

컴퓨터 구조 개요(1장, 2장)

성결대학교 컴퓨터공학부

최정열 교수

(passjay@sungkyul.ac.kr)

범사에 감사하라 이것이 그리스도
예수 안에서 너희를 향하신 하나님
의 뜻이니라(데살로니가전서5:18)

수업 목표

- ❑ 컴퓨터를 구성하는 주요 부품이 무엇인지 말할 수 있다
- ❑ 컴퓨터의 구성 요소의 기능을 설명할 수 있다
- ❑ 집적도 향상에 따른 효과가 무엇인지 이해한다
- ❑ 컴퓨터의 성능 향상을 위한 주요 설계 이슈들을 이해한다
- ❑ 컴퓨터 성능 평가 방법을 이해한다

목차

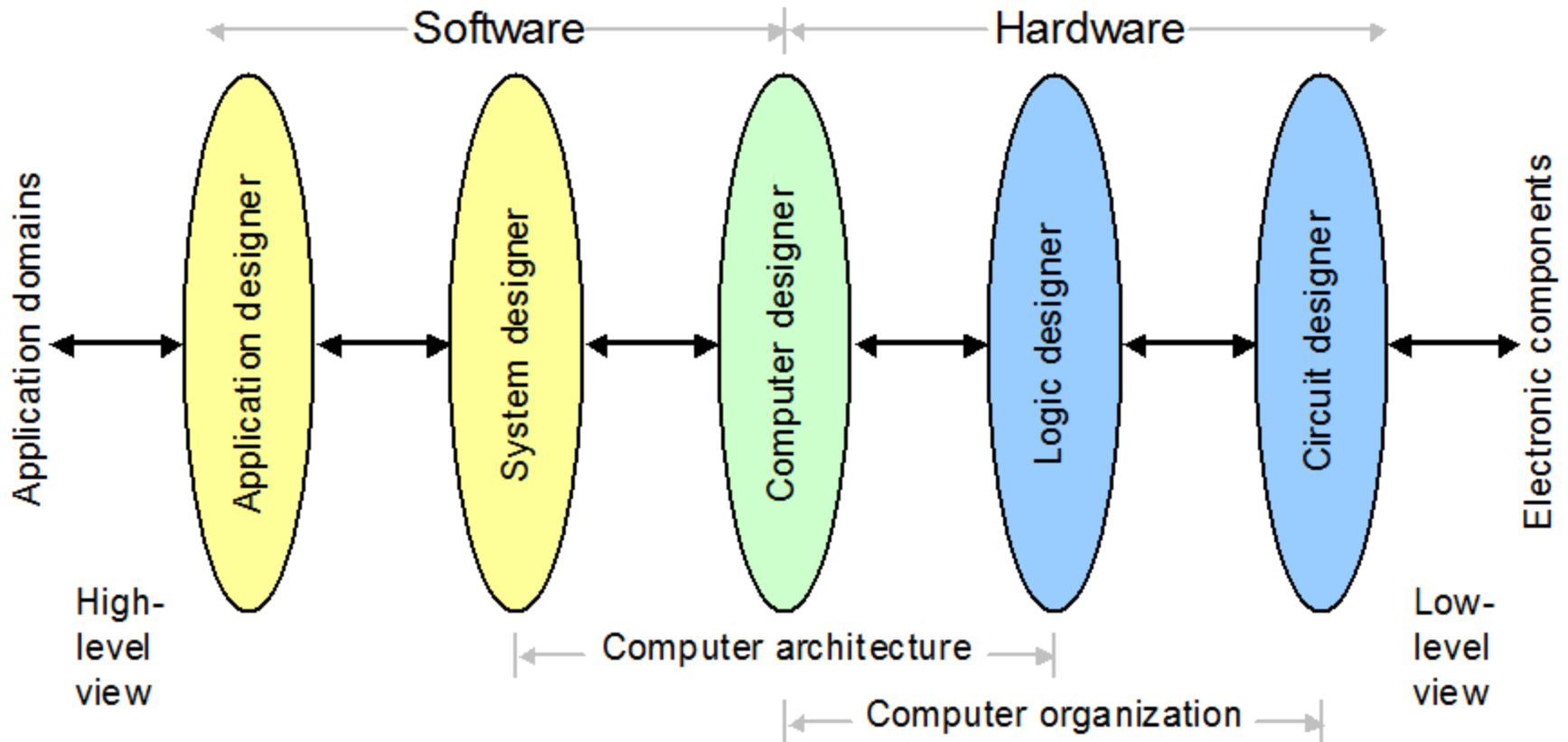
- ❑ 컴퓨터의 조직과 기능
- ❑ 컴퓨터의 구성
- ❑ 소프트웨어
- ❑ 집적회로
- ❑ 컴퓨터 성능 향상
- ❑ 컴퓨터의 성능 평가

❖ CPU는 어떻게 작동할까?

