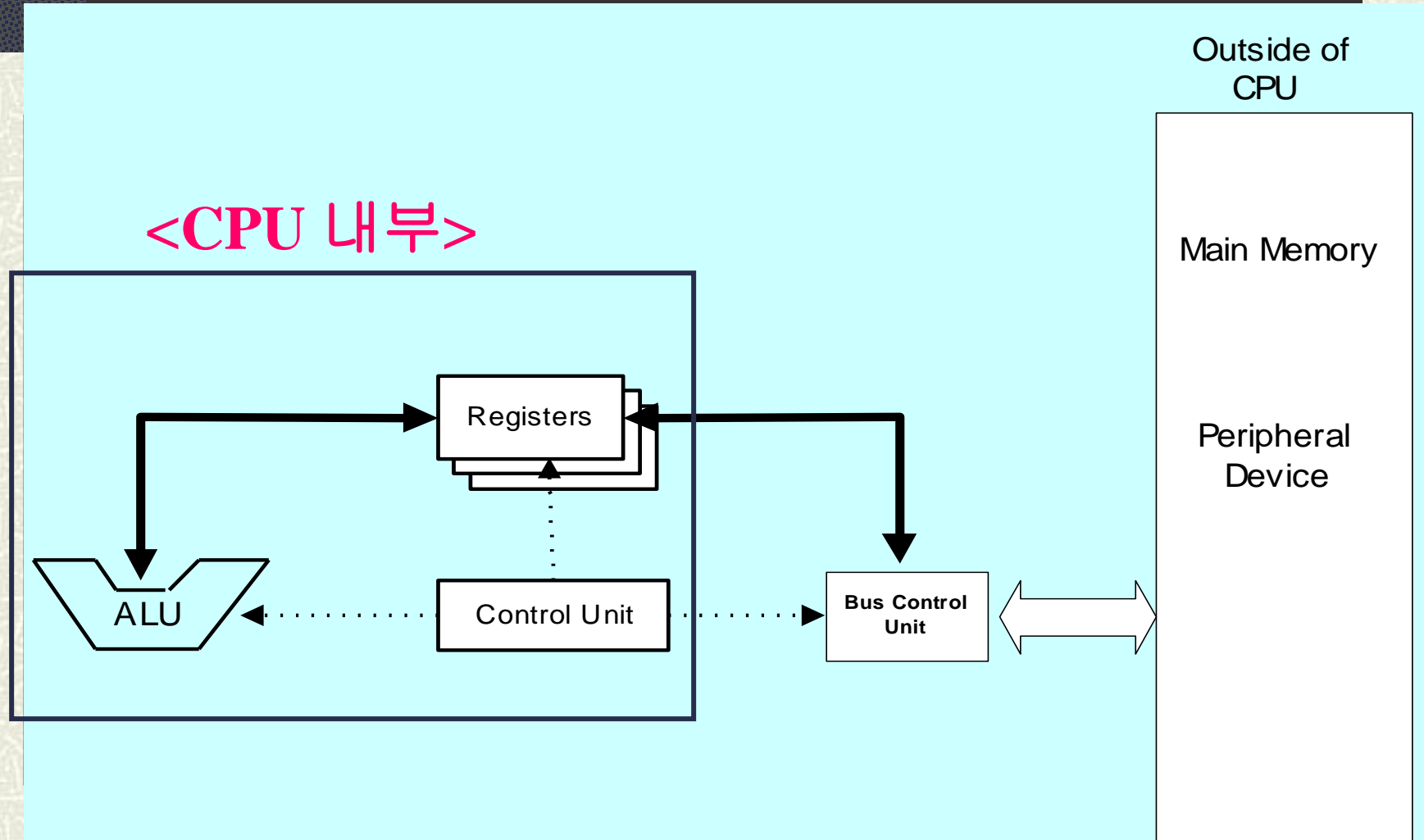


CPU의 기본 구조





CPU 내부 구조



CPU의 내부 구성 요소

Central Processing Unit (CPU)

■ 프로그램 실행하는 장치

① ALU(Arithmetic & Logic Unit)

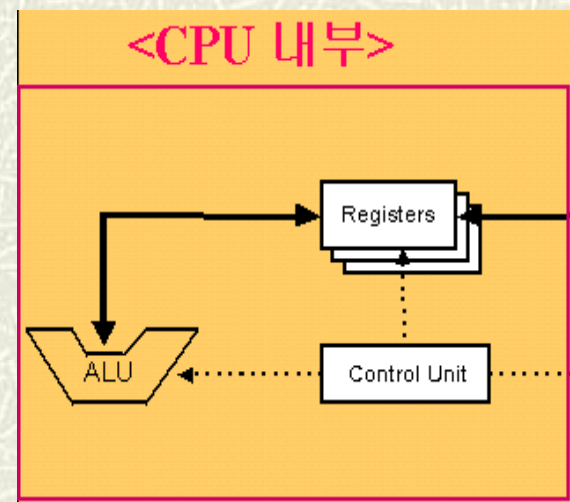
- 기계 명령어가 지시하는 내용을 실행

② Register set

- 필요한 정보를 임시로 저장

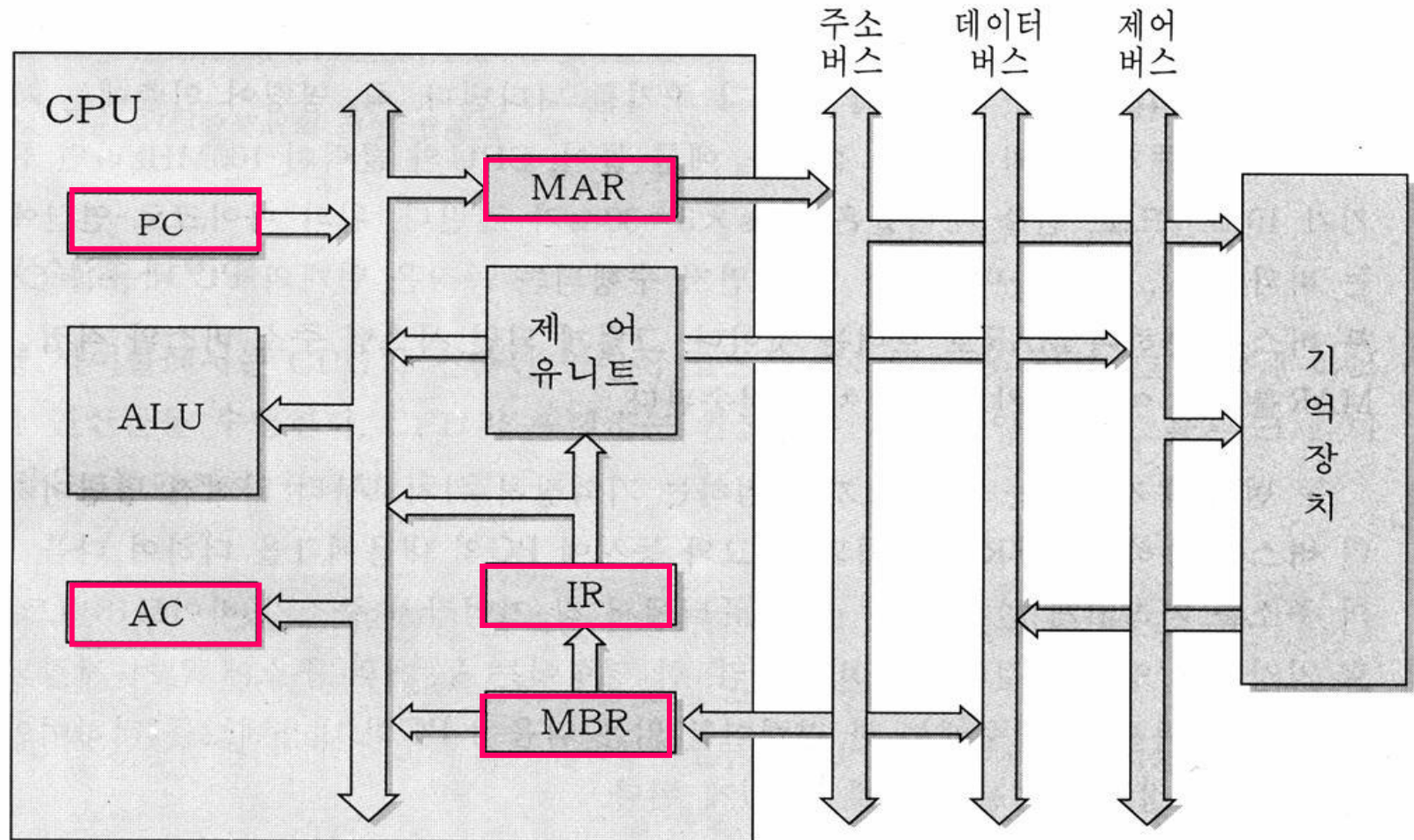
③ CU(Control Unit)

