제 2장 컴퓨터동작의 기본 개념

컴퓨터동작의 기본 개념

1. 컴퓨터 연산 단위
   - 비트, 음수의 표현, 바이트, Word의 개념
   - 2진수 연산과 16진수 표현
2. 컴퓨터 조직
   - 8086 프로세서, 시스템 버스, 기억장치, I/O장치
3. 명령어 실행 개념
   - 판독과 기록사이클
4. 기억장치 관리

2.1 컴퓨터 연산단위

자료의 단위

* 비트(bit): binary digit
  * 0과 1로 표현되는 가장 작은 기억 단위
  * 2가지 정보 밖에 저장할 수 없는 비트가 컴퓨터 기억장치와 정보 조직의 기본
  * 2비트=4(2^2)가지, 3비트 =8(2^3)가지, n비트는 2^n의 값을 기록
* 니블(nibble) 4비트
  * 4비트로 16개의 정보를 저장
* 바이트(byte): 고유주소:의미를 줄 수 있는 기본 기억 단위로서 8 개의 비트로 구성
  * 1바이트에 2^8정보를 저장
  * 1바이트로 정수 표현한다면 0부터 255까지 표현할 수 있다
  * 이렇게 표현 할 수 있도록 각 문자에 번호를 붙여놓은것을 ASCII코드라 한다.
  * 예 : A →65 이진수로 나타낸 값은 1000001-41(16)

자료의 단위

* 문자(character)
  - ➢ 코드 : 비트들이 모여서 문자를 표현하는 방법
* 단어(word) : 주기억장치와 CPU 사이 전송되는 정보의 단위
  - : 컴퓨터가 자료를 처리하는 기본 단위
  - 인텔8086 : 16비트
  - 인텔 80386,80486프로세서 : 4바이트
  - 인텔펜티엄프로세서 : 8바이트
  - – 단어의크기는 컴퓨터 시스템의 버스(bus)와 레지스터와 밀접한 관련
* 항목(item): 필드, 레코드 구성의 논리적 자료 단위
* 레코드, 파일
* 블록: 보조기억장치와 컴퓨터내부 사이 전송되는 다수의 레코드의 묶음

진법 및 진법변환

* **진법**: 자리수를 정하는 기수법

  예: k진법은 0에서 k−1까지의 숫자로 모든 수를 나타냄
  k 진법으로 표현된 수를 k진수라 함.

* **소수점이 있는 경우**

  2를 곱한 정수부분을 오른쪽에 쓰고 소수부분에 다시 2를 곱한다. 소수부분이 없어질 때까지

* 0.375(10) = 0.011(2)
* 2 → 8, 16 → 10
* 11000.0101(2) → 30.24(8) → 24.3125(10)

진법 및 진법변환

1. $(1011)_2$
2. $(123)_8$
3. $(AF)_{16}$
4. $(10010.11011)_2$
5. $(22.66)_8$
6. $(12.D8)_{16}$

진법 및 진법변환

1. $(1011)_2 = 1*2^3+0*2^2+1*2^1+1*2^0$
2. $(123)_8 = 1*8^2+2*8^1+3*8^0$
3. $(AF)_{16} = 10*16^1+15*16^0$

진법 및 진법변환

4. $(10010.11011)_2 = 1*2^4 + 1*2^1 + 1*2^{-1} + 1*2^{-2} + 1*2^{-4} + 1*2^{-5}$

5. $(22.66)_8 = 2*8^1 + 2*8^0 + 6*8^{-1} + 6*8^{-2}$

6. $(12.D8)_{16} = 1*16^1 + 2*16^0 + D*16^{-1} + 8*16^{-2}$

정수의 표현

* 가장 왼쪽의 한 비트를 부호비트로 한다.
* <u>음의 정수를 나타내는 방법</u>
  - **부호 절대값의 방법** : $-(2^{n-1}-1) \sim (2^{n-1}-1)$
  - **1의 보수 방법** : $-(2^{n-1}-1) \sim (2^{n-1}-1)$
  - **2의 보수 방법** : $-(2^{n-1}) \sim (2^{n-1}-1)$

음수의 표현

* r진수에서 r-1의 보수
  * r-1에서 각 자리 숫자를 뺀다
  * 2진수 0011에 대한 1의 보수
    1111-0011=1100
  * $-(2^{n-1}-1) \sim (2^{n-1}-1)$
* r진수에서 r의 보수
  * r-1의 보수에 1을 더함
  * 2진수 0011에 대한 2의 보수
    1100+1=1101
  * $-(2^{n-1}) \sim (2^{n-1}-1)$

2의 보수 방법의 장점

* 계산 결과가 동일한 2의 보수
* 뺄셈이 덧셈으로 계산
    * *cpu의 덧셈에 대한 논리회로가 뺄셈도 수행 가능

예 제

* 2 − 1 = 1

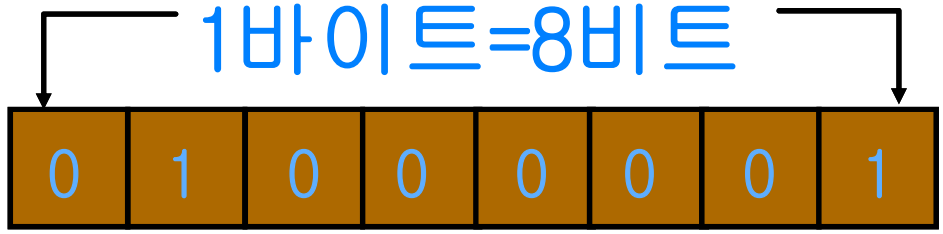
부호 절대치 방법	1의 보수 방법	2의 보수 방법
00000010 (2)	00000010 (2)	00000010 (2)
10000001 (−1)	11111110 (−1)	11111111 (−1)
10000011 (−3)	00000000 (0)	00000001 (1)

4비트에 의한 정수표

2진수	2의보수	1의보수	양수
0111	7	7	7
0110	6	6	6
0101	5	5	5
0100	4	4	4
0011	3	3	3
0010	2	2	2
0001	1	1	1
0000	0	0	0
1111	-1	0	15
1110	-2	-1	14
1101	-3	-2	13
1100	-4	-3	12
1011	-5	-4	11
1010	-6	-5	10
1001	-7	-6	9
1000	-8	-7	8

바이트의 개념

* 의미를 주는 기억단위
* **8개의 비트로 구성**(0~255까지 표현)
* ASCII 코드: 각 문자에 번호를 부쳐 놓은 것



'A'=$65_{(10)}$=$01000001_{(2)}$=$41_{(16)}$

8비트에 의한 정수표

2진수	정수 10진수	문자형(양수)	ASCII 문자
0111 1111	127	127	DEL
0111 1110	126	126	~
… …	……	……	……
… …	……	……	……
0100 0001	65	65	A
… …	……	……	……
… …	……	……	……
0000 0001	1	1	
0000 0000	0	0	Null
1111 1111	-1	255	
… …	……	……	……
… …	……	……	……
1000 0000	-128	128	……

단어(Word)의 개념

* 자료를 처리하는 기본단위
* 컴퓨터에 따라 달리 정의
* 1단어 = 8,16,32,64비트
  * Apple,인텔8086, 인텔80386/486,펜티엄 프로세서 이후
  * 중앙처리장치가 기억장치에서 한번 읽을때 8,16,32비트로 자료를 읽기 때문에 자료를 빨리 처리 하는 이유가 단어의 크기에 관계
* BUS나 register와 밀접한 관계

