



DISCIPLINA: Organização e Arquitetura de Processadores
CURSOS: Ciência da Computação e Engenharia de Computação
CÓDIGO: 98G03 **CRÉDITOS:** 04 **CARGA HORÁRIA:** 60 horas-aula
VALIDADE: 2021/2, apenas
REQUISITOS:
 Pré-requisito: 4646B-04 Fundamentos de Sistemas Digitais
 Co-requisito: Não há
 Requisito especial: Não há

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

1. Relacionar conceitos básicos de sistemas digitais com os conceitos abstratos subjacentes a arquiteturas e organizações de processadores.
2. Aprender os conceitos fundamentais relacionados a conjuntos de instruções de processadores, à linguagem de montagem destes e à linguagem objeto subjacente.
3. Dominar a programação de um processador usando sua linguagem de montagem.
4. Relacionar o código de programas escritos em linguagens de alto nível com o código em linguagem de montagem equivalente a este em um processador dado.
5. Estender o domínio da arquitetura e organização de um processador à compreensão de conceitos de processadores em geral.
6. Compreender como uma organização de processador executa programas escritos em linguagem de montagem.
7. Distinguir os componentes do bloco de dados e do bloco de controle de um processador quanto a função, composição e estrutura.
8. Ter conhecimentos sobre técnicas básicas usadas na descrição e implementação de organizações de processadores.
9. Compreender um conjunto de conceitos fundamentais úteis para aumentar o desempenho de processadores, incluindo o uso de paralelismo de execução em hardware e aceleradores de hardware específicos.
10. Entender os fundamentos e algumas das abstrações relacionados a operações de entrada e saída e a hierarquias de memória em processadores.

EMENTA:

Apresentação de modelos gerais para arquiteturas e organização de computadores. Introdução aos elementos definidores de uma arquitetura de processador: conjunto

Carimbo e Assinatura da Unidade:

Campus Central

Av. Ipiranga, 6681 – Prédio 30 – CEP: 90619-900

Fone: (51) 3320-3558

E-mail: politecnica@pucrs.br

<https://www.pucrs.br/politecnica/>



de registradores, conjunto de instruções, formatos de instrução, modos de endereçamento, linguagem de montagem, modelo de acesso a memória, programação em linguagem de montagem. Uso de linguagens de descrição de hardware para especificar processadores. Estudo do modelo bloco de dados-bloco de controle para o projeto de processadores. Implementação de módulos de sistemas computacionais de processadores.

Nº DA UNIDADE: 01

Nº DE HORAS EM PERCENTUAL: 10%

CONTEÚDO: O Nível RTL de Abstração e o Modelo Síncrono de Projeto

1.1. Um Estudo de Caso de Projeto Síncrono

1.2. Níveis de Abstração de Projeto

1.2.1. Projeto a nível de transistores

1.2.2. Projeto lógico

1.2.3. Projeto RTL

PROCEDIMENTOS E RECURSOS:

Esta Unidade deve iniciar retomando em 1 aula estudo de caso da disciplina pré-requisito desta (Fundamentos de Sistemas Digitais, 4646B-04) e explorando as características gerais do processo de projeto de circuitos digitais, nas demais aulas da Unidade. Como a ênfase é em circuitos digitais complexos como processadores, deve-se explorar os níveis de abstração de projeto, colocando foco sobretudo no domínio de representações no nível de transferência entre registradores (*register transfer level* ou RTL). Como o projeto síncrono será usado no restante do curso, este deve ser bem caracterizado e detalhado. Isto consiste em apresentar conceitos como o uso de registradores como barreiras temporais, a sensibilidade a borda do sinal de relógio e a caracterização da temporização de projetos síncronos, determinação da frequência máxima de operação etc. Os alunos devem ser informados que o projeto de sistemas digitais complexos é extensamente explorado em disciplina companheira desta (Sistemas Digitais, 98G02-04), obrigatória para alunos da Engenharia de Computação e optativa para alunos da Ciência da Computação. A prática de implementação de tais sistemas é também coberta na disciplina que segue Sistemas Digitais (Construção de Sistemas Digitais, 98G04-02), igualmente obrigatória para alunos da Engenharia de Computação e eletiva para alunos da Ciência da Computação.

