"O inventor, como a natureza de Linneu, não faz saltos; progride de manso, evolui" (Santos Dumont).

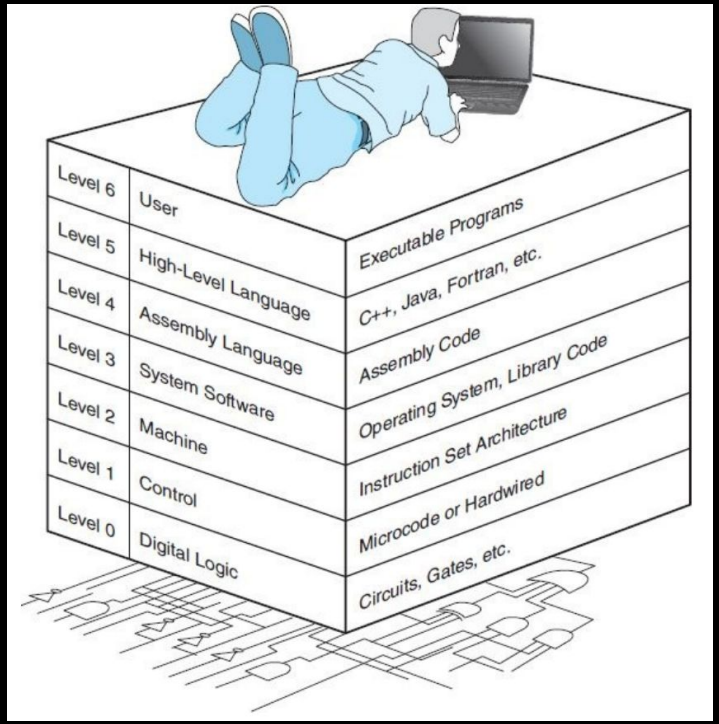
Arquiteturas e Abstrações

Paulo Lisboa de Almeida

DSBD

UFPR

Os níveis de abstração de um computador



Null, Lobur (2014)

Arquitetura de Von Neumann

A arquitetura de Von Neumann foi criada por ...

Arquitetura de Von Neumann

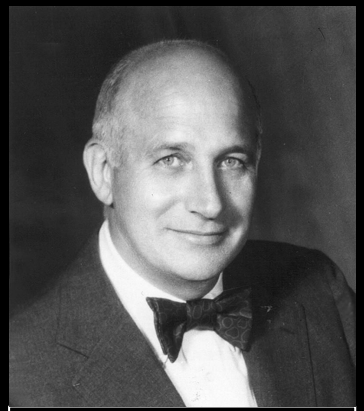
A arquitetura de Von Neumann foi criada por **John Mauchly** e **John Eckert**.

Criada enquanto eles trabalhavam no ENIAC.

A arquitetura seria empregada no sucessor do ENIAC, o EDVAC.

O projeto era secreto (WWII) e os pesquisadores não puderam publicar suas ideias.

John W. Mauchly
30/08/1907 - 08/01/1980
Físico Americano.
- ENIAC
- UNIVAC
- Mauchly's sphericity test
en.wikipedia.org/wiki/John_Mauchly



John Adam Presper Eckert Jr.
09/04/1919 - 03/06/1995
Engenheiro Eletricista Americano.
- ENIAC

https://en.wikipedia.org/wiki/J._Presper_Eckert

Arquitetura de Von Neumann

A arquitetura de Von Neumann foi criada por **John Mauchly** e **John Eckert**.

Criada enquanto eles trabalhavam no ENIAC.

A arquitetura seria empregada no sucessor do ENIAC, o EDVAC.

O projeto era secreto (WWII) e os pesquisadores não puderam publicar suas ideias.

John Von Neumann.

Matemático Húngaro.

Trabalhava em itens periféricos do projeto ENIAC.

Publicou e popularizou as ideias do EDVAC propostas por Mauchly e Eckert.

Foi um publicitário tão bom das ideias que creditaram a arquitetura em seu nome.

John von Neumann
28/12/1903 - 08/02/1957
Físico, Matemático e Cientista da Computação Húngaro.
- Abelian von Neumann algebra
- Affiliated operator
- Arithmetic logic unit
- Artificial life
- ...

en.wikipedia.org/wiki/John_von_Neumann#Illness_and_death

Arquitetura de Von Neumann

A arquitetura é famosa pelo conceito de “programa armazenado”.

Parece trivial, mas os primeiros programas de computadores eram “hardwired”.

Se precisar mudar o programa, mude o circuito!

