컴퓨터 구조 HW 1

2019320045 지동환

1

> Write the eight great ideas in computer architecture research mentioned during the class.

- Design for *Moore's Law*
- Use *abstraction* to simplify desing
- Make the common case fast
- Performance via *parallelism*
- Performance via *pipelining*
- Performance via *prediction*
- *Hierarchy* of memories
- *Dependability* via redundancy (for reliability)

2

> **Student Y** stated that the performance of the ARM processor using 2 GHz clock exhibits higher performance than the x86 Pentium processor that runs with 1.5 GHz clock. Explain why the statement by **Student Y** is not always true. Please take a counter example in your answer.

둘의 ISA가 다르기 때문이다. ARM cpu와 x86 Pentium을 구성하는 Instruction set에는 차이가 있다. 특히 x86은 CISC 구조로서 상대적으로 많은 Instruction set으로 구성되어 있다. 그렇기 때문에 두 개의 CPU에서 같은 작업을 시행하더라도, x86에서 실행해야 하는 instructions의 개수가 ARM에서 실행해야 하는 양보다 적거나 같다. 그렇기 때문에, ARM processor가 비록 높은 clock rate를 가지고 있지만 x86에서 실행하는 instructions 수가 더 적기 때문에 속도는 x86이 더 높은 performance를 보여 줄 수도 있다.

3

> Assume a color display using 8-bits for each of primary colors (red, green, blue) per pixel and a frame size is 1920 × 1080.

(a)

> What is the minimum size in bytes of the frame buffer to store a single frame?

한 pixel에 R G B 세 가지 색이 있고, 각각 1 byte를 사용한다. 즉, 3 byte / pixel.

frame의 크기가 1920 × 1080이므로, single frame을 저장하는 buffer의 최소 크기는 3 × 1920 × 1080 이다.

(b)

> Assume that you are requested to design a display interface that can transfer 60 frames per second. Assuming a single frame has a 1920 x 1080 resolution of 8-bits R/G/B pixels, calculate the minimum bandwidth of this display interface. Write your answer using MB/s. (1 KB = 1024 byte, 1 MB = 1024 KB)

초당 60개의 frame을 저장해야 한다. display interface의 최소 bandwidth는, 60 * 3 * 1920 * 1080 B/s = 364500 KB/s = 355.96 MB/s 이다.

4

> Consider three different processors P1, P2, and P3 executing the same instruction set. P1 has 2.4GHz clock rate and a CPI of 1.2. P2 has a 3.0GHz clock rate and CPI of 1.4. P3 has a 4.0GHz clock rate and has a CPI of 2.2.

(a)

> Which processor has the highest performance expressed in **instructions per second**?

instructions per second의 관점에서 high performance => 단위 시간 동안 더 많은 instruction을 실행한다.

CPU Time = Instruction Count x CPI / Clock Rate.

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

instruction per second를 알고 싶기 때문에 CPU Time이 1로 모두 동일하다고 생각하면,

$\text{Instruction Count per second}_{P_1} = \frac{2.4\text{GHz}}{1.2}$

$\text{Instruction Count per second}_{P_2} = \frac{3.0\text{GHz}}{1.4}$

$\text{Instruction Count per second}_{P_3} = \frac{4.0\text{GHz}}{2.2}$

instruction per second는 $P_2 > P_1 > P_3$이다. instruction per second가 클 수록 performance가 좋으므로,

Instruction per second의 관점에서 Performance의 순서는 $P_2 > P_1 > P_3$이다.

(b)

> If the processors each execute a program in 10 seconds, find the **number of cycles** and the **number of instructions**.

$\text{Clock Cycles} = \text{Clock Rate} \times \text{CPU time}$

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

이다.

