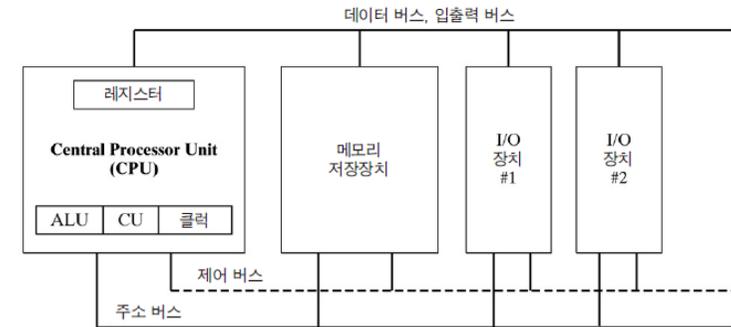


2장 x86 프로세서 구조

마이크로컴퓨터의 기본 구조



ALU: 산술논리연산

제어장치(CU): 실행순서 제어

클럭: 구성요소들의 동작 동기화

CPU + memory + I/O
+
bus



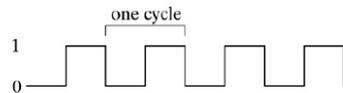
연세대학교

어셈블리어

2

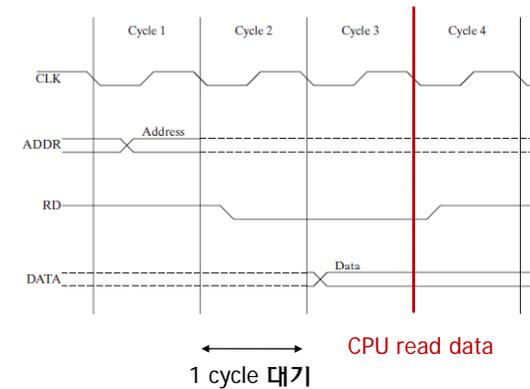
클럭 (Clock)

- CPU와 Bus 동작은 클럭에 동기되어 동작을 한다.



- 1 클럭 사이클 동안 간단한 동작을 수행한다.
 - 기계어 명령어 수행에 적어도 1 클럭 사이클이 필요함
 - 복잡한 명령어는 많은 클럭 사이클이 필요할 수 있다.
 - 메모리 접근 시에 느린 메모리 속도 때문에 **대기 상태**(wait state) 사이클들이 존재할 수 있다. (다음 페이지 그림 참조)
 - 메모리 접근 속도를 높이기 위해 고속 **캐시 메모리** 사용
 - 대기 상태가 없거나 작음
 - 최근에 읽은 메모리 내용을 복사함.

- 메모리 읽기 사이클과 대기상태



연세대학교

어셈블리어

4



연세대학교

어셈블리어

3

프로세서와 x86 프로세서

- 프로세서 - 명령어를 실행하고 데이터를 처리하는 장치
 - (예) CISC계열: 80x86, 680x0
 - (예) RISC계열: Sparc, Alpha, PowerPC, ARM 등
- x86 프로세서
 - Intel사의 프로세서인 80x86 및 Pentium 계열 프로세서의 일반적 이름
 - 16비트 architecture: IA-16
 - 32비트 architecture: IA-32
 - 64비트 architecture:
 - IA-64 (Itanium계열): 하위호환성 없음, 현재 단종됨
 - Intel64 (AMD의 x86-64 명령어 지원) : 하위호환성
 - AMD사에서도 x86 호환 프로세서를 제조함



x86 프로세서 역사

x86 계열 프로세서

이름	특징
8086, 8088	(1978) 16 bit register, 최대 1MB 메모리, real mode - 8087 math coprocessor 별도 사용
80286	(1982) 16 bit register, 최대 16MB 메모리, protected mode 추가 - 80287 math coprocessor 별도 사용
80386	(1985) 32 bit register, 최대 4GB 메모리, 페이징 및 가상 8086모드 기능을 가진 protected mode - 80387 math coprocessor 별도 사용
80486	(1989) + cache memory, math coprocessor 내장
P5 Pentium Pentium Pro	(1993) + 64 bit data bus , data cache, instruction cache (1995) Pentium + 36-bit address(최대64GB) , 내부구조 개선, L2 cache
PentiumMMX	(1996) Pentium + MMX instruction 추가



x86 프로세서 역사(계속)