Arquitetura de Von Neumann

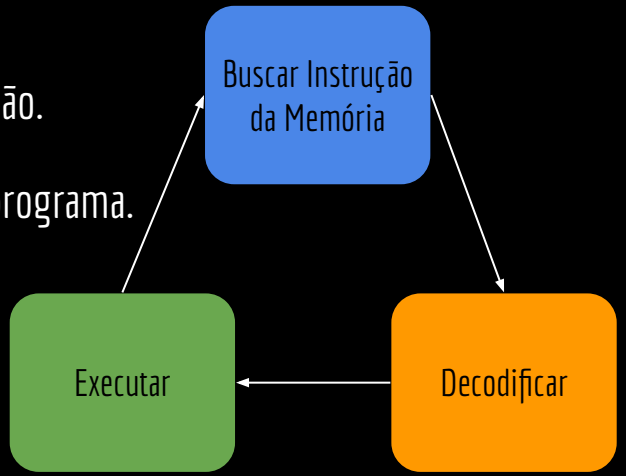
A arquitetura é famosa pelo conceito de “programa armazenado”.

Parece trivial, mas os primeiros programas de computadores eram “hardwired”.

Se precisar mudar o programa, mude o circuito!

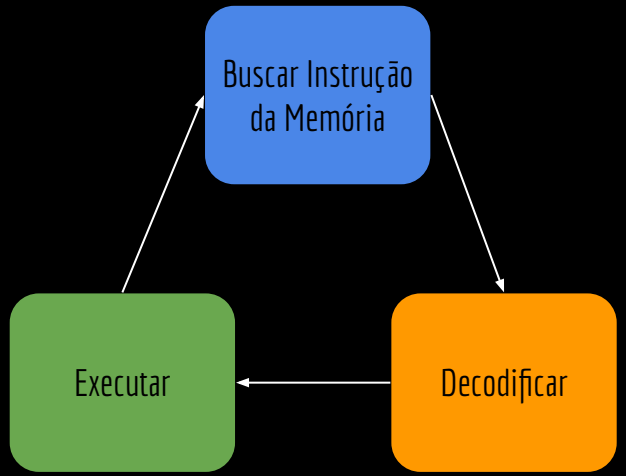
Os programas executam no ciclo de Von Neumann – Busca-decodificação-execução.

1. A CPU busca a próxima instrução da memória utilizando um contador de programa.
2. A instrução é decodificada.
3. Os operandos necessários são carregados.
4. A ALU executa a operação e o resultado é armazenado na memória.



Pergunta

Pergunta: a CPU RISC-V estudada até agora segue o ciclo de Von Neumann?



Versão Moderna da Arquitetura de Von Neumann

Arquitetura composta de:

- CPU com unidade de controle, ALU, registradores e contador de Programa.
- Uma memória principal que armazena o programa e seus dados.
- Sistema de E/S.

Capaz de executar instruções sequencialmente.

Um único caminho (lógico ou físico) até a memória principal.

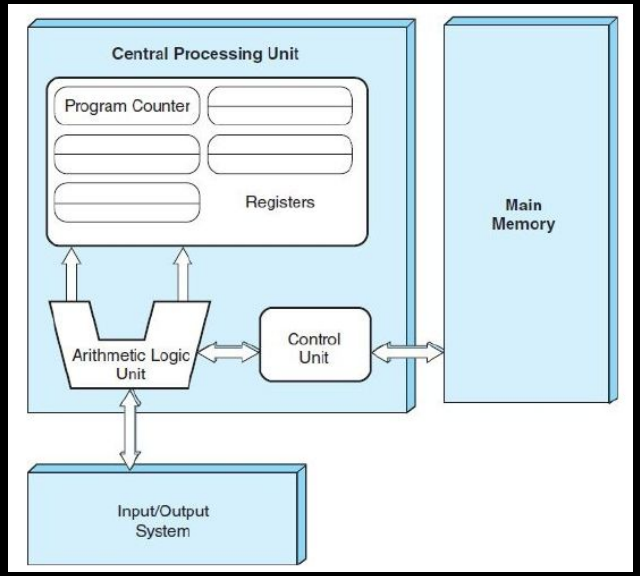
Versão Moderna da Arquitetura de Von Neumann

Arquitetura composta de:

- CPU com unidade de controle, ALU, registradores e contador de Programa.
- Uma memória principal que armazena o programa e seus dados.
- Sistema de E/S.

Capaz de executar instruções sequencialmente.

Um único caminho (lógico ou físico) até a memória principal.

