



Problema #2 – 2024.1

Barramento

1. Tema

Conversas entre Hardware e 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 C 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.

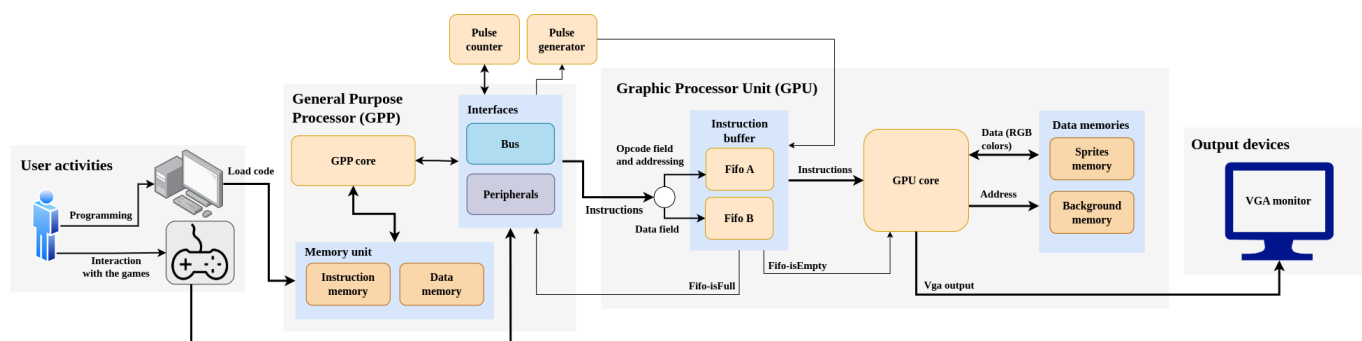


Figura 1: Arquitetura desenvolvida.

A plataforma DE1-SoC possui um processador ARM (HPS) que dispensa o uso do NIOS II. Essa “facilidade” resulta em um problema: Como o HPS “conversará” com o Processador Gráfico que está na FPGA?

Sua equipe está incumbida de desenvolver um módulo de Kernel no HPS para o Processador Gráfico. Visando facilitar a vida do programador, também será necessário desenvolver uma biblioteca em C. O trabalho de Conclusão de Curso de Gabriel pode ser acessado em: <https://drive.google.com/file/d/1MIIIpB9TSnoPGEMkocr36EH9-CFz8psO/view>

Observação: Compreender 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 deve ser escrito em linguagem C;
- 4.2. A biblioteca deve conter no mínimo uma função para cada Instrução do 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. Código em linguagem C ilustrando todos os entes disponíveis no Processador Gráfico. A figura 2 contém um exemplo de como deve ser a imagem resultante deste código;

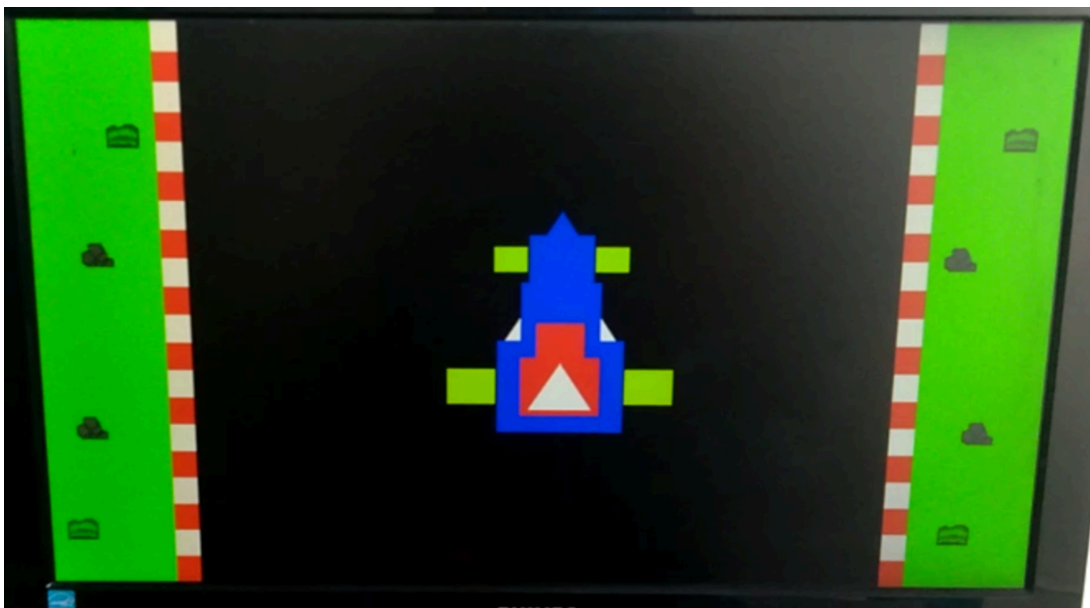


Figura 2: Exemplo de uso do Processador Gráfico.

- 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 README do projeto no GitHub, contendo, no mínimo:
 - 5.4.1. Detalhamento dos software usados no trabalho, incluindo softwares básicos;
 - 5.4.2. Arquitetura do Processador Gráfico;
 - 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
07	qua. - 01/mai.	Feriado - Dia do Trabalho
	sex. - 03/mai.	Apresentação do Problema 2
08	qua. - 08/mai.	Problema 2 – Tutorial/Desenvolvimento
	sex. - 10/mai.	Problema 2 – Tutorial/Desenvolvimento
09	qua. - 15/mai.	Problema 2 – Tutorial/Desenvolvimento
	sex. - 17/mai.	Problema 2 – Tutorial/Desenvolvimento
10	qua. - 22/mai.	Problema 2 – Tutorial/Desenvolvimento
	sex. - 24/mai.	Problema 2 – Tutorial/Desenvolvimento
11	qua. - 29/mai.	Problema 2 – Tutorial/Desenvolvimento
	sex. - 31/mai.	Provável ponto facultativo - Corpus Christi (30/maio)
12	qua. - 05/jun.	Problema 2 – Tutorial/Desenvolvimento
	sex. - 07/jun.	Problema 2 – Entrega/Avaliação

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