

Instituto Federal de Minas Gerais
Campus Ouro Branco

Evolução e desempenho do computador

Professor: Saulo Henrique Cabral Silva

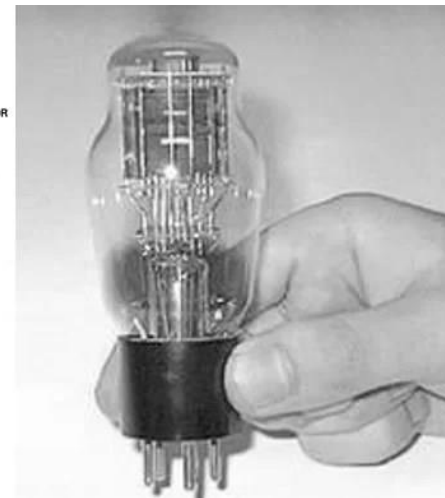
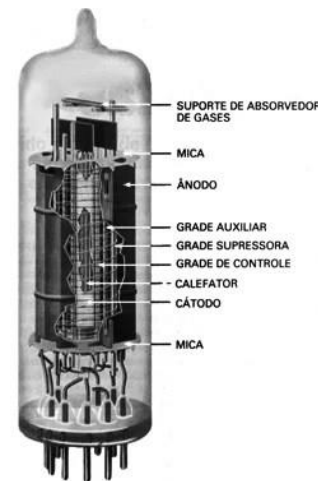
Principais Pontos

- A evolução dos computadores tem sido caracterizada pelo:
 - Aumento na **velocidade do processador**;
 - Redução no **tamanho dos componentes**;
 - Aumento no **tamanho da memória**;
 - Aumento na **capacidade e velocidade da E/S**.
- A redução no tamanho dos componentes do microprocessador **reduz a distância entre os componentes**.
- A organização do processador inclui:
 - Pipelines
 - Execução paralela
 - Execução especulativa
- É preciso **equilibrar** para que os **ganhos** de uma área não sejam **prejudicados** por atrasos de outra.
 - Memória **cache**
 - Caminhos de dados **mais largos**.



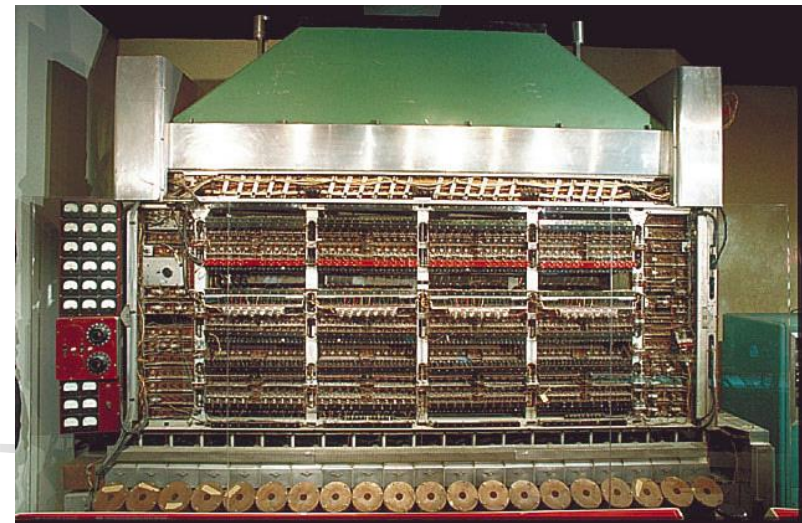
Breve Histórico

- **A primeira geração: Válvulas**
 - **ENIAC** (Electronic Numerical Integrator And Computer)
 - Desenvolvido na Universidade da Pensilvânia
 - Primeiro Computador Digital eletrônico de uso geral do mundo.
 - Atender as **necessidades dos EUA** durante a Segunda Guerra Mundial
- **Ballistics Research Laboratory (BRL)** do Exército, uma agência responsável por **desenvolver tabelas de faixa e trajetória** para novas armas.
 - Havia dificuldade para **fornecer essas tabelas com precisão** e dentro de um espaço de tempo razoável.



