**Universidade Federal de Santa Catarina**
**Centro Tecnológico**
**Departamento de Informática e Estatística**

# Programa de Ensino

## 1) Identificação

**Disciplina:** INE5350 - Arquitetura de Computadores II
**Carga horária:** 54 horas-aula Teóricas: 54 Práticas: 0
**Período:** início da oferta da disciplina até a presente data

## 2) Cursos

- Ciências da Computação (208)

## 3) Requisitos

- Ciências da Computação (208)
  - INE5346 - Integração Software/Hardware

## 4) Ementa

Máquinas de pilha. Computadores paralelos. Overlap e processamento paralelo. Tópicos especiais.

## 5) Objetivos

**Geral:** Descrever os mecanismos de hardware e técnicas de software para a aceleração de programas em sistemas computacionais e introduzir noções quantitativas quanto ao impacto dos recursos de hardware no desempenho de programas.

**Específicos:**

- Revisar o conceito de execução “pipelining” e estudar os “hazards” que reduzem seu desempenho.
- Introduzir técnicas para detecção e utilização de paralelismo existente entre as instruções de um programa (“instruction-level parallelism”).
- Introduzir o “estado da arte” das técnicas de exploração de paralelismo usadas em processadores RISC e super-escalares contemporâneos.
- Introduzir noções de requisitos e benchmarks para os mercados desktop e computadores embutidos.

## 6) Conteúdo Programático

6.1) Revisão de desempenho em computadores [6 horas-aula]
- Medida de desempenho.
- Programas para avaliar desempenho (“benchmarks”).
- Comparação de desempenho.
- Princípios quantitativos no projeto de sistemas de computação.
- Requisitos dos mercados desktop, servidores e computadores embutidos.

6.2) Arquitetura do conjunto de instruções [9 horas-aula]
- Classificação.
- Freqüência de uso de modos de endereçamento.
- Freqüência no uso de instruções.
- Freqüência do tamanho de operandos.
- Aspectos da codificação do conjunto de instruções.
- Revisão do papel do compilador.

- Introdução de um exemplo típico de arquitetura contemporânea: o MIPS64.

6.3) Paralelismo em nível de instrução e sua exploração dinâmica. [18 horas-aula]
- Conceito de "instruction-level parallelism" (ILP)
- Dependências.
- Escalonamento Dinâmico.
- Previsão dinâmcia de desvios.
- Entrega de instruções de alto desempenho.
- Emissão múltipla.
- Especulação.
- Exemplo real: a microarquitetura P6 (base das implementações IA-32 contemporâneas)

6.4) Explorando o paralelismo com abordages de software [21 horas-aula]
- Loop unrolling.
- Previsão estática de desvios.
- Emissão múltipla: abordagem VLIW.
- Suporte avançado de compilador para expor e explorar o ILP: software pipelining.
- Escalonamento global de código.
- Escalonamento de traçado.
- Suporte de HW para expor mais ILP em tempo de compilação.
- Exemplo real: a arquitetura IA-64 e o processador Itanium.

**7) Bibliografia Básica**

- John L. Hennessy and David A. Patterson, “Computer Architecture: A Quantitative Approach”, 3rd edition, Morgan Kaufmann Publishers, San Francisco, California, 2002.

**8) Bibliografia Complementar**

- David A. Patterson and John L. Hennessy, “Computer Organization and Design: The Hardware/Software Interface”, 3rd edition, Morgan Kaufmann Publishers, San Francisco, California, USA, 1998.