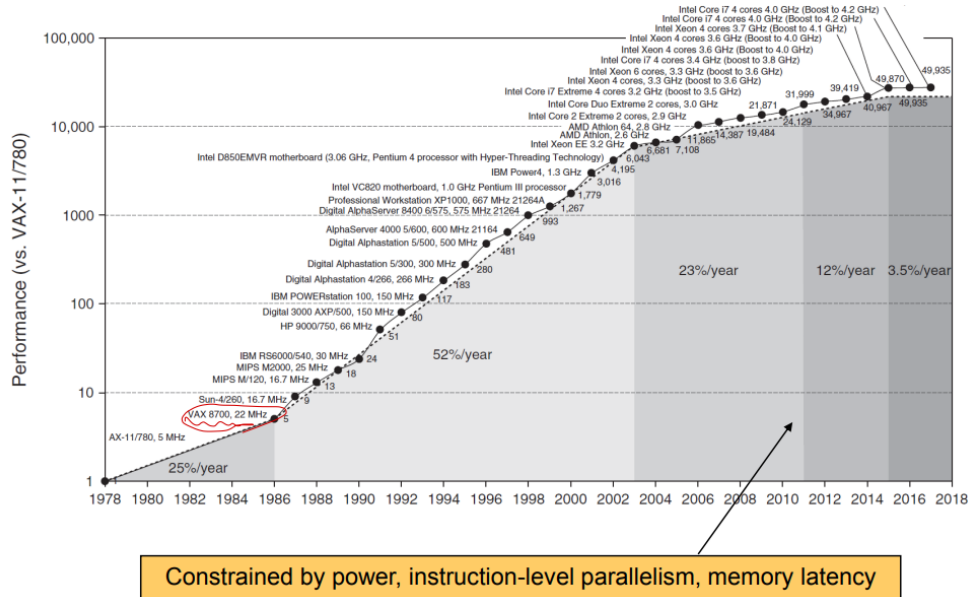




◦ 더 이상 전압을 줄일 수 없고 더 단열할 수 없음 → 어떻게 성능 향상?

→ core를 더 넣기!

## ▼ 1.8 Multiprocessors



- Multicore microprocessors → more than one processor per chip (여러개의 core)  
(size)  
◦ 더 많은 집적회로 → 처리량 개선에 효과 (memory)  
◦ no more free-lunch → 병렬 프로그램 작성 필요성이 생김
  - instruction level의 병렬성과 비교
    - h/w : 한 번에 여러 개의 instruction 실행
    - instruction : h/w와 다르게 programmer에게 눈에 띄지 않음
  - hard to do
    - 성능을 위해 programming → 프로그래밍 어려움
    - load balancing → 스케줄링, 조정하는 오버헤드가 작아야 함
    - optimizing communication and synchronization

## ▼ 1.9 실례 : intel core i7 벤치마킹

### ▼ SPEC CPU Benchmark

- **workload** : 실행시키는 프로그램 집합
  - 두 컴퓨터 시스템을 평가할 때 같은 workload의 실행시간 비교
  - ⇒ but, 대부분의 사용자는 이런 상황에 있지 않다.

→ 새로운 성능 평가 방법으로 자기 workload에 대한 컴퓨터 성능을 기대

⇒ Benchmark : 성능을 측정하기 위해 선택된 프로그램 집합

- **Benchmark** : 사용자의 실제 workload에 대한 성능을 잘 반영할 것이라고 생각함

- 자주 생기는 일은 더 빠르게 하기 위함 → 중요한 역할

- **SPEC** : 표준 벤치마크를 만들어 컴퓨터 시스템 성능 증진을 도움

- SPEC 2017 : 정수 벤치마크 10개 + 부동 소수점 벤치마크 13개

- SPECratio : 정규화한 결과 → 클수록 성능이 더 좋은 컴퓨터

⇒ SPECratio를 기하평균해서 SPEC 2017의 요약값을 구함

$$\sqrt[n]{\prod_{i=1}^n \text{Execution time ratio}_i}$$

↓  
작업보다 n개의 program 중  
첫번째 program .

## ▼ 1.11 오류 및 함정

▼ Amdahl's Law → 고정된 problem 사이즈에서만 적용가능

- pitfall :

- 컴퓨터의 한 부분만 개선 → 개선된 양에 비례 → 전체 성능이 좋아질 것이라고 기대 (속도만큼 좋아지는 것 X)

- 개선 후의 실행 시간 구하기 → 기술에 영향을 받는 부분과 받지 않는 부분이 존재

$$T_{old} = T_{affected} + T_{unaffected}$$

$$T_{new} = \frac{T_{affected}}{\text{improvement factor}} + T_{unaffected}$$

→ 성능향상분  
⇒ 해당부분만 줄어든다!