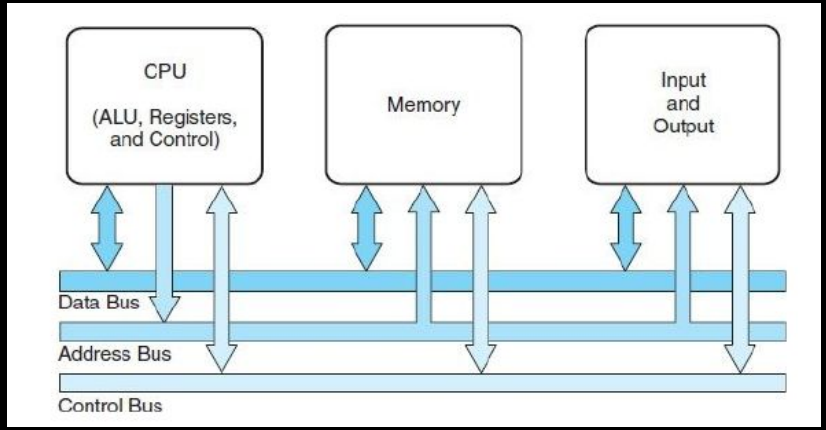


Null, Lobur (2014)

Versão Moderna da Arquitetura de Von Neumann

Versão modificada e mais próxima de uma utilizada em uma CPU real.

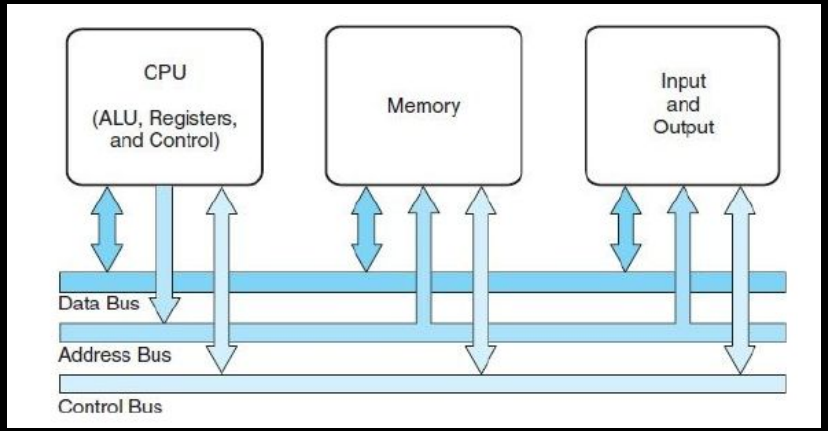
Apesar das modificações no diagrama, ainda é uma arquitetura de Von Neumann.



Null, Lobur (2014)

Pergunta

O processador RISC-V visto em aula segue a **Arquitetura** de Von Neumann?



Null, Lobur (2014)

Pergunta

O processador RISC-V visto em aula segue a **Arquitetura** de Von Neumann?

- Não exatamente.
- Temos duas memórias separadas.
  - Uma para instruções e uma para dados.

Pergunta

O processador RISC-V visto em aula segue a **Arquitetura** de Von Neumann?

- Não exatamente.
- Temos duas memórias separadas.
  - Uma para instruções e uma para dados.

A maioria das CPUs modernas **não seguem a arquitetura de Von Neumann de maneira estrita.**

- São variantes dessa arquitetura.
- Muitos autores argumentam que as mudanças são tão pequenas que na verdade essas CPUs podem ser consideradas como pertencendo à arquitetura de Von Neumann.

Arquitetura de Von Neumann

Na arquitetura de Von Neumann temos **uma memória principal** que armazena o programa e seus dados.

Problema?

Arquitetura de Von Neumann

Na arquitetura de Von Neumann temos **uma memória principal** que armazena o programa e seus dados.

Problema?

Gera um gargalo, conhecido como ***Von Neumann bottleneck*** (Gargalo de Von Neumann).

No pipeline da CPU RISC-V o problema fica claro.

Memória única geraria um hazard estrutural e, no mínimo, precisaríamos injetar algumas bolhas.

Arquitetura Harvard

Temos memórias para dados e instruções **separadas**.

- Como no processador RISC-V.

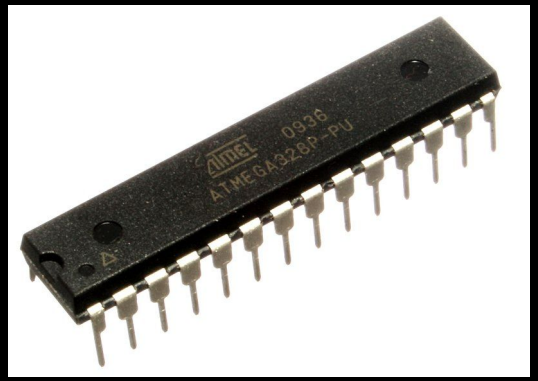
Máquinas com **Arquitetura Harvard Pura**.

- Seguem estritamente esse conceito.
- Comum em microcontroladores.

