

## Arquiteturas e Abstrações

Yuri Kaszubowski Lopes

UDESC

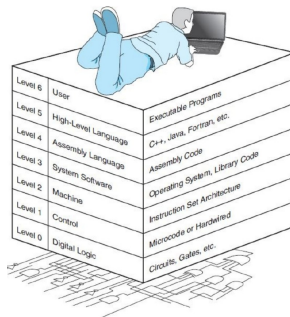
YKL (UDESC)

Arquiteturas e Abstrações

1 / 29

Anotações

## Os níveis de abstração de um computador



Null, Lobur (2014)

- "níveis de abstração" nos quais podemos trabalhar segundo Null, Lobur (2014)
- Na disciplina de AOC, em quais níveis estamos mais focados?
- Quais níveis dependem diretamente dos que estamos trabalhando?

YKL (UDESC)

Arquiteturas e Abstrações

2 / 29

Anotações

## Arquitetura de Von Neumann

- A arquitetura de Von Neumann foi criada por ...
- John W. Mauchly e J. Presper Eckert
- Criada enquanto eles trabalhavam no ENIAC
- A arquitetura seria empregada no sucessor do ENIAC, o EDVAC
- O projeto era secreto (Segunda Guerra Mundial) 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

YKL (UDESC)

Arquiteturas e Abstrações

3 / 29

Anotações

## Arquitetura de Von Neumann

- A arquitetura é famosa pelo conceito de "programa armazenado"
  - Parece algo trivial hoje, mas os primeiros programas de computadores eram "hardwired"
    - ★ Se precisar mudar o programa, precisa mudar o circuito
- Os programas executam no ciclo de Von Neumann
- Ciclo de busca-decodificação-execução:
  - 1 A CPU busca a próxima instrução da memória utilizando um contador de programa
  - 2 A instrução é decodificada para algo que uma ALU possa entender
  - 3 Os operandos necessários são carregados
  - 4 A ALU executa a operação e o resultado é armazenado na memória

Anotações

---

---

---

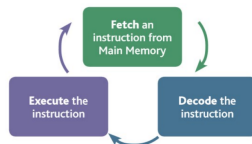
---

---

---

---

## Arquitetura de Von Neumann



### Nossa CPU MIPS segue o ciclo 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

Anotações

---

---

---

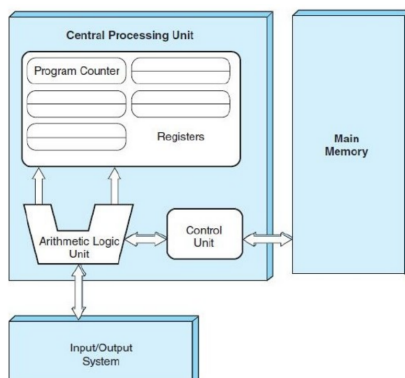
---

---

---

---

## Versão Moderna da Arquitetura de Von Neumann



Arquitetura de Von Neumann (Null, Lobur; 2014)

Anotações

---

---

---

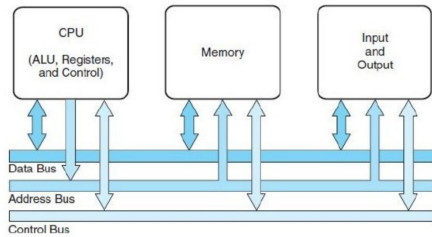
---

---

---

---

## Versão Moderna da Arquitetura de Von Neumann



Versão Modificada da arquitetura, com barramentos para dados, endereços e controle separados (Null, Lobur; 2014)

- Versão modificada e mais próxima de uma utilizada em uma CPU real
- Apesar das modificações no diagrama, ainda é uma arquitetura de Von Neumann

### Anotações

---

---

---

---

---

---

---

---

## Versão Moderna da Arquitetura de Von Neumann

### Nossa CPU MIPS segue o ciclo de Von Neumann?

- Não exatamente
  - Temos duas memórias separadas:
    - Uma memória de instruções
    - Uma memória para dados
- Vamos ver que 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 CPU's podem ser consideradas como pertencendo a arquitetura da Von Neumann

### Anotações

---

---

---

---

---

---

---

---

## Arquitetura de Von Neumann

- Na arquitetura de Von Neumann (estrita) temos:
  - Uma memória principal que armazena o programa e seus dados
  - Qual o problema?
    - Gera um gargalo, conhecido como Von Neumann bottleneck (Gargalo de Von Neumann)
    - No pipeline da CPU MIPS o problema fica claro
    - Geraríamos um hazard estrutural, e no mínimo precisaríamos injetar algumas bolhas

### Anotações

---

---

---

---

---

---

---

---

Arquitetura Harvard

- Temos memórias para dados e instruções separadas
  - Como no processador MIPS estudado
- Máquinas com Arquitetura Harvard Pura:
  - Sequem 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.

