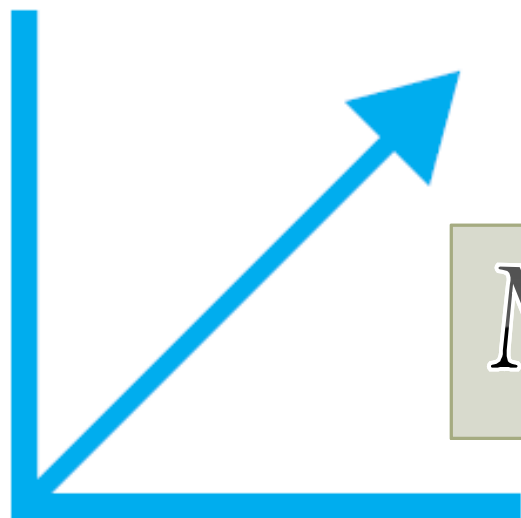


اصول سیستم‌های کامپیوتری

فصل ۱: فناوری و تجربدهای کامپیوتری

هشت ایده اساسی در معماری کامپیوتر

15



MOORE'S LAW

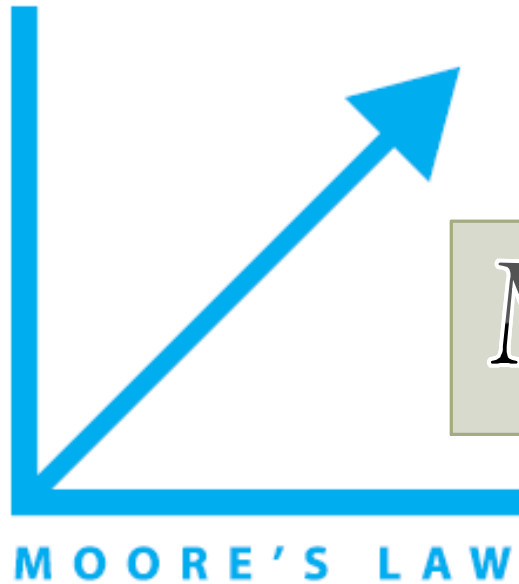
قانون Moore

تقریباً هر دو سال، تعداد ترانزیستورهای قابل مجتمع شدن در تراشه‌ها دو برابر می‌شود.

