


# Arquitetura e Organização de Computadores II

# Conteúdo Programático

## 1. Introdução

- 1.1. Evolução histórica das arquiteturas
- 1.2. Taxonomia de arquiteturas paralelas
- 1.3. Visão geral das arquiteturas existentes

## 2. Arquiteturas Superescalares

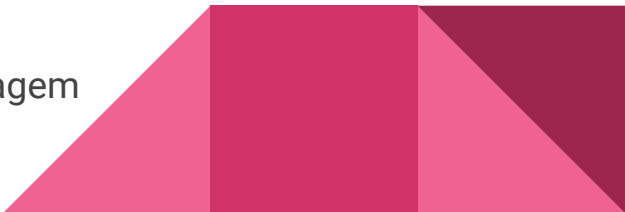
- 2.1. Paralelismo em nível de instrução
  - 2.2. Predição de desvio e execução especulativa
  - 2.3. Escalonamento dinâmico de instruções
  - 2.4. Arquitetura Multiple-Issue e Escalonamento Estático de instruções
  - 2.5. Desempenho e Eficiência
- 

# Conteúdo pragmático

## 3. Arquiteturas de Multiprocessamento Simétrico

- 3.1. Arquiteturas NUMA (Acesso Não Uniforme a Memória)
- 3.2. Arquiteturas multinúcleo e multithreading
- 3.3. Organizações de Arquiteturas Simétricas
- 3.4. Topologias de Rede
- 3.5. Coerência de cache e Protocolos
- 3.6. Desempenho e eficiência

## 4. Arquiteturas Não Convencionais

- 4.1. Processadores vetoriais
  - 4.2. Processadores VLIW
  - 4.3. Multiprocessadores embarcados
  - 4.4. Clusters e outros multiprocessadores de passagem de mensagem
- 

# Conteúdo pragmático

- 5. Aspectos Tecnológicos de Última Geração
  - 5.1. Sistemas de armazenamento
  - 5.2. Unidades de processamento gráfico (gpu)
- 6. Atividades com simulação de arquiteturas



# Critério de Avaliação

- Avaliações periódicas
  - 2 provas com valor 10,0 ambas com peso 1
  - 1 trabalho prático com valor 10,0 com peso 1
- Avaliação Final:
  - Uma prova com valor 10,0



# Evolução histórica das arquiteturas

- Durante os últimos 65 anos o progresso na tecnologia de computação foi muito alto:
  - Por menos de US\$ 500,00 se compra um computador com mais desempenho, memória principal e armazenamento que um computador de US\$ 1 milhão em 1985;
- Durante os primeiros 25 anos de existência dos computadores eletrônicos a melhoria de desempenho era cerca de 25% ao ano.
- Com o surgimento dos microprocessadores (final da década de 70) o crescimento subiu para 35% ao ano.



# Evolução histórica das arquiteturas

- Com uma alta taxa de crescimento combinado com as vantagens de produção microprocessadores (custo reduzido devido a produção em massa) a computação se tornou cada vez mais baseada em microprocessadores.
- Existiram também alterações no mercado que colaboraram com o sucesso de novas arquiteturas:
  - i. Eliminação da programação direta em assembly
  - ii. Criação de S.Os padronizados;



# Evolução histórica das arquiteturas

- Com todas essas mudanças um novo conjunto de arquiteturas com instruções mais simples tornou-se possível no início da década de 80:
  - RISC: Reduced Instruction Set Computer;
- As máquinas baseadas em RISC tiveram duas características as quais chamaram a atenção dos projetistas de arquiteturas:
  - Paralelismo em nível de instrução (Instruction-Level Parallelism - ILP);
  - Uso de cache;





# Evolução histórica das arquiteturas

- Tais evoluções maximizaram o padrão de desempenho:
  - Algumas arquiteturas acompanharam tal evolução e outras apenas deixaram de existir;
    - A arquitetura x86 da intel adaptou seus projetos traduzindo instruções do tipo 80x86 para instruções do tipo RISC;
    - Isso causou um overhead no hardware fazendo com que em algumas aplicações outras arquiteturas tivessem vantagem;



