

معماری کامپیوتر

جلسه یازدهم: برآورد کارایی

ارزیابی کارایی



- تعریف کارایی

- Performance: مفهوم کلی است به معنای ایده‌آل کردن فرایندی در سیستم که این فرایند برحسب کاربرد می‌تواند زمان پاسخ، کلاک، سرعت، توان مصرفی و ... باشد.

- Throughput: مجموع کاری که سیستم در زمان معین می‌تواند انجام دهد (سرعت اجرای برنامه در پردازنده)

- نقطه مشترک تمامی کاربران: سرعت کارکرد سیستم‌های کامپیوتری

$$\text{Performance} \propto \frac{1}{\text{Execution Time}}$$

ارزیابی کارایی



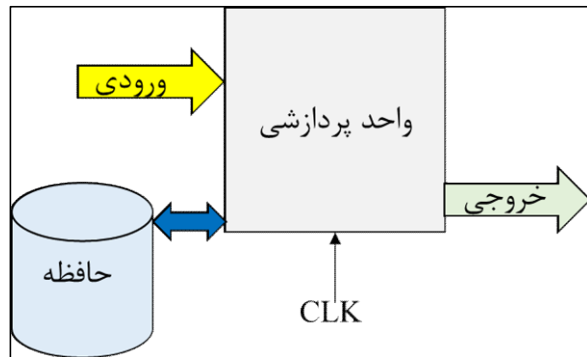
$$\text{Performance} \propto \frac{1}{\text{Execution Time}}$$

• منظور از زمان اجرا:

- مدت زمانی که طول می کشد تا یک سیستم کامپیوتری وظیفه مشخصی را به اتمام برساند
- شامل: زمان پردازش، دسترسی به حافظه، عملیات سیستم عامل و ...
- رابطه ذکر شده برای محاسبه کارایی استفاده می شود
- در ادامه هزینه هم در این رابطه وارد کرده و ارزیابی را جامع تر می کنیم



شیوه تخمین زمان اجرا



- مدل و اجزای سیستم‌های کامپیوتری

- انجام عملیات در این سیستم طی چندین کلاک

- فرکانس کاری پردازنده برابر f آنگاه طول هر کلاک برابر $1/f$

- اجرای هر برنامه تعداد مشخصی کلاک طول می‌کشد

$$\begin{aligned} \text{Execution Time} &= \text{تعداد کلاک‌های مصرفی برنامه} \times \frac{1}{f} \\ &= \sum_{i=1}^n \frac{\text{تعداد کلاک‌های مصرفی دستور } i \text{ ام}}{f} \end{aligned}$$

معیارهای تخمین کارایی



• پارامتر CPI: Clock Per Instruction

• CPI_i : تعداد کلاک لازم برای اجرای دستور i

• CPI پردازنده یا برنامه: میانگین CPI دستورات آن

$$\text{Execution time} = \sum_{i=1}^n \frac{CPI_i}{f}$$

$$CPI = \frac{\text{تعداد کلاک مصرفی برای دستور } i \text{ ام } \sum_{i=1}^n}{\text{تعداد دستورات } set \text{ instruction}}$$

نکته: در محاسبه زمان اجرا علاوه بر زمان پردازش، عوامل دیگری نیز دخیل هستند که فعلا از آن‌ها صرف نظر شده است

معیارهای تخمین کارایی



- پارامتر IPC: Instruction Per Clock

- تعداد دستورهای اجرا شده در هر کلاک

$$IPC_x = \frac{K}{\text{تعداد کلاک مصرفی برای هر } K \text{ دستور مجموعه } X}$$

- بدیهی است که: $CPI * IPC = 1$

- واحد CPI و IPC؟

معیارهای تخمین کارایی



• پارامتر MIPS: Million Instructions Per Second

$$\text{MIPS} = \frac{\text{تعداد دستورات اجرا شده}}{\text{زمان طی شده به ثانیه}} \times \frac{1}{10^6} \longrightarrow \text{چند میلیون دستور در یک ثانیه}$$

• در صورت اجرای دو برنامه کاملاً یکسان، سیستم با MIPS بالاتر، کارایی بیشتری دارد

• در حالتی که دستورات متفاوت باشند نمی‌توان مقایسه کرد

معیارهای تخمین کارایی



- در نتیجه کارایی توسط سه پارامتر اساسی قابل تخمین است:
- CPI دستورات برنامه که وابسته است به ساختمان و نحوه طراحی ریزپردازنده
- تعداد دستورات برنامه که وابسته است به مجموعه دستورالعمل و الگوریتم کامپایل برنامه
- فرکانس کاری ریزپردازنده

$$\text{زمان اجرای برنامه} = \frac{CPI \times \text{تعداد دستورات}}{f}$$

معیارهای تخمین کارایی



$$\text{زمان اجرای برنامه} = \frac{CPI \times \text{تعداد دستورات}}{f}$$

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

Algorithm	Instruction count, possibly CPI
Programming language	Instruction count, CPI
Compiler	Instruction count, CPI
Instruction set architecture	Instruction count, clock rate, CPI