-- Gordon Moore, 1965

هشت ایده اساسی در معماری کامپیوتر

15

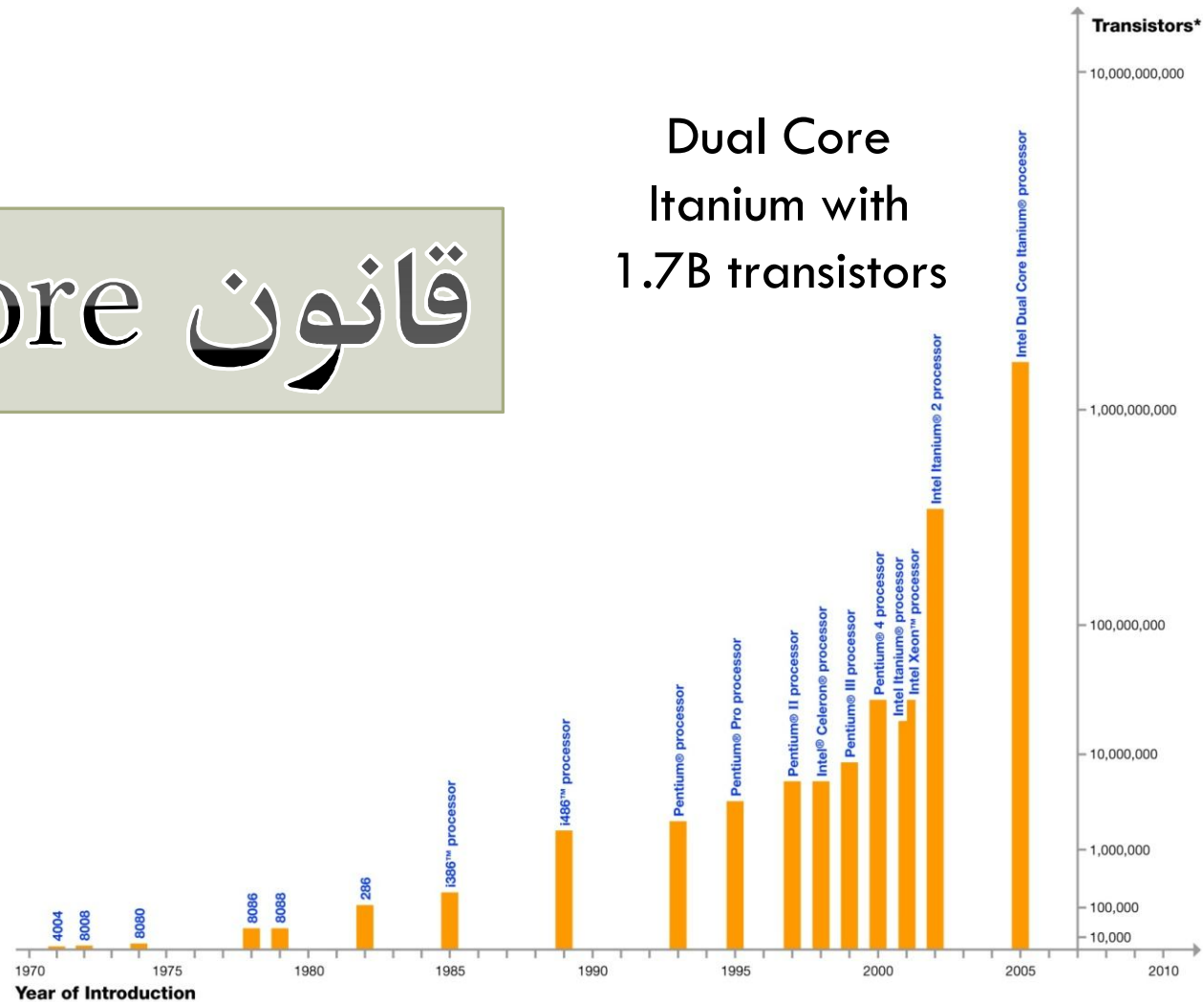


قانون Moore

تقریباً هر دو سال، تعداد ترانزیستورهای قابل مجتمع شدن در تراشه‌ها دو برابر می‌شود.

-- Gordon Moore, 1965

Dual Core
Itanium with
1.7B transistors



*Note: Vertical scale of chart not proportional to actual Transistor count.

