

# LOM3233

# Microprocessadores

Turma 20222F1

Aula inicial

24/08/2022

# Objetivos

- Apresentar os principais microprocessadores atuais
- Comparar microprocessadores e microcontroladores
- Apresentar ferramentas de simulação, programação e prototipação virtual
- Internet das Coisas: IoT
- Inteligência Artificial: AI
- Desenvolver competências técnicas e socioemocionais por meio de projeto integrador de microprocessadores

# Programa da disciplina

- Introdução
- Microprocessadores e Microcontroladores
- Simulação/programação/prototipação virtual e física
- Internet das Coisas: IoT
- Inteligência Artificial: AI
- Projeto

# Plataformas de apoio a disciplina

- Google Workspace (Meet, Drive, Docs, Forms, Site, etc)
- MOODLE e-disciplinas
- Simuladores Tinkercad e SimulIDE
- Sites de projeto Instructables, *Maker*, Hackster IO e profissionais

# Critério de avaliação

O critério de avaliação a ser aplicado na disciplina levará em conta três critérios:

- 1) participação nas atividades de aula;
- 2) realização de atividades didáticas complementares fora do horário de aula (apresentações e seminários);
- 3) execução de projeto final de aplicação real de microprocessadores.

A nota final será composta pela média ponderada das atividades acima.

# Introdução aos Microprocessadores

# Introdução

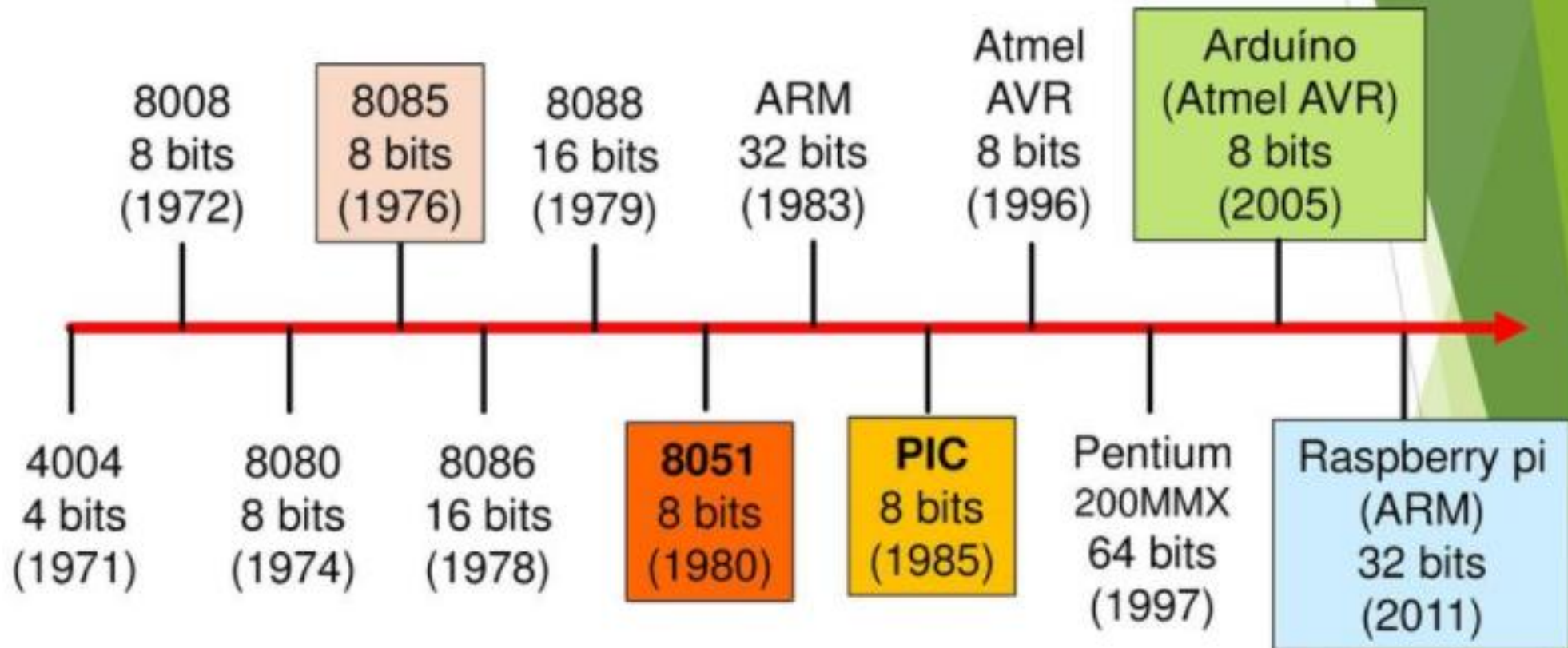


*O que eles têm em comum?*



# Introdução

## *Linha do Tempo*





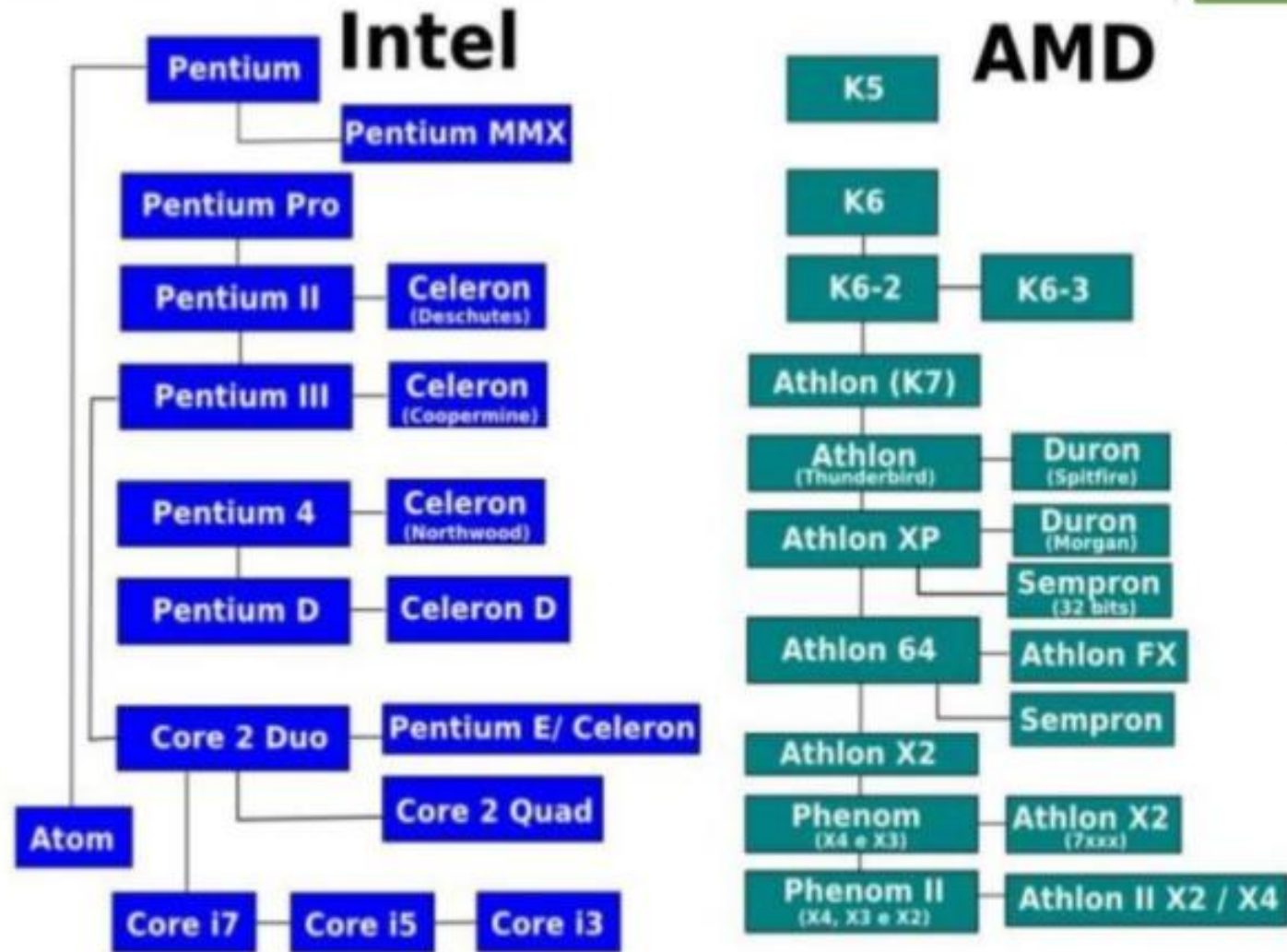
# Introdução

## Evolução dos Processadores

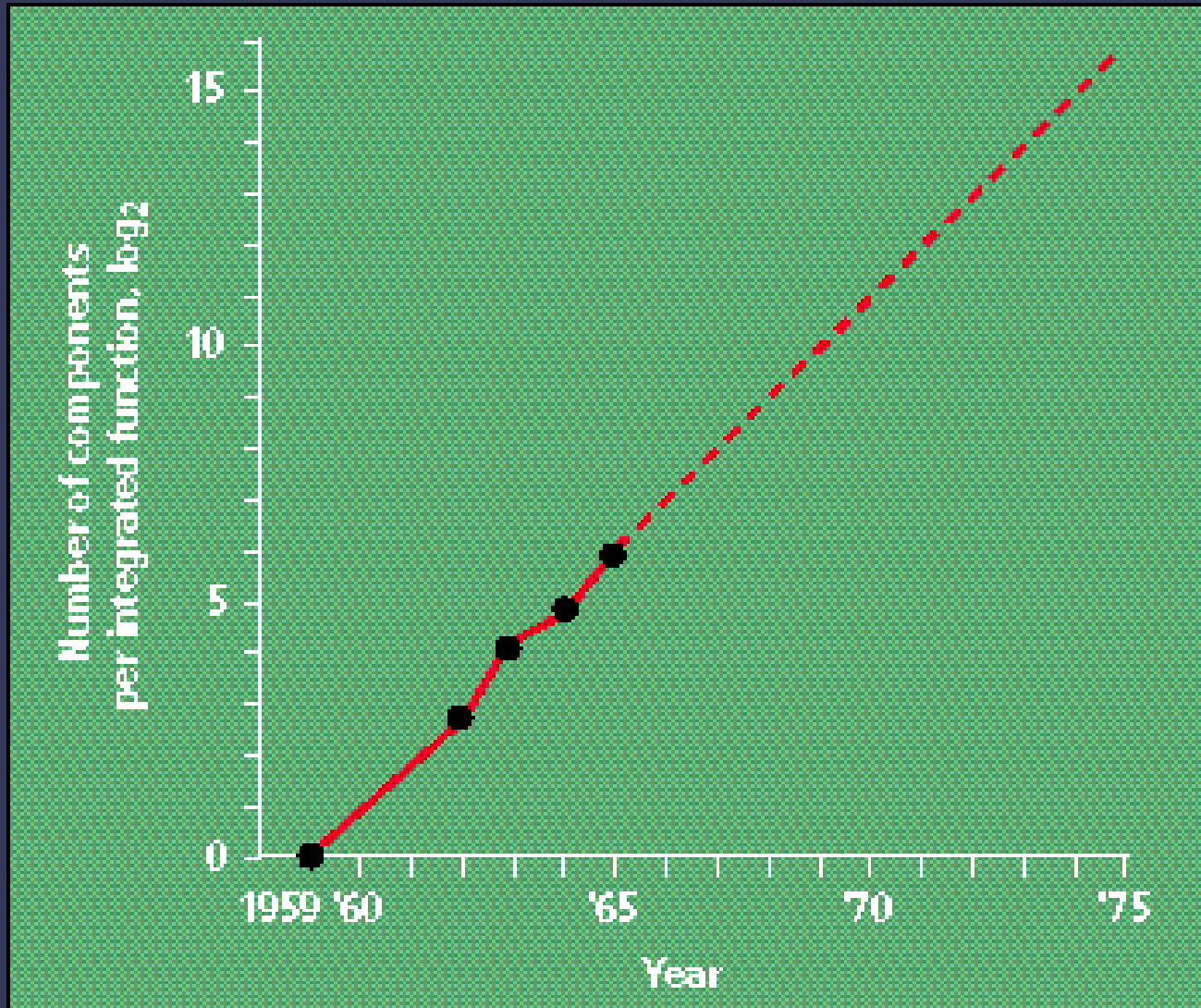


# Introdução

## Evolução dos Processadores



# Circuito integrado: Lei de Moore (1965)



Gordon Moore (Intel):

“O número de transistores num circuito integrado dobra a cada 18 meses”