Breve Histórico

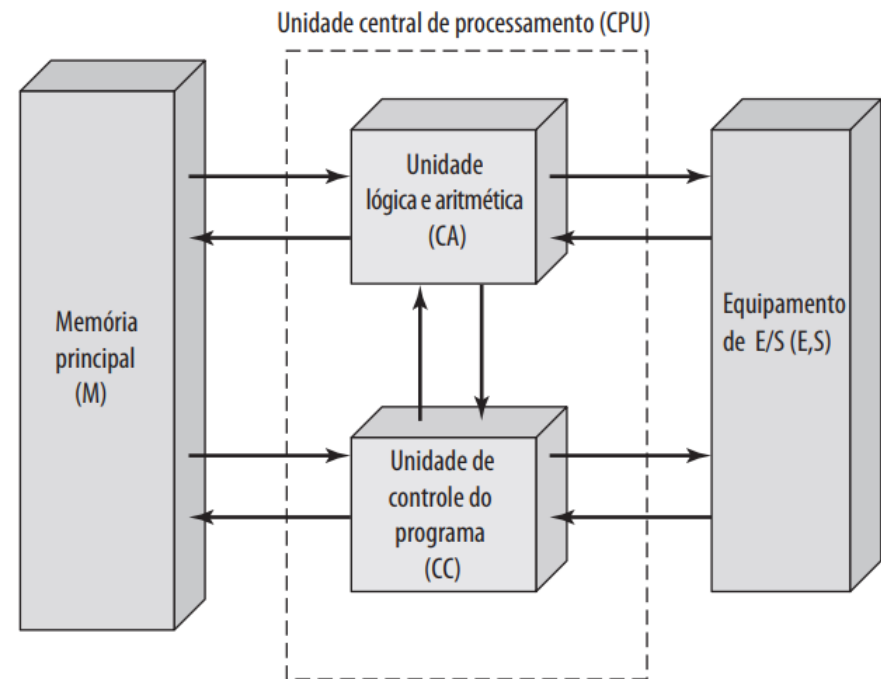
- Máquina de Von Neumann
 - Resolve o problema de **alterar programas** para o ENIAC.
 - Programa representado em um **forma adequada para armazenamento na memória** junto com os dados.
 - Computador poderia obter suas instruções **lendo-as da memória**
 - Um programa pode ser **criado ou alterado** definindo-se os valores de uma parte da memória.
 - Conceito de **Programa Armazenado**.
- Von Neumann publicou a ideia como EDVAC (Electronic Discrete Variable Computer) em 1945.
- Em 1946, Von Neumann iniciaram o projeto de um novo computador de programas armazenados (IAS).
 - Embora não concluído antes de 1952, é o **protótipo de todos os computadores** de uso geral.



Breve Histórico

- Estrutura do IAS

- Uma **memória principal**, que armazena dados e instruções
- Uma **unidade lógica e aritmética (ALU)** capaz de operar sobre dados binários.
- Uma **unidade de controle**, que interpreta as instruções na memória e faz com que sejam executadas.
- Equipamento de **entrada e saída (E/S)** operado pela unidade de controle.



Estrutura Von Neumann

- **Primeiro:** Realizar as **operações elementares** da aritmética mais frequentemente.
 - Adição, Subtração, Multiplicação e Divisão
 - Unidade aritmética
- **Segundo:** Controle lógico do dispositivo, sequenciação apropriada das operações que possam ser executadas por um órgão de controle central.
 - Se for flexível (propósito geral), então deve-se distinguir entre as instâncias específicas .
- **Terceiro:** Qualquer dispositivo que tiver que **executar sequências de operações longas**, precisa ter uma **memória considerável**.