هشت ایده اساسی در معماری کامپیوتر

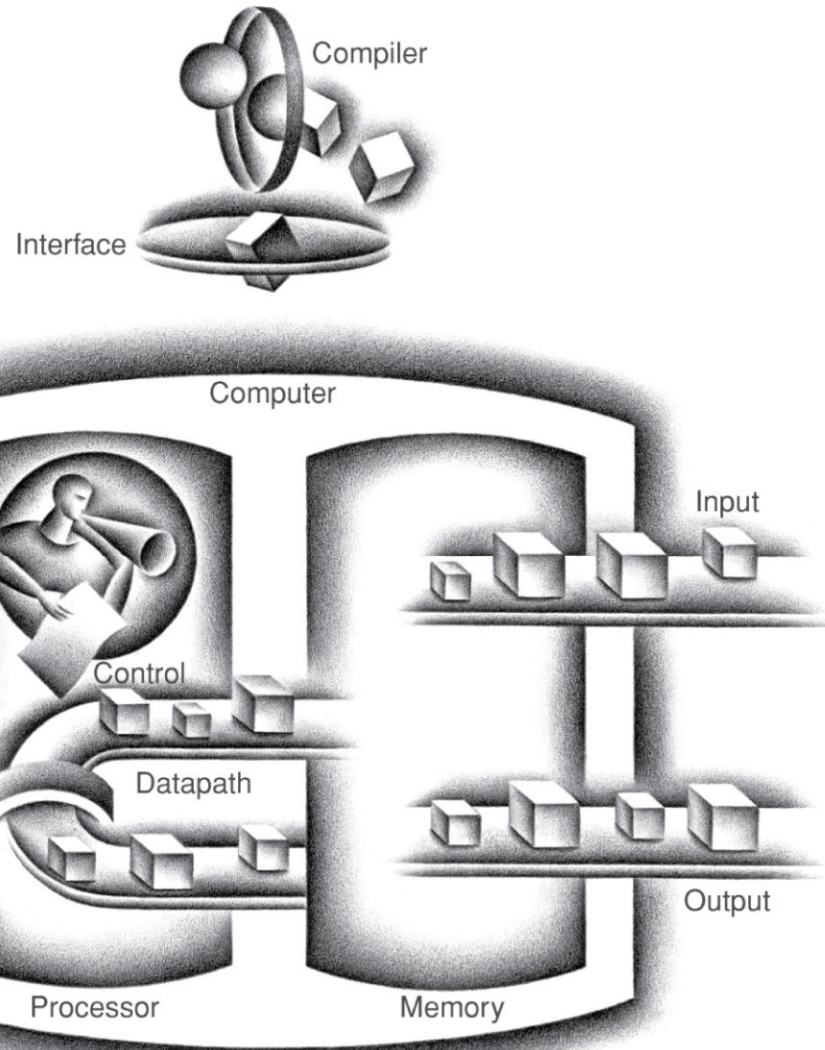
21



ABSTRACTION

ساده تر شدن
طراحی با
تجربید

اجزای یک کامپیوتر



□ پنج جزء متداول

1. ورودی

2. خروجی

3. حافظه

4. مسیر داده‌ها

5. واحد کنترل

□ پردازنده = مسیر
داده + واحد کنترل

هشت ایده اساسی در معماری کامپیوتر

32

بهبود
قسمت‌های
مشترک



COMMON CASE FAST

هشت ایده اساسی در معماری کامپیوتر

33



PARALLELISM

افزایش
کارایی
با
توازی

هشت ایده اساسی در معماری کامپیوتر

34

افزایش کارایی
با
لوله کشی
(Pipelining)



PIPELINING

هشت ایده اساسی در معماری کامپیوتر

35



PREDICTION

افزایش
کارایی
با
پیش بینی

هشت ایده اساسی در معماری کامپیوتر

36

سلسله مراتب حافظه



HIERARCHY

هشت ایده اساسی در معماری کامپیوتر

37

قابلیت اعتماد با
افزونگی
(Redundancy)