이름	특징
P6 Pentium II AMD K6	(1997) + MMX instructions 추가, 64GB메모리, 내부구조 개선 (1997) + 3DNow instruction
Pentium III	(1999) + SSE instructions 추가, 내부구조 개선 Pentium M (2003), Intel Core(2006) : for low power
P7 Pentium 4 Athlon(K7)	(2000) + SSE2 instructions, Hyperthreading, NetBurst 구조 (1999) + MMX, 3DNow, SSE
P8 Athlon64(K8) Intel Core2	(2003) + x86-64 (64-bit 연산) instructions (2006) + x86-64 (64 bit 정수연산), SSE3, SSE4
x86-64 P8/9 Phenom(K10) Core i3/5/7 Atom	(2007) + SSE4a (2008) + SSE4a, 구조개선, on-die L3 cache (2008) + for very low power, on-die GPU

- 참고: <http://en.wikipedia.org/wiki/X86>



IA-32 (x86) 의 동작 모드

- Real-address mode
 - 8086호환 모드
 - MS-DOS에서 사용, Windows 98에서도 사용가능
- Protected mode *
 - 80286 이상에서 사용되는 동작 모드
 - 메모리 보호 및 segmentation/paging 기능을 포함 (286은 paging제외)
 - Windows, Linux에서 사용
- System management mode(SMM)
 - 전원 관리, 시스템 보안, 진단 등의 기능 수행
- (cf) virtual 8086 mode
 - protected mode에서 real-address mode용 프로그램을 그대로 수행할 수 있도록 주소변환을 하는 동작 모드
 - 멀티 태스킹 환경에서 여러 개의 MS-DOS 프로그램을 실행시킬 수 있음.



주소 공간

주소 공간

- 선형주소 공간: 프로그램의 주소공간
- 물리적 메모리 주소공간: 반도체 메모리의 주소공간

동작모드와 (선형)주소 공간

동작 모드	주소 비트수	주소공간 크기
real-address mode, virtual-8086 mode	20	1MB
protected mode	32	4GB

물리적 메모리 주소 공간

- 8086 : 20비트 (1MB)
- 80286: 24비트 (16MB) ... 선형주소공간도 16MB임
- 386 ~ Pentium : 32비트 (4GB)
- Pentium Pro, P6~ : 36비트 (64GB) → PAE (physical address extension)



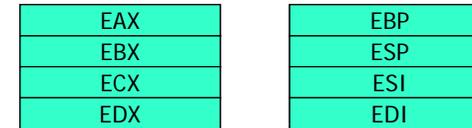
IA-32 레지스터

레지스터

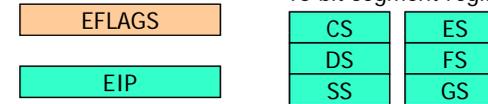
- 데이터/주소를 저장하기 위해서 프로세서 내에 제공되는 기억장소

IA-32의 레지스터

32 bit general purpose registers



16 bit segment registers



범용 레지스터

범용 레지스터

- 데이터와 주소 저장용으로 사용되는 32비트 레지스터
- (cf) x86의 레지스터는 범용 레지스터도 특수 용도를 갖고 있다.

