



**Documento de Arquitetura**

Problema 1 – A Linguagem do Computador

Universidade Estadual de Feira de Santana

**Build 1.0a**

# Histórico de Revisões

| Date       | Descrição                                               | Autor(s)                  |
|------------|---------------------------------------------------------|---------------------------|
| 15/03/2016 | Transcrição da documentação ".DOC" para o formato LaTeX | Lucas Anderson S. de Lima |

## SUMÁRIO

|          |                                         |          |
|----------|-----------------------------------------|----------|
| <b>1</b> | <b>Introdução</b>                       | <b>3</b> |
| 1        | Propósito do Documento . . . . .        | 3        |
| 2        | Stakeholders . . . . .                  | 3        |
| 3        | Visão Geral do Documento . . . . .      | 3        |
| 4        | Definições . . . . .                    | 3        |
| 5        | Acrônimos e Abreviações . . . . .       | 4        |
| <b>2</b> | <b>Visão Geral da Arquitetura</b>       | <b>5</b> |
| 1        | Diagrama de Classe . . . . .            | 5        |
| 2        | Definições de Entrada e Saída . . . . . | 5        |
| 3        | Datapath Interno . . . . .              | 6        |
| 3.1      | Memória . . . . .                       | 6        |
| 3.2      | Clock/Frequência . . . . .              | 6        |
| 3.3      | Entrada . . . . .                       | 6        |
| 3.4      | Processador . . . . .                   | 6        |
| 3.5      | Saída . . . . .                         | 6        |
| <b>3</b> | <b>Descrição da Arquitetura</b>         | <b>7</b> |
| 1        | Nome do Módulo . . . . .                | 7        |
| 1.1      | Diagrama de Classe . . . . .            | 7        |
| 1.2      | Definições de Entrada e Saída . . . . . | 7        |
| 1.3      | Datapath Interno . . . . .              | 7        |
| 1.4      | Máquina de Estados . . . . .            | 7        |
| 1.5      | Temporização . . . . .                  | 7        |

# 1 | Introdução

## 1. Propósito do Documento

Este documento descreve a arquitetura do projeto Problema 1 – A Linguagem do Computador, incluindo especificações do circuitos internos e máquinas de estados de cada componente. Ele também apresenta diagramas de classe, definições de entrada e saída e diagramas de temporização. O principal objetivo deste documento é definir as especificações do projeto Problema 1 – A Linguagem do Computadore prover uma visão geral completa do mesmo e suas funcionalidades.

## 2. Stakeholders

| Nome                                                | Papel/Responsabilidades |
|-----------------------------------------------------|-------------------------|
| Antonio Crispim; Lucas Anderson e Anderson Teixeira | Testes Assembly         |
| Antonio Crispim                                     | Montador                |
| Lucas Anderson                                      | Simulador               |
| Lucas Anderson                                      | Documentação Tecnica    |

## 3. Visão Geral do Documento

O presente documento é apresentado como segue:

- **Capítulo 2** – Este capítulo apresenta uma visão geral da arquitetura, com foco em entrada e saída do sistema e arquitetura geral do mesmo;  
[EDITAR]

## 4. Definições

| Termo    | Descrição |
|----------|-----------|
| [EDITAR] | [EDITAR]  |

## 5. Acrônimos e Abreviações

| Sigla | Descrição                                               |
|-------|---------------------------------------------------------|
| CI    | Circuitos Integrados                                    |
| MI    | Módulo Integrador                                       |
| SD    | Sistemas Digitais                                       |
| UEFS  | Universidade Estadual de Feira de Santana               |
| GT    | Grupo de Trabalho                                       |
| USB   | Universal Serial Bus                                    |
| PBL   | Aprendizagem baseada em Problema                        |
| NFR   | Requisitos não Funcionais                               |
| FPGA  | Field-Programmable Gate Array                           |
| FR    | Requisitos Funcionais                                   |
| E/S   | Entrada e Saída de informações                          |
| MIPS  | Microprocessador sem estágios intertravados de pipeline |
| ALU   | Unidade lógica e aritmética                             |
| NOP   | Nenhuma Operação                                        |