Estrutura Von Neumann

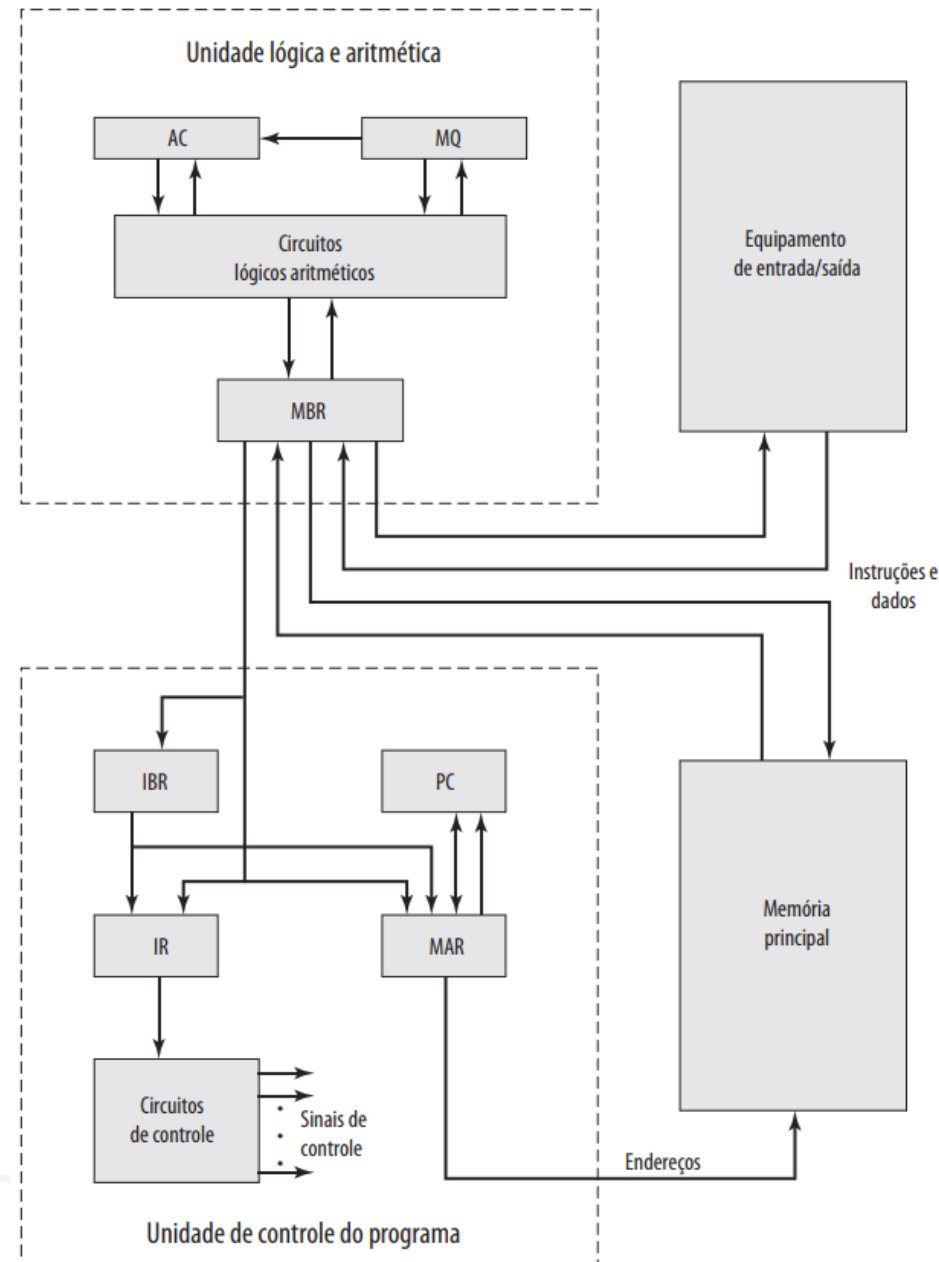
- **Quarto:** Dispositivo precisa ter unidades para transferir informações de E/S para suas partes específicas C e M.
 - Unidades formam sua entrada (I - input)
- **Quinto:** Dispositivo precisa ter unidades para transferir de C e M para E/S.
 - Unidades forma sua saída (o de output)
- **Importante:** Com raras exceções, todos os computadores de hoje têm essa **estrutura e função geral**, e portanto, são conhecidas como máquinas de Von Neumann.

Operação do computador IAS(Hayes)

- A memória do **IAS** consiste em **1000 locais** de armazenamento (*words*).
 - 40 dígitos binários
 - **Armazena dados e instruções.**
- Cada número é representado por um **bit de sinal e um valor de 39 bits.**
- Uma palavra pode conter duas instruções de 20 bits, com cada instrução consistindo em:
 - Código de operação de **8 bits (opcode)**
 - **Endereço de 12 bits**, designando uma das palavras da memória (**0 a 999**)

Operação do computador IAS(Hayes)

- A **unidade de controle** opera o IAS buscando instruções da memória e executando-as uma de cada vez.
 - **Registrador de buffer de memória (MBR)**
 - **Registrador de endereço de memória(MAR)**
 - **Registrador de instrução (IR)**
 - **Registrador de buffer de instrução (IBR)**
 - **Contador de programa (PC)**
 - **Acumulador (AC) e quociente multiplicador (MQ)**



Registradores

Memory buffer register (MBR)

- Contém a palavra a ser armazenada na memória ou enviada para unidade de I/O
- Ou utilizada para receber uma palavra da memória ou da unidade de I/O

Memory address register (MAR)

- Especifica o endereço da memória em que uma palavra deve ser lida ou escrita na MBR

Instruction register (IR)

- Contém os 8-bits referentes a instrução opcode que deve ser executada

Instruction buffer register (IBR)

- Retêm temporariamente a instrução da mão direita de uma palavra na memória

Program counter (PC)

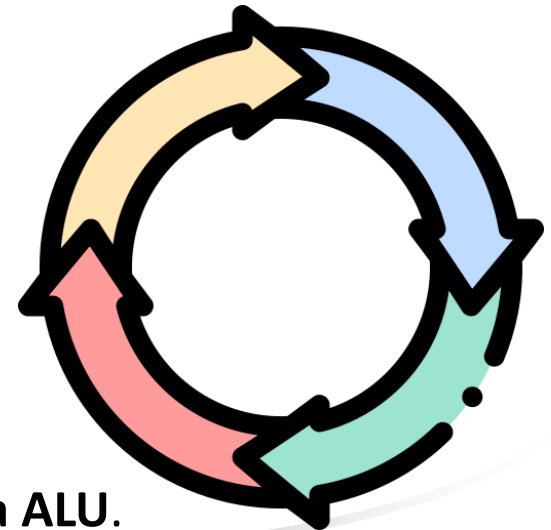
- Contém o endereço da próxima instrução a ser buscada da memória

