

Arquitetura e Organização de Computadores

Cap 1. Abstrações e Tecnologias Computacionais

Parte 1 - Desempenho

Prof. Dr. João Fabrício Filho

Universidade Tecnológica Federal do Paraná
Campus Campo Mourão
2023

O que é abstração?

Abstrair ajuda a resolver
problemas?

O que isso tem a ver com
arquitetura de computadores?

— — —

Problema

e.g, Procurar o registro de um aluno no banco de dados da UTFPR

Algoritmo

Busca binária

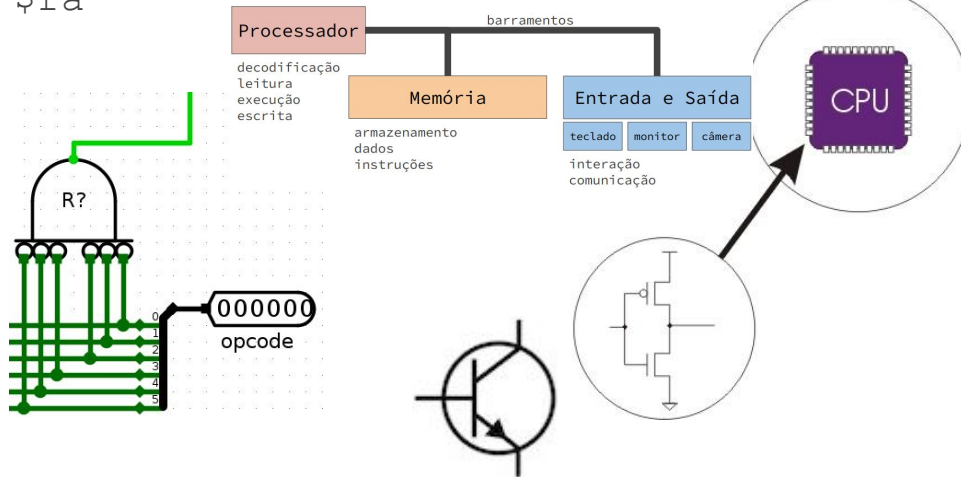
Programa

```
while (i<m) {
    if (A[i] == A[k])
        return i;
```

ISA

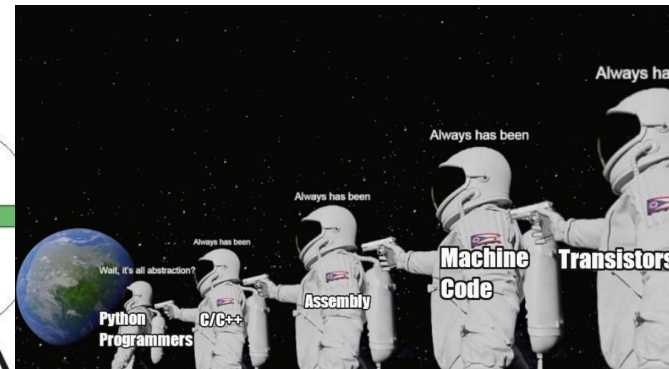
```
bne $s0, $s1, ELSE
add $v0, $t0, $0
jr $ra
```

Microarquitetura



Circuitos

Dispositivos



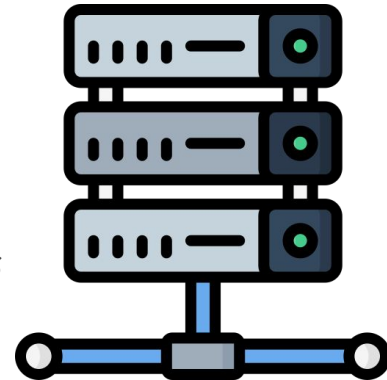
A (r)evolução dos computadores

- Novas aplicações
 - Smartphones
 - Computadores em automóveis
 - Motores de busca
 - Projeto de genoma humano

“Se o setor de transportes, por exemplo, tivesse tido o mesmo desenvolvimento da indústria da computação, hoje nós poderíamos viajar de Nova York até Londres em aproximadamente um segundo por apenas alguns centavos.” (Patterson & Henessy, 2017)

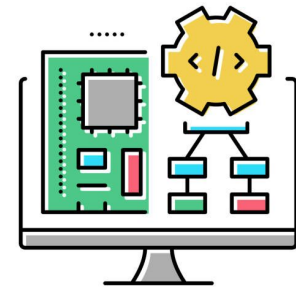
Classes de computadores

- Computadores pessoais
 - Propósito geral, variedade de softwares
 - Sujeito à variações de custo e desempenho
- Servidores
 - Baseados em rede
 - Alta capacidade, desempenho e confiabilidade
 - Variam de servidores pequenos a grandes *datacenters*

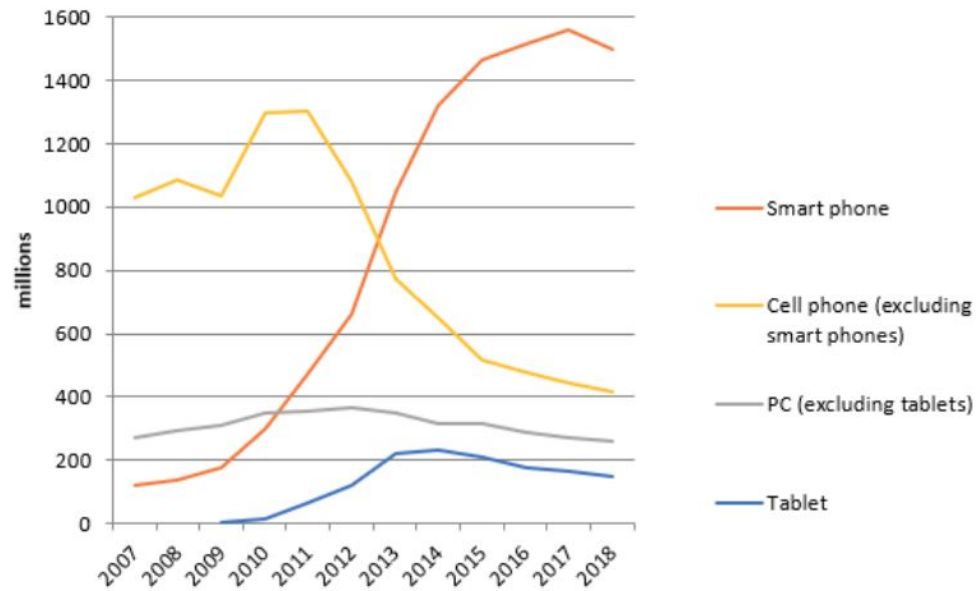


Classes de computadores

- Supercomputadores
 - Tipo de servidor
 - Cálculos científicos e de engenharia de alto nível
 - Alta capacidade
 - Pequena fração do mercado
- Computadores embarcados (embutidos?)
 - Escondidos como componentes de sistema
 - Rigorosamente limitados em desempenho, custo e energia



A era pós-PC



Onde a mudança possibilita maior ganho de desempenho?

Algoritmo

Define as operações para gerar as saídas

Programa

Estruturas de dados, compiladores, número de instruções

ISA

Comportamento das instruções, movimentação de dados

Processador

Determina o quão rapidamente as instruções serão executadas

Circuitos

Aspectos físicos da microarquitetura, parâmetros e construção de dispositivos

Níveis de código

- Linguagem de alto nível
 - Nível de abstração próximo ao problema
 - Provém produtividade e portabilidade
- Linguagem de montagem (*assembly*)
 - Representação textual de instruções
- Representação de *hardware*
 - Dígitos binários (*bits*)
 - Dados e instruções codificados

High-level
language
program
(in C)

```
swap(int v[], int k)
{int temp;
  temp = v[k];
  v[k] = v[k+1];
  v[k+1] = temp;
}
```

Compiler

Assembly
language
program
(for MIPS)

```
swap:
  muli $2, $5, 4
  add  $2, $4, $2
  lw   $15, 0($2)
  lw   $16, 4($2)
  sw   $16, 0($2)
  sw   $15, 4($2)
  jr   $31
```

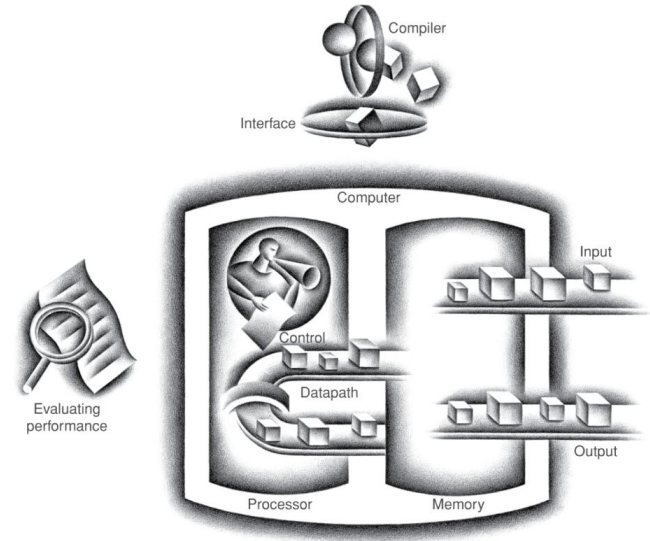
Assembler

Binary machine
language
program
(for MIPS)

```
000000001010000100000000000011000
0000000000001100000001100000100001
100011000110001000000000000000000
100011001111001000000000000000100
101011001111001000000000000000000
101011000110001000000000000000100
00000011111000000000000000001000
```