DEPENDABILITY

هشت ایده اساسی در معماری کامپیوتر

هشت ایده اساسی در معماری کامپیوتر

38



هشت ایده اساسی در معماری کامپیوتر

38



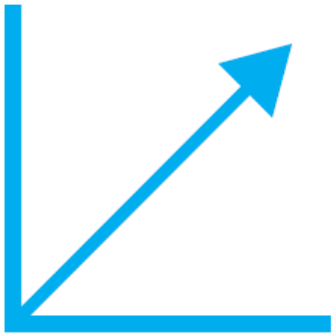
MOORE'S LAW



ABSTRACTION

هشت ایده اساسی در معماری کامپیوتر

38



MOORE'S LAW



ABSTRACTION



COMMON CASE FAST

هشت ایده اساسی در معماری کامپیوتر

38



MOORE'S LAW



ABSTRACTION



COMMON CASE FAST



PARALLELISM

هشت ایده اساسی در معماری کامپیوتر

38



MOORE'S LAW



ABSTRACTION



COMMON CASE FAST



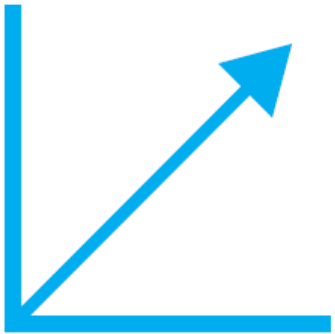
PARALLELISM



PIPELINING

هشت ایده اساسی در معماری کامپیوتر

38



MOORE'S LAW



ABSTRACTION



COMMON CASE FAST



PARALLELISM



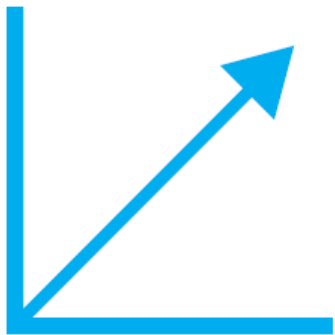
PIPELINING



PREDICTION

هشت ایده اساسی در معماری کامپیوتر

38



MOORE'S LAW



ABSTRACTION



COMMON CASE FAST



PARALLELISM



PIPELINING



PREDICTION



HIERARCHY

هشت ایده اساسی در معماری کامپیوتر

38



MOORE'S LAW



ABSTRACTION



COMMON CASE FAST



PARALLELISM



PIPELINING



PREDICTION



HIERARCHY



DEPENDABILITY

مکانی امن برای داده‌ها

□ حافظه‌های اصلی فرار

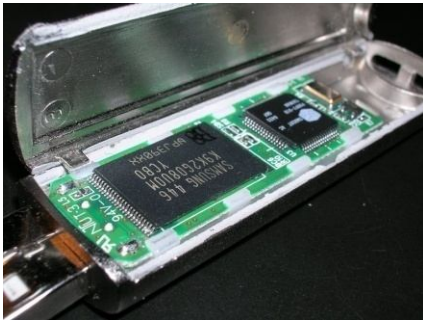
□ از دست دادن محتویات در صورت قطع جریان برق

مکانی امن برای داده‌ها

39



- حافظه‌های اصلی فرار
- از دست دادن محتویات در صورت قطع جریان برق
- حافظه‌های ثانویه غیرفرار



- دیسک مغناطیسی
- حافظه فلش (Flash)
- دیسک نوری



واحدهای اندازه‌گیری ظرفیت حافظه

Decimal term	Abbreviation	Value	Binary term	Abbreviation	Value	% Larger
kilobyte	KB	10^3	kibibyte	KiB	2^{10}	2%
megabyte	MB	10^6	mebibyte	MiB	2^{20}	5%
gigabyte	GB	10^9	gibibyte	GiB	2^{30}	7%
terabyte	TB	10^{12}	tebibyte	TiB	2^{40}	10%
petabyte	PB	10^{15}	pebibyte	PiB	2^{50}	13%
exabyte	EB	10^{18}	exbibyte	EiB	2^{60}	15%
zettabyte	ZB	10^{21}	zebibyte	ZiB	2^{70}	18%
yottabyte	YB	10^{24}	yobibyte	YiB	2^{80}	21%

واحدهای اندازه‌گیری ظرفیت حافظه



Decimal term	Abbreviation	Value	Binary term	Abbreviation	Value	% Larger
kilobyte	KB	10^3	kibibyte	KiB	2^{10}	2%
megabyte	MB	10^6	mebibyte	MiB	2^{20}	5%
gigabyte	GB	10^9	gibibyte	GiB	2^{30}	7%
terabyte	TB	10^{12}	tebibyte	TiB	2^{40}	10%
petabyte	PB	10^{15}	pebibyte	PiB	2^{50}	13%
exabyte	EB	10^{18}	exbibyte	EiB	2^{60}	15%
zettabyte	ZB	10^{21}	zebibyte	ZiB	2^{70}	18%
yottabyte	YB	10^{24}	yobibyte	YiB	2^{80}	21%