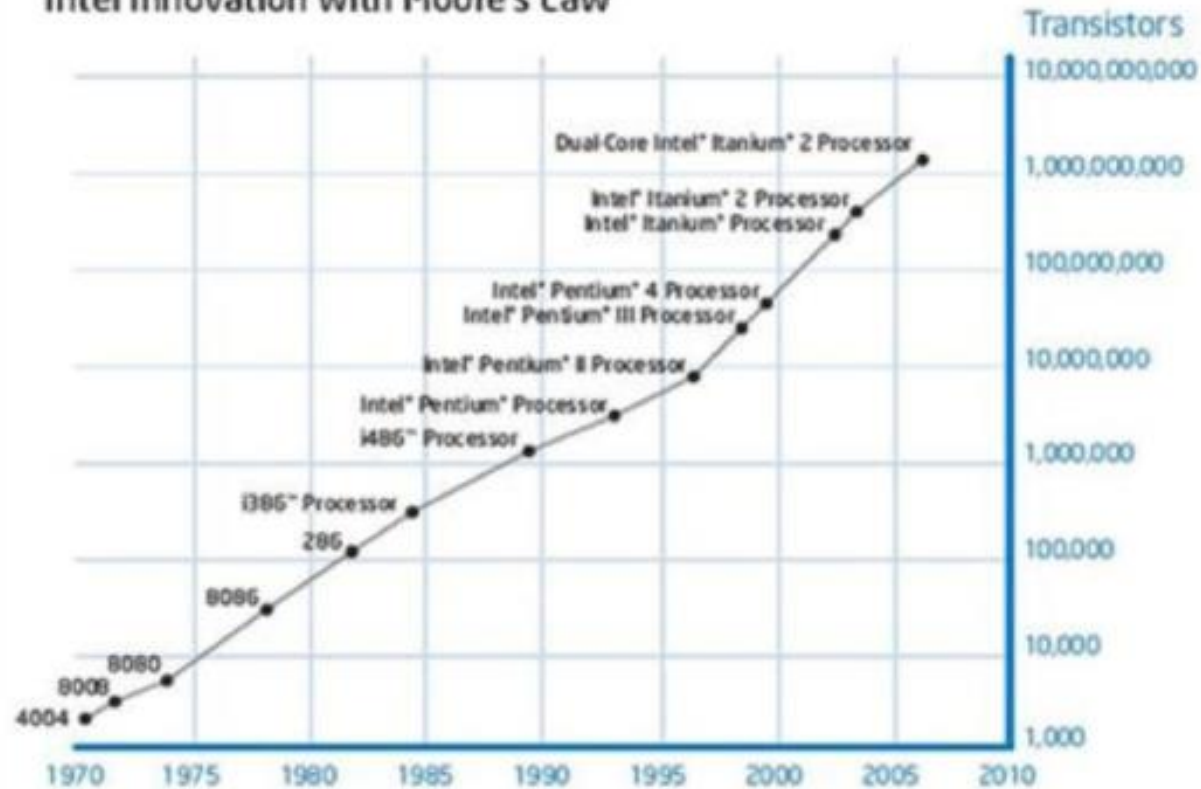
# Introdução: Lei de Moore

## Quantidade de Transistores

**8085** (1976) 3.600 transistores (5 MHz)  
370.000 instruções/segundo  
8 bits.

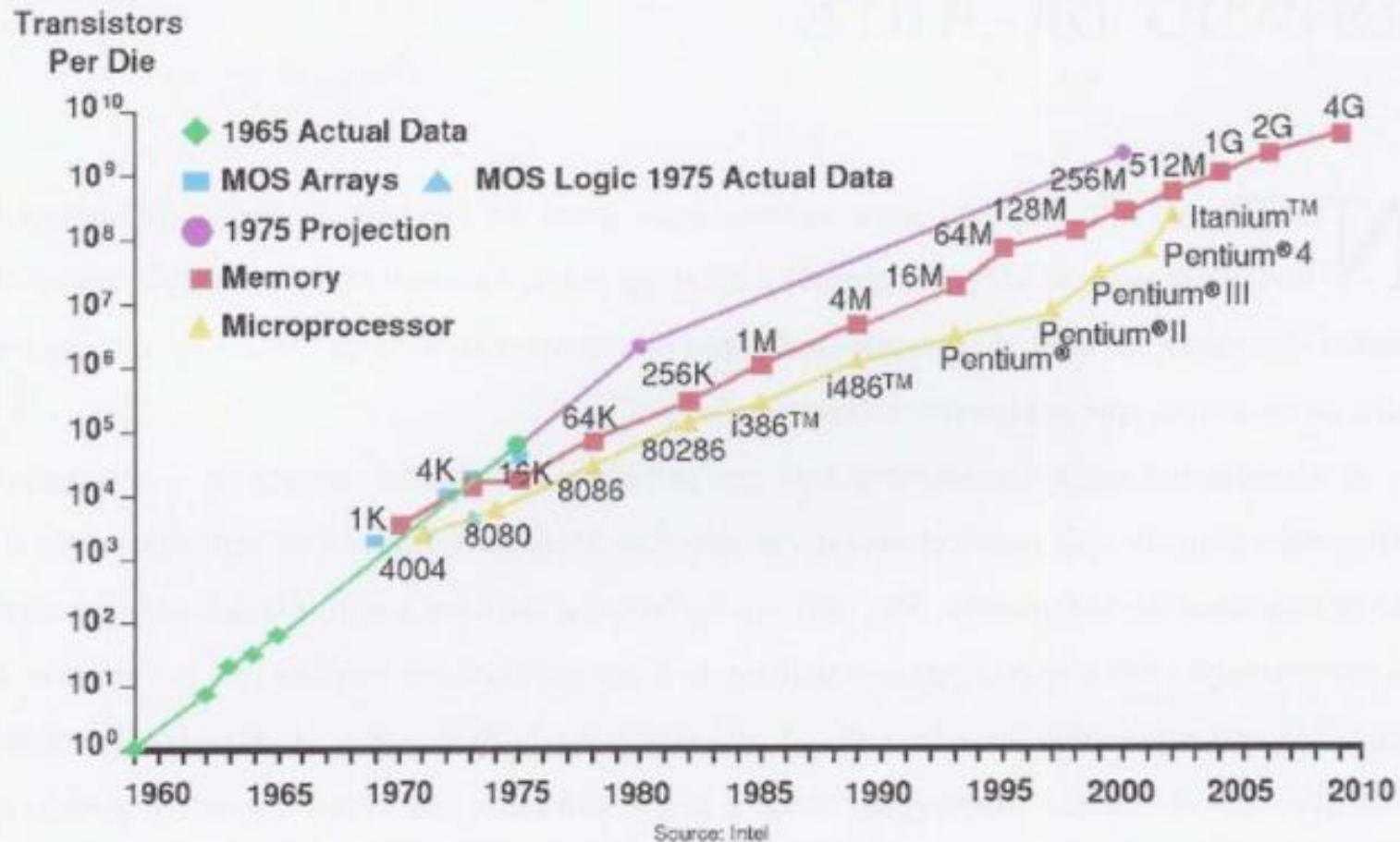
**Dual Core** (2007): 820 milhões de transistores  
53.000 milhões de instruções/segundo  
64 bits. Tecnologia: 45 nm. 3,33 GHz

Intel Innovation with Moore's Law

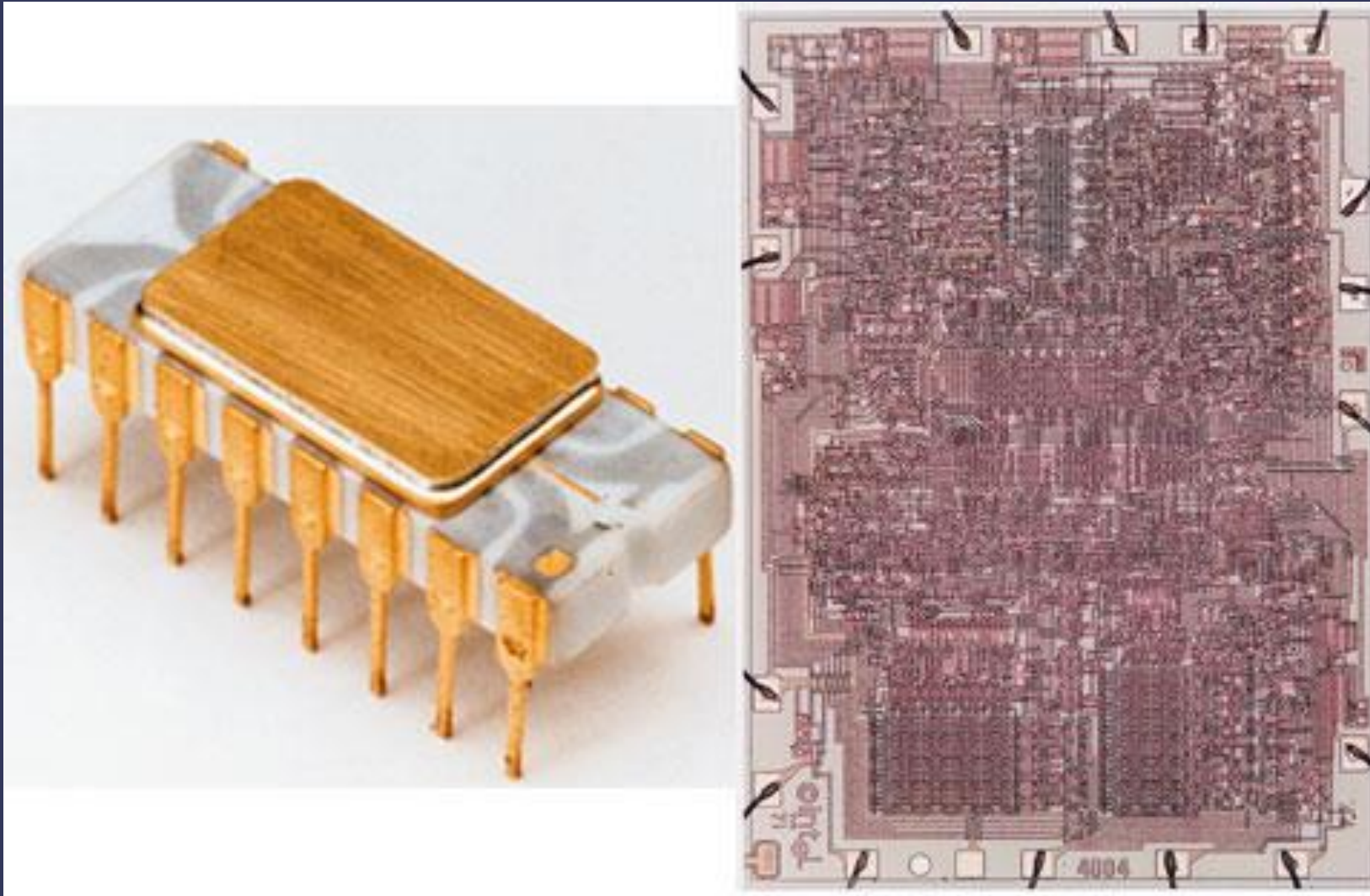


# Introdução: Lei de Moore

## Quantidade de Transistores e Memória

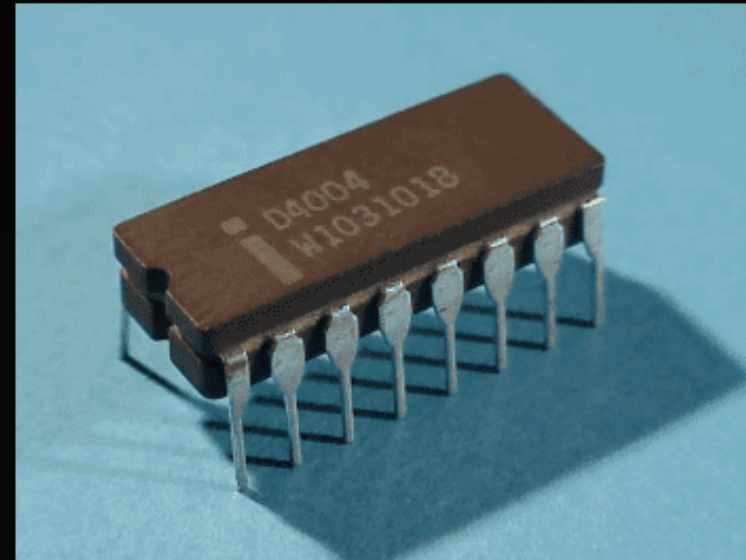


# Circuito integrado: Intel 4004 (1971)



# Circuito integrado: Intel 4004 (1971)

- World first “general purpose” micro-processor
- Lead designers
  - Ted Hoff, Federico Faggin, Stan Mazor, Masatoshi Shim
- Data
  - Word width: 4-bit
  - 2300 transistors
  - Clock: 108KHz/500/740
  - 46 instructions
  - Registers: 16 x 4-bit
  - Stack: 12 x 4-bit
  - Address space
    - 1Kb of program, 4Kb of data