- 프로그램 코드(명령어) 해석
- 각 명령어의 실행 과정을 관리, 제어

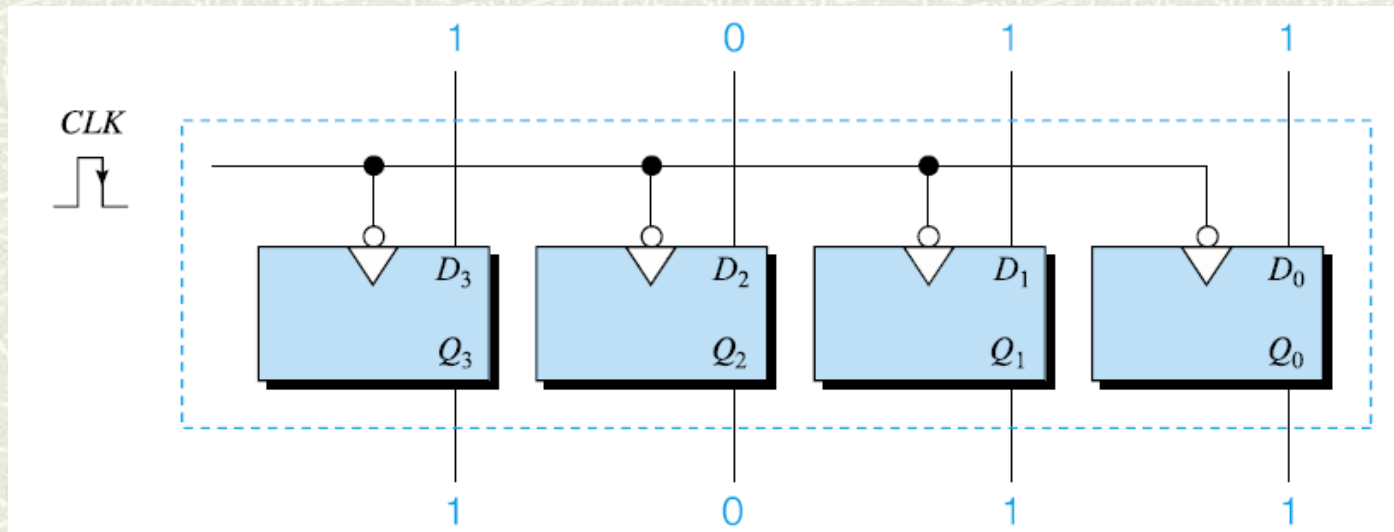


CPU구조

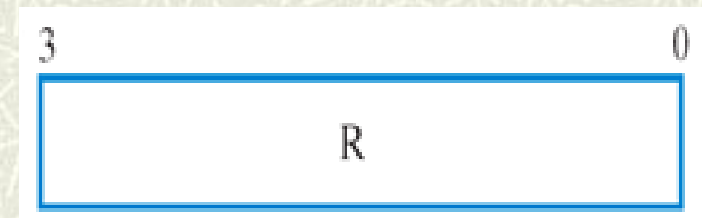
Register들과 데이터 통로를 이용한 표현



Register 예 (4-bit reg.)



- ☞ F/F(Flip-Flop)으로 구성
- ☞ 정보 이동 및 처리 결과 저장



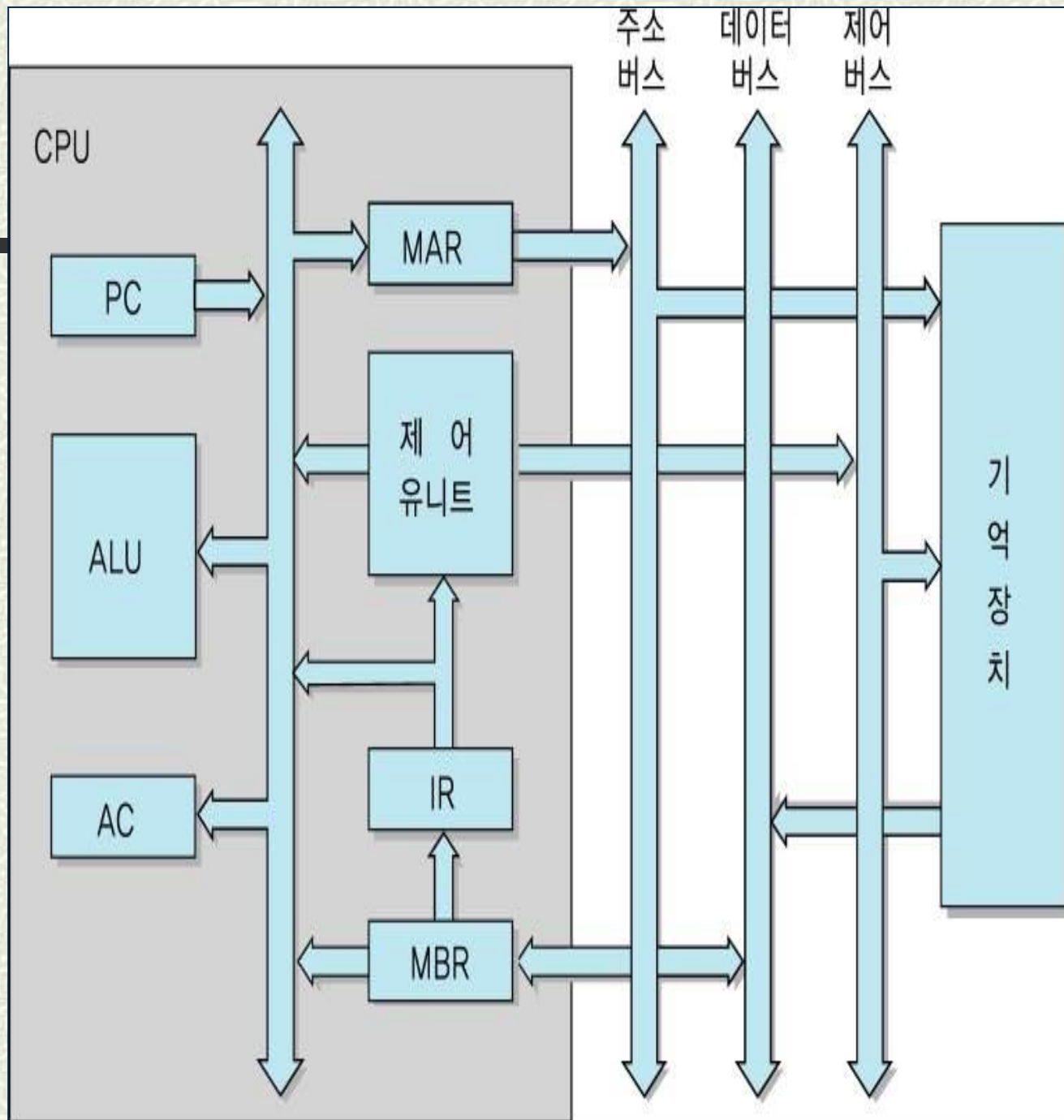
numbering of 4-bit register

레지스터(Register)

- # F/F(Flip-Flop)으로 구성
- # 정보 이동 및 처리 결과 저장
- # 종류
 - PC(program counter)
 - 다음에 실행할 명령어의 번지를 기억
 - ACC(accumulator)
 - 연산시 결과를 일시적으로 저장

레지스터(Register)

- MAR(Memory Address Register)
 - 수행될 명령어 또는 데이터가 있는 메모리 번지를 기억하는 레지스터
 - MBR(Memory Buffer Register)
 - 기억장소에 출입하는 자료를 기억
 - IR(Instruction Register)
 - 명령부분을 기억(현재 수행중인 명령기억)
 - GR(General Purpose Register)
 - 사용자가 사용할 수 있는 레지스터
-



CPU 레지스터와 메모리와의 관계

기억장치

- 4096 words
- 각 워드 16비트

MAR, PC

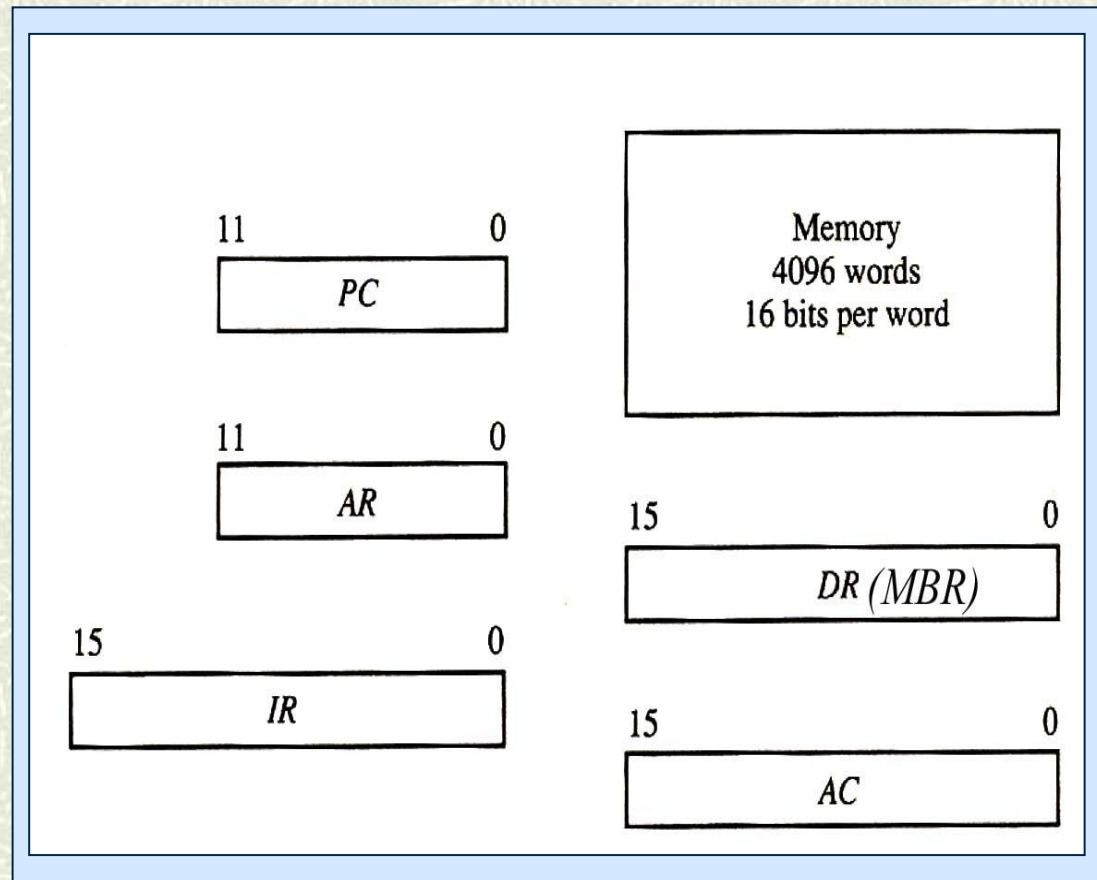
- 12bits

MBR, AC

- 16비트

IR

- 16비트



제2장 CPU의 구조와 기능 (계속)