Accumulator (AC) and multiplier quotient (MQ)

- Armazena temporariamente os operadores e resultados temporários das operações

Operação do computador IAS(Hayes)

- O IAS opera **repetidamente um ciclo de instrução**.
 - Cada ciclo de instrução consiste em **dois subciclos**
 - Durante o **ciclo de busca** (*fetch cycle*), o opcode da **próxima instrução** é **carregado no IR** e a parte de **endereço** é **carregada no MAR**.
 - **A instrução pode ser retirada do IBR** ou pode ser obtida da memória carregando-se uma palavra no MBR, e depois para o IBR, IR e MAR.
- Quando o **opcode está no IR**, o ciclo de execução é realizado.
 - O circuito **de controle interpreta o opcode** e **executa a instrução** enviando os sinais de controle apropriados para que os dados movidos ou uma **operação seja realizada pela ALU**.



Instruções IAS

- IAS tinha um total de **21 instruções**, agrupadas da seguinte forma:
 - **Transferência de dados:** movem dados entre memória e registradores da ALU ou entre dois registradores da ALU.
 - **Desvio incondicional:** normalmente, a unidade de controle executa instruções em sequência a partir da memória.
 - Essa sequência pode ser alterada por uma instrução de desvio, que facilita **operações repetitivas**.
 - **Desvio condicional:** o desvio pode se tornar dependente de uma condição, permitindo assim pontos de decisão.
 - **Aritméticas:** operações realizadas pela ALU.
 - **Modificação de endereço:** permite que os endereços sejam calculados na ALU e depois inseridos em instruções armazenadas na memória.
 - Isso permite a um programa uma flexibilidade de endereçamento considerável

Tipo de instrução	Opcod	Representação simbólica	Descrição
Transferência de dados	00001010	LOAD MQ	Transfere o conteúdo de MQ para AC
	00001001	LOAD MQ,M(X)	Transfere o conteúdo do local de memória X para MQ
	00100001	STOR M(X)	Transfere o conteúdo de AC para o local de memória X
	00000001	LOAD M(X)	Transfere M(X) para o AC
	00000010	LOAD – M(X)	Transfere – M(X) para o AC
	00000011	LOAD M(X)	Transfere o valor absoluto de M(X) para o AC
	00000100	LOAD – M(X)	Transfere - M(X) para o acumulador
Desvio incondicional	00001101	JUMP M(X,0:19)	Apanha a próxima instrução da metade esquerda de M(X)
	00001110	JUMP M(X,20:39)	Apanha a próxima instrução da metade direita de M(X)
Desvio condicional	00001111	JUMP+ M(X,0:19)	Se o número no AC for não negativo, apanha a próxima instrução da metade esquerda de M(X)
	00010000	JUMP+ M(X,20:39)	Se o número no AC for não negativo, apanha a próxima instrução da metade direita de M(X)
Aritmética	00000101	ADD M(X)	Soma M(X) a AC; coloca o resultado em AC
	00000111	ADD M(X)	Soma M(X) a AC; coloca o resultado em AC
	00000110	SUB M(X)	Subtrai M(X) de AC; coloca o resultado em AC
	00001000	SUB M(X)	Subtrai M(X) de AC; coloca o resto em AC
	00001011	MUL M(X)	Multiplica M(X) por MQ; coloca os bits mais significativos do resultado em AC; coloca bits menos significativos em MQ
	00001100	DIV M(X)	Divide AC por M(X); coloca o quociente em MQ e o resto em AC
	00010100	LSH	Multiplica o AC por 2; ou seja, desloca à esquerda uma posição de bit
	00010101	RSH	Divide o AC por 2; ou seja, desloca uma posição à direita
Modificação de endereço	00010010	STOR M(X,8:19)	Substitui campo de endereço da esquerda em M(X) por 12 bits mais à direita de AC
	00010011	STOR M(X,28:39)	Substitui campo de endereço da direita em M(X) por 12 bits mais à direita de AC

A SEGUNDA GERAÇÃO: TRANSISTORES

A Segunda geração: transistores

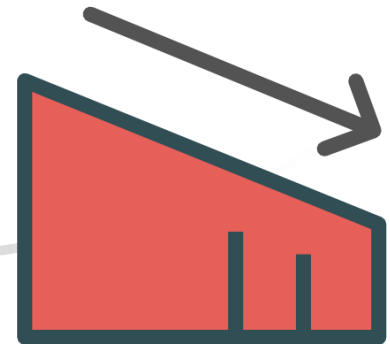
- A primeira mudança importante no computador eletrônico veio com a **substituição da válvula pelo transistor**.
 - O transistor é **menor, mais barato e dissipa menos calor** que uma válvula, e pode ser usado da mesma forma que uma válvula para construir computadores.
 - Diferente da válvula, que exige fios, placas de metal, uma cápsula de vidro e um vácuo, o transistor é um dispositivo de estado sólido, feito de silício.
- A segunda geração viu a introdução de **unidades lógicas e aritméticas e unidades de controle mais complexas**.
 - O uso de linguagens de programação de alto nível e a disponibilidade do software de sistema com o computador.