Componentes de um computador

- Os componentes são os mesmos para todas as classes de computadores
 - Desktop, servidores, embarcados
- E/S inclui:
 - Interface com usuário
 - Teclado, monitor, telas touchscreen, mouse
 - Dispositivos de armazenamento
 - Disco rígido, flash drives, SSD
 - Adaptadores de rede
 - Wifi, ethernet

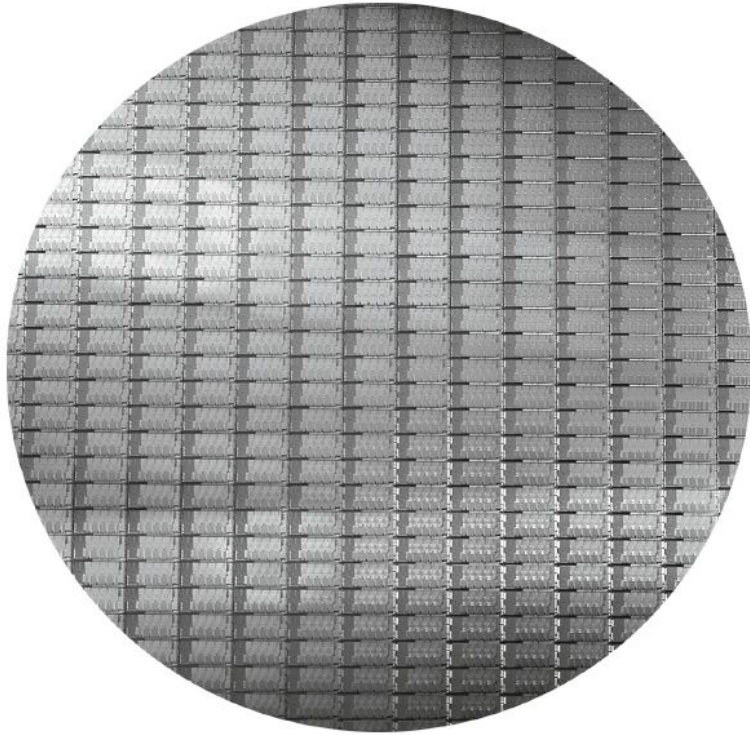


Tecnologia de semicondutores

- Silício: semicondutor
- Adição de materiais para transformar suas propriedades
 - Condutores

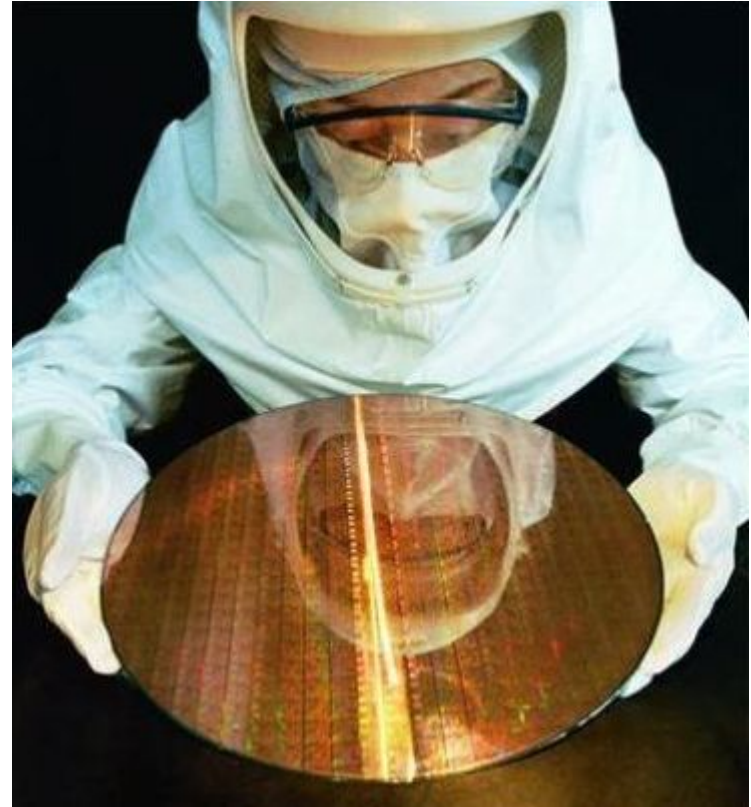


Wafer de fabricação de um Intel Core i7



30mm wafer, 280 chips, litografia
de 32nm

cada chip possui 20.7 x 10.5mm



Processo de fabricação de um processador



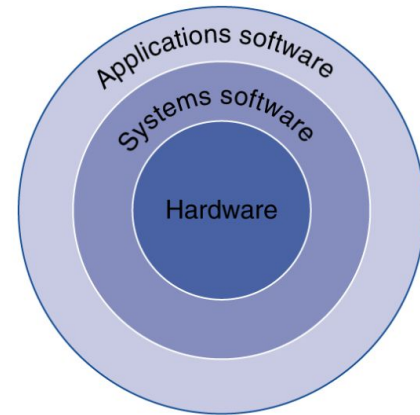
From Sand To Silicon: The Making of a Chip | Intel. https://www.youtube.com/watch?v=05paWn7bFg4&ab_channel=Intel

Custo de um circuito integrado

- Não-linear em relação à área e taxa de defeitos
- Custo do wafer é fixo
- Taxa de defeitos é determinada pelo processo de fabricação
- Área do die é determinada pela arquitetura e design do circuito

Por trás dos programas

- Software da aplicação
 - Escrito em linguagem de alto nível (HLL)
- Software de sistemas
 - Compilador: traduz código HLL em linguagem de máquina
 - Sistema operacional
 - Manipula E/S
 - Gerencia memória e armazenamento
 - Tarefas de escalonamento e compartilhamento de recursos
- Hardware
 - Processador
 - Memória
 - Controladores de E/S