Nº DA UNIDADE: 02

Nº DE HORAS EM PERCENTUAL: 25%

CONTEÚDO: Arquitetura de Processadores: Elementos e Linguagem de Montagem

2.1. Abstrações Definidoras de uma Arquitetura de Processador (*ISA*)

2.1.1. Os registradores acessíveis ao programador em linguagem de montagem

Carimbo e Assinatura da Unidade:

Campus Central

Av. Ipiranga, 6681 – Prédio 30 – CEP: 90619-900

Fone: (51) 3320-3558

E-mail: politecnica@pucrs.br

<https://www.pucrs.br/politecnica/>



- 2.1.2. O conjunto de instruções
- 2.1.3. Os formatos de instrução
- 2.1.4. Os modos de endereçamento
- 2.1.5. A linguagem de montagem
- 2.1.6. O modelo de acesso à memória
- 2.2. Programação em Linguagem de Montagem (*Assembly Language*)
 - 2.2.1. Instruções: mnemônicos, rótulos e operandos
 - 2.2.2. Definição de dados
 - 2.2.3. Diretivas de montagem
 - 2.2.4. Pseudo-instruções e macro-instruções
 - 2.2.5. Hierarquização de código em linguagem de montagem: sub-rotinas
 - 2.2.5.1. Chamada e retorno de sub-rotinas
 - 2.2.5.2. Passagem de argumentos via registradores e/ou pilha
 - 2.2.5.3. Retorno de valores via registradores e/ou pilha
- 2.3. Linguagens de Montagem versus Linguagens de Alto Nível

PROCEDIMENTOS E RECURSOS:

Os itens 2.1 e 2.2 desta Unidade deve ser cobertos em sequência, e o item 2.3 perpassa a abordagem do item 2.2. O item 2.1 é a base da Unidade, ele caracteriza uma arquitetura de processador como um conjunto de seis abstrações que interagem para definir a visão arquitetural de um hardware que executa software. O item 2.2, por outro lado, deve ser baseado em um estudo de caso de processador (recomenda-se o uso de um processador real, tal como o MIPS, uma arquitetura RISC) e deve explorar em detalhe a programação deste em linguagem de montagem. Sempre que possível, uma abordagem diferencial deve ser empregada para mostrar como decisões arquiteturais em diferentes processadores levam a diferentes características operacionais, usando o processador alvo de estudo e o comparando com outro(s) processador(es). Uma mera proposta, mas que é interessante para complementar o estudo de caso, é os alunos realizarem trabalho prático onde pesquisem um processador diferente do visto em aula. Para o item 2.3, recomenda-se o uso de uma linguagem de programação de alto nível ou pseudocódigo, desde que a escolha contenha estruturas para definir seleção, laços, expressões aritméticas etc.

Nº DA UNIDADE: 03

Nº DE HORAS EM PERCENTUAL: 25%

CONTEÚDO: Organização de Processadores: Fundamentos e Estudo de Caso

- 3.1. Tipos de Organizações Sequenciais de Processadores
 - 3.1.1. Organizações monociclo versus organizações multiciclo
 - 3.1.2. Organizações Harvard e von Neumann
 - 3.1.2.1. Organizações híbridas
- 3.2. Implementação de Organizações Sequenciais Multiciclo

Carimbo e Assinatura da Unidade:

Campus Central

Av. Ipiranga, 6681 – Prédio 30 – CEP: 90619-900

Fone: (51) 3320-3558

E-mail: politecnica@pucrs.br

<https://www.pucrs.br/politecnica/>



- 3.2.1. A interface processador-memória e sua operação
- 3.2.2. Bloco de dados (*data path*)
 - 3.2.2.1. Banco de registradores
 - 3.2.2.2. ULA
 - 3.2.2.3. Registradores de apoio e módulos auxiliares
- 3.2.3. Bloco de controle (*control unit*)
 - 3.2.3.1. Decodificação de instruções
 - 3.2.3.2. Máquina de estados de controle
 - 3.2.3.3. Registradores de controle e módulos auxiliares