2.1 CPU의 기본 구조

2.2 명령어 실행 /처리 과정

➤ Instruction Cycle

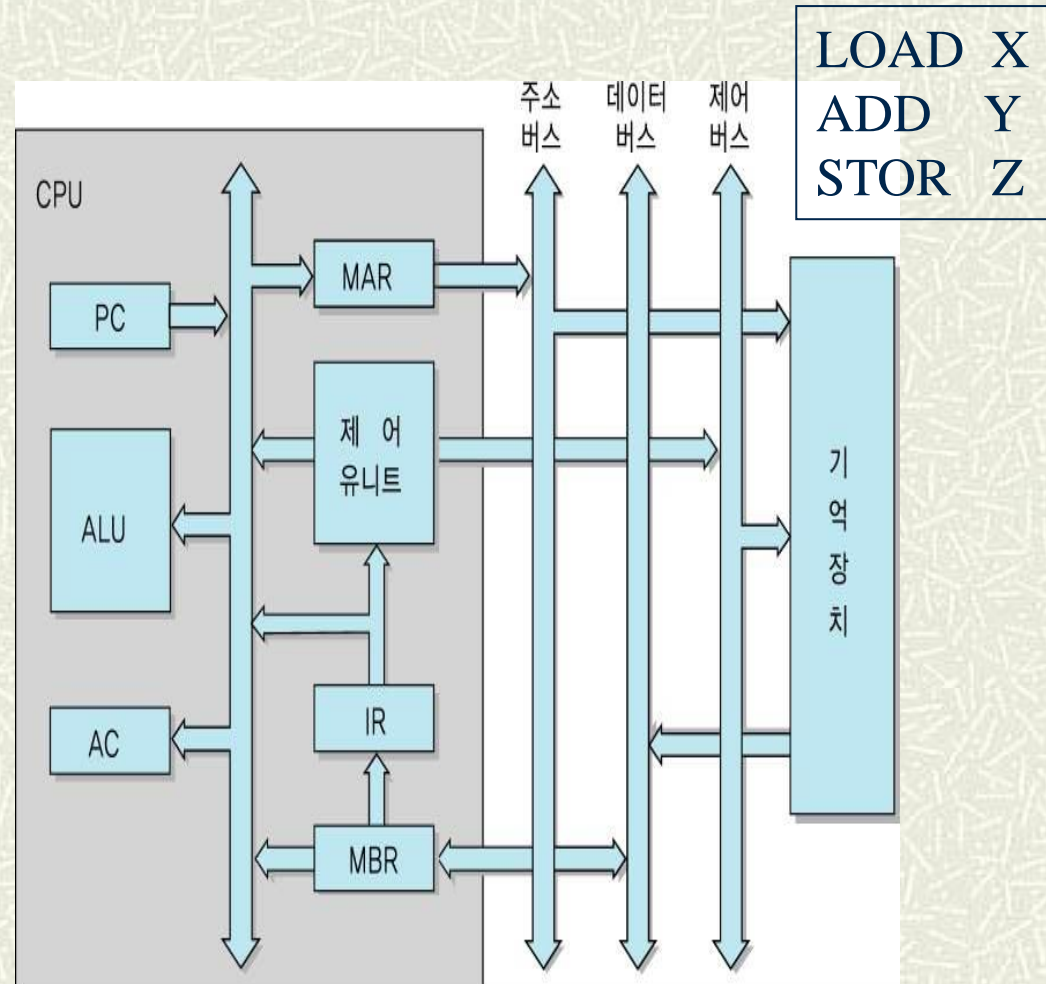
2.3 명령어 파이프라이닝

2.4 명령어 세트

CPU의 동작 원리

(프로그램 수행 기능)

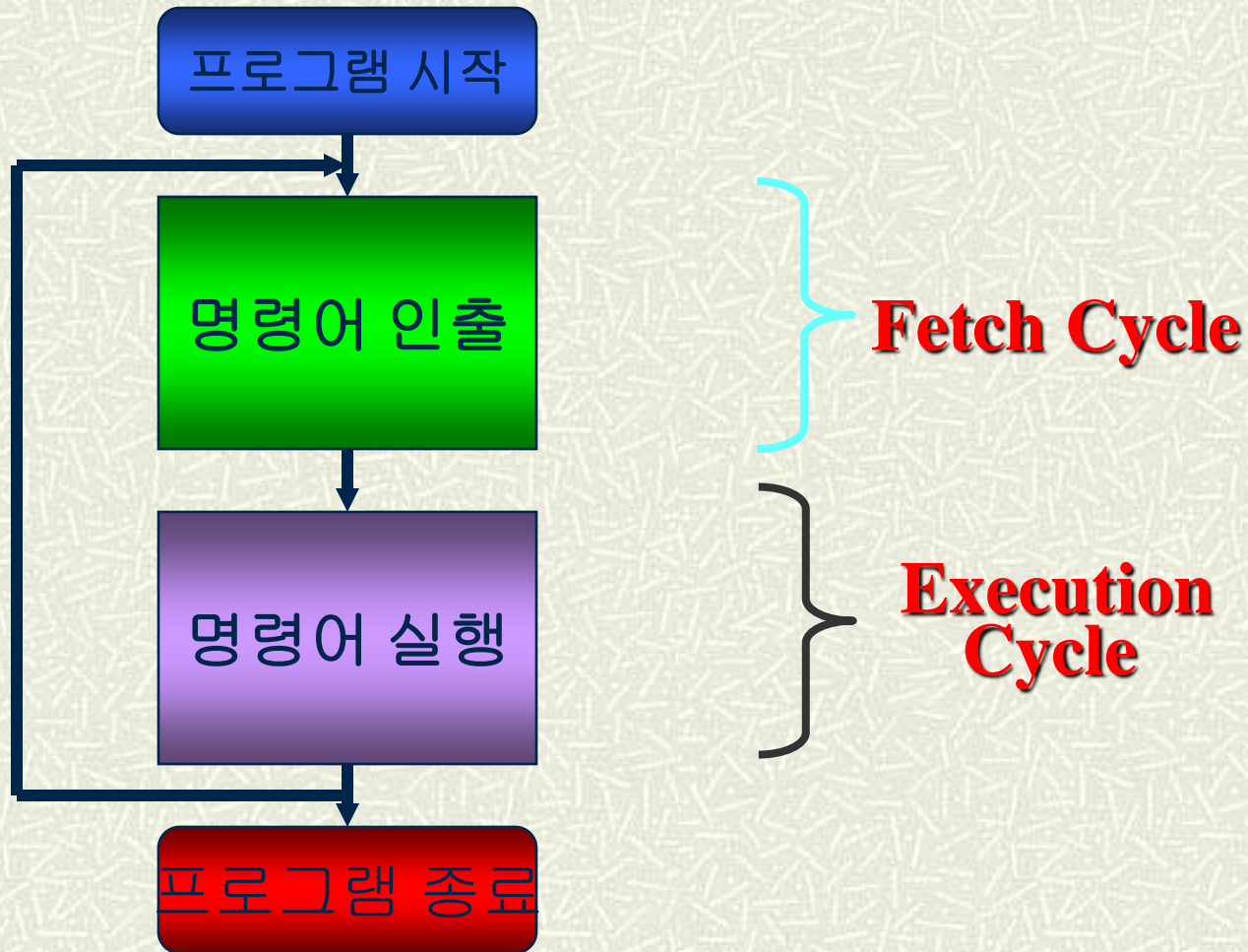
1. **Instruction** *Fetch*
2. Instruction *Decode*
3. **Data** *Fetch*
4. Data *Process*
5. Data *Store*



명령어 실행

- # CPU는 memory에 저장된 명령어들을 한 번에 한 개씩 읽어서(인출)하여 실행함.
- # 한 명령어 처리 과정(**Instruction Cycle**)
 - ① Instruction Fetch cycle (명령어 인출 단계)
 - memory로 부터 한개 명령어를 읽어 오는 단계.
 - ② Instruction Execution cycle (명령어 수행 단계)
 - 인출된 명령어를 수행(실행) 하는 단계

