

# Fundamentos de Arquitetura de Computadores

Tiago Alves

Faculdade UnB Gama  
Universidade de Brasília



## Conceitos Básicos de FAC

- Projetar pela Lei de Moore.
- Usar abstrações para simplificar
- Agilizar o caso comum
- Aumentar o **desempenho** por paralelismo, pipeline e predição
- Explorar a hierarquia de memórias
- Aumentar a confiabilidade pela redundância



## Desempenho

- Meça, resuma e informe
- Faça escolhas inteligentes
- Veja através da propaganda de marketing
- Vital para entender a motivação organizacional subjacente



## Desempenho

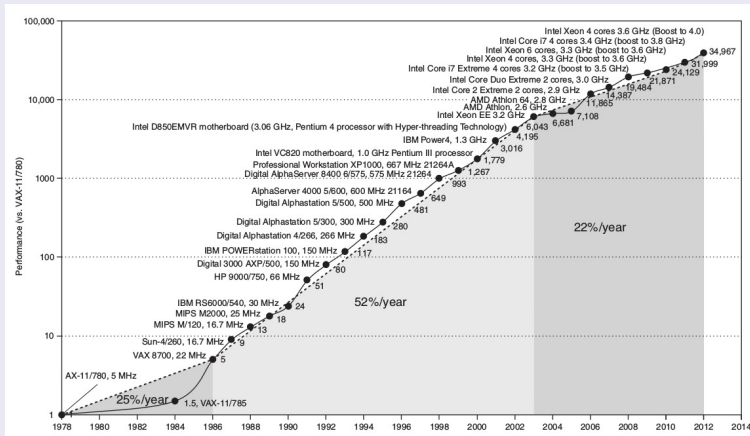
Q:

- Por que alguns hardwares são melhores do que outros para diferentes programas?
- Que fatores do desempenho de sistema são relacionados ao hardware? Por exemplo: Precisamos de uma nova máquina ou de um novo sistema operacional?
- Como o conjunto de instruções da máquina afeta o desempenho?

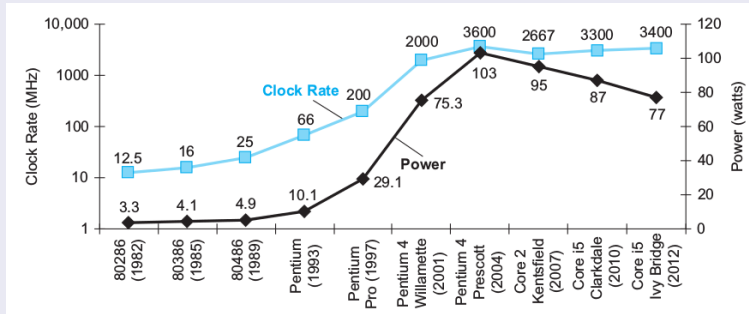


## Evolução do Desempenho das CPUs

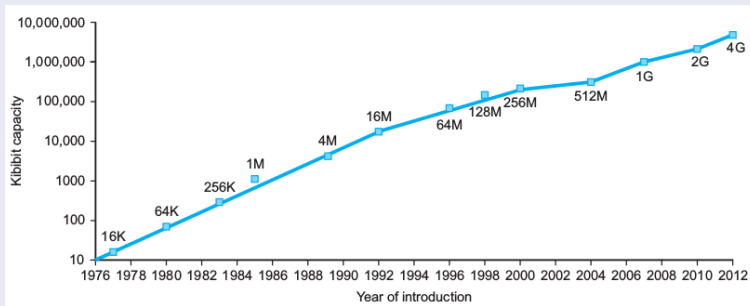
Resultados normalizados em relação ao desempenho de uma CPU VAX 11/780 usando como métrica SPECInt.



## Evolução da Velocidade de Clock e Potência (W)



## Evolução da Capacidade de Memória



## Avaliação de Desempenho: Aviões

Airplane	Passenger capacity	Cruising range (miles)	Cruising speed (m.p.h.)	Passenger throughput (passengers $\times$ m.p.h.)
Boeing 777	375	4630	610	228,750
Boeing 747	470	4150	610	286,700
BAC/Sud Concorde	132	4000	1350	178,200
Douglas DC-8-50	146	8720	544	79,424

- O quanto mais rápido é o Concorde comparado com o 747?
- O quanto maior é o 747 do que o Douglas DC-8?