ارزیابی کارایی



• مثال: دو کامپیوتر با مشخصات زیر داریم:

A: $\text{CPI} = 2$ و $\text{cycle time} = 250 \text{ ps}$

B: $\text{CPI} = 1.2$ و $\text{cycle time} = 500 \text{ ps}$

برای اجرای یک برنامه و دستورالعمل‌های یکسان، کدام سریع‌تر است و چه مقدار؟

ارزیابی کارایی



• مثال: دو کامپیوتر با مشخصات زیر داریم:

A: $\text{CPI} = 2$ و $\text{cycle time} = 250 \text{ ps}$

B: $\text{CPI} = 1.2$ و $\text{cycle time} = 500 \text{ ps}$

برای اجرای یک برنامه و دستورالعمل‌های یکسان، کدام سریع‌تر است و چه مقدار؟

$$\text{Execution time}_A = 2 * 250 = 500 \text{ ps}$$

$$\text{Execution time}_B = 1.2 * 500 = 600 \text{ ps}$$

$$\begin{aligned} \text{Performance}_A / \text{Performance}_B &= \text{Execution time}_B / \text{Execution time}_A \\ &= 600 / 500 = 1.2 \end{aligned}$$

ارزیابی کارایی



- **مثال:** برنامه‌ای در کامپیوتر A با نرخ کلاک 400 MHz در ۱۰ ثانیه اجرا می‌شود و در کامپیوتر B همین برنامه در ۶ ثانیه. اگر بدانیم هر دستور در کامپیوتر B به ۱/۲ برابر کلاک بیشتر نسبت به A نیاز دارد، نرخ کلاک B را بدست آورید.

$$\text{CPI_B} = 1.2 \text{ CPI_A}$$

$$\text{Execution_time_A} = 10 = \text{CPI_A} / 400 \text{ MHz} \rightarrow \text{CPI_A} = 4 \text{ e9}$$

$$\text{CPI_B} = 1.2 * 4\text{e9} = 4.8\text{e9}$$

$$\text{Execution_time_B} = 6 = 4.8\text{e9} / f_B \rightarrow f_B = 4.8\text{e9} / 6 = 800 \text{ MHz}$$

ارزیابی کارایی



- **مثال:** برنامه‌ای روی دو کامپیوتر با فرکانس 1MHz با دو کامپایلر مختلف اجرا شده است. اگر CPI کلاس دستورات A، B و C و تعداد دستورات هر کلاس در هر برنامه مطابق جدول باشد، مشخص کنید کدام کامپیوتر کارایی بالاتری دارد؟ کدام MIPS بیشتری دارد؟ CPI میانگین هر کامپیوتر چقدر است؟

	CPI for each instruction class		
	A	B	C
CPI	1	2	3
Code sequence	Instruction counts for each instruction class		
	A	B	C
1	2	1	2
2	4	1	1

ارزیابی کارایی



• حل:

$$\text{Execution Time}_1 = (2*1 + 1*2 + 3*2) / 1\text{MHz} = 10 \text{ micro second}$$

$$\text{Execution Time}_2 = (1*4 + 2*1 + 3*1) / 1\text{MHz} = 9 \text{ micro second}$$

$$\text{MIPS}_1 = (5 / 10\text{e-}6) * (1/1\text{e}6) = 0.5$$

$$\text{MIPS}_2 = (6 / 9\text{e-}6) * (1/1\text{e}6) = 2/3 = 0.66$$

$$\text{Performance} \propto \frac{1}{\text{Execution Time}} \longrightarrow \text{کارایی ۲} < \text{کارایی ۱}$$

ارزیابی کارایی



• ادامه حل:

