

## **SSC0113 - Elementos de Lógica Digital II**

### **Apresentação da Disciplina**

Professores: Eduardo do Valle Simões e Eduardo Marques/Alexandre Delbem

Email: simoes , emarques , acdb ...@icmc.usp.br

Departamento de Sistemas de Computação – ICMC - USP

Grupo de Sistemas Embarcados, Evolutivos e Robóticos

Laboratório de Robótica Móvel -- Laboratório de Computação Reconfigurável

**Site da disciplina:** <http://wiki.icmc.usp.br/index.php/SSC-113>

### **Elementos de Lógica Digital II**

Créditos Aula: 5

Créditos Trabalho: 0

### **Objetivos**

Estender o conhecimento do aluno em técnicas digitais, com ênfase no projeto de circuitos seqüenciais, bem como no projeto e implementação da arquitetura básica de processadores e sua programação em linguagens de máquina.

### **Programa Resumido**

Revisão de circuitos combinacionais. Projeto de lógica seqüencial e casos de estudos. Projeto de máquinas de estado finito - FSM (Finite State Machine). Implementação de máquinas de estado finito, utilizando-se blocos esquemáticos e linguagem de descrição de hardware em circuitos de lógica programável tipo FPGA (Field-Programmable Gate Array). Desenvolvimento e implementação dos elementos básicos que constituem a arquitetura de processadores, incluindo unidade lógica aritmética, unidade de controle, entrada e saída, banco de registradores e memória de dados e instruções. Programação em Linguagens Montadoras.

### **Avaliação**

- Serão realizados dois projetos (P1 e P2) sobre máquinas de estado e processadores, respectivamente.
- As aulas de laboratório envolverão a implementação de vários trabalhos práticos (Tn) avaliados individualmente ao final de cada aula prática.
- Não serão aprovados alunos com nota nos projetos ou média dos trabalhos práticos inferior a cinco.
- Os projetos P1 e P2 podem ser realizados em grupo de até 3 alunos.
- A avaliação dos projetos será composta por duas notas, nota do grupo (P) e nota individual (Arguição). A nota final  $P_i$  do aluno no projeto será calculada conforme a equação:  $P_i = (P * Arguição)/10$ .
- A nota final (NF) será calculada da seguinte maneira:  $NF = 0.3P1i + 0.4P2i + 0.3 * Média dos Tn$ .

**- Não tem SUB !**

Norma de Recuperação

Nota Final:  $= (NP-2) / 5 * Mrec + 7 - NP$  se  $Mrec \geq 5$ ; ou

$\text{Max}(NP, Mrec)$  se  $Mrec < 5$

Sendo NP = Nota da 1a Avaliação e

MRec = Média da Recuperação

### **Aulas Programadas**

- \* Aula 1 - Apresentação do curso
- \* Aula 2 - Linguagem de descrição de hardware VHDL
- \* Aula 3 - Máquina de estados finitos
- \* Aula 4 - Modelos de máquinas de estados finitos - Moore e Mealy (exercícios)  
o Máquina de venda de refrigerante
- \* Aula 5 - Máquina de estados finitos (continuação)
- \* Aula 6 - Apresentação P1 (06/08/2011)
- \* Aula 7 - Template de microprocessador (memória, mapa de caracteres, I/O, ULA, unidade de controle)
- \* Aula 8 - Análise e implementação: instruções de acesso a memória e I/O
- \* Aula 9 - Análise e implementação: instruções de operações aritméticas, lógicas e de deslocamento
- \* Aula 10 - Análise e implementação: instruções de desvios
- \* Aula 11 - Análise e implementação: instruções de subrotinas
- \* Aula 12 - Montador
- \* Aula 13 - Programação do P2 no Processador
- \* Aula 14 - Programação do P2 no Processador
- \* Aula 15 - Apresentação P2 (29/11/2011)

### **Bibliografia**

- PATTERSON, D.A.; HENNESSY, J.L. Computer Organization and Design: The Hardware/Software Interface, Morgan Kaufmann, 1994.
- TANENBAUM, A.S. Structured Computer Organization, Prentice Hall, 4th ed, 1999.
- MONTEIRO, M.A. Introdução à Organização de Computadores, 3a ed. Livros Técnicos e Científico Editora SA, 1996.
- BROWN, S.; VRANESIC, Z. Fundamentals of Digital Logic with VHDL Design, 2a ed, Mc Graw Hill, 2005.
- KATZ, R.H.; BORRIELLO, G. Contemporary Logic Design, 2a ed, Prentice Hall, 2005.

### **Bibliografia Complementar**

- HARRIS, D.; HARRIS. S. Digital Design and Computer Architecture, Morgan Kaufmann, 2007.

### **Programa da Disciplina:**

A seguir, a ementa da disciplina é apresentada com uma descrição da maneira com que cada tópico será abordado e dos capítulos dos Livros Texto (PATTERSON – 3a ed, 2003; MONTEIRO - 5a ed, 2007; BROWN - 2a ed, 2005; KATZ – 2a ed, 2005) que contém cada tópico:

1) Revisão de conceitos sobre a implementação de circuitos combinacionais e lógica seqüencial.

- As aulas iniciais devem realizar uma revisão de todas as estruturas necessárias para a disciplina: multiplexadores, registradores, memórias (construção, barramentos, operação), ULA, barramentos e interconexões.

