UNIVERSIDADE ESTADUAL DE FEIRA DE SANTANA
DEPARTAMENTO DE TECNOLOGIA
ÁREA DE ELETRÔNICA

Problema #2 – 2024.2

1. Tema

Assembly e bibliotecas de software

2. Objetivos de Aprendizagem

Ao final da realização deste problema, o/a discente deverá ser capaz de:

- Aplicar conhecimentos de interação hardware-software;
- Compreender o mapeamento de memória em uma arquitetura ARM;
- Utilizar a interface de conexão entre HPS e FPGA da DE1-SoC;
- Programar em assembly para um processador com arquitetura ARM;
- Entender políticas de gerenciamento de sistema operacional Linux em arquitetura ARM;
- Compreender os princípios básicos da arquitetura da plataforma DE1-SoC.

3. Problema

O discente Gabriel Sá Barreto Alves em suas atividades de iniciação científica e em seu trabalho de conclusão de curso, projetou e implementou um Processador Gráfico que permite mover e controlar elementos em um monitor VGA com resolução de 640×480 *pixels*. Este processador permite desenhar dois tipos de polígonos convexos (Quadrado e Triângulo) e uma quantidade determinada de *sprites*. Durante o desenvolvimento, Gabriel utilizou como unidade de processamento principal o NIOS II. Ele foi embarcado na mesma FPGA que o Processador Gráfico. A figura 1 ilustra a arquitetura desenvolvida.

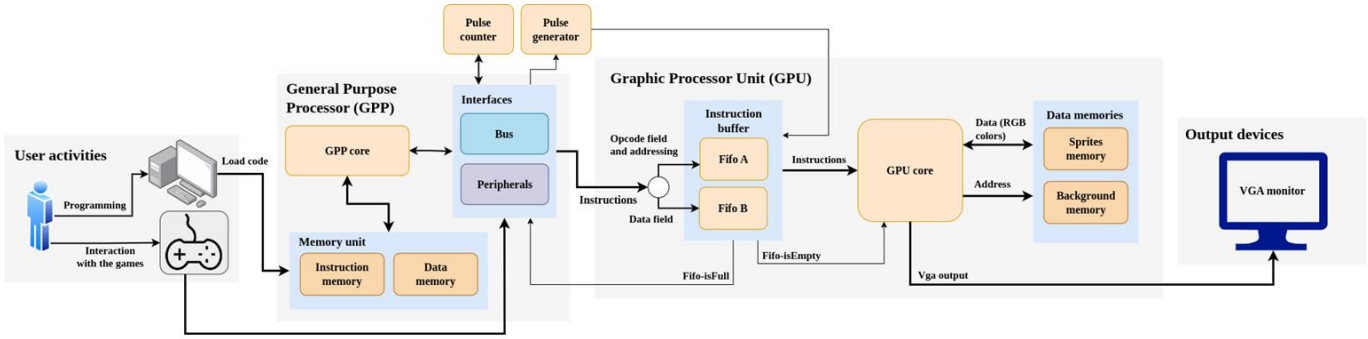

Figura 1: Arquitetura desenvolvida.

Sua equipe está incubida de desenvolver uma biblioteca com funções gráficas para o Processador Gráfico, que possam facilitar a implementação do jogo Tetris desenvolvido no problema anterior. O trabalho de Conclusão de Curso de Gabriel pode ser acessado em: https://drive.google.com/file/d/1MIIlpB9TSnoPGEMkocr36EH9-CFz8psO/view.

Observação: Compreender como usar o Processador Gráfico é mandatório para este desenvolvimento.

4. Requisitos

O problema a ser desenvolvido no Kit de desenvolvimento DE1-SoC deve atender às seguintes restrições:

4.1. O código da biblioteca deve ser escrito em linguagem aseembly;

4.2. A biblioteca deve conter as funções essenciais para que seja possível implementar a parte gráfica do jogo usando o Processador Gráfico;

4.3. A biblioteca deve seguir as recomendações descritas em: https://github.com/MaJerle/c-code-style

5. Produto

Todo o projeto deverá ser disponibilizado na plataforma GitHub. No prazo indicado no cronograma a seguir, cada equipe deverá apresentar:

5.1. Levantamento de requisitos;

5.2. Código

5.2.1. Códigos em linguagem C e assembly;

5.2.2. Todos os códigos deverão estar detalhadamente comentados;

5.3. Script de compilação tipo Makefile para geração do código executável;

5.4. Documentação técnica escrita no arquivo READ.ME do projeto no GitHub, contendo, no mínimo:

5.4.1. Detalhamento dos software usados no trabalho, incluindo softwares básicos;

5.4.2. Detalhamento da biblioteca criada incluindo a explicação dos métodos e como a mesma deve ser compilada e utilizada;

5.4.3. Descrição de instalação, configuração de ambiente e execução;

5.5. Descrição dos testes de funcionamento do sistema, bem como, análise dos resultados alcançados.

6. Cronograma

Semana	Data	Descrição
01	25/09	Apresentação do Problema II – Tutorial
	02/10	LAB
02	04/10	Tutorial
	09/10	LAB
03	11/10	Tutorial
	16/10	LAB

04	18/10	Tutorial
	23/10	LAB
05	25/10	Tutorial
	30/10	LAB
06	01/11	Apresentação do Problema II

7. Avaliação

Para avaliar o envolvimento do grupo nas discussões e na apresentação, o tutor poderá fazer perguntas variadas a qualquer membro, tanto nas sessões tutoriais quanto na apresentação. O estudante que não comparecer, ou se atrasar, no dia da sessão de apresentação, terá automaticamente nota 0,0 (zero) no problema, excetuando-se as condições que permitem 2ª chamada de avaliações, conforme regulamento do curso.

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

Critério	Critérios para a nota	Peso
Desempenho Individual	Participação individual nas sessões tutoriais, de acordo com o interesse e entendimento demonstrados pelo aluno, assim como sua assiduidade, pontualidade e contribuição nas discussões. Essa nota inclui o desempenho do estudante na apresentação do problema no laboratório.	3
Documentação	Documentação técnica de cada grupo, considerando qualidade da redação (ortografia e gramática), organização dos tópicos, definição do problema, descrição da solução, explicação dos experimentos, análise dos resultados, detalhando os itens não atendidos, se for o caso.	3
Códigos	Qualidade do código fonte (organização e comentários), e execução correta dos códigos binários de acordo com testes de validação que explorem as situações de uso.	4