

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL  
FACULDADE DE INFORMÁTICA**

**DEPARTAMENTO:** Fundamentos da Computação

**DISCIPLINA:** Organização de Computadores

**CÓDIGO:** 46183

**CRÉDITOS:** 04

**CARGA HORÁRIA:** 60 horas-aula

**REQUISITOS:**

**Pré-Requisito:** 4112D-04 - Estruturas Algébricas (INF)

**Requisito Especial:** 4490R-04 - Circuitos Digitais (INF)

**OBJETIVOS:** O cumprimento da disciplina busca dar ao aluno, ao final do semestre condições de:

1. Compreender o processo de projeto de sistemas digitais e a manipular modelos usados para representar este processo.
2. Dominar pelo menos um modelo de representação do processo de projeto de sistemas digitais.
3. Compreender o modelo de programa armazenado de von Neumann e sua relação com a organização de "hardware" de computadores modernos.
4. Conhecer os componentes básicos da organização de "hardware" de um computador e entender sua classificação segundo a finalidade a que se destinam, controle do fluxo de informação (bloco de controle) e transformação de informação (bloco de dados).
5. Distinguir bloco de dados e bloco de controle de um computador, quanto a função, composição e construção.
6. Identificar os aspectos importantes de linguagens de descrição de "hardware" (HDLs) e sua relação com outras técnicas conhecidas de projeto de sistemas digitais.
7. Aprender os conceitos básicos relacionados a conjuntos de instruções de processadores, linguagem de montagem e linguagem objeto, bem como programação de processadores nestas linguagens.
8. Dominar uma HDL específica, e ser capaz de empregá-la para descrever sistemas digitais.
9. Dominar um subconjunto mínimo de técnicas de implementação do bloco de dados e seus componentes e do bloco de controle e seus componentes.
10. Ter noções de conceito avançado em organização de computadores, tal como "pipelines".

**EMENTA:**

O modelo bloco de dados-bloco de controle: computador de programa armazenado. Fluxo de execução de instruções. Avaliação de desempenho. Otimização do bloco de dados e de controle: simulação e síntese. Ferramentas de projeto auxiliado por computador. Organização de computadores e execução de programas. O processo de projeto de sistemas digitais.

**UNIDADE:** 01

**Nº DE HORAS EM PERCENTUAL:** 20%

**CONTEÚDO:** O Processo de Projeto de Sistemas Digitais

### 1.1 Conceitos fundamentais de Sistemas Digitais

1.1.1 Álgebra Booleana

1.1.2 Circuitos Combinacionais

1.1.3 Circuitos Seqüenciais

1.1.4 Otimização e validação de projeto

### 1.2 O processo de projeto de sistemas digitais

1.2.1 Níveis de abstração e domínios de descrição - o diagrama Y

1.2.2 Outros modelos para representar o processo de projeto

### 1.3 Taxonomias de sistemas digitais

1.3.1 Critérios de classificação: programação, personalização, construção, etc

1.3.2 Taxonomias e a ortogonalidade de critérios

## **PROCEDIMENTOS E RECURSOS:**

A distribuição das aulas sugerida é explorar o item 1.1 em 6 horas aula, reservando em torno de 6 horas aula para os item 1.2 e 1.3.

Esta unidade introdutória visa fornecer modelos que servirão ao longo do curso de ferramental para colocar o assunto numa perspectiva mais ampla. Neste sentido, modelos para representar o processo de projeto como o diagrama Y são uma maneira de racionalizar a relação entre os métodos "ad hoc" de projeto de sistemas digitais vistos na disciplina Circuitos Digitais.

As classificações de sistemas digitais apresentadas devem empregar múltiplos critérios, visando dar uma idéia global dos avanços tecnológicos recentes e das alternativas de implementação de sistemas digitais. As justificativas de existência de formas diversas de implementação para uma dada funcionalidade devem ser múltiplas, econômicas, históricas e tecnológicas.

Aconselha-se que as etapas típicas de concepção de um sistema digital sejam apresentadas com base em exemplos práticos de descrições, seja por meio de ferramental de domínio prévio, como esquemáticos e diagramas de tempo, seja através de descrições ainda não dominadas pelos alunos, mas intuitivas, tais como uma especificação em português estruturado ou através de uso de um linguagem procedural como C ou Pascal. O objetivo é reduzir a aridez da descrição pura e simples de etapas de projeto, sem a necessária contextualização.