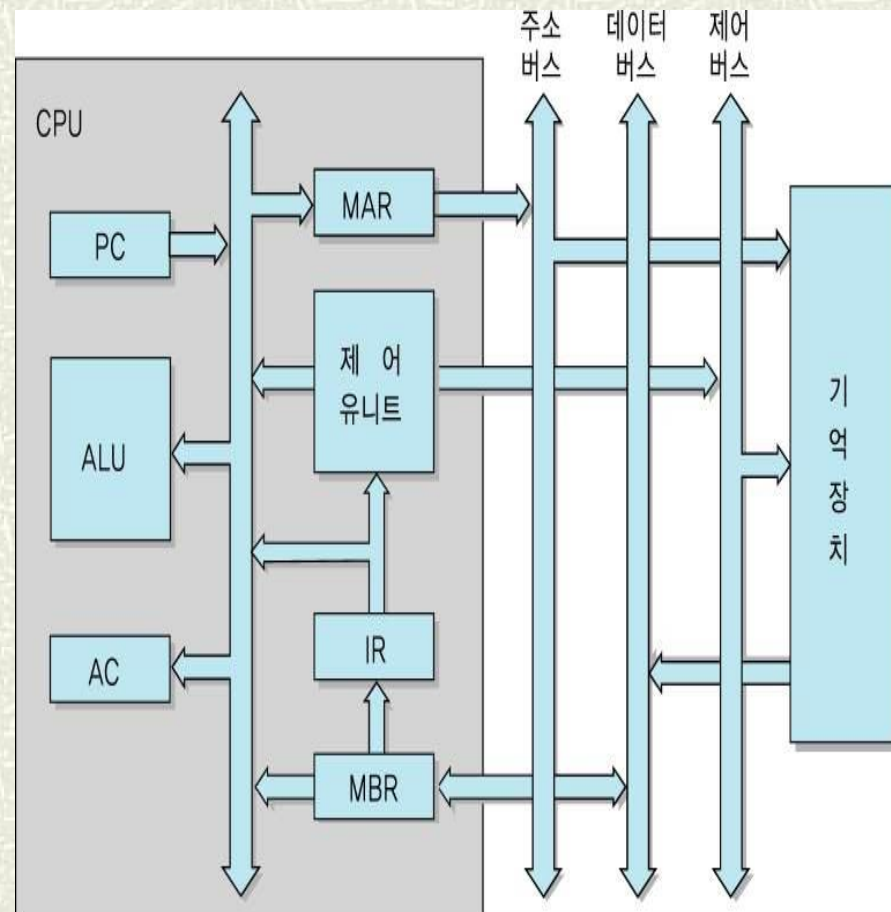
명령어 처리 과정

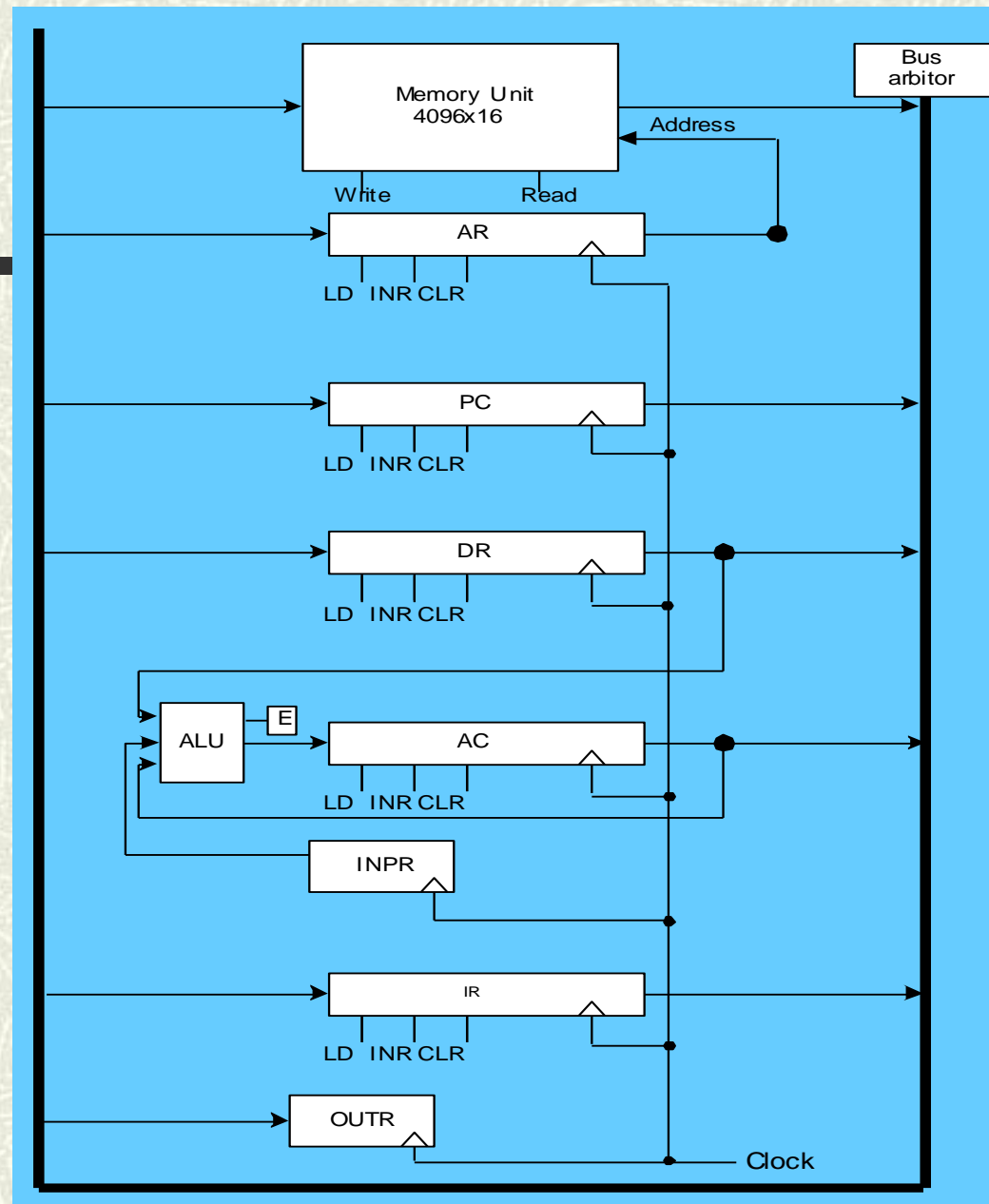
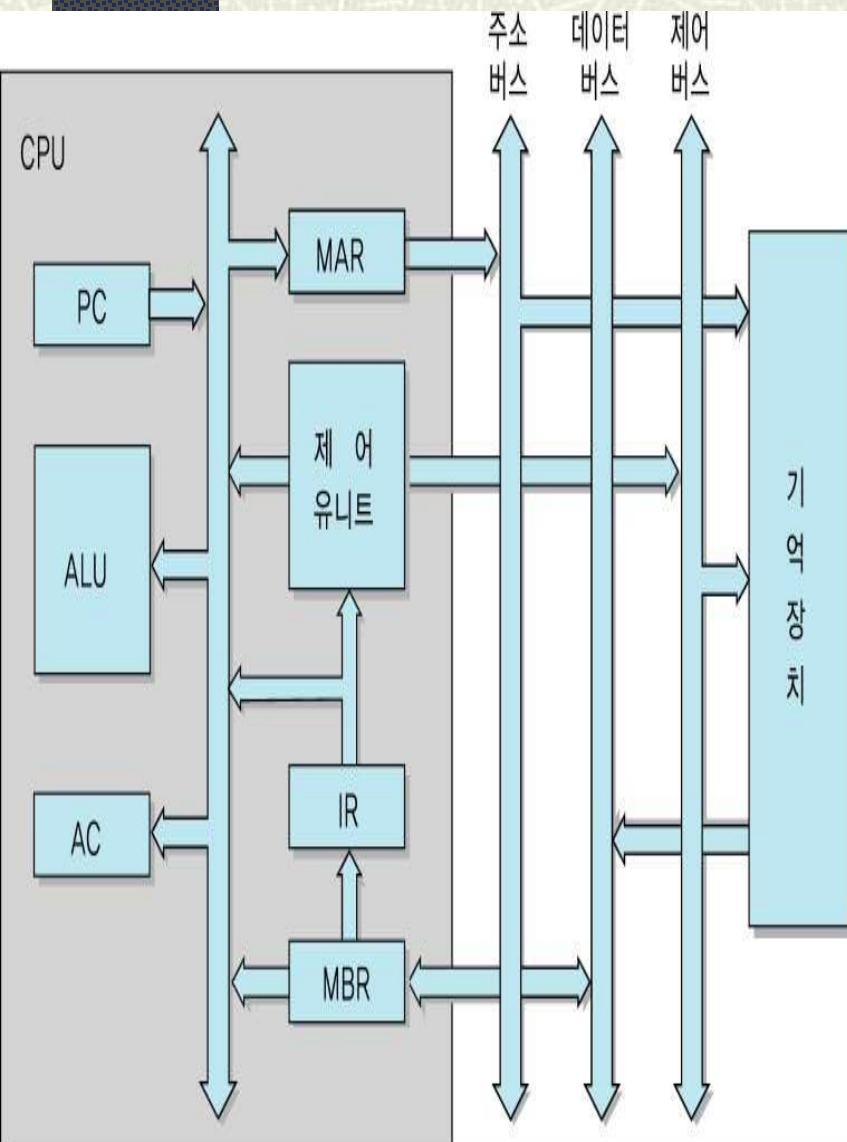


