

UNIVERSIDADE FEDERAL DA BAHIA - UFBA
ESCOLA POLITÉCNICA / DEPARTAMENTO DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
ENGG57 – LABORATÓRIO INTEGRADO IV-A – 2023.1 – PROF. WAGNER L. A. OLIVEIRA

Este Laboratório Integrado será formado por um problema composto por 2 produtos, cada qual gerador de uma nota, sendo a nota final na disciplina a média aritmética simples das notas obtidas nos produtos requisitados.

A nota de cada produto (**N**) será formada por 3 componentes, com ponderação ajustável:

- Nota Produto (**NP**), com peso de 50%;
- Nota Relatório (**NR**), com peso de 10%;
- Nota Participação Individual (**NI**), com peso de 40%.

Para cada produto, o não cumprimento do prazo acarretará na penalização da perda de 10% da nota **N** por semana de atraso. A nota **NI** será aferida através da participação do/da estudante nas sessões tutoriais e no cumprimento de metas dos projetos.

Calendário Proposto

Data	Atividade
21/03	Apresentação da Disciplina
28/03	Revisão Quartus / ModelSim
04/04	Ferramentas de Depuração no Quartus
11/04	Platform Designer: Especificação de Sistemas Embarcados e Elaboração de Componentes
18/04	Platform Designer: Simulação
25/04	Apresentação do Problema
02/05	Sessão PBL / Implementação
09/05	Sessão PBL / Implementação
16/05	Sessão PBL / Implementação
23/05	Sessão PBL / Implementação
30/05	Sessão PBL / Implementação
06/06	Sessão PBL / Implementação
13/06	Sessão PBL / Implementação
20/06	Sessão PBL / Implementação
27/06	Sessão PBL / Implementação
04/07	Apresentação de Resultados

Problema: Acelerador em Hardware

Tema: *Modelagem, implementação e validação de um sistema hardware/software amparado no Platform Designer.*

Contexto:

Coprojeto hardware/software é um termo que se refere ao projeto concorrente dos componentes de hardware e de software de um sistema, de forma a atender a funcionalidade pretendida e os requisitos de projeto, geralmente em curto prazo de tempo. Há uma grande variedade de componentes envolvidos, os quais podem ser utilizados de diferentes formas, proporcionando várias alternativas de projeto, as quais deverão ser testadas e avaliadas. Há várias metodologias de coprojeto, contudo, basicamente, o que se procura fazer é o particionamento adequado das funções do sistema em componentes de hardware ou de software, com base em restrições como, por exemplo, custo, desempenho, consumo de energia, potência dissipada e grau de programabilidade. Em linhas gerais, funções do sistema que necessitam de alta velocidade e baixo consumo são mapeadas para hardware (o qual será desenvolvido através de HDLs como Verilog), ao passo que funções que apresentam maiores complexidade e necessidade de retrabalho são associadas para execução em processadores pré-existentes (uso de programação).

Dispositivos do tipo FPGA (*Field-Programmable Gate Array*) são chips de lógica reconfigurável após o encapsulamento, cujo uso tem apresentado significativa ascensão nos últimos anos. Atualmente, é possível sintetizar um processador RISC em um chip FPGA por apenas US\$ 2.00. Outros exemplos de aplicação desse tipo de chip podem ser encontrados em áreas diversas:

- telecomunicações: circuitos de roteamento de estações rádio base de telefonia celular / satélites;
- controles industriais: circuitos de controle integrados ou como lógica “glue”, no interfaceamento de microcontroladores;
- sistemas de processamento digital de sinais: sistemas de reconhecimento de padrões (computação gráfica / análise facial / análise de íris / reconhecimento de placas de automóveis), decodificadores MPEG4 e HEVC;
- mod chips: dispositivos que “modificam” a função original de um sistema (desbloqueios de videogames);
- arquiteturas reconfiguráveis em tempo de execução;
- dispositivos acionados por computador: implementação de interfaces com barramentos padrão de PCs (PCI, PCI-Express, SCSI, dentre outros), possibilitando, inclusive, a síntese da própria aplicação de hardware desejada (*back-end* do dispositivo); e
- HPC (*High-Performance Computing*): coprocessadores para aceleração de partes de algoritmos em substituição/colaboração com GPUs (exemplo: Bing da Microsoft).

Muitas aplicações podem se beneficiar de aceleração em hardware especializado. Em arquiteturas heterogêneas, a execução do software pode ser distribuída numa CPU (responsável pelo gerenciamento da comunicação entre nós e organização de dados de E/S) e em um ou mais aceleradores (processadores de propósito específico, responsáveis pela execução das partes de computação intensa do algoritmo da aplicação alvo). Em tais plataformas, a otimização de desempenho pode decorrer de duas formas distintas:

- O algoritmo acelerado executa mais rápido no hardware dedicado do que em seu equivalente numa implementação comum usando somente CPU; ou
- O algoritmo acelerado executa tão rápido quanto seu equivalente em CPU, mas pode ser executado em paralelo em várias unidades de processamento dedicadas.