# Evolução histórica das arquiteturas

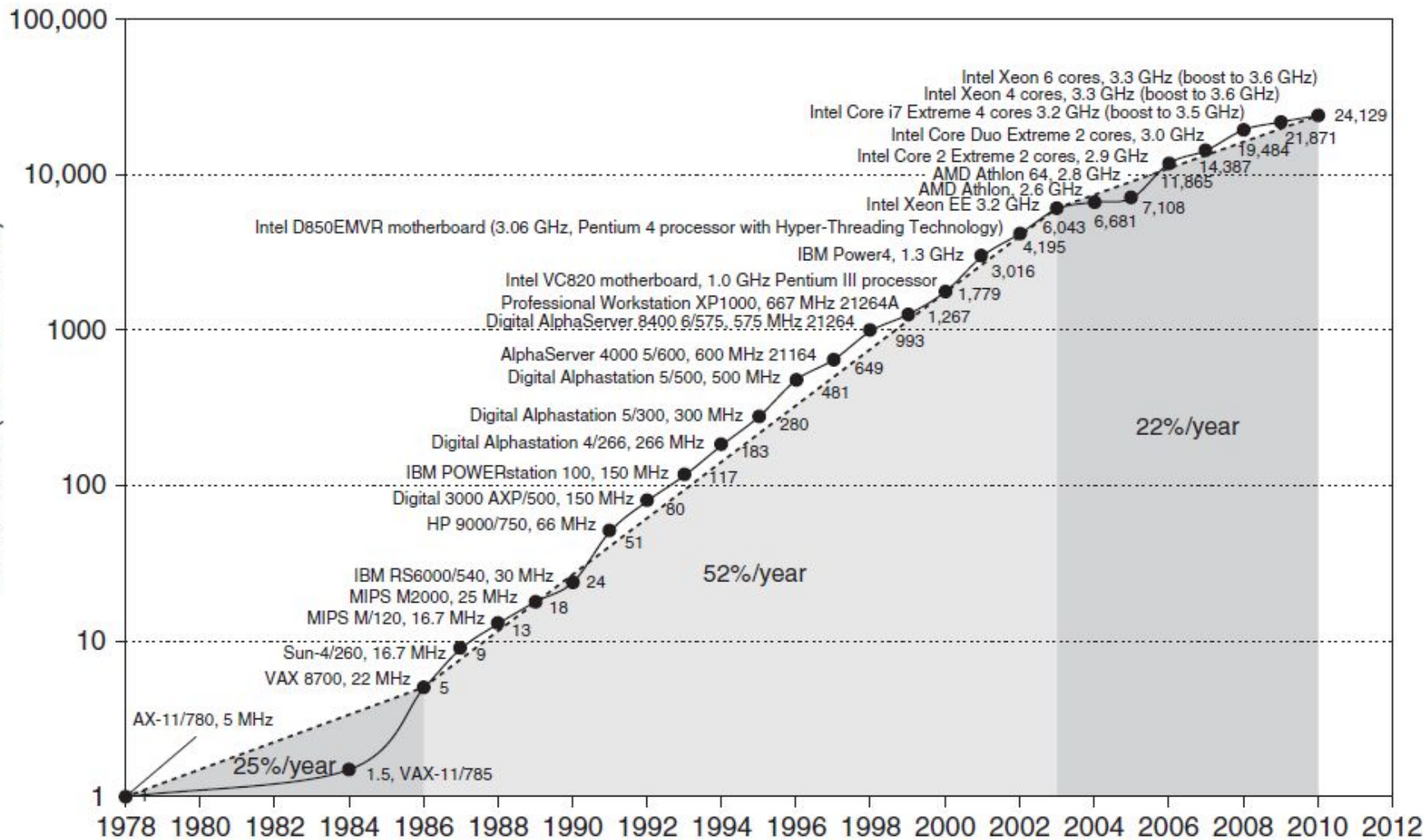
- A combinação de melhorias na organização e na arquitetura dos computadores trouxe um acréscimo anual de desempenho por volta de 50%.
- Esse crescimento trouxe 4 principais impactos:
  - Melhorou a capacidade para os usuários de computador;
  - Novas classes de computadores devido a redução de custo;
  - Dominância de projetos baseados em microprocessadores;
  - Desenvolvedores puderam trocar linguagens orientadas a desempenho por linguagens mais produtivas, mudando também a implementação de software.