PROCEDIMENTOS E RECURSOS:

A Unidade deve centrar esforços no estudo de organizações de processadores, explorando em alguma profundidade o projeto de uma organização para um estudo de caso de arquitetura de processador. Recomenda-se usar um subconjunto da arquitetura de processador estudada na Unidade 2. O item 3.1 desta Unidade deve explorar classificações de organizações segundo pelo menos os dois critérios listados aqui, mas outras podem ser abordadas. A essência da Unidade reside no item 3.2, que deve evoluir pela transcrição dos conceitos arquiteturais em hardware que implementa estes. Isto deve ocorrer via a apresentação de diagramas de blocos pelo professor, diagramas estes feitos equivalentes a descrições RTL simuláveis, conceitos abordados na Unidade 1. Partindo de um subconjunto muito restrito de arquitetura, deve-se compreender o hardware de base. O professor mostra e faz evoluir este hardware para acomodar novas instruções e estruturas para dar suporte a uma versão mais abrangente da arquitetura de base. Deve-se reforçar a importância de identificar e tratar separadamente os fluxos de informação de controle e de dados. O emprego do modelo Bloco de Dados/Bloco de Controle deve ser indicado como a chave para esta última tarefa.

Nº DA UNIDADE: 04

Nº DE HORAS EM PERCENTUAL: 20%

CONTEÚDO: Organização de Processadores: Paralelismo Básico e Aceleradores

- 4.1. Paralelismo Básico: a execução superposta de instruções
 - 4.1.1. Organizações pipeline
- 4.2. Aceleradores para Processamento Numérico
 - 4.2.1. Representações numéricas de ponto fixo e ponto flutuante
 - 4.2.2. O padrão IEEE-754
 - 4.2.3. Suporte a ponto flutuante em processadores

PROCEDIMENTOS E RECURSOS:

Após cobrir a parte essencial do curso nas Unidades 2 e 3, esta Unidade deve trazer uma primeira exploração de dois conceitos essenciais para obter processadores

Carimbo e Assinatura da Unidade:

Campus Central

Av. Ipiranga, 6681 – Prédio 30 – CEP: 90619-900
Fone: (51) 3320-3558
E-mail: politecnica@pucrs.br
<https://www.pucrs.br/politecnica/>



modernos com desempenho aceitável. O item 4.1 deve esboçar rudimentos de paralelismo, sobretudo a nível da execução de instruções (*instruction level parallelism* ou ILP), com a apresentação do conceito de *pipelines* de instruções. Isto deve tomar cerca de dois terços do tempo da Unidade. O item 4.2 então deve explorar como se acelera em hardware o processamento numérico, sobretudo para representações de números racionais. Isto se faz estudando representações de ponto fixo e ponto flutuante, com ênfase nas últimas. O padrão IEEE-754 deve ser apresentado em linhas gerais. A última parte do item consiste em brevemente apresentar como trabalhar com números em ponto flutuante em linguagem de montagem para manipular as representações estudadas.

Nº DA UNIDADE: 05

Nº DE HORAS EM PERCENTUAL: 20%

CONTEÚDO: Arquitetura de Processadores: Tópicos Fundamentais Adicionais

5.1. Entrada e Saída em Processadores (E/S)

5.1.1. Interface entre programas e dispositivos

5.1.2. E/S programada

5.1.2.1. Bloqueada

5.1.2.2. *Polling*

5.1.2.3. Interjeição

5.1.3. E/S não programada – interrupções

5.1.4. Acesso direto à memória (*direct memory access* ou DMA)

5.2. Hierarquia de Memória em Processadores

5.2.1. Justificativa, conceitos e características fundamentais

5.2.2. Memórias escondidas (*cache memories*)

5.2.2.1. Conceitos, características e organização

5.2.2.2. Mapeamento de endereços

PROCEDIMENTOS E RECURSOS:

Esta Unidade final do curso deve trazer conceitos essenciais de processadores para a definição da interface destes com o restante do sistema que ele sempre integra. O item 5.1 cobre operações de entrada e saída, explorando a interface entre hardware e software que habilita o processador a enviar e receber dados de/para seu exterior. A ênfase é nos tipos possíveis de interação processador-mundo externo. O item 5.2, por outro lado, deve explorar os rudimentos de hierarquias de memória, iniciando com a justificativa de sua existência (a disparidade de velocidades entre o processador e os diversos tipos de meios de armazenamento de informação), seguindo para a noção de memórias “escondidas” (memórias *cache*) geridas de forma automática por hardware dedicado. Deve-se enfatizar que o conteúdo desta Unidade, bem como conceitos adicionais de paralelismo são explorados em maior profundidade na disciplina que segue esta, Arquitetura e Organização de Computadores (98G05-04),

Carimbo e Assinatura da Unidade:

Campus Central

Av. Ipiranga, 6681 – Prédio 30 – CEP: 90619-900

Fone: (51) 3320-3558

E-mail: politecnica@pucrs.br

<https://www.pucrs.br/politecnica/>



obrigatória para alunos da Engenharia de Computação e optativa para alunos da Ciência da Computação.

AVALIAÇÃO:

$$G1 = 0,3 * P1 + 0,3 * P2 + 0,4 * TP$$

ONDE:

P1 – Prova 1, abrange as unidades 1 e 2;

P2 – Prova 2, abrange as unidades 3, 4 e 5;

TP – Nota envolvendo diversos trabalhos práticos desenvolvidos ao longo da disciplina.

BIBLIOGRAFIA:

• **BÁSICA:**

1. PATTERSON, D.; HENNESSY, J. Organização e projeto de computadores: a interface hardware/software. 4a ed., Elsevier, 2014, 709p.
2. STALLINGS, W. **Arquitetura e organização de computadores**. 8ª ed., Pearson, 2010, 625p.
3. HENESSY, J; PATTERSON, D. **Arquitetura de computadores: uma abordagem quantitativa**. 5ª ed., Campus, 2014, 744p.

• **COMPLEMENTAR:**

1. TANENBAUM, A.; AUSTIN, T. **Organização estruturada de computadores**. 6a ed., Capítulo 8 - Arquitetura de Computadores Paralelos, pp. 436-518, Pearson, 2013.
2. BAER, J.-L. **Arquitetura de Microprocessadores – do Simple Pipeline ao Multiprocessador em Chip**. 1ª ed., LTC, 2013, 326p.
3. MONTEIRO, M. **Introdução à organização de computadores**. 5a ed., LTC, 2012, 686p.
4. PEDRONI, V. **Eletrônica digital moderna e VHDL**. 1a ed., Campus, 2010, 648p.
5. PARHAMI, B. **Computer Architecture: From Microprocessors to Supercomputers**. Oxford University Press, 2014, 576p.

SOFTWARE DE APOIO:

Carimbo e Assinatura da Unidade:	Campus Central Av. Ipiranga, 6681 – Prédio 30 – CEP: 90619-900 Fone: (51) 3320-3558 E-mail: politecnica@pucrs.br https://www.pucrs.br/politecnica/
Página 6 de 7 Emitido em: 04/08/21 - 08:22	



1. MARS (<http://courses.missouristate.edu/kenvollmar/mars>) ou similar, i.e. ambientes de suporte a captura e simulação de programas escritos em linguagem de montagem de um processador selecionado.
2. Vivado, ModelSim ou similares, i.e. ambientes de captura, simulação e/ou prototipação de circuitos e sistemas digitais descritos em linguagem de descrição de hardware.

Carimbo e Assinatura da Unidade:

Campus Central

Av. Ipiranga, 6681 – Prédio 30 – CEP: 90619-900

Fone: (51) 3320-3558

E-mail: politecnica@pucrs.br

<https://www.pucrs.br/politecnica/>