https://www.youtube.com/watch?v=Fg00LN30Ezg&ab_channel=bRd3D

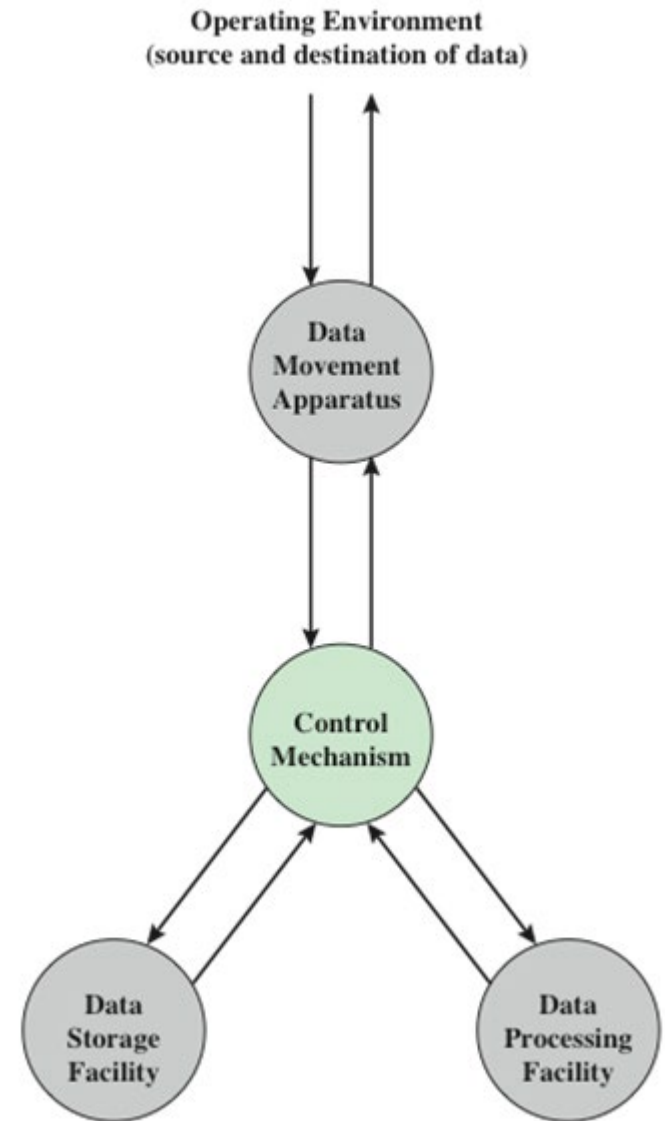
1. 컴퓨터의 조직과 기능

❑ Computer in SW/HW



컴퓨터의 기능

- ☐ _____
- ☐ _____
- ☐ _____
- ☐ _____

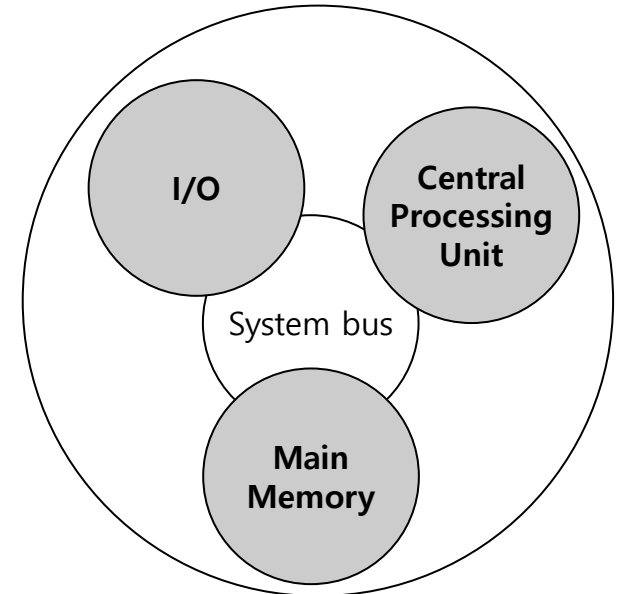


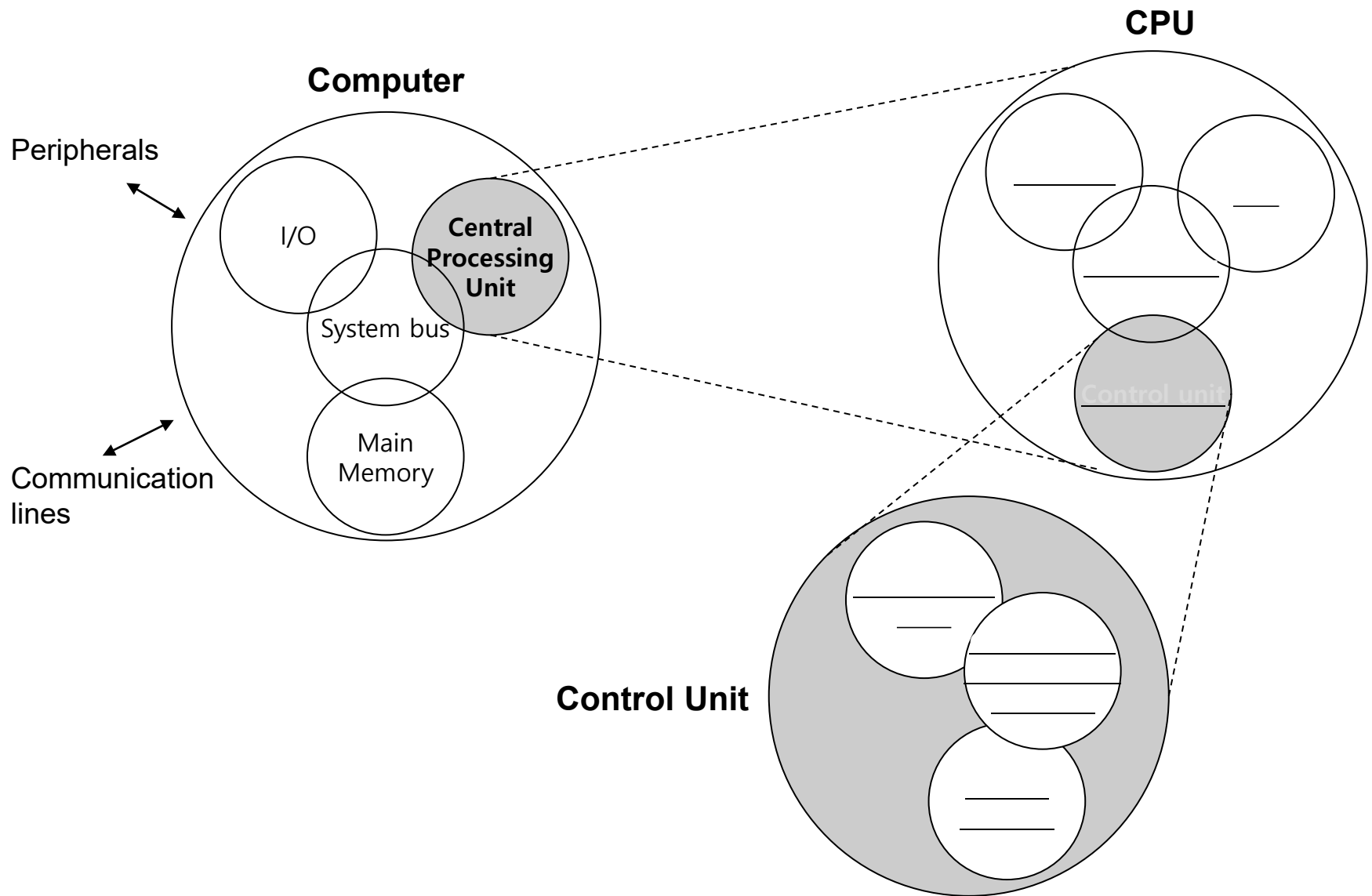
기능적 관점에서 본 컴퓨터

컴퓨터의 조직

□ 최상위에서 본 컴퓨터의 구성 요소

- _____
- _____
- _____
- _____



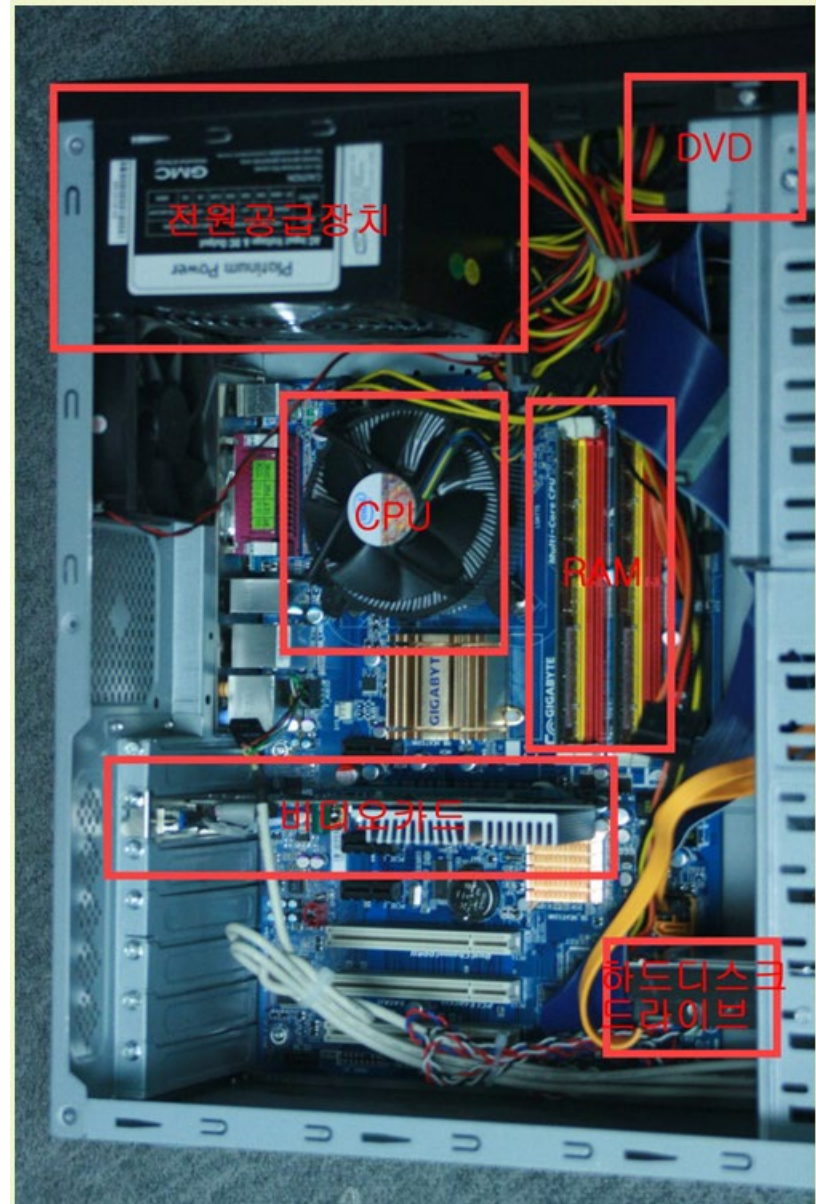