# Evolução histórica das arquiteturas

- Desde 2003 a melhoria de desempenho dos uniprocessadores caiu para cerca de 22% ao ano;
  - Consequência do limite de dissipação de calor.
- Isso trouxe o surgimento de uma nova tendência:
  - Vários processadores por chip;
- Assim sinaliza-se uma passagem histórica, onde, além do ILP, conta-se também com:
  - Paralelismo em nível de Thread (Thread-Level Parallelism - TLP);
  - Paralelismo em nível de dados (Data-Level Parallelism - DLP);
- Tanto TLP quanto DLP são explicitamente Paralelos
  - Tarefa do programador modelar o software para aproveitar tais recursos.

Performance (vs. VAX-11/780)



# Classes de Computadores

- Tais alterações geraram diferentes mercados de computadores, onde cada um deles tem suas diferentes necessidades e aplicações:
  - Dispositivo pessoal móvel;
  - Computação desktop;
  - Servidores;
  - Computadores cluster;
  - Computadores embarcados



# Classes de Computadores

- Dispositivo pessoal móvel:
  - Dispositivo sem fio e com interfaces de usuário;
    - Celulares, Tablets, etc..
  - Custo é a principal preocupação
  - Eficiência energética baseada frequentemente no uso de baterias;
  - Aplicativos baseados na web e orientados para mídia;
  - Armazenamento em memória flash;
    - Consequência de requisitos de energia e tamanho
  - Necessidade de Minimizar a memória e o consumo de potência;
    - A eficiência energética depende também da dissipação de calor, e não só da bateria



# Classes de Computadores

- Computação desktop:
  - Primeiro e maior mercado em termos financeiros;
  - Desde de 2008 mais da metade dos computadores desktop fabricados por ano corresponde a laptops;
  - Busca relação preço desempenho;
    - O desempenho medido é principalmente em desempenho de cálculo e gráfico;
  - A computação desktop é bem caracterizada em termos de aplicações e benchmarks;
    - Porém o uso de aplicações web vem crescendo o que impõe novos desafios em avaliação de sistemas desktop



# Classes de Computadores

- Servidores:
  - Com o início da computação desktop os servidores passaram a ser usados para oferecer serviços de arquivo e computação em maior escala;
  - Tem como principais características:
    - Disponibilidade;
    - Throughput eficiente;





# Classes de Computadores

- Clusters:

- Com o fornecimento de software como serviço a classe de computadores chamada cluster também cresceu;
- Fatores críticos para clusters são:
  - Preço-desempenho;
  - Consumo de potência;
    - Uma grande quantidade do custo de um cluster é associado a sua potencia e seu resfriamento.
- Sua escalabilidade é tratada por softwares;
- Enfatizam aplicações interativas, armazenamento em grande escala, etc.