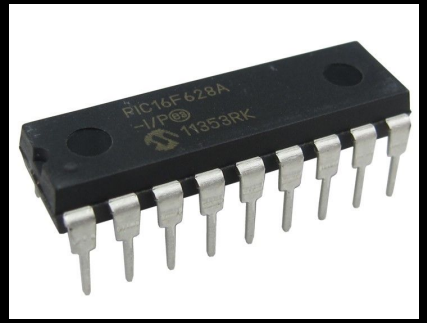
Um microcontrolador é um “computador completo” em um chip.

- Inclui memória, controladores de I/O, armazenamento de dados, processamento, etc.

Arquitetura Harvard



ATmega328P



PIC16F628a

Device	Program Memory	Data Memory	
	FLASH (words)	SRAM (bytes)	EEPROM (bytes)
PIC16F627A	1024	224	128
PIC16F628A	2048	224	128

begin with, the PIC16F627A/628A/648A uses a Harvard architecture, in which program and data are accessed from separate memories using separate busses. This improves bandwidth over traditional von Neu-

In order to maximize performance and parallelism, the AVR uses a harvard architecture – with separate memories and buses for program and data. Instructions in the program memory are executed with a single level pipelining. While one

Arquitetura Harvard

Máquinas com **Arquitetura Harvard Modificada**.

- Relaxa a separação física entre a memória de dados e instruções.
- Podemos encaixar a maioria dos PCs modernos (como os x86-64) nessa arquitetura.

Nos níveis de memória próximos ao processador, temos uma Arquitetura de Harvard.

- Temos memórias cache separadas para dados e instruções.

Em níveis de memória distantes do processador, temos uma máquina de Von Neumann.

- Os dados são acessados por um único barramento até a memória.

Faça você mesmo

Em um computador Linux, execute *lscpu* para informações detalhadas sobre sua CPU.

Exemplo em minha máquina.

L1d: cache de dados.

L1i: cache de instruções.

```
Arquitetura:                         x86_64
Modo(s) operacional da CPU:          32-bit, 64-bit
Ordem dos bytes:                     Little Endian
Address sizes:                       48 bits physical, 48 bits virtual
CPU(s):                              16
Lista de CPU(s) on-line:             0-15
Thread(s) per núcleo:                2
Núcleo(s) por soquete:               8
Soquete(s):                          1
Nó(s) de NUMA:                       1
ID de fornecedor:                    AuthenticAMD
Família da CPU:                      25
Modelo:                              33
Nome do modelo:                      AMD Ryzen 7 5800X 8-Core
Processor
…
cache de L1d:                        256 KiB
cache de L1i:                        256 KiB
cache de L2:                         4 MiB
cache de L3:                         32 MiB
…
```

Arquiteturas Paralelas

Computadores Paralelos.

- Desde seu computador pessoal até supercomputadores.

Existem diversos níveis de paralelismo.

- Pipelining, processadores superescalares, processadores multicore, multiprocessadores, grids, clusters, ...

Vamos focar em multicore como exemplo.

- Nome um tanto mercadológico, mas serve como um bom exemplo.

Processadores Multicore

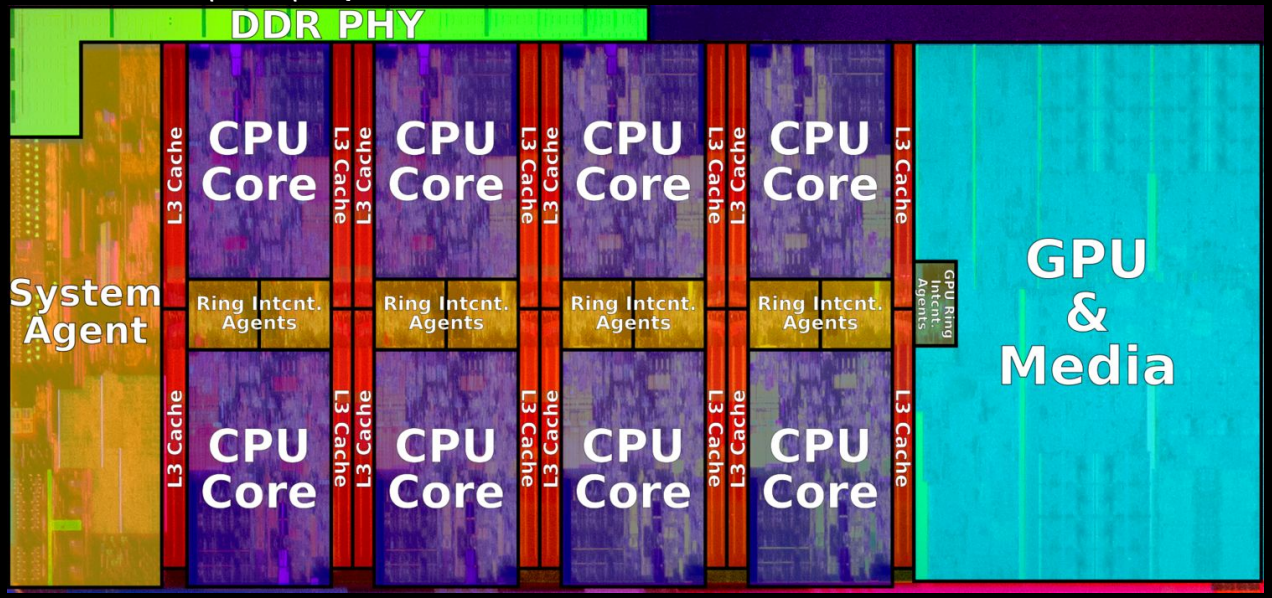
Um processador multicore (com múltiplos núcleos) possui vários núcleos de processamento (CPUs) em um único chip.