Exercícios intra- e extra-classe devem ser usados como recursos para reforçar os conteúdos vistos em aula, podendo envolver a pesquisa de exemplos de sistemas digitais, fluxos de projeto, ferramentas de projeto ou modelos formais.

**UNIDADE:** 02

**Nº DE HORAS EM PERCENTUAL:** 35%

**CONTEÚDO:** O Modelo Bloco de Dados - Bloco de Controle

**2.1.** O Modelo de von Neumann - o computador de programa armazenado

**2.2.** Um estudo de caso

**2.2.1.** Especificação da arquitetura - instruções e modos de endereçamento

**2.2.2.** Programação em linguagem de montagem

**2.2.3.** Particionamento e Implementação da especificação

**2.2.3.1.** Componentes do bloco de dados - operadores, memórias e conexões

**2.2.3.2.** Componentes do bloco de controle - a máquina de estados do controle

**2.2.4.** Análise da implementação – microsimulação e alternativas de implementação

### **PROCEDIMENTOS E RECURSOS:**

A distribuição das aulas sugerida é explorar o item 1 nas primeiras duas horas aula, reservando quatro horas aula para cada um dos itens 2.1, 2.2, 2.3.1 e 2.3.1, e duas horas para o item 2.3.

Esta unidade introduz o modelo de von Neumann, sendo assim a mais fundamental da disciplina. Partindo dos conhecimentos adquiridos na disciplina de Circuitos Digitais, apresenta-se o modelo e sua implementação em “hardware”, através do uso de um estudo de caso de organização de um processador hipotético. O estudo de caso é apresentado pronto aos alunos em sala de aula, e a justificativa da estrutura dos componentes da organização e da organização em si deve ser apresentada em termos qualitativos e quantitativos.

O estudo de caso inicialmente é apresentado sob forma de uma especificação abstrata, a arquitetura do processador, no item 2.1. Os itens restantes da unidade são usados para desenvolver a implementação da arquitetura, a organização do processador.

A implementação da organização do processador será feita em paralelo nesta disciplina e na disciplina de Laboratório de Organização de Computadores, motivo pela qual a segunda tem a primeira como co-requisito. Deve ser enfatizado o processo hierárquico de projeto, largamente apoiado em sucessivas decomposições da especificação inicial. Para levar a cabo a implementação, deverá ser empregado um sistema de projeto auxiliado por computador, preferencialmente um sistema comercial moderno baseado em dispositivos programáveis tais como Field-Programmable Gate-Arrays (FPGAs). O nível de abstração aconselhado para o desenvolvimento é o nível lógico, mediante emprego de ferramentas como editor de esquemáticos simuladores lógico e de temporização, todas as ferramentas tendo suporte para projetos hierárquicos. Esta unidade constitui-se assim, na exploração do processo clássico de projeto de sistemas digitais.

**UNIDADE:** 03

**Nº DE HORAS EM PERCENTUAL:** 35%

**CONTEÚDO:** Uma Linguagem de Descrição de “Hardware” (HDL)

### 3.1 Relação entre descrições gráficas e textuais

3.1.1 Nível de abstração

3.1.2 Tratamento automatizado

3.1.3 Exemplos

### 3.2 Estrutura da linguagem de descrição de “hardware”

3.2.1 Declarações

3.2.2 Comandos

3.2.3 Estruturas de Controle

### 3.3 Descrição do estudo de caso em HDL

3.3.1 Descrições comportamentais

3.3.2 Descrições estruturais

3.3.3 Implementação

## **PROCEDIMENTOS E RECURSOS:**

A distribuição das aulas sugerida é explorar o item 1 nas primeiras quatro horas aula, reservando seis horas aula para cada um dos itens 2 e 3. No item 1, aconselha-se apresentar exemplos de implementação de blocos de “hardware” conhecidos dos alunos, facilitando a apreensão de ambas, sintaxe e semântica da linguagem.

Esta unidade revisita o assunto da unidade anterior sob uma nova ótica, a de emprego de linguagens de descrição de “hardware” (“hardware description languages” ou HDLs) no projeto de sistemas digitais em geral, e de arquiteturas de computadores em particular. Uma vez que muitas das HDLs modernas permitem o emprego de diversos níveis de abstração de projeto, cabe salientar que a ênfase aqui não deve ser apenas em apresentar uma nova forma de se fazer projeto, mas um novo paradigma, aquele subjacente a algumas das estratégias de projeto habilitadas por HDLs: descrições abstratas comportamentais, verificação formalizada de atendimento de propriedades da especificação, etc.