Qual avião você escolheria para:

- Levar 1 passageiro de Recife para Lisboa? E 400?
- Levar 100 passageiros de Recife para Paris? E 300?
- Levar 100 passageiros de Brasília para Auckland? E 400?





## Desempenho do computador

Tempo de resposta (latência + execução)

- Quanto tempo leva para meu trabalho ser realizado?
- Quanto tempo leva para realizar um trabalho?
- Quanto tempo preciso esperar para a consulta ao banco de dados?

Vazão (throughput)

- Quantos trabalhos a máquina pode realizar em um intervalo de tempo?
- Qual é a velocidade de execução média?
- Quanto trabalho está sendo feito?

Se atualizarmos uma máquina com um novo processador (mais rápido), em que melhoramos?

Se acrescentarmos uma máquina ao laboratório, em que melhoramos?

**Q:** Diminuir o tempo de resposta aumenta a vazão?



## Tempo de execução

### Tempo decorrido

- conta tudo (acessos a disco e a memória, E/S etc.)
- um número útil, mas normalmente não é ideal para fins de comparação

### Tempo de CPU

- não conta E/S ou tempo gasto executando outros programas.
- Pode ser dividido em: tempo de sistema e tempo de usuário.

Nosso foco na disciplina: **tempo de CPU do usuário**, ou seja, tempo gasto executando as linhas de código que estão no nosso programa.



## Definição de desempenho

Para um programa sendo executado na máquina X:  $\text{Desempenho}_X = \frac{1}{\text{Tempo de Execução}_X}$

Fator de Desempenho: "X é n vezes mais rápido do que Y", onde  $n = \frac{\text{Desempenho}_X}{\text{Desempenho}_Y} = \frac{\text{Tempo de Execução}_Y}{\text{Tempo de Execução}_X}$

Exemplo:

- a máquina A executa um programa em 20 segundos
- a máquina B executa o mesmo programa em 25 segundos.

Qual o fator de desempenho de A em relação a B?



## Ciclos de clock

Em vez de informar o tempo de execução em segundos, normalmente usamos ciclos:

Tempo de Execução da CPU para um programa =

Ciclos de clock da CPU para um programa  $\times$  Tempo de ciclo de clock, de outra forma,

$$\frac{\text{segundos}}{\text{programa}} = \frac{\text{ciclos}}{\text{programa}} \times \frac{\text{segundos}}{\text{ciclos}}$$

Os pulsos de clock indicam quando iniciar as atividades (abstração)

- tempo de ciclo (período): tempo entre os pulsos de clock [segundos]
- velocidade de clock (frequência): ciclos por segundo [hertz]

Um sinal de relógio de 4GHz possui um tempo de ciclo de período =  $\frac{1}{\text{frequência}} = \frac{1}{4 \times 10^9} = 250 \times 10^{-12} = 250\text{ps}$



## Como melhorar o desempenho

$$\frac{\text{segundos}}{\text{programa}} = \frac{\text{ciclos}}{\text{programa}} \times \frac{\text{segundos}}{\text{ciclos}}$$

Portanto, para melhorar o desempenho (tudo mais sendo igual), você pode (aumentar ou diminuir?)

- o número de ciclos necessários para um programa, ou
- o tempo de ciclo de clock ou, dito de outra maneira,
- a frequência de clock.



## Quantos ciclos são necessários para um programa?

Poderíamos considerar que o número de ciclos é igual ao número de instruções do programa?

Essa suposição é **geralmente incorreta**; diferentes instruções geralmente levam a diferentes períodos em diferentes máquinas.