Anotações

---

---

---

---

---

---

---

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 (leia-se 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
    - ★ Nossas máquinas são de “múltiplos canais”, mas a CPU ainda requisita uma informação por vez de cada canal

Anotações

---

---

---

---

---

---

---

Exemplo

- Em um computador Linux, rode o comando `lscpu` para informações detalhadas sobre sua CPU

```
[yuri@ssdarch ~]$ lscpu
Architecture:                x86_64
CPU op-mode(s):              32-bit, 64-bit
Byte Order:                  Little Endian
Address sizes:                39 bits physical, 48 bits virtual
CPU(s):                      8
On-line CPU(s) list:         0-7
Thread(s) per core:          2
Core(s) per socket:          4
Socket(s):                   1
NUMA node(s):                1
Vendor ID:                   GenuineIntel
CPU family:                   6
Model:                       142
Model name:                   Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz
Stepping:                    12
CPU MHz:                     800.072
CPU max MHz:                  4600.0000
CPU min MHz:                  400.0000
BogoMIPS:                     4001.60
Virtualization:               VT-x
L1d cache:                   128 KiB
L1i cache:                   128 KiB
L2 cache:                     1 MiB
L3 cache:                     8 MiB
```

Anotações

---

---

---

---

---

---

---

## Arquiteturas Paralelas

- Computadores Paralelos
  - ▶ Atualmente utilizados desde seu computador pessoal até em supercomputadores
  - ▶ Existem diversos níveis de paralelismo:
    - \* Pipelining
    - \* Processadores superescalares
    - \* Processadores multicore
    - \* Multiprocessadores
    - \* Grids (heterogêneo)
    - \* Clusters (homogêneo)
    - \* ...
  - ▶ Vamos focar em multicore

### Anotações

---

---

---

---

---

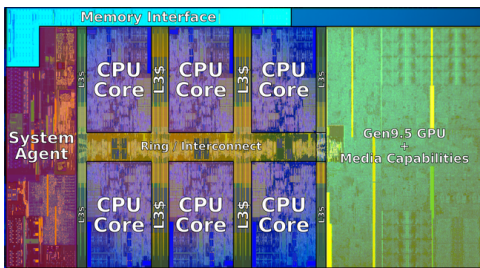
---

---

---

## Processadores Multicore

- Um processador multicore (com múltiplos núcleos) possui vários núcleos de processamento (processadores) em um único chip
  - ▶ Todos os processadores acessam a mesma memória principal por um mesmo barramento
  - ▶ Máquina UMA (Uniform Memory Access)
  - ▶ Comum em nossos computadores pessoais



Fonte: <https://en.wikichip.org/wiki/intel/core/J7/17-8700k>

### Anotações

---

---

---

---

---

---

---

---

## Processadores Multicore

- Por serem paralelos, temos efetivamente 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 CPU's como pertencendo a arquitetura de Von Neumann
  - ▶ Cada "core" é uma máquina de Von Neumann, cooperando para executar as tarefas
  - ▶ Segundo Null, Lobur (2014) podemos enxergar uma CPU moderna como uma máquina de Von Neumann que apresenta alguns aspectos de "non-von Neumannness"
    - \* ... alguns aspectos que não são de Von Neumann

### Anotações

---

---

---

---

---

---

---

---

## Mix and Match (Misturando)

- Podemos ver uma CPU x86-64 atual então como sendo
  - ▶ Dependendo do nível de memória
    - ★ Arquitetura Harvard
    - ★ Arquitetura de Von Neumann
  - ▶ Dependendo do seu entendimento:
    - ★ Arquitetura de Von Neumann
    - ★ Uma arquitetura separada, se encaixando exclusivamente em arquiteturas paralelas

### Anotações

---

---

---

---

---

---

---

## RISC versus CISC

- Alguns se referem a RISC e CISC como Arquiteturas
  - ▶ Na verdade são mais para “estilos de conjuntos de instrução”

### Anotações

---

---

---

---

---

---

---

## CISC

- Complex Instruction Set Computer
- Geralmente:
  - ▶ Contém um número (muito) grande de instruções
  - ▶ Instruções de tamanhos variados (no nosso MIPS32 todas as intruções são de 32-bits)
  - ▶ Instruções complexas, que podem executar “múltiplas coisas” em uma única operação:
    - ★ e.g., uma **mesma operação** busca o dado da memória, realiza uma operação com esse dado, e armazena o resultado na memória

### Anotações

---

---

---

---

---

---

---

RISC

- Reduced Instruction Set Computer
- Geralmente:
  - ▶ Possui um conjunto de instruções reduzido e simplificado
    - ★ Instruções são menos "poderosas"
    - ★ Objetivo é simplificar o projeto e criar instruções que executam mais rapidamente
  - ▶ Poucos formatos de instruções (e.g. tipo-R, tipo-I, tipo-J)
  - ▶ Instruções de tamanho fixo

Anotações

---

---

---

---

---

---

---

RISC versus CISC

- O processador MIPS visto em aula é RISC ou CISC?
  - ▶ RISC

Anotações

---

---

---

---

---

---

---

Tabela Comparativa

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

Atenção

Analise a tabela de maneira crítica. Essas não são regras, mas sim diferenças comumente encontradas em processadores RISC e CISC.

