

# Aula 1: Abstrações e Tecnologias Computacionais

Professor(a): Virgínia Fernandes Mota  
<http://www.dcc.ufmg.br/~virginiaferm>

OCS (TEORIA) - SETOR DE INFORMÁTICA



# A revolução dos computadores

- Progresso na tecnologia dos computadores
  - Lei de Moore: Será?
- Novas aplicações surgem rapidamente
  - Computadores em automóveis
  - Smart Phones
  - Mecanismos de busca (Street View Google Data Center\*)
  - World Wide Web
- Computadores são onipresentes!

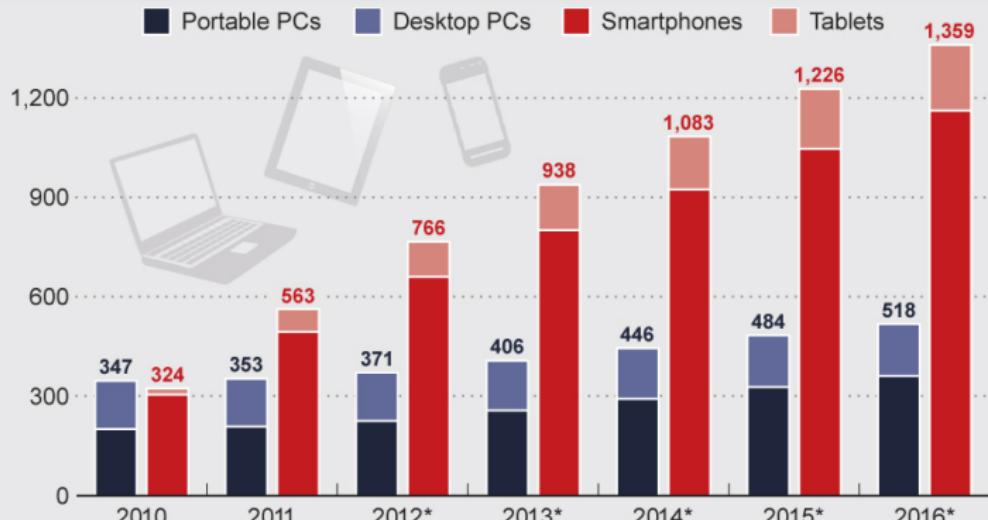
# Classes de computadores

- **Pessoais** (Desktop, Notebook): Propósito geral, vários programas, sujeito a custo/desempenho.
- **Servidores**: baseado em redes; grande capacidade, desempenho e confiança.
- **Supercomputadores**: alto desempenho de custo bem alto, grande capacidade, configurado como servidores (aplicações científicas).
- **Computadores embutidos**: dentro de algum outro dispositivo (carros, eletrodomésticos), propósito específico.

# Classes de computadores - Era Pós-PC

## The Post-PC Era Has Arrived

Global smartphone, tablet and PC shipments (in millions)



- Dispositivos móveis (Smartphones, tablets): funcionamento por bateria, conecta à internet, "menos custo".
- Nuvem: Warehouse Scale Computers (datacenter); software como serviço.

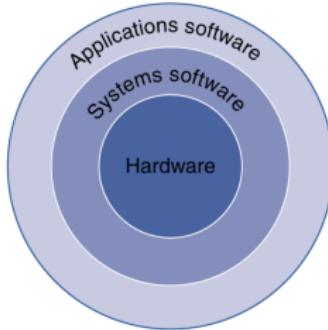
# O que iremos aprender

- Como programas são traduzidos para linguagem de máquina e como o hardware a executa.
- A interface hardware/software.
- O que determina o desempenho de um programa e como melhorar.
- O que determina o desempenho do hardware e como melhorar.
- O que é o processamento paralelo.
- Como o Sistema Operacional (SO) funciona.

- Algoritmo
  - Determina o número de operações executadas.
- Linguagem de programação, compilador e arquitetura.
  - Determina o número de instruções de máquina executadas por operação.
- Processador e memória
  - Determina a velocidade que as instruções serão executadas.
- Sistema de entrada/saída (I/O) - Incluindo o SO
  - Determina a velocidade de operações de entrada e saída.

