



# Ch.1-2 Computer Abstractions & Technology

## ▼ 1.6 Performance

- 고려해야 할 사항
  1. 왜 다른 프로그램에 대해 더 나은 하드웨어가 있는가?
  2. 하드웨어와 관련된 시스템 성능에는 어떠한 요인이 있는가?
  3. 어떻게 machine의 instruction이 성능에 영향을 주는가?

- depends on

Algorithm: affects IC, possibly CPI

Programming language: affects IC, CPI

Compiler: affects IC, CPI

Instruction Set Architecture: affects IC, CPI,  $T_c$

⇒ 성능은 어떤 환경에서 보느냐에 따라 다르게 측정됨.

## ▼ response time & throughput

- **response time** : 어떤 일로 결과를 만들어 내는 시간 (task로 처리하는데 걸리는 시간)
  - 컴퓨터의 processor를 더 빠른 버전으로 바꿈 → response time이 빨라짐
- **throughput** : 단위 시간 당 response time → 주어진 시간 동안 얼마나 많은 일을 처리?
  - processor를 더 많이 추가함 → throughput이 좋아짐

## ▼ Relative Performance 성능의 차이는 상대적으로 보게됨.

**performance = 1 / Execution time**

⇒ 컴퓨터의 성능은 실행 시간에 반비례

⇒ X is n time faster than Y

80%보다 빠른 때

$$\frac{Performance_X}{Performance_Y} = \frac{ExecutionTime_Y}{ExecutionTime_X} = n$$

- example
  - A : 10초, B : 15초
  - $E_B / E_A = 15/10 = 1.5$
  - Computer A : B보다 1.5 배 빠름  
↳ 1.5배

## ▼ Measuring Execution Time → 성능 측정하기

- **Elapsed time**(wall clock time, response time) programming 시에 들어가는 시간) → 시작부터 종료까지의 총 소요시간

- total response time, including all aspects(processing, I/O, OS overhead, idle time)
- determines system performance

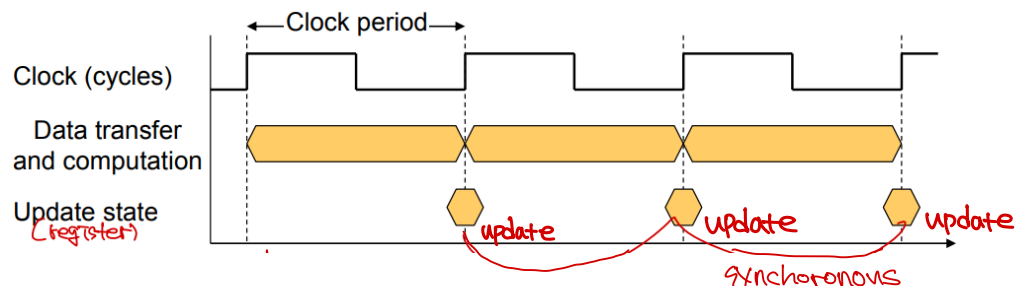
↳ 실행할 때마다  
컨디션에 따라 다른 수행시간을 지님.

## • CPU time = user CPU time + system CPU time

- 주어진 작업을 수행하는데 걸리는 시간 (not include I/O, other job's shares)
  - 각 프로그램은 CPU와 시스템 성능에 각각 영향 받음
  - OS의 각 작업이 어떤 프로그램을 위해서 수행되고 있는지 쉽게 알 수 없음
    - OS 간의 기능 차이도 존재!

## • CPU clocking → cycle.

- 거의 모든 컴퓨터는 hw 이벤트가 발생하는 시점을 clock이 결정
- clock cycle : clock의 시간 간격  
CPU clock cycle : CPU clock의 시간 간격 → CPU 내부에서 event가 발생하는 시점.