شبکه‌های کامپیوتری

41



□ ارتباط

□ تبادل اطلاعات بین کامپیوترها با سرعت زیاد

□ اشتراک منابع

□ مانند ابزارهای I/O

□ فایل‌ها و خدمات

□ دسترسی از راه دور

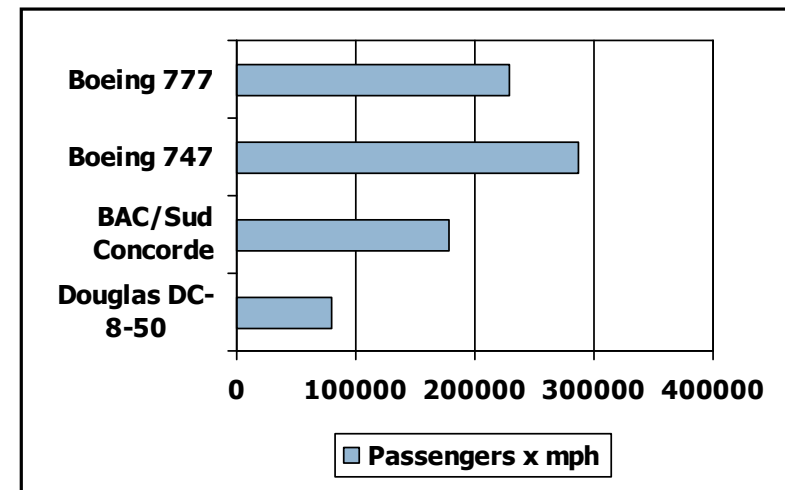
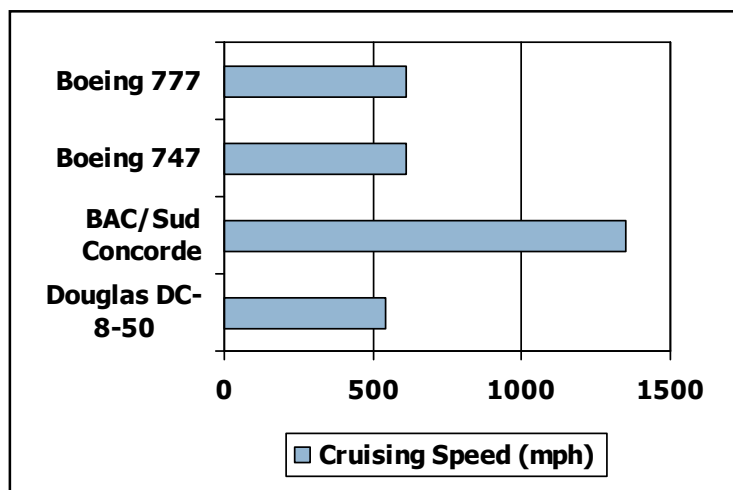
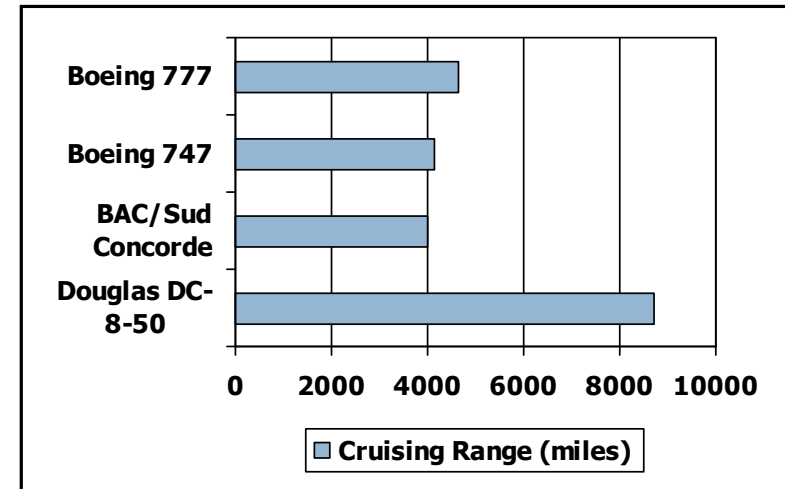
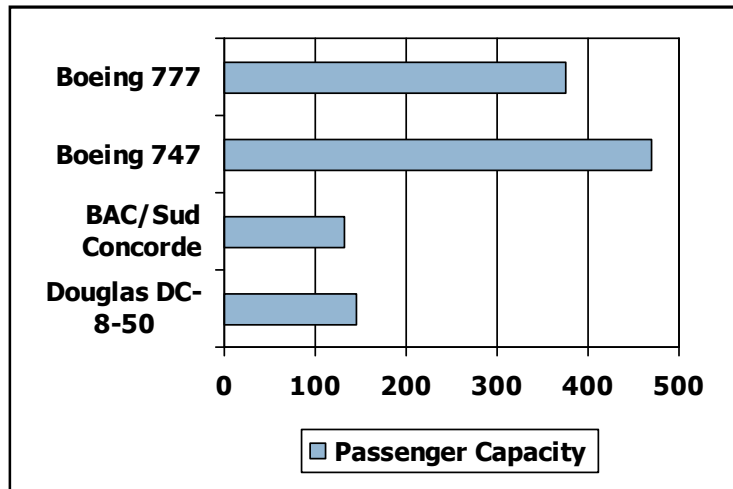
□ انواع