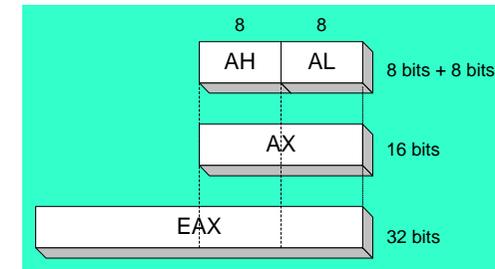
- EAX (Extended Accumulator)
- EBX (Base)
- ECX (Count)
- EDX (Data)

- EBP (Base Pointer)
- ESP (Stack Pointer)

- ESI (Source Index)
- EDI (Destination Index)



범용 레지스터의 부분 접근



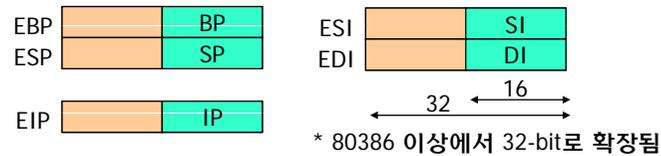
* 80386 이상에서 32-bit로 확장됨

32-bit	16-bit	8-bit (high)	8-bit (low)
EAX	AX	AH	AL
EBX	BX	BH	BL
ECX	CX	CH	CL
EDX	DX	DH	DL



범용 레지스터의 특수 목적 사용

레지스터	특정한 용도
EAX, AX	- 곱셈, 나눗셈, translate 연산용 accumulator - 입출력 연산: 입출력 데이터 보관
EBX, BX	- base주소 보관 (8086,80286)
ECX, CX	- 루프 반복 횟수 보관 (loop count) - shift 횟수 보관 (shift count)
EDX, DX	- 입출력 연산: 포트주소 보관 - 곱셈, 나눗셈에서의 operand 보관
ESP, SP (stack pointer)	- 스택의 top 위치 주소를 보관
EBP, BP (base pointer)	- 스택의 데이터 주소를 보관 (지역변수, 매개변수) frame pointer라고도 함
ESI, SI (source index) EDI, DI (dest. index)	- 메모리 블록 전송 명령어에서 사용



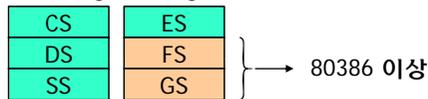
32-bit	16-bit
ESI	SI
EDI	DI
EBP	BP
ESP	SP



세그먼트 레지스터

- 세그먼트 레지스터 - segment의 시작(base)주소 지정에 사용됨

16 bit segment registers



- 세그먼트(segment)

- 데이터 또는 코드용으로 할당된 메모리 영역
- 세그먼트의 크기
 - 8086, 80286은 최대 64KB (2^{16}) - 16비트 offset 주소 사용
 - 80386이상은 최대 4GB (2^{32}) - 32비트 offset 주소 사용



세그먼트 종류

- segment 종류

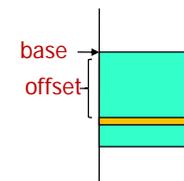
- code segment: 프로그램 코드를 포함
- data segment: 프로그램에서 정의된 데이터, 상수 등을 포함
- stack segment: 지역변수, 매개변수, 함수의 복귀주소 등을 포함

- segment registers - 각 세그먼트의 시작주소 정보를 보관

- CS: code segment
- DS, ES (, FS, GS): data segment
- SS: stack segment

- segment offset

- segment 시작주소에 상대적인 주소
- 8086/286: 16비트, 386이상: 32비트



- 메모리 주소 = 세그먼트 시작주소 + 오프셋 주소

- 세그먼트 시작주소 지정 방법은 동작 모드에 따라서 차이가 있음

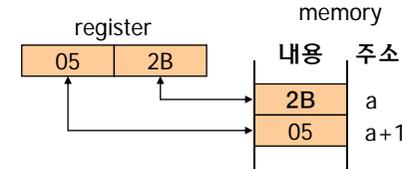


명령어 포인터

- 명령어 포인터(Instruction Pointer)
 - 실행할 다음 명령어의 offset 주소를 저장
- EIP (32비트)
 - 80386이상의 보호모드에서 사용
- IP (16비트)
 - 8086, 80286
 - 80386 이상의 real mode에서 사용

