



## Problema #3 – Coprocessador

### 1 Tema

Desenvolvimento de Coprocessador para o Processador NIOS

### 2 Objetivos de Aprendizagem

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

- Projetar circuitos digitais utilizando Linguagem de Descrição de Hardware (Verilog);
- Implementar testes de unidade e de integração para módulos Verilog;
- Compreender a comunicação entre o processador e um coprocessador;
- Identificar características de operação e restrições em projetos de sistemas digitais;
- Analisar a implementação de circuitos sob o ponto de vista de dispositivos FPGA;
- Compreender a integração de sistemas digitais em nível de microarquitetura.

### 3 Contexto

A era da Internet Industrial já começou e ela une máquinas inteligentes, análise computacional avançada e trabalho colaborativo entre pessoas conectadas para gerar profundas mudanças e trazer eficiência operacional para setores industriais diversos: manufatura, transporte, energia e saúde. Com isso, surge o conceito da Indústria 4.0 que propõe uma importante mudança de paradigma em relação à maneira como as fábricas operam nos dias de hoje.

Nessa visão de futuro, ocorre uma completa descentralização do controle dos processos produtivos e uma proliferação de dispositivos inteligentes interconectados, ao longo de toda a cadeia de produção e logística. Tornar a Indústria 4.0 uma realidade implicará a adoção gradual de um conjunto de tecnologias emergentes de TI e automação industrial, na formação de um sistema de produção físico-cibernético, com intensa digitalização de informações e comunicação direta entre sistemas, máquinas, produtos e pessoas; ou seja, a tão famosa Internet das Coisas. Esse processo promete gerar ambientes de manufatura altamente flexíveis e autoajustáveis à demanda crescente por produtos cada vez mais customizados. (adaptado de <http://www.sebrae.com.br/sites/PortalSebrae/artigos/saiba-o-que-e-a-industria-40-e-descubra-as-oportunidades-que-ela-gera,11e01bc9c86f8510VgnVCM1000004c00210aRCRD>)

A área de Visão Computacional se enquadra neste contexto. Devido a sua versatilidade um mesmo sistema pode ser utilizado na manufatura de produtos diversos. A visão computacional é um campo interdisciplinar que lida com a forma em que os computadores podem ser programados para obter um entendimento de alto nível a partir de imagens ou vídeos digitais. Ela envolve o desenvolvimento de uma base teórica e algorítmica para alcançar uma compreensão visual automática. (adaptado de [https://en.wikipedia.org/wiki/Computer\\_vision](https://en.wikipedia.org/wiki/Computer_vision))

Nesta base teórica, a detecção de bordas inclui uma variedade de métodos matemáticos que visam identificar pontos em uma imagem digital na qual o brilho da imagem muda acentuadamente ou, mais formalmente, tem descontinuidades. Os pontos nos quais o brilho da imagem muda drasticamente são tipicamente organizados em um conjunto de segmentos de linhas. A detecção de bordas é uma ferramenta fundamental no processamento de imagens, visão de máquina e visão computacional, particularmente nas áreas de detecção e extração de recursos. (adaptado de [https://en.wikipedia.org/wiki/Edge\\_detection](https://en.wikipedia.org/wiki/Edge_detection))

### 4 Problema

A Altera projetou especificamente o *core* do processador Nios II/e “econômico” para usar o menor número de recursos da FPGA. Ele é oferecido gratuitamente (para os processadores Nios II Classic e Nios II Gen2), sem necessidade de licença. O *core* Nios II/e tem desempenho superior, mas está na mesma classe de custo de uma

arquitetura 8051 típica, alcançando mais de 30 DMIPS em até 200 MHz e usando menos de 700 elementos lógicos (LEs). (retirado de <https://www.altera.com/products/processors/benefits/nios-ii-processor-cores.html>)