2. 컴퓨터의 구성

□ 컴퓨터 부품들

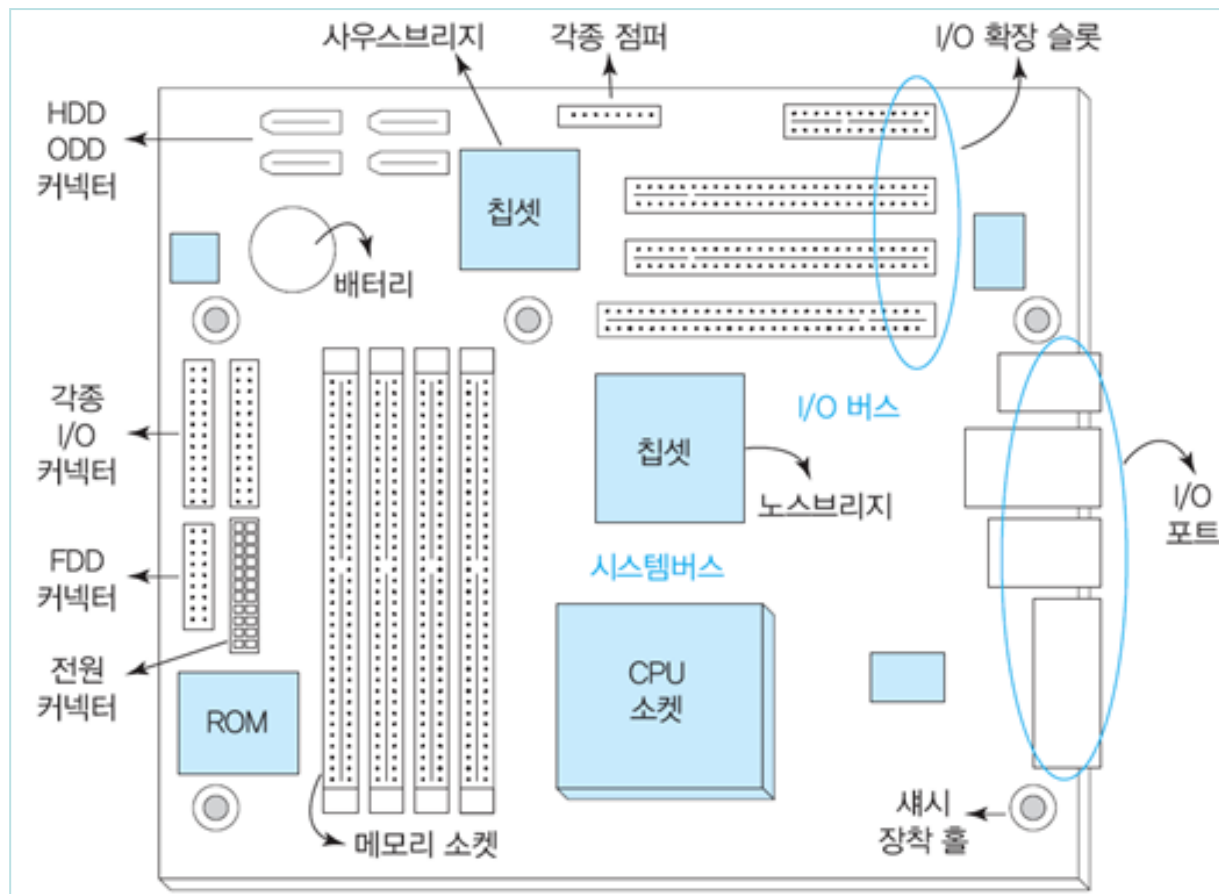
- Main board
- CPU
- RAM(main memory)
- Chipset
- HDD
- DVD
- Graphic card
- Power Supply
- ...

□ 컴퓨터를 조립해보자



Main board(System board, Mother Board)

- 공통의 버스 배선과 인터페이스 회로를 모아 놓고 슬롯 형태의 작은 보드를 꽂아 사용하는 기판 구조에서 어미가 되는 기판

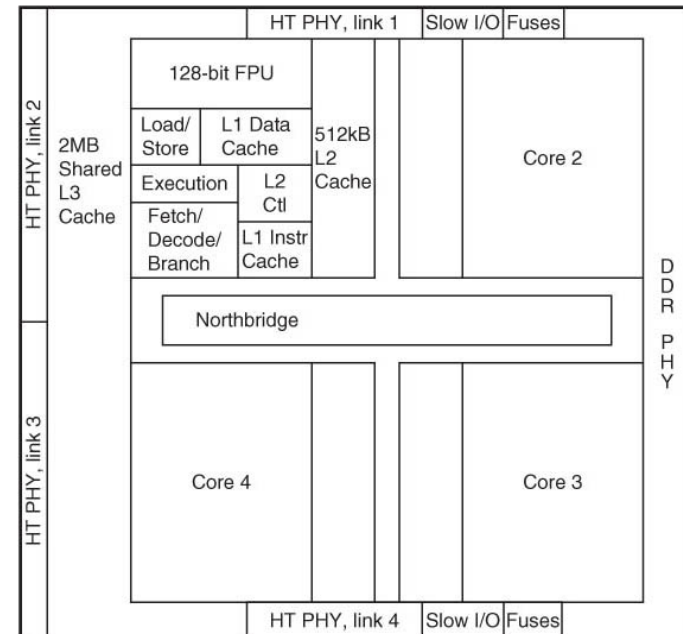
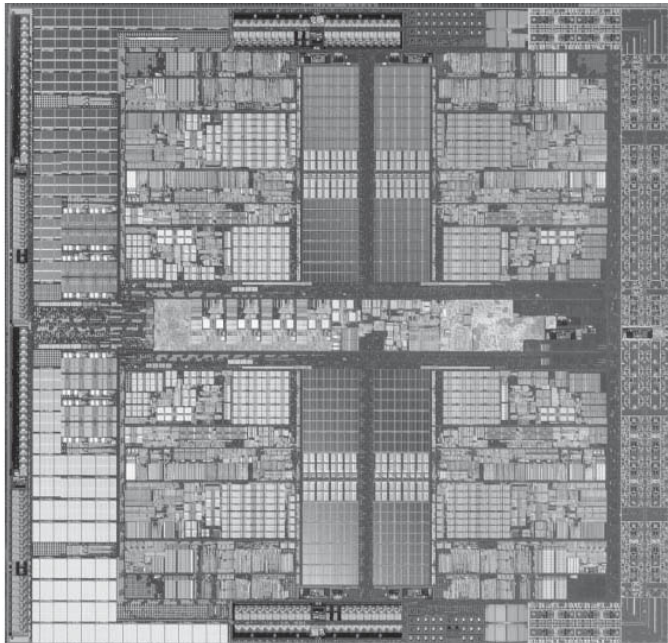


중앙처리장치 (Central Processing Unit, CPU)