Todos os processadores acessam a mesma memória principal **por um mesmo barramento.**

Máquina **UMA.**

Uniform Memory Access.

Comum em computadores pessoais.



Intel i9 9900k - en.wikichip.org/wiki/intel/core_i9/i9-9900k

Processadores Multicore

Por serem **paralelos**, temos múltiplas instruções sendo executadas por diferentes CPUs, **ao mesmo tempo**.

Sendo assim, muitos os classificam como não sendo máquinas de Von Neumann.

Não processam as instruções de forma sequencial.

Outros veem essas CPUs como pertencendo a arquitetura de Von Neumann.

Cada CPU ("core") é uma máquina de Von Neumann, cooperando para executar as tarefas.

Segundo Null, Lobur (2014) podemos ver uma CPU moderna como uma máquina de Von Neumann que apresenta alguns aspectos de "*non-von Neumannness*".

Algo como "não Von Neumoniosidade!!??"

Misturando Tudo

Podemos ver uma CPU x86-64 atual como sendo:

Dependendo do nível de memória.

- Arquitetura Harvard.
- Arquitetura de Von Neumann.

Dependendo dos seus gostos.

- Arquitetura de Von Neumann.
- Uma arquitetura separada, se encaixando exclusivamente em arquiteturas paralelas.

RISC versus CISC

Alguns se referem a RISC e CISC como Arquiteturas.

Na verdade são mais para “estilos de conjuntos de instrução”.

Ou melhor, são estilos de ISA – Instruction Set Architecture.

CISC

CISC - Complex Instruction Set Computer.

CISC geralmente:

- Contém um número (muito) grande de instruções.
- Instruções de tamanhos variados.
- Instruções complexas, que podem executar “múltiplas coisas” em uma única operação.
- Exemplo -› uma única operação:
    - Busca o dado da memória;
    - Realiza uma operação com o dado;
    - Armazena o resultado na memória.

RISC

RISC -Reduced Instruction Set Computer.

RISC geralmente:

- Possui um conjunto de instruções reduzido e simplificado.
- Instruções são menos “poderosas”.
  - Simplificar o projeto e criar instruções que executam mais rapidamente.
- Poucos formatos de instruções.
- Instruções de tamanho fixo.

RISC versus CISC

O processador RISC-V é RISC ou CISC?

RISC versus CISC

O processador RISC-V é RISC ou CISC?

RISC.

Tabela Comparativa

Atenção: analise a tabela de maneira **crítica**. Essas n**ão são regras**, mas sim diferenças comumente encontradas em processadores RISC e CISC.

RISC	CISC
Instruções de tamanho fixo.	Instruções de tamanho variado.
Muitos Registradores.	Poucos Registradores.
Instruções de 3 operandos.	Instruções com 1 ou 2 operandos.
Parâmetros passados via registrador.	Parâmetros passados via pilha.
Controle hardwired.	Controle microprogramado.
Pipeline Profundo e simplificado.	Pipeline raso e mais complexo.
Poucas instruções simples.	Muitas instruções sofisticadas e complexas.
Somente loads e stores acessam a memória.	Muitas instruções podem acessar a memória.

RISC versus CISC

Processor	ARM A8	Intel Core i7 920
Market	Personal Mobile Device	Server, Cloud
Thermal design power	2 Watts	130 Watts
Clock rate	1 GHz	2.66 GHz
Cores/Chip	1	4
Floating point?	No	Yes
Multiple Issue?	Dynamic	Dynamic
Peak instructions/clock cycle	2	4
Pipeline Stages	14	14
Pipeline schedule	Static In-order	Dynamic Out-of-order with Speculation
Branch prediction	2-level	2-level
1st level caches / core	32 KiB I, 32 KiB D	32 KiB I, 32 KiB D
2nd level cache / core	128 - 1024 KiB	256 KiB
3rd level cache (shared)	--	2 - 8 MiB

CISC

O seu x86-64 é CISC.

