



UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CAMPUS ARARANGUÁ - ARA  
CURSO DE ENGENHARIA DE COMPUTAÇÃO  
PLANO DE ENSINO

SEMESTRE 2014.2

**I. IDENTIFICAÇÃO DA DISCIPLINA:**

CÓDIGO	NOME DA DISCIPLINA	Nº DE HORAS-AULA SEMANAIS		TOTAL DE HORAS-AULA SEMESTRAIS
		TEÓRICAS	PRÁTICAS	
ARA7555	Linguagens de Descrição de Hardware	4	0	72

HORÁRIO		MÓDULO
TURMAS TEÓRICAS	TURMAS PRÁTICAS	Presencial
07655		

**II. PROFESSOR(ES) MINISTRANTE(S)**

Prof. Fábio Rodrigues de la Rocha, Dr.  
Fabio.rocha.ufsc@gmail.com

**III. PRÉ-REQUISITO(S)**

CÓDIGO	NOME DA DISCIPLINA
	Esta disciplina não possui pré-requisitos

**IV. CURSO(S) PARA O(S) QUAL(IS) A DISCIPLINA É OFERECIDA**

Engenharia de Computação

**V. JUSTIFICATIVA**

Esta disciplina deverá explorar conceitos e técnicas em circuitos digitais e motivar a realização de projeto extraclasse em ambiente de laboratório. Também irá induzir o aluno à realizar tais projetos em um ambiente que se utilize de linguagem de descrição de hardware

**VI. EMENTA**

Desenvolvimento de projetos em Alto Nível através de Linguagens de Descrição de Hardware (VHDL, Verilog), Máquina Finita de Estados, RTL (RegisterTransferLevel), em dispositivos como Field Programmable Gate Array (FPGA). São considerados Co-Projetos de Hardware/Software em ambientes System-on-Chip enfocando CORE e IP para o re-uso de sistemas. Para esses estudos são considerados os usos das ferramentas EDA (Eletronic Design Automation) da Xilinx e Altera. Considerações sobre: co-projeto de hardware/software; engenharia de software para o sistema; questões de sincronização de clock; protocolo de comunicação; escalonamento; RTOS (Real Time Operating System); validação e verificação; tolerância à falhas; programando sensores e atuadores; simulação, ferramentas EDA, ambiente distribuído.

**VII. OBJETIVOS**

**Objetivo Geral:**

- Compreender o funcionamento de uma FPGA e o ciclo de desenvolvimento de Hardware utilizando esse tipo de dispositivo.
- Compreender a capacidade desses dispositivos e as possibilidades de integração de Hard Cores, Soft Cores e código personalizado.
- Escrever código em linguagem de descrição de hardware, utilizar ferramentas de desenvolvimento e simulação
- Criar um pequeno projeto de dispositivo que explore os recursos de uma FPGA

**VIII. CONTEÚDO PROGRAMÁTICO**

**UNIDADE 1: FPGA [24 horas-aula]**

- FPGA, estrutura interna
- FPGA, ferramentas de desenvolvimento e simulação
- Ciclo de desenvolvimento, ferramenta de síntese de hardware, uso como ferramenta de prototipação
- Processadores implementados em hardware e em software, reconfiguração dinâmica
- mercado de FPGAs, empresas fabricantes, mercado de atuação
- propriedade intelectual de componentes de hardware

#### **UNIDADE 2: Programação para FPGA [12 horas-aula]**

- Linguagens existentes
- Código sintetizável e código não sintetizável
- Linguagem de programação VHDL
- Linguagem de programação Verilog

#### **UNIDADE 3: Estudo da linguagem VHDL [08 horas-aula]**

- Formato da linguagem, escrita de código básico
- Recursos avançados da linguagens
- Criação de código, simulação usando ambiente de desenvolvimento ALTERA e XILINX

#### **UNIDADE 4: Implementação de dispositivos em VHDL [16 horas-aula]**

- portas lógicas, latches, Flipflops, circuitos compostos por diferentes elementos
- decodificadores, mux, demux, decodificadores 7 segmentos
- Uso de decodificadores para habilitar partes de um circuito
- implementação de protocolos de comunicação rs232, I2C, endereçamento de dispositivos
- máquinas de estado em VHDL, máquinas moore, máquinas mealy

