



UNIVERSIDADE ESTADUAL DE FEIRA DE SANTANA  
DEPARTAMENTO DE TECNOLOGIA/ÁREA DE ELETRÔNICA E SISTEMAS  
TEC499 – MI – SISTEMAS DIGITAIS

## Problema #1 – Debaixo do Capô

### Tema

Linguagem de Máquina

### Objetivos de Aprendizagem

Ao final da realização deste problema você deve ser capaz de:

- Programar em linguagem de baixo nível (Assembly);
- Identificar conjunto de instruções em processadores;
- Compreender o fluxo de execução interna de um mono-processador RISC;

### 1 Contexto

Grande parte dos sistemas computacionais modernos lidam com alto volume de processamento de dados envolvendo busca, classificação e operações aritméticas compostas. Compiladores modernos são capazes de gerar padrões de códigos otimizados, de acordo com as características da linguagem de alto nível e através de parâmetros de compilação. Todavia, ao lidar com um sistema computacional, dotado de dezenas de unidades de coprocessamento e periféricos de entrada e saída, um projetista pode, facilmente, perder o controle dos parâmetros de desempenho do seu sistema.

Sob o ponto de vista da implementação de sistemas computacionais, destacam-se duas abordagens que diferem no que diz respeito a desempenho, potência e flexibilidade: **circuitos dedicados** (ASIC - *Application Specific Integrated Circuit*) e **dispositivos lógicos programáveis** (PLD - *Programmable Logic Devices*). Atualmente o projeto de sistemas baseados em PLDs vem sendo bastante difundido, principalmente em função da evolução do processo de fabricação dos *chips* FPGA (*Field Programmable Gate Array*). Dispositivos FPGA modernos são capazes de implementar circuitos complexos, equivalentes a mais de 100.000 portas lógicas e tem, cada vez mais, substituído projetos baseados em *standard cells*.

Existem dois grandes fabricantes de *chips* FPGA atualmente: Xilinx e Intel FPGA (ex ALTERA). Dispositivos Xilinx são capazes de implementar instâncias do seu processador MicroBlaze. Por outro lado, a Intel FPGA incorpora em seus dispositivos o NIOS. Ambas as soluções consistem de unidades de processamento de propósito geral RISC configuráveis e otimizadas para implementação em seus respectivos dispositivos.

### 2 Problema

A empresa **IP-SoC** atua no desenvolvimento de *IP-cores* digitais licenciáveis para projeto de sistemas computacionais há mais de dez anos. Durante sua trajetória de mercado, vários produtos foram desenvolvidos nas áreas de processamento gráfico, microprocessadores de propósito geral e processamento digital de sinais. Nos últimos anos, a empresa tem se destacado no mercado de semicondutores com a venda dos seus *cores* licenciáveis.

Atenta às mudanças de paradigma no projeto de sistemas, a IP-SoC está de olho no mercado de plataformas embarcadas projetadas a partir de dispositivos FPGA. Neste sentido, a empresa começou uma força tarefa para apropriação do conhecimento acerca das soluções tecnológicas que podem ser exploradas com o uso desse tipo de plataforma, a começar pelas opções de núcleos de processamento geral.

Duas equipes de projeto foram destacadas para investigar as duas tecnologias de processadores para dispositivos reconfiguráveis. Diante deste cenário, sua equipe é aquela que deve apresentar informações técnicas

sobre o projeto de sistemas em FPGAs ALTERA, usando o processador NIOS. Para isso, além de conhecer o funcionamento básico do processador, você deve avaliar o nível de complexidade de programação e limitações do dispositivo, a partir do desenvolvimento de uma série de rotinas de códigos, usando a linguagem assembly associada a este processador. Para isso, contará com o software *open source* JNIOSEmu, disponível online.

### 3 Produto

No prazo indicado no cronograma apresentado neste documento, a sua equipe deverá apresentar:

1. Programas de teste:

- (a) Gerador da Sequência de Fibonacci (recursivo)
- (b) Ordenação de um vetor, com tamanho variável, usando o algoritmo Bubble Sort
- (c) Cálculo do fatorial de um número inteiro (recursivo)
- (d) Geração de números primos entre dois valores dados
- (e) Teste se um dado ano é ou não bissexto

Os programas devem considerar que a entrada e saída de parâmetros será realizada através das interfaces seriais emuladas pelo simulador. Além disso, é facultativo à cada grupo, utilizar as demais interfaces de E/S presentes na ferramenta.

### 4 Cronograma

| Semana | Data         | Descrição   |
|--------|--------------|---|
| 01     | seg - 11/set | —   |
|        | qui - 14/set | Apresentação – Problema 1                         |
| 02     | seg - 18/set | Sessão Tutorial                                   |
|        | qui - 21/set | Lab0 – Introdução à Plataforma de Desenvolvimento |
| 03     | seg - 25/set | Sessão Tutorial                                   |
|        | qui - 28/set | Lab1 – Layout Físico de Circuitos em FPGA (1)     |
| 04     | seg - 02/out | Sessão Tutorial                                   |
|        | qui - 05/out | Lab1 – Layout Físico de Circuitos em FPGA (2)     |
| 05     | seg - 09/out | <i>Entrega do Problema 1</i>                      |
|        | qui - 12/out | <b>Feriado – Semana Santa</b>                     |

### 5 Avaliação

Para aferir o envolvimento do grupo nas discussões e na apresentação, o tutor poderá fazer perguntas sobre o funcionamento de qualquer componente, a qualquer aluno, tanto nas sessões tutoriais quanto na apresentação.

#### Formato da Avaliação

A nota final será a composição de 3 (três) notas parciais:

**Desempenho Individual**    nota de participação individual nas sessões tutoriais, de acordo com o interesse e entendimento demonstrados pelo aluno, assim como sua assiduidade, pontualidade e contribuição nas discussões;  
**Valor: 5,0 pontos.**

|                             |  |
|-----------------------------|--|
| <b>Documentação Técnica</b> | nota atribuída à cada grupo referente ao relatório técnico, referente ao desenvolvimento do produto;<br><b>Valor: 2,0 pontos</b> |
| <b>Programas de Teste</b>   | nota atribuída à demonstração de funcionamento dos programas de teste;<br><b>Valor: 3,0 pontos.</b>                              |

## 6 Recursos

Visite a página da disciplina para ter acesso ao acervo de documentos e modelos de documentação indicados para uso no decorrer do semestre.

## Referências

JNiosEmu: Making it easy to learn assembly. *<http://stpe.github.io/jniosemu/>*. 2017.

Nios II Processor. *<https://www.altera.com/products/processors/support.html>*. 2017.