□ LAN

□ WAN

تعریف کارایی: کارایی کدام هواپیما بالاتر است؟

42



زمان پاسخ و توان عملیاتی

زمان پاسخ و توان عملیاتی

- زمان پاسخ (Response Time)
- زمان مورد نیاز برای انجام یک کار

زمان پاسخ و توان عملیاتی

□ زمان پاسخ (Response Time)

□ زمان مورد نیاز برای انجام یک کار

□ توان عملیاتی (Throughput)

□ جمع کل کار انجام شده در واحد زمان

■ مانند تعداد کار، وظیفه، تراکنش و ... در هر ساعت

زمان پاسخ و توان عملیاتی

- زمان پاسخ (Response Time)
 - زمان مورد نیاز برای انجام یک کار
- توان عملیاتی (Throughput)
 - جمع کل کار انجام شده در واحد زمان
 - مانند تعداد کار، وظیفه، تراکنش و ... در هر ساعت
- تاثیر ...
 - جایگزینی پردازنده با یک پردازنده سریعتر
- ... بر زمان پاسخ و توان عملیاتی چگونه است؟

زمان پاسخ و توان عملیاتی

□ زمان پاسخ (Response Time)

□ زمان مورد نیاز برای انجام یک کار

□ توان عملیاتی (Throughput)

□ جمع کل کار انجام شده در واحد زمان

■ مانند تعداد کار، وظیفه، تراکنش و ... در هر ساعت

□ تاثیر ...

□ جایگزینی پردازنده با یک پردازنده سریعتر

□ اضافه کردن پردازنده بیشتر

□ ... بر زمان پاسخ و توان عملیاتی چگونه است؟

بهبود هر دو



زمان پاسخ و توان عملیاتی

□ زمان پاسخ (Response Time)

□ زمان مورد نیاز برای انجام یک کار

□ توان عملیاتی (Throughput)

□ جمع کل کار انجام شده در واحد زمان

■ مانند تعداد کار، وظیفه، تراکنش و ... در هر ساعت

□ تاثیر ...

□ جایگزینی پردازنده با یک پردازنده سریعتر

□ اضافه کردن پردازنده بیشتر

□ ... بر زمان پاسخ و توان عملیاتی چگونه است؟

بهبود هر دو



فقط بهبود توان عملیاتی



زمان پاسخ و توان عملیاتی

□ زمان پاسخ (Response Time)

□ زمان مورد نیاز برای انجام یک کار

□ توان عملیاتی (Throughput)

□ جمع کل کار انجام شده در واحد زمان

■ مانند تعداد کار، وظیفه، تراکنش و ... در هر ساعت

□ تاثیر ...

□ جایگزینی پردازنده با یک پردازنده سریعتر

□ اضافه کردن پردازنده بیشتر

□ ... بر زمان پاسخ و توان عملیاتی چگونه است؟

□ فعلا بر زمان پاسخ تمرکز می کنیم.

بهبود هر دو



فقط بهبود توان عملیاتی



عملکرد نسبی

□ کارایی = ۱ / زمان اجرا

عملکرد نسبی

- کارایی = $1 / \text{زمان اجرا}$
- « X ، n بار از Y سریع تر است»

عملکرد نسبی

□ کارایی = ۱ / زمان اجرا

□ «X، n بار از Y سریع تر است»

$$\begin{aligned} & \text{Performance}_x / \text{Performance}_y \\ & = \text{Execution time}_y / \text{Execution time}_x = n \end{aligned}$$

