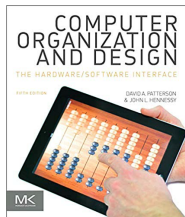


Aula 01 –Introdução

Prof. Dr. Clodoaldo Aparecido de Moraes Lima



Apresentação da Disciplina

Sumário

- Objetivos da disciplina
- Programa e Conteúdo
- Ferramentas Utilizadas
- Critérios de Avaliação
- Recuperação
- Bibliografia

Objetivos

- Apresentar conceitos básicos em projeto, organização e arquitetura de computadores
- Focalizando em aspectos de interface software-hardware.

Conteúdo

- Revisão de conceitos sobre operações e expressões lógicas e tabela-verdade.
- Circuitos lógicos combinacionais e sequenciais, sua aplicação na construção de blocos mais complexos como autômatos finitos e bancos de memória.
- Subsistemas de memória, organização, síntese e análise.
- Unidades de processamento: elementos básicos, relação com autômatos finitos, operação geral, fluxo de dados , macro instruções e microinstruções, unidade de controle, fundamentos, desenvolvimento e implementação.
- Conceitos básicos sobre pipeline .

Conteúdo

- Técnicas para organização de E/S, barramentos, uso de Acesso Direto à Memória (DMA).
- Arquiteturas de conjunto de instruções
- Linguagem de montagem
- Relações entre organização, arquitetura, sistema operacional e compiladores para o desempenho geral de um sistema computacional.

Programa e conteúdos

- Aula 1 - Apresentação da disciplina: Programa, Cronograma, Avaliações
- Aula 2 - Operações e expressões lógicas e tabela-verdade
- Aula 3 - Circuitos lógicos combinacionais e sequenciais
- Aula 4 - Construção de blocos mais complexos como autômatos finitos
- Aula 5 - Subsistemas de memória
- Aula 6 - Unidades de processamento - Parte 1
- Aula 7 - Unidades de processamento - Parte 2
- Aula 9 - Unidades de processamento - Parte 3
- Aula 10 - Introdução a Pipeline
- Aula 11 - Prova P1 (13/05)

Programa e conteúdos

- Aula 12 - Técnicas para organização de E/S - Parte 1
- Aula 13 - Técnicas para organização de E/S - Parte 2
- Aula 14 - Arquiteturas de conjunto de instruções - Parte 1
- Aula 15 - Arquiteturas de conjunto de instruções - Parte 2
- Aula 16 - Linguagem de montagem - Parte 1
- Aula 17 - Linguagem de montagem - Parte 2
- Aula 18 - Relações entre organização, arquitetura
- Aula 19 - Prova P2 (26/06)
- Aula 20 - Prova Substitutiva (01/07)
- REC

Critérios de Avaliação

Avaliação

- Frequência Mínima: 70%
- 2 Trabalhos (EP1, EP2)
- Média do trabalho (MP)
- $MT = (EP1 + EP2) / 2$
- 2 Provas Individual (P1, P2)
- Média das Provas (MP)
- $MP = (P1 + P2) / 2$
- Média Final (MF)
- $MF = 0.8 * MP + 0.2 * MT$
- Se $MF \geq 5.0$ então Aluno APROVADO
- Senão Se $MF \geq 3.0$ então Aluno em Recuperação
- Senão Aluno REPROVADO

Básica

- W. STALLINGS - Arquitetura e organização de computadores. Pearson, 10a. edição, 2017.
- PATTERSON, D.A.; HENNESSY, J.L. Computer Organization and Design MIPS Edition: The Hardware/Software Interface, Morgan Kaufmann, 6a edition, 2021
- SA. S. TANENBAUM - Organização estruturada de computadores. Pearson, 6a. edição, 2013.

Arquitetura versus Organização

Arquitetura

- Refere-se a atributos que tem impactos diretos sobre a execução lógica de um programa. Esses atributos são:
 - conjunto de instruções,
 - numero de bits que representa um determinado dado,
 - mecanismos de entrada e saída, entre outros.
- Lida com o funcionamento do Sistema Computacional.
- Corresponde aos aspectos visíveis a um programador em linguagem de máquina, tais como repertório de instruções, número de bits utilizado para representar vários tipos de dados, mecanismo de E/S e modos de endereçamento.