# Calculadora eletrônica: Texas Instruments (1967)

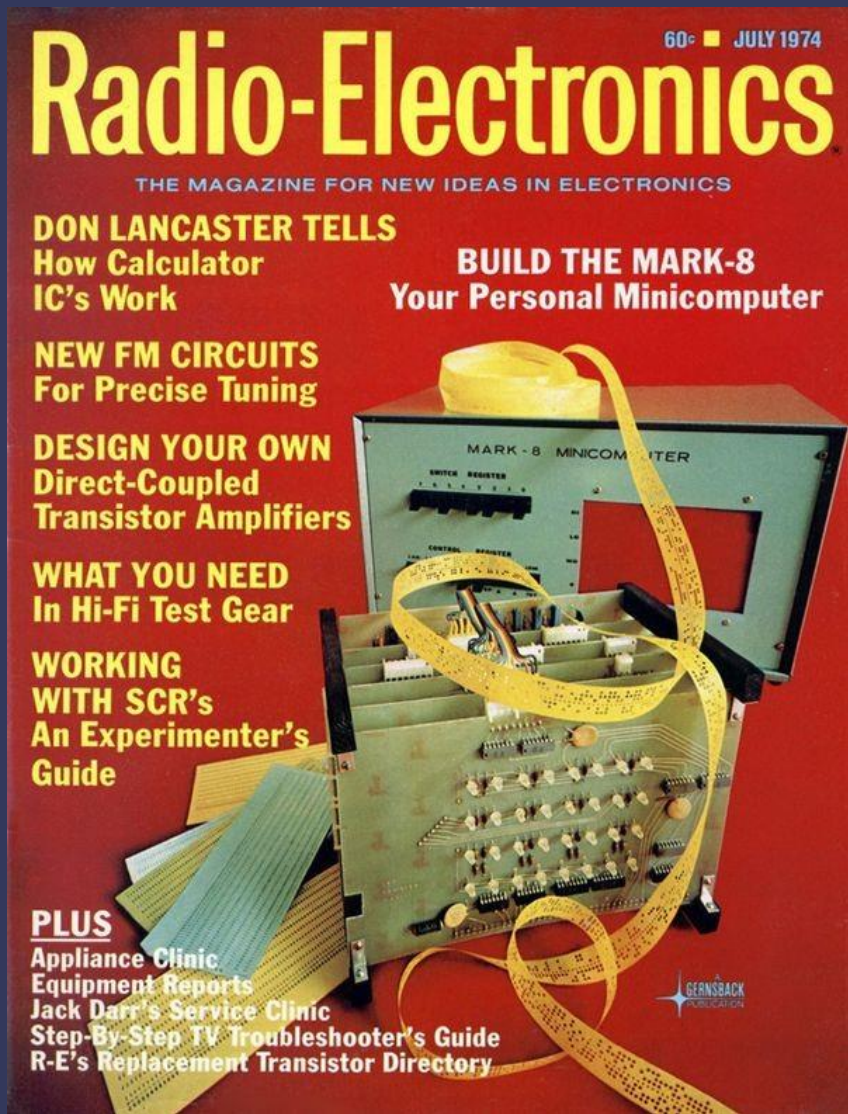


 TEXAS  
INSTRUMENTS

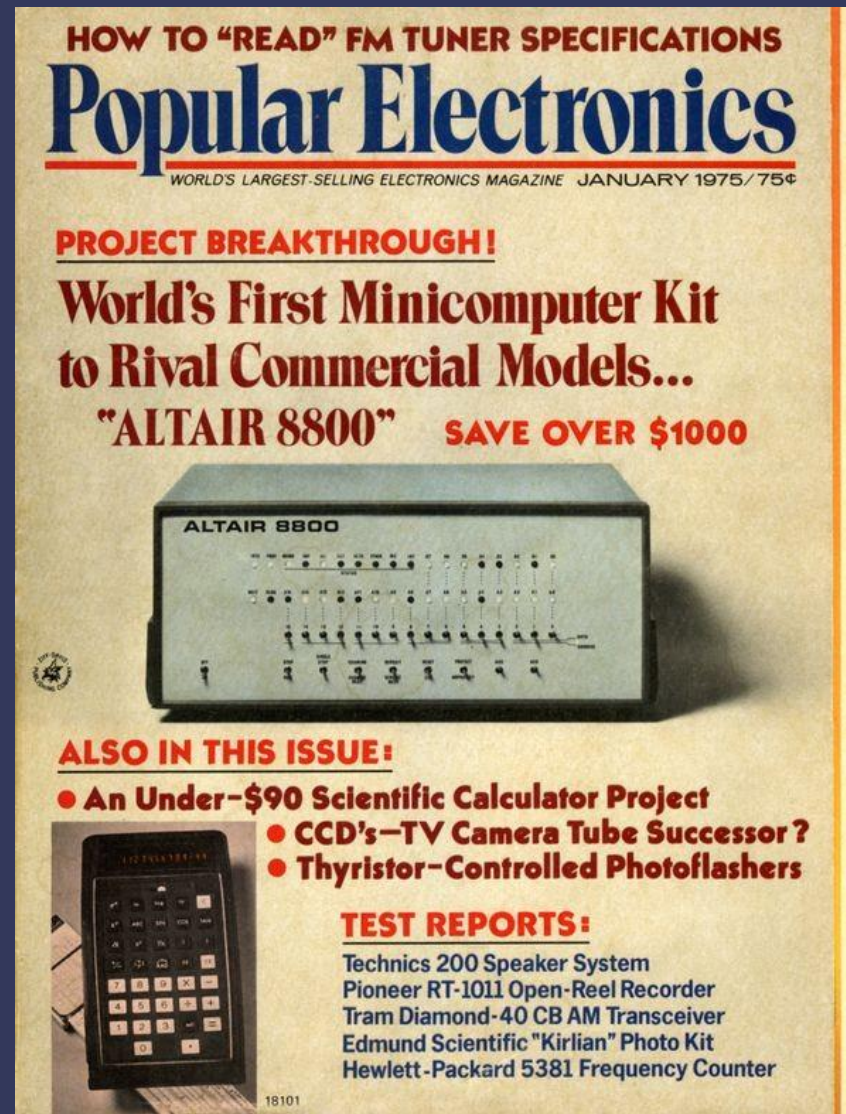
0975-60



# Altair 8800: primeiro microcomputador

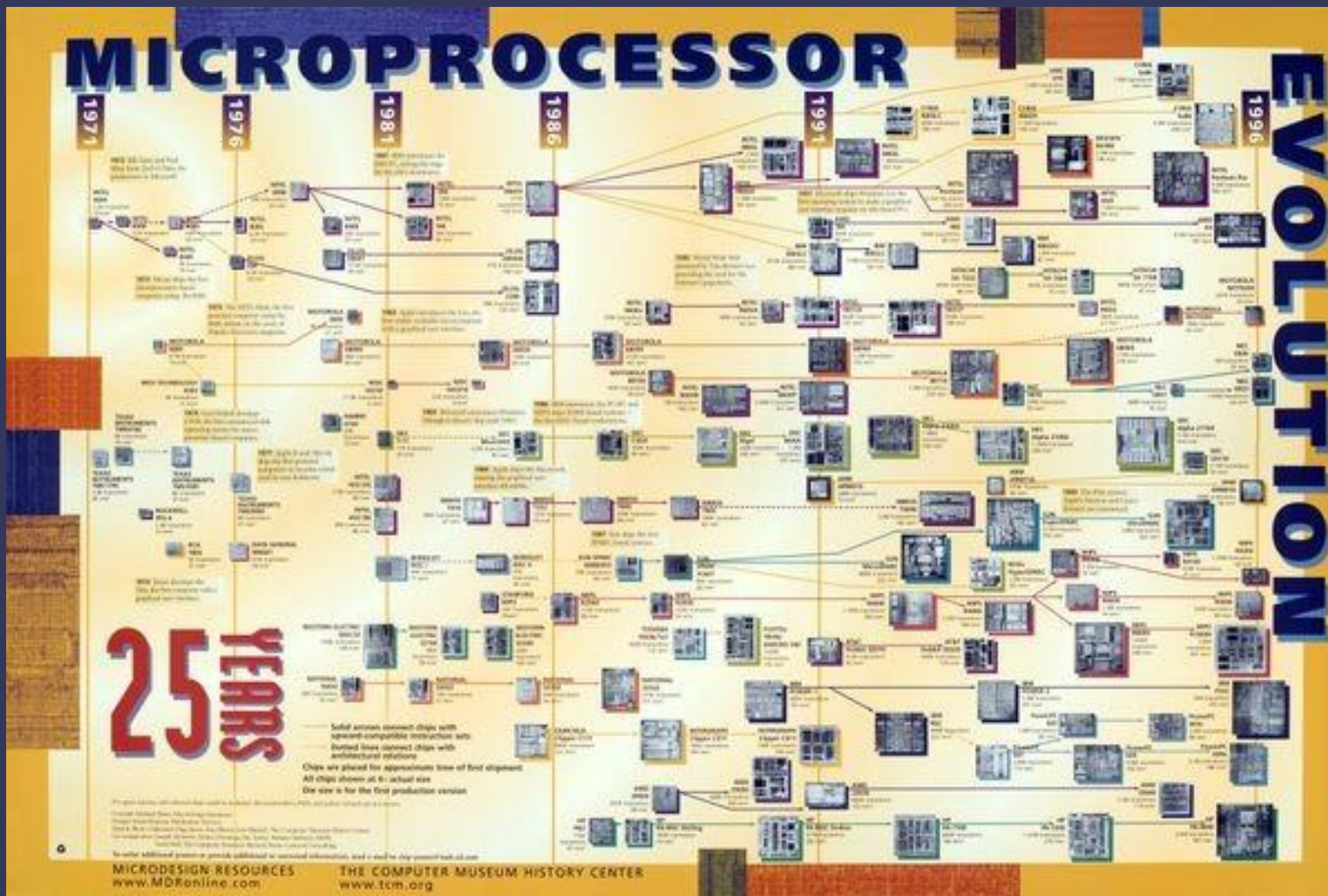


Julho 1974



Janeiro 1975

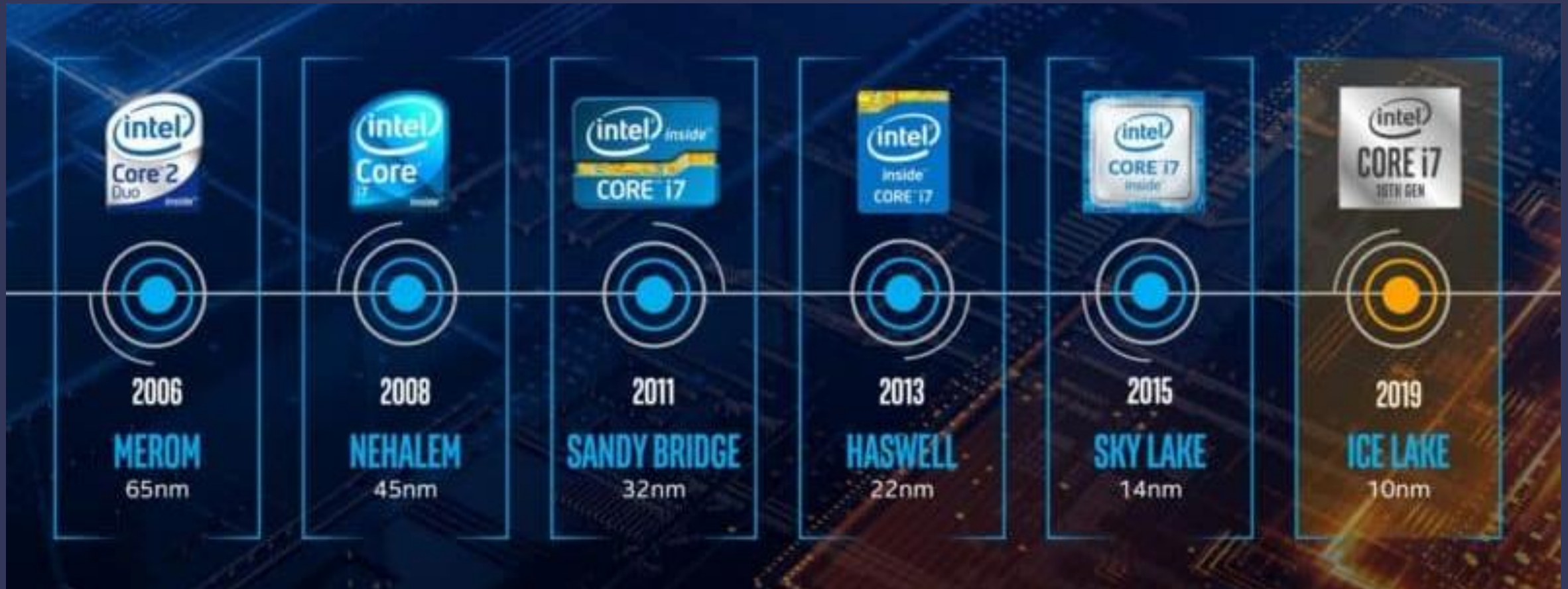
# Microprocessadores 1971-1996



# Microprocessadores e microcomputadores



# Microprocessadores Intel



# Microprocessador Intel i9 (2019-2020)

NEW 10<sup>TH</sup> GEN INTEL® CORE™ i9-10900K



WORLD'S FASTEST GAMING PROCESSOR<sup>1</sup>

UP TO  
**5.3\***  
GHZ

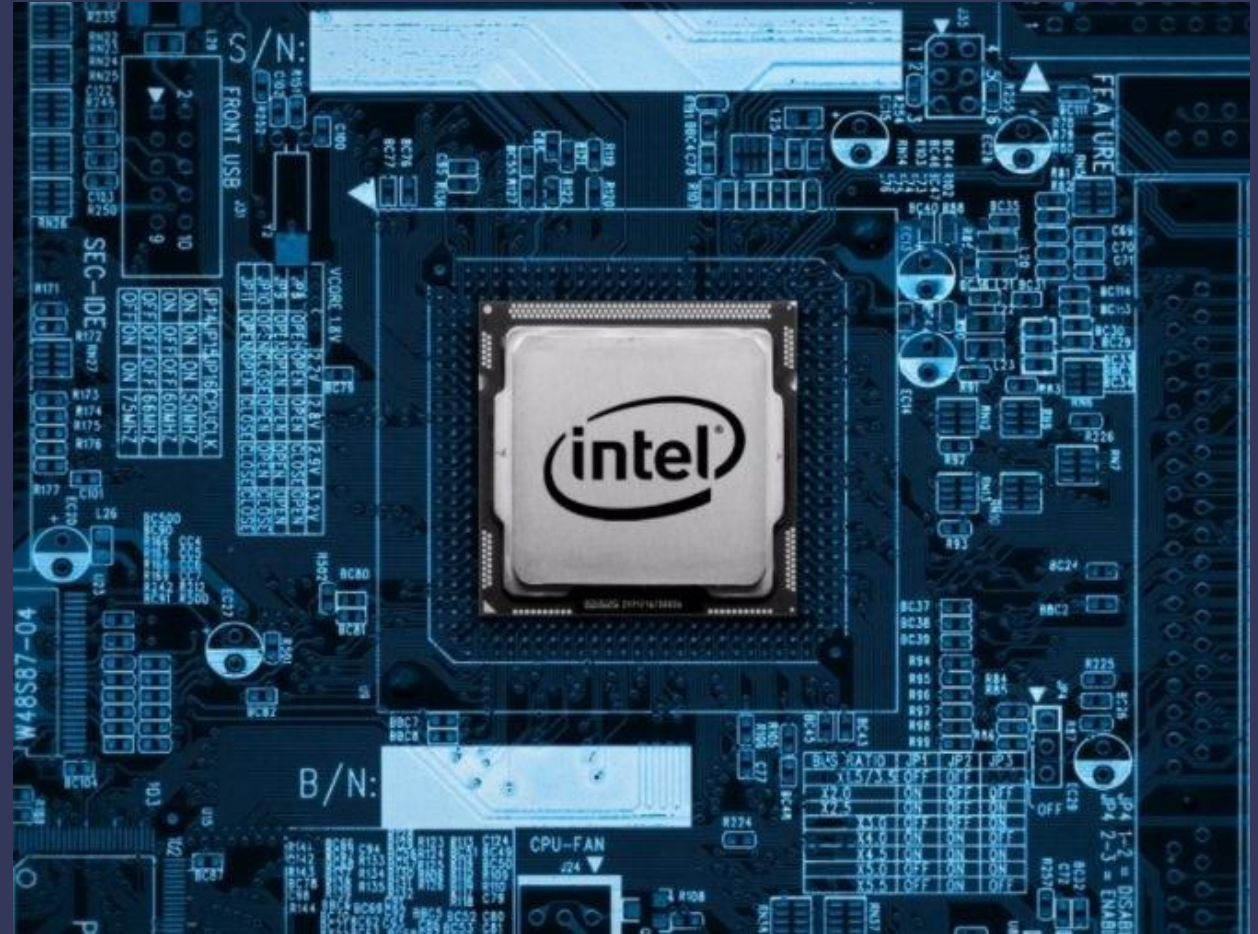
**10**  
CORES

**20**  
THREADS

intel  
**CORE™ i7**  
10TH GEN

intel  
**CORE™ i9**  
10TH GEN

intel  
**CORE™ i5**  
10TH GEN



# Microprocessador Intel i9 (2019-2020)

## NEW 9<sup>TH</sup> GEN INTEL<sup>®</sup> CORE<sup>™</sup> I9 PROCESSORS DESKTOP CALIBER PERFORMANCE ON THE GO



**5GHZ<sup>1</sup>**  
**8 CORES**  
**16 THREADS**

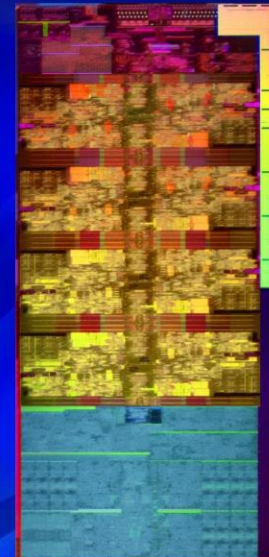
FULLY UNLOCKED  
**19-9980HK<sup>22</sup>**  
FOR MUSCLEBOOKS