❑ 컴퓨터 시스템 전체를 제어



중앙처리장치의 종류



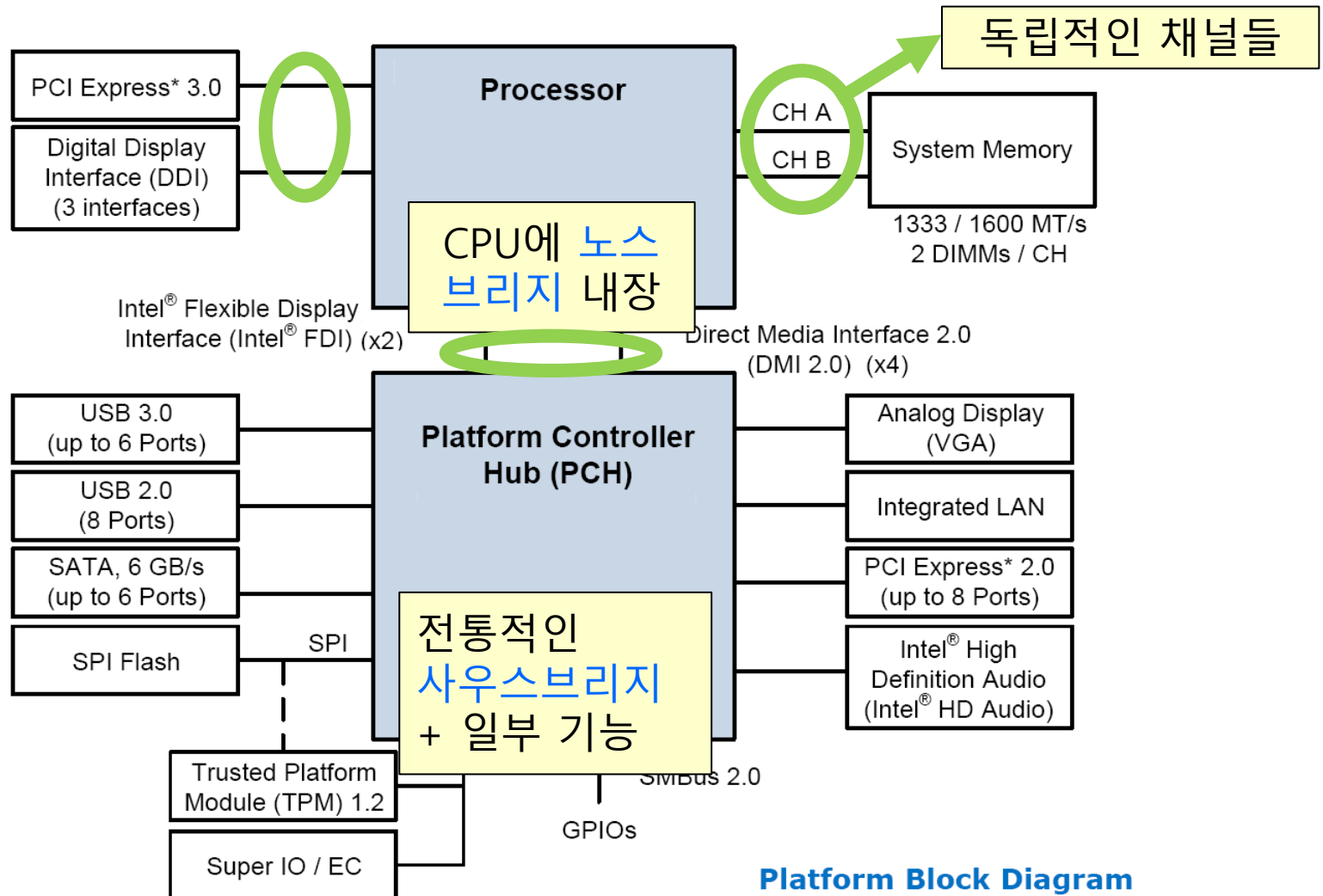
Inside the AMD Barcelona microprocessor

Chipset

- ❑ 여러 칩과 회로가 모여 서로 연관된 기능을 수행하도록 설계된 _____

- ❑ CPU 프로세서와 함께 _____
 - 특정 CPU를 사용해 하드웨어를 구성하도록 설계
- ❑ 대개 메인보드 상에 몇 개의 제어 칩들로 구성
- ❑ 칩셋 내부 회로
 - CPU를 지원하는 각종 제어 장치들
 - 버스 컨트롤러, 메모리 컨트롤러, I/O 컨트롤러, 인터럽트 컨트롤러, 타이머 등

인텔 4세대 Core 프로세서의 플랫폼(2013)



주기억장치(Main Memory)

- ❑ 수행 중인 프로그램과 이를 위해서 필요한 데이터를 기억하고 있는 장치
- ❑ 비교적 CPU에 접근 속도가 빠르며 많은 용량을 기억
- ❑ RAM(Random Access Memory)이 주로 사용

주기억장치



보조기억 장치

- 외부 기억장치
- 반영구적으로 데이터를 저장하고 보존
- 중앙처리장치와 직접 정보를 교환할 수 없기 때문에 _____ 옮겨
진 후 처리
- 주기억장치에 비해 가격은 저렴하고 저장 용량 또한 크지만 속도가 느린 단점
- 자기테이프, 자기디스크, CD-ROM, DVD, Flash Memory, 광디스크



여러 가지 보조 기억 장치(HDD, DVD, CD)

입출력 장치

□ 입력 장치

- 컴퓨터 프로그램과 데이터를 읽어 들이는 역할



다양한 입력장치
(마우스, 키보드, 스캐너, 조이스틱)

□ 출력 장치

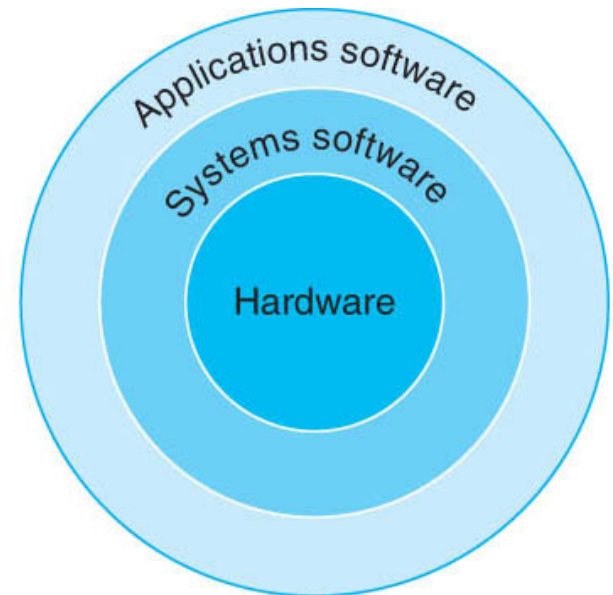
- 컴퓨터 내부에서 처리된 결과를 사용자가 보거나 들을 수 있도록 하는 역할



출력 장치(모니터, 프린터)

3. 소프트웨어

- ❑ 정보의 이동 방향과 정보처리의 종류를 지정하고 그러한 동작들이 일어나는 시간을 지정하는 명령들의 집합
- ❑ 시스템 소프트웨어(System Software)
 - 컴퓨터 시스템을 제어하고 효율적으로 사용하기 위해 만들어진 소프트웨어
 - 운영체제, 디바이스 드라이버, 컴파일러, 인터프리터



프로그래밍에서의 모델 및 추상 수준

More abstract, machine-independent;
easier to write, read, debug, or maintain

More concrete, machine-specific, error-prone;
harder to write, read, debug, or maintain

Very
high-level
language
objectives
or tasks

High-level
language
statements

Assembly
language
instructions,
mnemonic

Machine
language
instructions,
binary (hex)

Swap $v[i]$
and $v[i+1]$

```
temp=v[i]
v[i]=v[i+1]
v[i+1]=temp
```

```
add $2,$5,$5
add $2,$2,$2
add $2,$4,$2
lw  $15,0($2)
lw  $16,4($2)
sw  $16,0($2)
sw  $15,4($2)
jr  $31
```

```
00a51020
00421020
00821020
8c620000
8cf20004
acf20000
ac620004
03e00008
```



One task =
many statements



One statement =
several instructions



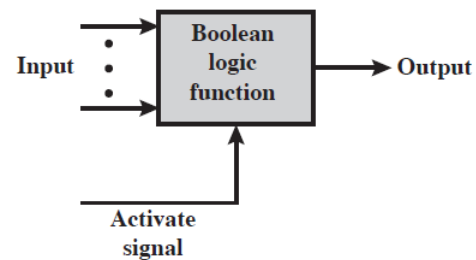
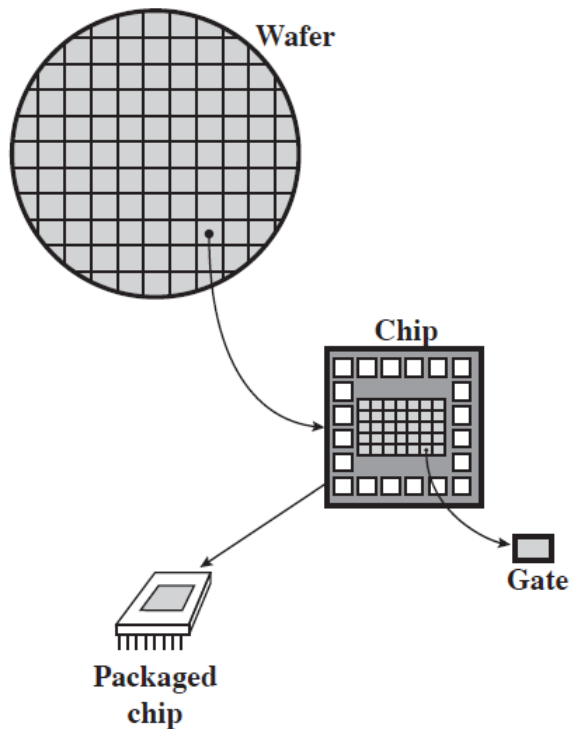
Mostly one-to-one

