

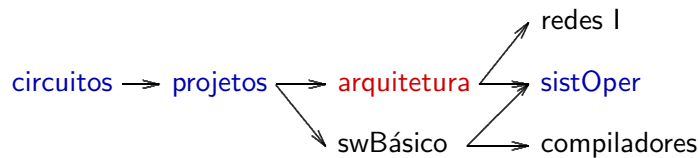
# CI212 – Organização e Arquitetura de Computadores

Roberto A Hexsel  
roberto@inf.ufpr.br  
www.inf.ufpr.br/roberto/CI212.html  
roberto-ci212@inf.ufpr.br

Material adicional para este curso:

www.inf.ufpr.br/roberto/ci212  
www.inf.ufpr.br/roberto/ci210  
www.inf.ufpr.br/roberto/ci064

## De Onde para Onde?



### O que se aprende nesta disciplina?

- como um programa em C é executado num computador;
- qual a interface entre o hardware e o software;
- o que determina o desempenho de um programa;
- o que determina o desempenho de um computador;
- o que um programador pode fazer quanto aos dois itens acima;
- o que é depuração (otimização) de desempenho.

## Bibliografia

livro texto:

**Computer Organization and Design**, P&H-COD  
D A Patterson e J L Hennessy, 4a Ed, Morgan Kaufmann, 2009

**Projeto e Organização de Computadores**,  
D A Patterson e J L Hennessy, trad (ruim) da 3a Ed (capa azul),  
Ed Campus, 2005

edições anteriores podem ser usadas **mas**  
números de seções, páginas e de exercícios serão da 4a Ed

**Leitura do livro é fundamental  
fazer os exercícios também**

cópia dos slides pode substituir o caderno  
**mas não substitui o livro!**

# Programa

- ★ apresentação da disciplina, introdução (1/2)
- ★ rev aritmética de ponto fixo, mult+div, aritm de ponto flutuante (2)
- ★ conjuntos de instruções - MIPS **TODOS DEVEM ESTAR COM O LIVRO!** (2)
- ★ circuito de dados do MIPS (ciclo longo) (1)
- ★ segmentação do circ de dados (pipeline), super-escalares (5)
- ★ revisão de sistemas de memória, memória cache (3)
- ★ prova Caps 1-5<sub>1/2</sub>
- ★ hierarquia de memória, memória virtual (3)
- ★ sistemas de E/S, interfaces de E/S com CPU, SO (2)
- ★ processamento paralelo (3)
- ★ prova Caps 1-7

## Por que estudar Arquitetura?

- ser um programador competente:
  - usar caches e memória virtual de forma vantajosa;
- aprender algoritmos e técnicas para alto desempenho:
  - tirar proveito da hierarquia de memória;
- usar bem recursos da máquina:
  - idem com relação a E/S, segmentação;
- poder trabalhar com eletrônica embarcada:
  - mercado novo e com poucos programadores competentes;
- entender artigos das revistas da ACM, IEEE;
- e, principalmente, ser feliz a vida é de vocês...

## Classes de computadores

- **desktop** – bom desempenho para usuário a custo baixo
- **servidor** – bom desempenho para uma aplicação grande (científica, engenharia), ou muitas pequenas (bDados, serv web)
  - \* desktop envenenado  $\approx$  3k US\$
  - \* servidor com múltiplas CPUs [1M, 10M] US\$
  - \* supercomputador >512 CPUs > 5M US\$
  - \* *warehouse scale computer* >50.000 CPUs > 100M US\$
- **embutido** – uso dedicado e escondido em algum produto celular, forno de microondas, freio ABS
  - \* vende-se > 20x  $\mu$ Proc de 32bits para uso embarcado do que para uso em desktops
  - \*  $\approx$  70% de todo sw novo é para ser embarcado
  - \* 6-7 engenheiros de SW para cada engenheiro de HW

## O quê vendeu onde (em 2012)

- 1,10 bilhões de celulares (decrecendo)
- 0,70 bilhões de smartphones (crescendo forte)
- 0,38 bilhões PCs desktops, mais de metade são laptops (estagn)
- 0,01 bilhões de tablets (crescendo forte)
- 0,02 bilhões servidores (crescendo pouco)
- 20,00 bilhões embarcados (em 2010, crescendo forte)  
6,10 bilhões ARM de USB a roteador...

## Modelo de Von Newman (1945)

### Computador com programa armazenado

Memória é um vetor de bits;  
interpretação dos bits definida por  
arquiteto do computador e/ou programador

Parte da memória contém instruções  
Parte da memória contém dados

## Abstrações

---

linguagem de alto nível	C, Haskell, Java, Python
linguagem de montagem	MIPS, ARM, 80x86
linguagem de máquina	bits, bytes e palavras
álgebra booleana	portas, flip-flops
equações diferenciais	resistores, transistores
mecânica quântica	portadores de carga

---

## Abstrações

---

Eng eletrônico, físico	portadores de carga, corrente, tensão
Eng “digital”	portas, flip-flops
Arquiteto	instruções, ULA, registradores
Programador de SO	ABI, instruções, interrupções, C
Programador de aplicação	API, bibliotecas
Usuário	navegador, telefone, caixa eletrônico

---

## Abstrações e Interfaces

Interface entre HW e SW = conjunto de instruções (Cdl)

Interface entre aplicativo e SO = *application binary interface* ABI

Interface entre programa e máquina = Cdl + ABI

Cada interface esconde detalhes e simplifica raciocínio

Cdl é a interface que facilita a programação do HW API do HW?