1. clock period : 한 clock cycle이 걸리는 시간, (register를 update)

- ex. 250ps = 0.25ns = 250×10<sup>-12</sup>s ⇒ 4GB 정도의 clock speed

KB 10<sup>3</sup> ≐ 2<sup>10</sup>  
 MB 10<sup>6</sup> ≐ 2<sup>20</sup>  
 GB 10<sup>9</sup> ≐ 2<sup>30</sup>  
 TB 10<sup>12</sup> ≐ 2<sup>40</sup>  
 Peta 10<sup>15</sup>

(Clock 속도)

2. clock frequency (rate) : cycles per second → clock period의 역수

• ex. 4.0GHz = 4000MHz =  $4.0 \times 10^9$ Hz

⇒ 사용자 cpu 시간으로 계산한 것을 cpu 성능이라고 생각할 것

## ▼ CPU Time

$$\begin{aligned} \textcircled{2} \text{ CPU Time} &= \text{CPU Clock Cycles} \times \text{Clock Cycle Time} (= \text{clock period}). \\ &= \frac{\text{CPU Clock Cycles}}{\text{Clock Rate}} \end{aligned}$$

clock 간격      clock 걸리는 시간  
clock 간격      clock 속도

• 컴퓨터의 성능 up ⇒ 어떻게?

1. clock 개수 down, 시간 down
2. clock speed up

• example

- **Computer A** : 2GHz clock, 10s CPU
- **Computer B** :  $1.2 \times \text{Clock Cycles}_A$ , 6s CPU

⇒ B의 clock이 더 빠른가?

$$\textcircled{A} \text{ Clock Cycles}_A = \text{CPU Time}_A \times \text{Clock Rate}_A$$

$$= 10 \times 2 \times 10^9 = 2 \times 10^{10}$$

$$\textcircled{B} \text{ Clock Rate}_B = \frac{\text{Clock Cycles}_B}{\text{CPU Time}_B} = \frac{1.2 \times \text{Clock Cycles}_A}{6}$$

$$= \frac{1.2 \times 10 \times 2 \times 10^9}{6} = 4 \times 10^9 = 4\text{GHz}$$

∴ B가 A보다 2배 빨라야 한다

## ▼ 명령어(instruction) 성능 and CPI(clock cycles per instructions)

실행시간 ⇒ 명령어와 큰 관련!

①  $\text{Clock Cycles} = \text{Instruction Count} \times \text{Cycles per Instruction}$  (Instruction 개수 × 명령어당 평균 사이클 수)

②  $\text{CPU Time} = \text{Instruction Count} \times \text{CPI} \times \text{Clock Cycle Time}$   
 $= \frac{\text{Instruction Count} \times \text{CPI}}{\text{Clock Rate}}$  (명령어당 clock cycle 수, 명령어에 따라서 다르다!)

- ① 명령어 개수
- instruction count  $\Rightarrow$  ISA(instruction set architecture), compiler에 의해서 달라짐
  - CPI(명령어 당 평균 clock cycles) ② 명령어당 clock cycle 수
    - 각 명령어가 다른 CPI를 가짐  $\Rightarrow$  명령어 조합에 따라 달라질 수 있음(필연x)
  - example  $\rightarrow$  같은 ISA 가정
    - Computer A : Cycle Time = 250ps, CPI = 2.0
    - Computer B : Cycle Time = 500ps, CPI = 1.2

$$\begin{aligned} \text{CPU Time}_A &= \text{Clock Cycles}_A \times \text{Clock Cycle Time} \\ &= \text{Instruction Count} \times \text{CPI}_A \times \text{Clock Cycle Time}_A \\ &= I \times 2 \times 250\text{ps} = I \times 500\text{ps} \end{aligned}$$

$$\text{CPU Time}_B = I \times 1.2 \times 500\text{ps} = I \times 600\text{ps}$$

$$\text{Speedup} = \frac{\text{CPU Time}_B}{\text{CPU Time}_A} = 1.2 \quad \therefore A \text{가 } 1.2\text{배} \text{ 빠름?}$$

(2배 빠를 것이라고 기대했음)

- instruction
  - 다른 instruction  $\rightarrow$  다른 개수의 cycle

①  $\text{Clock Cycles} = \sum_{i=1}^n (\text{CPI}_i \times \text{Instruction Count}_i)$   $\rightarrow T \text{ time에 실행되는 총 cycle 명령어}$