LOAD	X
ADD	Y
STOR	Z

CPU 레지스터와 데이터 통로

- ‡ **PC(Program Counter)**
 - 다음에 인출할 명령어의 주소를 가지고 있음.
- ‡ **MAR(Memory Address Reg.)**
 - 수행될 명령어의 주소 번지(PC내용) 또는 데이터가 있는 메모리 주소 번지를 주소버스로의 출력전에 일시적으로 기억하는 레지스터
- ‡ **MBR(Memory Buffer Register)**
 - 기억장치에 쓰여질 혹은 읽혀질 데이터를 일시적으로 저장하는 버퍼 레지스터
- ‡ **IR(Instruction Register)**
 - 인출된 명령어가 저장
- ‡ **AC(Accumulator)**
 - 데이터가 계산되기 전/후에 임시적으로 저장
 - AC크기는 데이터가 한번에 처리할 수 있는 데이터 비트 수

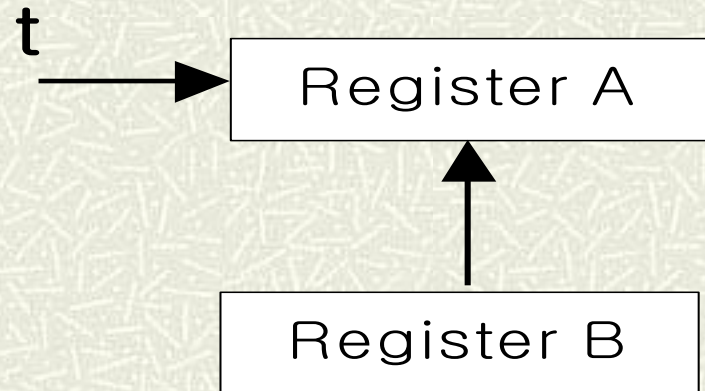




명령어 실행과 제어함수(신호) (Control Function (Signal))

$t : A \leftarrow B$

- 위의 마이크로 동작 (*Micro-Operation*)은 제어선 t 가 1일 때만 B의 내용이 A로 저장 (옮겨짐)



CPU Clock



μ -operation :
CPU 클럭 주기 동안
수행되는 기본적인 동작



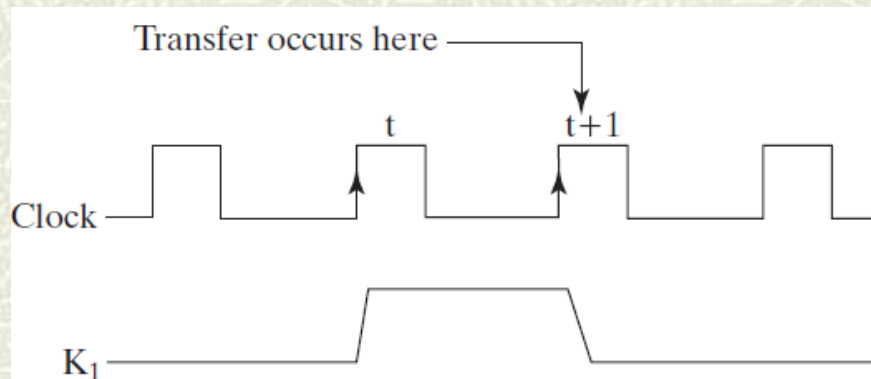
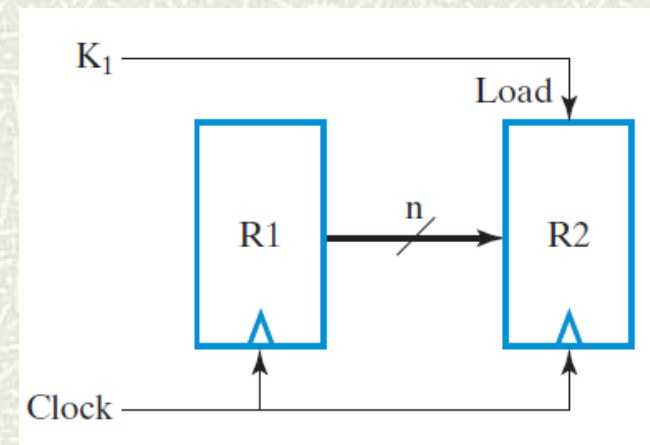
Timing & Control

명령어 실행과 제어함수(신호) (Control Function (Signal))

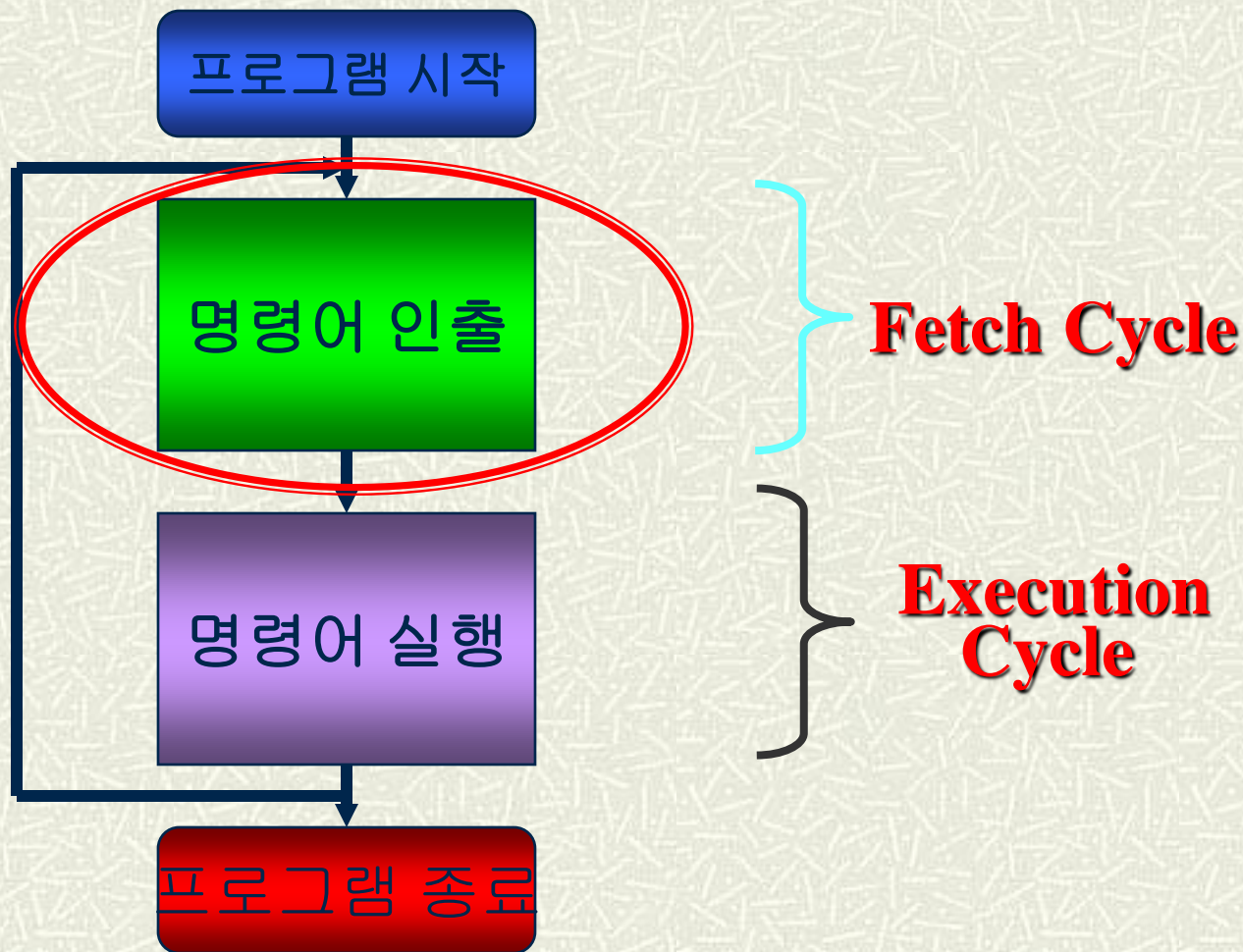
- * 제어신호 : 데이터 처리동작을 활성화시키는 2진 신호
- * Micro operation : 레지스터나 메모리의 데이터에 가해지는 기본동작

K1: $R2 \leftarrow R1$

- 위의 마이크로 동작(*Micro-Operation*)은 제어선 K1이 1일 때만 R1의 내용이 R2로 저장(옮겨짐)



명령어 처리 과정 (*Instruction Cycle*)



LOAD	X
ADD	Y
STOR	Z

Fetch Cycle

Execution Cycle

컴퓨터 사이클

Micro-Operation

Fetch(인출) 사이클

- $t_0 : \text{MAR} \leftarrow \text{PC}$
- $t_1 : \text{MBR} \leftarrow \text{M}, \text{PC} \leftarrow \text{PC} + 1$
- $t_2 : \text{IR} \leftarrow \text{MBR}(\text{OP})$