- A multiplicação leva mais tempo do que a adição;
- As operações de ponto flutuante levam mais tempo do que as operações de inteiros;
- Acessar a memória leva mais tempo do que acessar os registradores;

Importante: mudar o tempo de ciclo normalmente muda o número de ciclos necessários para várias instruções



## Medidas Básicas

$$\frac{\text{segundos}}{\text{programa}} = \frac{\text{instruções}}{\text{programa}} \times \frac{\text{ciclos\_clock}}{\text{instrução}} \times \frac{\text{segundos}}{\text{ciclo\_clock}}$$

- Tempo de Execução da CPU para um programa: quantidade segundos demandada para executar o programa
- Contagem de Instruções: instruções executadas para o programa
- Ciclos de clock por instrução (CPI): número médio de ciclos de clock por instrução
- Tempo do ciclo de clock: segundos por ciclo de clock ou simplesmente período.



## Visão Geral Apresentada

Um determinado programa exigirá:

- um determinado número de instruções (instruções de máquina)
- um determinado número de ciclos
- um determinado número de segundos

Vocabulário que relaciona essas quantidades:

- tempo de ciclo (segundos por ciclo)
- frequência de clock (ciclos por segundo)
- CPI (ciclos por instrução): uma aplicação com excessivo uso de ponto flutuante pode ter uma CPI mais alta





## Desempenho

O desempenho é determinado pelo tempo de execução.

Qualquer uma das outras variáveis igualam o desempenho?

- número de ciclos para executar o programa?
- número de instruções no programa?
- número de ciclos por segundo (frequência)?
- número médio de ciclos por instrução?
- número médio de instruções por segundo?

Armadilha comum: pensar que uma das variáveis é indicadora do desempenho, quando na realidade não é.



## Desempenho

Se duas máquinas possuem a mesma ISA (Arquitetura do Conjunto de Instruções), qual das quantidades serão idênticas?

- velocidade de clock;
- CPI (Ciclos por Instrução);
- tempo de execução;
- número de instruções.



## Exemplo de CPI

Suponha que tenhamos duas implementações da mesma arquitetura do conjunto de instruções (ISA)

Para um determinado programa,

- A máquina A tem um tempo de ciclo de clock de 250 ps e uma CPI de 2,0;
- A máquina B tem um tempo de ciclo de clock de 500 ps e uma CPI de 1,2;

**Q:** Que máquina é mais rápida para esse programa e o quanto?



## Exemplo de CPI

**A:**

Premissas:

- Mesmo ISA, logo, o programa tem o mesmo número de instrução (I);
- Ciclos de clock da CPUa = 2,0 I
- Ciclos de clock da CPUb = 1,2 I

Desenvolvimento:

- Tempo de CPUa = Ciclos de Clock de A x tempo do ciclo de clockA =  $1 \times 2,0 \times 250\text{ps} = 500 \times 1 \text{ ps}$
- Tempo de CPUb = Ciclos de Clock de B x tempo do ciclo de clockB =  $1 \times 1,2 \times 500\text{ps} = 600 \times 1 \text{ ps}$

Dessa forma:  $\frac{\text{Desempenho\_CPUa}}{\text{Desempenho\_CPUb}} = \frac{\text{Tempo de Execução}_b}{\text{Tempo de Execução}_a} = \frac{600 \times I}{500 \times I} = 1,2$

Conclui-se que a CPUa é 1,2 vezes mais rápida que o CPUb para esse programa.



## Componentes que afetam os fatores

Componente	Afeta o que?	Como?
Algoritmo	Contagem de Instruções e possivelmente CPI	Número e tipo de instruções
Linguagem de Programação	Contagem de Instruções e CPI	Instruções da linguagem são traduzidos para instruções do processador
Compilador	Contagem de Instruções e CPI	Eficiência do compilador.
Conjunto de Instruções	Contagem de Instruções, frequência de clock e CPI	Afeta os 3 aspectos do desempenho



## Comparando Segmentos de Código

Um projetista de compilador está tentando decidir entre duas sequências de código para um determinada máquina. Baseado na implementação de hardware, existem três classes diferentes de instruções: Classe A, Classe B e Classe C, e elas exigem um, dois e três ciclos (CPI), respectivamente.

- A primeira sequência de código possui 5 instruções: 2 de A, 1 de B e 2 de C.
- A segunda sequência possui 6 instruções: 4 de A, 1 de B e 1 de C.