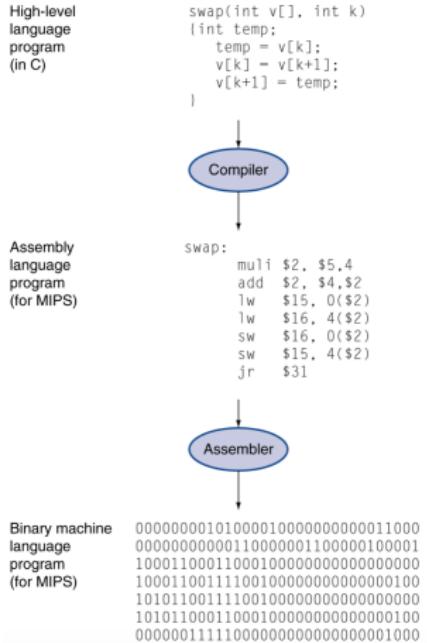
# Por baixo do seu programa

- Software de Aplicação
  - Escrita em linguagem de alto nível
- Softwares de Sistema
  - Compilador: Traduz a linguagem de alto nível para linguagem de máquina.
  - Sistema Operacional: gerencia entrada/saída, uso de memória, escalonamento dos processos e recursos.
- Hardware
  - Processador, memória, controles de entrada/saída.

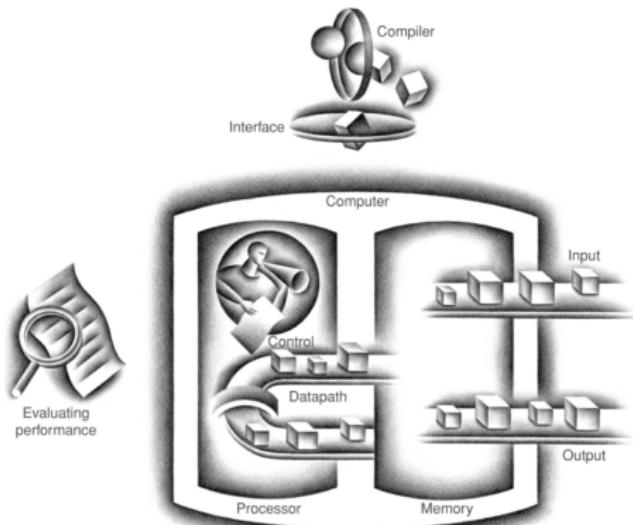


# Por baixo do seu programa

Programa em linguagem de alto nível → Programa em Assembly  
→ Programa binário em linguagem de máquina



# Componentes de um computador



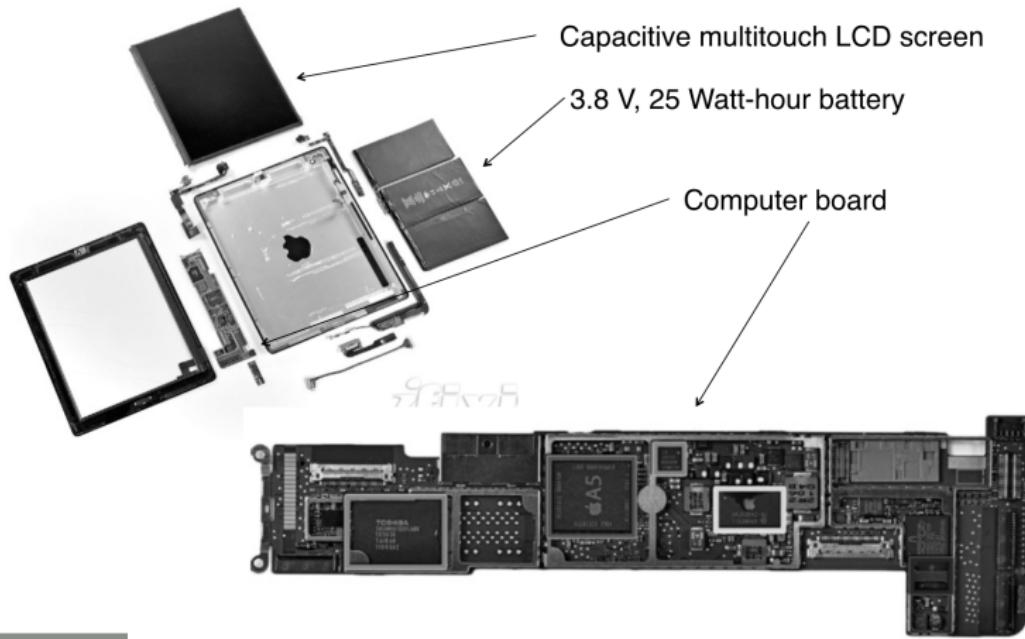
- Cinco componentes de um computador: entrada, saída, memória, caminho de dados (datapath) e controle, sendo que, às vezes, datapath e controle são chamados de processador.

# Touchscreen - Tela sensível ao toque

- Dispositivos da era Pós-PC.
- Substitui teclado e mouse.
- Resistiva vs. Capacitiva
  - Resistiva: Palm com canetinha.
  - Capacitiva: A maior parte dos dispositivos de hoje, permite multi-touch (toques simultâneos).