Uma das formas de superar a limitação de recursos é o uso de coprocessadores. Um coprocessador é um processador usado para suplementar as funções do processador primário (a CPU). As operações executadas pelo coprocessador podem ser aritmética de ponto flutuante, gráficos, processamento de sinais, processamento de strings, criptografia ou interface de E/S com dispositivos periféricos. Ao descarregar tarefas que requerem muito processamento, os coprocessadores podem acelerar o desempenho do sistema. Os coprocessadores permitem que uma linha de computadores seja personalizada, para que os clientes que não necessitam do desempenho extra não precisem pagar por isso. (adaptado de <https://en.wikipedia.org/wiki/Coprocessor>)

Sendo assim, a **Integrated IP LLC**. precisa de desenvolver um sistema digital para análise de imagens em uma linha de produção de peças. Sua equipe de projeto foi destacada para desenvolver um módulo de segmentação que recebe uma imagem e filtra suas bordas. Para testar o módulo será utilizado o processador desenvolvido em projetos anteriores.

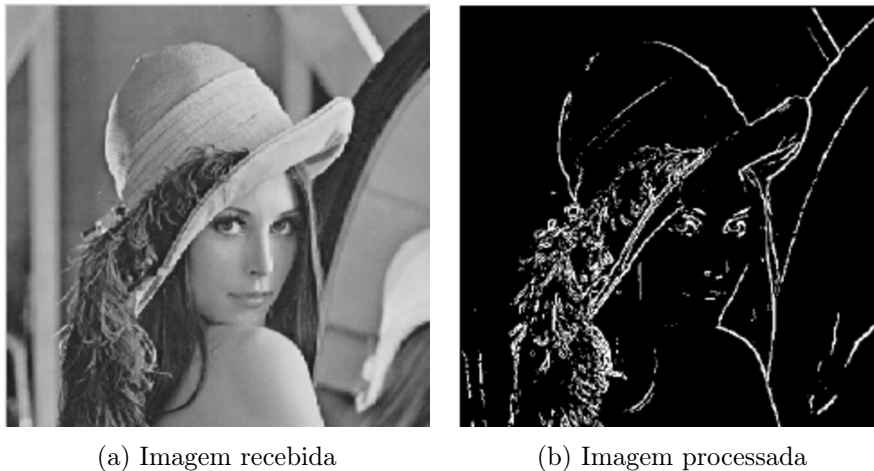


Figura 1: Descrição do problema.

## 5 Requisitos

O SoC desenvolvido precisa ser embutido em dispositivo FPGA e deve atender as seguintes restrições:

- A imagem será recebida em níveis de cinza com 8 bites de profundidade;
- A imagem será recebida por meio de uma interface serial RS-232;
- A segmentação será implementada como um coprocessador implementado em *hardware* utilizando Verilog;
- Utilizar o operador de Sobel para a detecção de bordas;
- A imagem filtrada deve ser apresentada em um monitor CRT.

## 6 Produto

No prazo indicado no cronograma a seguir, cada equipe deverá apresentar:

1. SoC composto pelo processador, coprocessador e o controlado de interface VGA;
2. Código em linguagem assembly para recepção e apresentação da imagem filtrada;
3. Uma apresentação sucinta contendo informações acerca das etapas de síntese lógica e física, incluindo, mas não limitando-se a: (i) indicação do caminho crítico do circuito; (ii) área total ocupada pelo circuito em função dos elementos internos do dispositivo FPGA (número de LEs por modo de operação, número de LABs, número de registradores, número de blocos de memória); (iii) testes de unidade (quando necessário) e integração automatizados; e (iv) descrição dos cenários de teste de funcionamento.

