



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 usando o algoritmo Bubble Sort
 - (c) Cálculo do fatorial de um número inteiro (recursivo)
 - (d) Geração de números primos
 - (e) Cálculo da potência de um número inteiro

Os programas devem considerar que os parâmetros serão introduzidos através de uma das entradas seriais do simulador.

4 Cronograma

Turmas P01 e P02

Semana	Data	Descrição
01	seg - 20/mar	Aula de apresentação da disciplina
	qui - 23/mar	Lab – Introdução à Plataforma de Desenvolvimento
02	seg - 27/mar	Sessão Tutorial
	qui - 30/mar	Lab – Layout Físico de Circuitos em FPGA
03	seg - 03/abr	Sessão Tutorial
	qui - 06/abr	Lab – Síntese em Verilog e FSM (1)
04	seg - 10/abr	Sessão Tutorial
	qui - 13/abr	Feriado – Semana Santa
05	seg - 17/abr	Sessão Tutorial
	qui - 20/abr	Lab2 - Síntese em Verilog e FSM (2)
06	seg - 24/abr	<i>Entrega do Problema 1</i>

Turmas P03 e P04

Semana	Data	Descrição
01	seg - 20/mar	Aula de apresentação da disciplina
	qui - 23/mar	Sessão Tutorial
02	seg - 27/mar	Lab0 – Introdução à Plataforma de Desenvolvimento
	qui - 30/mar	Sessão Tutorial
03	seg - 03/abr	Lab1 – Layout Físico de Circuitos em FPGA
	qui - 06/abr	Sessão Tutorial
04	seg - 10/abr	Lab2 – Síntese em Verilog e FSM (1)
	qui - 13/abr	Feriado – Semana Santa
05	seg - 17/abr	Lab2 - Síntese em Verilog e FSM (2)
	qui - 20/abr	Sessão Tutorial
06	seg - 24/abr	<i>Entrega do Problema 1</i>

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.

Documentação Técnica nota atribuída à cada grupo referente ao documento de especificação da arquitetura de instruções do processador;

Valor: 2,0 pontos

Programas de Teste nota atribuída à demonstração de funcionamento dos programas de teste;

Valor: 3,0 pontos.

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.