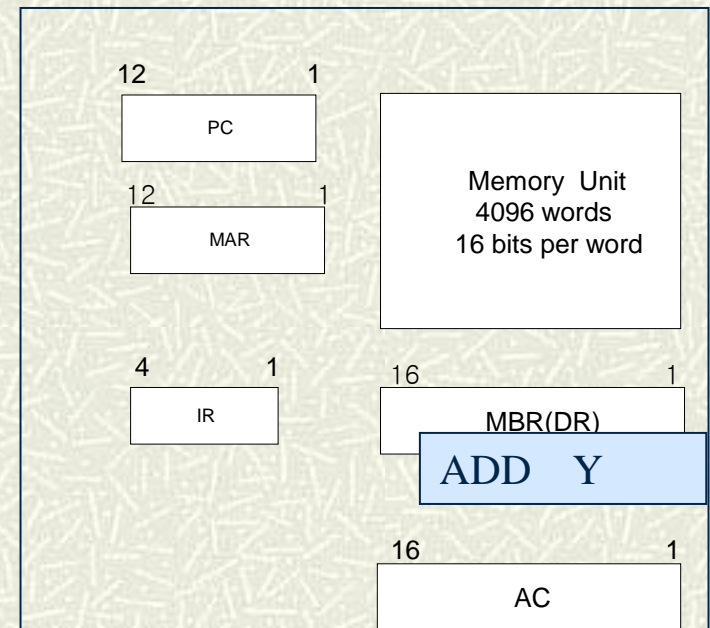
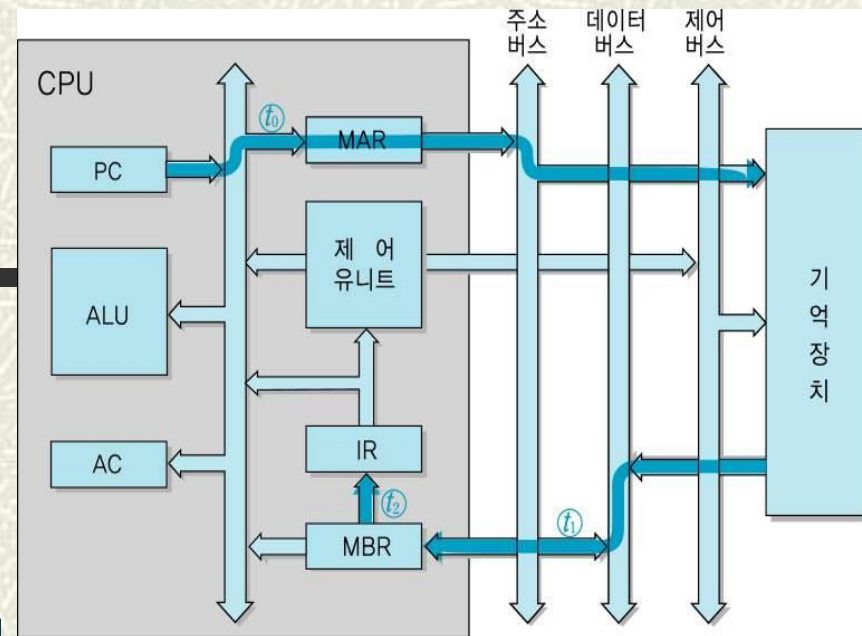
교재

$t_0 : \text{MAR} \leftarrow \text{PC}$

$t_1 : \text{MBR} \leftarrow \text{M}[\text{MAR}], \text{PC} \leftarrow \text{PC} + 1$

$t_2 : \text{IR} \leftarrow \text{MBR}$

MBR	Operation (OP)	Operand(주소)
-----	-------------------	-------------



Fetch에 대한 처리 흐름

(t_1 에서의 마이크로 동작의 동시 수행)

Fetch 사이클(계속)

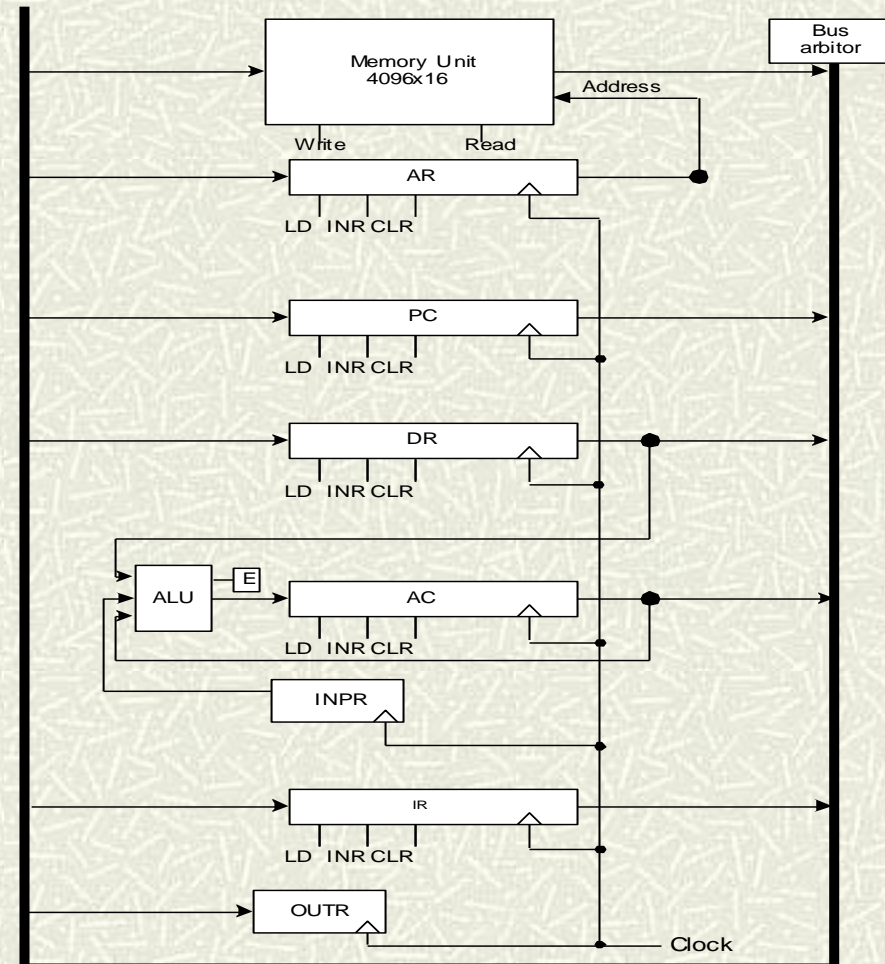
$t_0 : \text{MAR} \leftarrow \text{PC}$

$t_1 : \text{MBR} \leftarrow \text{M}, \text{PC} \leftarrow \text{PC} + 1$

$t_2 : \text{IR} \leftarrow \text{MBR}(\text{OP})$

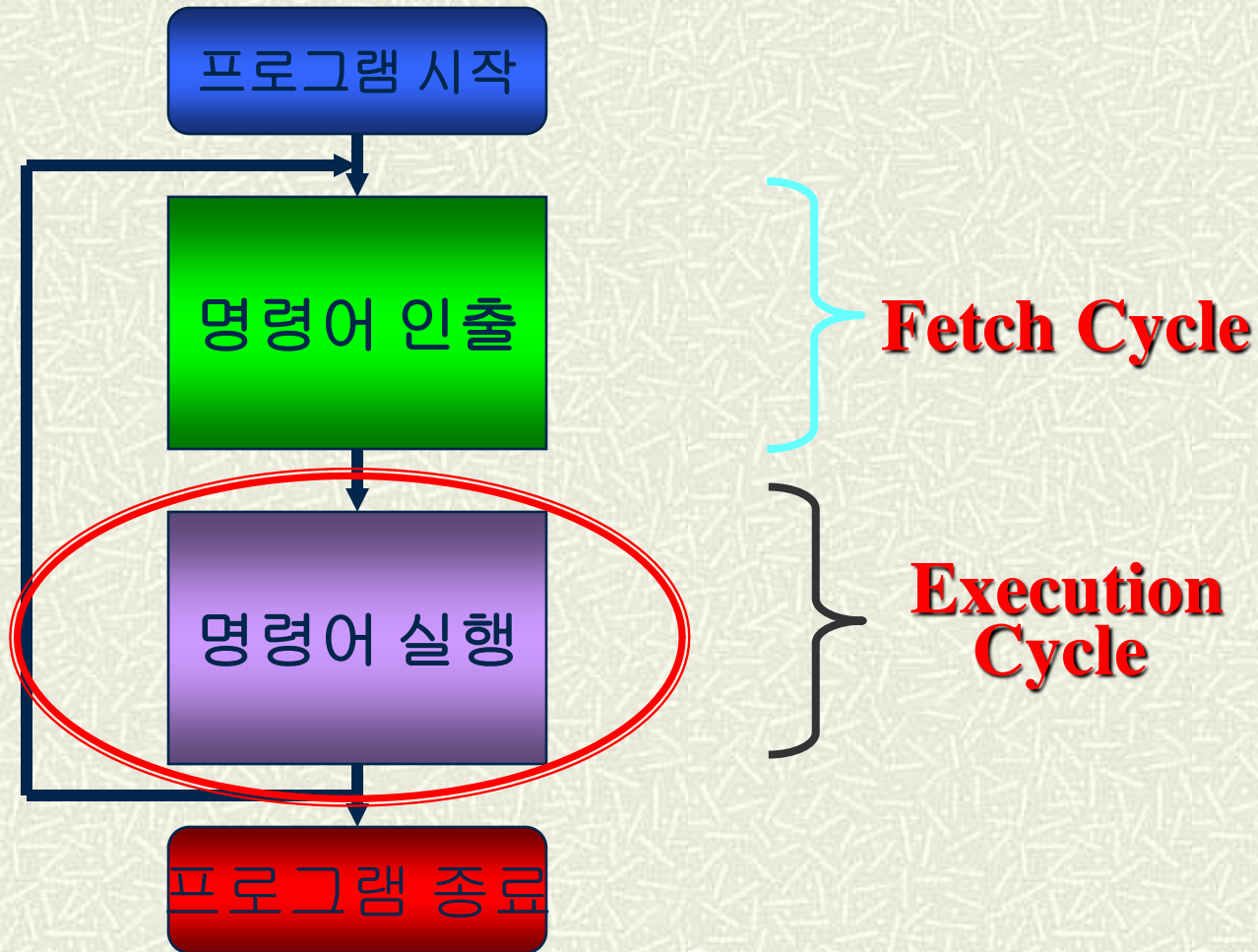
MBR

Operation	Operand(Addr)
-----------	---------------





명령어 처리 과정



실행 사이클

(예, Load, Move , Add, Store, Jump 등)

