



نظري

2

14

1680 sp

كلية الهندسة المعلوماتية

السنة الثالثة

Performance

(مراجعة و تنمية)

د. خولة العلي

محتوى مجاني غير مخصص للبيع التجاري

RB Informatcs

07/10/2024

بنیان الحواسيب 2



مفهوم الأداء (performance) يجب عن أسئلة كثيرة معقدة، مثلاً:

- كم هي سرعة المعالج؟
- ما سرعة تنفيذ التطبيق؟
- ما سرعة استجابته؟
- كم سرعة تنفيذ كمية كبيرة من الأعمال؟
- كمية الطاقة المستخدمة في الآلة؟

Measures of performance:

لنتذكر وحدات القياس المستخدمة في مفاهيم قياس الأداء:

Clock speed (هرتز) أو دورة في الثانية (Hz)

- 1 KHz = 10^3 Hz : cycle in 1 millisecond (10^{-3} second)
- 1 MHz = 10^6 Hz : cycle in 1 microsecond (10^{-6} second)
- 1 GHz = 10^9 Hz : cycle in 1 nanosecond (10^{-9} second)
- 1 THz = 10^{12} Hz : cycle in 1 picosecond (10^{-12} second)

يعني أنه: إذا تم قياس سرعة نبضة الساعة بالكيلو هرتز (Hz) فيقابلة دورة بالميلي ثانية وإذا تم قياسها بالميغا هرتز فيقابلة دورة بالميكرو ثانية وهكذا...



Instruction/Application Performance:

MIPS: Millions of Instructions Per Second.

FLOPS: Floating Point Instruction Per Second.

يقيس كم تعليمة فاصلة عائمة في الثانية، و إذا كُتب MFLOPS فيكون كم مليون تعليمة فاصلة عائمة في الثانية.

مثال GPUs: GeForce GTX Titan (2,688 cores, 4.5 Tera flops, 7.1 billion transistors, 42 Gigapixel/sec fill rate, 288 GB/sec).

و هذا النوع بطاقة رسومات (كرت شاشة) كما نرى يمكنه تنفيذ (288GB/sec و 4.5 تريليون عملية فاصلة عائمة بالثانية).

Benchmarks (SPEC) معايير الأداء

SPEC: Standard Performance Evaluation Corporation (مؤسسة تقييم الأداء القياسي)

وهي منظمة عالمية للقياس تضع المعايير الأساسية لأداء أنظمة الحاسوب.

CPI: Cycles Per Instruction

IPC: Instructions Per Cycle = $1/CPI$

تستخدم بشكل متكرر أكثر من CPI.

تُفضل أكثر لأن (الأكبر هو الأفضل) أي أنه كل ما زاد CPI يكون الأداء أقل لأن عدد دورات التعليمة يكون أكثر، أما إذا زاد ال IPC يصبح الأداء أفضل لأنه كلما زاد وسطي عدد التعليمات في كل دورة كانت سرعة الأداء أكبر.

التعليمات المختلفة تمتلك عدد دورات مختلف:

مثلاً: تعليمة (add) تأخذ دورة واحدة.

تعليمة (divide) تأخذ أكثر من عشر دورات.

و ذلك يعتمد نسبياً على ترددات التعليمة.



أنت مسؤول عن السعي،

لا النتيجة

برنامج فيه تعليمات (أعداد صحيحة، تعليمات ذاكرة (تخزين و تحميل)، تعليمات فاصلة عائمة)
 بنسب متساوية (عدد تعليمات الأعداد الصحيحة = عدد تعليمات الفاصلة = عدد تعليمات الذاكرة)
 و عدد دورات كل تعليمة منهم كالآتي:

integer : 1 -

memory : 2 -

Floating point : 3 -

فكم يكون الـ CPI في هذه الحالة؟

نسب متساوية أي نسبة كل نوع $\frac{100}{\text{عدد الأنواع}} = 33\%$ يعني $\frac{100}{3} = 33\%$

$$\Rightarrow CPI = (33\% \times 1) + (33\% \times 2) + (33\% \times 3) \approx 2$$

حساب الـ CPI يهمل عدة تأثيرات مثل: الزمن اللازم للوصول للذاكر، الزمن اللازم للإدخال و الإخراج و غيرها...

أي معالج يمكن شراؤه من بين المعالجات الآتية؟

معالج A فيه: clock=5GHz, CPI=2

معالج B فيه: clock=3GHz, CPI=1

غالباً المعالج A فهو يملك تردد الساعة أكبر من B، لكن B أسرع

وذلك بسبب CPI الأقل (بافتراض أن ISA و Compiler نفسهما في المعالجات).

مثلاً:

- 800MHz في Pentium3 أسرع من 1GHz في Pentium4، أي أن الأداء لا يعتمد فقط على تردد الساعة بل يتأثر بعوامل أخرى أيضاً كبنية المعالج و CPI.
- Corei7 أسرع من core2 أي أنه يتأثر أيضاً بعدد الأنوية أي من بنية المعالج و ذلك بافتراض ISA و Compiler متشابهان في كلا الجهازين اللذين نقارن بينهما.