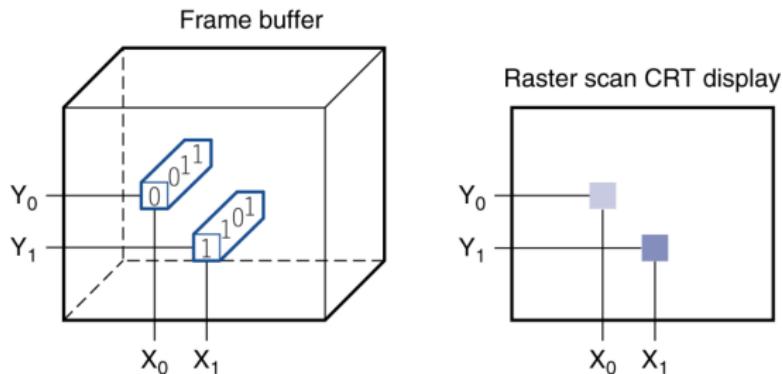


# Touchscreen - Tela sensível ao toque



# Diante do espelho

- Monitores de tubos de raios catódicos (CRT), monitores de tela plana, cristal líquido, LED.



# Abrindo o gabinete ou a tampa



# Abrindo o gabinete ou a tampa

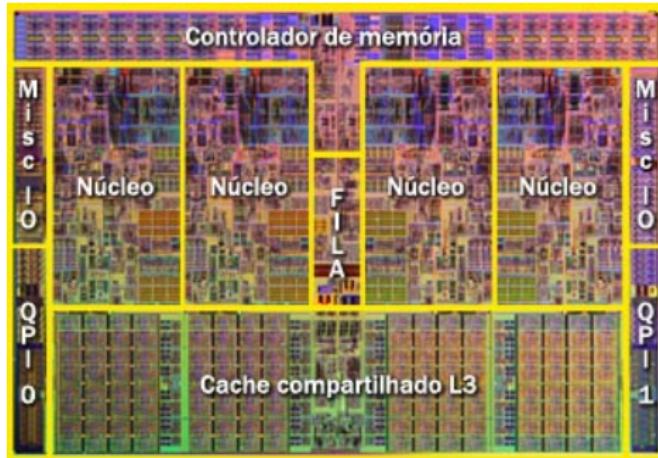
- Placa-mãe
- Memória principal e secundária
- Unidade central de processamento (CPU)

# Por dentro do processador (CPU)

- Caminho de dados (datapath): realiza as operações aritméticas.
- Controle: diz ao caminho de dados, à memória e aos dispositivos de E/S o que fazer com as instruções de um programa.
- Cache: memória pequena e rápida, um buffer para a memória RAM (que falaremos mais tarde).

# Por dentro do processador (CPU)

Processador i7



- Memória volátil vs não-volátil.
- Memória volátil: Memória principal do computador, a RAM (Random Access Memory). Perde as informações quando desligada.
- Memória não-volátil: Discos magnéticos, discos de estado sólido, memória flash, mídias óticas. Uma memória que não esquece mesmo quando é desligada.
- Qual a diferença quanto a volatilidade, tempo de acesso e custo entre esses tipos de memória?

- Redes - Network
  - Local area network (LAN): Ethernet
  - Wide area network (WAN): Internet
  - Wireless network: WiFi, Bluetooth

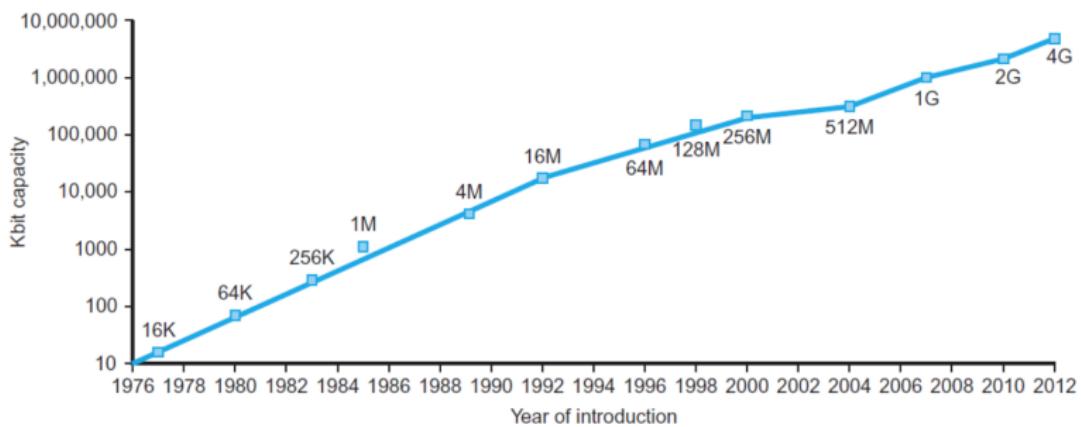
# Tendências da tecnologia

- Os processadores e a memória melhoraram em uma velocidade espantosa!
- A tecnologia continua a evoluir:
  - aumento da capacidade e desempenho
  - custo reduzido
- Circuito integrado: a grande revolução (combinação de transistores em um único chip).