Organização

- Refere-se as unidades operacionais e suas interconexões. Os atributos que representa a organização de um computador são:
 - detalhes de hardware tais como sinais de controle,
 - interfaces entre computadores e periféricos,
 - tecnologias de memórias utilizadas.
- Diz respeito às unidades operacionais (UCP, unidade de memória, barramentos, sinais de controle, etc) necessárias para implementar as especificações de uma arquitetura. A organização é em geral transparente ao programador.

Visão geral

- A computação em si requer a utilização de vários conceitos de áreas correlatas: matemática, física, química, eletromagnetismo, etc.
- A revolução dos últimos anos tornou os computadores menores, mais rápidos e eficientes.
- Uma das palavras-chave na computação: Inovação!!

Visão geral

- A computação levou à terceira revolução: a informação.
- Oportunidades na computação estão diretamente relacionadas ao seu avanço na capacidade de processamento.
- Lei de Moore: “Número de transistores em um circuito integrado dobrará a cada 2 anos...”.
- Novas aplicações se tornam possíveis a medida que avanços são atingidos.

Visão geral

- Algumas aplicações que a bem pouco tempo eram ficção:
 - Computadores embarcados;
 - Telefones móveis / smartphones;
 - Projeto genoma;
 - World Wide Web;
 - Inteligência artificial;
 - item etc. . .

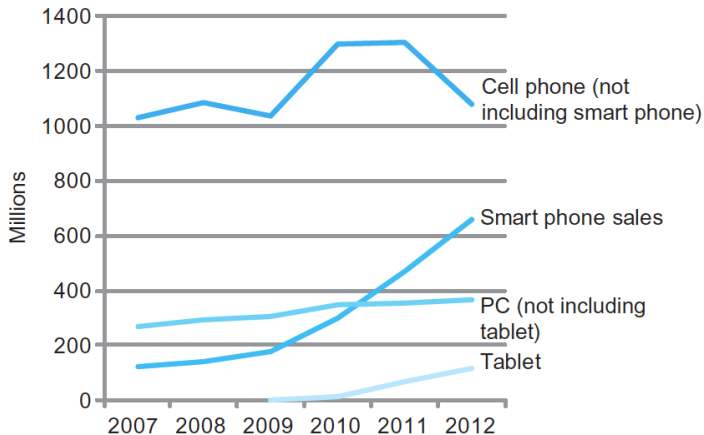
Classes de computadores

- Computadores pessoais (PCs):
 - Tipo de computador mais conhecido atualmente.
 - Bom desempenho aliado ao baixo custo.
- Servidores:
 - Grandes computadores com maior capacidade de processamento.
 - Geralmente com recursos compartilhados e acessados via rede.

Classes de computadores

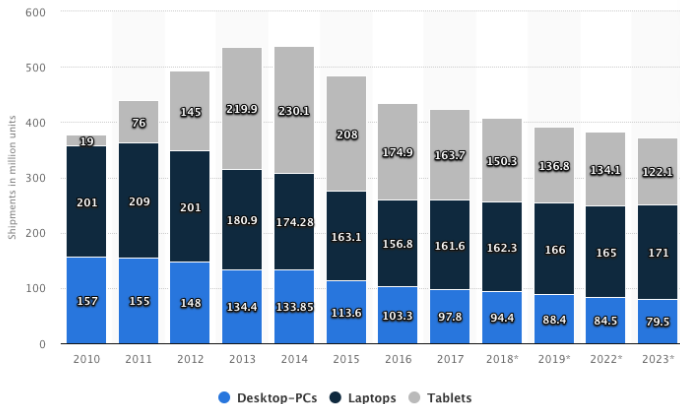
- Computadores embarcados:
 - Maior classe de computadores existente.
 - Soluções com propósito específico: carros, aviões, eletrodomésticos, etc.
 - Combinam desempenho desejado com limitações de custo e consumo energético

Introdução



Número de dispositivos manufaturados por ano.

Introdução



Número de dispositivos entregues por ano.

8 conceitos em arquitetura de computadores

Design para a lei de Moore

- Gordon Moore (1965): Capacidade de um circuito integrado irá dobrar a cada 18—24 meses.
- O design de um novo chip pode levar anos.
- É necessário prever onde a tecnologia estará quando da finalização de um novo produto.