$\text{Clock Cycles}_A = \text{Clock Rate}_A \times \text{CPU time} = 2.4\text{GHz} \times 10 = 2.4 \times 10^{10}$

$\text{Clock Cycles}_B = \text{Clock Rate}_A \times \text{CPU time} = 3.0\text{GHz} \times 10 = 3.0 \times 10^{10}$

$$\text{Clock Cycles}_A = \text{Clock Rate}_A \times \text{CPU time} = 4.0\text{GHz} \times 10 = 4.0 \times 10^{10}$$

$$\text{Instruction Count}_A = \frac{\text{CPU Time} \times \text{Clock Rate}_A}{\text{CPI}_A} = \frac{10 \times 2.4\text{GHz}}{1.2} = 2.0 \times 10^{10}$$

$$\text{Instruction Count}_B = \frac{\text{CPU Time} \times \text{Clock Rate}_B}{\text{CPI}_B} = \frac{10 \times 3.0\text{GHz}}{1.4} = 2.14 \times 10^{10}$$

$$\text{Instruction Count}_C = \frac{\text{CPU Time} \times \text{Clock Rate}_C}{\text{CPI}_C} = \frac{10 \times 4.0\text{GHz}}{2.2} = 1.82 \times 10^{10}$$

5

> Computer A can execute a C program 10 times in one second and Computer B can execute the same C program 20 times in one second. If the MIPS (million instructions per second) rate of Computer A is MIPS_A and MIPS rate of Computer B is MIPS_B, then an engineer concludes that MIPS_B = MIPS_A x 2. Under what conditions is this calculation correct?

MIPS = Milions of Instructions Per Second

Computer A와 B는 같은 프로그램을 실행시킨다.

그러나 A와 B의 ISA가 다를 수도 있기 때문에, 실행되는 instruction이 B가 2배 많다고 단정 지을 수 없다.

즉, A와 B의 ISA가 같다는 상태이면 가능하다.

6

> Assume that for a certain program compiler A results in a dynamic instruction count of 1.1×10^9 and has an execution time of 1.2 sec, while compiler B results in a dynamic instruction count of 1.5×10^9 and an execution time of 1.7 sec.

(a)

> (a) Find the average CPI for each program given that the processor has a clock cycle time of 1 ns.

Compiler A :

clock time = 1ns. excution time = 1.2s. Clock Cycles = 1.2s / 1ns = 1.2×10^9

average CPI of A = Clock Cycles / Instruction Count = $\frac{1.2 \times 10^9}{1.1 \times 10^9} = \frac{12}{11} \simeq 1.09$

Compiler B :

clock time = 1ns. excution time = 1.7s. Clock Cycles = 1.7s / 1ns = 1.7×10^9

average CPI of B = Clock Cycles / Instruction Count = $\frac{1.7 \times 10^9}{1.5 \times 10^9} = \frac{17}{15} \simeq 1.13$

(b)

> Assume that the compiled programs run on two different processors **X** and **Y**. If the execution times on the two processors are the same, how much faster is the clock of the **processor Y** running **compiler B**'s code versus the clock of the **processor X** running **compiler A**'s code? Assume that the processors have the same microarchitecture deploying the same ISA.

execution time_X = execution time_Y = t

instruction count_A = 1.1×10^9 , instruction count_B = 1.5×10^9

execution time = clock count * CPI * instruction count = CPI * instruction count / clock rate

Clock Rate_Y = k * Clock Rate_X. 상수 k의 값을 알고 싶다.

A,B의 execution time은 같고, processor X, Y의 ISA가 같고, compiler A, B의 average CPI도 알 수 있다.

CPI_A * instruction count_A / Clock rate_A = CPI_B * instruction count_B / Clock rate_B

$$\frac{\frac{12}{11} * 1.1 \times 10^9}{Clock\ rate_A} = \frac{\frac{17}{15} * 1.5 \times 10^9}{Clock\ rate_B}$$

$$\frac{Clock\ rate_B}{Clock\ rate_A} = \frac{\frac{17}{15} * 1.5 \times 10^9}{\frac{12}{11} * 1.1 \times 10^9}$$

$$= \frac{17}{12} \simeq 1.42$$

(c)

> A new compiler is developed that uses only 6.0×10^8 instructions and has average CPI of 1.1. What is the speedup of using this new compiler versus using compiler A or B on the original processor?