# Tendências da tecnologia

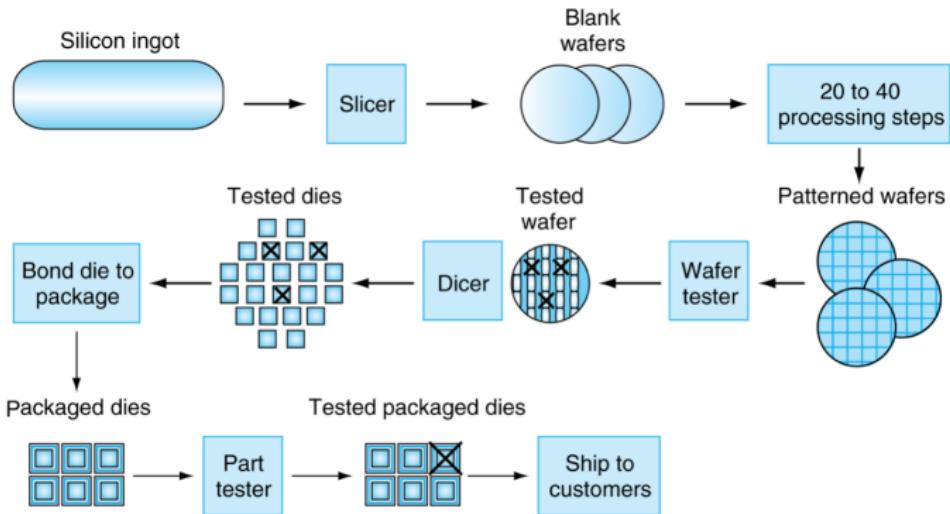
Ano	Tecnologia	Desempenho relativo/custo unitário
1951	Válvula	1
1965	Transistor	35
1975	Circuito integrado (IC)	900
1995	Very large scale IC (VLSI)	2,400,000
2013	Ultra large scale IC (ULSI)	250,000,000,000

# Tendências da tecnologia



DRAM capacity

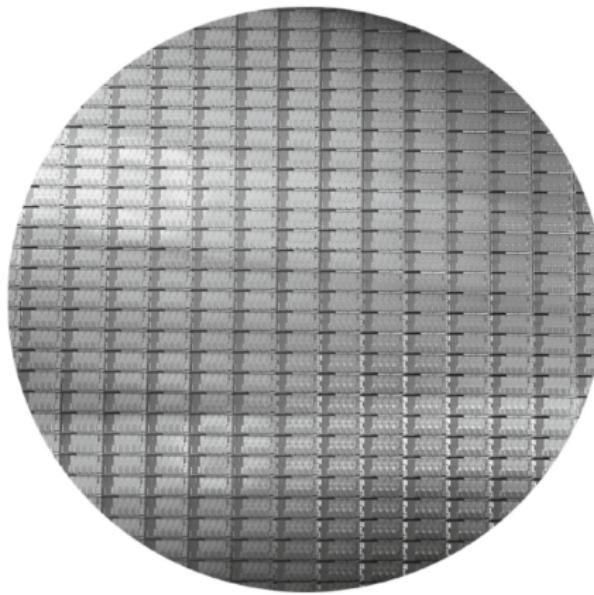
# Fabricando um chip



- Aproveitamento (yield) de um processo: A porcentagem de dies bons do número total de dies em um wafer.