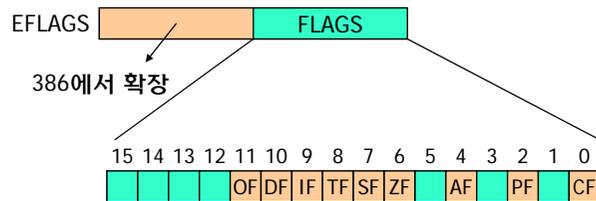
메모리 주소

- 메모리 주소
 - 메모리의 각 위치에 주소(address)가 부여됨 - 대개 byte 단위
- 메모리 상의 데이터 주소 지정
 - 2 byte 이상의 데이터는 2개 이상의 연속적인 주소의 메모리에 저장됨
 - 80x86계열은 little-endian 방식을 사용 - 하위 바이트가 낮은 주소

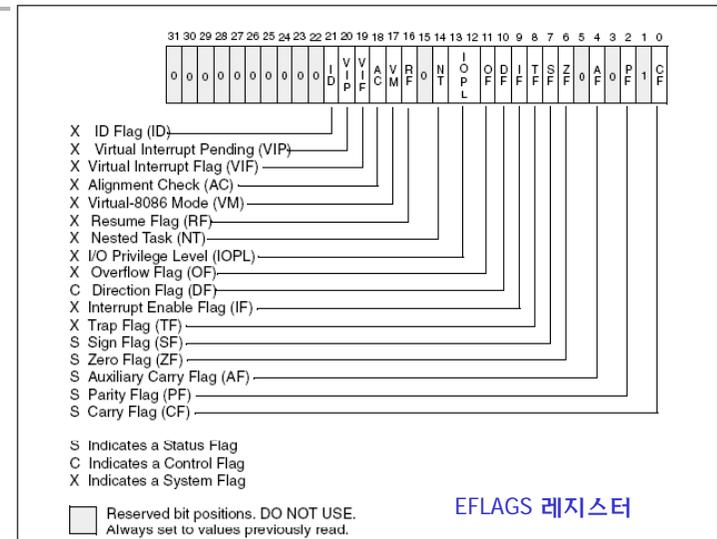


플래그(flag) 레지스터

- 플래그 레지스터
 - 프로세서의 현재 상태와 연산결과를 나타내는 레지스터



- 상태플래그: 연산결과를 나타내는 비트:
 - SF(sign), ZF(zero), CF(carry), OF(overflow)
 - AF(auxiliary carry), PF(parity)
- 제어플래그: 프로세서 동작을 제어하는 비트:
 - DF(direction), IF(interrupt), TF(trap)



MMX, XMM 레지스터

MMX Registers



MMX Registers

XMM Registers (SSE, SSE2 registers)



XMM Registers

32-bits

MXCSR Register



- Carry (CF)
 - unsigned arithmetic out of range
- Overflow (OF)
 - signed arithmetic out of range
- Sign (SF)
 - result is negative
- Zero (ZF)
 - result is zero
- Auxiliary Carry (AF)
 - carry from bit 3 to bit 4
- Parity (PF)
 - sum of 1 bits is an even number

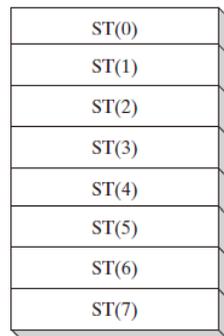


FPU 레지스터

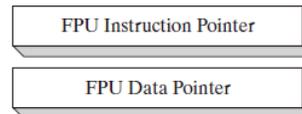
■ 부동소수점 연산장치에서 사용

데이터 레지스터: 스택구조

80-bit Data Registers



48-bit Pointer Registers



16-bit Control Registers



Opcode Register



세그먼트와 주소 지정

■ 선형주소

- 8086(또는 real mode)에서 20 bit 크기 → 최대 1MB 메모리 참조 가능
- 80286 보호모드에서 24 bit 크기 → 최대 16MB 메모리 참조 가능
- 80386 이상 보호모드에서 32 bit 크기 → 최대 4GB 메모리 참조 가능