CPU Time = Instruction count * CPI / Clock rate

CPU Time_new = $6.0 \times 10^8 * 1.1 / Clock\ rate_{original}$

CPU Time_A = $1.1 \times 10^9 * \frac{12}{11} / Clock\ rate_{original}$

CPU Time_B = $1.5 \times 10^9 * \frac{17}{15} / Clock\ rate_{original}$

새로운 컴파일러는 A보다 $\frac{6.0 \times 10^8 * 1.1}{1.1 \times 10^9 * \frac{12}{11}} = \frac{66}{120}$배 빠르게 실행된다. 즉, $\frac{120}{66} = 1.82$ 배 빠르게 실행된다.

새로운 컴파일러는 B보다 $\frac{6.0 \times 10^8 * 1.1}{1.5 \times 10^9 * \frac{17}{15}} = \frac{66}{170}$배 빠르게 실행된다. 즉, $\frac{170}{66} = 2.58$ 배 빠르게 실행된다.

7

> Consider two different implementations of **the same instruction set architecture**. The instructions can be divided into four classes according to their CPI (classes A, B, C, and D). Consider the following two processors and an application.

> **P1: Clock frequency = 2.0GHz, CPIs for each instruction class = 1, 2, 3, 3**
>
> **P2: Clock frequency = 3.0GHz, CPIs for each instruction class = 2, 2, 2, 2**
>
> **Application:**
>
> **Instruction count =** 1.0×10^6**,**
>
> **fractions by instruction classes: class A = 20%, class B = 10%, class C = 40%, class D = 30%**

(a)

> Which processor is faster : P1 or P2?

Clock Cycles = $\sum_{i=1}^{n}(CPI_i \times Instruction\ Count)$

P1:

Classes	**A**	**B**	**C**	**D**
Instruction Count	$\frac{20}{100} \times 1.0 \times 10^6$	$\frac{10}{100} \times 1.0 \times 10^6$	$\frac{40}{100} \times 1.0 \times 10^6$	$\frac{30}{100} \times 1.0 \times 10^6$
CPI	1	2	3	3
Clock Cycles Count	$1 \times \frac{20}{100} \times 1.0 \times 10^6$	$2 \times \frac{10}{100} \times 1.0 \times 10^6$	$3 \times \frac{40}{100} \times 1.0 \times 10^6$	$3 \times \frac{30}{100} \times 1.0 \times 10^6$
=	2.0×10^5	2.0×10^5	12.0×10^5	9.0×10^5

실행에 필요한 Clock의 총합은 2.5×10^6

CPU Time = Clock Cycles Count / Clock frequency = $\frac{2.5 \times 10^6}{2GHz} = \frac{2.5 \times 10^6}{2 \times 10^9 Hz} = \frac{1.25}{10^3}s = 1.25 \times 10^{-3}s$

P2:

Classes	**A**	**B**	**C**	**D**
Instruction Count	$\frac{20}{100} \times 1.0 \times 10^6$	$\frac{10}{100} \times 1.0 \times 10^6$	$\frac{40}{100} \times 1.0 \times 10^6$	$\frac{30}{100} \times 1.0 \times 10^6$
CPI	2	2	2	2
Clock Cycles Count	$2 \times \frac{20}{100} \times 1.0 \times 10^6$	$2 \times \frac{10}{100} \times 1.0 \times 10^6$	$2 \times \frac{40}{100} \times 1.0 \times 10^6$	$2 \times \frac{30}{100} \times 1.0 \times 10^6$
=	4.0×10^5	2.0×10^5	8.0×10^5	6.0×10^5

실행에 필요한 Clock의 총합은 2.0×10^6

CPU Time = Clock Cycles Count / Clock frequency = $\frac{2.0 \times 10^6}{3GHz} = \frac{2.0 \times 10^6}{3 \times 10^9 Hz} \simeq \frac{0.67}{10^3}s = 0.67 \times 10^{-3}s$

그러므로, P2가 P1보다 빠르다.

(b)

> What is the global CPI for each implementation?