비트,니블,바이트,단어,긴단어

비트 = 1비트, 표현범위 = 0~1

0

니블 = 4비트, 표현범위 = 0 ~ 15

3 0

바이트 = 8비트, 표현범위 = 0 ~ 255

7 4 3 0

단어 = 16비트, 표현범위 = 0 ~ 65535

15 좌측니블 8 7 우측니블 0

긴단어 = 32비트, 표현범위 = 0 ~ 4294967295

31 좌측 단어 15 우측 단어 0

2진수의 연산규칙

* 1 + 1 = 10
* 1 + 0 = 1
* 1 * 0 = 0
* 1 * 1 = 1
* 1 − 1 = 0
* 10 − 1 = 1

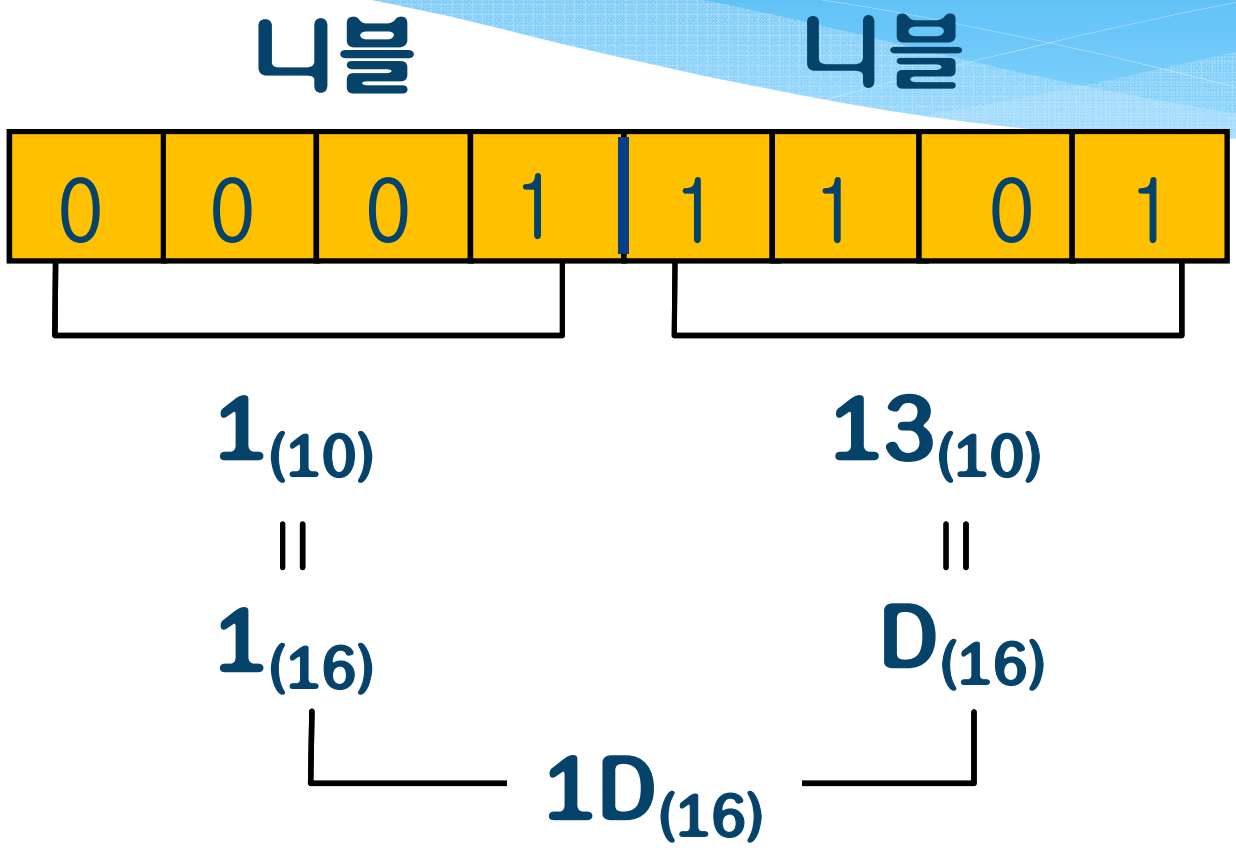
2진수 연산의 예제

$10111 = 2^4 + 0 + 2^2 + 2^1 + 2^0 = 23$

$+)\ 11101 = 2^4 + 2^3 + 2^2 + 0 + 2^0 = 29$

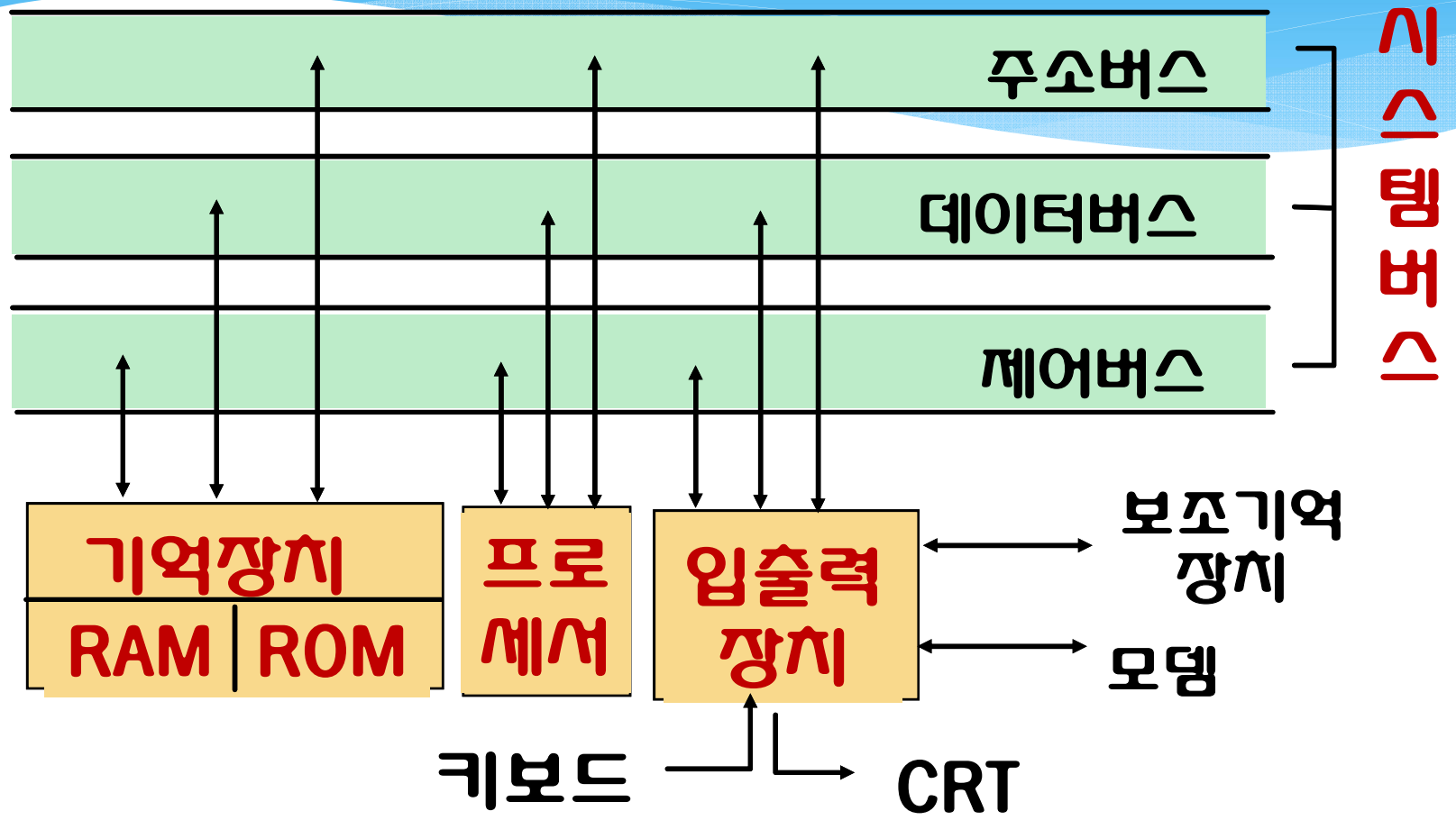
$110100 = 2^5 + 2^4 + 0 + 2^2 + 0 + 0 + = 52$