LOCKED  
**19-9880H**  
FOR T&L CREATOR LAPTOPS

UP TO  
**18%<sup>2</sup>**  
HIGHER FPS

UP TO  
GAME  
+STREAM  
+RECORD  
**2.1X<sup>3</sup>**  
HIGHER FPS

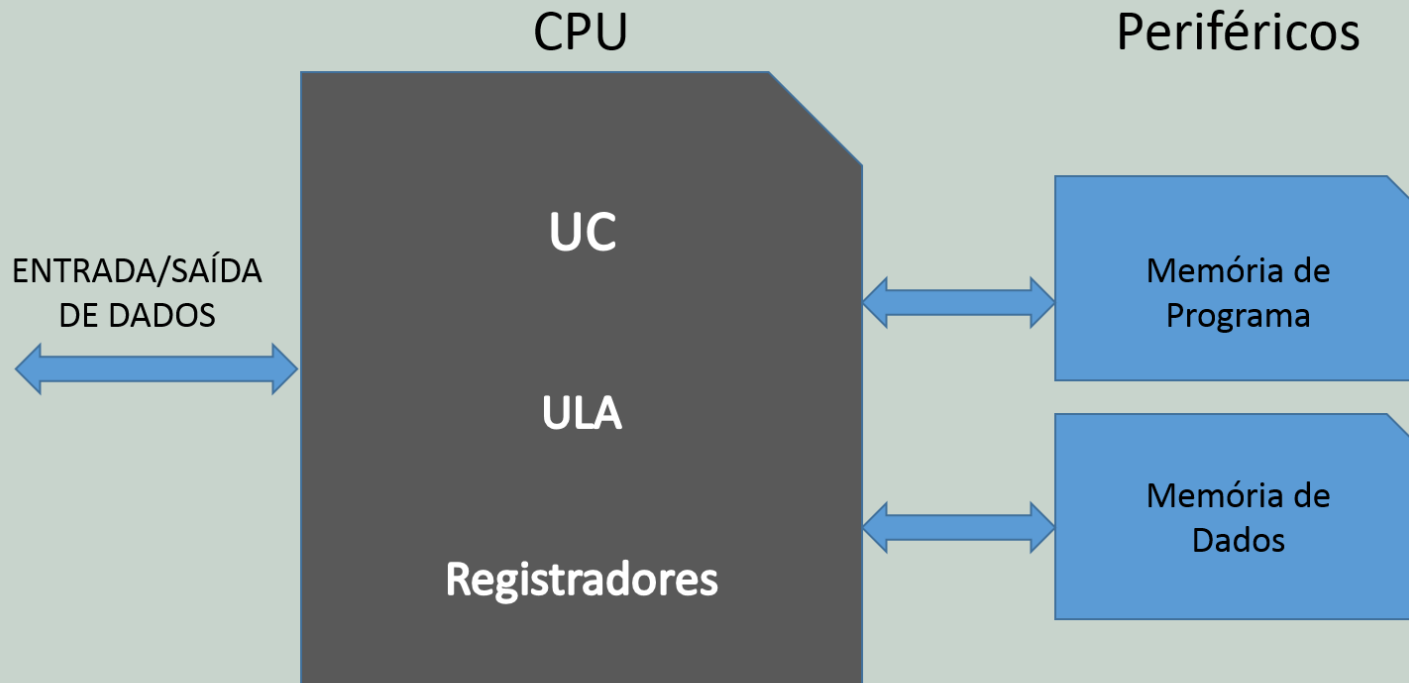
LESS WAIT, MORE CREATE  
**28%<sup>4</sup>**  
FASTER 4K VIDEO EDITING



For more complete information about performance and benchmark results, visit [www.intel.com/benchmarks](http://www.intel.com/benchmarks).



# Arquitetura de microprocessador



CPU = Central Processing Unit (microprocessador)

UC = Unidade de Controle

ULA = Unidade Lógica e Aritmética

Registradores = pequenos espaços de memória necessários à realização das operações da CPU.

# Introdução

- Bit: (Binary Digit - 0 ou 1) - Menor informação digital possível;
- Byte: Um termo especial, usado para designar palavra de oito bits. Um byte sempre é constituído de oito bits, sendo este tamanho da palavra da maioria dos sistemas microcontrolados atuais.

Bit	0
Nibble	0000
Byte	0000 0000
Word	0000 0000 0000 0000

- Registradores: conjunto de flip-flops para Armazenamento temporário de dados- 8 bits, 16 bits, 32 bits, etc;



# Introdução

## Número de bits dos Registradores

1 nibble



					Decimal	Hexa
Menor valor positivo	0	0	0	0	0	0
	0	0	0	1	1	1
	0	0	1	0	2	2
	0	0	1	1	3	3
	...	...	...	...		
Maior valor positivo	1	1	1	1	15	F

Capacidade:  $2^4 = 16$  nibbles

# Introdução

## Número de bits dos Registradores

1 byte

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-------	-------	-------	-------	-------	-------	-------	-------

1 byte = 8 bits

									Decimal	Hexa
Menor valor positivo	0	0	0	0	0	0	0	0	0	00
	0	0	0	0	0	0	0	1	1	01
	0	0	0	0	0	0	1	0	2	02
	0	0	0	0	0	0	1	1	3	03
	...	...	...	...	...	...	...	...		
Maior valor positivo	1	1	1	1	1	1	1	1	255	FF

Capacidade:  $2^8 = 256$  bytes

Intervalo dos valores positivos e negativos (notação com sinal):

0 a 127 = 00h a 7Fh

-1 a -128 = FFh a 80h

# Introdução

## Número de bits dos Registradores

1 word = 16 bits

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--------	--------	--------	--------	--------	--------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

### Intervalo dos valores positivos

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Capacidade:  $2^{16} = 65.536$  words

Intervalo dos valores positivos: 0 a 65.535 = 0000h a FFFFh

Intervalo dos valores positivos e negativos (notação com sinal):

0 a 32.767 = 0000h a 7FFFh

-1 a -32.768 = FFFFh a 8000h

# Introdução

## Número de bits dos Registradores

**double word = 32 bits**

Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Capacidade:  $2^{32} = 4.294.967.296$  double-words

Intervalo dos valores positivos: 0 a 4.294.967.295 = 0000 0000h a FFFF FFFFh

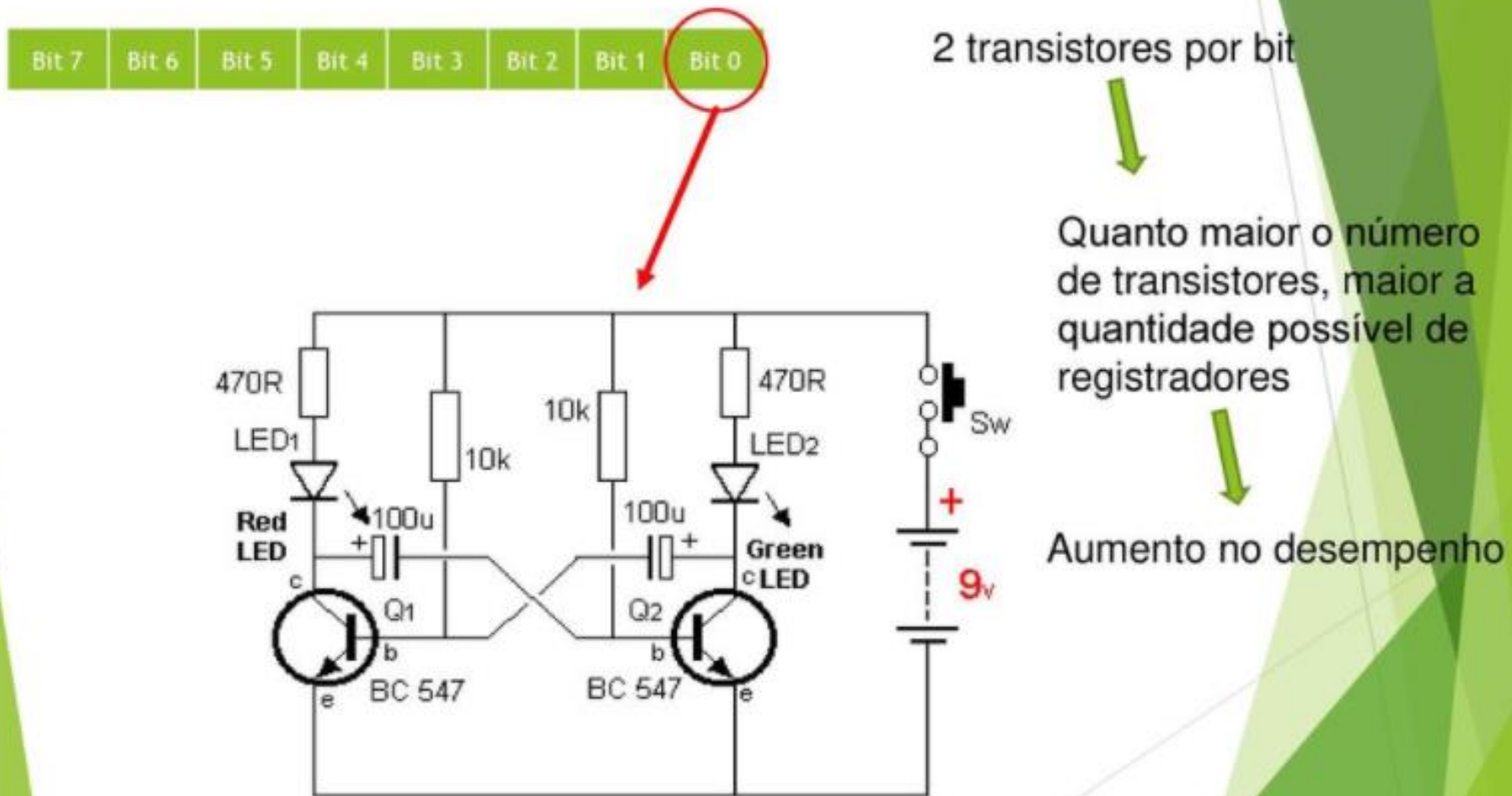
### **Intervalos considerando números com sinal:**

Intervalo positivo: 0 a 2.147.483.647 = 0000 0000 a 7FFF FFFFh

Intervalo negativo: -1 a -2.147.483.648 = FFFF FFFF a 8000 0000h

# Introdução

## Quantidade de Transistores



# Introdução

## Frequência de Clock

Ano	Processador	Frequência de Clock	Tecnologia (tamanho transistor)
1971	4004	108 kHz	10 $\mu\text{m}$
1972	8008	200 kHz	
1976	8085	5 MHz	
1978	8086	5 a 10 MHz	
1979	8088	5 a 10 MHz	
1989	80486	25 a 50 MHz	
1997	Pentium II	450 MHz	0,25 $\mu\text{m}$
1999	Pentium III	500 MHz até 1,2 GHz	0,25 $\mu\text{m}$
2004	Pentium 4 (Prescott)	4 GHz	90 nm
2007	Core 2	3,33 GHz	45 nm
2010	Core i5	3,33 GHz	32 nm

Aumento do desempenho

# Introdução

## Índice de Desempenho de um Microprocessador (velocidade de processamento)

$$\text{Tempo de processamento} = \frac{\text{tempo}}{\text{ciclo}} \times \frac{\text{número de ciclos}}{\text{instrução}} \times \frac{\text{instruções}}{\text{programa}}$$

1. Aumento da frequência de clock
2. Aumento do número interno de bits
3. Aumento do número externo de bits
4. Aumento da capacidade e velocidade da memória cache
5. Redução do número de ciclos para execução de cada instrução
6. Execução de instruções em paralelo

# Introdução

## Redução do número de ciclos para executar cada instrução

A execução de uma instrução normalmente é feita em duas etapas: **busca** (onde a instrução é transferida da memória para a unidade de decodificação) e **execução** (onde os sinais de controle ativam, em uma sequência lógica, todas as unidades envolvidas na execução).

No microprocessador 8085 as instruções mais rápidas são executadas em quatro ciclos de clock; as mais lentas, em até 16 ciclos de clock. A redução do número de ciclos de clock na execução de uma instrução torna o processamento mais rápido.



# Introdução

## **Aumento da capacidade e velocidade da memória cache**

Como já foi dito anteriormente, ao longo dos anos, o aumento de velocidade de processamento dos microprocessadores tem sido muito maior do que o aumento da velocidade de acesso à memória principal. Assim, a velocidade de acesso à memória principal torna-se um limitador de desempenho dos processadores. Em razão desse problema foi criada a memória cache. A memória cache (constituída de memória RAM estática) é usada para acelerar a transferência de dados entre a CPU e a memória principal (constituída de RAM dinâmica, de menor volume, porém mais lenta). O aumento da capacidade e da velocidade da memória cache resulta no aumento da velocidade de transferência de dados entre a CPU e a memória principal e, conseqüentemente, resulta no aumento do desempenho global do sistema.

# Introdução

## **Execução de instruções em paralelo**

O microprocessador 8085 compartilha um barramento comum entre suas unidades internas e seus periféricos, o que significa dizer que não permite a execução simultânea de duas operações que utilizem o barramento. Assim, apenas uma instrução é executada por vez. Uma arquitetura que permita que duas ou mais operações sejam executadas simultaneamente torna o processamento mais rápido.

# Introdução

• Hardware: Atualmente não é mais considerada a parte “imutável” do Microcomputador

• Software:

– Instrução: Comando específico de um microprocessador (movimentação de dados, operações aritméticas e lógicas, etc.);

– Programa: Seqüência lógica de instruções que podem ser reconhecidas pelo microprocessador, que quando executadas em uma ordem correta, resulta na finalização de uma função ou algoritmo

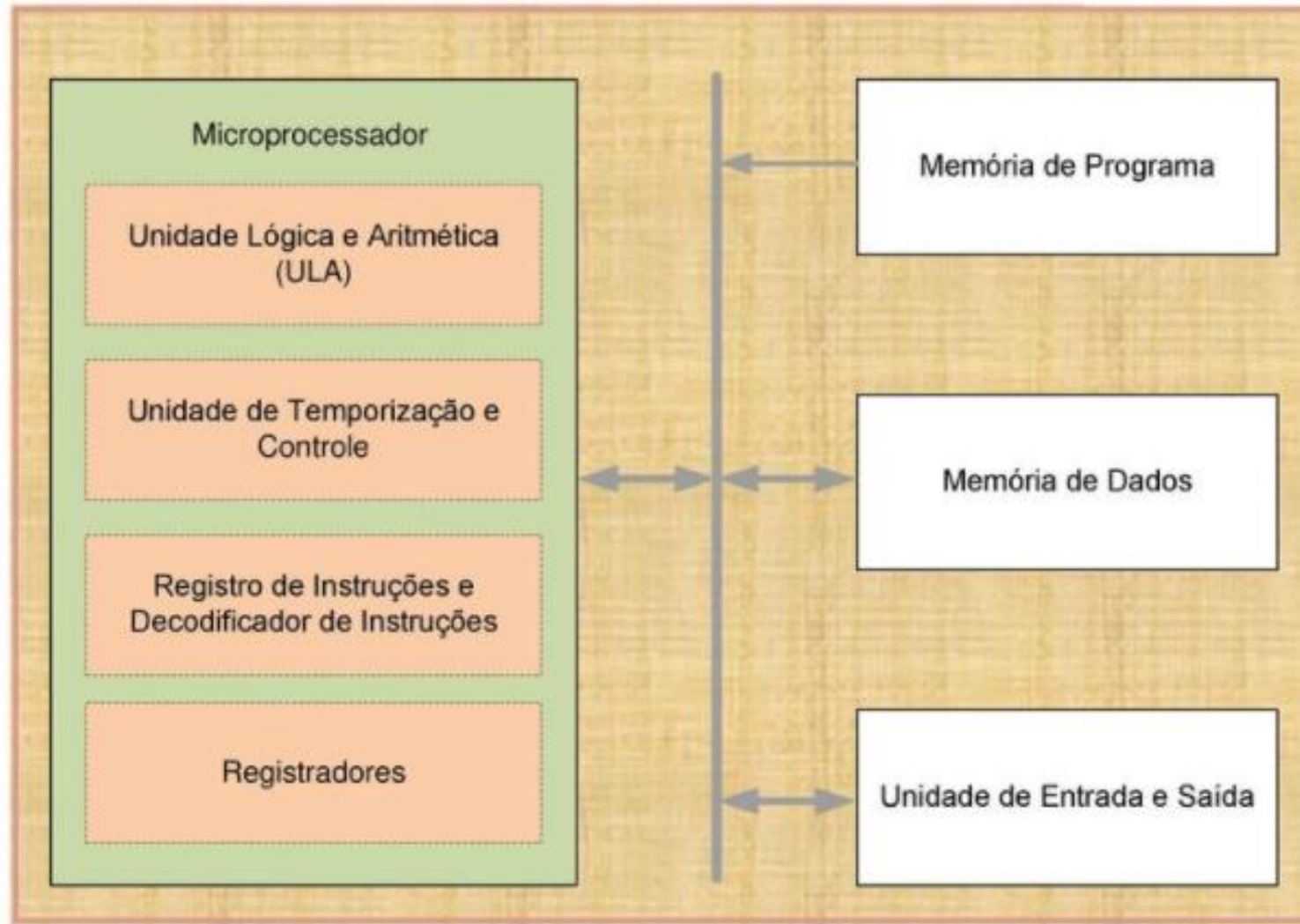
• Firmware: Programa de função bastante específica que está armazenado em memória não volátil