■ 선형주소와 물리적 메모리 주소의 관계

- paging을 사용하지 않으면: 선형주소 = 메모리 주소
- paging을 사용하면: 선형주소 ≠ 메모리 주소

■ 세그먼트 레지스터와 세그먼트 시작 주소 지정

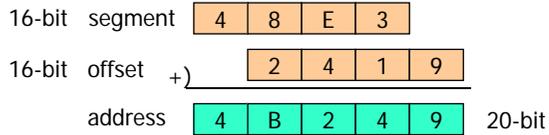
동작모드	segment register값의 내용
8086 또는 80286이상 real mode	20-bit segment 시작주소의 상위 16-bit
80286 보호모드	24-bit segment 시작주소들을 저장한 테이블의 인덱스(→ segment selector라고 부름)
80386이상 보호모드	32-bit segment 시작주소들을 저장한 테이블의 인덱스 (그리고 32-bit offset 사용)



real address mode에서의 주소 지정

segment와 offset을 사용한 주소 지정

- 8086의 register는 16 bit 크기 → 최대 64KB 메모리 참조 가능
- 8086에서 16 bit segment register들을 추가로 사용하여 → 20 bit 메모리주소 제공



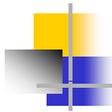
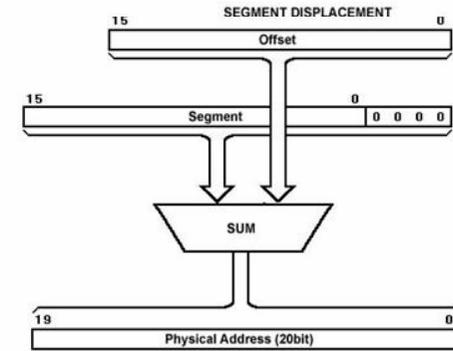
address = (segment << 4) + offset

segment : offset
= 48E3:2419
= 4B249

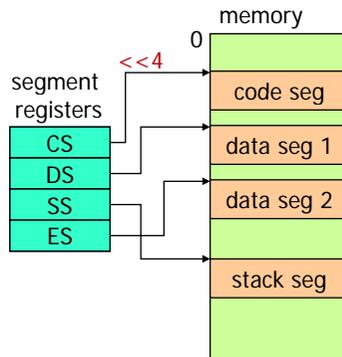


physical address = segment시작주소(20-bit) + offset(16-bit)