- Latency: (التأخير-الزمن الضائع) How long to finish the program.
- Throughput: (الإنتاجية) How much work finished per unit time.

الحالة المثالية: تكون فيها الإنتاجية عالية، الزمن الضائع قليل، استهلاك طاقة اقل، ذات تكلفة أقل...

مثال:

Car: speed=60 miles/hour, capacity=5

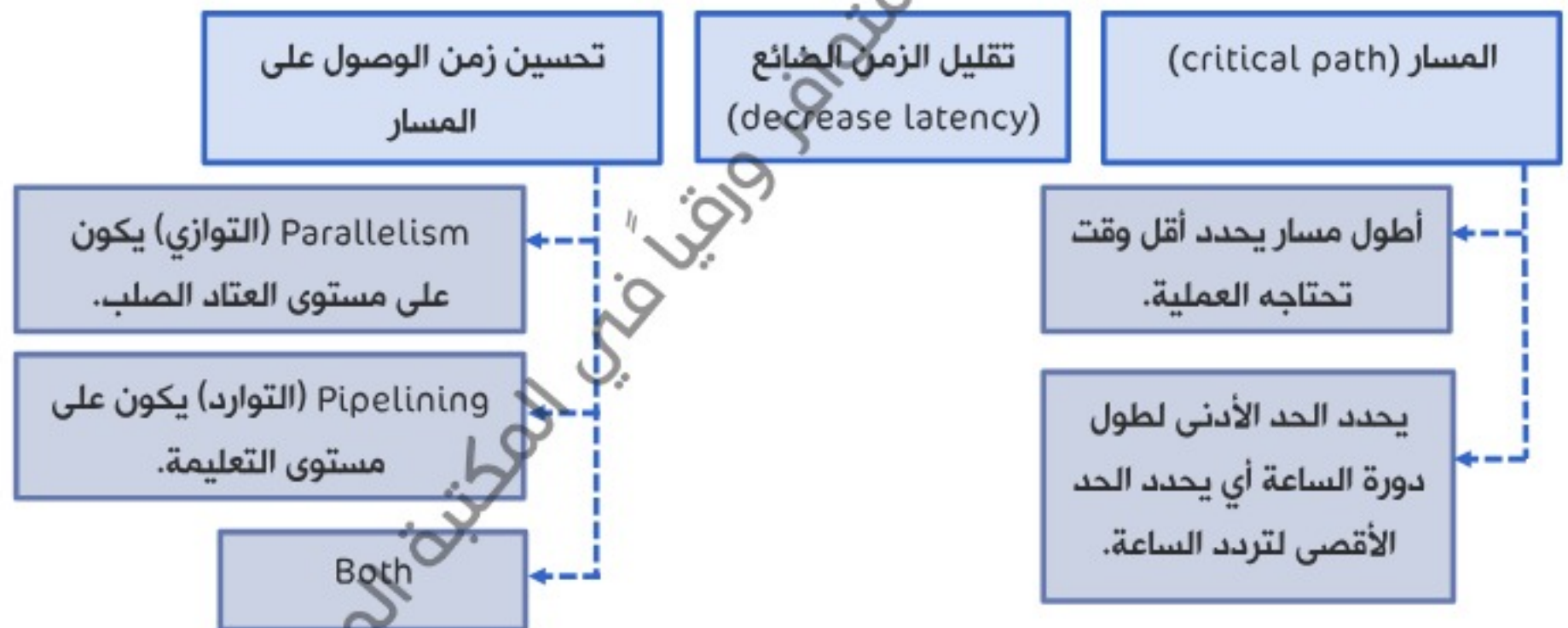
Bus: speed=20 miles/hour, capacity=60

Task: transport passengers 10 miles.

	Latency(min)	Throughput(pph)
Car	10 min	15 pph
Bus	30 min	60 pph

الأداء متعلق بالمهمة الموكلة.

كيف نجعل الحاسوب أسرع؟



Total Time:

$$CPU\ Time = Instructions \times CPI \times Clock\ cycle\ time$$

نستخدم هذا القانون إذا كنا قد حسبنا CPI وسطي لكل التعليمات، أما إذا لم نحسب الوسطي فيكون:

$$CPU\ time = \left(\sum_{i=1}^n I_i \times CPI \right) \times clock\ cycle\ time$$

عدد التعليمات في البرنامج: هو عدد التعليمات التي ينفذها البرنامج وقت التشغيل ويتم تحديده بواسطة البرنامج أو المترجم أو ISA.

CPI عدد دورات كل تعليمة: يتم تحديده بواسطة البرنامج أو المترجم أو ISA أو البنية الدقيقة (المكروية).
Seconds per cycle: زمن كل دورة، يُحدد بواسطة البنية الصغيرة و المعاملات التقنية.