# Introdução

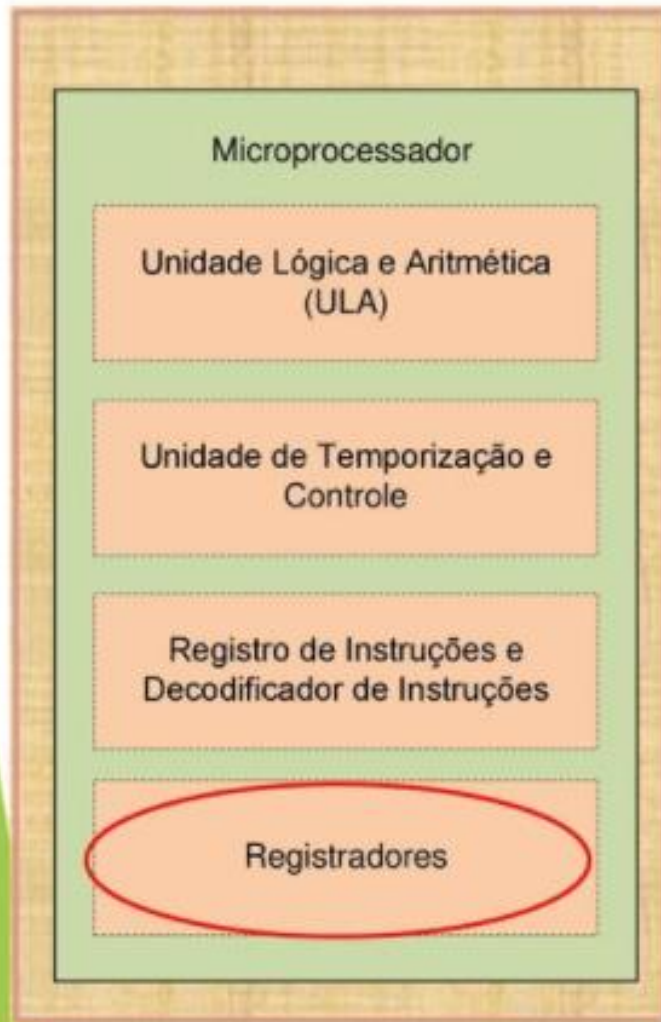
## Dinâmica de um sistema microprocessado

- *O microcomputador é um sistema de processamento capaz de buscar e executar instruções (programas) alocados em memória;*
- *Após a energização do microcomputador, é gerado um sinal de reset que zera o Program Counter (PC);*
- *O microprocessador irá buscar e executar a instrução que está localizada no endereço de memória definida pelo PC;*

# Microprocessador: arquitetura



# Microprocessador: arquitetura

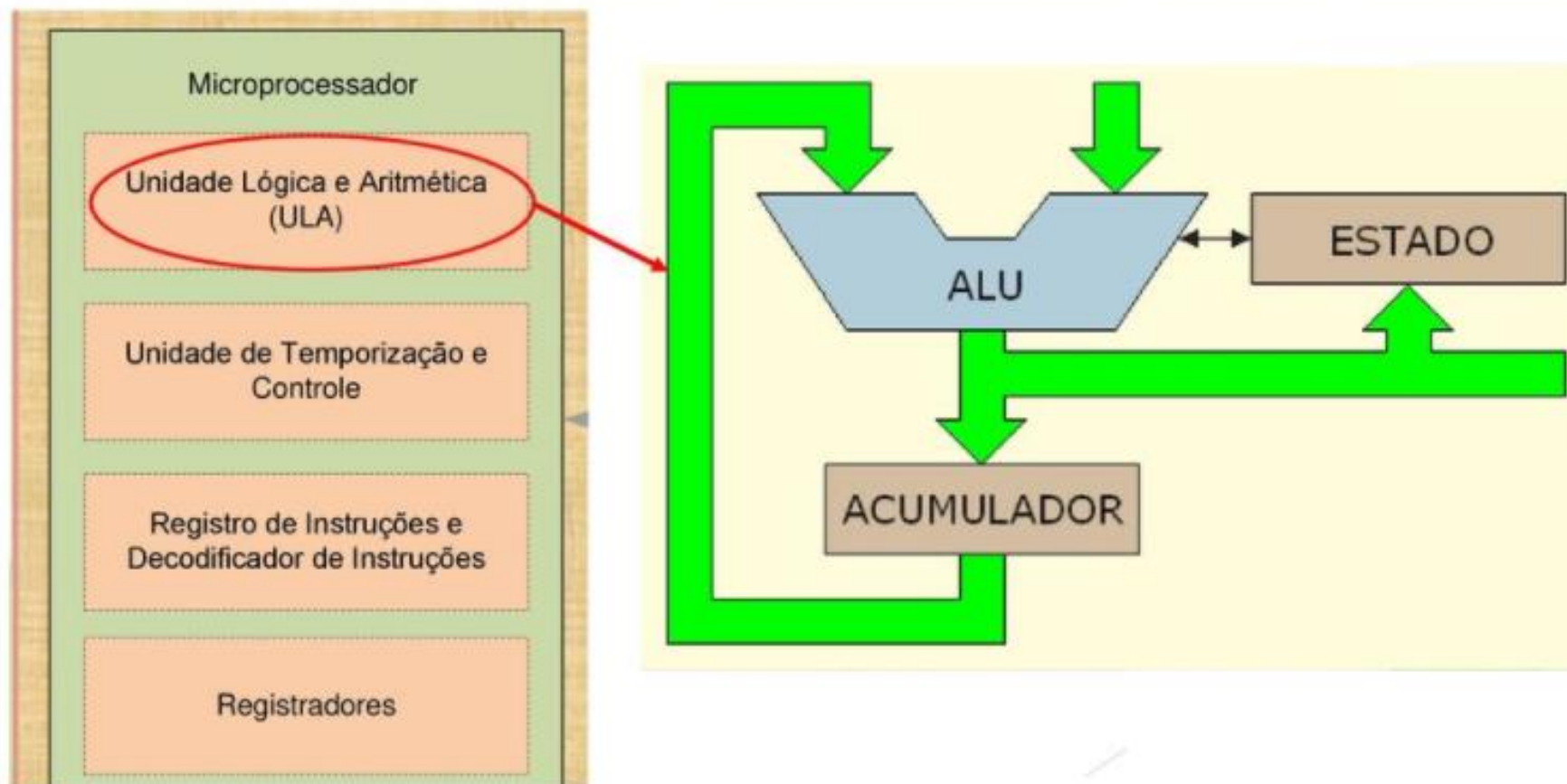


**Registradores** - São usados para o armazenamento interno da CPU. Existem diversos registradores na CPU e o principal deles é chamado de **Acumulador**.

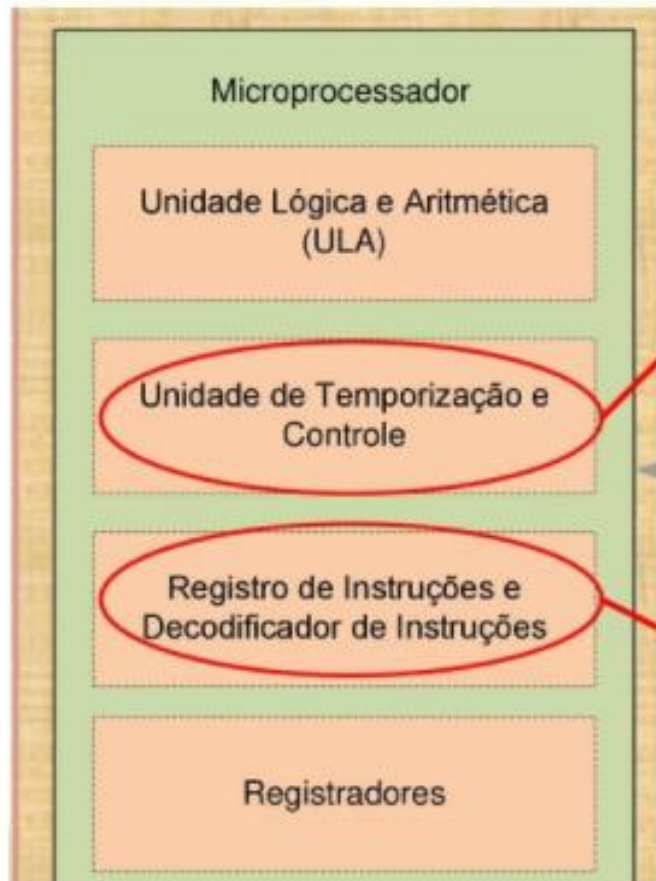
Os registradores são construídos com flip-flops, que podem reter (armazenar) dados. O acumulador contém um dos dados usados na operação que se deseja e ainda o resultado da operação, que substitui o dado original.

# Microprocessador: arquitetura

**Unidade Lógica e Aritmética (ULA ou ALU)** - realiza funções básicas de processamento de dados (adição, subtração, funções lógicas, etc.).



# Microprocessador: arquitetura



## Unidade de Temporização e Controle:

Responsável pela geração dos sinais de controle para todas as unidades, a partir da informação da Unidade de Decodificação. Os sinais de controle são sincronizados, de acordo com o sinal de clock.

## Unidade de Decodificação:

Registrador de Instruções e Decodificador de Instruções

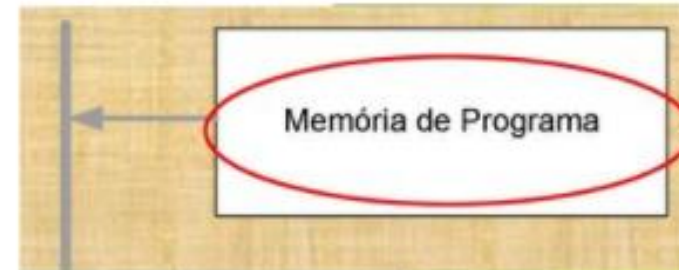
Responsável pela identificação da instrução a ser executada, a partir do código de operação (opcode).



# Microprocessador: tipos de memória

## Memória de Programa:

**ROM** (*Read-Only Memory*) – Memória que permite apenas a leitura, ou seja, as suas informações são gravadas pelo fabricante uma única vez e após isso não podem ser alteradas ou apagadas, somente acessadas.



**PROM** (*Programmable Read-Only Memory*) – Podem ser escritas com dispositivos especiais mas não podem mais ser apagadas.

**EPROM** (*Erasable Programmable Read-Only Memory*) – Podem ser apagadas pelo uso de radiação ultravioleta permitindo sua reutilização.

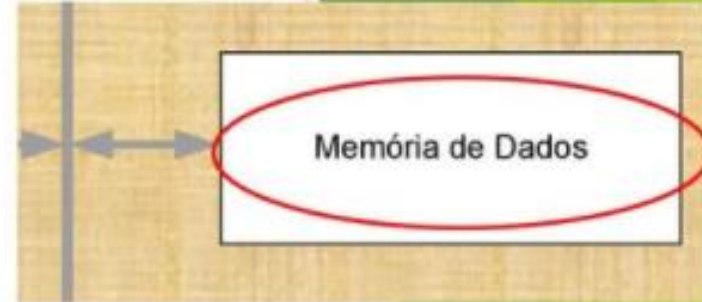
Exemplo para o caso do 8051: Microcontrolador **8751**

**EEPROM** – Permite apagar eletricamente e gravar várias vezes.

**FLASH** – Equivalente à memória EEPROM. Porém, ocupa menos espaço; menor consumo de energia; alta durabilidade.

# Microprocessador: tipos de memória

## Memória de Dados:



Memória **RAM** – Permite a leitura e a gravação de dados.

Memória Dinâmica (**DRAM**) – Baixa densidade, mas lenta. Capacitores com circuitos com “atualização de dados - *refresh*”.

Memória estática (**SRAM**) – Alta densidade. Rápida. Baseada em Flip-flops.

Memória **CACHE** - Pequena quantidade de memória RAM estática (SRAM) usada para acelerar o acesso à memória principal (RAM dinâmica).

Quando há necessidade de transferir dados da (para) memória dinâmica, estes são antes transferidos para a memória cache

# Microprocessador: arquitetura

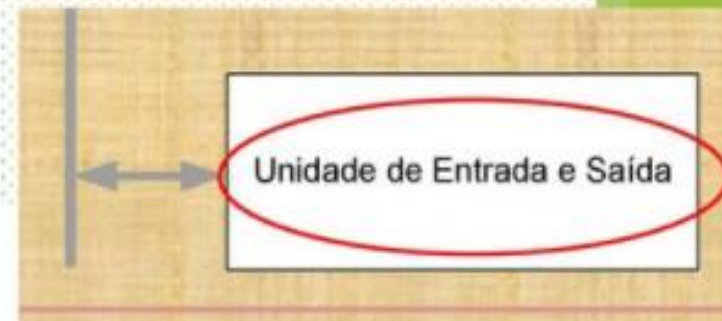
## Unidade de Entrada e Saída (I/O)

A entrada de dados de um microprocessador (via teclado, mouse ou outros dispositivos) e a saída de dados (via vídeo, impressora ou outros) exige circuito integrado adicional como interface.