⇒ 특정부분 (성능향상분)에 집중해서 시간↓ ⇒ 성능

\*영향을 받지 않는 부분  $\Rightarrow 1 - \text{frac}_{\text{affected}}$

$$\text{frac}_{\text{affected}} = \frac{T_{\text{affected}}}{T_{\text{old}}}$$

영향을 받고 있는 부분이 전체에서 어느 정도?

$$T_{\text{new}} = T_{\text{old}} \times \left( (1 - \text{frac}_{\text{affected}}) + \frac{\text{frac}_{\text{affected}}}{\text{speedup}_{\text{affected}}} \right)$$

시간이 바뀌지 X  
변하지 X  
시간이 점점 ↓  
변함

$$\text{SpeedUp} = \frac{T_{\text{old}}}{T_{\text{new}}} = \frac{1}{(1 - \text{frac}_{\text{affected}}) + \frac{\text{frac}_{\text{affected}}}{\text{speedup}_{\text{affected}}}}$$

변하지 않는 부분 발생  $\rightarrow$  고정 상수.

#### • example

##### ◦ assumption:

$$T_{\text{old}} = 80 + 20 = 100$$

- 80초의 multiplication 시간과 함께 100초 동안 프로그램이 실행된다고 가정

1. 4배 빠르게 하기 위해서는 multiplication 속도를 얼마나 올려야 함?

$$T_{\text{new}} = \frac{100}{4} = 25 = 5(\text{multiplication}) + 20(\text{no affect})$$

$$\text{speed} = \frac{80}{5} = 16 \text{ times}$$

2. 5배 빠르게  $\rightarrow$  multiplication time이 0이어야 함  $\rightarrow$  불가

### ▼ Low Power at Idle $\Rightarrow$ power of CPU 성능은 비례가 X

- **fallacy** : 이용률이 낮은 컴퓨터는 전력 소모가 낮다
  - 서버는 workload가 가변적  $\rightarrow$  이용률 낮을 때의 전력 효율
  - 에너지에 비례하는 컴퓨팅  $\rightarrow$  hw 재설계 필요
- **fallacy** : 성능에 초점을 둔 설계와 에너지 효율에 초점을 둔 설계는 무관하다

### ▼ MIPS as a performance metric

- **pitfall** : performance equation  $\rightarrow$  performance metric의 일부로 활용하는 것
- MIPS : Millions of Instructions Per second  $\rightarrow$  초당 몇백만개의 명령어?

$$\text{MIPS} = \frac{\text{Instruction count}}{\text{Execution time} \times 10^6}$$

$\rightarrow$  millions.

$$= \frac{\text{Instruction count}}{\frac{\text{Instruction count} \times \text{CPI}}{\text{Clock rate}} \times 10^6} = \frac{\text{Clock rate}}{\text{CPI} \times 10^6}$$

성능 지표가 결정됨.

$$\text{Execution time} = \frac{\text{Instruction Count} \times \text{CPI}}{\text{Clock rate}}$$

문제점

1. ISA가 달라지면 instruction 개수도 달라지기에 MIPS 값으로 성능 비교 어려움

$$CPU\ Time = \frac{IC \times CPI}{clock\ rate} = \frac{5+2+3}{4 \times 10^9}$$

2. 같은 컴퓨터에서도 프로그램에 따라 MIPS 값이 다르다

$$MIPS = \frac{clock\ rate}{CPI \times 10^6}$$

$clock\ rate = 4 \times 10^9$   
 $IC_1 = 7 \times 10^9$   
 $IC_2 = 12 \times 10^9$

example

Two different compilers are being tested for a 4 GHz. machine

▶ With three different classes of instructions

Inst. type	Class A	Class B	Class C
CPI	1	2	3

▶ Both compilers are used to produce code for a large piece of software.

Inst. Type	Class A	Class B	Class C
The 1st compiler's code	5 billion	1 billion	1 billion
The 2nd compiler's code	10 billion	1 billion	1 billion

total  
→ 7 billion  
→ 12 billion

1) MIPS = 4GHz, ⇒ 몇배 더 빠름?

$$IC_1 = 7 \times 10^9$$

$$CPI_1 = \frac{(1 \times 1 + 2 \times 1 + 3 \times 1) \times 10^9}{7 \times 10^9} = \frac{10}{7} = \frac{40}{28}$$

$$IC_2 = 12 \times 10^9$$

$$CPI_2 = \frac{(1 \times 10 + 2 \times 1 + 3 \times 1) \times 10^9}{12 \times 10^9} = \frac{15}{12} = \frac{5}{4} = \frac{35}{28}$$

$$CPI = \frac{cycles.}{IC}$$

$$\therefore MIPS_1 = \frac{4 \times 10^9}{\frac{40}{28} \times 10^6} = \frac{28 \times 10^9}{10^7} = 2800$$

$$MIPS_2 = \frac{4 \times 10^9}{\frac{5}{4} \times 10^6} = \frac{16}{5} \times 10^3 = 3200$$

→ 2배 더 빠름

$$2) MIPS = \frac{IC}{Exe\ time \times 10^6}$$

$$Ex\ time = \frac{IC}{MIPS \times 10^6} = \frac{IC}{\frac{clock\ rate}{CPI \times 10^6} \times 10^6}$$

$$= \frac{CPI \times IC}{clock\ rate}$$

$$Ex_1 = \frac{10}{7} \times 7 \times 10^9 \times \frac{1}{4 \times 10^9} = 2.5$$

$$Ex_2 = \frac{5}{4} \times 12 \times 10^9 \times \frac{1}{4 \times 10^9} = \frac{15}{4} \approx 3.5$$

∴ Ex : 10배 더 빠름, (CPI가 작다고 무조건 좋은건 X)