Fetch한 명령을 Decoding한 후에 Execution

CPU가 수행하는 연산자들의 분류

- 데이터 이동
 - CPU와 메모리간 또는 I/O 장치등으로 데이터 이동
 - 데이터 처리
 - 산술 또는 논리 연산을 수행
 - 데이터 저장
 - 연산 처리 결과를 기억장치에 저장
 - 제어
 - 프로그램의 수행 순서를 결정
-

LOAD X

MBR

Operation

Operand(Addr)

실행 사이클(데이터 이동)

#LOAD 명령 실행

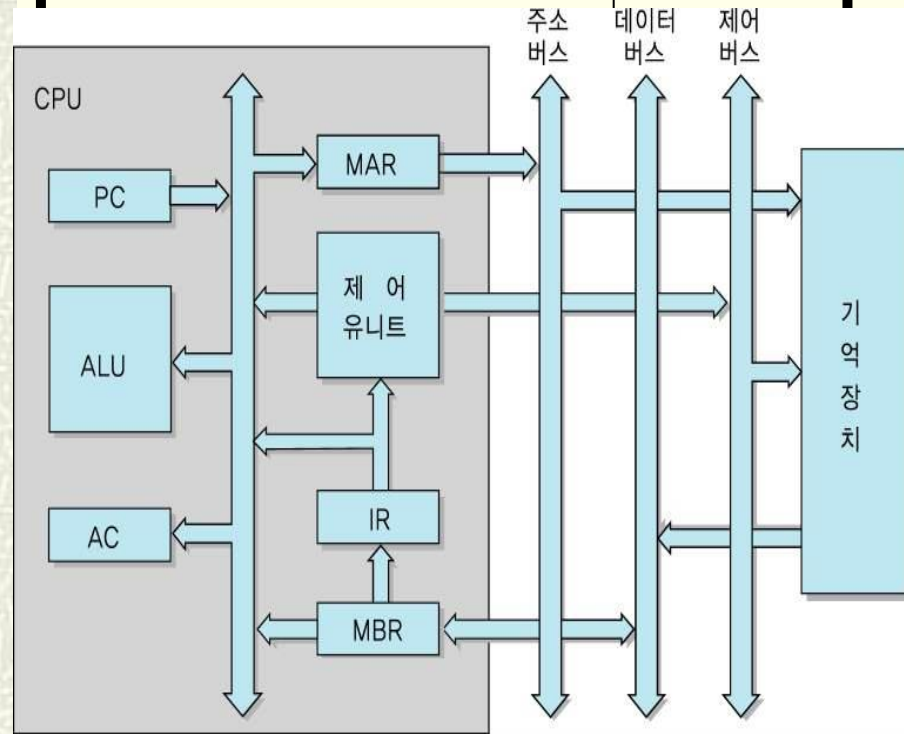
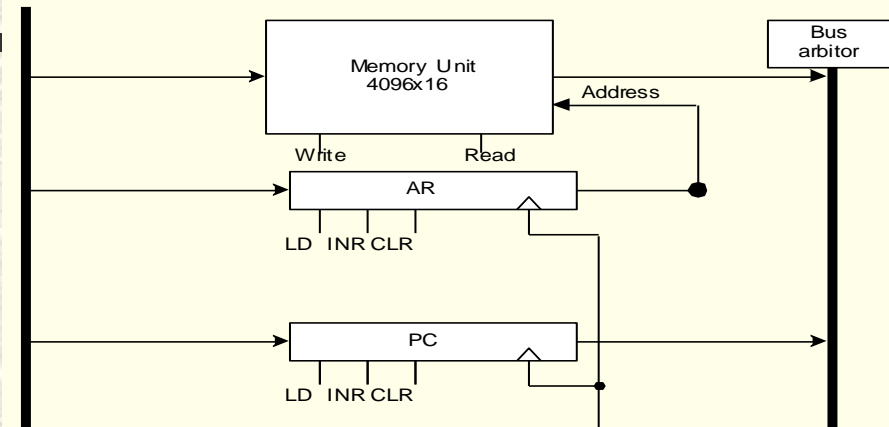
- $t_0 : MAR \leftarrow MBR(AD)$
- $t_1 : MBR \leftarrow M, AC \leftarrow 0$
- $t_2 : AC \leftarrow AC + MBR$

교재

$t_0 : MAR \leftarrow IR(addr)$

$t_1 : MBR \leftarrow M[MAR]$

$t_2 : AC \leftarrow MBR$



IR

Operation

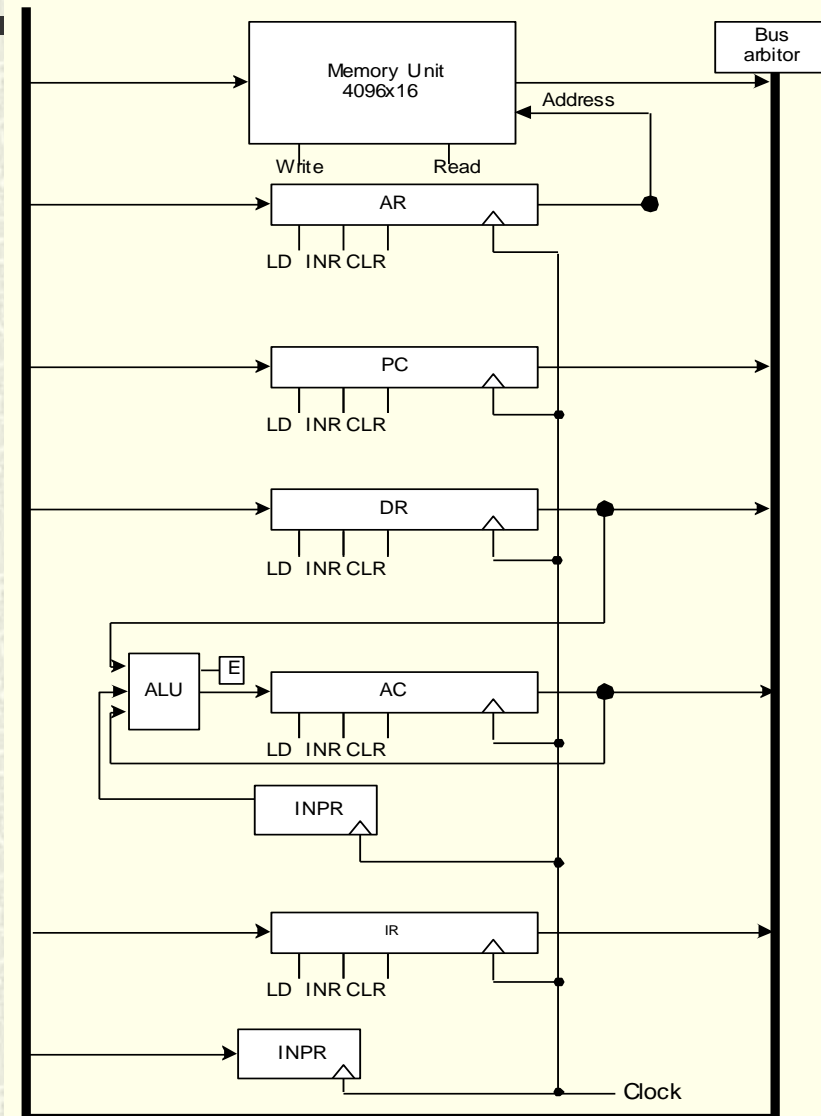
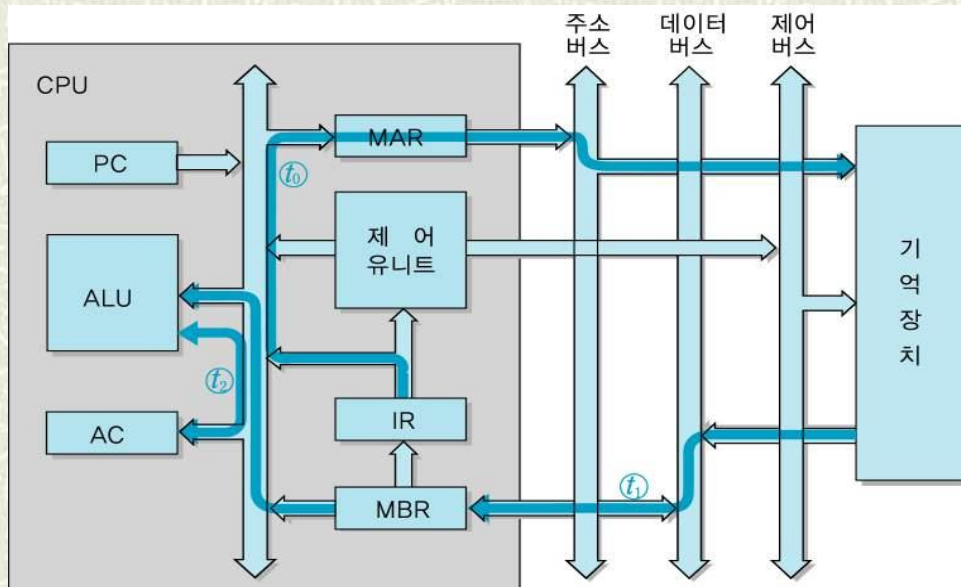
Operand(Addr)