4. 집적 회로

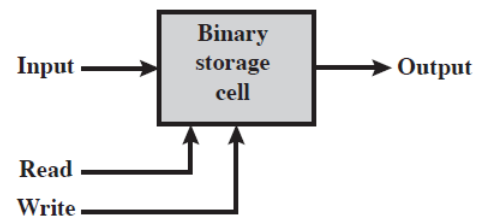
❑ 집적 회로 제조 공정

- 참고 동영상) <https://youtu.be/gld0IHfbyF8>

❑ 웨이퍼, 칩, 게이트의 상관관계



(a) Gate



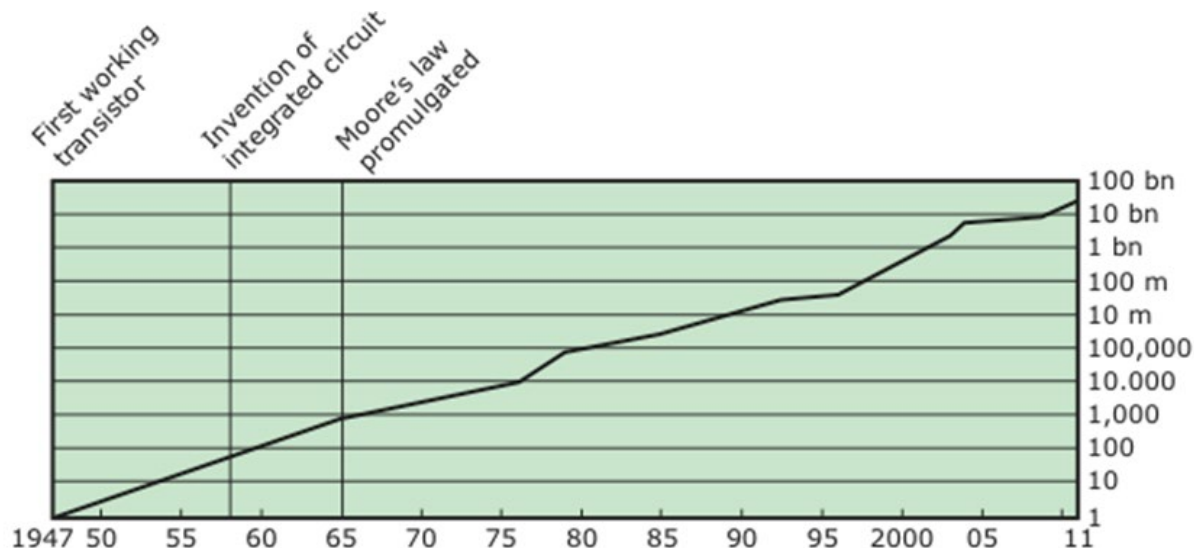
(b) Memory cell

기본적인 컴퓨터 요소

무어의 법칙(Moore's law)

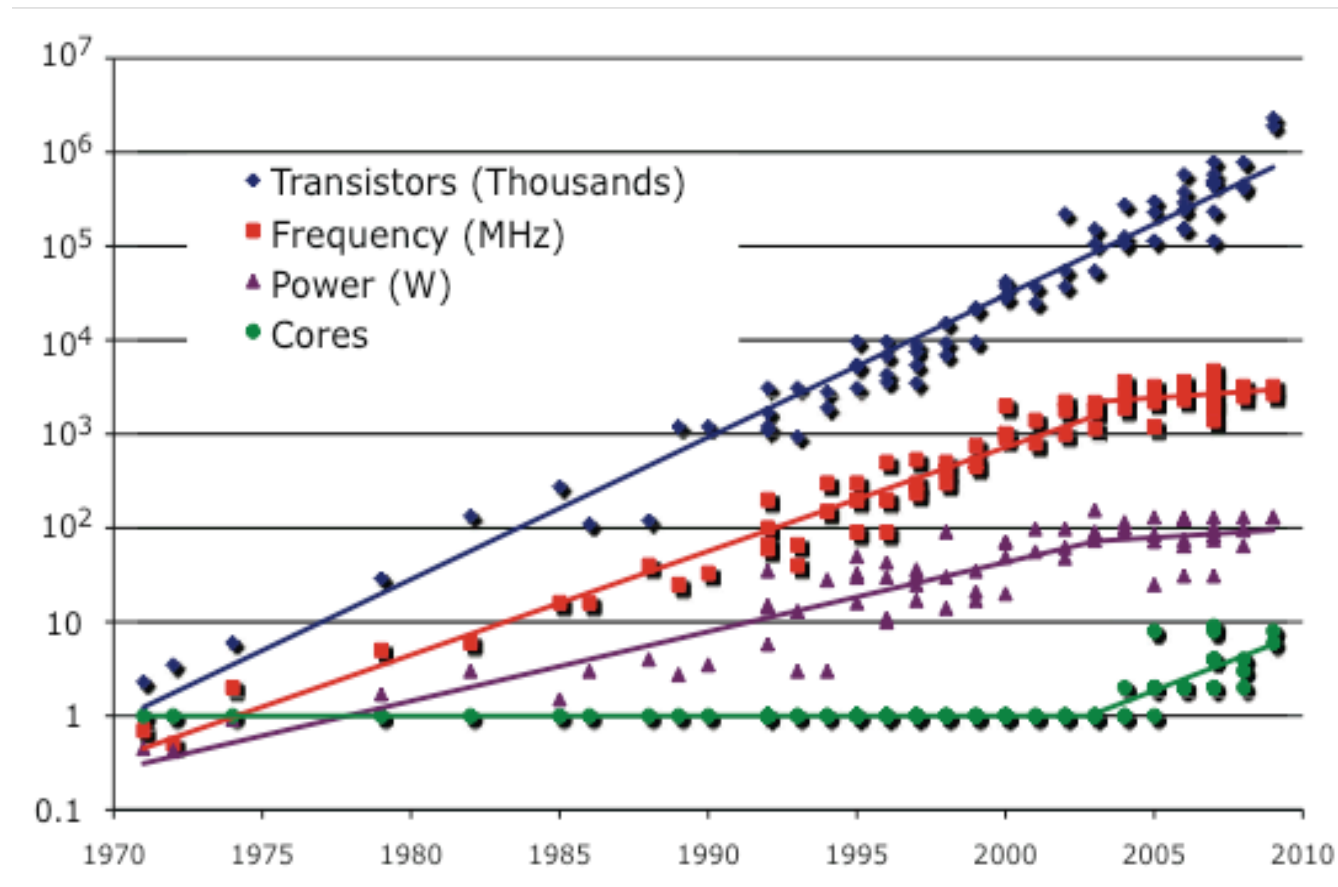
- ❑ 컴퓨터의 크기가 작아졌다
- ❑ 논리회로/기억 소자들이 가까워져 전기적 통로의 길이가 줄어들어 동작 속도가 상승했다
- ❑ 집적회로 내에서 연결되어 납땜 연결보다 신뢰성이 높아졌다
- ❑ 집적도가 상승되는 동안 칩 가격은 변하지 않았다
- ❑ 전력 소모가 줄어들었다

CPU내 트랜지스터 개수의 증가율



5. 컴퓨터의 성능 향상

□ 마이크로프로세서 성능 향상



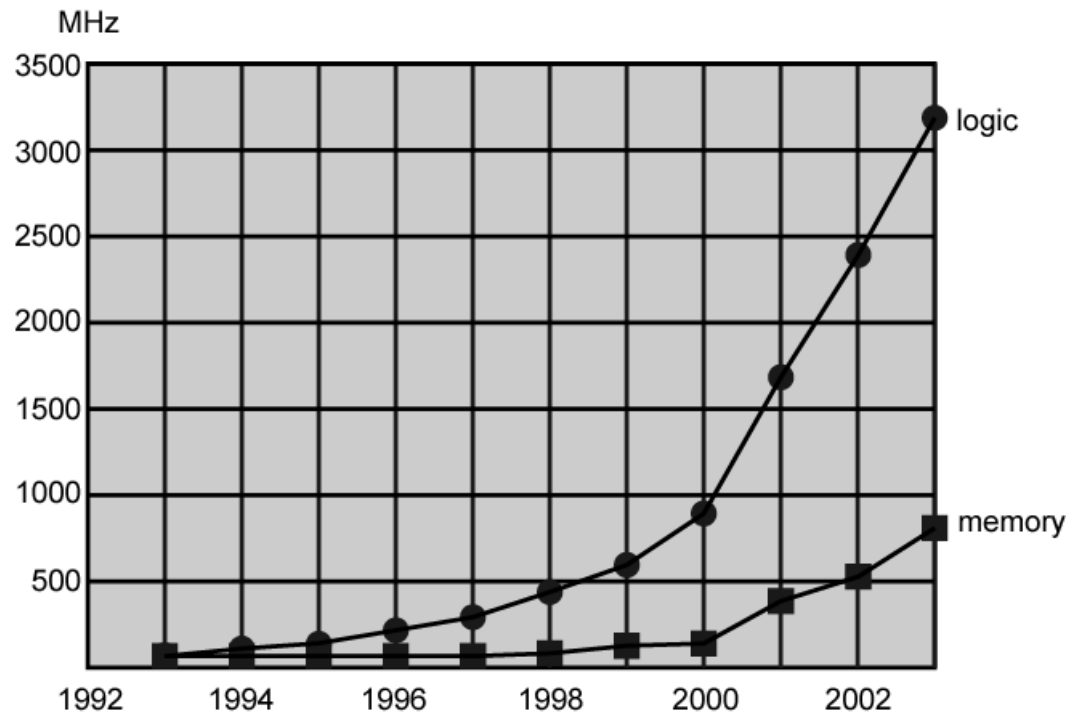
성능 균형 문제(performance balance)