- A cada 2 anos, o número de componentes em um chip dobrará e o custo se manterá



Dentro do processador

Intel Pentium 4

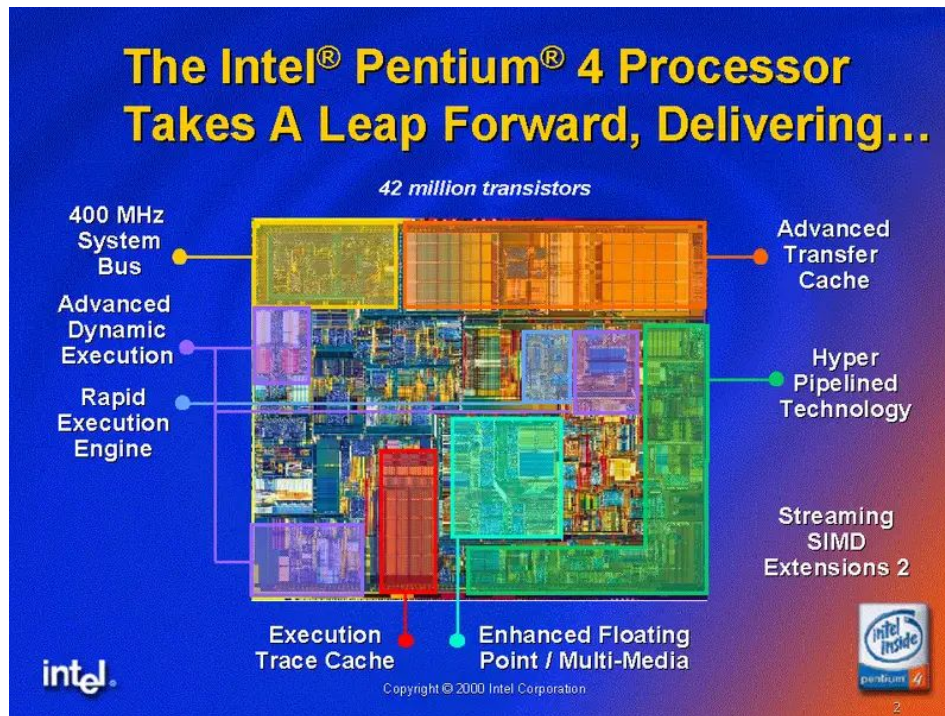
Frequência-base: 3,2 GHz

Cores: 1

Potência: 82W

Cache: 512KB

55×10^6 transistores



https://en.wikichip.org/wiki/File:p4_die_slide.png 17

Dentro do processador

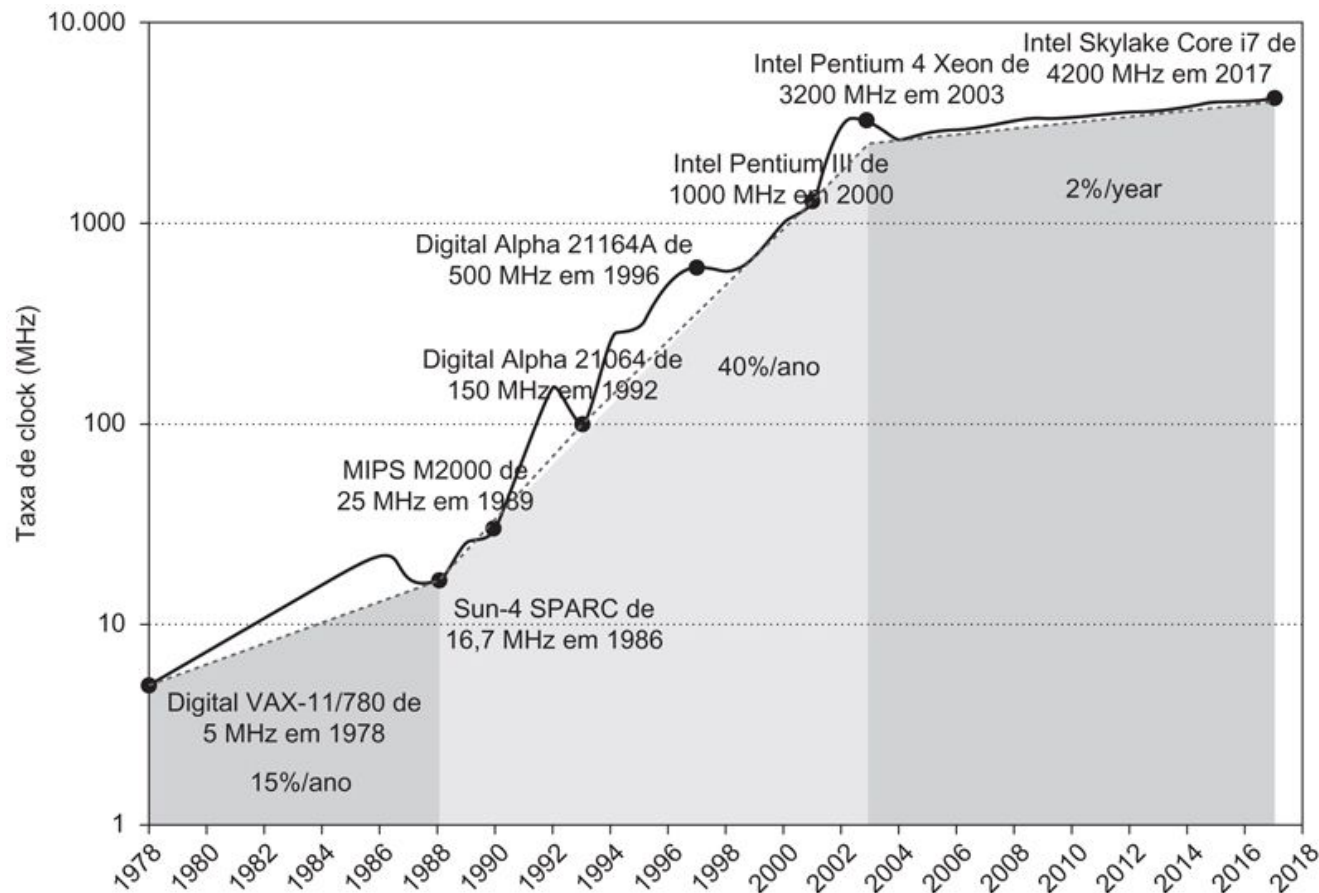
Intel Core i9-13900K

Frequência-base: 3,2 GHz
Cores: 8+16P/32E
Potência: 125W
Cache: 32MB L2 + 36MB L3
4,2 x 10⁹ transistores

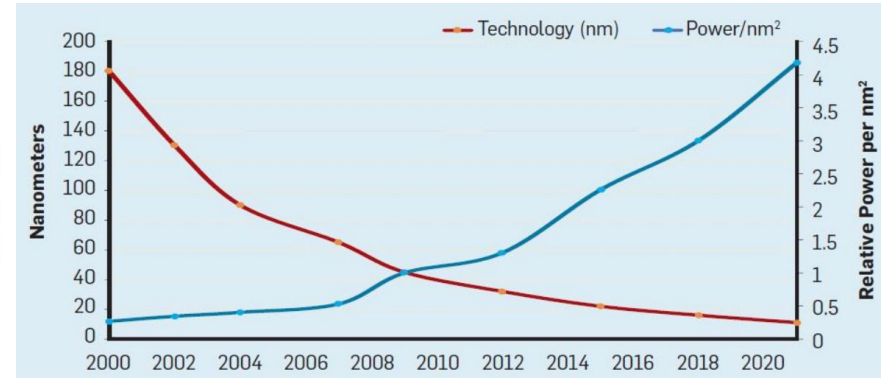
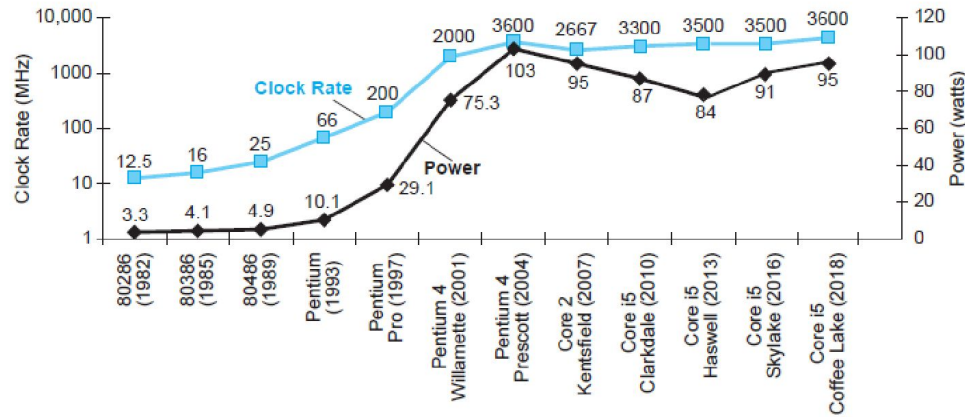


<https://www.techpowerup.com/review/intel-core-i9-13900k/2.html>

Crescimento na taxa de clock



Tendências em potência

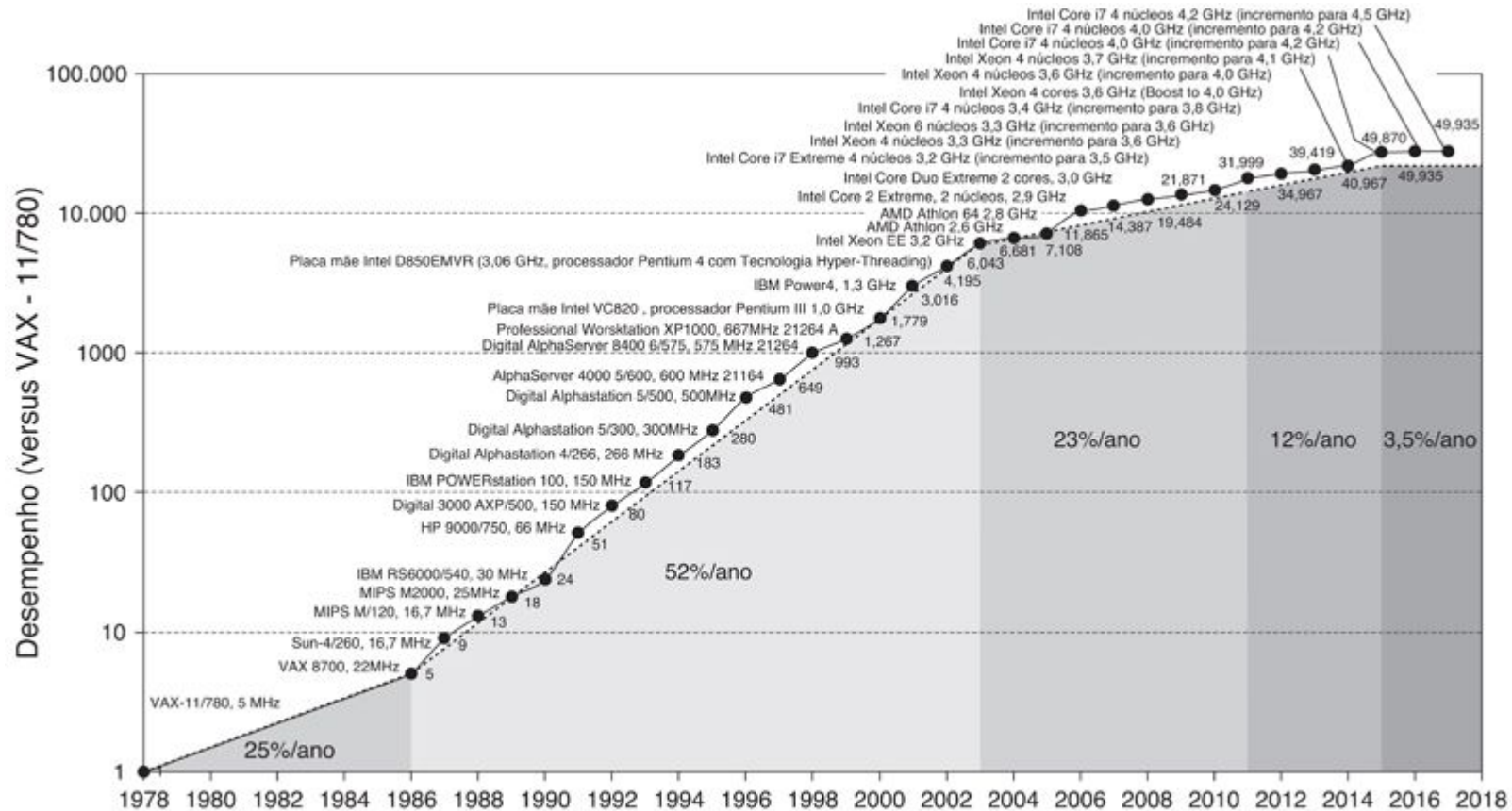


- Na tecnologia CMOS
 - $\text{Potência} = \text{carga capacitiva} \times (\text{tensão elétrica})^2 \times \text{frequência}$

Multiprocessadores

- Mais de um processador em um chip
- Exige programação paralela explícita
 - Comparação com paralelismo a nível de instrução
 - O hardware executa múltiplas instruções ao mesmo tempo
 - Escondido do programador
 - Dificuldades
 - Programar com objetivo o desempenho
 - Balanceamento de carga
 - Otimização de comunicação e sincronização

Desempenho de um processador único



O que é desempenho?