# Fabricando um chip

## Wafer do Intel Core i7



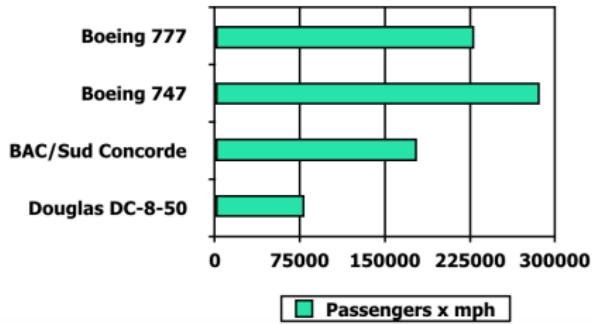
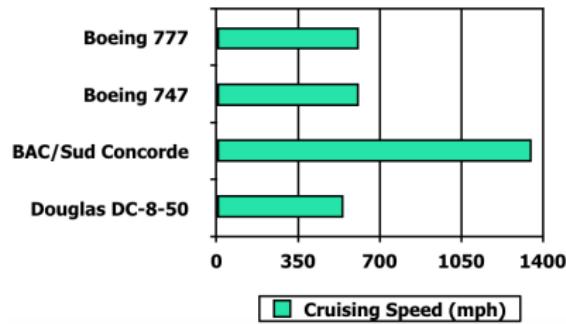
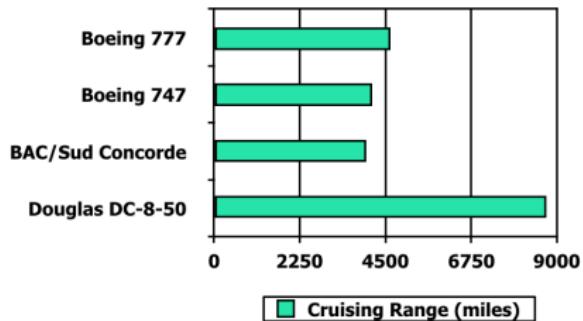
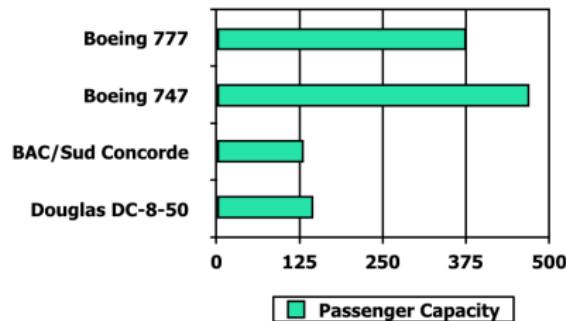
- 300mm wafer, 280 chips, tecnologia 32nm
- Cada chip tem 20.7 x 10.5 mm

# Fabricando um chip - o custo de um IC

- $Custo\ por\ die = \frac{Custo\ por\ wafer}{Dies\ por\ wafer \times Aproveitamento}$
- $Custo\ por\ wafer \approx \frac{Area\ do\ Wafer}{Area\ do\ Die}$
- $Aproveitamento = \frac{1}{(1 + (Defeitos\ por\ area \times Area\ do\ Die/2))^2}$

# Fabricando um chip - definindo desempenho

- Qual avião tem o melhor desempenho (performance)?



- Tempo de resposta e Vazão (*Response time* e *Throughput*).
  - Tempo de resposta: Quanto tempo uma tarefa demora para ser executada.
  - Vazão: Trabalho total executado por unidade de tempo (tarefas por hora, por exemplo).
- Como tempo de resposta e vazão são afetados por:
  - Colocar um processador mais rápido?
  - Adicionar mais processadores?
- **Vamos focar no tempo de resposta por enquanto...**

- Seja: Desempenho = 1/Tempo de execução
- "X é  $n$  vezes mais rápido que Y"

$$\frac{\text{Desempenho } X}{\text{Desempenho } Y} = \frac{\text{Tempo de execução } Y}{\text{Tempo de execução } X} = n$$

- Exemplo: Um mesmo programa roda em 10s no processador A e em 15s no processador B. Qual é o desempenho relativo dos processadores?

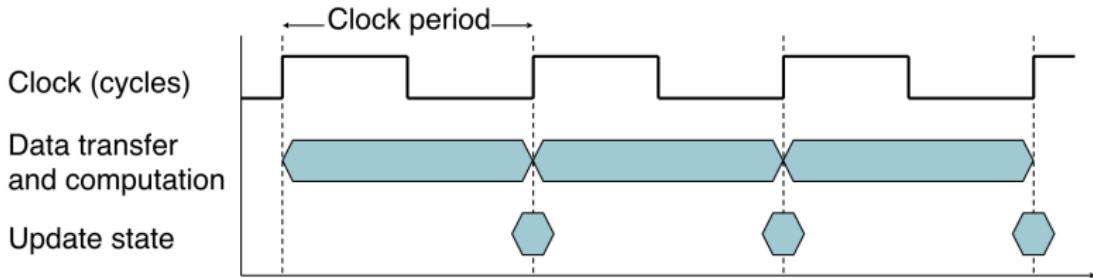
Tempo de Execução B / Tempo de Execução A = 15s / 10s = 1,5

Logo A é 1,5 vezes mais rápido que B.