8 conceitos em arquitetura de computadores

Abstração para simplificação

- Visa aumentar a produtividade de arquitetos e desenvolvedores.
- A abstração representa o design de uma solução em diferentes níveis de representação.
- Detalhes de baixo-nível são escondidos de modo a oferecer um modelo mais simples às camadas de alto nível.

Fazer o “caso comum” mais rápido

- Tende-se a obter uma melhoria significativa do desempenho mais rapidamente.
- O “caso comum” é frequentemente mais simples de se otimizar que o(s) “caso(s) raro(s)”.
- Definir qual é o “caso comum” pode requerer experimentações e diferentes análises.

8 conceitos em arquitetura de computadores

Desempenho via paralelismo

- Paralelismo visa a execução de tarefas de forma simultânea.
- Aumentar o número de componentes físicos é uma das maneiras de melhorar o desempenho.

Desempenho via pipelining

- É um tipo de implementação de paralelismo muito comum na computação.
- Tipo de abstração que permite enxergar a organização interna de computador como se fosse um “encanamento” de recursos.
- A divisão do “pipe” em estágios facilita a otimização

8 conceitos em arquitetura de computadores

Desempenho via predição

- Algumas vezes pode ser mais rápido, na média, “chutar” o caminho a tomar.
- Esperar para ter “certeza” pode ser mais custoso que “chutar” uma opção.
- Se “errar pouco”, o custo médio de cada erro “pode compensar”.

8 conceitos em arquitetura de computadores

Hierarquia de memórias

- Desenvolvedores querem que a memória seja rápida, grande e barata.
- O desempenho da memória normalmente limita o tamanho/velocidade dos problemas que podem ser resolvidos.
- As demandas pela memória podem ser endereçadas através do uso de uma organização hierárquica específica.

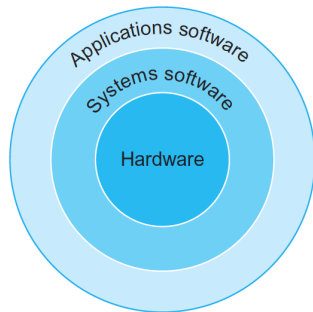
8 conceitos em arquitetura de computadores

Dependabilidade via redundância

- Sistemas computacionais devem suportar certos níveis de falhas.
- A “dependabilidade” é alcançada através da inclusão de componentes que podem tratar/detectar uma falha quando de sua ocorrência.
- Analogia do caminhão que possui dois pares de rodas em cada lado: redundância

Integração software / hardware

- Uma aplicação (software) simples pode possuir milhares de linhas de código.
- O hardware consegue executar somente instruções extremamente simples de baixo nível.
- A tradução das instruções de alto nível em baixo nível envolve a abstração de várias camadas de software.



Software de sistema (systems software)

- Dois tipos são centrais a qualquer sistema computacional moderno: Sistemas operacionais e compiladores.
- Sistemas operacionais:
 - Gerenciamento de operações de I/O (Input / Output);
 - Alocação e gerenciamento das memórias;
 - Isolamento de aplicações entre si e entre o sistema;
 - Gerenciamento de recursos em geral;
 - etc.

Software de sistema (systems software)

- Dois tipos são centrais a qualquer sistema computacional moderno: Sistemas operacionais e compiladores.
- Compiladores:
 - Tradução de programas em alto-nível para instruções compreendidas pelo hardware;
 - Otimizações e “ajudas” ao processador;
 - etc.

Linguagem de alto-nível para linguagem de hardware

- O hardware só compreende sinais elétricos: on e off.
- Dois bits (ou linguagem binária) é utilizada para a comunicação com os computadores.
- A linguagem binária (de máquina) é utilizada para codificar as instruções enviadas à CPU. Ex.: 1000110010100000.
- A linguagem assembly é utilizada para codificar as instruções de maneira simplificada: add A,B.

Linguagem de alto-nível para linguagem de hardware

- Apesar de mais simples, a linguagem assembly ainda é muito complexa.
- A necessidade de produtividade levou a criação das linguagens de programação de alto-nível.
- Os compiladores surgiram como forma de auxiliar nesse processo de tradução.

Integração software / hardware

Linguagem de alto-nível para linguagem de hardware