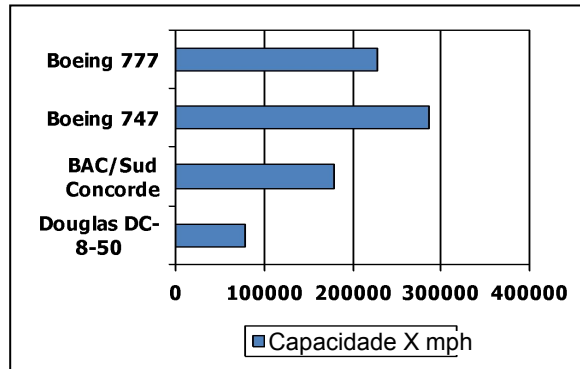
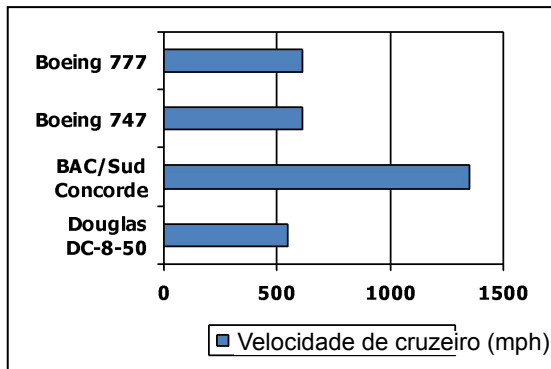
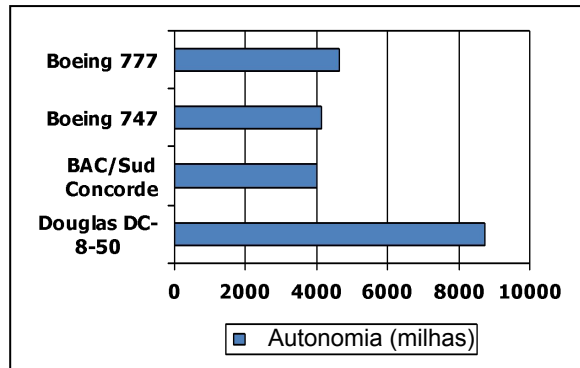
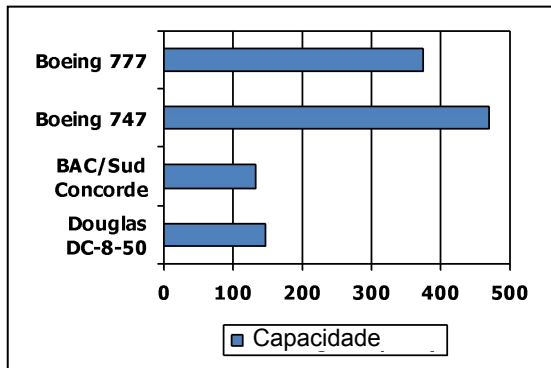
Existem medidas de desempenho?

Qual a diferença entre
desempenho e eficiência?

— — —

Definindo desempenho

Qual avião possui melhor desempenho?



Desempenho em um sistema computacional

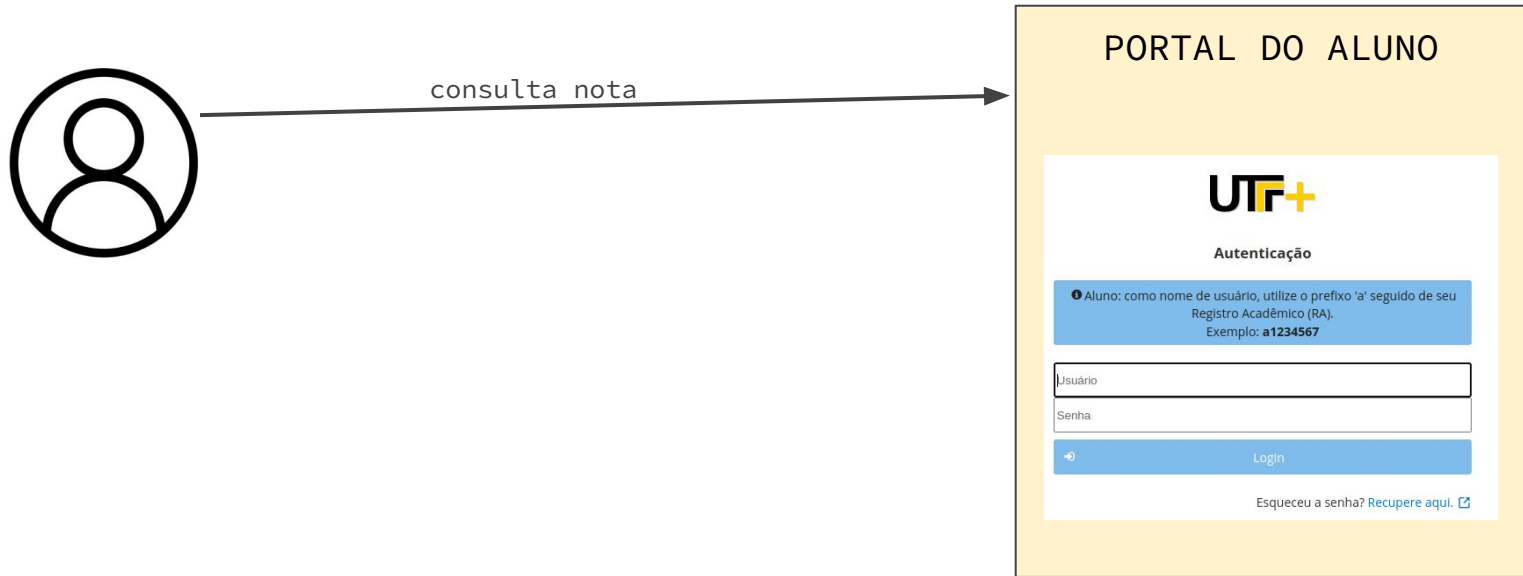
Tempo de Resposta

- Tempo entre início e término de uma tarefa.

Throughput

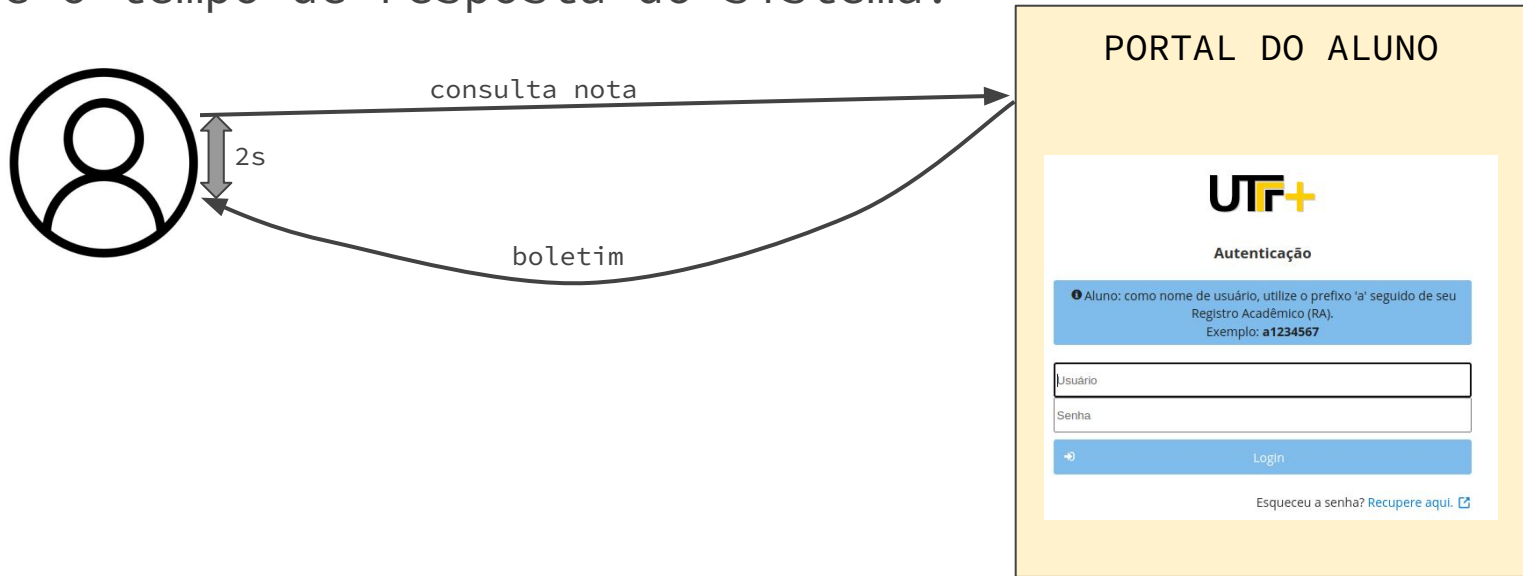
- Quantidade de trabalho realizada por unidade de tempo.