ADD Y

실행 사이클 (데이터 처리)

#ADD 명령 실행

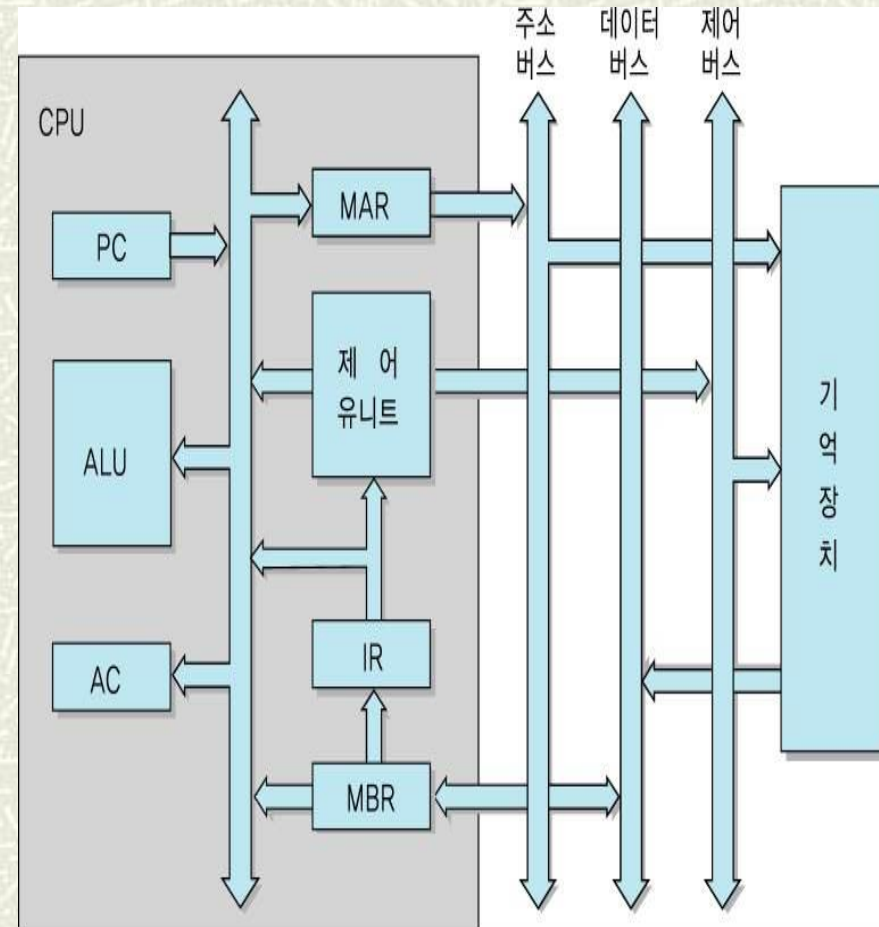
- $t_0 : \text{MAR} \leftarrow \text{IR}(\text{addr})$
- $t_1 : \text{MBR} \leftarrow \text{M}[\text{MAR}]$
- $t_2 : \text{AC} \leftarrow \text{AC} + \text{MBR}$



실행 사이클 (데이터 저장)

#STA 명령 실행

- $t_0 : \text{MAR} \leftarrow \text{IR}(\text{addr})$
- $t_1 : \text{MBR} \leftarrow \text{AC}$
- $t_2 : \text{M}[\text{MAR}] \leftarrow \text{MBR}$



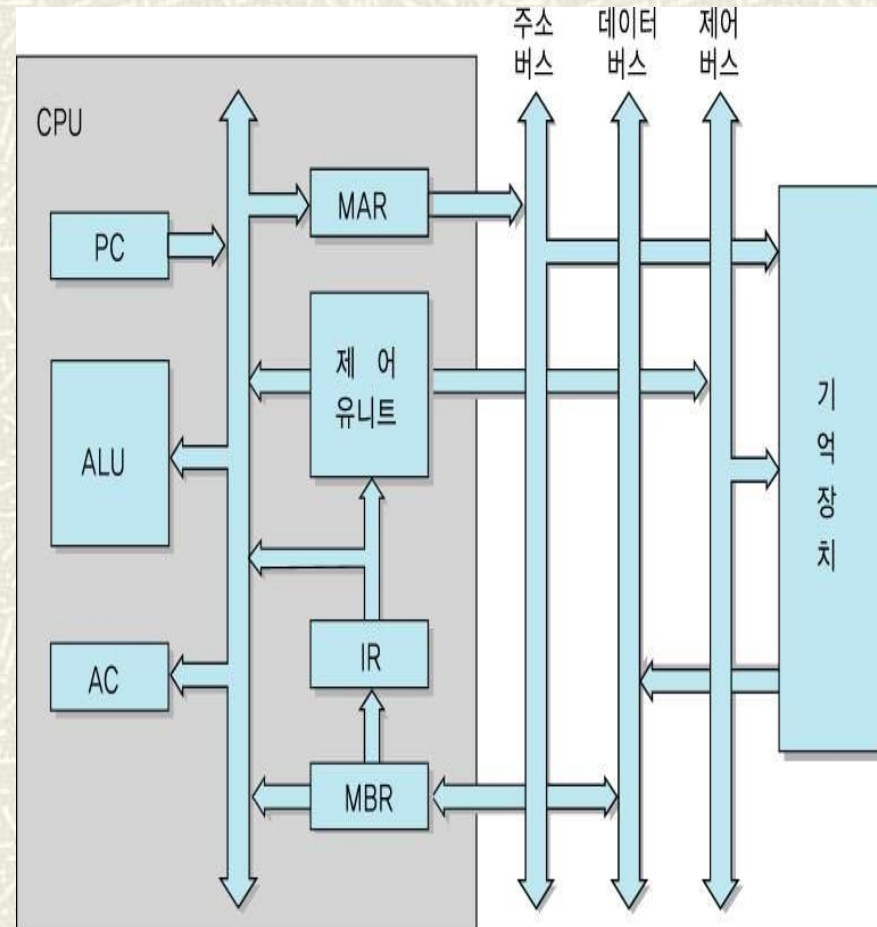
실행 사이클 (흐름제어 처리)

제어 명령

JUMP *addr* 명령

■ to : $PC \leftarrow IR(addr)$

이후 Fetch cycle에서
현재 PC에 저장된 주소
에 있는 명령어를 인출
하여 수행하게 된다.



정보처리기사 1급 기출 문제

46. 4096×16 용량을 가진 RAM이 있다. 메모리 버퍼 레지스터(MBR)는 몇 비트의 레지스터인가?

가. 8

나. 16

다. 32

라. 4096

47. 다음은 fetch cycle에서 일어나는 micro instruction이다. 시행 순서에 맞는 것은?

① $MAR \leftarrow PC$

② $MBR \leftarrow M$

③ $PC \leftarrow PC + 1$

④ $OPR \leftarrow MBR(OP)$

(단, MAR은 Memory Address Register이고, MBR은 Memory Buffer Register이며, PC는 Program Counter이고, OPR은 Operation Code Register이다.)

가. ② → ① → ③ → ④

나. ① → ② → ③ → ④

다. ② → ④ → ① → ③

라. ③ → ① → ② → ④

기사 1급 문제

12. 다음 마이크로오퍼레이션(micro operation)은 무엇을 수행하는 것인가?

MAR \leftarrow MBR(AD)
MBR \leftarrow M, AC \leftarrow 0
AC \leftarrow AC + MBR

- 가. STORE AC 나. LOAD TO AC
다. AND TO AC 라. ADD TO AC

13. 기억 장치가 1024word로 구성되고 각 word는 16bit로 이루어질 때 PC, MAR, MBR의 각 비트수를 바르게 나타낸 것은?

- 가. 16, 10, 10 나. 10, 10, 16
다. 10, 16, 16 라. 16, 16, 10

CPU에서의 명령어 실행 과정

예

주소	명령어		기계 코드
100	LOAD	250	1250
101	ADD	251	5251
102	STA	251	2251
103	JUMP	170	8170

- # LOAD 명령 : 1
- # STA 명령 : 2
- # ADD 명령 : 5
- # JUMP 명령 : 8