CISC

O seu x86-64 é CISC.

Mentira!

x86-64

Construímos um pipeline simples na CPU RISC-V.

Imagine construir isso em uma CPU CISC!

Quais são as complexidades extras?

x86-64

Construímos um pipeline simples na CPU RISC-V.

Imagine construir isso em uma CPU CISC!

Quais são as complexidades extras?

- Instruções de tamanhos variados;
- Uma infinidade de instruções diferentes;
- Diversos formatos de instrução;
- Muitas instruções capazes de acessar a memória;
- ...

x86-64

Adicione diversos outros conceitos que complicam o hardware.

- Despacho múltiplo dinâmico.

x86-64

Adicione diversos outros conceitos que complicam o hardware.

- Despacho múltiplo dinâmico.
- Multithreading simultâneo (A Intel chama de Hyper-threading).

x86-64

Adicione diversos outros conceitos que complicam o hardware.

- Despacho múltiplo dinâmico.
- Multithreading simultâneo (A Intel chama de Hyper-threading).
- Instruções com quantidade de ciclos de clock variável.

x86-64

Adicione diversos outros conceitos que complicam o hardware.

- Despacho múltiplo dinâmico.
- Multithreading simultâneo (A Intel chama de Hyper-threading).
- Instruções com quantidade de ciclos de clock variável.
- Buffers de reordenação de instruções.

x86-64

Adicione diversos outros conceitos que complicam o hardware.

- Despacho múltiplo dinâmico.
- Multithreading simultâneo (A Intel chama de Hyper-threading).
- Instruções com quantidade de ciclos de clock variável.
- Buffers de reordenação de instruções.
- Predição dinâmica de desvios com tabelas dinâmicas.

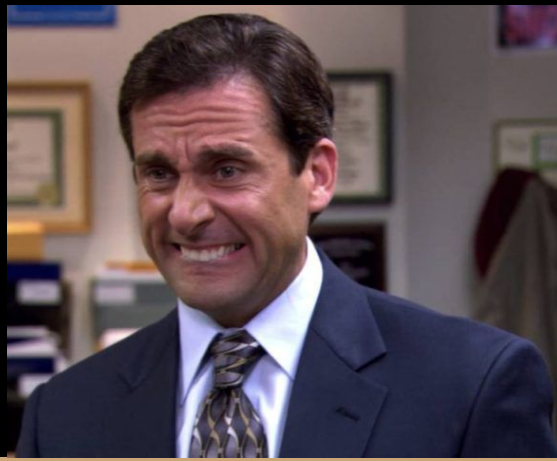
x86-64

Adicione diversos outros conceitos que complicam o hardware.

- Despacho múltiplo dinâmico.
- Multithreading simultâneo (A Intel chama de Hyper-threading).
- Instruções com quantidade de ciclos de clock variável.
- Buffers de reordenação de instruções.
- Predição dinâmica de desvios com tabelas dinâmicas.

E tudo de complicado que você pode imaginar ...

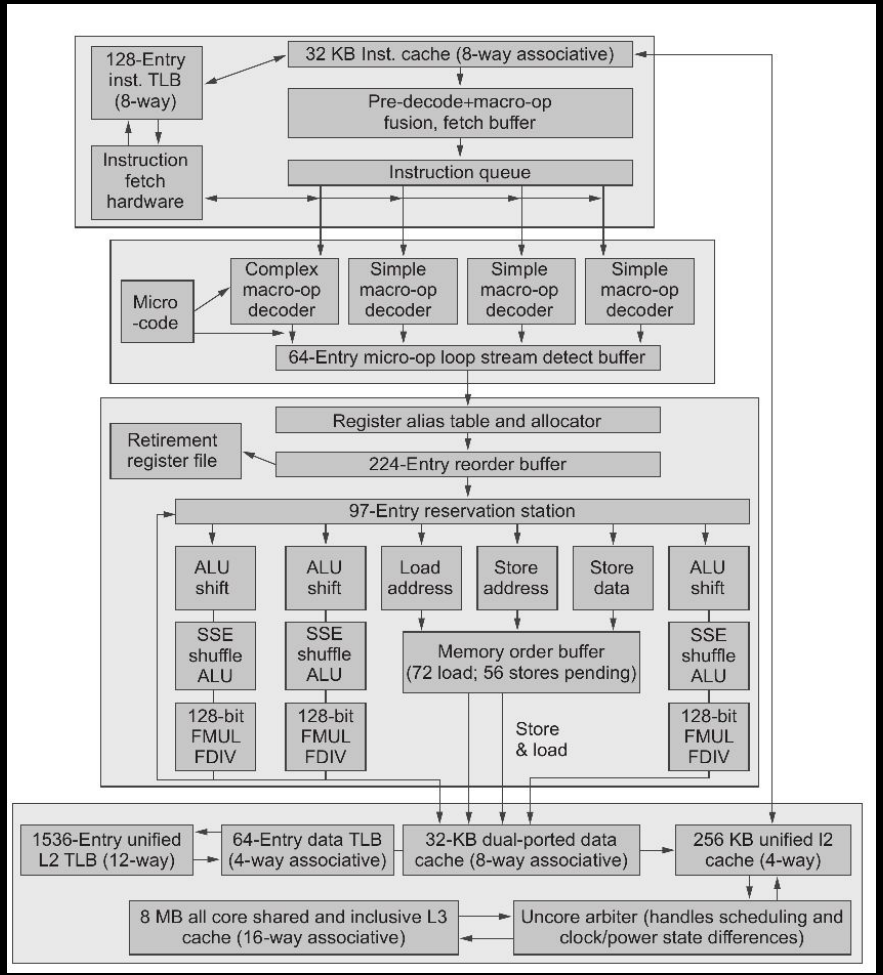
Junte isso a uma CPU CISC e você vai ter um projeto um tanto complicado.