Tempo de resposta | Exemplo

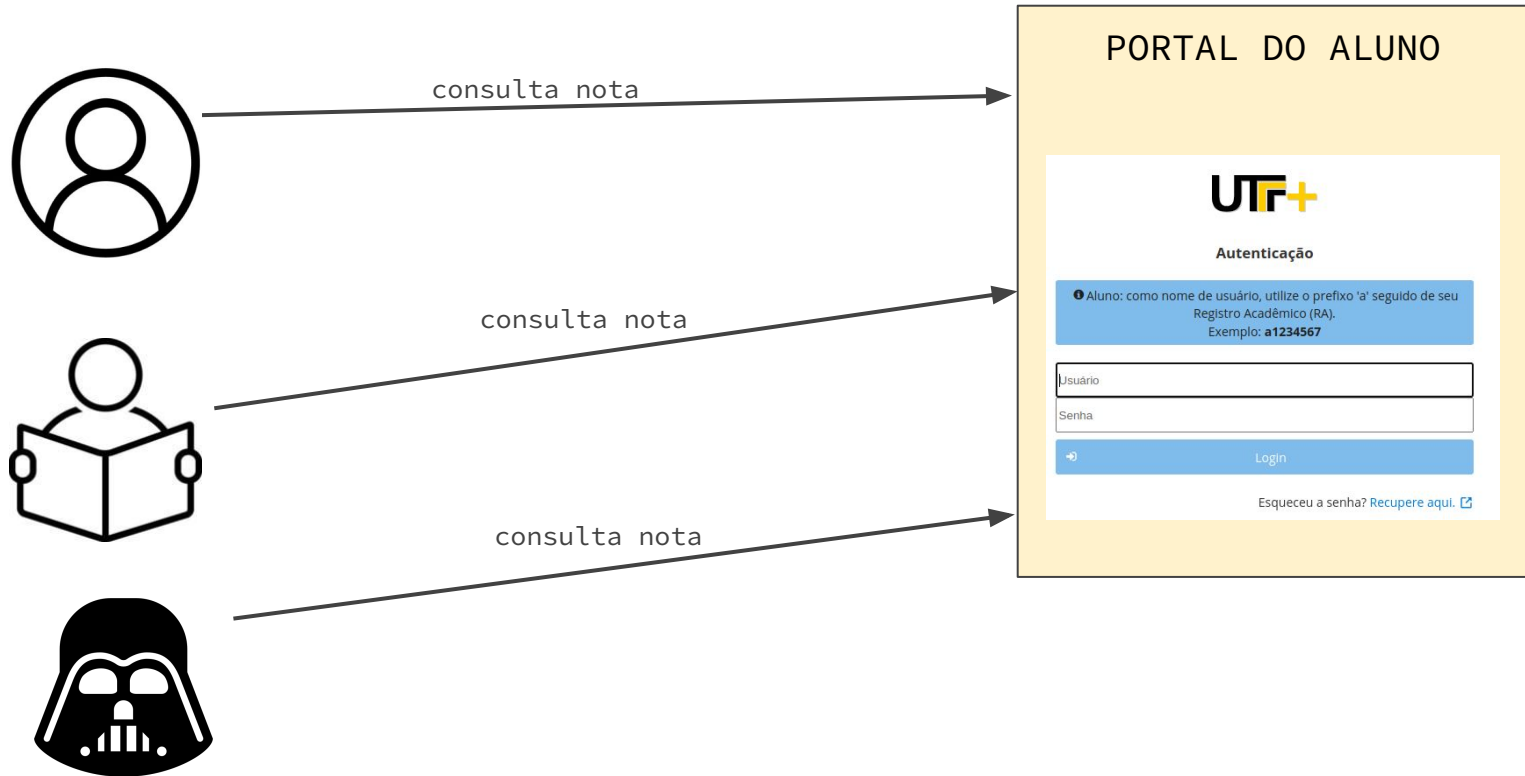


Tempo de resposta | Exemplo

Qual é o tempo de resposta do sistema?

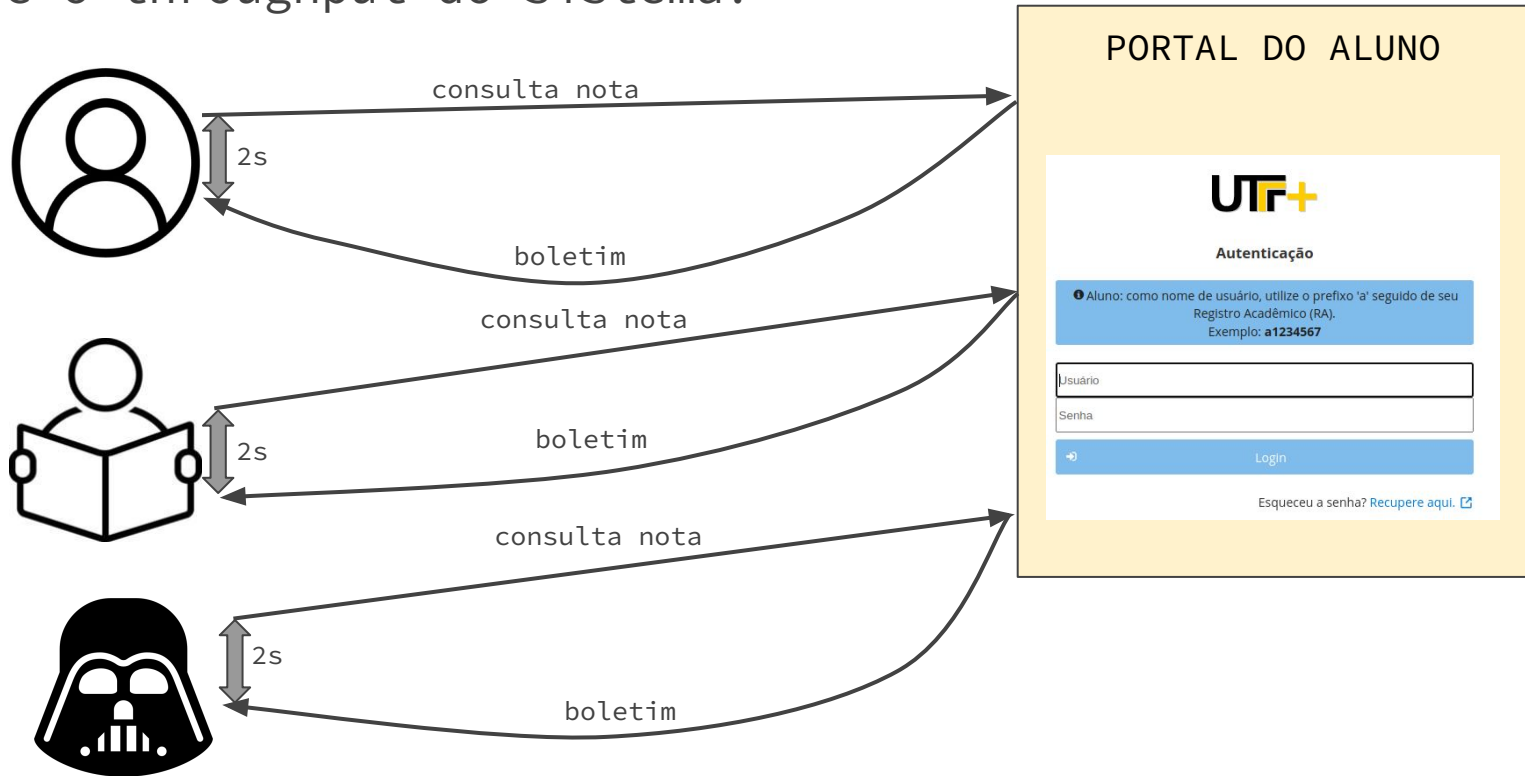


Throughput | Exemplo



Throughput | Exemplo

Qual é o throughput do sistema?



Tempo de resposta vs throughput

Como o tempo de resposta e o throughput são afetados:

1. Substituindo o processador por uma versão mais rápida?
2. Incluindo mais processadores?

Medindo o desempenho

- Tempo decorrido
 - Tempo de resposta total, incluindo todos os aspectos
 - Processamento, E/S, atraso do SO, tempo ocioso
 - Determina o desempenho do sistema
- Tempo de CPU
 - Tempo despendido processando um dado trabalho
 - Desconta tempo de E/S e compartilhamento de outros trabalhos
 - Compreende tempo de CPU de usuário e de sistema
 - Diferentes programas são afetados diferentemente pelo desempenho da CPU e do sistema

Métrica de desempenho

- Desempenho do sistema x:

$$Desempenho_x = \frac{1}{Tempo\ de\ CPU_x}$$

1. Qual o desempenho do portal do aluno com tempo de resposta de 2 segundos?
2. Se houver um novo servidor com tempo de resposta de 1 segundo, qual o novo desempenho?

Desempenho relativo

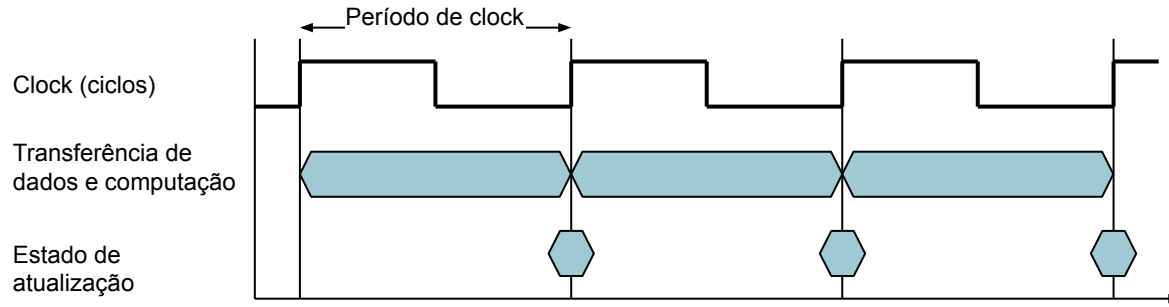
- “X é n vezes mais rápido que Y”

$$\frac{Desempenho_x}{Desempenho_y} = \frac{Tempo\ de\ CPU_y}{Tempo\ de\ CPU_x}$$

- Quantas vezes mais rápido o novo servidor do exemplo anterior é do que o antigo?