#### **UNIDADE 5: Estudo da linguagem Verilog [04 horas-aula]**

- Formato da linguagem, estrutura de decisão, repetição

### **IX. METODOLOGIA DE ENSINO / DESENVOLVIMENTO DO PROGRAMA**

- Aulas teóricas: desenvolvidas em sala e com emprego de meios audiovisuais tais como transparências e apresentações sobre PC portátil de produção própria expostas com projetor. Todo o material didático estará disponível "a priori" para os alunos na página do professor: [fabiodelarocha.paginas.ufsc.br](http://fabiodelarocha.paginas.ufsc.br)

### **X. METODOLOGIA E INSTRUMENTOS DE AVALIAÇÃO**

A verificação do rendimento escolar compreenderá frequência e aproveitamento nos estudos, os quais deverão ser atingidos conjuntamente. Será obrigatória a frequência às atividades correspondentes a cada disciplina, no mínimo a 75% das mesmas (Frequência Suficiente - FS), ficando nela reprovado o aluno que não comparecer a mais de 25% das atividades (Frequência Insuficiente - FI).

Serão realizadas duas avaliações :

- Prova Escrita 1 (P1) , peso 6
- Trabalho prático T1, peso 4

A média final do semestre será a própria média ponderada da prova e do trabalho e assim a nota mínima para aprovação na disciplina será **MF>=6,0** (seis) e Frequência Suficiente (FS). (Art. 69 e 72 da Res. nº 17/CUn/1997).

O aluno com Frequência Suficiente (FS) e média das notas de avaliações do semestre MF entre 3,0 e 5,5 NÃO terá direito a uma nova avaliação no final do semestre (REC) de acordo com o art.70, § 2º. A Nota Final (NF) será calculada por meio da média aritmética entre a média das notas das avaliações parciais. (Art. 70 e 71 da Res. nº 17/CUn/1997).

Ao aluno que não comparecer às atividades práticas ou não apresentar trabalhos no prazo estabelecido será atribuída nota 0 (zero). (Art. 70, § 4º da Res. nº 17/CUn/1997)

**Observações:**

**Avaliação de recuperação**

Não há avaliação de recuperação nas disciplinas de **caráter prático** que envolve atividades de laboratório. (Res.17/CUn/97).

**Nova avaliação**

Pedidos de segunda avaliação somente para casos em que o aluno, por motivo de força maior e plenamente justificado, deixar de realizar avaliações previstas no plano de ensino, e deverá ser formalizado via requerimento de avaliação à Secretaria Acadêmica do Campus Araranguá dentro do prazo de 3 dias úteis apresentando comprovação. (Ver formulário)