Um Intel core i7

Unidades Funcionais de um Intel Core i7 6700 - 6a geração.

Henessy, Patterson (2020)



Um Intel core i7

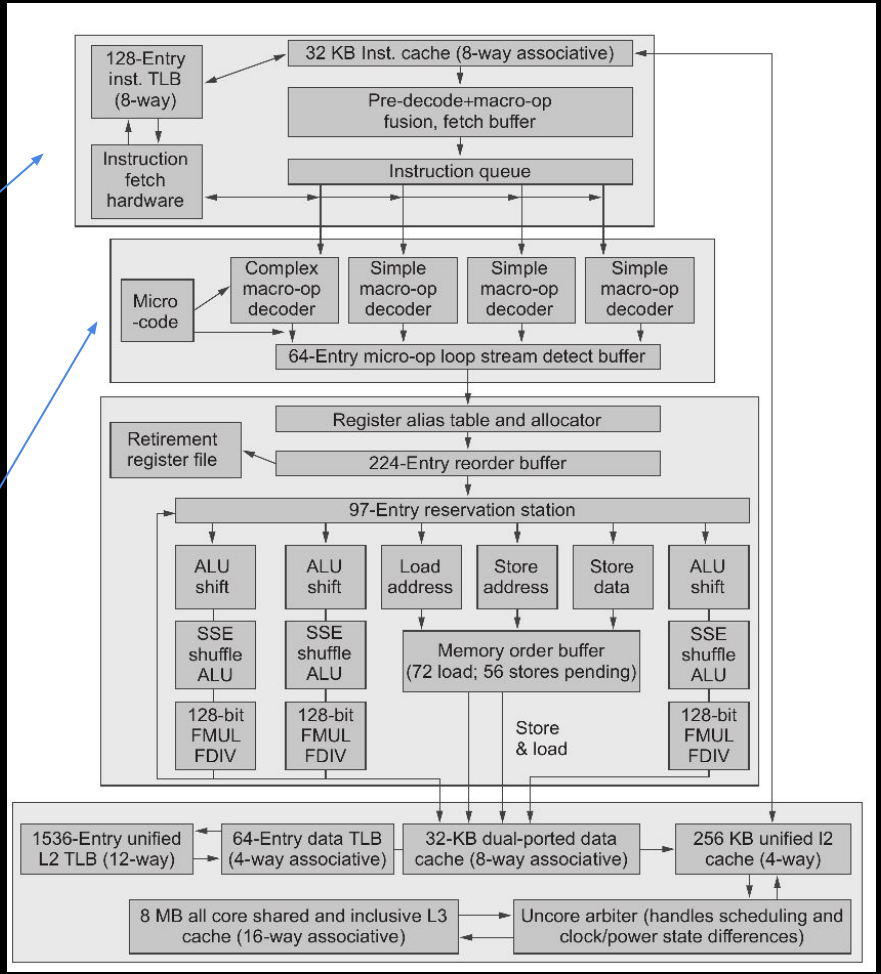
A unidade inicial faz uma pré-tradução. Instruções com tamanhos variados são difíceis de tratar.

Essas unidades traduzem as instruções CISC para **micro-ops.**

- Micro-ops são instruções RISC.
- Conceito introduzido no Pentium Pro de 1997.

Processadores AMD utilizam a mesma estratégia, mas dão nomes diferentes aos bois.