Clock de CPU

- Operação de hardware digital regulada por uma taxa constante de clock



- Período de clock
 - Duração de um ciclo de clock
- Frequência de clock
 - Ciclos por período (geralmente em ciclos/segundo)

Tempo de CPU

$$\begin{aligned}\text{Tempo de CPU} &= \text{Ciclos de Clock CPU} \times \text{Período clock} \\ &= \frac{\text{Ciclo de clock CPU}}{\text{Frequência do Clock}}\end{aligned}$$

- Desempenho é melhorado:
 - Reduzindo o número de ciclos de clock
 - Aumentando a frequência de clock
 - Projeto do hardware:
 - Frequência de clock X Quantidade de ciclos

Tempo de CPU | Exemplo

A UTFPR quer um novo projeto de servidor para o portal do aluno. O tempo de CPU de 1s do novo servidor não é o suficiente. A nova demanda é de um tempo de CPU de 600ms.

Vamos escolher um novo processador para esse servidor?

- Devemos manter a arquitetura (mesma ISA)
- Frequência do processador antigo = 2 GHz
- Qual deve ser a frequência para atingir o tempo de 600ms?

Tempo de CPU | Exemplo - Resolução

Máquina atual

- Tempo CPU = 1 segundo
- Frequência = 2 GHz = 2×10^9 ciclos/segundo
- Ciclos:
 - Tempo CPU = Ciclos / Frequência ;
 - $1 = \text{Ciclos} / 2 \times 10^9$;
 - Ciclos = 2×10^9

Máquina necessária

- Tempo CPU = 0,6 segundos
- Ciclos = 2×10^9
- Frequência:
 - $0,6 = 2 \times 10^9 / F$
 - $F = 2 \times 10^9 / 0,6 = \sim 3,3 \times 10^9 = 3,3 \text{ GHz}$

Tempo de CPU | Exemplo 2

- Computador A:
 - Frequência 2GHz
 - Tempo de CPU 10s
- Computador B a ser projetado:
 - Objetivo: 6s de tempo de CPU
 - Ciclos de clock B: 1,2 x ciclos de clock A
- Qual a frequência de clock que o computador B deve ter?

Tempo de CPU | Exemplo 2 - resolução

Tempo de CPU A: $\text{Tempo} = \text{ciclos} / \text{Frequência}$

- $10 = \text{ciclos} / 2 \times 10^9 \mid 10 \times 2 \times 10^9 = \text{ciclos}$

- $\text{ciclos} = 20 \times 10^9$

Tempo de CPU B: $\text{tempo} = \text{ciclos} / \text{frequência}$

- $6 = 1,2 \times 20 \times 10^9 / \text{frequência} \mid \text{frequência} = 1,2 \times 20 \times 10^9 / 6$

- $\text{frequência} = 4 \times 10^9 = \mathbf{4GHz}$

R: Para executar o programa em 6s, o computador B deverá ter 4 GHz de frequência de clock.

Número de instruções e CPI

$$\begin{aligned}\text{Ciclos de Clock} &= \text{Número de Instruções(IC)} \times \\ &\quad \times \text{Ciclos por Instrução(CPI)} \\ \text{Tempo de CPU} &= \text{IC} \times \text{CPI} \times \\ &\quad \times \text{Período do Clock} \\ &= \frac{\text{IC} \times \text{CPI}}{\text{Frequência do Clock}}\end{aligned}$$

- Número de instruções de um programa
 - Determinado pelo programa, ISA e compilador
- Média de ciclos por instrução
 - Determinada pela CPU
 - Se instruções diferentes possuem CPI diferente
 - CPI médio é afetado pela mistura de instruções

Exemplo de CPI

- Computador A
 - Período de clock: 250ps, CPI = 2,0
- Computador B
 - Período de clock: 500ps, CPI = 1,2
- Mesmo conjunto de instruções
- Qual é o mais rápido e quão mais rápido?

Exemplo de CPI

- Tempo de CPU = Número de Instruções X CPI X Período de clock
- Suponha um programa que tenha I instruções
- Computador A
 - $T = I \times 2,0 \times 250\text{ps} = I \times 500\text{ps}$
- Computador B
 - $T = I \times 1,2 \times 500\text{ps} = I \times 600\text{ps}$
- Desempenho relativo
 - $\text{Tempo de resposta B} / \text{Tempo de Resposta A} = I \times 600\text{ps} / I \times 500\text{ps} = 1,2$

CPI em mais detalhes

- Se classes de instruções diferentes possuem diferentes números de ciclos:

$$\text{CiclosClock} = \sum_{i=1}^n (\text{CPI}_i \times \text{IC}_i)$$

- Média ponderada CPI

$$\text{CPI} = \frac{\text{CiclosClock}}{\text{IC}} = \sum_{i=1}^n \left(\text{CPI}_i \times \frac{\text{IC}_i}{\text{IC}} \right)$$

Exemplo de CPI para diferentes classes

- Sequência de código compiladas usando instruções das classes A, B e C

Classe	A	B	C
CPI para a classe	1	2	3
IC na sequência 1	2	1	2
IC na sequência 2	4	1	1

- Qual sequência de códigos executa mais instruções?
- Qual é mais rápida?
- Qual o CPI médio para cada sequência?

Exemplo de CPI para diferentes classes

- Qual sequência de códigos executa mais instruções?
 - A sequência 1 executa $2 + 1 + 2 = 5$ instruções.
 - A sequência 2 executa $4 + 1 + 1 = 6$ instruções.
 - Portanto, a sequência 1 executa menos instruções.
- Qual é mais rápida?
 - Ciclos de clock = $\text{soma}_{i=1}^n (\text{CPI}_i \times \text{IC}_i)$
 - Ciclos 1 = $(2 \times 1) + (1 \times 2) + (2 \times 3) = 10$ ciclos
 - Ciclos 2 = $(4 \times 1) + (1 \times 2) + (1 \times 3) = 9$ ciclos
 - Assim, a sequência de código 2 é mais rápida, mesmo executando uma instrução a mais.
- Qual o CPI médio para cada sequência?
 - O CPI médio para cada sequência é calculado dividindo o número de ciclos de clock pela contagem de instruções:
 - $\text{CPI } 1 = 10 / 5 = 2,0$
 - $\text{CPI } 2 = 9 / 6 = 1,5$

Exemplo de CPI para diferentes classes (2)

- Sequência de código compiladas usando instruções das classes A, B e C

Classe	ULA	Controle	Memória
CPI para a classe	1	3	10
Programa 1	1000	100	100
Programa 2	2000	300	50

- Qual sequência de códigos executa mais instruções?
- Qual é mais rápida?
- Qual o CPI médio para cada sequência?

Exemplo de CPI para diferentes classes (2)

- Qual sequência de códigos executa mais instruções?
 - O programa 1 executa 1200 instruções.
 - O programa 2 executa 2350 instruções.
 - Portanto, o programa 1 executa menos instruções.
- Qual é mais rápida?
 - Ciclos 1 = $(1000 \times 1) + (100 \times 3) + (100 \times 10) = 2300$ ciclos
 - Ciclos 2 = $(2000 \times 1) + (300 \times 3) + (50 \times 10) = 3400$ ciclos
 - Assim, o programa 1 é mais rápido além de executar menos instruções.
- Qual o CPI médio para cada sequência?
 - $\text{CPI 1} = 2300 / 1200 = 1,91$
 - $\text{CPI 2} = 3400 / 2350 = 1,44$
 - Assim, o CPI médio do programa 2 é menor, o que quer dizer que cada instrução sua demanda menor tempo de CPU.

Resumo de desempenho

$$\text{Tempo CPU} = \frac{\text{Instruções}}{\text{Programa}} \times \frac{\text{CiclosClock}}{\text{Instrução}} \times \frac{\text{Segundos}}{\text{Ciclos de Clock}}$$

- Desempenho depende:
 - Algoritmo: afeta IC, possivelmente CPI
 - Linguagem de programação: afeta IC, CPI
 - Compilador: afeta IC, CPI
 - ISA: afeta IC, CPI, clock

Benchmark de CPU SPEC

- Programas usados para medir desempenho
 - Cargas de trabalho típicas
- Standard Performance Evaluation Corp (SPEC)
- SPEC CPU
 - Tempo decorrido para executar o conjunto de programas
 - Tempo E/S desprezível, foco no desempenho da CPU
 - Normalizado em relação a uma máquina referência
 - Sumarizado como a média geométrica das taxas de desempenho