Geração	Datas aproximadas	Tecnologia	Velocidade típica (operações por segundo)
1	1946 – 1957	Válvula	40.000
2	1958 – 1964	Transistor	200.000
3	1965 – 1971	Integração em escala pequena e média	1.000.000
4	1972 – 1977	Integração em escala grande	10.000.000
5	1978 – 1991	Integração em escala muito grande	100.000.000
6	1991-	Integração em escala ultragrande	1.000.000.000

A TERCEIRA GERAÇÃO: CIRCUITOS INTEGRADOS

A terceira geração: circuitos integrados

- Circuito integrado
 - A **segunda geração** continha cerca de 10.000 transistores
 - O número cresceu para **centenas e milhares, tornando a manufatura de máquinas mais poderosas cada vez mais difícil.**
 - O circuito integrado explora o fato de que componentes como transistores, resistores e condutores podem ser fabricados a partir de um semicondutor como o silício.
 - O computador é um conjunto de portas e células.
- Microeletrônica: Tendência persistente e consistente em direção à **redução o tamanho dos circuitos eletrônicos digitais.**



Relacionamento entre *Wafer*, Chip e Porta

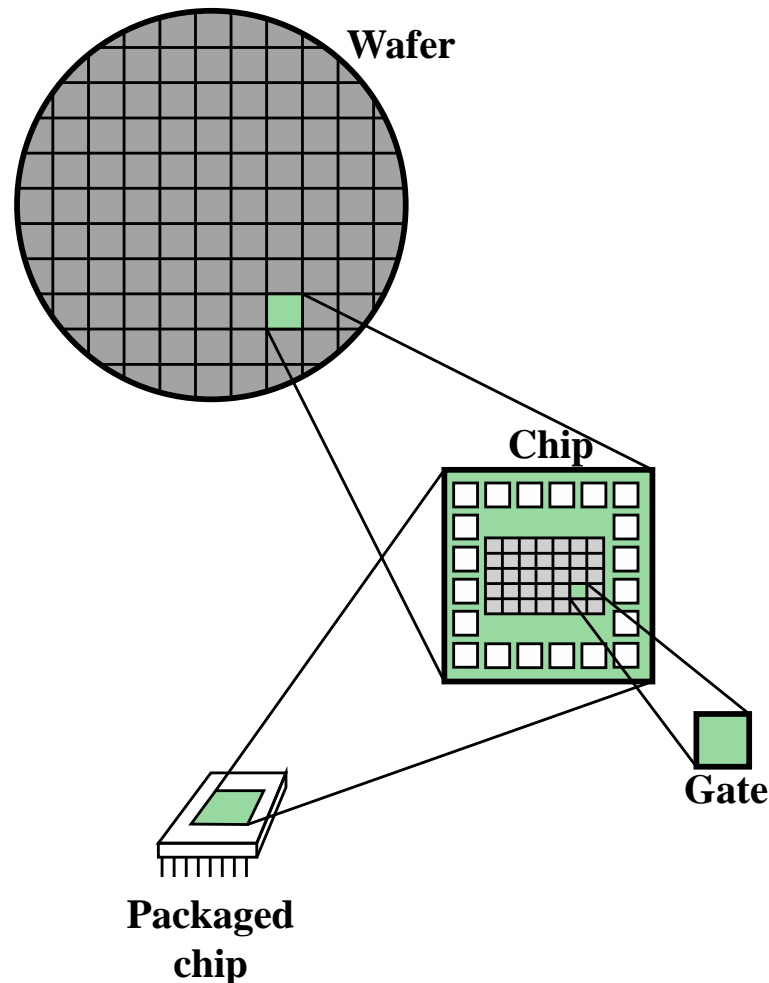


Figure 1.11 Relationship Among Wafer, Chip, and Gate

Circuito integrado

- SSI → Small-Scale Integration
 - Inicialmente **somente algumas portas** ou células poderiam ser empacotadas.
 - Com o passar do tempo mais e mais componentes foram empacotados no mesmo chip
- Lei de Moore
 - Gordon Moore cofundador da Intel
 - Número de transistores **dobrava a cada ano.**
 - O ritmo diminuiu a partir de **1970** (18 meses)



Gordon Moore, cofundador da Intel e 'profeta' da tecnologia, morre aos 94 anos

Além de fundar empresas de sucesso no ramo da tecnologia, Moore fez previsão que se tornou 'lei' e estimulou revolução dos computadores pessoais e smartphones.