## 2 | Visão Geral da Arquitetura

### 1. Diagrama de Classe

Foi utilizado para o desenvolvimento da arquitetura desse projeto o padrão MIPS, a arquitetura MIPS foi elaborada para privilegiar a simplicidade do conjunto de instruções sem, contudo, sacrificar sua flexibilidade. Devido à limitação de todas as instruções possuírem exatamente o mesmo tamanho, instruções em geral disponíveis em outros processadores estão ausentes no MIPS. Contudo, foi tomado cuidado no projeto desta arquitetura para que tal funcionalidade possa ser suprida de forma simples via o conceito de pseudo-instruções. Pseudo-instruções são instruções inexistentes em uma arquitetura, mas disponibilizadas ao programador em linguagem de montagem. Sua implementação mediante uso de uma sequência de instruções existentes é feita no código objeto pelo programa montador. Pseudo-instruções costumam ser substituídas pelo montador ao gerar instruções para o computador na forma de Linguagem de Máquina, essas informações são armazenadas no processador, tendo base que a memória de instruções armazena apenas instruções. Cada instrução na MIPS ocupa exatamente 32 bits. Assim, uma instrução ocupa 4 posições de memória, sendo assim as mudanças de instruções acompanham mudanças no percorrer da memória até sua saída. Com base nas informações de saída é possível realizar uma comparação de resultados utilizando um software, o simulador "MARS"; o simulador MARS pode gerar arquivos de saída contendo o código objeto das áreas de dados e de instruções de qualquer programa, entre outras informações. Tais arquivos podem ser salvos usando a opção de menu File Dump Memory do simulador. Ele será útil para gerar a entrada do processo de simulação da implementação de hardware a ser realizada neste trabalho. O código objeto pode ser gerado após a leitura e montagem de um arquivo contendo um programa em linguagem de montagem correto, podendo ser utilizado como comparação para o montador desenvolvido.

A imagem a seguir representa os passos da entrada até a saída do processador.  
[COLOCAR IMAGEM]

### 2. Definições de Entrada e Saída

A arquitetura MIPS (utilizada neste projeto) é do tipo load-store, ou seja, as operações lógicas e aritméticas são executadas exclusivamente entre registradores da arquitetura ou entre constantes imediatas e registradores. As operações de acesso à memória só executam ou uma leitura da memória (load) ou uma escrita na memória (store). Sendo as entradas via USB: instruções e pseudo-instruções em Assembly capazes de realizar várias funções tendo como base principalmente a ALU e tendo como saída os resultados das execuções e seus valores.

### **3. Datapath Interno**

#### **3.1. Memória**

Este requisito define uma memória mínima de 64 KB = 65536 Bytes (in binary) para o funcionamento de todas as funções do processador.

#### **3.2. Clock/Frequência**

Não sendo necessária devido a utilização de memória assíncrona.

#### **3.3. Entrada**

Saída do Montador; arquivo de texto contendo as informações em binário.

#### **3.4. Processador**

Hardware ou Simulador capaz de executar e armazenar no mínimo 64 instruções compatíveis com seu modelo (no caso, modelo deste projeto).

#### **3.5. Saída**

Arquivo de texto contendo a execução, valores e resultados das instruções solicitadas e processadas pelo processador/simulador.

## 3 | Descrição da Arquitetura

### 1. Nome do Módulo

#### 1.1. Diagrama de Classe

#### 1.2. Definições de Entrada e Saída

#### 1.3. Datapath Interno

#### 1.4. Máquina de Estados

#### 1.5. Temporização

A partir das discursões anteriores percebe-se que ambas as interfaces de memória são assíncronas, ou seja, não dependem de sinais de relógio tais como o sinal clock.