데이터의 16진수 표현



2.2 컴퓨터 조직

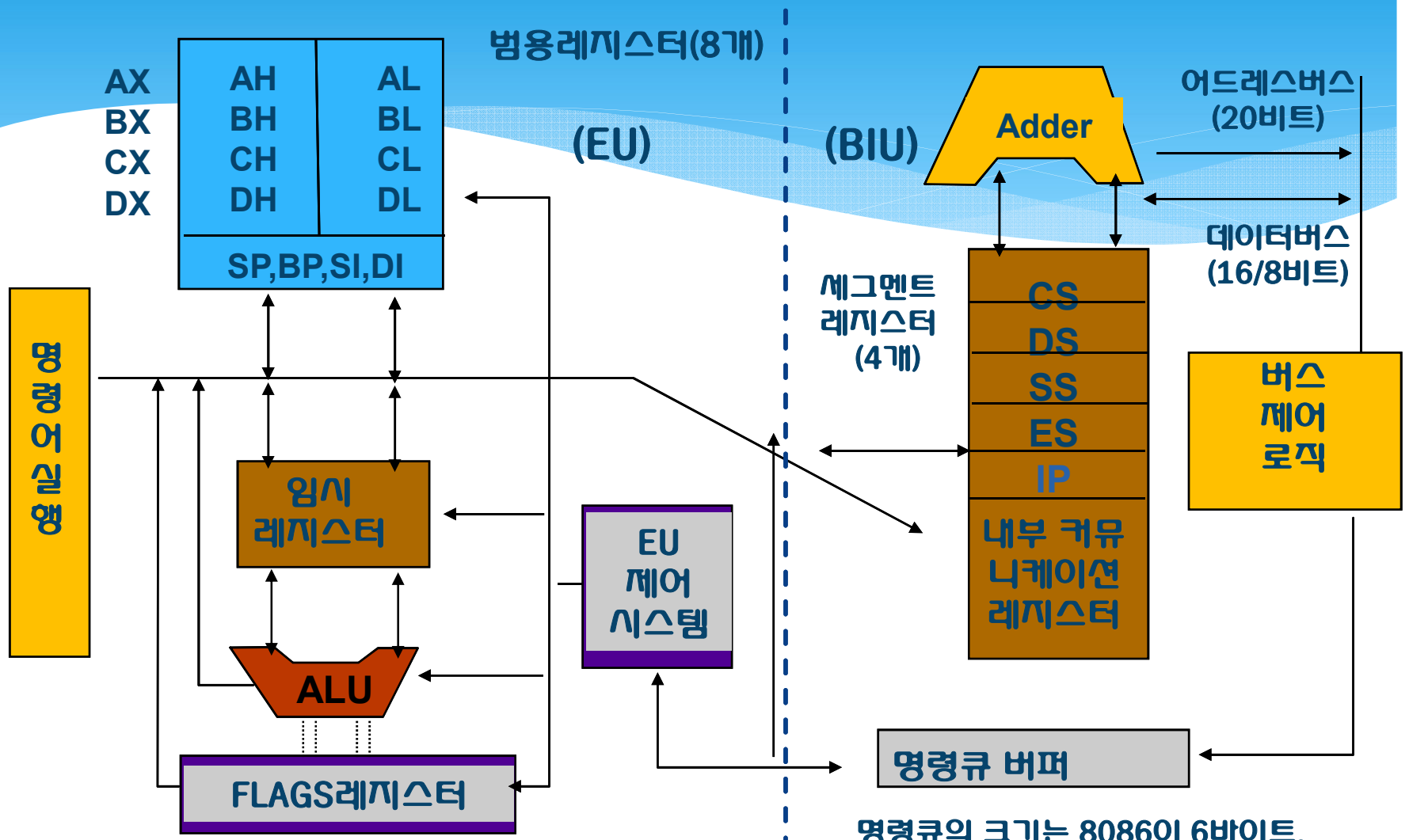
컴퓨터의 조직


주소버스
데이터버스
제어버스
시스템버스
기억장치
RAM
ROM
프로세서
입출력장치
보조기억장치
모뎀
키보드
CRT

프로세서

* 명령어 실행 장치(EU:Execution unit)
  * 명령이 실행
  * 산술 논리 연산장치, 범용 레지스터등
* 버스 인터페이스(BIU : bus interface unit)
  * 명령어 페치
  * 기억 장치나 입출력장치로 부터 데이터를 읽어 오거나 출력을 행함
  * 내부 통신을 위한 세그먼트 레지스터
  * 유효 주소를 계산하기 위한 가산기
  * 명령어 큐
    * 실행할 명령어를 기억 장치로 부터 가져와 임시로 보관하는 버퍼
* 내부버스
  * 프로세서 내부에는 각 부분 사이의 자료를 전달해주는

마이크로프로세서 내부구조



연산 결과가 이 플래그 레지스터에 반영된다.

명령큐의 크기는 8086이 6바이트, 8088이 4바이트를 미리 읽어 명령 코드를 쌓아둔다.

레지스터

* 자료의 임시 저장소
  * 신속히 이용하기 위한 일종의 기억 장소
* 범용레지스터, 세그먼트레지스터, 명령어 포인터, 인덱스 레지스터
* 8086/8088 : 16비트 레지스터 14개
* 펜티엄 프로세스 :

  32 비트 레지스터 10개 +

  16비트 세그먼트 레지스터 6개

산술 논리 연산 장치(ALU)

* 데이터를 처리하기 위하여
  * 사칙 연산과
  * 논리 연산 등을 처리하는 장치
* 데이터를 원활히 처리하기 위하여 사용되는 레지스터의 일종
  * 누산기
  * 계산기
* 2진수로 숫자를 처리하며 여러 개의 비트로 되어 있음
  * 연산장치가 8비트로 되어 있으면 2^8범위내의숫자는
  * 한 시스템 사이클 시간에 처리 할 수 있다.

범용 레지스터

32비트	16비트	8비트 (상위)	8비트 (하위)
EAX	AX	AH	AL
EBX	BX	BH	BL
ECX	CX	CH	CL
EDX	DX	DH	DL

A : Accumulator (연산 레지스터로서 연산의 결과나 중간값등을 저장하는데 쓰인다.)
<보통 CPU가 처리해야 할 코드/데이터를 지정>
B : Base (베이스 어드레스 지정에 쓰인다. 간접 어드레스 지정시에 어드레스 레지스터, 트랜슬레이터 명령에 있어서 변환 테이블, 베이스레지스터로써 사용한다.)
<보통 메모리중의 특정 블럭의 시작 주소를 지정>
C : Counter (반복 실행문의 횟수를 지정할 때 쓰인다.)
<보통처리해야 할 회수를 지정>
D : Data (어큐레이터의 보조로 활용되거나 간접 어드레스에 의한 입출력 명령시 어드레스 지정에 사용된다. 곱셈, 나눗셈 작업을 할 때 사용된다.)
<보통 처리할 데이터의 시작 주소를 지정>

16비트 시절에는 AX,BX, CX, DX 라는 이름을 가지고 있다가, 386 CPU부터 레지스터의 크기가 32비트로 확장되면서 모두 이름앞에 E(Extended)가 붙여졌다. 16비트 시절의 레지스터들은 8비트 크기로 나누어서 접근이 가능했다. AX의 경우 상위 8비트가 AH, 하위 8비트가 AL이라는 이름을 가진다.
(H : High, L : Low)

세그멘트 레지스터

CS(code segment) :CPU가 실행할 명령 코드가 저장되어 있는 부분의 시작 어드레스를 가리킨다.

DS(data segment) :프로그램에서 사용할 데이터가 놓여 있는 부분의 시작 어드레스를 가리킨다.

SS(stack segment) :주로 스택 조작에서 데이터를 처리하는 부분이며, 이 부분의 시작 어드레스를 가리킨다.

ES(extra segment) :주로 스트링 명령을 실행할 경우에 보조로 사용하는 부분이며, 이 부분의 시작 어드레스를 가리킨다

인덱스 레지스터

인덱스 레지스터는 여러개의 메모리 블록을 한꺼번에 이동하거나 가르칠때 사용됩니다. 일반적으로 번지를 간접 지정하는 레지스터라고 합니다.

인덱스 레지스터는 16비트일 경우 SI,DI 로 명명하고, 32비트일 경우 ESI,EDI 로 명명합니다. SI 는 Source Index의 약자이며 DI 는 Destination Index의 약자입니다. 이 두개의 레지스터에 메모리 주소를 설정하고 나면 해당 메모리를 이동시키거나 복사하는 명령으로 데이터를 이동 및 복사를 할 수가 있습니다.

시스템 버스

* 프로세스,기억장치,입출력장치는 버스라 불리우는 신호선을 통하여 결합
* 주소 버스, 데이터 버스, 제어 버스
* 주소 버스 : 기억용량의 크기 결정
  * 16비트이면 한 사이크클시간에 2^{16}가지 주소 전달,65535,64kb 기억용량
* 데이터 버스의 크기
  * 기억장치에서 한 사이클 시간에 읽어올 수 있는 데이터 양 결정

시스템 버스

* 제어버스 :
* 제어버스는 CPU가 현재 무엇을 하고 있는가를 메모리나 입출력 기에 알림
* 또 CPU가 어떤 상태이어야 하는가를 알리기 위한 제어 신호를 싣는데 사용하는 전송로이며 수개로 구성
* 단방향성의 기능
* 버스방식은 입출력기기나 메모리를 시스템 버스에 접속하는형을 채용하므로 회로가 매우 단순
* 주소 신호는 메모리와 입출력 포트에 공용으로 사용

버스

* 여러 개의 신호선 으로 구성
* n개로 구성되어 있으면 버스의 크기를 n비트라 부름
* N비트 버스는 => 한 사이클 시간에 2^n가지의 정보를 전달
* 예 : 데이터 버스가 8비트로 구성되어 있으면 한 사이클 시간에
* 2^8가지의 정보를 전달,16비트는 2^16가지 주소 전달
* 이때 주소는 매 바이트 마다 붙이기 때문에 이 컴퓨터는
* 64k 바이트의 기억용량을 가질 수 있다.