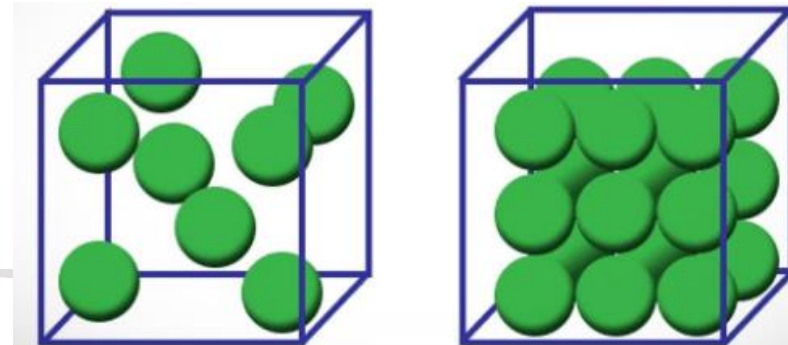
Por g1

24/03/2023 21h20 · Atualizado há uma semana



Consequências de Moore

- O **tamanho** de um chip permaneceu praticamente inalterado durante esse período de rápido crescimento em **densidade**.
- Elementos são **colocados muito próximos** (chips mais densamente empacotados)
- Computador **menor**
- Redução nos requisitos de **potência e resfriamento**
- **Interconexões** do circuito integrado são **mais confiáveis** do que as conexões com solda



FAMÍLIA DE COMPUTADORES

Família de computadores

- Computadores compatíveis, cliente **pode atualizar para uma máquina mais rápida com mais memória**, sem sacrificar o investimento no software já desenvolvido.
 - Conjunto de instruções **semelhante ou idêntico**: mesmo conjunto de instruções de máquina é aceito em todos os membros da família.
 - Sistema operacional **semelhante ou idêntico**: Mesmo Sistema operacional está disponível para todos os membros da família.
 - **Velocidade aumentada**: taxa de execução de instruções aumenta (do mais baixo ao mais alto)
 - Número de portas de E/S maior
 - Tamanho de memória crescente: tamanho da **memória principal aumenta**
 - Maior custo: em um dado tempo t , sistema aumenta em relação aos membros mais baixos até os mais altos.
 - Em um tempo t_1 , ocorre redução



GERAÇÕES POSTERIORES

Gerações posteriores

- Após a terceira geração, existe **pouco consenso** sobre uma definição.
 - **LSI** → large-scale integration, mais de 1.000 componentes colocados em um único chip de circuito integrado.
 - **VLSI** → very-large-scale integration alcançou mais de 10.000 componentes por chip
 - **ULSI** → ultra-large-scale integration, pode conter mais de um milhão de componentes.

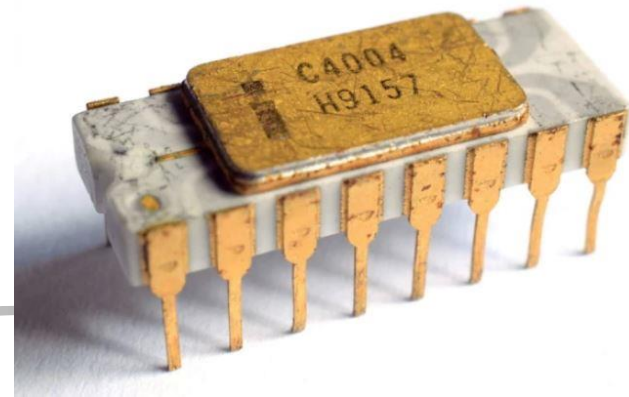
Memória Semicondutora

- Uso do circuito integrado para construção dos processadores.
- Possibilidade de construção de **memória utilizando a tecnologia de circuito integrado**.
 - Memórias de 1950 a 1960 eram construídas a partir de aneis de ferrite.
- Desde 1970 a memória semicondutora vem passando por várias gerações:
 - 1K, 4K, 16K, 64K, 256K, 1M, 4M, 16M, 64M, 256M, 1G, 4G, 16G
 - Cada geração fornecendo **quatro vezes mais densidade** de armazenamento.



Microprocessadores

- Assim como a densidade dos elementos nos chips de memória continuavam subindo, **a densidade dos elementos** dos chips do processador também subiram
 - Menos e menos **chips de diversos componentes** eram necessários para construir um único computador.
- 1971 a Intel desenvolveu o 4004, que **continha todos os componentes** de uma CPU
 - Microprocessador
 - Soma de valores com representação de 4 bits
 - Multiplicação com operações de adições repetidas.
 - Marca o início de uma evolução contínua de **capacidade e poder do microprocessador**.
- 1972 lançamento do 8008
 - Operações com 8 bits
 - Dobro de complexidade



1970's

	4004	8008	8080	8086	8088
Introduced	1971	1972	1974	1978	1979
Clock speeds	108 kHz	108 kHz	2 MHz	5 MHz, 8 MHz, 10 MHz	5 MHz, 8 MHz
Bus width	4 bits	8 bits	8 bits	16 bits	8 bits
Number of transistors	2,300	3,500	6,000	29,000	29,000
Feature size (µm)	10	8	6	3	6
Addressable memory	640 Bytes	16 KB	64 KB	1 MB	1 MB