كيف نرفع الأداء؟

نحتاج لتقليل زمن المعالج (CPU time)

- تقليل عدد التعليمات.
- خفض CPI (بإستبدال التعليمات التي تحتاج لدورات أكثر بتعليمات لها عدد دورات أقل).
- خفض وقت دورة الساعة.

Example:

لدينا برنامج فيه التعليمات التالية:

25% load/store, $CPI=3$

60% arithmetic, $CPI=2$

15% branches, $CPI=1$

نريد جعل Multi-cycle 30 MHz CPU (15 MIPS) أسرع بمرتين بجعل التعليمات الحسابية أسرع (لأنها ذات التأثير الأكبر تشكل 60%)

$$CPI\ الحالي = 0.25 \times 3 + 0.6 \times 2 + 0.15 \times 1 = 2.1$$

جعل المعالج أسرع بمرتين أي جعل MIPS=30 بدلاً من 15.

لنجعل CPI للعمليات الحسابية = 1 فيكون:

$$CPI = 0.25 \times 3 + 0.6 \times 1 + 0.15 \times 1 = 1.5$$

نلاحظ أن CPI الوسطية انخفضت و لكن ليس للنصف، لذلك لنعرف كم يجب أن تكون CPI العمليات الحسابية:

$$CPI = (0.25 \times 3) + (0.6 \times x) + (0.15 \times 1) = 1.05$$

$$\Rightarrow x = 0.25$$

وهي نصف الـ 2.1

إذاً لجعل الأداء يصبح أسرع مرتين يجب خفض Arithmetic CPI من 2 إلى 0.25.

ملخص لبعض الأفكار:

تستخدم معظم الحواسيب نبضة ساعة ذات تردد ثابت لوحدة المعالجة المركزية و تعتمد نبضة الساعة على تصميم وحدة المعالجة المركزية CPU و على التقنية المتبعة في تصنيع الدارات المتكاملة IC حيث:

$$\text{clock rate} = \frac{1}{\text{clock cycle time}}$$

$$CPU_{time} = CPU_{\text{clock cycles for a program}} \times \text{clock cycle time}$$

بتعويض العلاقة الأولى:

$$CPU_{time} = \frac{CPU_{\text{clock cycles for a program}}}{\text{clock rate}}$$

■ متوسط عدد الدورات في التعليمة CPI في البرنامج يساوي إلى متوسط عدد الدورات CPI لجميع التعليمات المنفذة في البرنامج على عدد التعليمات.

$$CPI = \frac{CPU_{\text{clock cycles for a program}}}{I} \rightarrow \text{عدد التعليمات}$$

■ نقول عن حاسوب أداؤه أفضل من آخر إذا كان زمن التنفيذ الكلي أقصر (يتناسب الأداء عكساً مع زمن التنفيذ)

$$\text{performance } A = \frac{1}{\text{execution time } A}$$

■ معادلة الأداء لوحدة المعالجة المركزية:

$$T = I \times CPI \times C$$

T: زمن تنفيذ البرنامج

I: عدد تعليمات البرنامج

CPI: عدد الدورات الوسطي لتنفيذ كل تعليمة

C: زمن الدورة

أي تحسين بتلك المحددات يحسن من أداء وحدة المعالجة المركزية.



■ من معايير أداء الحاسوب هي MIPS:

و يعطى بالعلاقة:

$$MIPS\ rating = \frac{\text{عدد التعليمات}}{\text{زمن التنفيذ} \times 10^6}$$

Amdahl's Law

$$\text{زمن التنفيذ بعد التحسين} = \frac{\text{زمن التنفيذ المتأثر بالتحسين}}{\text{كمية التحسين}} + \text{زمن التنفيذ غير المتأثر بالتحسين.}$$

عندما نجري تحسين في وحدة المعالجة، ليس مشروطاً أن ينعكس على كامل البرنامج.

Speed up (مقدار التحسين):

يمكن حسابه بين حاسوبين مختلفين أو لحاسوب واحد قبل التحسين و بعده، وقانونه:

حاسوب واحد:

$$speed\ up(n) = \frac{\text{زمن التنفيذ بعد التغيير}}{\text{زمن التنفيذ قبل التغيير}}$$

$n > 1$
التغيير أسوأ للأداء

$n < 1$
التغيير حسن من الأداء

أداء حاسوبين:

$$n = \frac{\text{execution time A}}{\text{execution time B}}$$

$n < 1$ أفضل A
 $n > 1$ أفضل B

$$n = \frac{\text{performance A}}{\text{performance B}}$$

$n < 1$ أفضل B
 $n > 1$ أفضل A



أمثلة عن معايير الأداء:

إذا كان لدينا معالج A و معالج B ينفذان برنامج بنفس عدد الدورات، فما هو المعيار مما يلي الذي يحدد أيهما أسرع؟