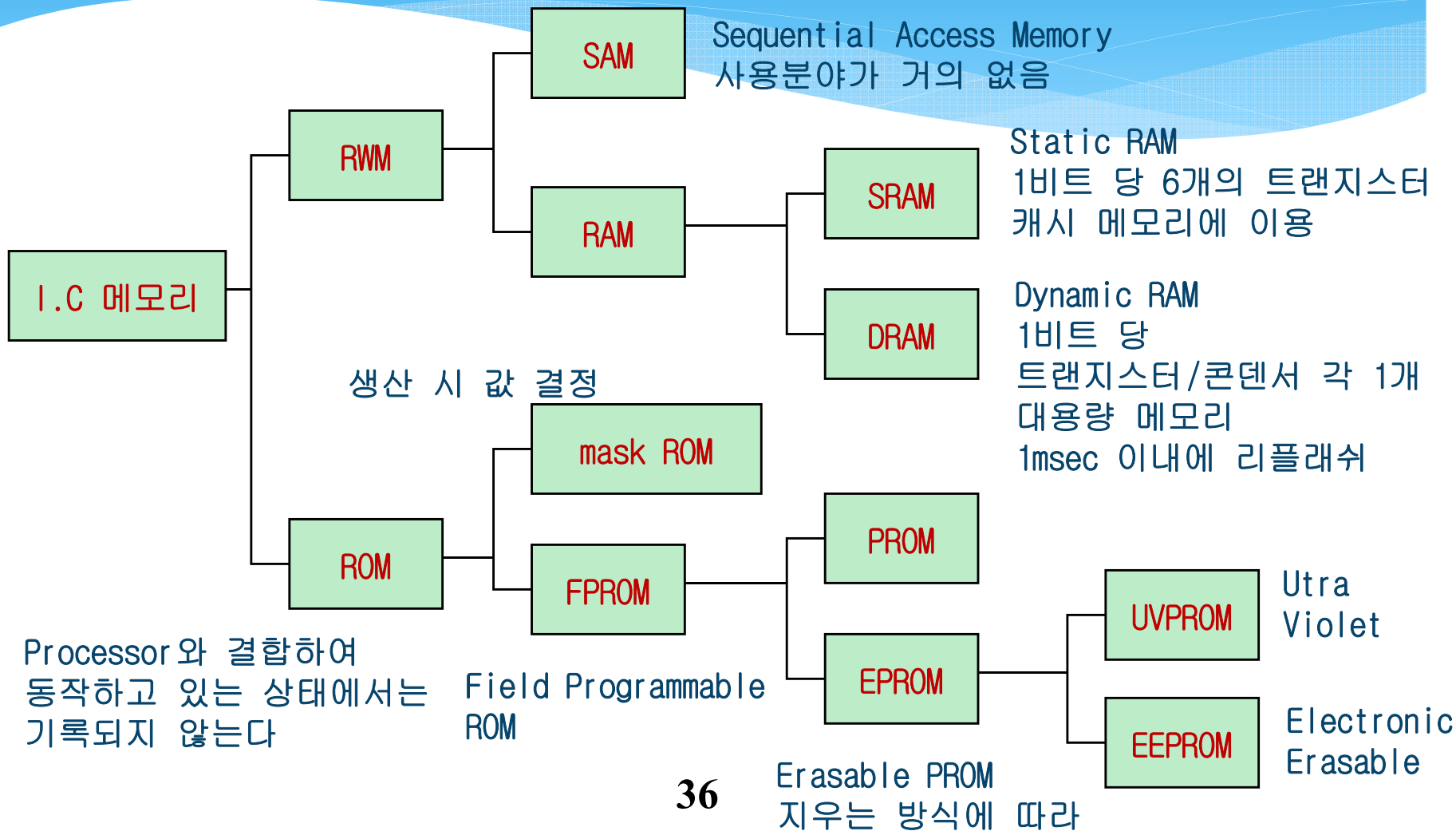
컴퓨터 성능 결정 요소

* 레지스터의 크기
* 산술 논리 연산장치의 크기
  * 한 사이클 시간에 CPU에 자료를 저장하고 연산할 수 있는 정보의 양을 결정
* 데이터 버스 크기
  * 기억 장치에서 한 사이클 시간에 읽어 올 수 있는 데이터 양을 결정
* 주소버스 크기
  * 기억장치의 크기

기억 장치

* RAM과ROM 으로 구성
  * RAM : 데이터를 기록하고 읽을 수 있는 기억장치
  * ROM : 새로이 기록하지 못하고 읽을 수만 있는 기억장치
* 주기억 장치:
  * 프로그램이나 데이터를 잠시 보관하기 위한 기억 장소
* 보조기억장치 :
  * 데이터나 프로그램을 장기적으로 보관할 때는 값이 싼 보조 장치를 이용
* 기억장치는 전자적인 소자로 구성되어 있어 자료가 전기적인 신호 펄스형태로 저장

주기억장치


I.C 메모리
RWM
SAM
Sequential Access Memory
사용분야가 거의 없음
RAM
SRAM
Static RAM
1비트 당 6개의 트랜지스터
캐시 메모리에 이용
DRAM
Dynamic RAM
1비트 당
트랜지스터/콘덴서 각 1개
대용량 메모리
1msec 이내에 리플래쉬
생산 시 값 결정
ROM
mask ROM
FPROM
PROM
EPROM
UVPROM
Utra
Violet
EEPROM
Electronic
Erasable
Processor와 결합하여
동작하고 있는 상태에서는
기록되지 않는다
Field Programmable
ROM
Erasable PROM
지우는 방식에 따라

2.3 명령어 실행 개념

컴퓨터 기본 동작

* 판독 (Read)
  * 프로세서가 자료를 기억장치에서 읽는 것

* 기록 (Write)
  * 프로세서가 자료를 기억장치에 저장하는 것

판독 사이클

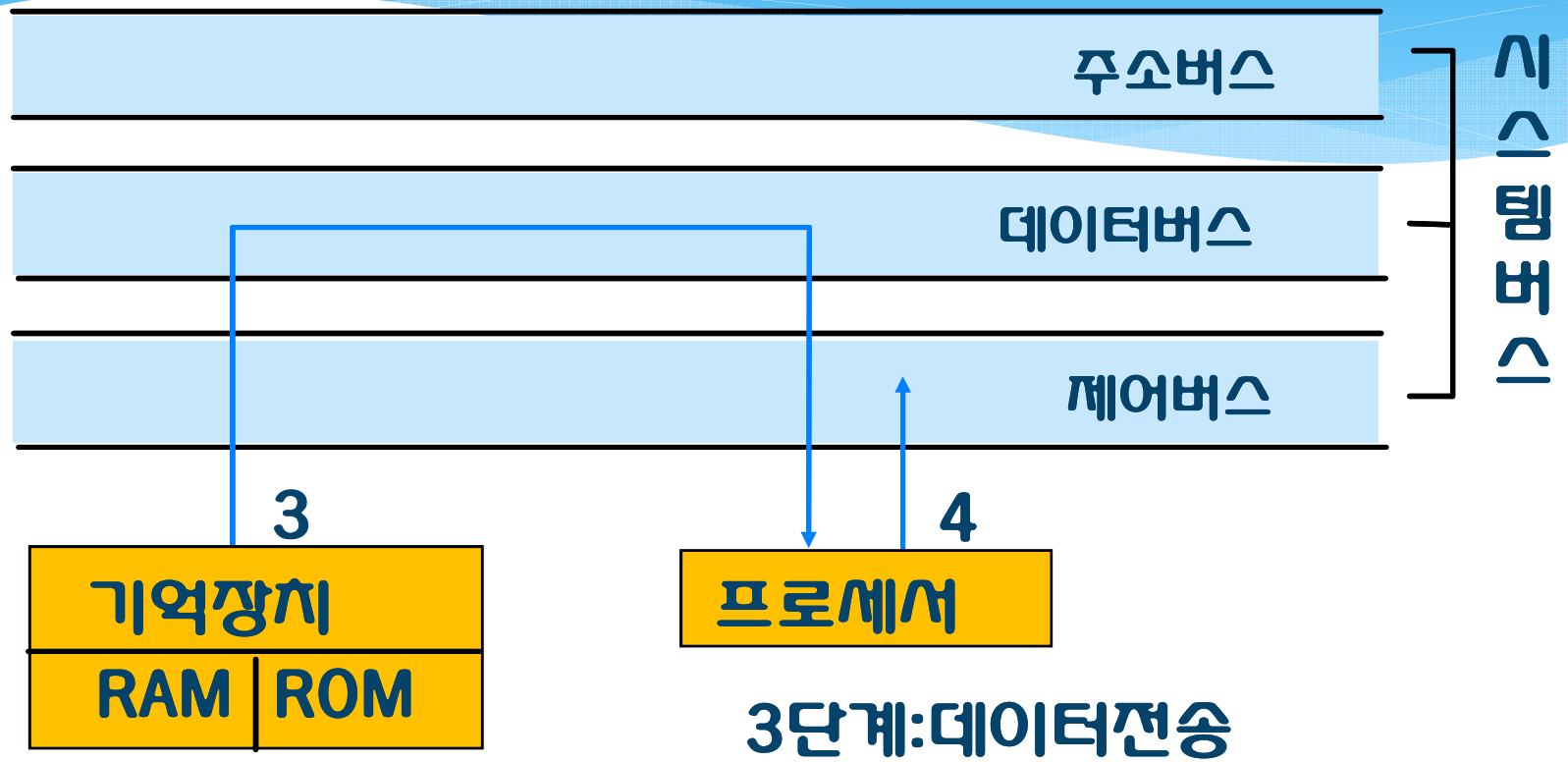
* 판독 요구 신호(Read Request)
  * 제어버스를 통해 버스 사용을 요구
* 주소전송(send address)
  * 판독할 자료가 있는 기억 소자의 주소를 주소 버스에 준다
* 데이터 수신(Receive data)
  * 기억장치에서 전송되어온 데이터를 프로세서에서 받는다
* 판독완료 신호(signal ok)
  * 프로세서가 데이터를 받으면 버스 사용이 끝났다는 신호를 제어 버스에 보냄