□ 프로세서 속도 증가

□ 기억용량 증가

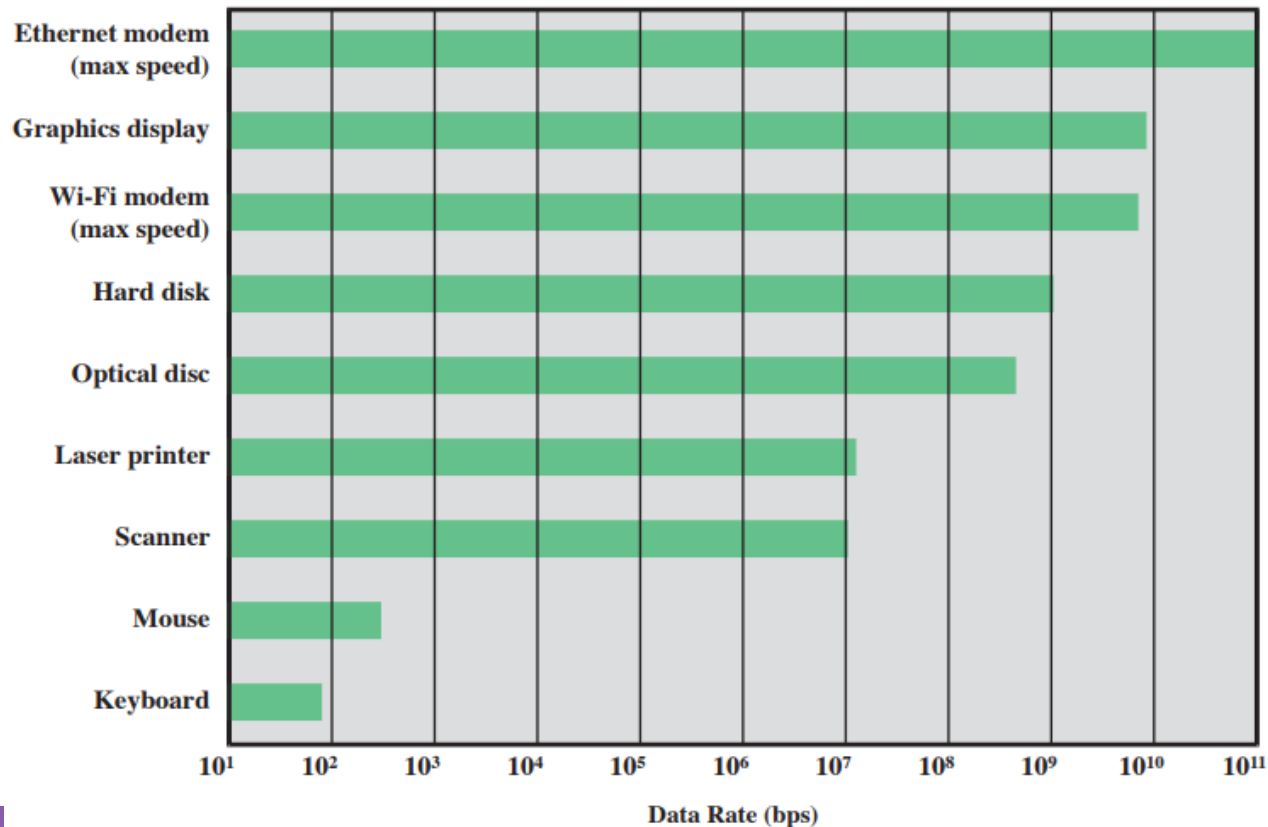
❖ _____ 인터페이스에서 병목 발생

논리 소자와 기억 장치 간의 성능 차이



□ 입출력 장치에서의 성능 균형 문제

- 높은 I/O 요구율을 갖는 주변장치를 사용하는 응용이 등장
- I/O 장치의 데이터 처리 속도의 비약적 증가
- 프로세서와 입출력 장치 간의 데이터 이동은 여전히 문제임



프로세서의 속도 향상 방안

❑ 프로세서의 하드웨어 속도 증가

- 논리 게이트 크기를 줄이고, 더 많은 게이트를 더욱 조밀하게 넣고_____
- 신호들의 전파 시간 감소

❑ 프로세서-주기억장치 사이의 _____

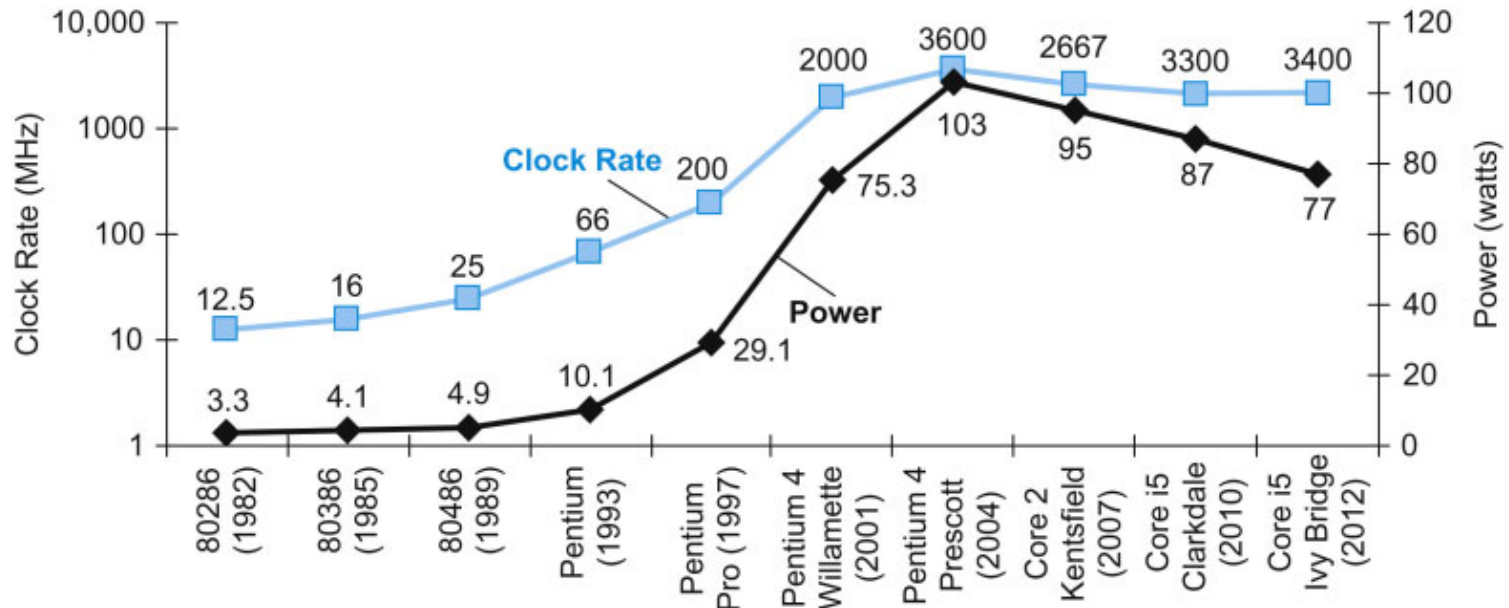
- 프로세서 칩 내부에 캐시를 배치하여 캐시 액세스 시간을 감소

❑ 명령어 실행 속도를 높일 수 있도록 프로세서 조직과 구조를 변경

- 파이프라이닝, 수퍼스칼라 등의 _____ 제공

클럭 속도와 회로 밀도 문제

- ❑ 칩 내부 회로의 밀도와 클럭 속도가 높아지면 _____(Watts/cm²) 증가
- ❑ 트랜지스터 하나가 소비하는 전력은 논리 값이 한 번 바뀔 때 소모되는 에너지와 시간당 논리 값이 바뀌는 빈도수의 곱