- Tempo decorrido:
  - Tempo total gasto, incluindo aspectos de processamento, E/S, overhead do SO, tempo ocioso.
  - Determina o desempenho do sistema.
- Tempo de CPU:
  - Tempo gasto executando a tarefa dada: Desconta E/S e outras tarefas compartilhadas.
  - Abrange tempo de usuário e sistema na CPU.
- Programas diferentes são afetados de maneira diferente pelo desempenho da CPU e do sistema.

Já calcularam o tempo de execução usando a biblioteca `time.h` do C?

# Clock da CPU (CPU clocking)



- Clock: Sinal de temporização usado em uma transmissão síncrona.
- Período (tempo) de clock: duração do ciclo de clock  
 $250\text{ps} = 0,25\text{ns} = 250 \times 10^{-12}\text{s}$
- Frequência de clock (taxa/rate): ciclos por segundo  
 $4,0\text{GHz} = 4000\text{MHz} = 4,0 \times 10^9\text{Hz}$

$$\begin{aligned} \text{Tempo de CPU} &= \text{Ciclos de Clock} \times \text{Tempo de ciclo de clock} = \\ &= \frac{\text{Ciclos de Clock}}{\text{Frequencia de clock}} \end{aligned}$$

- Desempenho melhorado por:
  - Reduzir número de ciclos de clock.
  - Aumentar a frequência de clock.
- Designer de hardware normalmente escolhe a taxa no lugar da contagem de clocks.

- Exemplo: O Computador A tem 2GHz clock e tempo de CPU 10s . Queremos desenvolver o computador B com 6s de tempo de CPU. Ele pode ter um clock mais rápido, mas gasta 20% ciclos de clock a mais. Quão rápido o clock de B deve ser?

- ① O que desejamos calcular? Quão rápido → velocidade
- ② Velocidade? Que termo mede velocidade? Ciclos por segundo → Hz → Frequência de clock
- ③ 20% ciclos de clock a mais? B gasta 1,2 Ciclos de clock de A

# Tempo de CPU

$$\text{Tempo de CPU} = \frac{\text{Ciclos de Clock}}{\text{Frequencia de clock}}$$

Logo,

$$\text{Frequencia de clock} = \frac{\text{Ciclos de Clock}}{\text{Tempo de CPU}}$$

$$\text{Ciclos de Clock} = \text{Frequencia de clock} \times \text{Tempo de CPU}$$

Assim,

$$\text{Frequencia de clock}_B = \frac{1,2 \text{ Ciclos de Clock}_A}{6s}$$

$$\text{Ciclos de Clock}_A = 2\text{GHz} \times 10s = 20 \times 10^9$$

$$\text{Frequencia de clock}_B = \frac{1,2 \times 20 \times 10^9}{6s} = 4\text{GHz}$$

B deve ter frequência de clock de 4GHz.

# Contagem de instruções e Ciclos por instrução (CPI)

*Ciclos de Clock = Contagem de instruções x CPI*

*Tempo de CPU = Contagem de instruções x CPI*

*x Tempo de ciclo de clock =*

$$= \frac{\text{Contagem de instruções x CPI}}{\text{Frequencia de clock}}$$

- Contagem de instruções para um programa: determinada por programação, arquitetura do conjunto de instruções (Instruction set architecture - ISA) e compilador.
- Média de ciclos por instrução: determinada pelo hardware da CPU e pelo conjunto de instruções a ser executado.

# Contagem de instruções e Ciclos por instrução (CPI)

- Exemplo: O computador A tem tempo de ciclo de clock de 250ps e CPI 2,0. O computador B tem tempo de ciclo de clock de 500ps e CPI 1,2. Ambos possuem a mesma ISA. Qual é o mais rápido e por quanto?

# Contagem de instruções e Ciclos por instrução (CPI)

- ① O que desejamos calcular? Quão rápido, mas não temos informação de frequência.
- ② Que informações temos? Tempo de ciclo e CPI.
- ③ Então? Precisamos saber o tempo de CPU, ele também nos diz qual é o mais rápido.
- ④ Mas e se não temos o número de instruções? Assumindo a mesma arquitetura, consideramos o mesmo número de instruções para os 2.

# Contagem de instruções e Ciclos por instrução (CPI)

$$\begin{aligned} \text{Tempo de CPU} &= \text{Contagem de instruções} \times \text{CPI} \\ &\quad \times \text{Tempo de ciclo de clock} \end{aligned}$$

Assim,

$$\text{Tempo de CPU}_A = I \times 2,0 \times 250\text{ps} = I \times 500\text{ps}$$

$$\text{Tempo de CPU}_B = I \times 1,2 \times 500\text{ps} = I \times 600\text{ps}$$

A é mais rápido que B! Quantas vezes mais rápido?

$$\frac{\text{Tempo de CPU}_B}{\text{Tempo de CPU}_A} = \frac{I \times 600\text{ps}}{I \times 500\text{ps}} = 1,2$$

A é 1,2 mais rápido que B, ou 20% mais rápido que B.