컴퓨터 동작의 판독사이클

1단계:판독요구신호

2단계:주소전송

컴퓨터 동작의 판독사이클



3단계:데이터전송

4단계:판독완료신호

기록 사이클

* 기록 요구 신호(Write Request)
  * 제어버스에 버스 사용 신호를 보냄
* 주소전송(send address)
  * 버스를 이용 할 수 있게 되면 주소를 통하여 데이터 가 기록될 주소를 보낸다
* 데이터 전송(data send)
  * 데이터 버스를 통하여 기록할 데이터 보낸다.
* 기록 완료 신호(signal ok)
  * 기록을 완료하면 기록 작업이 끝났다는 신호를 제어 버스에 보낸다.

컴퓨터 동작의 기록사이클

1단계:기록요구신호

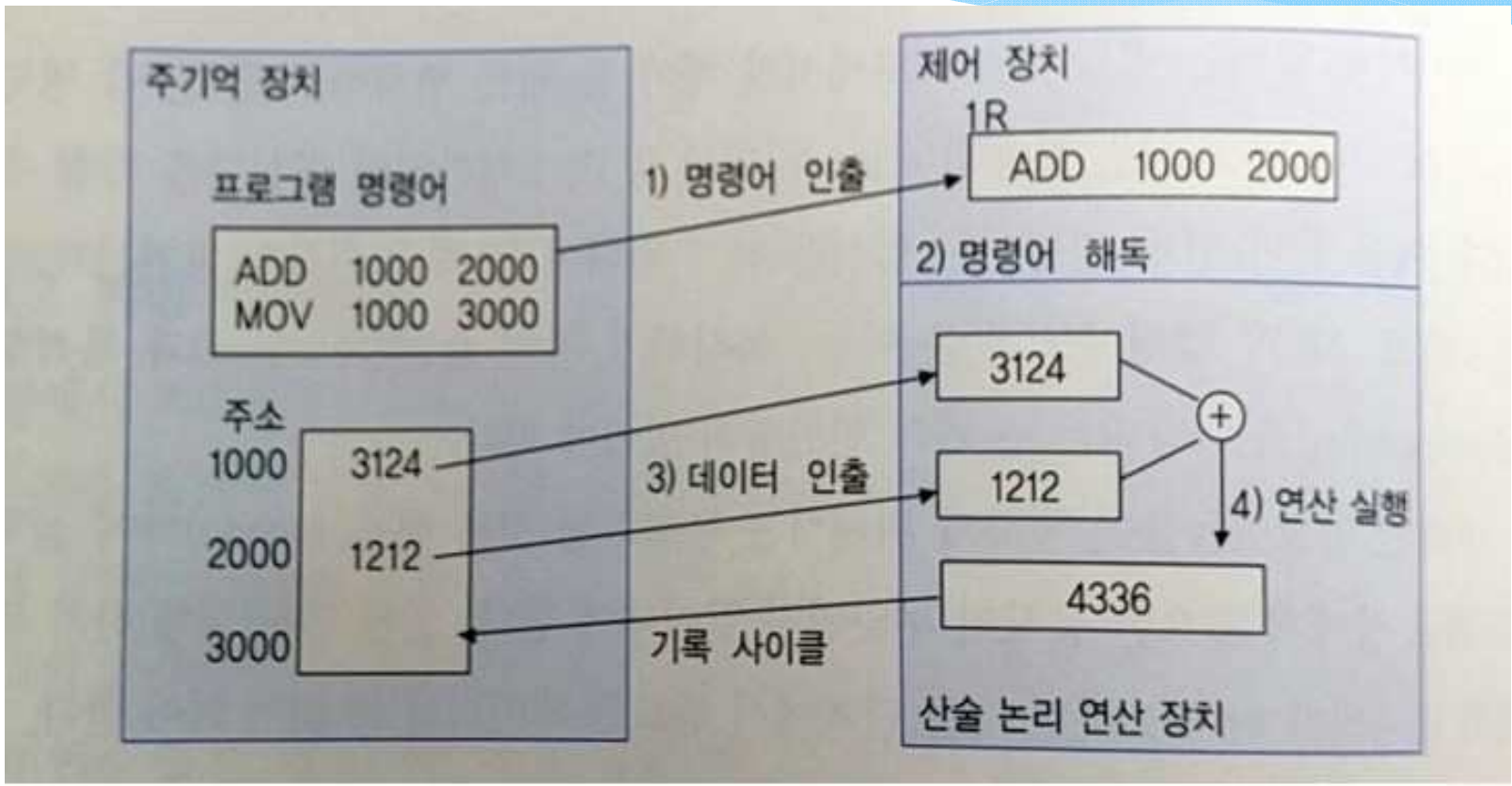
2단계:주소전송

컴퓨터 동작의 기록사이클

3단계:데이터전송

4단계:기록완료신호

명령어의 실행 단계


주기억 장치
프로그램 명령어
ADD 1000 2000
MOV 1000 3000
주소
1000
2000
3000
3124
1212
1) 명령어 인출
3) 데이터 인출
기록 사이클
제어 장치
IR
ADD 1000 2000
2) 명령어 해독
3124
1212
4) 연산 실행
4336
산술 논리 연산 장치

명령어의 실행 단계

* 명령어 인출(instruction fetch)
  * IP가 기억한 주소를 주소 버퍼를 통하여 주소버스에 줌
  * 데이터 버스를 통하여 기억장치에서 명령어를 읽어옴
  * 읽어온 명령어를 내부버스를 통하여 버스 인터페이스 장치안에 있는 명령어 큐에 저장
  * 명령어 를 읽어 오면 IP주소를 증가시키는데 이것은 다음 명령어가 기억된곳을 가리킴
  * 일반적으로 주소는 단어 단위로 증가

명령어의 실행 단계

* 명령어 해독(instruction decode)
  * 명령어 큐에 기억해 놓은 명령어를 해독하기 위하여
  * 명령어 실행 장치에 보낸다.
  * 연산될 자료가 저장된 레지스터 번호나 기억 장소의 주소를 연산항에 주는 경우가 많기 때문