## 7 Cronograma

Semana	Data	Descrição
15	25/06/2018 (seg)	Apresentação do Problema #3
	28/06/2018 (qui)	Entrega do Problema #2
16	02/07/2018 (seg)	Feriado
	05/07/2018 (qui)	Lab4 – Simulação e teste (2)
17	09/07/2018 (seg)	Sessão Tutorial #1
	12/07/2018 (qui)	Lab5 – Gate Level Simulation (1)
18	16/07/2018 (seg)	Sessão Tutorial #2
	19/07/2018 (qui)	Lab5 – Gate Level Simulation (2)
19	23/07/2018 (seg)	Sessão Tutorial #3
	26/07/2018 (qui)	Feriado
20	30/07/2018 (seg)	Sessão Tutorial #4
	02/08/2018 (qui)	Sessão de Acompanhamento
21	06/08/2018 (seg)	Entrega do Problema #3
	09/08/2018 (qui)	— — —

## 8 Avaliação

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

### Formato da Avaliação

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

<b>Desempenho Individual</b>	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; <b>Peso: 3,0 pontos.</b>
<b>Apresentação</b>	nota atribuída à cada grupo, referente à apresentação técnica do produto; <b>Peso: 3,0 pontos</b>
<b>Protótipo</b>	nota atribuída à cada grupo, oriunda da análise da implementação em Verilog, incluindo sua descrição funcional e estruturas de teste/validação; <b>Peso: 4,0 pontos.</b>

## 9 Orientações

Vide Problema #1.

## 10 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

BITTENCOURT, J. C. N. *Projeto de IP-cores para Aplicações Gráficas com Prototipação em FPGA Utilizando o ipPROCESS*. Feira de Santana - BA: [s.n.], 2012. Trabalho de Conclusão de Curso. Disponível em: <<http://www.ecomp.uefs.br/a/ecomp.uefs.br/cccomp/trabalhos-de-conclusao-de-curso/TCC%20Jo%C3%A3o%20Carlos%20Bittencourt%202012.2.pdf>>.

- BROWN, S. D.; VRANESIC, Z. G. *Fundamentals of Digital Logic with Verilog Design*. 2nd. ed. [S.l.]: McGraw-Hill Higher Education, 2008.
- GONZALEZ, R. C.; WOODS, R. E. *Processamento Digital de Imagens*. 3. ed. São Paulo: Pearson / Prentice Hall, 2010.
- HARRIS, D. M.; HARRIS, S. L. *Digital Design and Computer Architecture*. 2. ed. USA: Elsevier, 2013. ISBN 9780123944245.
- KILTS, S. *Advanced FPGA Design: Architecture, Implementation, and Optimization*. [S.l.]: John Wiley & Sons, 2007.
- MATURANA, P. S. *Algoritmos de Detecção de Bordas Implementados em FPGA*. Dissertação (Mestrado) — Universidade Estadual Paulista, Ilha Solteira - SP, 2010. Disponível em: <[http://www.feis.unesp.br/Home/departamentos/engenhariaeletrica/pos-graduacao/273-dissertacao\\_patricia.pdf](http://www.feis.unesp.br/Home/departamentos/engenhariaeletrica/pos-graduacao/273-dissertacao_patricia.pdf)>.
- PATTERSON, D. A.; HENNESSY, J. L. *Arquitetura de Computadores*. 3. ed. Brasil: Editora Elsevier, 2014. 709 p. Impresso. ISBN 9788535235852.
- SOUZA, F. A. de. *Implementação em FPGA de Detecção de Bordas em Imagens Digitais*. Feira de Santana - BA: [s.n.], 2012. Trabalho de Conclusão de Curso. Disponível em: <[http://www.ecomp.uefs.br/trabalhos-de-conclusao-de-curso/2011.2\\_TCC\\_Fladmy\\_pos\\_Banca\\_Final.pdf](http://www.ecomp.uefs.br/trabalhos-de-conclusao-de-curso/2011.2_TCC_Fladmy_pos_Banca_Final.pdf)>.
- SPEAR, C. *System Verilog for Verification : A Guide to Learning the Testbench Language Features*. 2nd. ed. [S.l.]: Springer, 2008.