인텔 마이크로프로세서의 클럭 속도와 전력의 변화

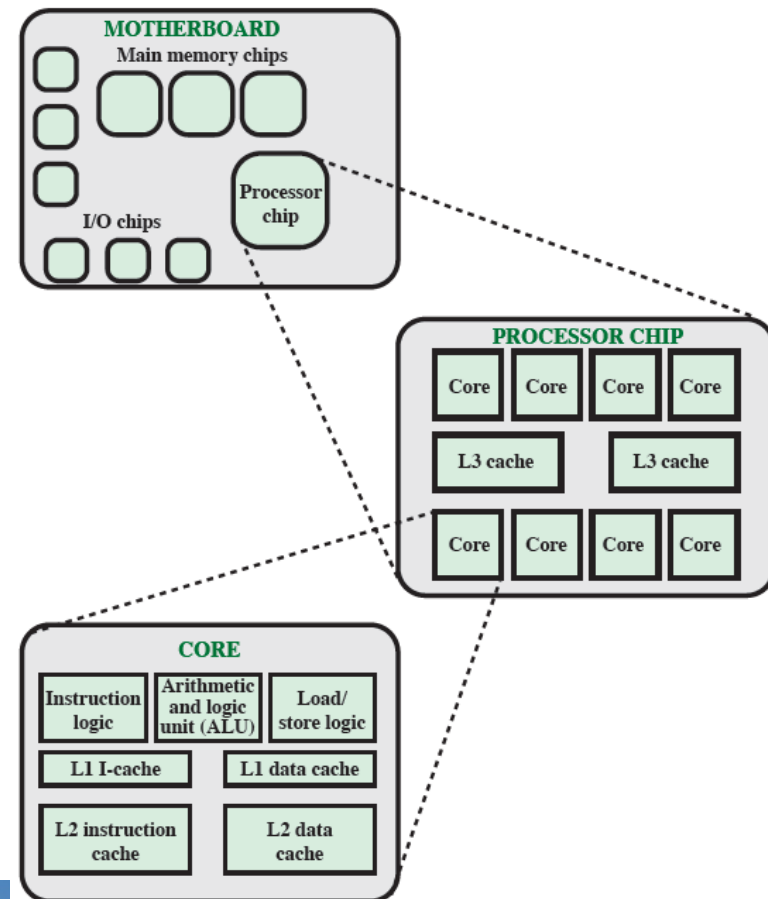
성능 향상을 위한 새로운 시도 - _____

□ _____ 하는 여러 프로세서를 하나의 칩에 배치

- 기억장치 회로의 소비전력이 프로세서 회로보다 훨씬 적음

□ 하나의 복잡한 프로세서를 사용하기보다는 칩 내에 더 간단한 여러 프로세서를 배치

- 프로세서 내부에서, 성능 향상은 복잡도 증가의 제공근에 비례
- _____가 여러 개의 프로세서들을 효과적으로 사용하도록 지원한다면 프로세서 개수 만큼의 성능 향상을 기대



6. 컴퓨터의 성능 평가

❑ 프로세서의 평가 요소

- _____ 등

❑ 성능 평가의 예: 여객기

Airplane	Passenger capacity	Cruising range (miles)	Cruising speed (m.p.h.)	Passenger throughput (passengers × m.p.h.)
Boeing 777	375	4630	610	228,750
Boeing 747	470	4150	610	286,700
BAC/Sud Concorde	132	4000	1350	178,200
Douglas DC-8-50	146	8720	544	79,424

관점에 따른 컴퓨터 성능 평가

❑ 사용자 관점: _____ 이 중요

- 실행시간: 컴퓨터가 태스크를 완료하기까지의 총 소요시간
- 디스크 접근, 메모리 접근, 입출력 작업, 운영체제의 오버헤드 및 CPU 시간을 포함

❑ 시스템 관점: _____ 이 중요

- 대역폭: 일정한 시간 동안 처리하는 작업의 양

❑ 시스템의 종류에 따라 다른 성과 평가 지표가 적용되어야

- 임베디드 시스템: _____
- 데스크톱 PC: _____
- 서버: _____

성능의 정의

□ 프로그램의 실행 시간

- 벽시계 시간, 응답 시간, 경과 시간
 - 한 작업을 끝내는데 필요한 전체 시간
 - 디스크 접근, 메모리 접근, 입출력 작업, 운영체제 오버헤드 등 모든 시간을 더한 것
- CPU 실행 시간(CPU 시간)
 - 프로세서가 순수하게 이 프로그램을 실행하기 위해서 소비한 시간

□ 성능의 정의

- 성능 = 1 / 수행시간

- _____

- _____

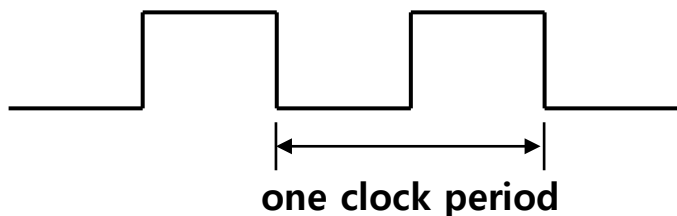
성능에 영향을 미치는 요소

□ 클록 율(clock rate, clock cycle, clock tick, cycle time)

- 클록 신호는 연속적인 신호 파형을 발생하는 수정 결정판(quartz crystal)에 의해 생성



- 프로세서의 속도는 클록에 의해서 발생하는 펄스 주파수에 의해 결정



10 nsec clock cycle	=>	100 MHz clock rate
2 nsec clock cycle	=>	500 MHz clock rate
1 nsec (10^{-9}) clock cycle	=>	1 GHz (10^9) clock rate
200 psec clock cycle	=>	5 GHz clock rate

- 프로세서에 의해 수행되는 연산들은 시스템 클록에 의해 통제

□ 명령어 당 사이클 수(Cycles Per Instruction, CPI)

- 하나의 명령어를 수행하기 위해서 소요되는 클럭 사이클 수

- CPI_i : 명령어 유형 i 를 위해 필요한 사이클 수
- I_i : 주어진 프로그램에서 실행된 유형 i 의 명령어들의 수
- I_c : 주어진 프로그램에서 실행된 명령어들의 수

- 주어진 프로그램을 수행하는 데 필요한 프로세서 시간 T

- τ : 클럭 사이클 시간

- 명령어 실행 및 기억장치 이동시간 등을 고려한 프로세서 시간 T

- p : 명령어를 해독하고 실행하는 데 필요한 프로세서 사이클의 수
- m : 필요한 기억장치 참조 수
- k : 기억장치 사이클 시간과 프로세서 사이클 시간 간의 비율

□ 명령어 수: 하나의 프로그램을 수행하기 위해서 필요한 명령어 개수

- 명령어 세트의 예

$A = B + C$

// 고급 언어

add mem(B), mem(C), mem(A)

// CISC 명령어 세트

load mem(B), reg(1)

// RISC 명령어 세트

load mem(C), reg(2)

add reg(1), reg(2), reg(3)

store reg(3), mem(A)

예제

1. 같은 명령어 집합 구조를 구현한 컴퓨터 두 종류가 있다. 컴퓨터 A의 클럭 사이클 시간은 250ps이고 어떤 프로그램에 대한 CPI는 2.0이다. 컴퓨터 B의 클럭 사이클 시간은 500ps이고, CPI는 1.2이다. 이 프로그램에 관해서는 어떤 컴퓨터가 얼마나 더 빠른가?
2. 어떤 컴파일러 설계자가 같은 상위 수준 언어 문장에 대해 생성된 두 가지 코드 1과 2 중 하나를 선택하려고 한다. 어떤 코드가 더 많은 명령어를 실행하는가? 어떤 코드가 더 빠른가? 각 코드의 CPI는 얼마인가?