BROWN Cap. 2.1 a 2.7, 4.1 a 4.7, 6.1 a 6.5, 7.1 a 7.11 e 7.14, 10.1 (2a ed, 2005)

## 2) Projeto de máquinas de estado finito - FSM (Finite State Machine).

- A primeira fase da disciplina irá abordar a implementação de máquinas de estado finito, utilizando-se blocos esquemáticos e linguagem de descrição de hardware em circuitos de lógica programável tipo FPGA (Field-Programmable Gate Array).
- Será apresentada a linguagem VHDL e sua utilização na ferramenta Quartus II, utilizado-se placas de FPGA para implementação das máquinas de estado estudadas.
- Serão abordadas nas aulas teóricas a descrição de máquinas de Moore e Mealy.
- As aulas práticas irão abordar a implementação de vários exemplos das máquinas de Moore e Mealy, como por exemplo, o projeto de um alarme de automobilístico, o controlador de uma máquina de refrigerante, e o controlador de um vídeo cassete player.
- O primeiro projeto (P1) deverá ser a implementação de um jogo simples utilizando máquinas de estado finito.

BROWN Cap. 8.1 a 8.5 (2a ed, 2005)

KATZ Cap. 7.1 a 7.4 (2a ed, 2005)

## 3) Desenvolvimento e implementação de uma arquitetura de processador do tipo RISC de acordo com a arquitetura de Von Neumann.

- As aulas teóricas seguintes apresentam o projeto incremental de um processador contendo ULA, Máquina de Controle, Memória, registradores e dispositivos de I/O
- As aulas práticas servirão de revisão da implementação destes sub-sistemas e suas interconexões, envolvendo a implementação incremental do conjunto de instruções deste processador (ex.: ADD, LOAD, STORE, CALL, JUMP, PUSH, PULL, SHIFT ...).

- PATTERSON Cap. 1.1 a 1.3, 3.1 a 3.5 (3rd ed, 2003)

- MONTEIRO Cap. 1.1.3, 3.3.1.2, 3.4.1, 4.2.1, 4.2.3, 4.3.3, 6.2.1, 6.2.2, 6.3, 6.4, 6.5., E.1 (5a ed, 2007)

- Estes pontos serão vistos em uma seqüência de várias aulas, nas quais o processador e a unidade de controle serão construídos progressivamente para executar o conjunto de instruções, uma a uma, revendo então seus fundamentos, desenvolvimento e implementação.

- PATTERSON Cap. 2.1 a 2.8, 5.1 a 5.5, 5.7, 7.1 a 7.2 (3rd ed, 2003)

- MONTEIRO Cap.8.1 a 8.5, 11.1, 11.2, 11.3.1, 11.3.2, 11.3.3 (5a ed, 2007)

## 4) Programação em Linguagens Montadoras.

- Serão apresentados exemplos de utilização de montadores para converter linguagem montadora em linguagem de máquina, apresentando exemplos simples de transformação de linguagem C em linguagem montadora, como, por exemplo, inicialização de variáveis ("Int A = 5;"), Loops (For), e condicional (Se-Entao).

- Também serão vistos vários exemplos de desenvolvimento de programas em linguagem Assembly realizando diversas funcionalidades, inclusive E/S.

- O segundo projeto P2 irá abordar a implementação de um jogo simples utilizando o conjunto de instruções do processador desenvolvido.

- PATTERSON Cap. 2.5 a 2.8, 2.10, 2.13, 2.15 (3rd ed, 2003)

- MONTEIRO Cap. C.2, C.3, C.6 (5a ed, 2007)

## **Metodologia:**

- As aulas serão principalmente expositivas explicando-se na lousa os conteúdos do livro. Sendo assim, a cópia do material dado na lousa possibilitará a obtenção de uma síntese bem exemplificada do conteúdo do livro.
- Serão utilizadas transparências, apresentando os principais tópicos de máquinas de estados finitos e arquitetura de processadores e sua implementação em linguagem VHDL.
- Será disponibilizada uma ampla lista de exercícios sobre toda a matéria e marcada uma aula específica para tirar as dúvidas dos alunos sobre os exercícios.
- Será disponibilizada uma ferramenta que é um simulador de operação de um processador semelhante ao que será apresentado e implementado em aula, desenvolvida por alunos do ICMC, incluindo os códigos (feitos em linguagem C no Visual Studio) do algoritmo que simula os módulos do processador, o algoritmo da máquina de controle e um montador para os programas em linguagem de máquina. Com esta ferramenta, os alunos podem editar o processador, incluir novas instruções, reprogramar a máquina de controle para controlar estas novas instruções e editar o montador para montá-las para o código de máquina do processador.
- Será também apresentada a linguagem VHDL na ferramenta Quartus II e a utilização das placas de FPGA para a implementação de processadores próprios dos alunos. Serão fornecidas transparências com a síntese desta metodologia para os alunos.
- Alguns alunos serão convidados a apresentar implementações para seus colegas utilizando as ferramentas apresentadas para implementar software em linguagem montadora, bem como os que implementarem processadores próprios em software com o simulador ou em hardware na placa de FPGA.
- Todo este material será disponibilizado diretamente aos alunos durante as aulas e através do site da disciplina (<http://wiki.icmc.usp.br/index.php/SSC-113>).