No item 3.1, aconselha-se mostrar o mapeamento entre trechos de descrições em HDL de módulos de sistemas digitais e a descrição estrutural correspondente em diagramas de portas lógicas, bem como vice-versa. O objetivo aqui é facilitar a interpretação da semântica da linguagem, evidenciando a distinção entre linguagens de programação e HDLs.

No item 3.2, aconselha-se dar menos ênfase ao detalhamento da estrutura sintática e léxica da linguagem escolhida, concentrando-se na semântica das construções e no seu mapeamento para estruturas de “hardware” conhecidas. O domínio pelo aluno da sintaxe das estruturas deverá resultar da realização de exercícios intra e extra classe.

No item 3.3, aconselha-se empregar o mesmo estudo de caso utilizado na unidade 2, usando as diferentes descrições para comparar as vantagens e inconvenientes de trabalhar em distintos níveis de abstração. Os estudos de caso da unidade anterior devem ter sua implementação realizadas através desta nova forma de descrição, atendendo aos critérios do paradigma estabelecido pelo emprego da HDL em questão.

**UNIDADE:** 04

**Nº DE HORAS EM PERCENTUAL:** 10%

**CONTEÚDO** Tópicos Avançados em Organização de Computadores

**4.1** O conceito de "pipeline"

**4.1.1** Introdução ao conceito

**4.1.2** Pipeline em computadores

**4.1.3** Bloco de dados com pipeline

**4.1.4** Bloco de controle com pipeline

**4.1.5** Problemas associados a pipelines

**PROCEDIMENTOS E RECURSOS:**

Esta unidade deve servir para abrir perspectivas de estudos futuros sobre o tema de organização de computadores, mostrando os caminhos iniciais empregados na superação do gargalo de von Neumann, através de: paralelismo a nível de instrução no caso de pipelines e aumento da complexidade das operações realizadas diretamente pelo hardware.

**AVALIAÇÃO:**

$$G1 = (P1 + P2 + TP) / 3$$

Onde:

P1 – Prova abrange as unidades 1 e 2

P2 – Prova abrange as unidades 3 e 4

TP - Trabalho(s) Prático(s)

## **BIBLIOGRAFIA:**

### **• LIVRO(S) TEXTO**

1. PATTERSON, D. A. & HENNESSY, J. L. "Organização e Projeto de Computadores - A interface Hardware/Software - editora LTC, 2000, 551p.

### **• LIVRO(S) REFERENCIADO(S)**

1. BOUTH, D. The Practical Xilinx Designer Lab Book. Prentice Hall, 1997, 327p.
2. HAYES, J. P. "Introduction to digital logical design". Addison-Wesley Publishing Company, 1994.
3. HENNESSY, J. L. & PATTERSON, D. A. "Computer architecture: a quantitative approach". Morgan Kaufmann Publishers, Inc, San Francisco, CA, 1996. 2nd. edition.
4. HERZOG, J. H. "Design and organization of computer structures". Franklin, Beedle and Associates, 1996. (transparências de apoio disponíveis na url: <http://www.ece.orst.edu/~herzog/docs.html>).
5. KATZ, R. H. "Contemporary logic design". Benjamin Cummings /Addison-Wesley Publishing Company, 699 páginas, 1994. (livro disponível na url: <http://http.cs.berkeley.edu/~randy/CLD/CLD.html>).
6. MANO, M. M. "Computer system architecture". Prentice-Hall, Englewood Cliffs, NJ, 1993. 525p.
7. MAZOR, S. & LANGSTRAAT, P. "A guide to VHDL". Kluwer Academic Publishers. Norwell, MA, 1992.
8. J. V. OLDFIELD, R. C. DORF. "Field-programmable Gate Arrays: reconfigurable logic for rapid prototyping and implementation of digital systems". Wiley-Interscience, 327p.1995.
9. PATTERSON, D. A. & HENNESSY, J. L. "Computer organization and design: the hardware/software interface". Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1993.
10. Z. SALCIC, A. SMAILAGIC. "Digital systems design and prototyping using field programmable logic". Kluwer Academic, 340p. 1997.
11. ROTH JR., C. H. "Fundamentals of logic design". PWS Publishing Company, 1995.

### **• SOFTWARE DE APOIO**

1. ACTIVE - HDL 3.5 (Simulador da Linguagem de descrição de hardware UHDL)
2. XILINX FOUNDATION - (Ferramenta de síntese de circuitos digitais)