- weighted average CPI  $\Rightarrow$  가중치를 이용한 평균

$$\text{CPI} = \frac{\text{Clock Cycles}}{\text{Instruction Count}} = \sum_{i=1}^n \left( \text{CPI}_i \times \frac{\text{Instruction Count}_i}{\text{Instruction Count}} \right)$$

$\rightarrow T \text{ time Count}$   
 $\rightarrow \text{total Count}$

$$\text{CPI} = \frac{\text{CPU Time} \times \text{Clock Rate}}{\text{Instruction Count}}$$

Relative frequency  $\Rightarrow$  빈도끼리  
다 더하면 1이 됨

$$\text{CPU Time} = \text{Clock Cycle Time} \times \sum_{j=1}^n \text{CPI}_j \times \text{IC}_j$$

	CPI for each instruction class		
	A	B	C
CPI	1	2	3

total 5 6	Code sequence	Instruction counts for each instruction class		
		A	B	C
	1	2 $\frac{2}{6}$	1 $\frac{1}{6}$	2 $\frac{2}{6}$
	2	4 $\frac{4}{6}$	1 $\frac{1}{6}$	1 $\frac{1}{6}$

[1회] Clock Cycle =  $1 \times 2 + 2 \times 1 + 3 \times 2 = 10$

Avg. CPI =  $1 \times \frac{2}{6} + 2 \times \frac{1}{6} + 3 \times \frac{2}{6} = \frac{10}{6} = 2.0$

[2회] Clock Cycle =  $1 \times 4 + 2 \times 1 + 3 \times 1 = 9$

Avg. CPI =  $1 \times \frac{4}{6} + 2 \times \frac{1}{6} + 3 \times \frac{1}{6} = \frac{9}{6} = 1.5$

∴ 2번이 더 빠르다.

- example : calculating CPI bottom up

- Base Machine(Reg / Reg)

Operation	Cycles	Frequency	CPI <sub>i</sub>	Time(%)
ALU	1	50% 0.5	= 0.5	33% = $\frac{0.5}{1.5}$
Load	2	20% 0.2	= 0.4	27%
Store	2	10% 0.1	= 0.2	13%
Branch	2	20% 0.2	= 0.4	27%

→ 전체 프로그램에서  
해당 operation이  
얼마나?

Average CPI total : 1.5

Typical Mix of instruction types in a program clock cycle.

→ frequency

• Time = Instruction Count / 1.5

$CPI = Cycles \times \boxed{\text{Instruction Count}}$

→ Frequency weighted average.

total CPI = 1.5

- example : Branch Stall Impact ⇒ instruction이 판단-실행까지 걸리는 시간

- 가정

1. Base Machine : CPI = 1.0 (ideal) ignoring branch → 한 사이클에 하나

2. A branch instruction is stalling for 3 cycles → 3 cycle마다 한 번

- instruction 중 30%가 branch일 때, branch instructions이 3 cycle  
70% ideal

Operation	Cycles	Frequency	CPI(i)	Time(%)
Others	① → CPI	70%	= 0.7	37%
Branch	④ CPI	30%	= 1.2	63%

↓  
1.9 (평균 CPI) > 1.0 (ideal)  
→ 성능 저하

- New CPI = 1.9

- New machine is  $1/1.9 = 0.52$  times faster(시간이 두 배 정도 늘어남)

↓  
ideal 변경

## The BIG Picture

clock speed가 무조건 빠르게 바뀐다 \*  
→ 코드가 달라져 다른 수 있음

$$\text{CPU Time} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Clock cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Clock cycle}}$$

$$\text{CPI} = \frac{\text{clock cycles}}{I, C}$$

$$= \frac{\text{CPU Time} \times \text{Clock Rate}}{I, C}$$

$$\text{CPU Time} = \text{cycle Time} \times \sum_{i=1}^n \text{CPI}_i \times \text{IC}_i$$

$$\text{CPI} = \frac{\sum_{i=1}^n \text{CPI}_i \times \text{IC}_i}{\text{IC}}$$

↓  
 $f_i$

→ 5 cycle A