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

Armadilha: lei de Amdahl

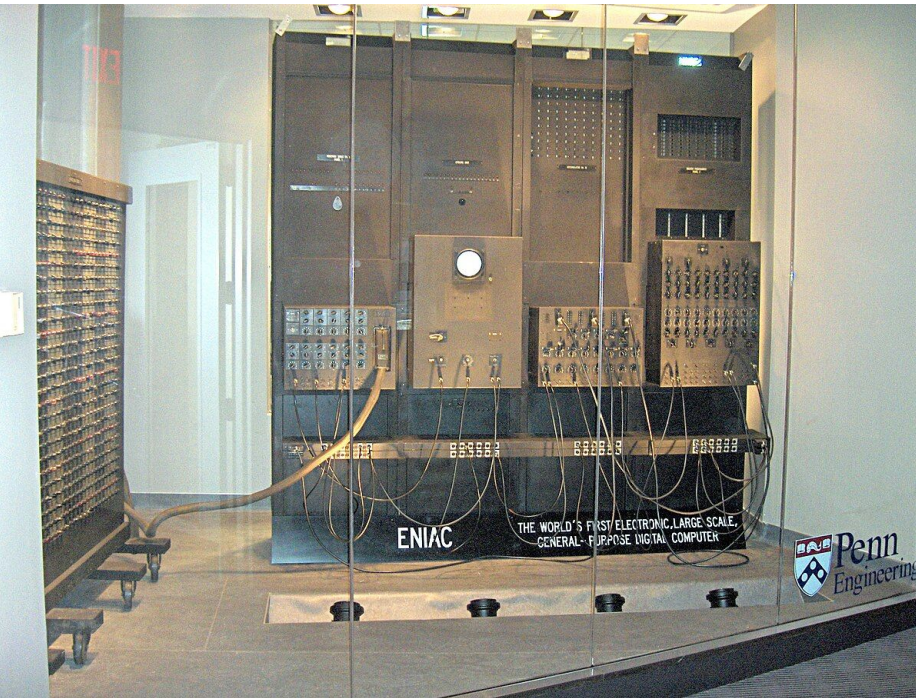
- Melhorar um aspecto do computador e esperar uma melhoria proporcional em todo o desempenho

$$T_{\text{improved}} = \frac{T_{\text{affected}}}{\text{improvement factor}} + T_{\text{unaffected}}$$

- Exemplo: multiplicação que corresponde 80s/100s
 - Quanta melhoria é possível no desempenho total se tivermos 5x na multiplicação?
- Corolário: faça o caso comum ser rápido

Considerações finais

- Custo e desempenhos estão melhorando
 - Devido ao desenvolvimento da tecnologia
- Camadas hierárquicas de abstração
 - Tanto no hardware quanto no software
- Conjunto de instruções da arquitetura (ISA)
 - A interface entre hardware e software
- Tempo de execução: a melhor medida de desempenho
- Potência é um fator limitante
 - Use o paralelismo para melhorar o desempenho



“Enquanto o Eniac é equipado com 18.000 válvulas e pesa 30 toneladas, os computadores no futuro poderão ter 1.000 válvulas e talvez pesar apenas 1,5 tonelada.”

Revista Popular Mechanics, Março de 1949.

Exercícios sugeridos

Capítulo 1 – Abstrações e tecnologias computacionais (P&H)

Seção “Verifique você mesmo”: 1.1, 1.6, 1.10

Seção do final do capítulo: 1.1, 1.3, 1.5, 1.6, 1.7, 1.9, 1.11.(1,3-4,5-11), 1.12, 1.13, 1.14

Referências

Materiais disponibilizados pelos professores Lucas Wanner, Paulo Gonçalves, Ricardo Pannain e Rogério Ap. Gonçalves.

Patterson, David A. Hennessy, John L. Organização e Projeto de Computadores. Disponível em: Minha Biblioteca, (5a. edição). Grupo GEN, 2017.

Slides do livro PATTERSON, David A. e HENNESSY, John L. *Computer Organization and Design Risc-V Edition: The Hardware Software Interface*. Estados Unidos, Ed. Morgan Kauffman, 2020

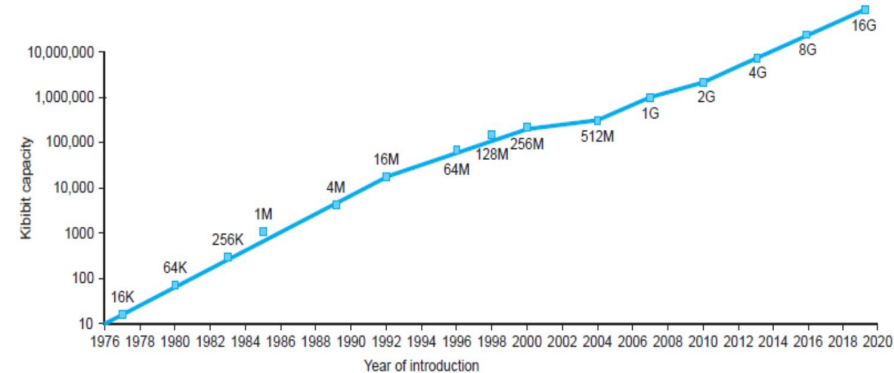
Abstrações

- Abstrações ajudam a lidar com complexidade
 - Esconde detalhes menores
- Conjunto de instruções da arquitetura (ISA)
 - Interface hardware/software
- Interface binária de aplicação
 - ISA e interface de software do sistema
- Implementação
 - Detalhes abaixo das abstrações

Tendências de tecnologia

- Tecnologia de eletrônicos continua a evoluir
 - Capacidade e desempenho aumentados
 - Custo reduzido

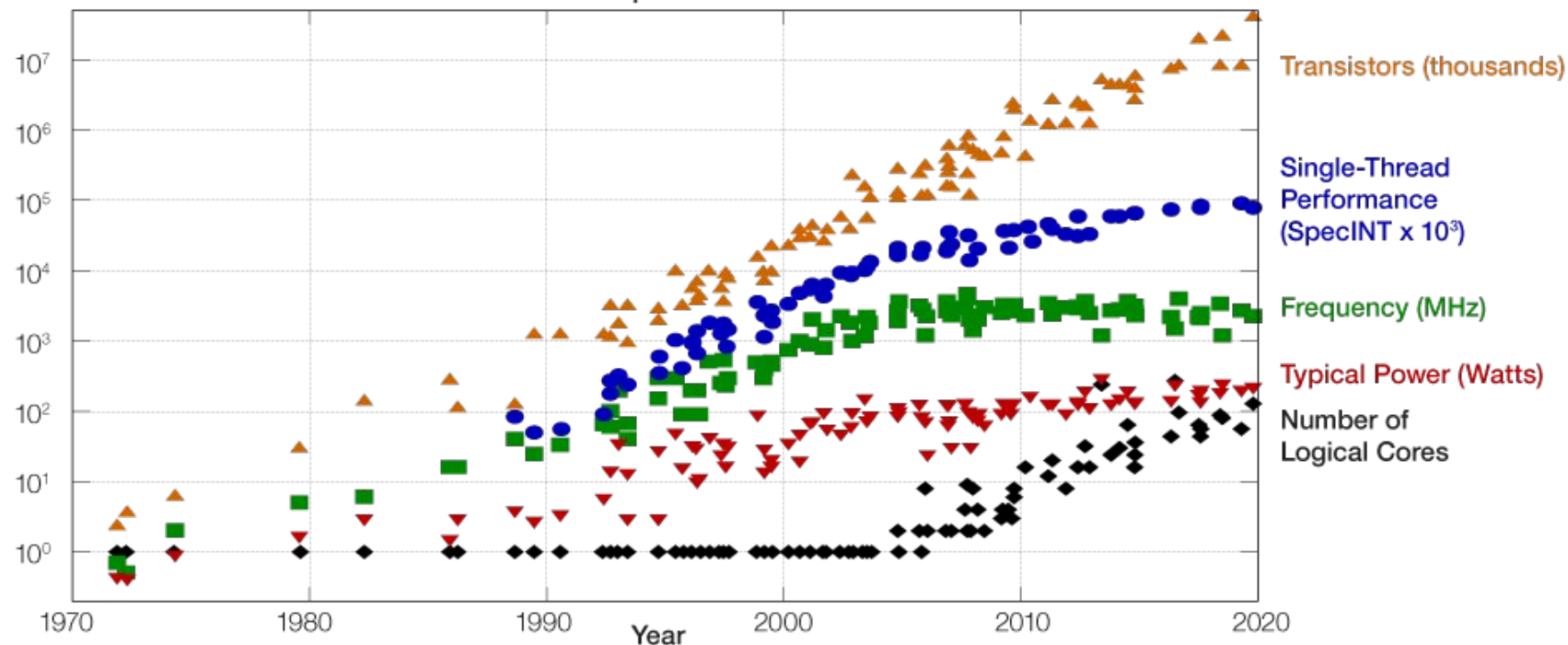
Capacidade da memória principal



Ano	Tecnologia	Relação desempenho/custo
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