**Q:** Que sequência será mais rápida? O quanto mais rápida? Qual é a CPI para cada sequência?



## Comparando Segmentos de Código

**A:**

Premissas:

- Seq. 1:  $2 + 1 + 2 = 5$  inst.
- Seq. 2:  $4 + 1 + 1 = 6$  inst.
- Pode-se utilizar a seguinte expressão para definir o números de clock da CPU:  
$$\text{ciclos\_clock\_CPU} = \sum_{i=1}^n \text{CPI}_i \times \text{Instrução}_i$$

Desenvolvimento:

- Para a CPUseq1 =  $(2 \times 1) + (1 \times 2) + (2 \times 3) = 10$  ciclos
- Para a CPUseq2 =  $(4 \times 1) + (1 \times 2) + (1 \times 3) = 9$  ciclos

Conclusão: Seq. Cod.2 mais rápida!!

$$\text{CPI} = \frac{\text{ciclos\_clock\_cpu}}{\text{contagem\_instruções}}$$
. Logo:  $\text{CPI1} = 10/5 = 2$  e  $\text{CPI2} = 9/6 = 1,5$



## Outra medida de Desempenho: Consumo de Potência.

- Os processadores atuais são construídos utilizando tecnologia de circuitos denominada CMOS. (CMOS: Complementary Metal Oxide Semiconductor)
- O consumo de energia (Watts) é realizado no processo de “chaveamento” (Switching) das operações nos elementos CMOS;
- A potência é medida levando em consideração: Carga capacitiva de cada transistor, voltagem aplicada, frequência de “chaveamento”.

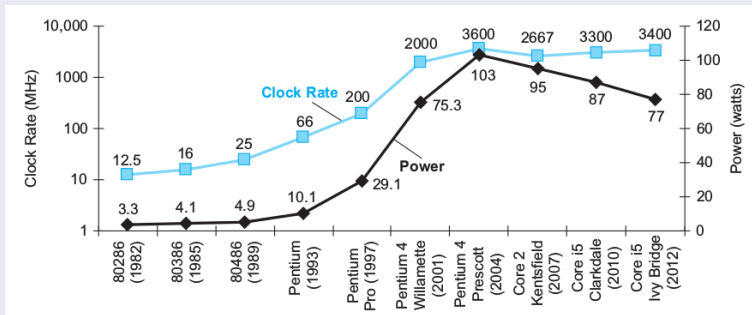
Green Computing





## Evolução da Velocidade de Clock e Potência (W)

Revisitando...



## Outra medida de Desempenho: Consumo de Potência.

Uma relação para esta medida:

$$\text{Potência} = \text{Carregamento Capacitivo} \times \text{Tensão Aplicada}^2 \times \text{Frequência de Chaveamento}$$

Essas variáveis são dependentes:

- Frequência de Chaveamento: Velocidade de Clock
- Carregamento Capacitivo: Número de transistores conectados a uma saída
- Tensão aplicada: Varia com a tecnologia (Anos 80: 5V e atualmente 1V aumentando o poder de processamento em 30x!)

Algumas melhorias: Resfriamento Forçado. Mesma técnica utilizada no Overclocking.



## Unidade de Medida: MIPS

$$\text{MIPS} = \frac{\text{Contador\_Instruções}}{\text{Tempo\_Execução}} \times 10^{-6}$$

Vantagens:

- Fácil de entender.
- Um computador é capaz de processar 100 MIPS é mais rápido que outro de 50 MIPS

Porém:

- Não leva em consideração a capacidade das instruções.
- O MIPS varia entre programas no mesmo processador.
- O MIPS pode variar inversamente com o desempenho!



## Unidade de Medida: MIPS

Dois compiladores diferentes estão sendo testados para uma máquina de 4GHz com três classes diferentes de instruções: Classe A, Classe B e Classe C, e elas exigem um, dois e três ciclos, respectivamente. Ambos os compiladores são usados para produzir código para um grande software.

- O código do primeiro compilador usa 5 bilhões de instruções da Classe A, 1 bilhão de instruções da Classe B e 1 bilhão de instruções da Classe C.
- O código do segundo compilador usa 10 bilhões de instruções da Classe A, 1 bilhão de instruções da Classe B e 1 bilhão de instruções da Classe C.

Que sequência será mais rápida de acordo com o MIPS?

Que sequência será mais rápida de acordo com o tempo de execução?