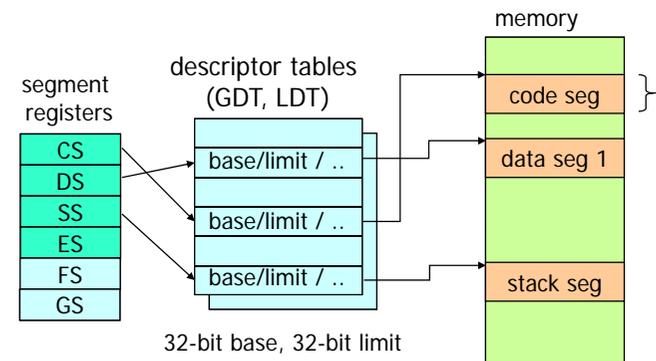
16의 배수 segment 크기는 최대64KB



segment와 segment register



Protected mode 주소 지정

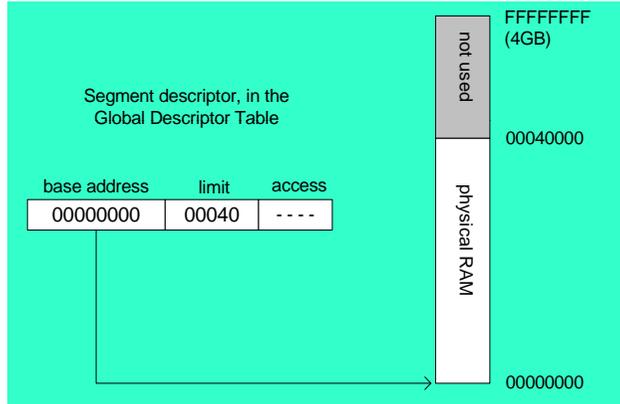


메모리 주소 = 32-bit 세그먼트 시작주소 + 32-bit 오프셋 주소
(오프셋은 해당 세그먼트의 limit보다 작아야 함)



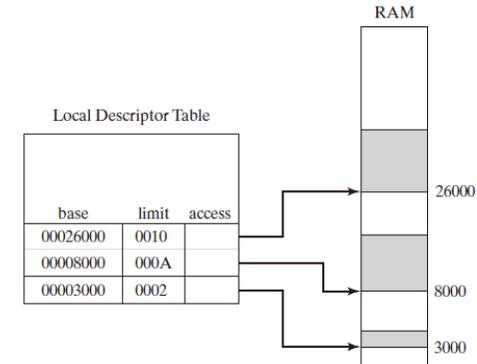
균일 세그먼트(Flat segment) 모델

- 모든 세그먼트가 32-bit 전체 메모리 공간에 맵핑
 - 전체 메모리를 단일 세그먼트로 사용
 - GDT(global descriptor table)에 segment descriptor를 정의함



다중 세그먼트 모델

- 각 프로그램은 자신의 세그먼트 테이블을 가짐
 - 자신의 LDT에 세그먼트들의 segment descriptor를 정의함.



- Windows, Linux 등의 운영체제는 다중 세그먼트 모델을 사용하지 않고 균일 세그먼트 모델을 사용함



x86-64

- x86-64 (Intel64)
 - 64비트 선형주소 공간과 정수연산을 제공하는 명령어 집합
 - IA32 x86 프로세서와의 하향 호환성을 제공함
 - AMD가 처음으로 구현
- IA-32e 동작모드
 - 64비트 모드:
 - 64비트 주소, 피연산자
 - 세그먼테이션 비활성화, 균일 선형주소 공간 사용
 - 호환모드:
 - 16비트, 32비트 응용프로그램을 실행가능하도록 함
 - 32비트 주소, 16/32비트 피연산자



Intel 64 64-bit 범용 레지스터

- 8개의 범용 레지스터 추가(R8-R15)
- 64비트 레지스터 명칭
 - RAX, RBX, RCX, RDX, RDI, RSI, RBP, RSP, R8-R15
- 추가된 레지스터들도 32, 16, 8비트 레지스터로 사용 가능
 - 레지스터 이름 끝에 D (32비트), W (16비트), L (8비트)를 사용

Register Type	With REX
Byte Registers	AL, BL, CL, DL, SIL, BPL, SPL, R8L - R15L
Word Registers	AX, BX, CX, DX, SI, BP, SP, R8W - R15W
Doubleword Registers	EAX, EBX, ECX, EDX, EDI, ESI, EBP, ESP, R8D - R15D
Quadword Registers	RAX, RBX, RCX, RDX, RDI, RSI, RBP, RSP, R8 - R15



CISC와 RISC

- CISC – complex instruction set computer
 - 많은 수의 명령어들로 구성됨
 - 복잡한 동작의 고수준 연산을 수행하는 명령어들이 포함됨
 - **마이크로코드(micro-code)**에 의한 기계어 해독
 - (예) Intel 80x86 계열, 680x0계열
- RISC – reduced instruction set computer
 - 적은 수의 짧고 간단한 명령어들로 구성됨
 - 하드웨어에 의한 기계어 처리
 - (예)
 - ARM (Advanced RISC Machines)
 - Alpha (Digital, now HP)
 - SPARC (Sun, now Oracle)
 - PowerPC (IBM)