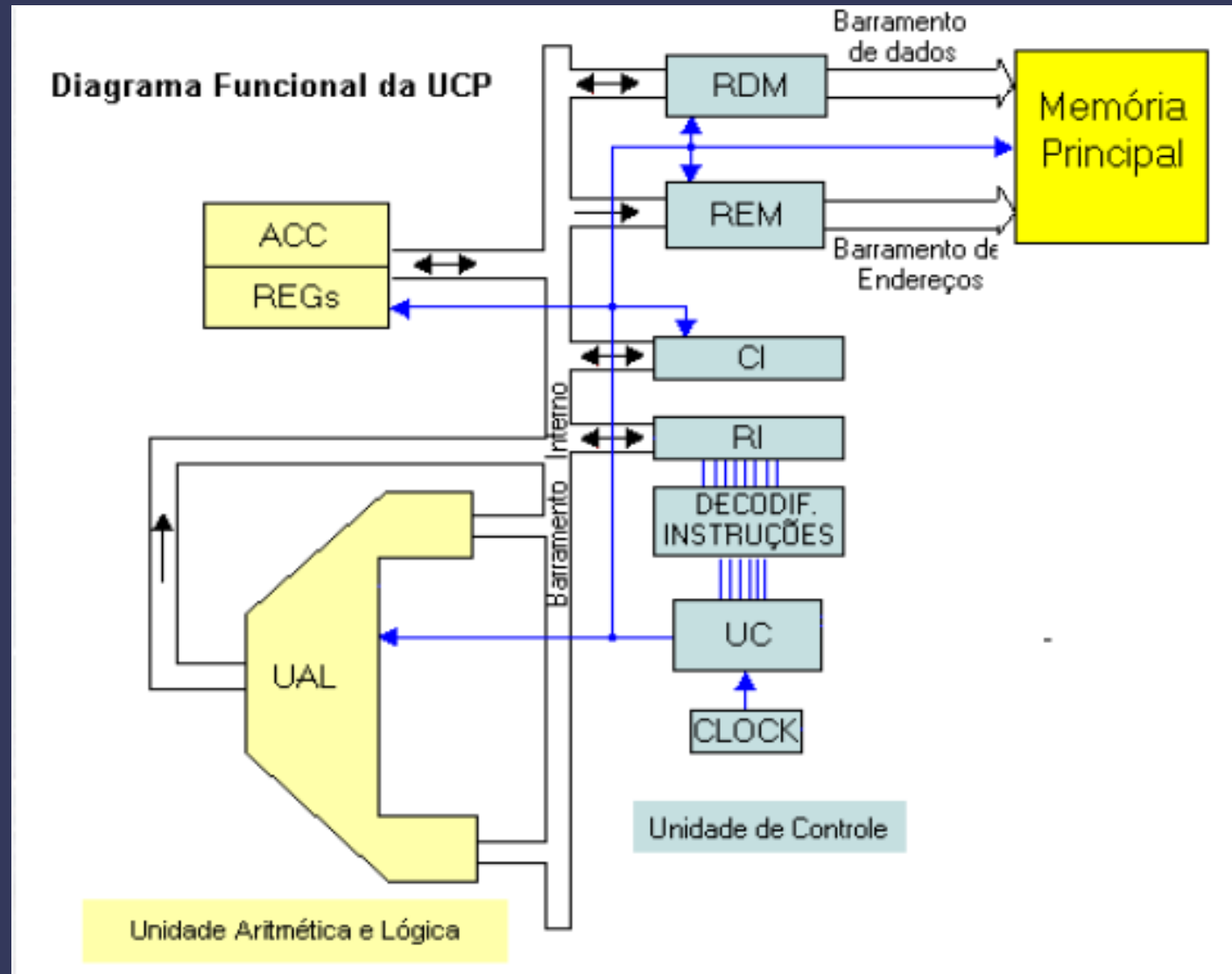
O microcontrolador já possui essa unidade internamente.

Exemplos de periféricos usados como unidade de entrada e saída:

- CI 8156 – RAM e porta de entrada e saída
- CI 8355 – ROM e portas de entrada e saída)



# Microprocessador: arquitetura

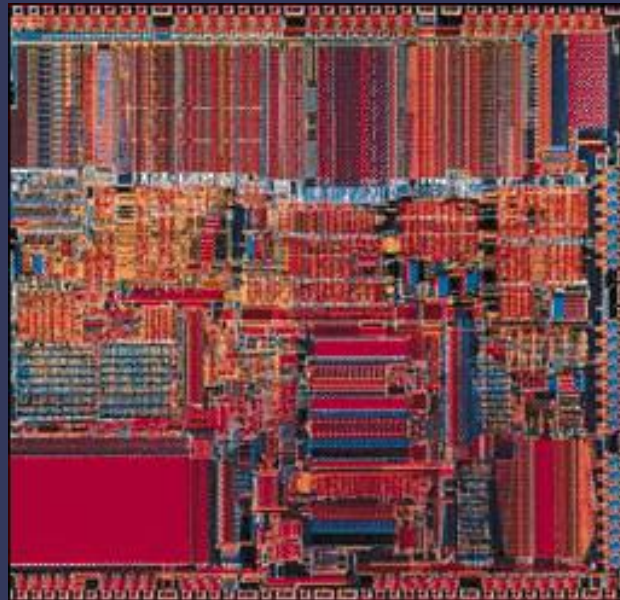


# Circuito integrado: microprocessadores



O primeiro microprocessador:

Intel 4004 (1960)

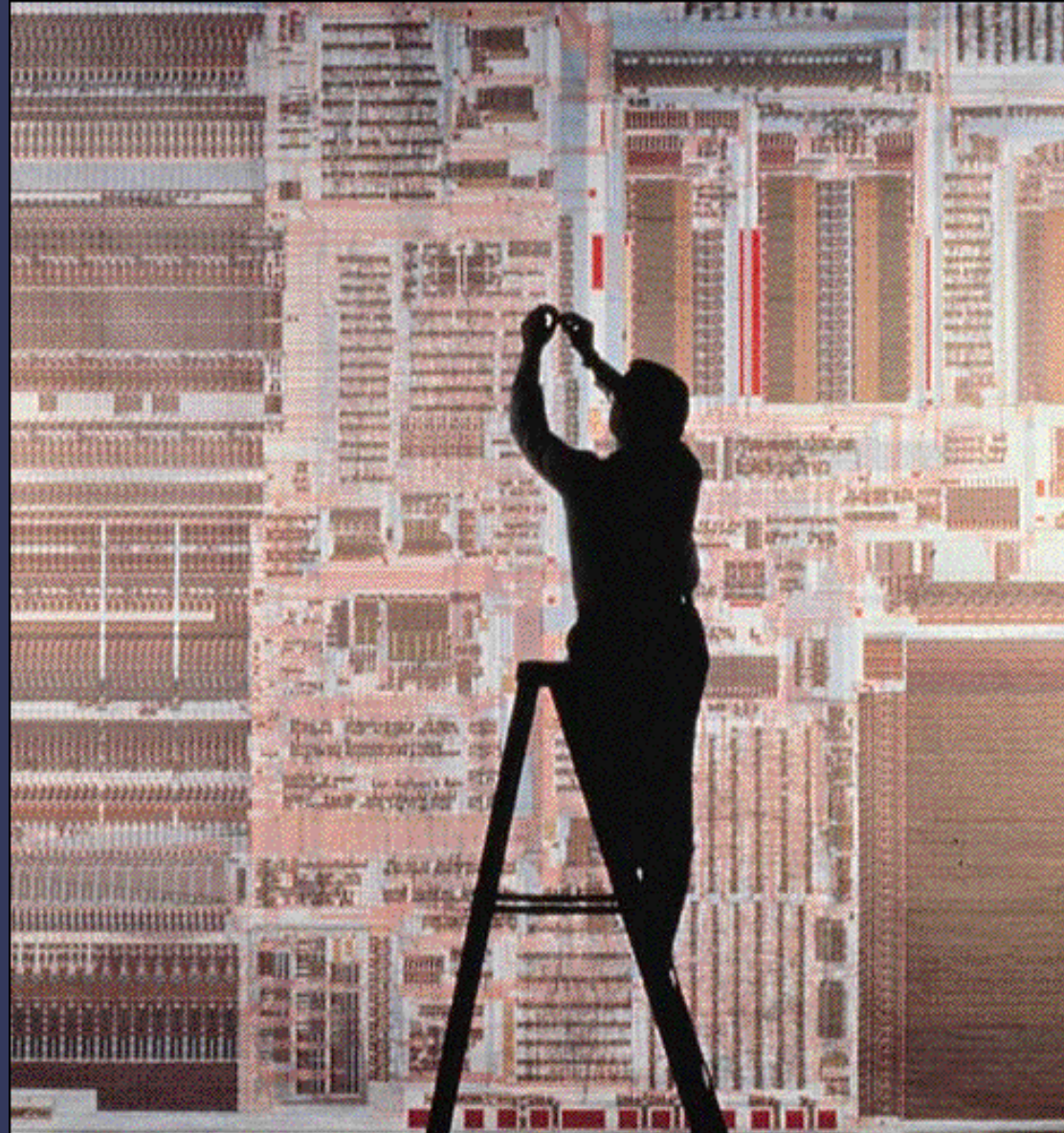


Processador Intel 386  
(1986)

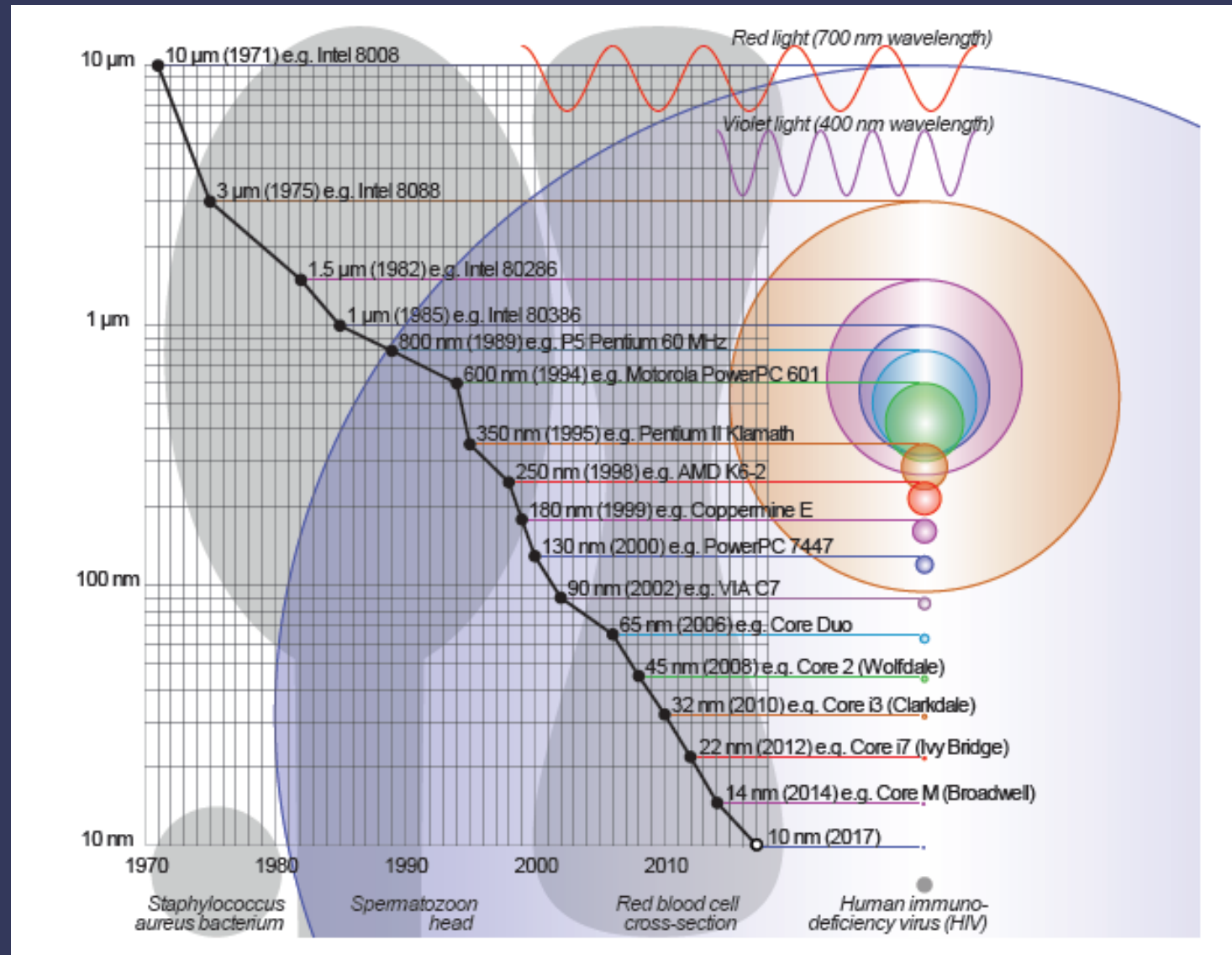


Processador Intel  
PentiumPro (1996)