CPI = Clock Cycles / Instruction Count

P1 :

$$CPI_{P1} = \frac{2.5 \times 10^6}{1.0 \times 10^6} = 2.5$$

P2 :

$$CPI_{P2} = \frac{2.0 \times 10^6}{1.0 \times 10^6} = 2$$

가 된다.

(c)

> Figure out the clock cycles required in both cases.

이는 (a)에서 이미 계산 했었다.

Clock Cycles_P1 : 2.5×10^6 , Clock Cycles_P2 : 2.0×10^6 이다.

8

> An enhancement is proposed to a computer (called the **baseline**). The enhancement merges some commonly occurring instructions into a single instruction. But this enhancement increases the clock cycle time by 20% for all instructions. Assume that 80% of the instructions in the baseline computer are merged into just 40%. (That means 2 instructions are merged into one instruction.) Furthermore, assume that in the baseline machine the CPI (clock per instruction) of the instructions that **cannot** be merged is 50% higher than the CPI of the instructions that **can** be merged. What is the speedup of the enhanced computer over the baseline?

baseline의 Instruction Set을 두 가지, A 와 B로 나눌 수 있다. A는 merge 될 수 있고, B는 없다.

A의 instruction count는 절반으로 줄었다. B는 유지된다.

Clock Cycle time은 baseline의 1.2배가 된다.

B의 CPI는 A의 CPI보다 1.5배이다. CPI_B = 1.5 * CPI_A

이를 바탕으로 계산하면,

CPU Time_baseline = Clock Cycles Time_baseline × (CPI_B × 0.2 * Instruction Count_baseline + CPI_A × 0.8 * Instruction Count_baseline)

= 1.1 * Clock Cycles Time_baseline × CPI_A × Instruction Count_baseline

CPU Time_new = (1.2 * Clock Cycles Time_baseline) × (CPI_B × (0.2 * Instruction Count_baseline) + CPI_A × 0.4 × Instruction Count_baseline)

= 0.84 * Clock Cycles Time_baseline × CPI_A × Instruction Count_baseline

새로운 CPU의 실행 시간은 0.84배가 되었다. => performance는 $\frac{1.1}{0.84} \simeq 1.31$ 배 증가했다. 31% improved.

9

> Let us assume that 40% of instructions in a certain program can be parallelized. Note that if 100% of instructions is parallelizable, the execution time of this application will be half if running on a dual-core processor.

100% parallelizable하다면 core 하나 당 50%의 instruction을 할당한다. 그러므로 실행해야 하는 instruction의 수가 50%가 된다. 그래서 실행 시간이 절반이 된다. => CPU Time은 정확히 분배 하는 만큼 줄어든다. 다른 영향을 주는 요인이 없다.

(a)

> What are the performance improvements of this application when running on dual-core (2 cores), quad-core (4 cores), and octa-core (8 cores) processors, respectively. Assume that the baseline machine for comparison is a single-core system.

single-core에서 돌아갈 때 instruction의 수 k.

Dual-core : 두 개의 core에 40%의 instruction을 분배한다. 60%는 하나에서만 실행 된다.

=> (0.4/2 + 0.6)k = 0.8k 1.25배 성능 향상

Quad-core : 네 개의 core에 40%의 instruction을 분배한다. 60%는 하나에서만 실행 된다.

=> (0.4/4 + 0.6)k = 0.7k 1.43배 성능 향상

Octa-core : 여덟 개의 core에 40%의 instruction을 분배한다. 60%는 하나에서만 실행 된다.

=> (0.4/8 + 0.6)k = 0.65k 1.54배 성능 향상

(b)

> Assuming the dynamic power consumption of the single-core processor is P, what are the power consumption of the dual-core, quad-core, and octa-core processors, respectively? Use the equation learned during the class and ignore the activity factor.