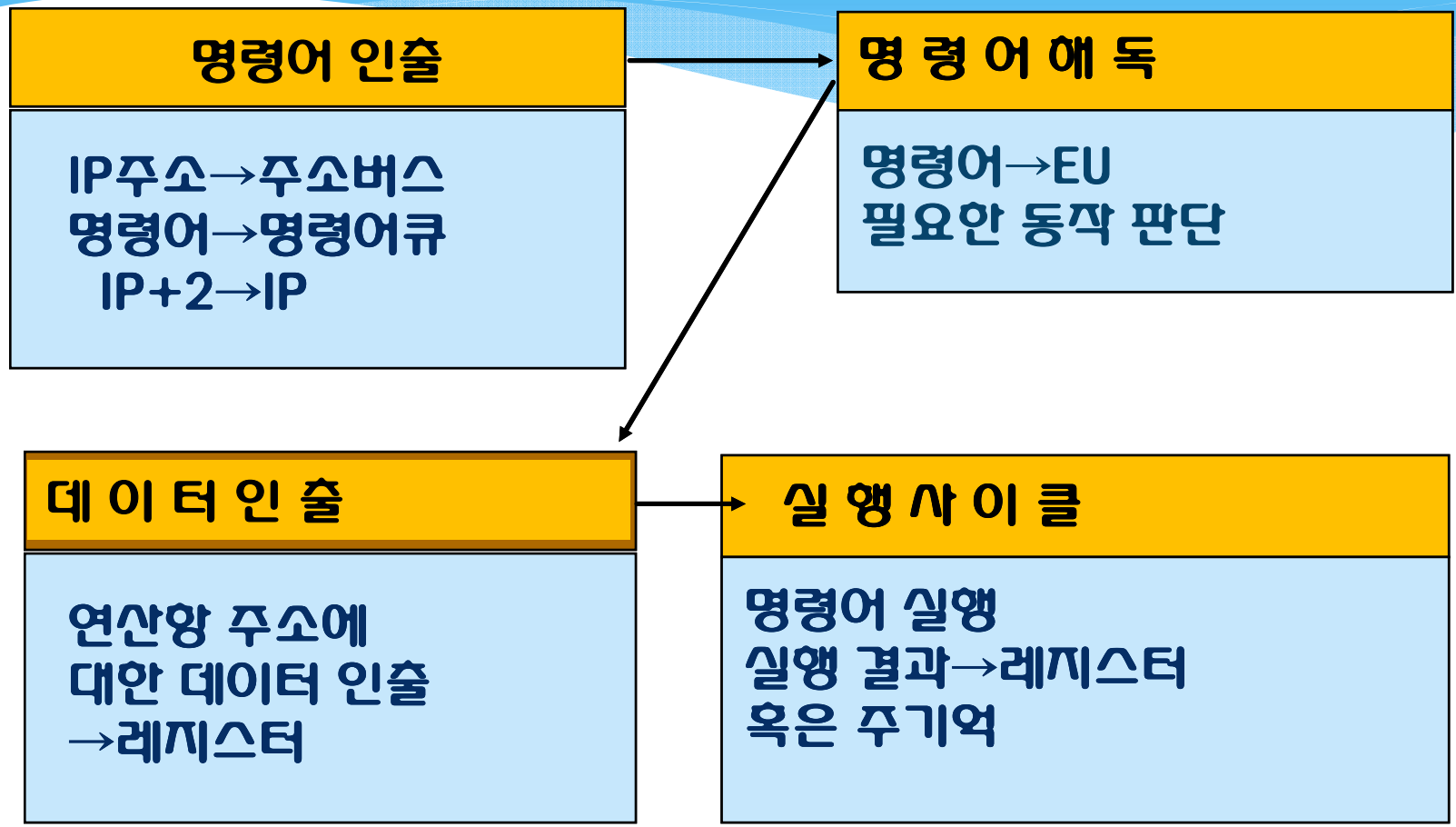
명령어의 실행 단계

* 데이터 인출(data fetch)
  * 해독된 명령어의 연산항에 주소가 주어져 있으면
  * 이 주소에 기록된 명령어를 읽어와야한다
  * 판독 사이클에 따라 읽어온다.
  * 읽어온 명령어 는 내부 버스를 통하여 레지스터에 저장된다

명령어의 실행 단계

* 실행 사이클(Execution cycle)
  * 프로세서가 명령어를 실행할 준비가 되어 있다.
  * 무엇을 가지고 (데이터 인출)
  * 무엇에 의해서(산술 논리 연산장치)
  * 무엇을 할 것인가(명령어 해독)하는 것을 모두 알았기 때문이다.
  * 연산을 실행하면 연산결과는 내부 버스를 통하여 레지스터에 저장되든지
  * 기억장치에 보내기 위해서 데이터 버퍼에 잠시 보관된다.
  * 후자의 경우에는 저장될 주소를 명시하여 기록 사이클을 시작한다.

명령어의 실행 단계


명령어 인출
IP주소→주소버스
명령어→명령어큐
IP+2→IP
명 령 어 해 독
명령어→EU
필요한 동작 판단
데 이 터 인 출
연산항 주소에
대한 데이터 인출
→레지스터
실 행 사 이 클
명령어 실행
실행 결과→레지스터
혹은 주기억

명령어 실행속도

* 한 명령어가 실행되는 시간은 명령어 호출, 명령어 해독, 데이타 호출, 실행 시간을 합
* fetch time(I-time) : 명령어 호출 시간
* execution time(E-time) : 명령어 해석하여 데이터 인출하고 연산 수행하는 시간
* instruction cycle = I-time + E-time

* MIPS(Million Instruction Per Second) 결정요소 : 사이클시간, 데이타버스크기, 레지스터크기, 논리연산장치크기, 명령어크기

MIPS란?

MIPS(Microprocessor without Interlocked Pipe Stage)

의미 : 1초당 1,000,000번의 연산 능력

용도 : 컴퓨터의 계산 속도를 나타내는 단위

컴퓨터의 계산 속도를 나타내는 단위로서 성능을 평가하는 지표로 사용된다. 일반적인 PC보다는 슈퍼컴퓨터와 같은 중대형 컴퓨터에서 주로 사용되며, 1mips는 1초에 100만 번의 계산을 의미한다. 컴퓨터 시스템에서 1초간 실행할 수 있는 최대 명령어의 개수를 체크한 후, 이 값을 100만으로 나누면 해당 컴퓨터의 성능이 몇 밉스인지 알 수 있다.

밉스 칩(MIPS Chip)

스탠퍼드대학에서 1982년경 개발된 축소 명령 집합형 컴퓨터(RISC) 마이크로프로세서의 이름. 이 프로세서는 초창기의 RISC 칩이며, 이후 밉스컴퓨터시스템즈사가 R2000, R3000, R4000 등으로 이름으로 상품화해 여러 종류의 공학용 워크스테이션에 탑재되었다. MIPS 프로세서는 RISC 구조를 가지고 있으며, 고가의 워크스테이션에 주로 사용됨은 물론, 가정용 게임기나 HPC 등에서도 활발히 사용되고 있다.

밉스[MIPS (million instructions per second)]란?

밉스는 프로세서의 성능을 나타내는 단위로 초당 몇 백만개의 명령어를 처리할 수 있는지를 나타내는 수치이다. 예를 들어 50 MIPS 성능을 갖는 프로세서가 있다면 초당 5천만개의 명령어를 처리할 수 있음을 뜻한다.