FPGAs e GPUs são os dois principais tipos de dispositivos utilizados como aceleradores, com maior ou menor adequação conforme a carga de trabalho. FPGA é um hardware altamente configurável, de temporização determinística e interfaces de projeto diversificadas. Já GPUs fornecem unidades de execução paralela maciça e alta largura de banda de memória, através de uma estrutura regular, fortemente acoplada. De modo geral, FPGAs se destacam em aplicações que envolvam streaming de dados em ponto fixo, enquanto GPUs são mais adequadas para operações em ponto flutuante para grandes conjuntos de dados sem interdependência.

Esta disciplina visa a aquisição de conhecimentos básicos relativos à modelagem, implementação e validação de sistemas embarcados em FPGA. Tais sistemas podem ser construídos a partir de módulos IP (*Intellectual Property*) já existentes ou a serem desenvolvidos em HDL como, por exemplo, Verilog. Tais módulos facilitam o desenvolvimento destes sistemas, abreviando o fluxo de projeto dos mesmos.

Atualmente, muitos módulos IP têm sido utilizados como aceleradores em hardware. Módulos IP desenvolvidos em Verilog podem ser integrados ao ambiente **Platform Designer (Qsys)** e a sua validação de desempenho pode ser amparada no processador **Nios II**.

Os trabalhos desta disciplina exigirão duas formas de implementação:

- Exclusivamente em software, usando linguagem C e o Nios II, para o propósito de identificação de funções críticas do sistema; e
- Mista (coprojetado hardware/software), usando linguagem C, Nios II e os módulos implementados para as funções críticas do sistema identificadas.

Neste contexto, o trabalho atual solicita à sua equipe o desenvolvimento de uma aplicação embarcada cujos componentes (funções) poderão ser implementados tanto em hardware (através de Verilog) quanto em software (através do Nios II). Caberá à sua equipe tomar as decisões de tal particionamento, a partir da identificação da parte crítica do algoritmo a ser tratado, conforme especificação na sequência.

Problema:

RANSAC (*Random Sample Consensus*) é um algoritmo bastante utilizado em aplicações de visão computacional, tais como junção de imagens, estimação de movimento, detecção e rastreamento de objetos, dentre outras. Independentemente de suas variantes/aperfeiçoamentos, basicamente é um algoritmo utilizado para a detecção de superfícies matematicamente parametrizáveis (planos, cones, cilindros, etc.) a partir da estimação dos parâmetros do modelo matemático que descreve tais superfícies.

Supõe-se que o conjunto de dados a serem processados contém *inliers* e *outliers*. Os *inliers* são dados que podem ser explicados por um modelo matemático, enquanto os *outliers* são dados que não se ajustam bem ao modelo, como ruído.

O algoritmo RANSAC permite tratar dados afetados por uma grande quantidade de *outliers*. Por exemplo, considere o problema de ajustar uma reta a um conjunto de pontos no plano, com a presença de inúmeros *outliers* (ou seja, pontos fora da reta). Para iniciar, seleciona-se aleatoriamente dois pontos bem afastados para calcular os parâmetros da reta. Se estes pontos não envolverem erros grosseiros, então uma grande quantidade de pontos remanescentes deverá concordar com esta realização da equação da reta; caso contrário, o inverso ocorre e um novo par de pontos deverá ser escolhido e testado. Quando ocorrer a escolha correta, todos os pontos consistentes devem ser utilizados para a estimativa dos parâmetros da reta usando procedimentos de estimação convencionais, como o baseado no Método dos Mínimos Quadrados (MMQ). Procedimentos de análise estatística podem ser utilizados após a aplicação do MMQ para eventualmente eliminar dados afetados por erros grosseiros e novos pontos, rejeitados anteriormente, serem incorporados aos pontos aceitos. O critério de parada normalmente se baseia no índice de pontos que se espera validar. Por exemplo, se a expectativa é que pelo menos 50% dos pontos estão corretos, o critério de parada poderá basear-se na validação desses 50% de pontos.

Neste trabalho, vocês deverão adotar o algoritmo RANSAC para a resolução de um dado problema na área de computação visual.

Inicialmente, vocês farão uma implementação exclusivamente em software, considerando o processador **NIOS II**, no ambiente **Platform Designer** do **Quartus**, voltado para o FPGA disponível no kit **DE2-115** da Intel. Através de simulações, vocês traçarão o perfil de execução da aplicação (o chamado *profile*), o qual permitirá reconhecer qual o procedimento de maior gasto de tempo do algoritmo RANSAC.

Em seguida, vocês implementarão um módulo em Verilog para este procedimento crítico, encapsularão tal módulo no ambiente **Platform Designer** e o utilizarão como um acelerador em hardware, numa implementação hardware/software, em conjunto com a aplicação já desenvolvida para propósitos de *profile*. Por meio de exemplos, vocês deverão provar o ganho de desempenho obtido.

Produtos:

Vocês deverão entregar os seguintes artefatos:

- P1.** A pasta de projeto **LAB_IV_A\Quartus\RANSAC_NIOS** compactada, contendo a implementação desenvolvida exclusivamente em software, juntamente com um arquivo .pdf contendo o *profile* da aplicação;
- P2.** A pasta de projeto **LAB_IV_A\Quartus\RANSAC_Acelerado** compactada, contendo a implementação mista hardware/software solicitada;
- Rel.** Um relatório em .pdf, contendo a descrição das implementações desenvolvidas e a participação individual e aprendizado adquirido por cada membro.