# Depuração de um projeto de circuito integrado na década 1980



# Microprocessadores: Lei de Moore até 2017



# Arquiteturas de microprocessadores

**CISC (Intel)**

*Complex Instruction Set Computing*

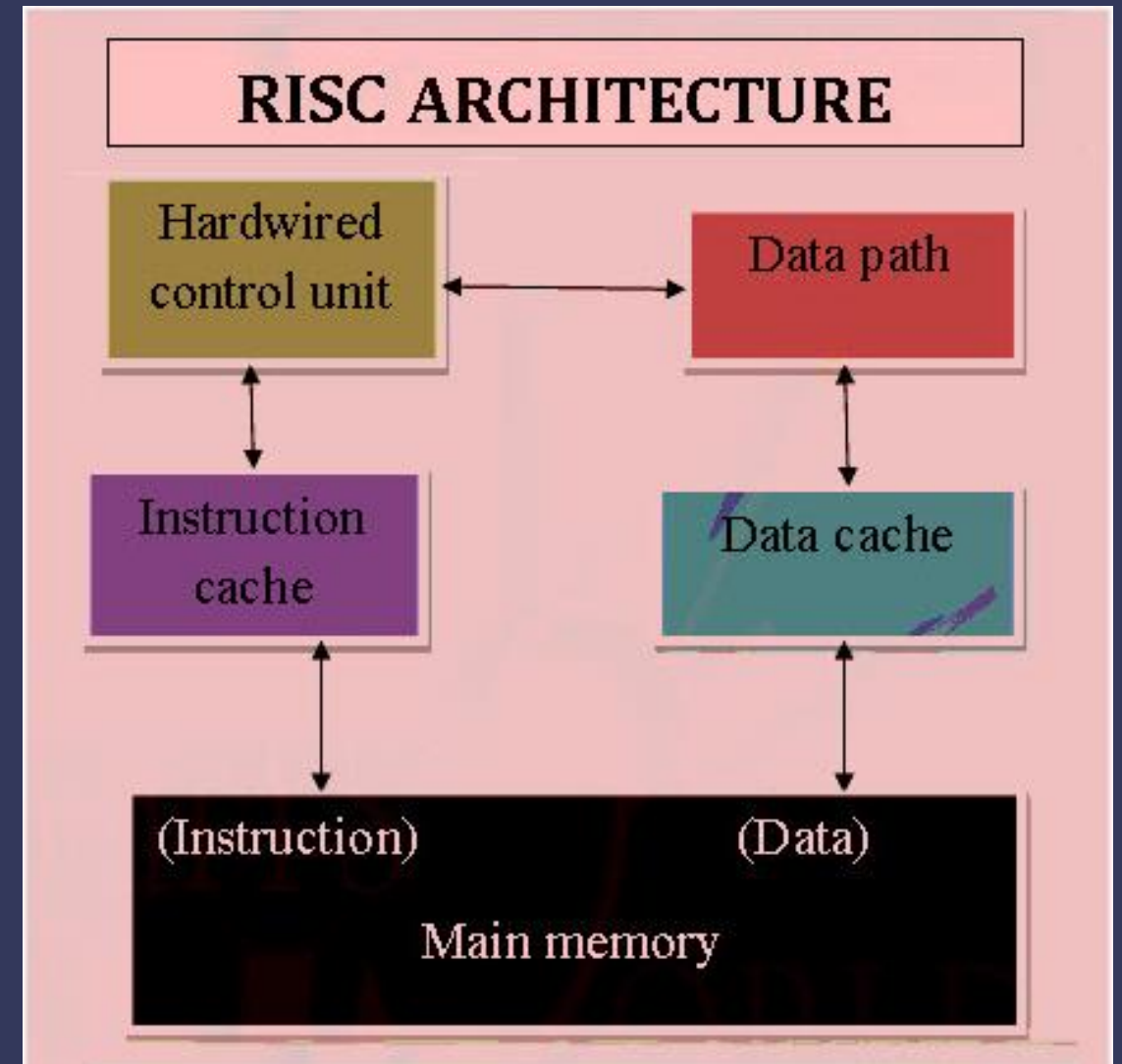
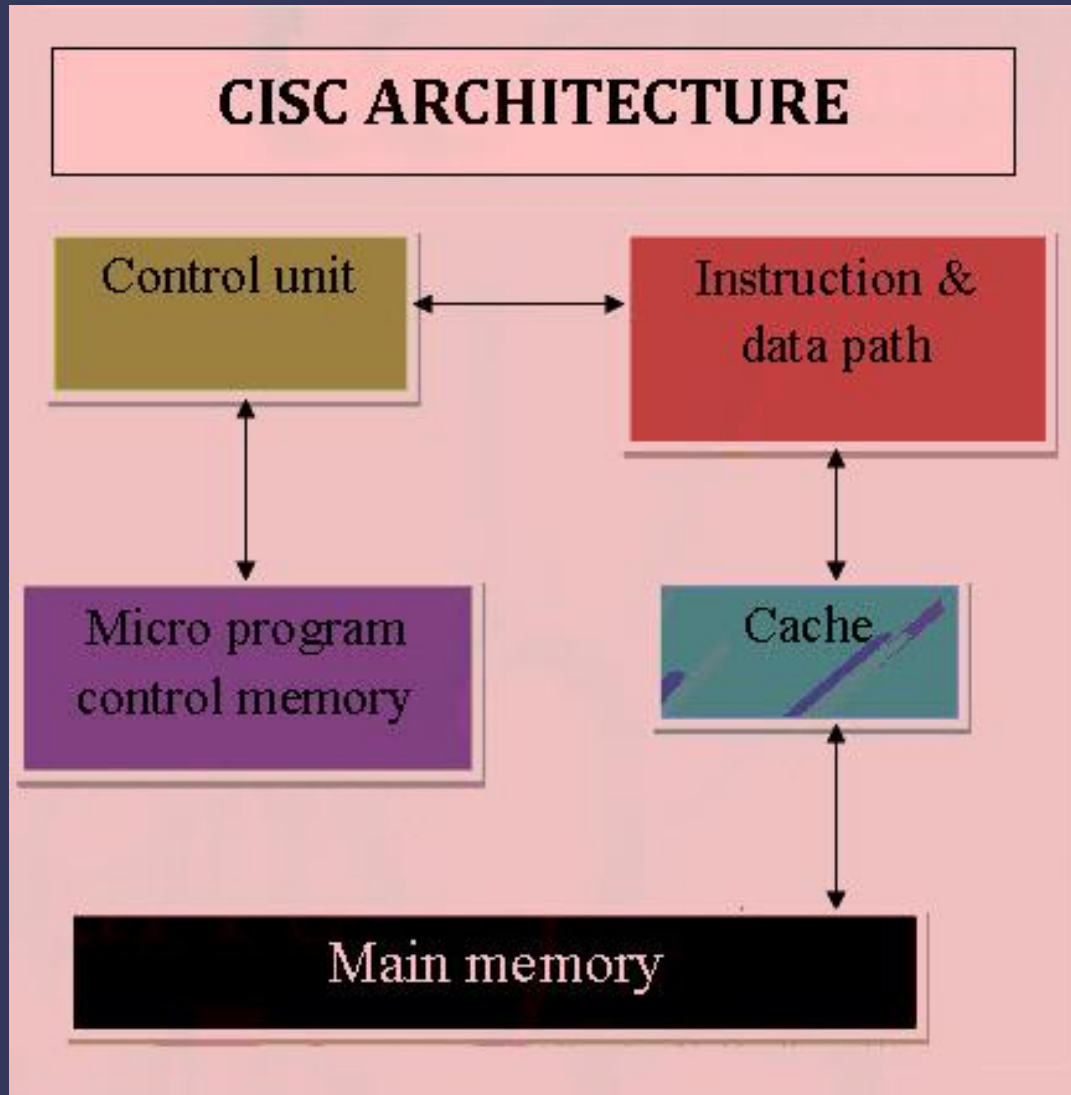
**X**

**RISC (ARM)**

*Reduced Instruction Set Computing*



# Arquiteturas de microprocessadores CISC e RISC



# Arquiteturas de microprocessadores CISC e RISC

Arquiteturas em uso nos computadores atuais:

- **CISC** - Complex Instruction Set Computing (Computador com Conjunto Complexo de Instruções) Exemplos: Intel e AMD
- **RISC** - Reduced Instruction Set Computing (Computador com Conjunto Reduzido de Instruções) Exemplos: PowerPC (da Apple, Motorola e IBM), SPARC (SUN) e MIPS R2000
- **Híbrida** - Combinação de ambas arquiteturas. Exemplo: Pentium Pro. O núcleo mais interno usa filosofia RISC.

# Arquiteturas de microprocessadores CISC

A arquitetura CISC tenta minimizar o número de instruções por programa, sacrificando o número de ciclos por instrução. Os computadores baseados na arquitetura CISC são projetados para diminuir o custo da memória. Assim, programas grandes precisam de mais armazenamento, aumentando o custo da memória tornando mais caro o dispositivo final. Para resolver esses problemas, o número de instruções por programa pode ser reduzido incorporando o número de operações em uma única instrução, tornando as instruções mais complexas.

# Arquiteturas de microprocessadores RISC

A arquitetura RISC é usada em dispositivos portáteis devido à sua eficiência energética. Por exemplo, os dispositivos portáteis Apple iPod e Nintendo DS e os *smartphones*. RISC é um tipo de arquitetura de microprocessador que usa um conjunto de instruções altamente otimizado. O RISC faz o oposto, reduzindo os ciclos por instrução ao custo do número de instruções por programa. O *pipelining* é um dos recursos exclusivos do RISC. É realizado sobrepondo a execução de várias instruções em um pipeline. Ele tem uma vantagem de alto desempenho sobre o CISC.

# Arquiteturas de microprocessadores CISC e RISC

Algumas Características RISC X CISC:	
RISC	CISC
Instruções básicas executadas em apenas 1 ciclo - uso intenso de superposição na execução de instruções (através de pipeline)	No mínimo 4 ciclos de clock (8085) para executar uma instrução. Tem pouca ou nenhuma superposição na execução de instruções
Uso reduzido da memória - basicamente 2 instruções de acesso à memória (load/store).	Muitas instruções com acesso à memória. Uso intenso da memória (load, store, mov... )
Instruções de tamanho fixo. Exemplo: as instruções do PIC 16F628 têm tamanho fixo de 14 bits.	Instruções de tamanho variável. Exemplo: 8085 tem instruções de 1, 2 e 3 bytes.
Muitos registradores. Exemplo: PIC típico possui de 32 a 128 registradores.	Poucos registradores. Exemplo: 8085 possui 7 registradores de propósito geral (A, B, C, D, E, H e L)
Não há necessidade de decodificação das instruções antes de executá-las. As instruções são semelhantes às micro-instruções da arquitetura CISC.	Ciclo de busca inclui busca na memória e identificação em decodificadores. Uso de micro-instruções gravadas no processador.

# Arquiteturas de microprocessadores CISC e RISC

## Algumas Características RISC X CISC:

RISC	CISC
Número pequeno (ou médio de instruções). Exemplo: PIC 12 e PIC 16 possuem 32 instruções.	Número elevado de instruções. Exemplo: 8085 possui cerca de 74 instruções, que resultam em cerca 255 códigos de operação (opcodes)
Instruções simples - 1 ciclo para execução	Instruções complexas - múltiplos ciclos para execução.
Programa compilado tem maior número de instruções em assembly, comparado com mesmo programa implementado em arquitetura CISC. Uso maior de memória	Menor número de instruções assembly, comparado com mesmo programa implementado em arquitetura RISC. No entanto, é mais lento na execução. Uso de uma quantidade menor de memória.

# Arquiteturas de microprocessadores CISC e RISC

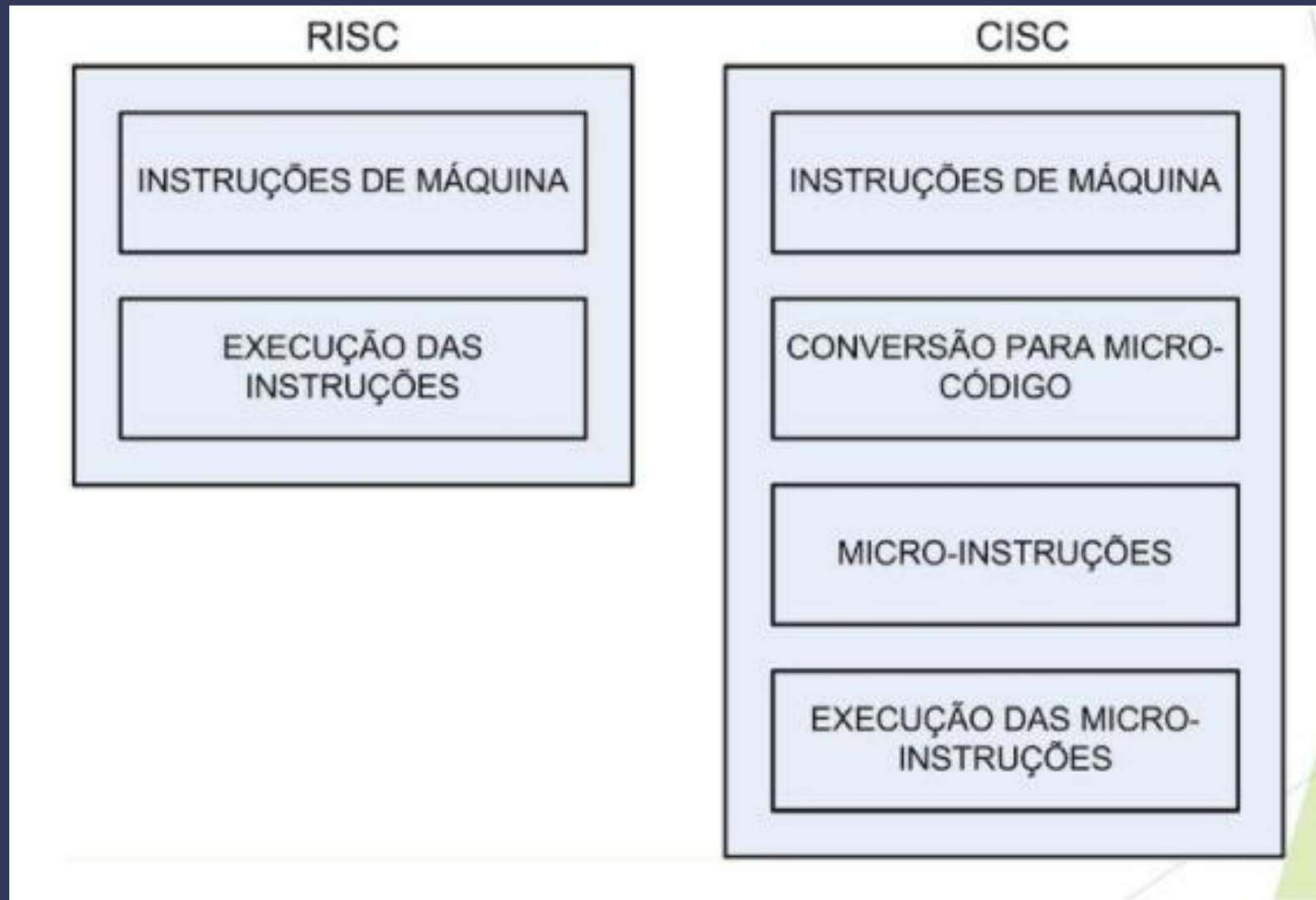
## Algumas Características RISC X CISC:

$$\text{Tempo de processamento} = \frac{\text{tempo}}{\text{ciclo}} \times \frac{\text{número de ciclos}}{\text{instrução}} \times \frac{\text{instruções}}{\text{programa}}$$

Na arquitetura **RISC**, esse é um item fundamental para aumento do desempenho – a execução em paralelo de instruções (com pipeline) reduz o número de ciclos na execução de cada instrução.

Na arquitetura **CISC**, esse é um item fundamental para aumento do desempenho – o número de instruções por programa pode ser reduzido, uma vez que cada instrução executa várias tarefas.

# Arquiteturas de microprocessadores CISC e RISC





# Arquiteturas de microprocessadores CISC e RISC

## CONJUNTO DE INSTRUÇÕES:

Grupos de instruções mais comuns em processadores de qualquer arquitetura:

**Instruções de desvio** (No CISC o valor de retorno é guardado na pilha; no RISC é guardado em um registrador.)

**Instruções de transferência entre registradores e memória**

(No RISC: load/store; no CISC: load, store, mov etc)

**Instruções de transferência entre registradores**

**Instruções de transferência entre posições de memória**

**Operações aritméticas** (soma, subtração ...)

**Operações lógicas** (and, or, not, rotação ...)

# Arquiteturas de microprocessadores CISC e RISC

## CICLO DE EXECUÇÃO:

**RISC** → As instruções são executadas em um único ciclo de via de dados. São instruções muito parecidas com as micro-instruções da arquitetura CISC. Não precisam de decodificação.

Não é possível ter instruções de multiplicação e divisão, por exemplo, por exigir muitos ciclos para execução. Multiplicações são resolvidas com adições e deslocamentos.

**CISC** → Antes de executar uma instrução, há necessidade de busca da instrução na memória e de decodificação. Utiliza-se micro-códigos gravados no processador, para a execução das instruções.

# Arquiteturas de microprocessadores CISC e RISC

## MEMÓRIA E REGISTRADORES:

**RISC** → Possui uma quantidade muito grande de registradores (em média 512 – com 32 visíveis por vez: 8 para variáveis globais e ponteiros, 8 para parâmetros de entrada, 8 para variáveis locais e 8 para parâmetros de saída).