عملکرد نسبی

□ کارایی = ۱ / زمان اجرا

□ «X، n بار از Y سریع تر است»

$$\text{Performance}_X / \text{Performance}_Y \\ = \text{Execution time}_Y / \text{Execution time}_X = n$$

□ مثال:

□ 10s برای A، 15s برای B

عملکرد نسبی

□ کارایی = ۱ / زمان اجرا

□ «X، n بار از Y سریع تر است»

$$\text{Performance}_X / \text{Performance}_Y \\ = \text{Execution time}_Y / \text{Execution time}_X = n$$

□ مثال:

□ 10s برای A، 15s برای B

□ $\text{Execution Time}_B / \text{Execution Time}_A$
 $= 15s / 10s = 1.5$

عملکرد نسبی

□ کارایی = ۱ / زمان اجرا

□ «X، n بار از Y سریعتر است»

$$\text{Performance}_X / \text{Performance}_Y \\ = \text{Execution time}_Y / \text{Execution time}_X = n$$

□ مثال:

□ 10s برای A، 15s برای B

□ $\text{Execution Time}_B / \text{Execution Time}_A$
 $= 15s / 10s = 1.5$

□ بنابراین، A، ۱.۵ برابر سریعتر از B است!

اندازه‌گیری زمان اجرا

□ فاصله بین زمان شروع و زمان پایان

اندازه‌گیری زمان اجرا

□ فاصله بین زمان شروع و زمان پایان

□ شامل

■ زمان پردازش، I/O ، سربار سیستم‌عامل، زمان تلف شده و ...

□ تعیین کننده کارایی سیستم

اندازه‌گیری زمان اجرا

□ فاصله بین زمان شروع و زمان پایان

□ شامل

■ زمان پردازش، I/O، سربار سیستم‌عامل، زمان تلف شده و ...

□ تعیین کننده کارایی سیستم

□ زمان CPU

□ زمانی که فقط صرف امور پردازشی برنامه شده

■ بدون در نظر گرفتن زمان I/O، زمان مصرف شده برای اجرای سایر برنامه‌ها، ...

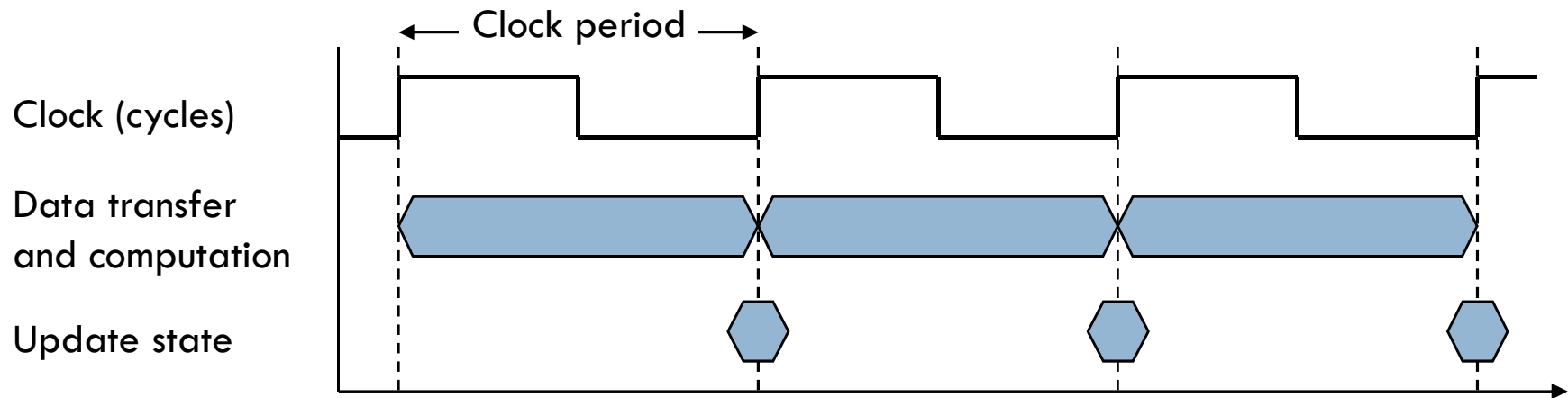
□ شامل زمان CPU کاربر و OS

□ تاثیرپذیری برنامه‌های مختلف نسبت به OS و کارایی سیستم متفاوت است.