# Contagem de instruções (CI) e Ciclos por instrução (CPI)

- Existe diferentes classes de instruções que tem número de ciclos de clock diferentes?

$$\text{Ciclos de Clock} = \sum_{i=1}^n \text{Contagem de instrucoes}_i \times CPI_i$$

- Média ponderada de CPI

$$\begin{aligned} CPI &= \frac{\text{Ciclos de Clock}}{\text{Contagem de instrucoes}} = \\ &= \frac{\sum_{i=1}^n \text{Contagem de instrucoes}_i \times CPI_i}{\text{Contagem de instrucoes}} \end{aligned}$$

# Contagem de instruções (CI) e Ciclos por instrução (CPI)

- Exemplo: Sequências de códigos compilados usando instruções de classes A, B, C. Calcule a média de CPI.

Classe	A	B	C
CPI por classe	1	2	3
CI sequência 1	2	1	2
CI sequência 2	4	1	1

# Contagem de instruções (CI) e Ciclos por instrução (CPI)

$$\text{Ciclos de Clock} = \sum_{i=1}^n \text{Contagem de instrucoes}_i \times \text{CPI}_i$$

Sequência 1

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

$$\text{CI} = 5$$

$$\text{Média de CPI} = 10 / 5 = 2$$

Sequência 2

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

$$\text{CI} = 6$$

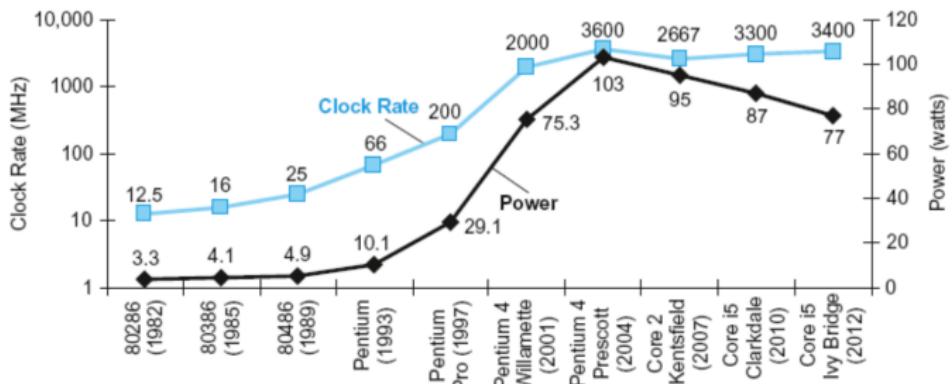
$$\text{Média de CPI} = 9 / 6 = 1,5$$

# Resumo do desempenho

$$\text{Tempo de CPU} = \frac{\text{Instrucoes}}{\text{Programa}} \times \frac{\text{Ciclos de clock}}{\text{Instrucao}} \times \frac{\text{Segundos}}{\text{Ciclos de clock}}$$

- Desempenho depende de:
  - Algoritmo: afeta CI, possivelmente CPI
  - Linguagem de programação: afeta CI, CPI
  - Compilador: afeta CI, CPI
  - Arquitetura do conjunto de instruções: afeta IC, CPI, Tc  
(tempo de clock)

# Tendência da potência



- Tecnologia CMOS (Complementary Metal Oxide Semiconductor): tecnologia dominante para os circuitos integrados.  
Potência = Carga capacitiva x Voltagem<sup>2</sup> x Frequência de chaveamento

Quanto maior a potência, mais rápido o processador, melhor ele é???