Número reduzido de acesso à memória

(o acesso à memória torna o processamento mais lento).

Alocação de variáveis em registradores.

Um ou dois modos de endereçamento para acesso à memória

**CISC** → Possui um número reduzido de registradores, comparado com o RISC.  
Alocação de variáveis em posições de memória, ao invés de registradores.

Vários modos de endereçamento para acesso à memória.

# Arquiteturas de microprocessadores CISC e RISC

## MICRO-CÓDIGOS:

**RISC** → As instruções geradas por um compilador para uma máquina RISC são executadas diretamente no hardware, sem o uso de micro-códigos.

A ausência de interpretação contribui para o aumento da velocidade de execução.

**CISC** → As instruções básicas são gravadas na forma de micro-códigos, que atuam no hardware estabelecendo os passos de cada instrução.

Há necessidade de busca e decodificação das instruções.

O programa compilado tem uma quantidade menor de instruções assembly do que um programa RISC, mas é mais lento na execução.

# Arquiteturas de microprocessadores CISC e RISC

## VANTAGENS RISC:

- Velocidade de execução
- O uso de pipeline torna os processadores RISC duas a quatro vezes mais rápidos que um CISC de mesmo clock
- Simplicidade de Hardware
- Ocupa menos espaço no chip, devido ao fato de trabalhar com instruções simples.
- Instruções de máquina simples e pequenas, o que aumenta sua performance.

# Arquiteturas de microprocessadores CISC e RISC

## **DESVANTAGENS RISC:**

- O desempenho de um processador RISC depende diretamente do código gerado pelo programador. Um código mal desenvolvido pode resultar em tempo de execução muito grande.
- Um programa originalmente compilado para uma máquina CISC tem um equivalente compilado para máquina RISC com uma quantidade muito maior de códigos assembly, ocupando um espaço maior na memória.
- A arquitetura RISC requer sistema de memória rápida para alimentar suas instruções. Normalmente possuem grande quantidade de memória cache interna, o que encarece o projeto.

# Arquiteturas de microprocessadores CISC e RISC



P8085AH-2  
K936546  
INTEL © 1976

## 8085

- 1976
- CPU de 8 bits
- Clock até 6 MHz
- Vcc = +5V; 40 pinos;
- Endereça até 64 kB de memória;
- 6 conjuntos de pinos:

- Endereçamento;
- Dados;
- Controle e status;
- Energização e clock;
- Pinos seriais;
- Sinalização externa.



8085

Pin	Signal	Pin	Signal
1	X <sub>1</sub>	20	V <sub>DD</sub>
2	X <sub>2</sub>	21	HOLD
3	RESET OUT	22	HLDA
4	SDO	23	CLK
5	SIO	24	RESET IN
6	TRAP	25	READY
7	RST 7.5	26	IO/M
8	RST 6.5	27	S1
9	RST 5.5	28	RD
10	INTR	29	WR
11	INTA	30	ALE
12	AD0	31	S0
13	AD1	32	A15
14	AD2	33	A14
15	AD3	34	A13
16	AD4	35	A12
17	AD5	36	A11
18	AD6	37	A10
19	AD7	38	A9
20	V <sub>SS</sub>	39	A8
		40	A7

# Arquiteturas de microprocessadores CISC e RISC

## 8085 × 8086 / 8088

Característica	Microprocessador 8085	Microprocessador 8088	Microprocessador 8086
Barramento de endereço	16 bits	20 bits	20 bits
Capacidade de endereçamento de memória	65.536 ( 64 kB )	1.048.576 ( 1 MB )	1.048.576 ( 1 MB )
Barramento de dados	8 bits	Interno: 16 bits Externo: 8 bits	Interno: 16 bits Externo: 16 bits
Manipulação de STRINGS	NÃO	SIM	SIM
Registradores Internos	8 bits e 16 bits	16 bits	16 bits
Uso de segmentação para endereçamento	NÃO	SIM	SIM
Aritmética Decimal completa	NÃO	SIM	SIM
Etapas de Busca e Execução	Em sequência:  <b>Busca → Executa</b>	Unidades Independentes:  Unidade de Interfaceamento com Barramento ( <b>BIU</b> ) – responsável pela <b>Busca</b> e  Unidade de Execução ( <b>EU</b> )	Unidades Independentes: Unidade de Interfaceamento com Barramento ( <b>BIU</b> ) – responsável pela <b>Busca</b> e  Unidade de Execução ( <b>EU</b> )



# Arquiteturas de microprocessadores

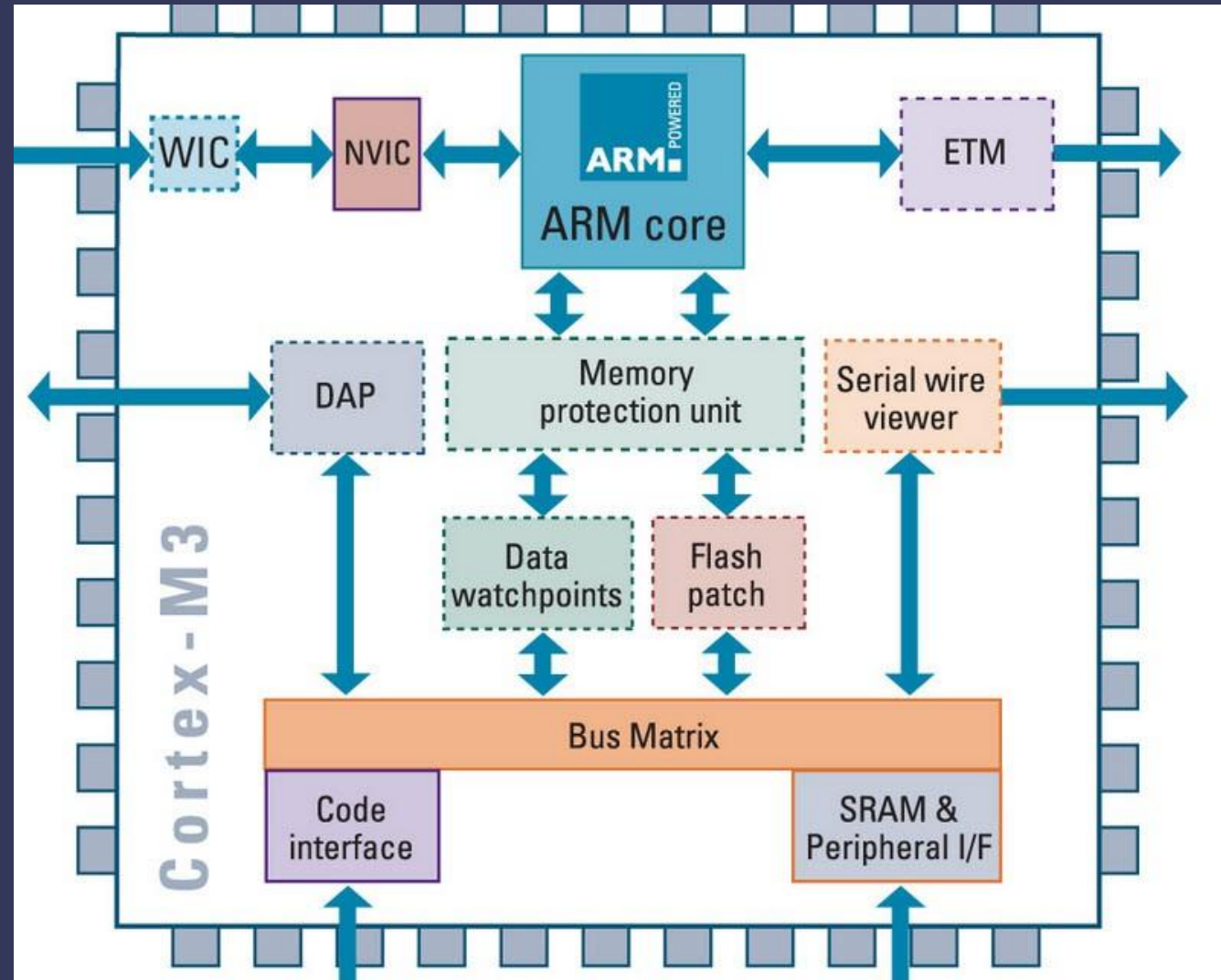
**ARM**

*Advanced RISC Machine*

# Microprocessador ARM

O microprocessador ARM (Advanced RISC Machine) é um dos núcleos de processador mais extensos e licenciados do mundo. O primeiro processador ARM foi desenvolvido no ano de 1978 pela Universidade de Cambridge, e o primeiro processador ARM RISC foi produzido pelo Grupo Acorn de Computadores no ano de 1985. Esses processadores são usados especificamente em dispositivos portáteis como câmeras digitais, telefones celulares, redes domésticas, módulos e tecnologias de comunicação sem fio e outros sistemas embarcados devido aos benefícios, como baixo consumo de energia, desempenho razoável, etc.

# Arquitetura de microprocessador ARM



# Microprocessador ARM

ARM, originalmente Acorn RISC Machine, e depois Advanced RISC Machine, é uma família de arquiteturas RISC desenvolvida pela empresa britânica ARM Holdings. Tais arquiteturas são licenciadas pela ARM para outras empresas, que implementam-nas em seus próprios produtos. A ARM também desenvolve chips que utilizam tal arquitetura e que são licenciados para uso exclusivo de outras empresas em seus produtos.

# Microprocessador ARM

Microprocessadores com uma arquitetura RISC em geral necessitam de menos transistores do que microprocessadores CISC, como os da arquitetura x86, comumente encontrada em computadores pessoais. Essa característica permite um consumo menor, custo menor, e dissipação de calor menor, o que faz dessa arquitetura algo desejado por fabricantes de dispositivos pequenos, portáteis, e movidos a bateria, como smartphones, laptops, e outros sistemas embarcados. Supercomputadores também podem se beneficiar dessas características, e alguns utilizam arquitetura ARM.

# Microprocessador ARM

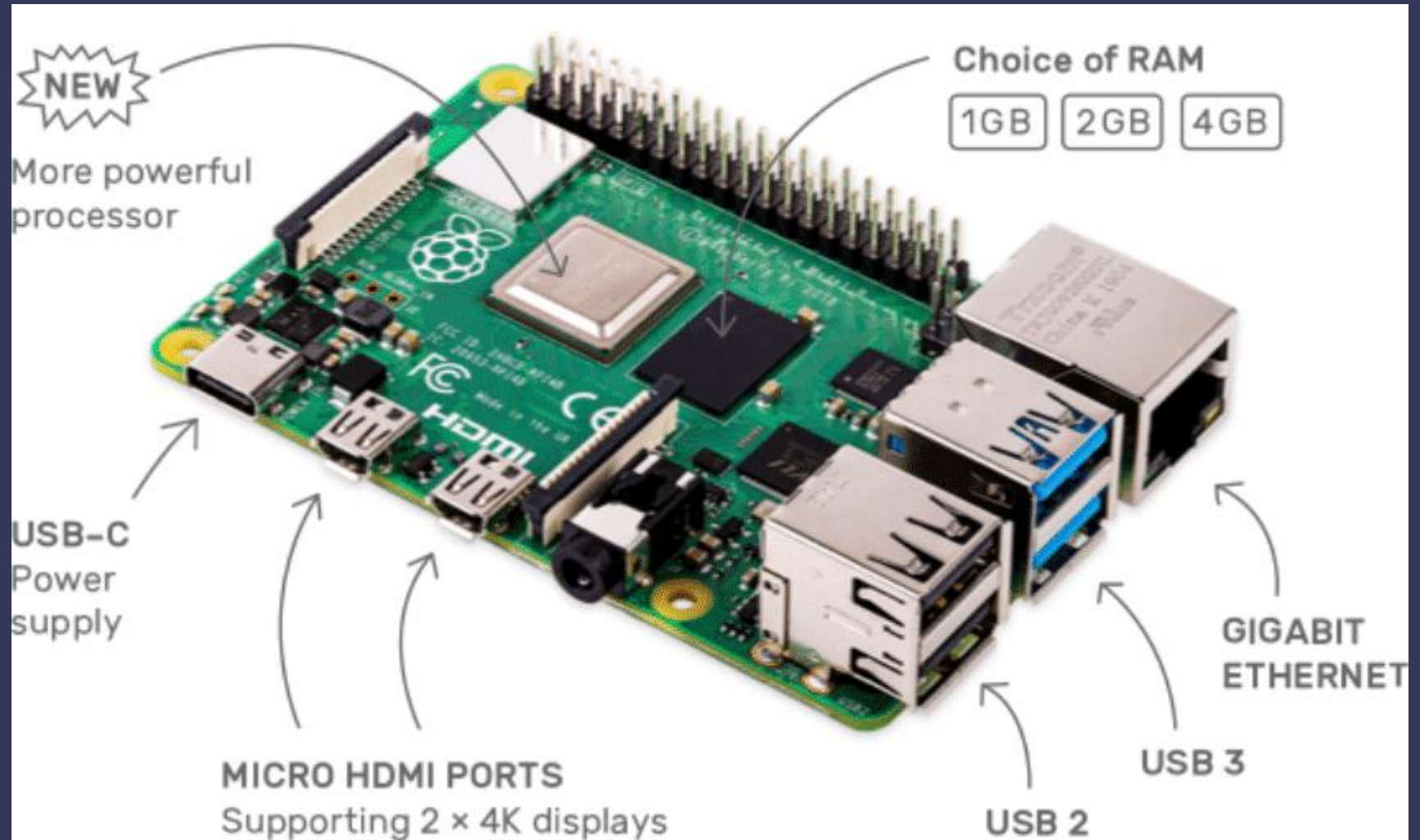
A ARM Holdings publica periodicamente atualizações para suas arquiteturas, todas elas com espaço de endereçamento de 32 bits e aritmética de 32 bits; instruções para chips da ARM possui tamanho fixo de 32 bits, mas versões mais recentes também suportam instruções de tamanho variável para melhorar a densidade de código. Alguns núcleos também providenciam a execução em hardware de bytecode Java. A arquitetura ARMv8-A, anunciada em Outubro de 2011, suporta endereçamento de 64 bits e aritmética de 64 bits, com instruções de tamanho fixo de 32 bits.

Com 100 bilhões de cores produzidos até 2017, ARM é a arquitetura mais popular em termos de quantidade.

# Raspberry Pi 4: ARM Cortex A72 64 bits

## Raspberry Pi 4

RAM: 1GB 2GB 4GB  
Chipset: ARM Cortex A72  
Speed: 1.5GHz  
Arch: 64-bit quad core



# Referências

- **COMPUTER History Museum.** Microprocessors flourish. Disponível em: <<https://www.computerhistory.org/revolution/story/284>>
- **NERYS, J. W. L.** Microprocessadores. Escola de Engenharia Elétrica, Mecânica e de Computação, Universidade Federal de Goiás, 2016. Disponível em: <<http://www.emc.ufg.br/~jwilson>>
- **WATELECTRONICS.** What is RISC and CISC architecture with advantages and disadvantages. Disponível em: <<https://www.watelectronics.com/what-is-risc-and-cisc-architecture/>>