명령어 유형	A	B	C
명령어 유형별 CPI	1	2	3
코드 1	2	1	2
코드 2	4	1	1

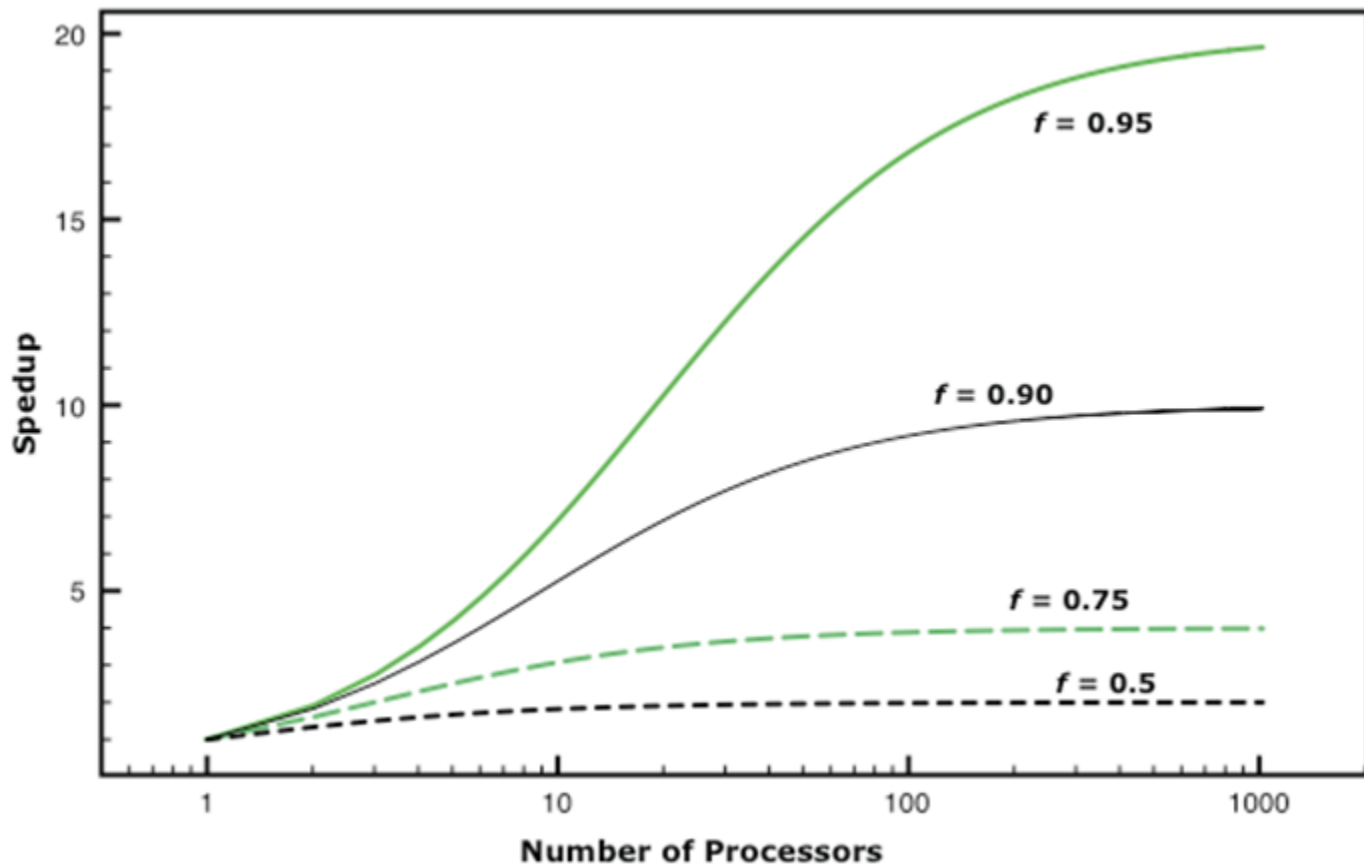
3. 2GHz 클럭의 컴퓨터 A에서 10초에 수행되는 프로그램이 있다. 이 프로그램을 6초 동안에 실행할 컴퓨터 B를 설계하고자 한다. 클럭 속도는 얼마든지 빠르게 만들 수 있는데, 이렇게 하면 CPU 다른 부분의 설계에 영향을 미쳐 같은 프로그램에 대해 A보다 1.2배 많은 클럭 사이클이 필요하게 된다고 한다. 컴퓨터 B의 클럭 속도는 얼마로 해야 하겠는가?

암달의 법칙(Amdahl's law)

- 단일 프로세서와 비교하여 _____한 프로그램의 잠재적 속도 향상 정도를 다룬
- N개의 프로세서를 가진 병렬 프로세서를 사용하여 그 프로그램의 병렬 부분을 완전히 이용하는 경우에 얻을 수 있는 속도 향상

$$Speedup = \frac{\text{time to execute program on a single processor}}{\text{time to execute program on } N \text{ parallel processors}}$$

- T : 특정 프로그램을 단일 프로세서를 이용하여 처리하는데 걸리는 시간
- N : 병렬 프로세서의 개수
- f : 한 프로그램에서 스케줄링 오버헤드 없이 무한히 병렬화할 수 있는 코드의 비율



다중프로세서에 대한 암달의 법칙

암달의 법칙 적용 예제

1. 부동소수점 연산의 수행 시간이 50%를 차지하는 프로그램이 있을 때, 다음 물음에 답하라.

- 1) 부동소수점 연산을 위해 하드웨어를 2배 빠르게 만든다면 수행 시간이 얼마나 향상되는가?

- 2) 이 프로그램을 2배 빠르게 수행하려면 부동소수점 연산을 위한 하드웨어를 얼마나 개선해야 하는가?

2. 어떤 프로세서 칩이 응용 프로그램을 위해 사용되고 있는데, 이 프로그램의 수행 시간의 30%는 부동 소수점 덧셈에, 25%를 부동 소수점 곱셈에, 그리고 10%는 부동 소수점 나눗셈에 소요한다. 이 프로세서의 새로운 모델을 위해 설계팀은 세 가지의 가능한 향상 방식을 고안해 냈는데, 이 각각은 설계 노력과 제조에 거의 같은 비용이 든다. 이들 향상 방식 중에 어느 하나를 선택해야 하는가?

- 1) 부동 소수점 덧셈기를 2배 빠르게 설계한다
- 2) 부동 소수점 곱셈기를 3배 빠르게 설계한다
- 3) 부동 소수점 나눗셈기를 10배 빠르게 설계한다

프로세서의 성능을 나타내는 척도

❑ MIPS(millions of instructions per second)

- 초당 백만 명령어 수

$$\text{MIPS} = \frac{I_c}{T \times 10^6} = \frac{f}{\text{CPI} \times 10^6}$$

I_c : 명령어 개수
 T : 실행 시간
 f : 클럭 속도 ($=1/\tau$)

❑ MFLOPS(millions of floating point operations per second)

- 부동소수점 성능

= 한 프로그램에서 수행되는 부동소수점 연산들의 수 / (수행시간* 10^6)

벤치마크(benchmark)

❑ 기계 성능의 상대적 평가를 위해 선택되거나 설계된 프로그램

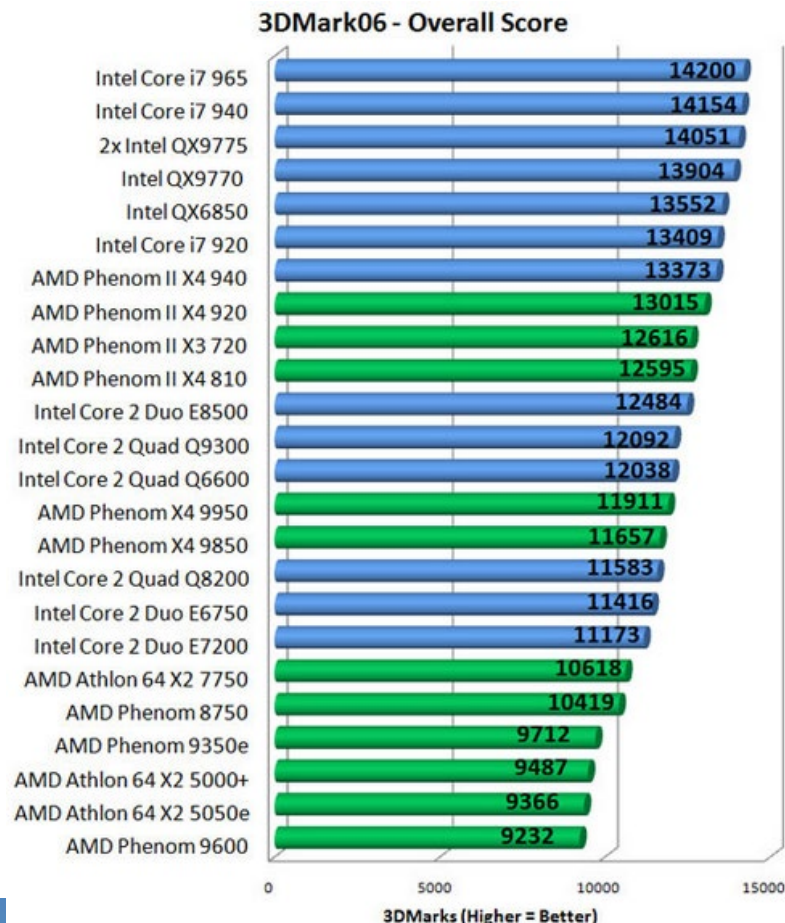
❑ 벤치마크 프로그램의 바람직한 특성

- 고급 언어로 작성되며, 서로 다른 기계들에서 호환성을 가져야 한다
- 특정 종류의 유형에 대표적이어야 한다
- 쉽게 측정되고 널리 보급될 수 있어야

❑ SPEC(System Performance Evaluation Corporation)

- 참조모델(reference model)을 이용하여 각 벤치마크 프로그램에 대한 기본 처리 시간을 정의한다
- SPEC CPU2017, SPECweb2009

CPU 벤치마크 테스트 결과 예





게으름이 사람으로 깊이 잠들게 하나니
태만한 사람은 주릴 것이니라(잠언

19:15)