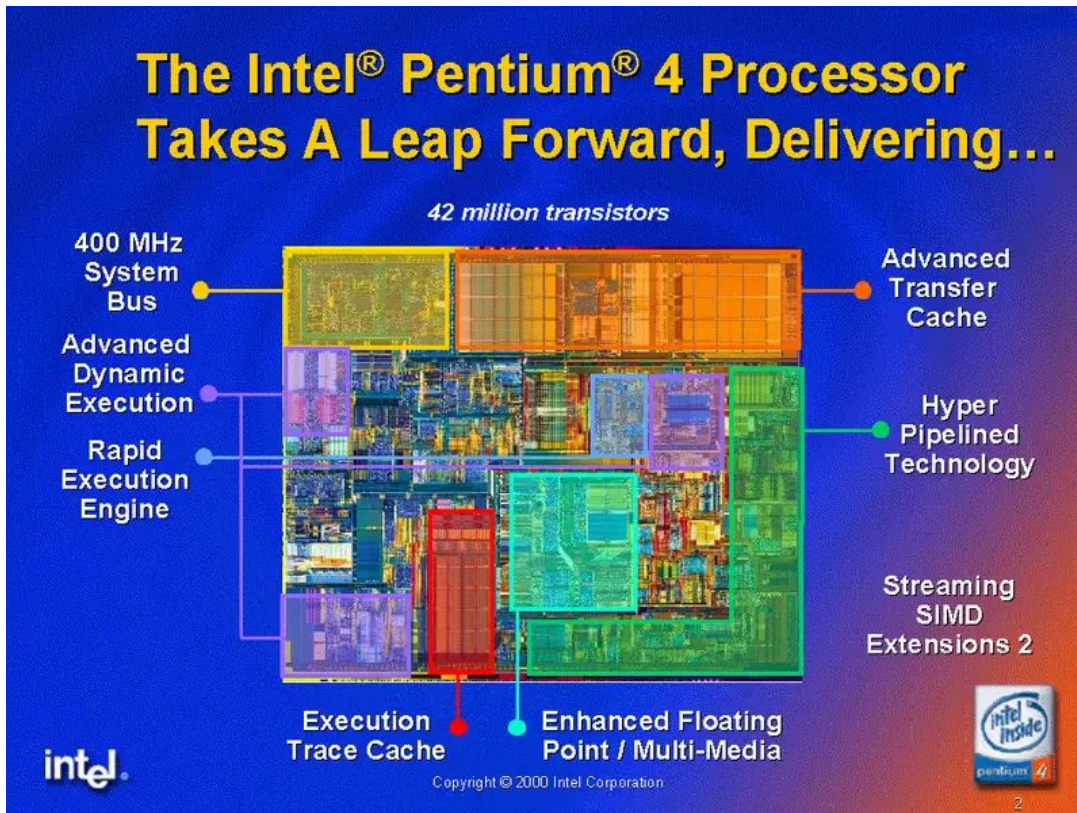
48 Years of Microprocessor Trend Data



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2019 by K. Rupp

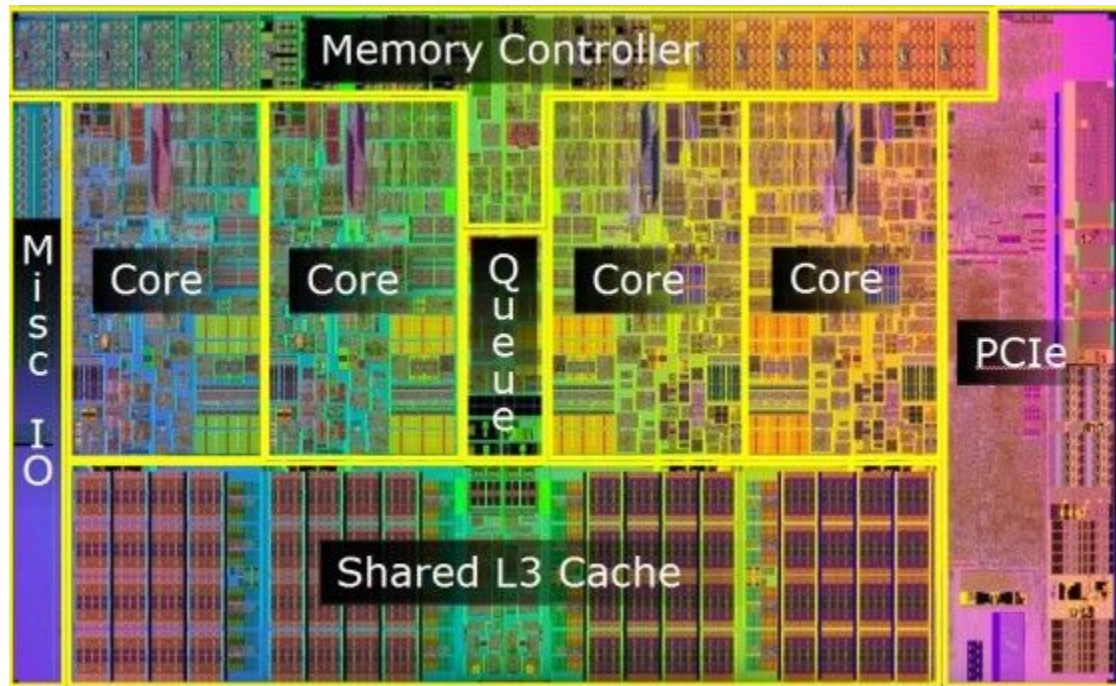
Dentro do processador

Intel Pentium 4



Dentro do processador

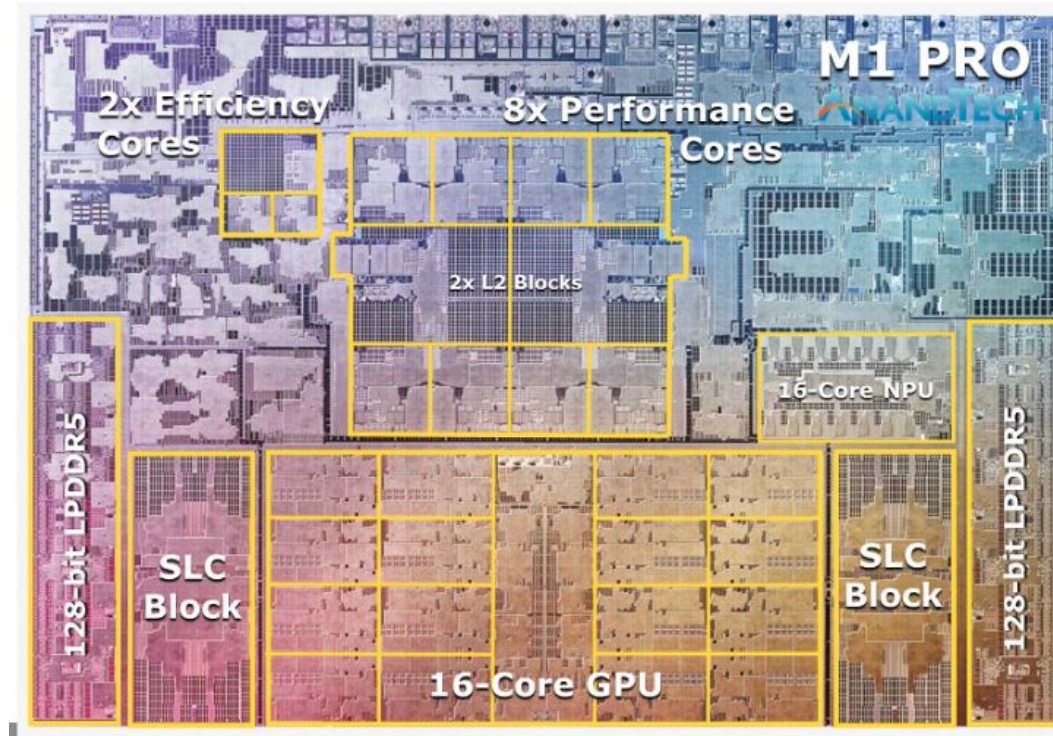
Intel core i7 Nehalem



<https://www.cs.uaf.edu/2009/fall/cs441/proj1/russell/index.html>

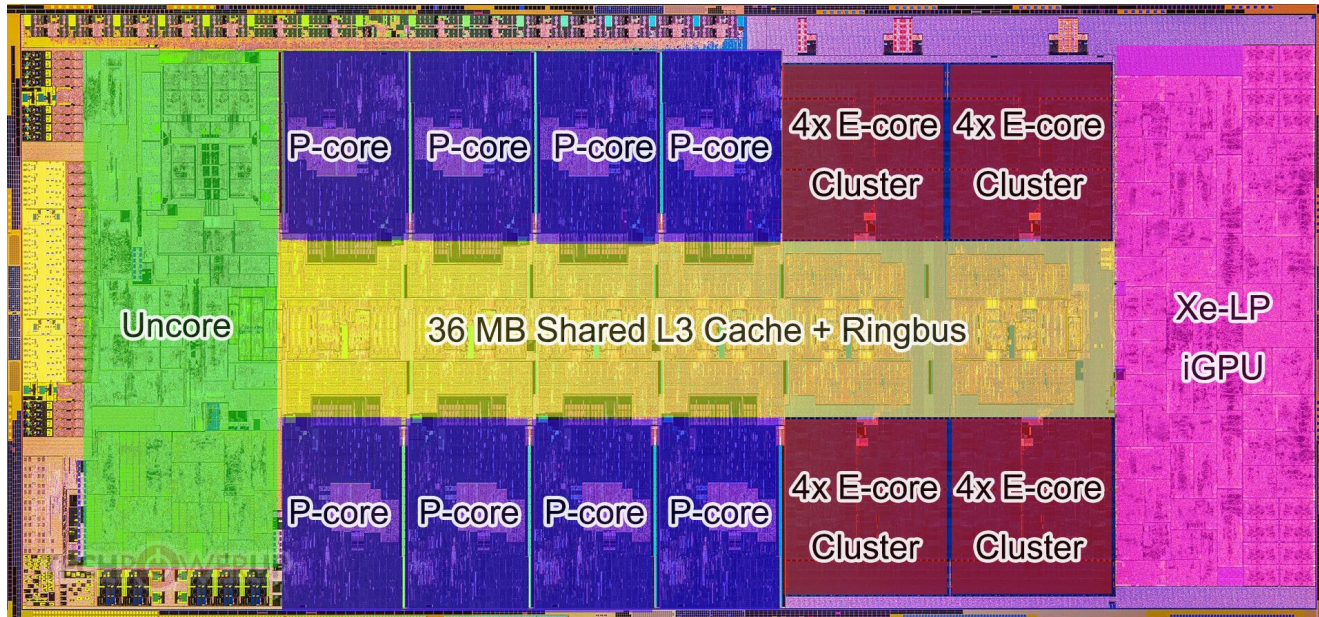
Dentro do processador

Apple M1 Pro SoC



Dentro do processador

Intel Core i9-13900K



<https://www.techpowerup.com/review/intel-core-i9-13900k/2.html>