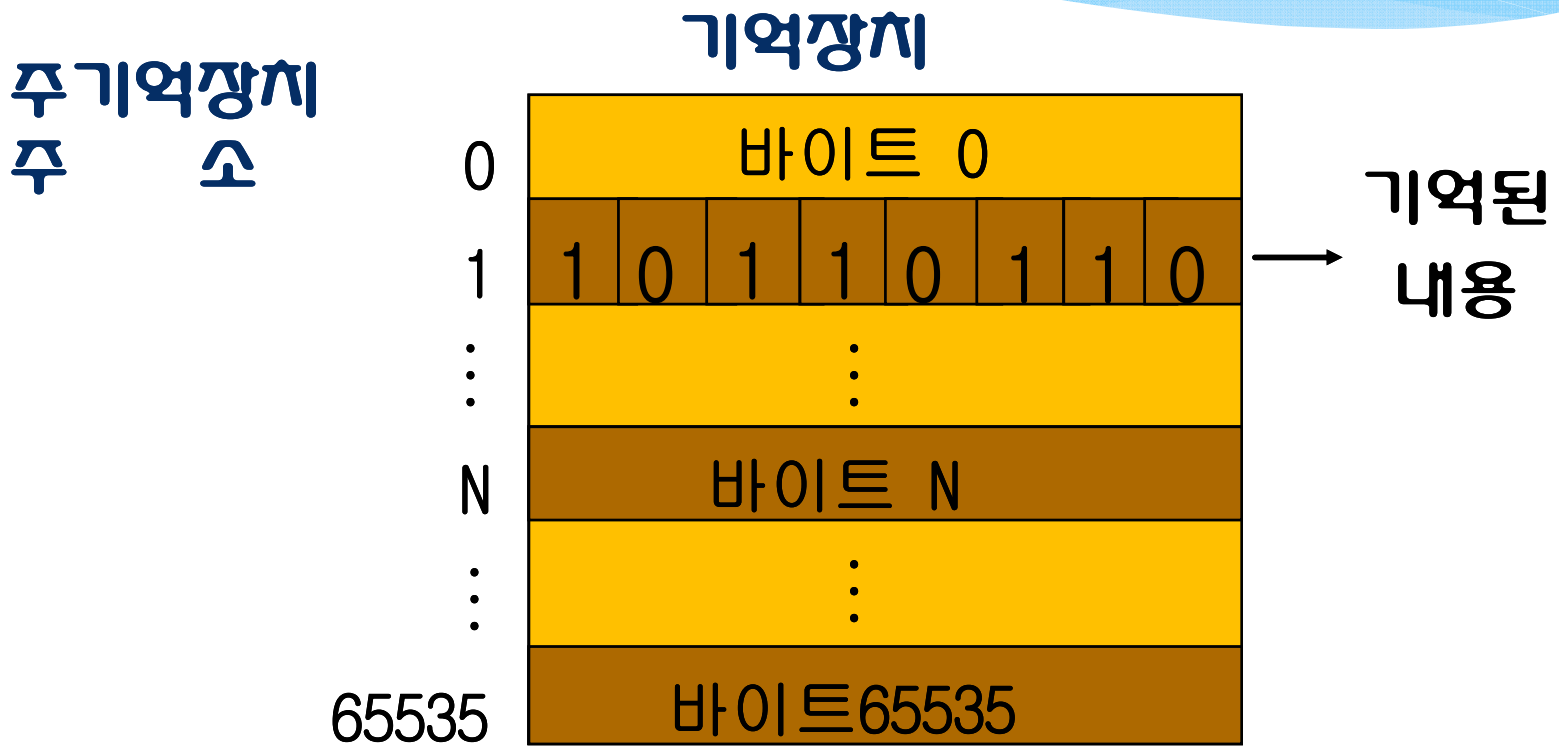
2.4 기억장치 관리

기억장치의 주소

* 주기억 장치
  * 바로 실행할 프로그램이나 데이터를 저장
* 보조 기억장치
  * 바로 필요하지 않는 프로그램이나 데이터를 저장
  * 기억 장치에 자료를 기억하고 나중에 읽어 오기 위하여 기억 장소에 주소를 붙임
  * 이 주소는 각 바이트 마다 주어짐
* 다음 그림은 기억 장소에 주소를 붙이는 방법
* 그 속에 저장되는 자료의 모양을 보여줌