چرخه‌های ساعت (CPU Clocking)

46

□ عملکرد سخت‌افزار دیجیتال تابع ساعتی با دوره تناوب ثابت است.



□ دوره تناوب ساعت (Clock Period)

■ طول یک چرخه ساعت، برای نمونه، $250\text{ps} = 0.25\text{ns} = 250 \times 10^{-12}\text{s}$

□ فرکانس ساعت (Clock Frequency (Rate))

■ تعداد چرخه در هر ثانیه، برای نمونه، $4.0\text{GHz} = 4000\text{MHz} = 4.0 \times 10^9\text{Hz}$

یادآوری!

Factor	Prefix	Symbol	Example
$1,000,000,000 = 10^9$	giga	G	1 gigameter (Gm) = 10^9 m
$1,000,000 = 10^6$	mega	M	1 megameter (Mm) = 10^6 m
$1,000 = 10^3$	kilo	k	1 kilogram (kg) = 10^3 g
$100 = 10^2$	hecto	h	1 hectogram (hg) = 100 g
$10 = 10^1$	deka	da	1 dekagram (dag) = 10 g
$0.1 = 10^{-1}$	deci	d	1 decimeter (dm) = 0.1 m
$0.01 = 10^{-2}$	centi	c	1 centimeter (cm) = 0.01 m
$0.001 = 10^{-3}$	milli	m	1 milligram (mg) = 0.001 g
$^*0.000001 = 10^{-6}$	micro	μ	1 micrometer (μm) = 10^{-6} m
$^*0.000000001 = 10^{-9}$	nano	n	1 nanosecond (ns) = 10^{-9} s
$^*0.000000000001 = 10^{-12}$	pico	p	1 picosecond (ps) = 10^{-12} s

زمان CPU

48

$$\begin{aligned} \text{CPU Time} &= \text{CPU Clock Cycles} \times \text{Clock Cycle Time} \\ &= \frac{\text{CPU Clock Cycles}}{\text{Clock Rate}} \end{aligned}$$

□ بهبود کارایی با

زمان CPU

48

$$\begin{aligned} \text{CPU Time} &= \text{CPU Clock Cycles} \times \text{Clock Cycle Time} \\ &= \frac{\text{CPU Clock Cycles}}{\text{Clock Rate}} \end{aligned}$$

- بهبود کارایی با
- کاهش تعداد چرخه‌های لازم

زمان CPU

48

$$\begin{aligned} \text{CPU Time} &= \text{CPU Clock Cycles} \times \text{Clock Cycle Time} \\ &= \frac{\text{CPU Clock Cycles}}{\text{Clock Rate}} \end{aligned}$$

- بهبود کارایی با
- کاهش تعداد چرخه‌های لازم
- افزایش نرخ ساعت

زمان CPU

48

$$\begin{aligned} \text{CPU Time} &= \text{CPU Clock Cycles} \times \text{Clock Cycle Time} \\ &= \frac{\text{CPU Clock Cycles}}{\text{Clock Rate}} \end{aligned}$$

- بهبود کارایی با
 - کاهش تعداد چرخه‌های لازم
 - افزایش نرخ ساعت
- اما، در اغلب اوقات طراحان سخت‌افزار لازم است تا بین نرخ ساعت و تعداد چرخه مصالحه‌ای را اتخاذ کنند!

مثال

- کامپیوتر A: 2GHz clock, 10s CPU time داده شده است
- قصد داریم تا کامپیوتر B را به گونه‌ای طراحی کنیم تا
 - زمان CPU به 6s برسد
 - افزایش نرخ ساعت منجر به $1.2 \times$ clock cycles می‌شود
- نرخ ساعت کامپیوتر B چقدر باید باشد؟

$$\begin{aligned} \text{CPU Time} &= \text{CPU Clock Cycles} \times \text{Clock Cycle Time} \\ &= \frac{\text{CPU Clock Cycles}}{\text{Clock Rate}} \end{aligned}$$