Computer Architecture
Fall, 2019
Week 2
2019.9.16

組員簽名：＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

[group7] **(對抗賽)**
1. 請問以下敘述何者正確?若是敘述錯誤，請說明何處錯誤。
    (a) Moore's law 說明每過 1 年，一個晶片上的電晶體數目會加倍。
    (b) CPU time = CPU clock cycles/clock rate，故可以同時增加 clock rate 跟減少 clock cycle 的數目來減少 CPU time。
    (c) 從 Amdahl's law 可以得知只能對可升級的部分效能作升級。
    (d) Power = Capacitive load x voltage^2 x Fequency，雖然至 2007 年為止 Fequency 成長了約 1000 倍，但由於科技進步，Voltage 也從原本的 5V 下降至 1V。為了有效降低 Power，我們可以選擇繼續降低 Voltage。

Ans:

      (A) 1.5 年
      (B) 增加 clock rate 跟減少 clock cycle 的數目互相衝突
      (C) 正確
      (D) 現今技術無法再降低 Voltage 太多

[group10]
2. A CPU is processing A、B、C three different type(class) of instruction, the CPI are 2、3、5. Under the same compiler, the CPU processes three programs 甲、乙、丙 and complies them into three Object Code. Followings are the instruction count respectively.

	A	B	C
Object Code 甲	80	40	50
Object Code 乙	90	30	40
Object Code 丙	40	60	50

    1. What is the three Object Code speed order? Rank it from fast to slow.
    2. Assume the cycle time of this CPU is 500ps, what is the CPU time of 甲?

Ans:
    1. CPU clock cycle for 甲 =2*80+3*40+5*50=530
       CPU clock cycle for 乙 =2*90+3*30+5*40=470
       CPU clock cycle for 丙 =2*40+3*60+5*50=510, so 乙>甲>丙.
    2. CPU time = instruction count * CPI * cycle time =(2*80+3*40+5*50)*500=530*500=265000ps

3. Given a program with a dynamic instruction count of 10^5 instructions divided into classes as follows: A: 20%, B: 30%, C: 20%, D: 30%

Clock rate	CPI			
	A	B	C	D
1. 4GHz	1	2	2	2
2. 3.5GHz	2	1	2	1

Which implementation is faster? And by how much?

Ans:

Average CPI:

1. **1 x 0.2 + 2 x 0.3 + 2 x 0.2 + 2 x 0.2 = 1.8**

2. **2 x 0.2 + 1 x 0.3 + 2 x 0.2 + 1 x 0.2 = 1.4**

CPU time:

1. $1.8 \times 10^5 \times \frac{1}{4.5 \times 10^9} = 4.5 \times 10^{-5}\text{s}$

2. $1.4 \times 10^5 \times \frac{1}{3.5 \times 10^9} = 4 \times 10^{-5}\text{s}$

$\frac{4.5 \times 10^{-5}}{4 \times 10^{-5}} = 1.125$

2 is faster by 1.125.

(對抗賽)

4. 有一個 Computer A: clock rate 為 *f* GHz, CPU time 為 *t* seconds，要設計一種 Computer B: CPU time 為 *x* seconds，且 clock cycle 是 A 的 n 倍；再設計 Computer C：他的 clock Rate 是 B 的 m 倍，clock cycle 跟 A 一樣，試求 C 的 CPU time?

Ans:

x/(mn) s

(對抗賽)

5. 什麼是摩爾定律，它是否會永遠生效

Ans:
摩爾定律：當價格不變時，積體電路上可容納的元器件的數目，約每隔 18-24 個月便會增加一倍，性能也將提升一倍。換言之，每一美元所能買到的電腦性能，將每隔 18-24 個月翻一倍以上。
該定律不會永遠生效，隨著新工藝節點的不斷推出，電晶體中原子的數量已經越來越少，種種物理極限制約著其進一步發展。比如當閘極長度足夠短的時候，量子隧穿效應就會發生，會導致漏電流增加。隨著新工藝節點的不斷推出，電晶體中原子的數量已經越來越少，種種物理極限制約著其進

一步發展。比如當閘極長度足夠短的時候，量子隧穿效應就會發生，會導致漏電流增加。所以該定律不會永遠生效

[group1] **(對抗賽)**

6. Three processors P1, P2, P3, with same instruction set.

P1: clock rate=3.0GHz, CPI=1.5

P2: clock rate=2.5GHz, CPI=1.0

P3: clock rate=4.0GHz, CPI=2.2

1) Which processor has the best performance?

2) We want to reduce the execution time of P1 by 30%. But it will cause the CPI to increase by 20%. What clock rate should we have to achieve this time reduction?

Ans:

1.

$$P1 = \frac{1.5 * I}{3 * 10^9} = 0.5 * 10^{-9} * I \ sec$$

$$P2 = \frac{1 * I}{2.5 * 10^9} = 0.4 * 10^{-9} * I \ sec$$

$$P3 = \frac{2.2 * I}{4 * 10^9} = 0.55 * 10^{-9} * I \ sec$$

Hance, P2 has the best performance.

2. $(0.5 * 0.7) * 10^{-9} * I \ sec = \frac{1.5*(1.2*I)}{x}$

$x \approx 5.14 GHz$

Increase clk rate to 5.1GHz

[group4]

7.　甚麼是 benchmarks？

Ans:

很多公司共同成一間公司 **SPEC**，**SPEC** 寫了公平、公正用來評分各公司產出系統的程式，而這類程式就稱為 **benchmarks**

[group11] **(對抗賽)**

8.　How to increase the performance of a processor?

Performance of a processor can be measured in terms of latency(time taken to execute a program) or throughput(number of programs executed per second). Latency of executing a program is given as

Latency = IC x CPI x CC

IC: Instruction Count. Number of instructions in the program

CPI: Clock per Instruction

CC: Clock Cycle time

Ans:

Get better performance if latency is smaller. To increase performance, you can do the following

Decrease number of instructions. This can be done by either choosing better instruction set or writing a smarter compiler generating compact code

Decrease clocks per instruction by pipelining

Decrease clock cycle width by having faster transistors

Ger better throughput by executing multiple programs on multiple threads(logical cores) or on multiple processors(physical cores)