$$\text{CPI}_1 = (2*1 + 1*2 + 3*2) / 5 = 2$$

$$\text{CPI}_2 = (1*4 + 2*1 + 3*1) / 6 = 3/2 = 1.5$$

دو سبک طراحی

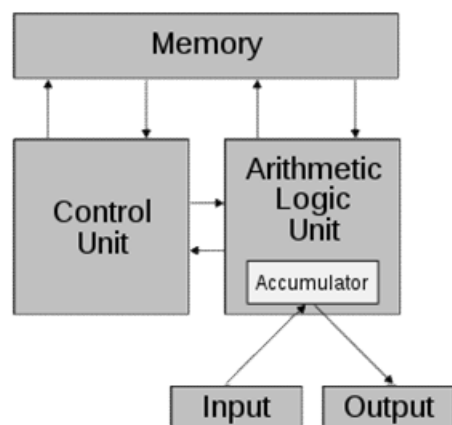


- کامپیوترهای اولیه: مجموعه دستورالعمل‌های محدود و ساده
 - ناکارآمدی کامپیوترهای اولیه با پیشرفت فناوری و پیچیده شدن کاربردها
 - سوق طراحی به سمت کامپیوترها با مجموعه دستورالعمل‌های پیچیده
 - تبعات طراحی کامپیوترهای پیچیده:
 - پیچیدگی و گران شدن هزینه طراحی
 - افزایش زمان تولید و عرضه به بازار
 - پیچیدگی عیب‌یابی و تست تضمین کیفیت محصولات
- ← بازگشت طراحان به طراحی اولیه

کامپیوترهای RISC و CISC



- RISC: Reduced Instruction Set Computers
- CISC: Complex Instruction Set Computers
- دو سبک طراحی و ایده در طراحی معماری سیستم‌های کامپیوتری برپایه مدل Von Neumann



کامپیوترهای RISC و CISC

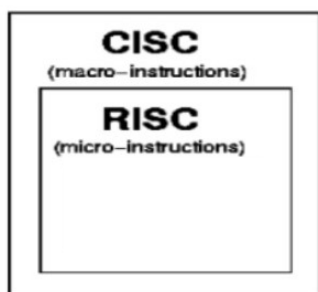


CISC	RISC
دستورات پیچیده‌تر دارند (تعداد کلاک‌های اجرای هر دستور (CPI) زیاد است)	دستورات ساده‌تری دارند (تعداد کلاک‌های اجرای هر دستور (CPI) ثابت و کمتر است)
زیاد بودن تعداد دستوراتالعمل‌ها	محدود بودن تعداد دستوراتالعمل‌ها
زیاد و متنوع بودن شیوه‌های آدرس‌دهی	کم بودن شیوه‌های آدرس‌دهی
استفاده از ثبات‌های بیشتر	استفاده از ثبات‌های کم‌تر
تعداد کم دستورات در برنامه	تعداد زیاد دستورات در برنامه
تعداد زیاد و متنوع عملوندهای دستورات	تعداد کم عملوندهای دستورات
متغیر و زیاد بودن تعداد بایت مصرفی هر دستور	ثابت و کم بودن تعداد بایت مصرفی هر دستور

کامپیوترهای RISC و CISC



- از نظر کاربرد، هریک استفاده‌های خود را دارند
- استفاده از CISC در سرورها و سیستم‌هایی که محاسبات پیچیده و زیاد لازم است
- استفاده از RISC در کامپیوترهای معمولی با کاربردهای عادی
- استفاده از هردونوع معماری در قالب یک طرح:
- پردازنده پنتیوم اینتل که هسته RISC و پوشش بیرونی CISC دارد. در واقع دستورات پیچیده را با تبدیل به دستورات ساده اجرا می‌کند



قانون Amdahl



- در طراحی سیستم‌های کامپیوتری اجزای مختلفی باید در نظر گرفته شوند
- کارایی و بازده بالا در طراحی سیستم‌های کامپیوتری بسیار حائز اهمیت است
- تشخیص اینکه سرمایه‌گذاری روی کدام بخش از سیستم منجر به بازده بالاتری می‌شود: قانون امدال
- بهبود کارایی سیستم در نتیجه بهبود کارایی اجزای آن حاصل می‌شود
- بهبود اجزایی از سیستم که محاسبات و عملیات سنگین را برعهده دارند منجر به بهبود بیشتر در کارایی می‌شوند
- بخش ترتیبی سیستم که قابلیت اجرای موازی ندارد: محدودیت سرعت اجرا
- پارامتر تسریع (speed up):
$$\frac{\text{Old execution time}}{\text{New execution time}}$$



قانون Amdahl



- اگر نسبت دستورات ترتیبی یک برنامه که قابلیت تسریع در آن‌ها نیست به کل دستورات آن f باشد، و بقیه دستورات را بتوان با اجرای موازی p برابر سریع‌تر کرد، میزان افزایش سرعت اجرای برنامه (speed up) حالت دوم به حالت اول برابرست با:

$$\text{Speed-Up} = \frac{1}{f + \frac{1-f}{p}} = \frac{p}{(p-1).f + 1}$$

- در حالت کلی اگر a_i بخش از برنامه را بتوان به اندازه p_i موازی کرد:

$$\text{Speed-Up} = \frac{1}{(1 - \sum_{i=1}^n a_i) + \sum_{i=1}^n (\frac{a_i}{p_i})}$$



قانون Amdahl