A- MIPS	D- Clock rate
B- CPI	E- Nothing
C- Dynamic instruction count	

معالج A و معالج B لهما نفس تردد الساعة (clock rate)، و لكن يدعمان ISA (Instruction Set Architecture) مختلفة، فالذي يحدد أيهما أسرع هو:

A- MIPS	D- Clock rate
B- CPI	E- Nothing
C- Dynamic instruction count	

معالجان A و B يدعمان نفس ال ISA، فالأسرع بينهما يحدد حسب:

A- MIPS	D- Clock rate
B- CPI	E- Nothing
C- Dynamic instruction count	

Fallacy: Low Power at Idle

باتخاذ الطاقة معيار، في معالج core i7، يكون:

At 100% load: 258W (100%)

At 50% load: 170W (%66)

At 10% load: 121W (%47)

$\frac{10}{100}$

نلاحظ انخفاض استهلاك الطاقة كلما كاد التحميل ينتهي، مثلاً لتحميل

استهلك 47% من الطاقة، وعندما وصل للنصف كان قد استهلك 66%

وبمقارنتهما نلاحظ انخفاض استهلاك الطاقة.



MIPS (Million of Instructions Per Second)

لا يأخذ MIPS بالاعتبار:

- الاختلاف بين ISA لأجهزة الحاسوب.
- الاختلاف في التعقيد بين التعليمات.

$$MIPS = \frac{\text{instruction count}}{\text{execution time} \times 10^6}$$

$$= \frac{\frac{\text{instruction count}}{\text{instruction count} \times CPI}}{\text{clock rate}} \times 10^6 = \frac{\text{clock rate}}{CPI \times 10^6}$$

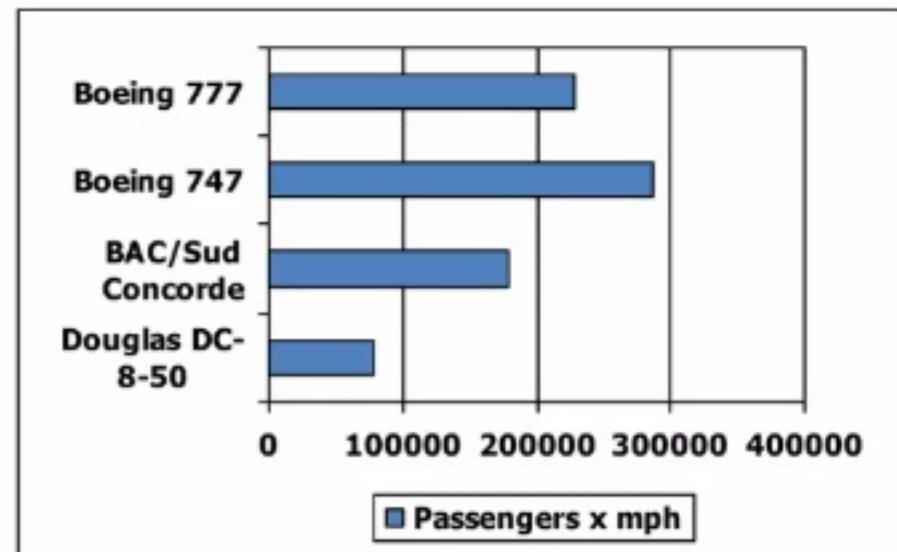
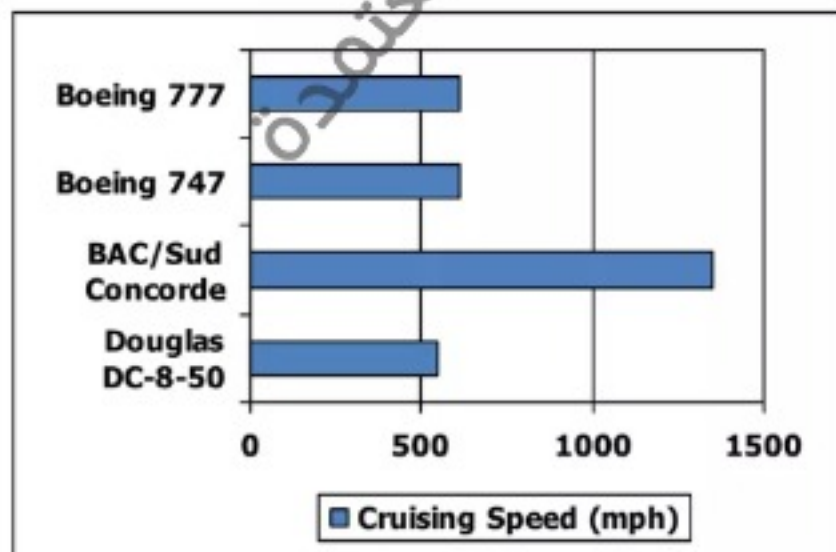
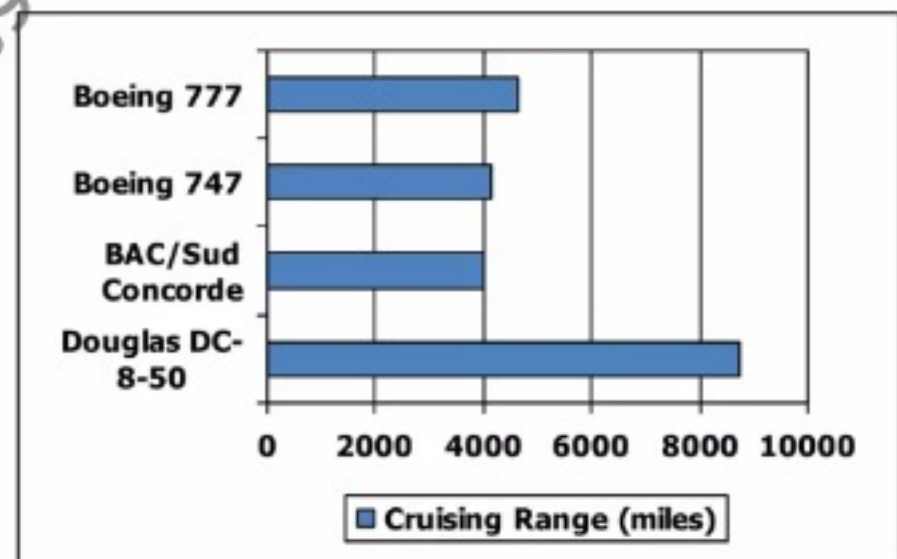
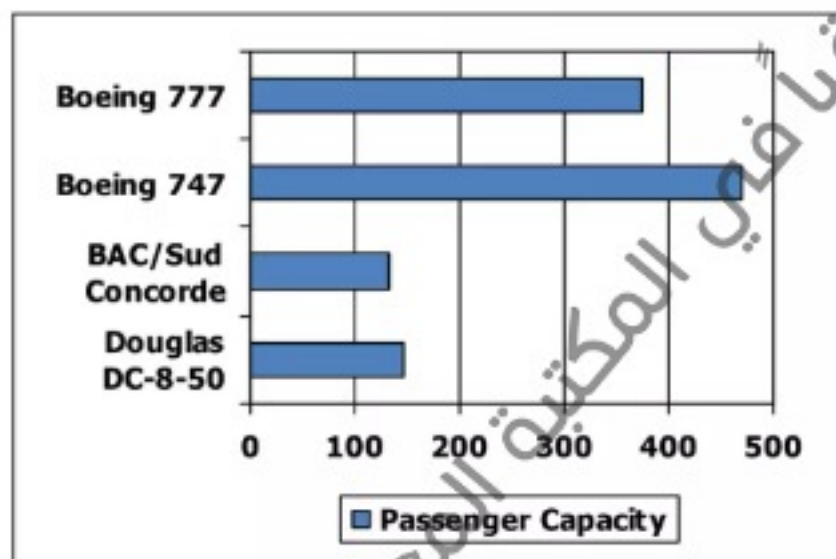
حيث:

$$\text{execution time} = \text{instruction count} \times CPI \times \text{clock cycle time}$$

$$= \frac{\text{instruction count} \times CPI}{\text{clock rate}}$$

Defining performance

Which airplane has the best performance?



حاسوب A يأخذ 10 ثواني لتنفيذ برنامج، و حاسوب B يأخذ 15 ثانية لتنفيذ نفس البرنامج، نلاحظ أن A أسرع من B لأن وقت تنفيذه أقل، و إذا أردنا استخدام قانون:

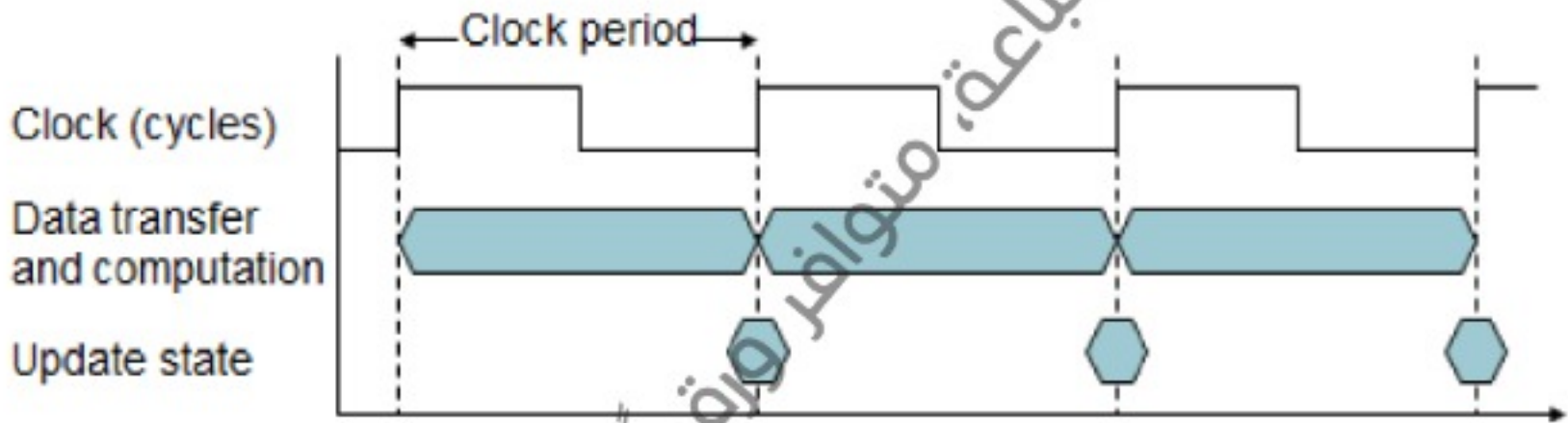


$$\frac{\text{execution time A}}{\text{execution time B}} = \frac{10}{15} = \frac{2}{3} < 1$$

← A أسرع من B

CPU Clocking

إن تشغيل الأجهزة الرقمية تتحكم به ساعة ذات معدل ثابت.



المسافة من جهة صاعدة إلى نهاية جهة هابطة أو من بداية جهة هابطة إلى نهاية جهة صاعدة تسمى نبضة الساعة.

- يقاس بالثانية و أجزائها و مضاعفاتها (زمن): Clock period
- يقاس بالهرتز و أجزاؤه و مضاعفاته (تردد): Clock rate (frequency)

يتم تحسين الأداء من خلال:

يجب على مصمم العتاد الصلب للجهاز معادلة معدل الساعة مقابل عدد الدورات.

رفع معدل الساعة (تردد)

تقليل عدد دورات الساعة

Instruction Count and CPI

نتذكر أن:

$$CPU \text{ time} = \text{clock cycles in CPU} \times \text{clock cycle time}$$

$$\text{clock cycle time} = \frac{1}{\text{clock rate}}$$

و لدينا:

$$\text{clock cycles} = \text{Instruction count} \times \text{cycles per instruction}$$

$$\Rightarrow CPU \text{ time} = \frac{\text{Instruction count} \times CPI}{\text{clock rate}}$$

- عدد التعليمات في البرنامج تحدد من خلال الـ ISA و المترجم (compiler)
- الـ CPI الوسطي: يحدد عتاد الـ CPU فعندما يوجد تعليمات مختلفة لها CPI مختلف فيتأثر الوسطي بمزيج هذه التعليمات.

Example:

حاسوب A فيه: cycle time=250 ps و CPI=2.0

حاسوب B فيه: cycle time=500 ps و CPI=1.2

و لهما نفس الـ ISA أيهما أسرع، و بكم؟

نحسب CPU time لكل منهما:

$$CPU \text{ time (A)} = \text{Instruction Count} \times CPI_A \times \text{cycle time}_A$$

$$= I \times 2.0 \times 250 \text{ ps} = I \times 500 \text{ ps}$$

$$CPU \text{ time (B)} = \text{Instruction Count} \times CPI_B \times \text{cycle time}_B$$

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

بالمقارنة نجد أن A أسرع من B.

$$\frac{CPU \text{ time}(B)}{CPU \text{ time}(A)} = \frac{I \times 600 \text{ ps}}{I \times 500 \text{ ps}} = 1.2$$

⇐ مقدار الفرق (النسبة) هو 1.2

هنا لم نعتمد على CPI فقط لأنه يوجد cycle time يؤثر بشكل كبير على الأداء.



CPI in more Detail

إذا كان هناك تعليمات مختلفة تأخذ عدد مختلف من الدورات:

$$\text{clock cycles} = \sum_{i=1}^n (CPI_i \times \text{Instruction count}_i)$$

لحساب الـ CPI الوسطي:

$$CPI = \frac{\text{clock cycles}}{\text{Instruction count}} = \sum_{i=1}^n \left(CPI_i \times \underbrace{\frac{\text{Instruction count}_i}{\text{Instruction count}}}_{\text{التردد النسبي}} \right)$$

Example:

ما هو الـ CPI الوسطي و clock cycles لكل من التسلسلين:

IC: Instruction Count

Class	A	B	C
CPI for Class	1	2	3
IC in sequence 1	2	1	2
IC in sequence 2	4	1	1

التسلسل الأول:

عدد التعليمات = 5 تعليمات

$$\text{Clock cycles} = 2 \times 1 + 1 \times 2 + 2 \times 3 = 10$$

$$\text{Avg.CPI} = \frac{10}{5} = 2$$

التسلسل الثاني:

عدد التعليمات = 6 تعليمات

$$\text{Clock cycles} = 4 \times 1 + 1 \times 2 + 1 \times 3 = 9$$

$$\text{Avg.CPI} = \frac{9}{6} = 1.5$$

ملخص:

سنكتب قانون CPU time كواحدات:

$$\text{CPU time} = \frac{\text{instruction}}{\text{program}} \times \frac{\text{clock cycles}}{\text{instruction}} \times \frac{\text{seconds}}{\text{clock cycle}}$$

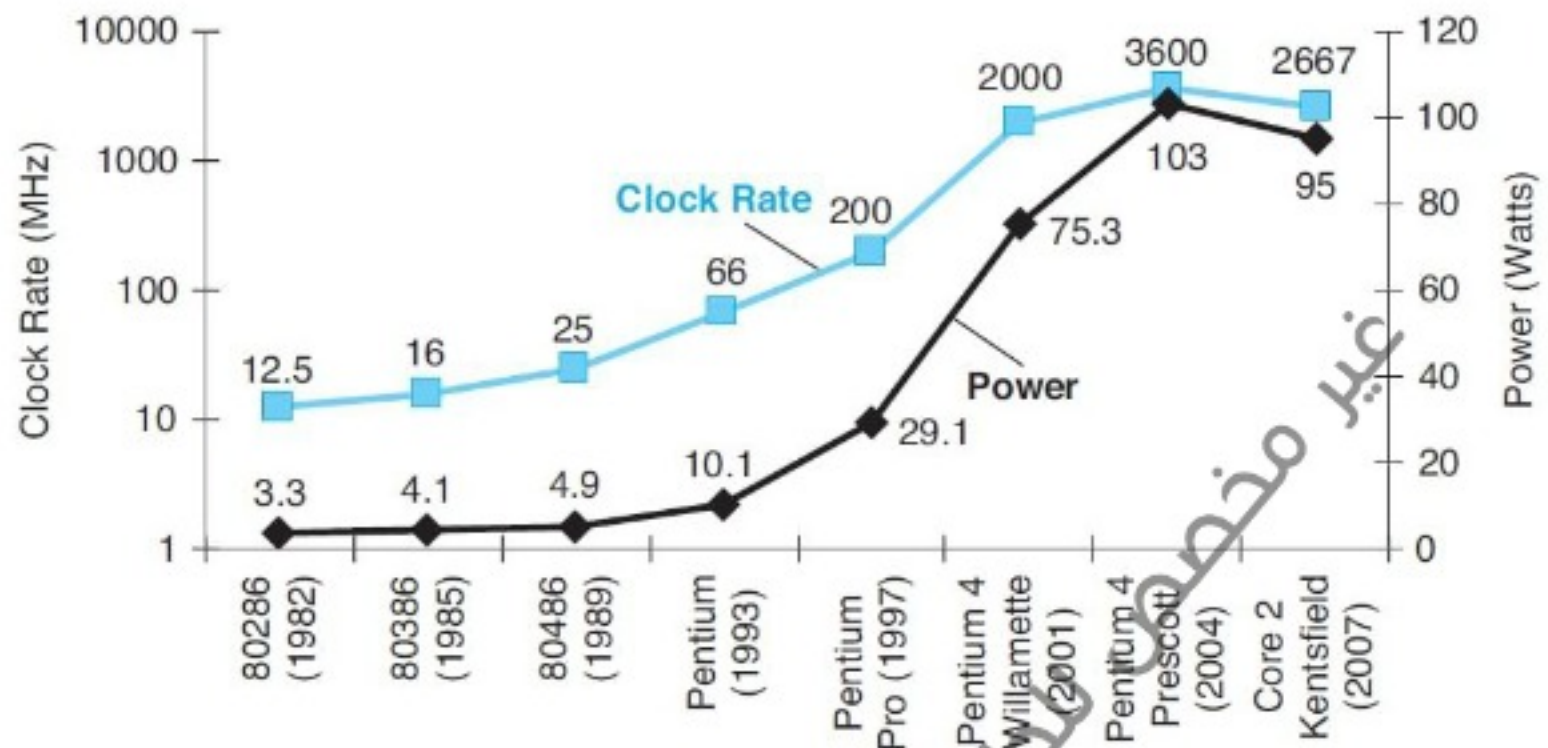
الأداء يعتمد على:

Compiler: يؤثر على عدد التعليمات و CPI.

الخوارزمية: تؤثر على عدد التعليمات و CPI.