$\text{Power} = \text{Capacitive load} \times \text{Voltage}^2 \times \text{Frequency}$

각 core가 병렬적으로 연결되어 있다. 각 코어마다 Capacitivie load, Voltage, Frequency가 같다.

dual-core = 2P

quad-core = 4P

octa-core = 8P

(c)

> Energy (= power × time) is the appropriate metric that considers both performance and cost. Namely we can recognize the total of consumed power that is required for executing applications by using energy. Calculate the energy consumption of single-core, dual-core, quad-core, and octa-core processors respectively using A (execution time) and P (power consumption) of the single-core processor. Figure out the best configuration considering the only energy consumption.

single-core : Energy = P × A = PA

dual-core :

(a)에 따라 dual-core에서의 실행 시간은 0.8A, (b)에 따라 dual-core의 전력 소모는 2P이다. Energy = 2P × 0.8A = 1.6PA

quad-core :

(a)에 따라 quad-core에서의 실행 시간은 0.7A, (b)에 따라 dual-core의 전력 소모는 4P이다. Energy = 4P × 0.7A = 2.8PA

ocata core :

(a)에 따라 octa-core에서의 실행 시간은 0.65A, (b)에 따라 dual-core의 전력 소모는 8P이다. Energy = 8P × 0.65A = 5.2PA

그러므로, 전력 소모의 측면에서 single-core가 가장 좋다.

10

> Assume a program requires the execution of 50×10^6 FP instructions, 110×10^6 INT instructions, 80×10^6 load/store instructions, and 16×10^6 branch instructions. The CPI for each type of instruction (FP, INT, load/store, and branch types) is 1, 1, 4, and 2, respectively. Assume that the processor has a 2GHz clock rate.

(a)

> By how much must we improve the CPI of FP instructions if we want the program to run two times faster?

현재 average CPI = $\frac{10^6 \times (50*1 + 110*1 + 80*4 + 16*2)}{(50+110+80+16) \times 10^6} = \frac{512}{256} = 2$

Clock Cycles = $\sum_{i=1}^{n}(CPI_i \times instruction\ count_i) = 1*(50 \times 10^6) + 1*(110 \times 10^6) + 4*(80 \times 10^6) + 2*(16 \times 10^6) = 512 \times 10^6$

CPU Time = Clock Cycles / Clock Rate = $\frac{512 \times 10^6}{2GHz} = \frac{2^9 \times 10^6}{2 \times 10^9} s = 2^8 \times 10^{-3} s = 0.256s$

여기서 2배 더 빨리 실행하고 싶다. => CPU Time을 0.128s로 줄이고 싶다. => Clock Cycles을 256×10^6로 줄인다.

새로운 CPI of FP를 x라고 하자. Clock Cycles을 다시 계산하면,

Clock Cycles = $\sum_{i=1}^{n}(CPI_i \times instruction\ count_i) = x*(50 \times 10^6) + 1*(110 \times 10^6) + 4*(80 \times 10^6) + 2*(16 \times 10^6) = 256 \times 10^6$

$= (50x + 462) \times 10^6$ 이다.

$50x + 462 = 256$을 만족해야 한다. 그러나 CPI는 음수가 될 수 없으므로, FP의 CPI를 improve해도 2배 빠르게 실행하는 것은 불가능하다.

(b)

> By how much is the execution time of the program improved if the CPI of INT and FP instructions is reduced by 40% and the CPI of load/store and branch instructions is reduced by 30%? [6]

Clock Cycles =
$\sum_{i=1}^{n}(CPI_i \times instruction\ count_i) = 0.6 * 1 * (50 \times 10^6) + 0.6 * 1 * (110 \times 10^6) + 0.7 * 4 * (80 \times 10^6) + 0.7 * 2 * (16 \times 10^6) = 342.4 \times 10^6$

CPU Time = Clock Cycles / Clock Rate = $\frac{342.4 \times 10^6}{2GHz} = \frac{342.4 \times 10^6}{2 \times 10^9} s = 171.2 \times 10^{-3} s = 0.1712s$ 로 실행 시간이 줄어든다.

원래 실행 시간은 0.256s. 실행 시간이 약 $\frac{0.1712}{0.256} = 0.66875$배가 된다. performance는 약 1.50배, 50% 좋아졌다.