# Classes de Computadores

- Computadores embarcados:
  - São mais limitados que um dispositivo pessoal móvel em questão de hardware e software;
    - Em geral não executam software de terceiros;
    - Alguns exemplos são computadores que são encontrados em:
      - Fornos de micro-ondas;
      - Carros;
      - Switches;
      - Impressoras;
      - etc
  - Tem a mais extensa gama de poder de processamento e custo;
    - Podem variar de US\$0,10 até US\$100,00;





# Classes de Paralelismo e Arquiteturas Paralelas

# Classes de Paralelismo

- Existem basicamente dois tipos de paralelismo:
  - Paralelismo em nível de dados (Data-Level Parallelism - DLP):
    - Dados que podem ser operados ao mesmo tempo;
  - Paralelismo em nível de tarefas (Task-Level Parallelism - TLP):
    - Tarefas que podem ser operadas de maneira independente e em paralelo;



# Arquiteturas Paralelas

- Os dois tipos de paralelismo podem ser explorados pelo hardware de quatro principais maneiras:
  - Paralelismo em nível de instrução;
  - Arquiteturas Vetoriais e unidades de processador gráfico;
  - Paralelismo em nível de thread;
  - Paralelismo em nível de requisição;



# Arquiteturas Paralelas

- **Paralelismo em nível de instrução:**
  - Explora o paralelismo em nível de dados;
  - Usa duas principais ideias:
    - **Pipelining;**
      - Executar os estágios de uma instrução de maneira paralela
    - **Execução especulativa;**
      - Executar instruções com base em especulações e descartá-las quando a especulação estiver errada
- **Arquiteturas Vetoriais e unidades de processador gráfico:**
  - Explora paralelismo em nível de dados;
  - Realiza a aplicação da mesma instrução em uma coleção de dados diferentes;

# Arquiteturas Paralelas

- Paralelismo em **Nível de Thread:**
  - Explora o paralelismo:
    - Em nível de dados;
    - Em nível de tarefas;
      - O hardware utilizado é fortemente acoplado, porém é permitida a interação entre threads;
- **Paralelismo em nível de requisição:**
  - Explora o paralelismo em tarefas muito desacopladas
  - Tais tarefas são especificadas pelo programador ou pelo S.O.



# Taxonomia de Flynn

- Essas 4 maneiras de suporte a paralelismo por parte do hardware podem ser classificadas de acordo com a taxonomia criada por Michael Flynn por volta de 1960.

		Fluxo de dados	
		Simples	Multiplo
Fluxo de instruções	Simples	SISD	SIMD
	Multiplo	MISD	MIMD



# Taxonomia de Flynn

- Fluxo simples de instruções, fluxo simples de dados (Single instruction stream, single data stream - SISD):
  - Categoria composta pelos uniprocessadores;
    - Pode explorar paralelismo em nível de instrução;
  - Exemplos:
    - Pentium 4;
    - Athlon XP;
    - etc...



# Taxonomia de Flynn

- Fluxo simples de instruções, fluxos múltiplos de dados (Single instruction stream, multiple data streams - SIMD)
  - Categoria composta por computadores de múltiplos processadores;
    - O paralelismo ocorre em nível de dados;
    - Todos os processadores executam a mesma instrução;
      - Cada um em seu conjunto de dados.
  - Exemplos:
    - Arquiteturas vetoriais;
    - Unidades de Processamento Gráfico;
    - etc...



# Taxonomia de Flynn

- Fluxos múltiplos de instruções, fluxo simples de dados (Multiple instruction streams, single data stream - MISD);
  - Categoria composta por computadores de múltiplos processadores;
    - Todos eles executam diferentes instruções sobre o mesmo conjunto de dados;
  - Não existem processadores comerciais deste tipo;



# Taxonomia de Flynn

- Fluxos múltiplos de instruções, fluxos múltiplos de dados (Multiple instruction streams, multiple data streams - MIMD)
  - Categoria composta por computadores de múltiplos processadores;
    - Cada processador tem suas executa suas próprias instruções sobre seu próprio conjunto de dados;
    - Podem ser arquiteturas que exploram o paralelismo principalmente de duas maneiras:
      - Em nível de thread;
      - Em nível de requisições;



# Referencias

PATTERSON, D. A.; HENNESSY, J. L. Computer Architecture: A Quantitative Approach. Fifth Edition

PATTERSON, D. A.; HENNESSY, J. L. Computer Organization and Design: The Hardware/Software Interface. Fourth edition