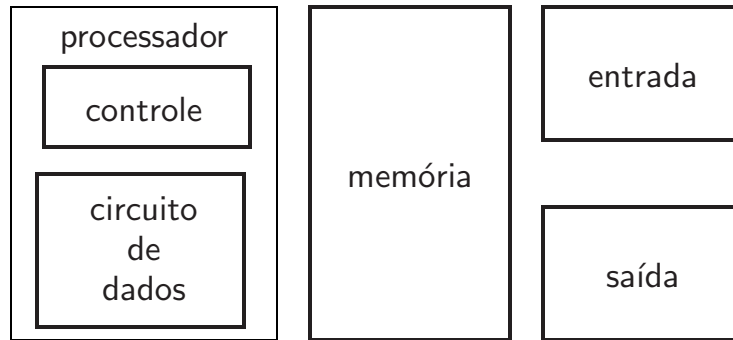
Cdl é a arquitetura do processador

uma arquitetura pode ser implementada de várias maneiras  
8086, 186, 286, 386, 486, Pentium, pPro, P-II, P-III, P-IV

## Traduções

alto nível	tradução
compilador, interpretador	C → MIPS
montador + ligador	MIPS → executável
carregador	executável → processo
simulador digital	esquema → circuito
simulador de circuitos	circuito → voltagens

# Organização de Computadores



## Definição de Arquitetura de Computadores

1. **arquitetura do conjunto de instruções (Cdl):**  
conjunto de instruções e registradores visíveis ao programador  
**Instruction Set Architecture = ISA**
2. **organização (micro-arquitetura):**  
blocos tais como sistema de memória, barramentos, CPU;  
mais de uma implementação de mesmo conjunto de instruções  
(AMD e Intel, 80{1,2,3,4,5,6}86)
3. **hardware:**  
tecnologia de implementação,  
circuitos integrados (CMOS), *pipelining* vs ciclo-longo

## Princípios de Projeto em Arquitetura

**Princípio 1:** simplicidade favorece regularidade

**Princípio 2:** menor é mais rápido (quase sempre)

**Princípio 3:** um bom projeto demanda compromissos

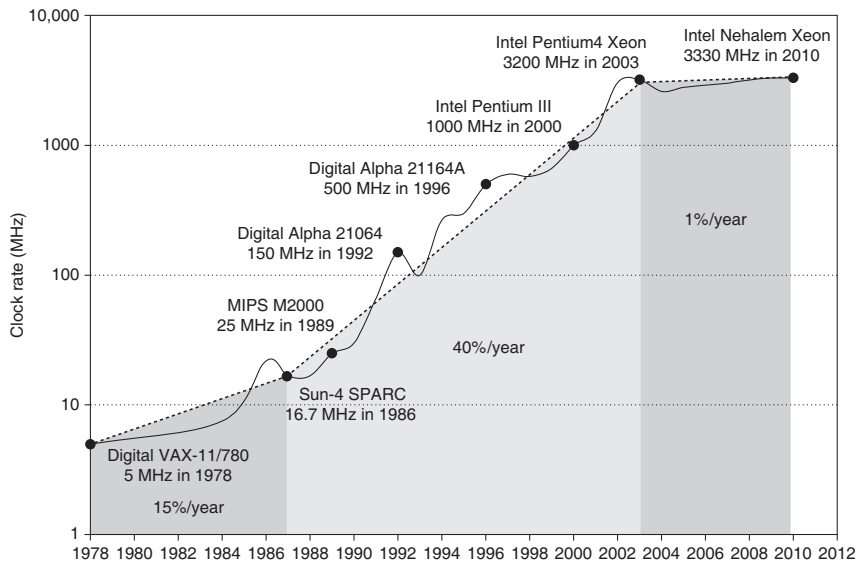
**Princípio 4:** o caso comum deve ser o mais rápido

## Família Intel

processador	ano	trans 10 <sup>6</sup>	área [mm <sup>2</sup> ]	clock [MHz]	
4004	1971	0.0023	12	0.75	
8008	1972	0.0035			
8080	1974	0.0045	190		
8085	1976	0.0065		1-5	
8086	1978	0.029		2-10	
80186	1981				
80286	1982	0.134	47	12.5	
80386	1985	0.275	43	16-33	
80486	1989	0.9-1.6	160-80	20-100	
Pentium	1993	3.3	148	60-200	
PentiumPro	1995	5.5	308	150-200	
PentiumII	1997	7.5		233-450	
Celeron	1999	19		300-700	
P3 xeon	1999	28		500-900	
P3 server	2001	48		1.400	
Pentium4	2002	55	250	2.400	70W
iCore7	2010	1170	257	3.330	130W

P4 dissipa 80W/cm<sup>2</sup> @ 3.2GHz

## Tendências I – Relógio



## Tendências 2 – Lei de Moore

Número de Transistores dobra a cada 18–24 meses  
Lei de Moore