1980's

	80286	386TM DX	386TM SX	486TM DX CPU
Introduced	1982	1985	1988	1989
Clock speeds	6 MHz - 12.5 MHz	16 MHz - 33 MHz	16 MHz - 33 MHz	25 MHz - 50 MHz
Bus width	16 bits	32 bits	16 bits	32 bits
Number of transistors	134,000	275,000	275,000	1.2 million
Feature size (µm)	1.5	1	1	0.8 - 1
Addressable memory	16 MB	4 GB	16 MB	4 GB
Virtual memory	1 GB	64 TB	64 TB	64 TB
Cache	—	—	—	8 kB

1990's

	486TM SX	Pentium	Pentium Pro	Pentium II
Introduced	1991	1993	1995	1997
Clock speeds	16 MHz - 33 MHz	60 MHz - 166 MHz,	150 MHz - 200 MHz	200 MHz - 300 MHz
Bus width	32 bits	32 bits	64 bits	64 bits
Number of transistors	1.185 million	3.1 million	5.5 million	7.5 million
Feature size (µm)	1	0.8	0.6	0.35
Addressable memory	4 GB	4 GB	64 GB	64 GB
Virtual memory	64 TB	64 TB	64 TB	64 TB
Cache	8 kB	8 kB	512 kB L1 and 1 MB L2	512 kB L2

2000's

	Pentium III	Pentium 4	Core 2 Duo	Core i7 EE 4960X
Introduced	1999	2000	2006	2013
Clock speeds	450 - 660 MHz	1.3 - 1.8 GHz	1.06 - 1.2 GHz	4 GHz
Bus width	64 bits	64 bits	64 bits	64 bits
Number of transistors	9.5 million	42 million	167 million	1.86 billion
Feature size (nm)	250	180	65	22
Addressable memory	64 GB	64 GB	64 GB	64 GB
Virtual memory	64 TB	64 TB	64 TB	64 TB
Cache	512 kB L2	256 kB L2	2 MB L2	1.5 MB L2/15 MB L3
Number of cores	1	1	2	6

PROJETANDO VISANDO AO DESEMPENHO

Velocidade do microprocessador

- Busca implacável de velocidade por parte dos fabricantes de chips de processadores.
 - Comprovação da lei de Moore, nova geração a cada 3 anos com **quatro vezes a quantidade de transistores**
 - Nos microprocessadores, a adição de novos circuitos, e o aumento de velocidade que vem **da redução das distâncias entre eles**, melhorou o desempenho de quatro a cinco vezes a cada três anos ou mais.
- **A velocidade bruta** do microprocessador não alcança seu potencial a menos que receba um **fluxo constante de trabalho**.
 - É necessário a aplicação de técnicas bem elaboradas para alimentar esse “monstro”

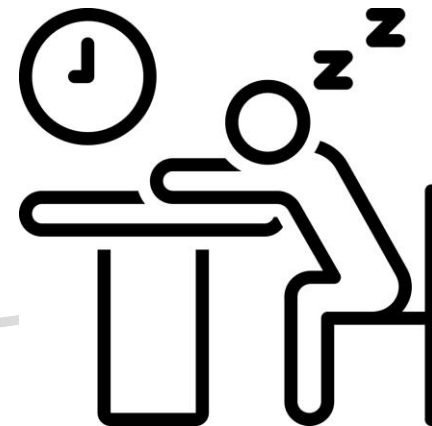
Técnicas embutidas nos processadores

- **Previsão de desvio:** Processador **antecipa o código da instrução** apanhando da memória e prevê quais desvios, ou grupos de instruções provavelmente serão utilizados em seguida.
 - Se houver acerto na maior parte do tempo, pode armazenar as informações em um buffer.
- **Análise de fluxo de dados:** O processador analisa quais instruções são dependentes dos resultados uma da outra, ou dos dados.
 - Cria uma sequência otimizada de instruções
 - Executando o que está sempre pronto
- **Execução especulativa:** Executam instruções antes mesmo do seu surgimento real na execução de um programa, mantendo os resultados em locais temporários.
 - Manter mecanismo de execução **sempre ocupado**.