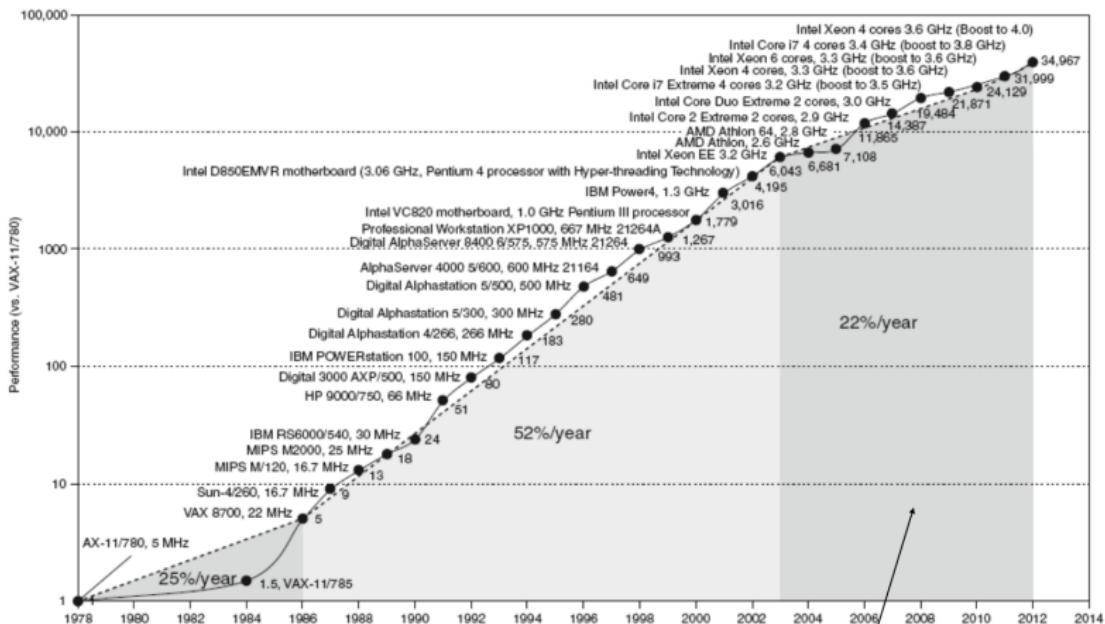
# Tendência da potência



# Reduzindo a potência

- Suponha uma nova CPU com Carga capacitiva 85% de uma antiga CPU, redução de 15% da voltagem e redução de 15% da frequênciā.
- Chegamos a um limite no qual não conseguimos mais reduzir nem voltagem, nem calor...
- Se não conseguimos mudar esses parâmetros, como é possível melhorar o desempenho?

# Reduzindo a potência



Constrained by power, instruction-level parallelism,  
memory latency

- Microprocessadores multicore: Mais de um processador por chip.
- Requer programação paralela explícita
  - Comparando com paralelismo em nível de instrução
    - Hardware executa múltiplas instruções ao mesmo tempo.
    - Escondido do programador.
  - Difícil de fazer
    - Programação para desempenho.
    - Balanceamento de carga.
    - Otimização de comunicação e sincronização.

- Programas usados para medir desempenho
- Standard Performance Evaluation Corp (SPEC): Desenvolve benchmarks para CPU, E/S, Web,...
- Aritmética de inteiros vs Aritmética de ponto flutuante
- Comparação de CPUs?

- **Armadilha:** Melhorar um aspecto do computador e esperar melhoria proporcional de todo o desempenho.
- **Lei de Amdahl:** usada para encontrar a máxima melhora esperada para um sistema em geral quando apenas uma única parte do mesmo é melhorada.

$$T_{melhorado} = T_{naoafetado} + \frac{T_{afetado}}{fator\ de\ melhoria}$$

- Corolário: faça o caso comum ficar mais rápido.

- *Falácia:* Computadores com baixa utilização usam menos potência.
- i7 power benchmark:
  - 100% de carga: 258W
  - 50%: 170W (66%)
  - 10%: 121W (47%)
- Google data center
  - Trabalha normalmente com 10% - 50% de carga.
  - Com 100% em menos de 1% do tempo.
- Ideia futura: Projetar processadores com carga-potência proporcionais.

- *Armadilha:* Usar parte da equação de desempenho como métrica de desempenho.
- MIPS: Millions of Instructions Per Second (Milhões de Instruções por Segundo).
  - Não leva em consideração: Diferença entre ISA, diferença entre instruções.

$$\begin{aligned} \text{MIPS} &= \frac{\text{Contagem de instruções}}{\text{Tempo de execução} \times 10^6} = \\ &= \frac{\text{Contagem de instruções}}{\frac{\text{Contagem de instruções} \times \text{CPI}}{\text{Frequência clock}} \times 10^6} = \\ &= \frac{\text{Frequência clock}}{\text{CPI} \times 10^6} \end{aligned}$$

- CPI varia entre programas em uma dada CPU

- Custo/desempenho está melhorando
  - Por causa do desenvolvimento da tecnologia.
- Camadas hierárquicas de abstração
  - Em hardware e em software.
- Arquitetura do conjunto de instruções (ISA)
  - A interface hardware/software.
- Tempo de execução: a melhor medida de desempenho.
- Potência é fator limitante (superaquecimento)
  - Usar paralelismo para melhorar o desempenho.

Instruções: A linguagem de Máquina