Sem isso, o pipeline e os demais itens seriam muito complexos.



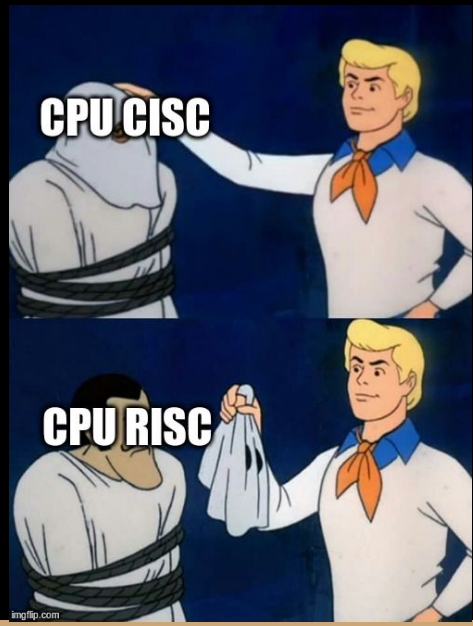
Profundidade dos Pipelines

Microprocessador	Ano	Clock	Estágios Pipeline	Tamanho Despacho	Fora de ordem?	CPUs por Chip	Potência
486	1989	0,025 GHz	5	1	Não	1	5 W
Pentium	1993	0,066 GHz	5	2	Não	1	10 W
Pentium Pro	1997	0,2 GHz	10	3	Sim	1	29 W
Pentium 4 Willamette	2001	2 GHz	22	3	Sim	1	75 W
Pentium 4 Prescott	2004	3,6 GHz	31	3	Sim	1	103 W
Intel Core	2006	3 GHz	14	4	Sim	2	75 W
Core i7 Nehalem	2008	3,6 GHz	14	4	Sim	2-4	87 W
Core Westmere	2010	3,73 GHz	14	4	Sim	6	130 W
Core i7 Ivy Bridge	2012	3,4 GHz	14	4	Sim	6	130 W
Core Broadwell	2014	3,7 GHz	14	4	Sim	10	140 W
Core i9 Skylake	2016	3,1 GHz	14	4	Sim	14	165 W
Intel Ice Lake	2018	4,2 GHz	14	4	Sim	16	185 W

Resumindo

Seu processador x86-64 é uma casca CISC envolvendo um processador RISC.

Pense em quanto hardware é “jogado fora” para criar essa casca que traduz de CISC para RISC!



Exercícios

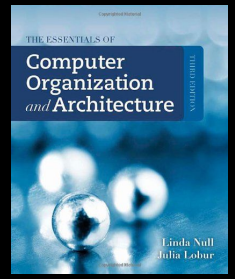
1. Quais as vantagens de uma CPU CISC sobre uma CPU RISC, e vice-versa?
2. Na aula vimos a CPU do seu computador pessoal é uma casca CISC que envolve uma CPU RISC. Discuta com os colegas o motivo de usarmos essa abordagem. Por que não utilizar uma CPU RISC de uma vez?

   Note que na aula foi contado só um lado da história. CPUs RISC tendem a gerar seus problemas. Utilize a resposta da questão 1 na discussão.
3. Estude a história de vida e morte do Itanium.

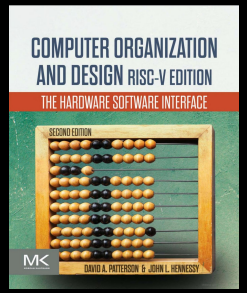
   Dica: Hennessy, Patterson. Arquitetura de Computadores: uma abordagem quantitativa. 6a Ed. 2019.

Referências

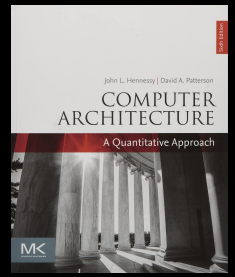
Null L., Lobur J. The Essentials of Computer Organization and Architecture. 2014.



Patterson, Hennessy. Computer Organization and Design RISC-V Edition: The Hardware Software Interface. 2020.

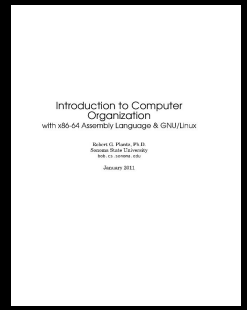


Hennessy, Patterson. Arquitetura de Computadores: uma abordagem quantitativa. 2019.



Intel. Manual x86-64. https://www.intel.com/content/www/us/en/developer/articles/technical/intel-sdm.html.

Plantz, R. G. Introduction to Computer Organization with x86-64 Assembly Language & GNU/Linux. 2011.



Licença