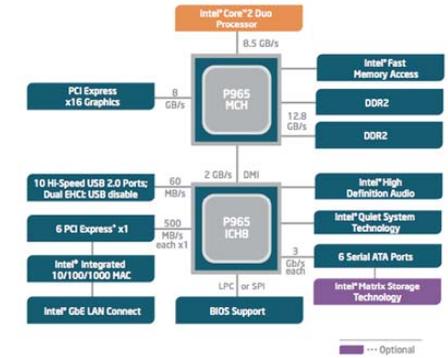
x86 컴퓨터 구성 요소

■ 프로세서



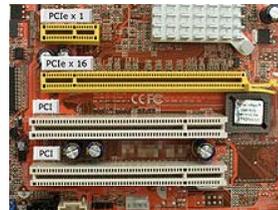
■ 주기판 칩셋

- north bridge
 - 메모리 제어기 허브
- south bridge
 - 입출력 제어기 허브



x86 컴퓨터 구성 요소 - 버스 등

- PCI / PCI-express 버스 - 주변기기 제어기와 연결
 - PCI 버스 : 병렬 연결
 - PCI express 버스: 고속 직렬 연결
- 메모리
 - ROM (EPROM)
 - DRAM, SRAM
 - VRAM
 - CMOS RAM – 배터리 사용
- 비디오 출력
 - CRT 모니터용 출력 – VGA (raster scanning 영상 신호)
 - LCD 모니터 전용 출력 – DVI (영상), HDMI (영상, 음성)



x86 컴퓨터 구성 요소 - 입출력 포트 등

- 범용 직렬 버스 (USB)
 - USB 1.x 최대 12 Mbps
 - USB 2.0 최대 480 Mbps
 - USB 3.0 최대 4 Gbps
- 저장장치용 인터페이스
 - ATA host adapter
 - SATA (serial ATA) host adapter
- 네트워크 인터페이스
 - LAN – Ethernet, Wi-Fi – 무선 Ethernet
 - Bluetooth
- PS/2 포트 – 키보드/마우스 용
- 병렬 포트 – 프린터 용
- 직렬 포트
 - RS232 인터페이스, 터미널/모뎀 접속용

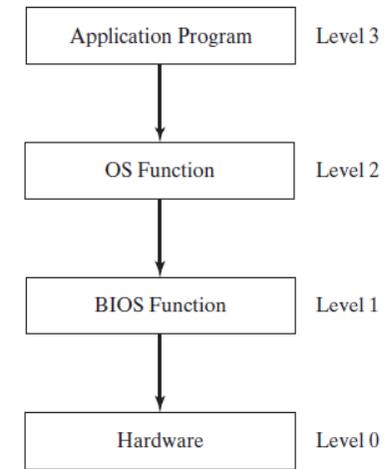


입출력 시스템 - 입출력 접근 레벨

- Level 3: 고급언어 함수 호출 (C++, Java)
 - 사용하기 쉬움, 하드웨어 추상화(세부사항을 숨김)
 - 성능이 가장 느림
- Level 2: 운영체제 서비스 함수 호출
 - 운영체제에 따라 다름, device independent 서비스
 - 중간 성능
- Level 1: BIOS (basic input-output system) 서비스 함수 호출
 - 시스템에 따라서 다른 결과가 나올 수 있음
 - 하드웨어에 대한 지식이 필요함
 - 대개 성능이 좋음
- Level 0: 하드웨어에 직접 접근
 - 일부 운영체제에서는 허용하지 않음
 - MS-DOS : 허용
 - Window95 이후 : 허용하지 않음



고급언어 프로그램의 입출력 접근



어셈블리 프로그램의 입출력 접근 수준