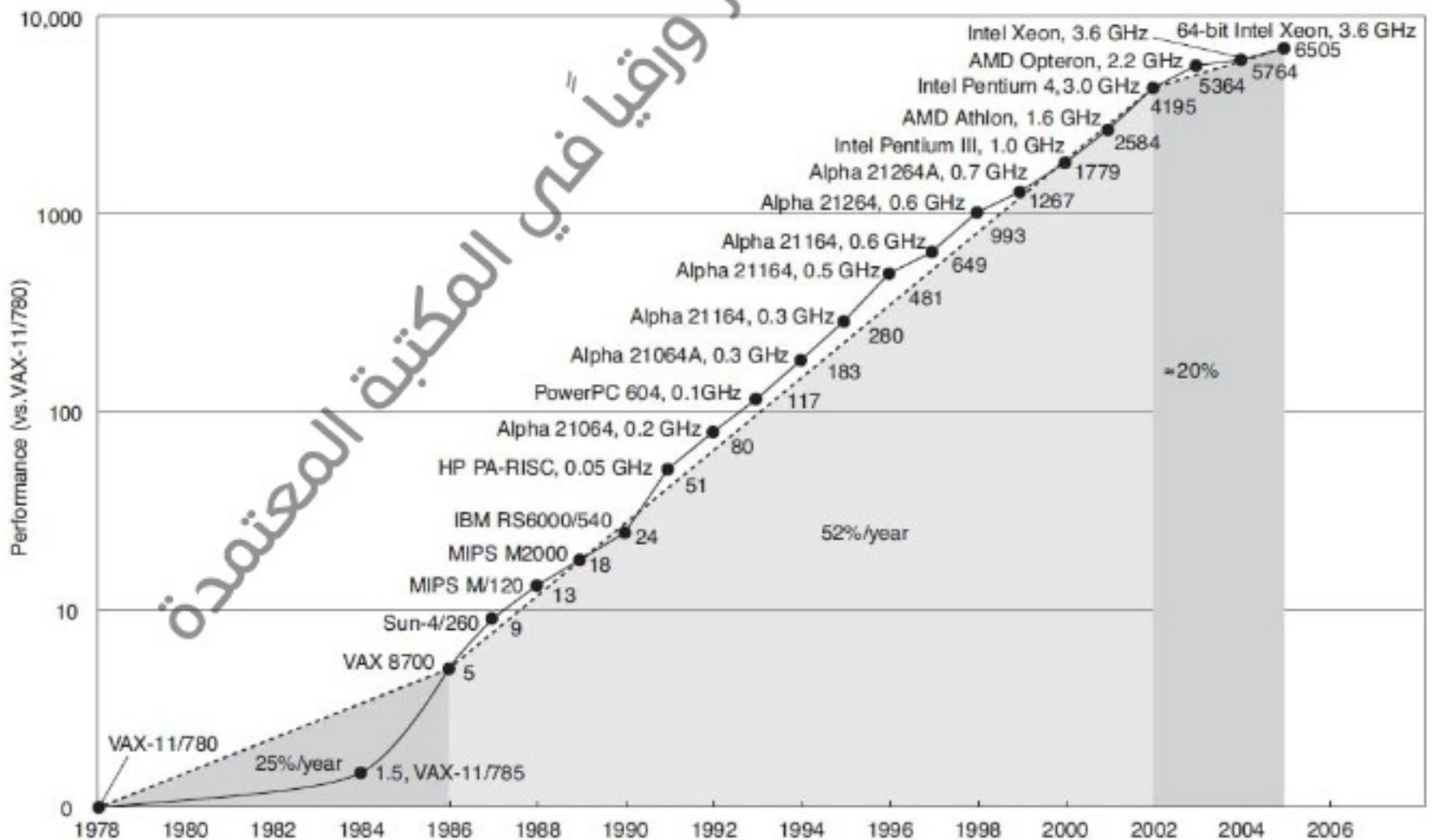
ISA: يؤثر على عدد التعليمات و CPI و زمن التنفيذ.

لغة البرمجة: تؤثر على عدد التعليمات و CPI.



$$Power = 0.5 \times capacitive\ load \times voltage^2 \times frequency$$

Uni processor Performance:



Multi Processors (المعالجات الصغيرة)

- المعالجات الدقيقة متعددة النوى: تحتوي على أكثر من معالج في الشريحة.
- تتطلب برمجة متوازية واضحة (parallel programming) تقارن مع تعليمات متوازية المستوى.
- -العتاد ينفذ تعليمات عدة مرة واحدة و غير مرئي للمبرمج.
- صعوبة التنفيذ (العمل)
 - برمجة الأداء.
 - موازنة التحميل
 - تطوير التواصل والمزامنة

SPEC CPU Benchmark (Standard Performance Evaluation corporation)

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

ملاحظات ختامية:

- يتم تحسين الأداء والكلفة تبعاً لتقنية التطوير الأساسية.
- الطبقات المتسلسلة للتجريد في العتاد الصلب وفي البرمجيات (مثل الذاكر الموجودة بأماكن مختلفة من الحاسوب، فكل نوع من الذاكر حسب استخدامه يكون موجود في الجهاز و لكل ذاكرة حجم و تقنية مختلف، و لتذكر أيضاً أنه يوجد هرمية للذاكر بحسب الحجم و الكلفة و الأداء.....) Hierarchical layers of abstraction
- ISA (بنية مجموعة التعليمات) مما يؤثر على الواجهة البرمجية والصلبة.
- وقت التنفيذ هو أفضل مقياس للأداء.
- الطاقة هو عامل محدود يستخدم التوازي للتحسين من الأداء.

The End