Balanço do desempenho

- Necessidade de procurar **equilíbrio do desempenho**
 - Potência do processador X desempenho de outros componentes
- Problema crítico na interface do processador com a memória principal.
 - Embora a **velocidade do processador tenha aumentado** rapidamente, a velocidade com que os dados podem ser transferidos entre a memória principal e o processador ficou para trás.
 - Caminho crítico (gargalo)
- Se a memória ou o caminho deixar de manter o ritmo com as demandas insistentes do processador, este entra em um **estado de espera.**



Maneiras de um arquiteto de sistemas atacar esse problema

- Aumentar o **número de bits** que são recuperados ao mesmo tempo.
 - DRAM mais larga (caminhos mais largos)
- Alterar a **interface da DRAM** para torná-la mais eficiente
 - Uso de cache
 - Ou esquema de buffer no chip
- Reduzir a **frequência de acesso à memória**
 - Uso de estruturas de cache mais complexas e eficientes entre processador e memória.
 - Uso de mais de uma cache
- Aumentar a **largura de banda** de interconexão entre processador e memória
 - **Barramento de velocidade mais alta**



Dispositivos de entrada e saída

- Tratamento dos dispositivos de E/S.
 - Aplicações mais sofisticadas com demandas intensas de E/S.
 - Demanda de vazão de dados
 - Problemas na movimentação dos dados
 - Estratégias de **caching e buffering**
- Necessidade de equilibrar as **demandas de vazão e processamento dos componentes.**
 - O projeto precisa ser constantemente repensado.
 - Taxa de mudança nas diversas áreas da tecnologia (processador, barramento, memória e periféricos)
 - Novas aplicações e novos dispositivos periféricos

MELHORIAS NA ORGANIZAÇÃO E ARQUITETURA DO CHIP

Melhorias na organização e na arquitetura do chip

- Além de **balancear o desempenho do processador com o da memória principal**, ainda é necessário alocar esforços com o intuito de aumentar a velocidade do processador.
 - Aumentar a velocidade de hardware do processador: escolha do tamanho de portas lógicas, para que possam ser posicionadas mais de perto (aumentando o clock).
 - Aumento do clock, implica em executar operações individuais mais rápido
 - Aumentar tamanho e velocidade das **caches** entre processar e memória principal.
 - Modificar a organização e arquitetura: Aumentar a velocidade da execução da instrução.

Obstáculos para obter desempenho

- **Potência:** à medida que a densidade da lógica e a velocidade do clock em um chip aumentam, também aumenta a densidade de potência (Watts/cm²).
 - Dificuldade de **dissipar o calor** gerado em chips de **alta densidade e velocidade**
- **Atraso de RC:** A **velocidade em que os elétrons** podem fluir no chip entre os transistores é limitada pela resistência (R) e capacitância (C) dos fios de metal
 - A medida que os componentes no chip diminuem de tamanho, as interconexões de fio se tornam mais finas, aumentando a resistência e os fios estão mais próximos aumentando a capacitância
- **Latência da memória:** A velocidade de memória limitam as velocidades do processador.

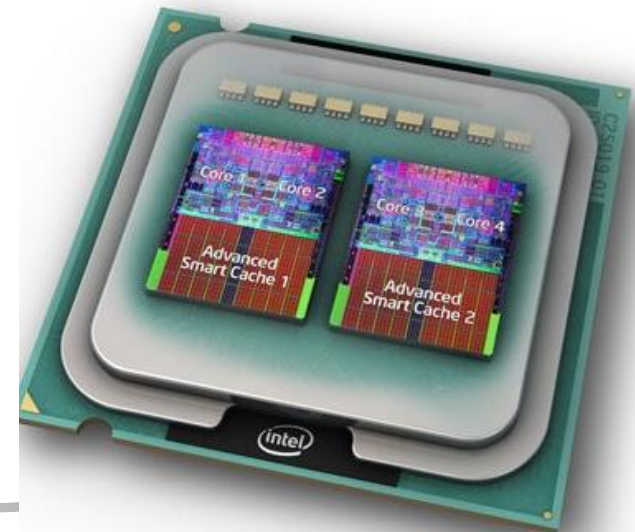


Duas estratégias utilizadas desde 1980

- Sem aumentar a velocidade do clock, duas estratégias são sistematicamente utilizadas para aumentar a velocidade:
 - **Aumento na capacidade** da cache: uso de dois ou três níveis de cache entre memória e processador.
 - A medida que a densidade do chip aumenta, mais da memória cache é incorporada ao chip.
 - Permite acesso mais rápido.
 - **A lógica de execução** de instrução dentro do processador está cada vez mais complexa.
 - Uso de **pipeline** e superescalar
 - Permite múltiplos pipelines dentro de um único processador
 - Instruções que não dependem umas das outras possam ser executadas em paralelo.

Multicore

- Uso de múltiplos processadores no **mesmo chip** (multicore)
- Cache **compartilhada**
- Aumento de **desempenho sem aumentar taxa de clock**
- Se o software puder suportar o uso efetivo de múltiplos processadores, é possível **quase dobrar o seu desempenho**.
- **Caches maiores** passam a ser justificadas.
- **Computação paralela!!!**



Questões

- Qual a principal contribuição da Máquina de Von Neumann no que diz respeito a evolução dos computadores?
 - Por quê?
- Cite os ganhos inerentes à transição do uso de válvulas para transistores.
 - Por quê?
- Até aqui você ainda acredita que o clock é a única medida de desempenho a ser observada em um processador?
 - Por quê?
- Como a evolução do desempenho dos computadores impactou outras tecnologias?

Dúvidas