**XI. CRONOGRAMA TEÓRICO**

<b>AULA (semana)</b>	<b>DATA</b>	<b>ASSUNTO</b>
1 <sup>a</sup>	11/08 – 15/08	Apresentação do plano de ensino, site da disciplina, Introdução à FPGA, visão geral da disciplina. Hardware concorrente, reconfiguração dinâmica.
2 <sup>a</sup>	18/08 – 23/08	Introdução VHDL, Ciclo de desenvolvimento, ferramenta de síntese de hardware, uso como ferramenta de prototipação ferramentas de simulação, ferramentas de desenvolvimento ALTERA (windows) escrita de pequenos dispositivos, teste, simulação. Hardware configurável, processadores em hardware (Power PC) e em software (Soft-core), existência de softcores para implementar microcontroladores, caso da ARM, sistemas operacionais voltados para hardware embarcado
3 <sup>a</sup>	25/08 – 30/08	Introdução VHDL, ferramentas de desenvolvimento e simulação, exemplos, entidade e arquitetura
4 <sup>a</sup>	01/09 – 06/09	Ferramenta de desenvolvimento Xilinx (windows), criação de código, síntese e simulação. Instalação da ferramenta Xilinx para linux. Sinais, variáveis, tipos de dados, direção de dados (entrada, saída, entrada-saída), vetores
5 <sup>a</sup>	08/09 – 13/09	Criação de projetos, implementação de portas lógicas, latches, FlipFlops, MUX, decodificadores, memória RAM, demux, somadores (half adder), somadores (full adder)
6 <sup>a</sup>	15/09 – 20/09	decodificadores 7 segmentos, protocolos de comunicação em paralelo, protocolos de comunicação serial RS232, I2C, protocolos alternativos SPI. Estudo de um CI real (memória I2C).
7 <sup>a</sup>	22/09 – 27/09	Criação de um sistema de transmissão de mensagens seriais. Trabalho com vários módulos (uso de componentes e vários arquivos fonte)
8 <sup>a</sup>	29/09 – 04/10	Máquinas de estado: revisão sobre o tema. Máquinas de estado em VHDL, máquina do tipo MOORE. Resolução de problemas usando máquinas de estado: máquina de refrigerante,
9 <sup>a</sup>	06/10 – 11/10	Sinalização de freio de carro, máquina de lavar, detector de sequencias. Máquina de estados do tipo Mealy, comparação entre as soluções usando os dois tipos de máquinas. Discussão sobre clock de hardware e divisão de clock para ser usado nos projetos.
10 <sup>a</sup>	13/10 – 18/10	PROVA P1, uso de funções e procedimentos, visão RTL do projeto. Correção da prova, apresentação da linguagem Verilog.
11 <sup>a</sup>	20/10 – 25/10	Verilog, apresentação da linguagem, uso do ambiente de desenvolvimento para escrever código.
12 <sup>a</sup>	27/10 – 01/11	Escrita de código em verilog
13 <sup>a</sup>	03/11 – 08/11	Uso de bibliotecas de componentes
14 <sup>a</sup>	10/11 – 15/11	Apresentação do trabalho (que será avaliado em defesa posterior), escrita de código em verilog. Comparar resultado da síntese em VHDL e Verilog
15 <sup>a</sup>	17/11 – 22/11	Portas lógicas, mux, latches, FF, decodificadores, etc.
16 <sup>a</sup>	24/11 – 29/11	Memória RAM, interface paralela e serial
17 <sup>a</sup>	01/12 – 06/12	Máquinas de estado em verilog, moore e mealy
18 <sup>a</sup>	08/12 – 13/12	Defesa de trabalho T1 Prova REC Fechamento das notas

**XII. Feriados previstos para o semestre 2014.2:**

DATA	

**XIII. BIBLIOGRAFIA BÁSICA**

- 1] Hamblen, J.O. e Furman M.D. - Rapid Prototyping of Digital Systems - A tutorial approach; kluver Academic Publisher, 2001.
- [2] Bhasker - VHDL Primer, Prentice Hall, 1999.
- [3] Pedroni - Circuit Design with VHDL; MIT Press, 2005.


**XIV. BIBLIOGRAFIA COMPLEMENTAR:**

- [4] D'Amore, R. - VHDL: Descrição e Síntese de Circuitos Digitais, LTC, 2005
- [5] Pong P. Chu, "FPGA Prototyping by VHDL Examples: Xilinx Spartan-3 Version", Wiley, 2008.
- [6] Andrew Rushton, "VHDL for Logic Synthesis", Wiley, 2011.
- [7] Steve Kilts, "Advanced FPGA Design: Architecture, Implementation, and Optimization", Wiley, 2007.
- [8] Richard Sandige, Michael Sandige, "Fundamentals of Digital and Computer Design with VHDL", McGraw-Hill, 2011.

Os livros acima citados constam na Biblioteca Universitária e Setorial de Araranguá. Algumas bibliografias também podem ser encontradas no acervo da disciplina, via sistema Moodle.

Prof. Fábio Rodrigues de la Rocha

Aprovado na Reunião do Colegiado do Curso \_\_\_/\_\_\_/\_\_\_

  
Prof. Dr. Eugênio Simão  
Coordenador do Curso de Graduação  
em Engenharia de Computação  
SIAPE: 392745 Portaria nº 1071