Anotações

---

---

---

---

---

---

---

RISC versus CISC

- Processadores RISC
  - O processador do seu celular
  - O processador MIPS
  - Microcontroladores
- Processadores CISC
  - O processador do seu computador pessoal

Anotações

---

---

---

---

---

---

---

Seu computador Pessoal: CISC?

- O Processador do seu computador pessoal é CISC
  - Não é bem assim
- Seu computador Pessoal
  - Construímos um pipeline simples na CPU MIPS
  - Imagine construir isso em uma CPU CISC
    - \* Instruções de tamanhos variados
    - \* Uma infinidade de instruções diferentes
    - \* Diversos formatos de instruções
    - \* Muitas instruções capazes de acessar a memória
    - \* ...
  - 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: e.g., unidade de ponto flutuante
    - \* Buffers de reordenação de instruções
    - \* Predição dinâmica de desvios com tabelas dinâmicas
    - \* E tudo de complicado que você pode imaginar ...

Anotações

---

---

---

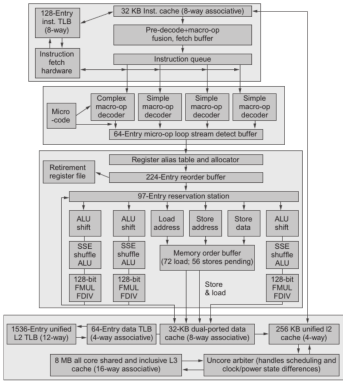
---

---

---

---

Um Intel core i7



Unidades Funcionais de um Intel Core i7 - 6ª geração

Anotações

---

---

---

---

---

---

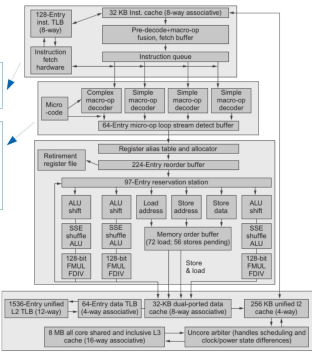
---



Um Intel core i7

A unidade inicial faz uma pré-tradução  
Instruções com tamanhos variados são difíceis de tratar, e  
precisam desse tratamento

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 se tornariam muito  
complexos



Unidades Funcionais de um Intel Core i7 - 6ª geração

Anotações

Profundidade dos Pipelines

Microprocessor	Year	Clock Rate	Pipeline Stages	Issue Width	Out-of-Order/Speculation	Cores/Chip	Power
Intel 486	1989	25 MHz	5	1	No	1	5 W
Intel Pentium	1993	66 MHz	5	2	No	1	10 W
Intel Pentium Pro	1997	200 MHz	10	3	Yes	1	29 W
Intel Pentium 4 Willamette	2001	2000 MHz	22	3	Yes	1	75 W
Intel Pentium 4 Prescott	2004	3600 MHz	31	3	Yes	1	103 W
Intel Core	2006	2930 MHz	14	4	Yes	2	75 W
Intel Core i5 Nehalem	2010	3300 MHz	14	4	Yes	1	87 W
Intel Core i5 Ivy Bridge	2012	3400 MHz	14	4	Yes	8	77 W

- Traduzir as instruções CISC para micro-ops (RISC): introduzido no Pentium Pro de 1997

Anotações

Resumindo

- Seu processador é uma casca CISC envolvendo um processador RISC
- Pense em quanto hardware é “jogado fora” só para criar essa casca que traduz de CISC para RISC!

Anotações

Referências

- D. Patterson; J. Henessy. **Organização e Projeto de Computadores: Interface Hardware/Software**. 5a Edição. Elsevier Brasil, 2017.
- Null L., Lobur J. The Essentials of Computer Organization and Architecture. Jones & Bartlett Publishers, 2014.
- STALLINGS, William. Arquitetura e organização de computadores. 10. ed. São Paulo: Pearson Education do Brasil, 2018.
- J. Henessy; D. Patterson. Arquitetura de computadores: Umaabordagem quantitativa. 6a Edição. Elsevier Brasil, 2014

Anotações

---

---

---

---

---

---

---

---

Anotações

---

---

---

---

---

---

---

---

Anotações

---

---

---

---

---

---

---

---