기억장치의 주소

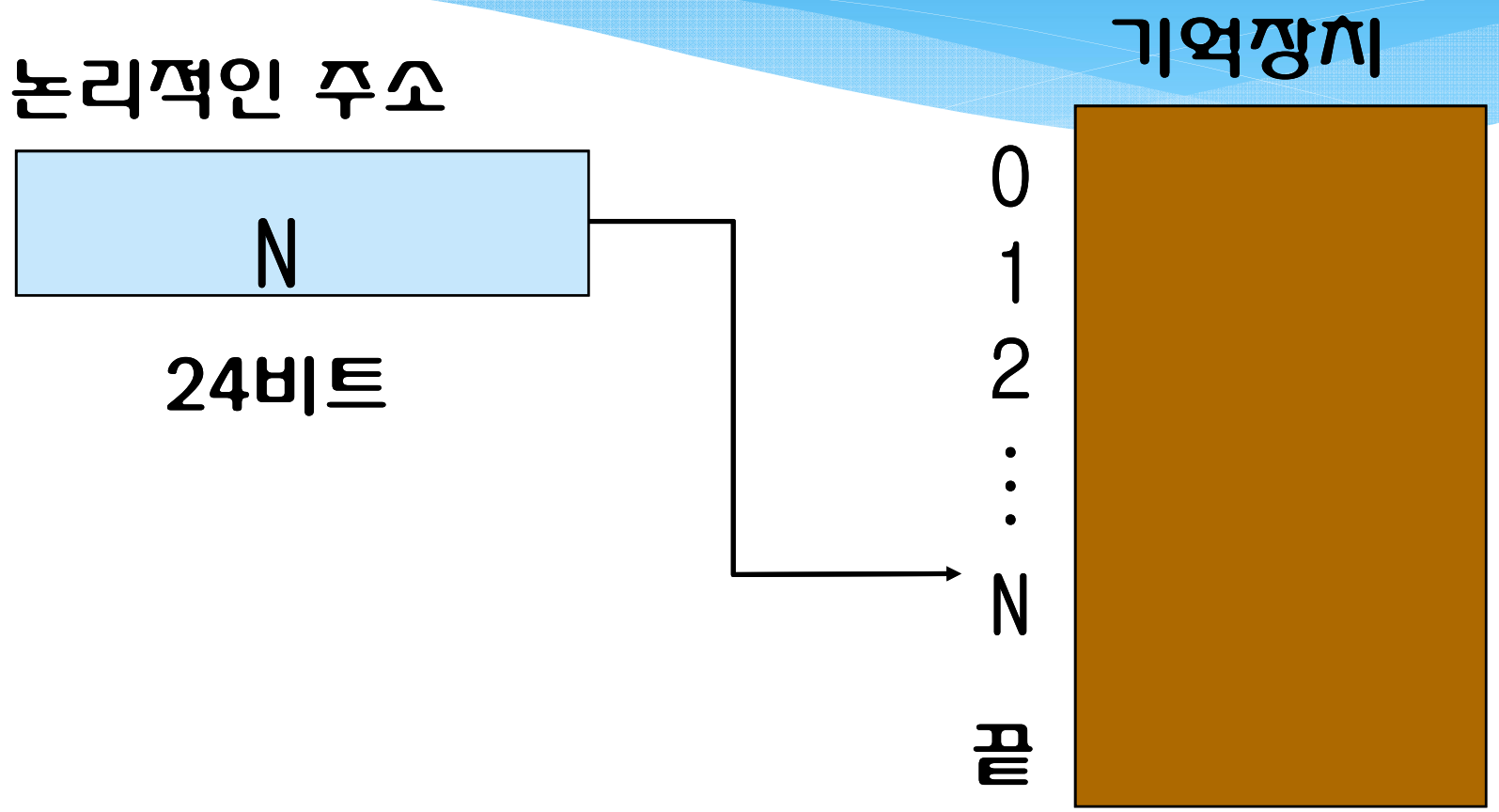
* 64KB의 기억장치 주소 예



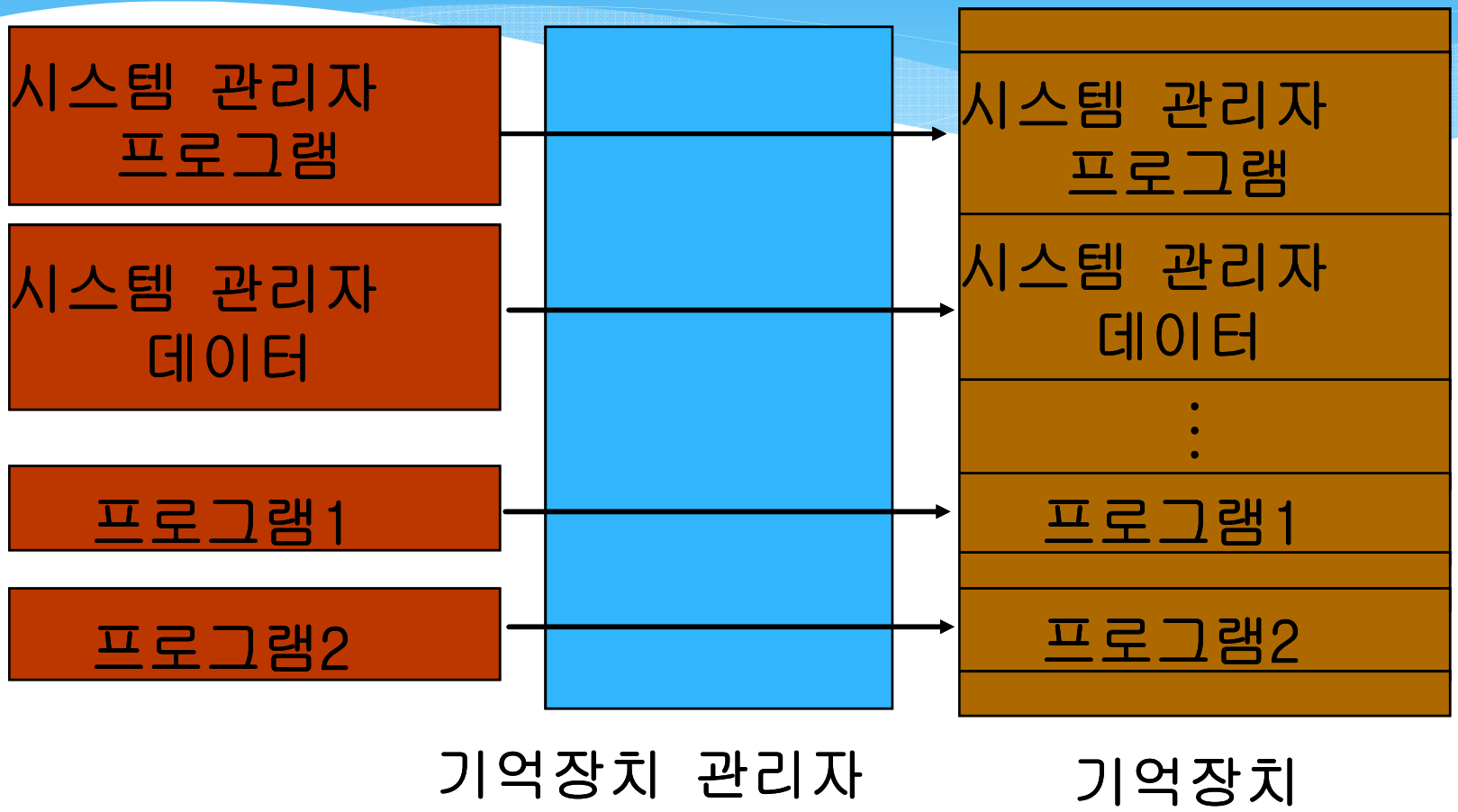
기억장치 관리방법

* 선형 기억장치 관리
  * 주소를 선형으로 배열
  * 논리주소 = 물리주소(기계적 주소)
  * M68000 프로세서 등
* 세그먼트 기억장치 관리
  * 기억장소를 세그먼트로 구분
  * 논리주소≠물리주소
  * 물리주소=논리주소+세그먼트번호
  * 인텔 프로세서, 대형 컴퓨터

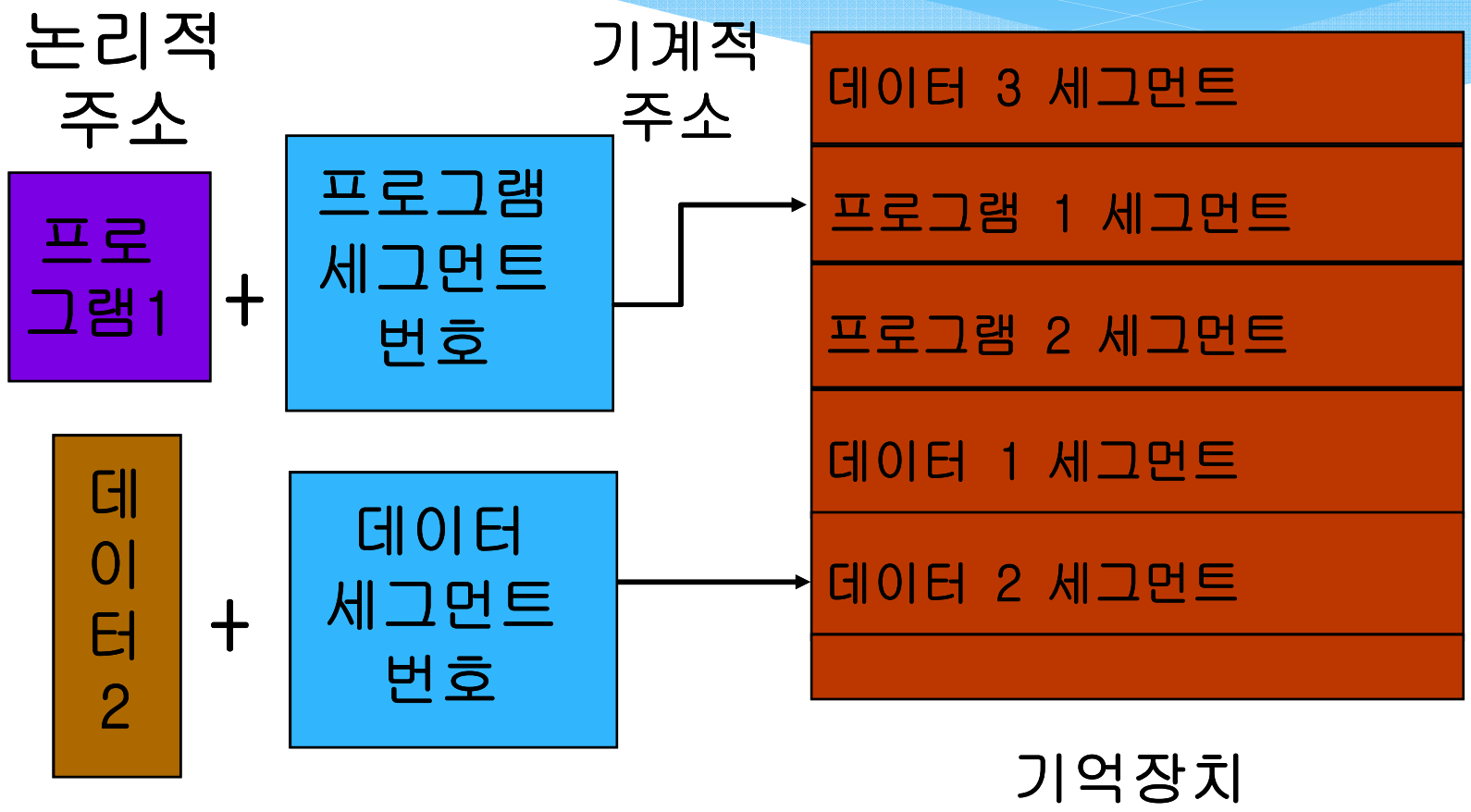
선형기억장치 관리형태



선형기억장치 관리형태


시스템 관리자
프로그램
시스템 관리자
데이터
프로그램1
프로그램2
시스템 관리자
프로그램
시스템 관리자
데이터
프로그램1
프로그램2
기억장치 관리자
기억장치

세그먼트 기억장치 관리형태



제2장 요약

* 비트,바이트,단어의 개념 이해
* 2진수 연산과 관련된 개념이해
* 시스템버스와 프로세서의 이해
* 컴퓨터동작의 기본사이클의 이해
* 선형 및 세그먼트기억장치 관리방법 이해

연습문제

1. N비트 정수(signed) 표현방법(2의 보수)에서 표현 범위는 어떻게 되는가?

2. 단어의 길이가 16비트인 컴퓨터에서 4바이트로 된 데이터를 읽어 오려면 몇 번의 판독(read)사이클이 필요한가?

3. 4비트 정수표현(2의보수)에서 7+1의 결과는 어떻게 될까?

4. 주기억장치가 64k(65536)바이트의 주소 공간을 가지고, 하나의 주소에 대응하는 기억공간이 한 바이트 크기라면, 데이터버스와 주소 버스의 크기는 얼마가 적합한가?

연습문제

5.컴퓨터의 성능은 기억용량과 처리속도에 의해서 결정된다. 기억용량과 처리속도를 결정하는 요소들을 각각 설명하라.

6.어느 컴퓨터의 기억용량이 64k바이트이다. 이 기억용량을 2배로 확장하려고 할 때 고려해야 할 요소는 무엇인가?

7.다음 용어를 설명하라.

1)ASCII

2)비트

3)바이트

연습문제

4)단어
5)긴단어(double word)
6)RAM
7)ROM
8)기억장치 관리방법
9)프로세서
10)레지스터
11)판독 및 기록사이클
12)사이클시간, 시스템 클럭, 펄스
13)부울대수, 논리회로
14)시스템 버스

연습문제

8.명령어 실행 4단계를 설명하라.

9.문자가 어떻게 컴퓨터에서 2진수로 저장될 수 있는가 설명하라.
왜 8비트가 필요한가도 설명하라.

10.8비트에 다음과 같은 숫자가 기록되어 있다.
01000111
이것은 십진수로 얼마를 나타내는가?
그리고 문자로는 어떤 문자를 나타내는가?