



**Documento de Arquitetura**

MUSA

Fazemos Qualquer Negócio Inc.

**Compilação 2.0**

# Histórico de Revisões

| Date       | Descrição  | Autor(s)              |
|------------|--|-----------------------|
| 25/06/2014 | Concepção do documento   | joaocarlos            |
| 15/10/2014 | Adição da subseção de acesso à memória   | Weverson Gomes        |
| 16/10/2014 | Adição da seção "Leitura da Instrução" com dados preliminares e modificação do nome do projeto no documento. | santana22 e gabri4el. |
| 19/10/2014 | Modificações na seção "Leitura da Instrução"   | santana22             |
| 20/10/2014 | Correções na subseção de acesso à memória  | Weverson Gomes        |

## SUMÁRIO

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introdução</b>                       | <b>4</b>  |
| 1        | Propósito do Documento . . . . .        | 4         |
| 2        | Stakeholders . . . . .                  | 4         |
| 3        | Visão Geral do Documento . . . . .      | 4         |
| 4        | Acrônimos e Abreviações . . . . .       | 5         |
| <b>2</b> | <b>Visão Geral da Arquitetura</b>       | <b>6</b>  |
| 1        | Restrições . . . . .                    | 6         |
| 2        | Codificação das instruções . . . . .    | 6         |
| 3        | Descrição dos Componentes . . . . .     | 10        |
| <b>3</b> | <b>Descrição da Arquitetura</b>         | <b>11</b> |
| 1        | Leitura da Instrução . . . . .          | 11        |
| 1.1      | Diagrama de Classe . . . . .            | 11        |
| 1.2      | Definições de entrada e saída . . . . . | 11        |
| 1.3      | Datapath Interno . . . . .              | 12        |
| 2        | Decodificação da Instrução . . . . .    | 12        |
| 2.1      | Diagrama de Classe . . . . .            | 12        |
| 2.2      | Definições de entrada e saída . . . . . | 12        |
| 2.3      | Datapath Interno . . . . .              | 12        |
| 3        | Calculos . . . . .                      | 12        |
| 3.1      | Diagrama de Classe . . . . .            | 12        |
| 3.2      | Definições de entrada e saída . . . . . | 12        |
| 3.3      | Datapath Interno . . . . .              | 13        |
| 4        | Acesso à memória . . . . .              | 13        |

|     |   |    |
|-----|---|----|
| 4.1 | Diagrama de Classe . . . . .            | 13 |
| 4.2 | Definições de entrada e saída . . . . . | 13 |

# 1 | Introdução

## 1. Propósito do Documento

Este documento descreve a arquitetura do projeto MUSA, incluindo especificações do circuitos internos de cada componente. Ele também apresenta diagramas de classe, definições de entrada e saída. O principal objetivo deste documento é definir as especificações do projeto MUSA e prover uma visão geral completa do mesmo.

## 2. Stakeholders

| Nome  | Papel/Responsabilidades |
|---|-------------------------|
| Manuelle Macedo   | Gerência                |
| Patrick   | Análise                 |
| Dilan Nery, Lucas Almeida, Mirela Rios, Cabele e Vinícius Santana | Desenvolvimento         |
| Antônio Gabriel e Weverson Gomes                                  | Testes                  |
| Tarles Walker e Anderson Queiroz                                  | Implementação           |

## 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;
- **Capítulo 3** – Este capítulo descreve a arquitetura interna do IP a partir do detalhamento dos seus componentes, definição de portas de entrada e saída e especificação de caminho de dados.

#### 4. Acrônimos e Abreviações

| Sigla  | Descrição       |
|--------|-----------------|
| PC     | Program Counter |
| OPCODE | Operation Code  |

## 2 | Visão Geral da Arquitetura

### 1. Restrições

- Restrições –

### 2. Codificação das instruções

Instrução é uma palavra da linguagem de máquina, sua codificação é de fundamental importância para o processamento das operações. Todas as instruções contêm 32 bits. Existem 4 formatos de instruções: R (R-type), I (I-type), Load/Store e Jump. Os OPCO-DES são os códigos de operação da instrução, neste documento ele é representado em números hexadecimais.

| Formato da instrução | Instrução | Descrição                                       |
|----------------------|-----------|---|
| R-type               | ADD       | Soma dois valores                               |
|                      | SUB       | Subtrai dois valores                            |
|                      | MUL       | Multiplica dois valores                         |
|                      | DIV       | Divide dois valores                             |
|                      | AND       | AND lógico                                      |
|                      | OR        | OR lógico                                       |
|                      | CMP       | Compara dois valores                            |
|                      | NOT       | NOT lógico                                      |
| I-type               | ADDI      | Soma dois valores, um destes imediato.          |
|                      | SUBI      | Subtrai dois valores, um destes imediato.       |
|                      | ANDI      | AND lógico de dois valores, um destes imediato. |
|                      | ORI       | OR lógico de dois valores, um destes imediato.  |
| Load/Store           | LW        | Leitura de um dado da memória de dados          |
|                      | SW        | Armazena um dado na memória de dados            |
| Jump                 | JP        | Desvia para um destino                          |
|                      | JPC       | Desvia para um destino relativo ao PC           |
|                      | BRFL      | Desvia para um destino se RF==CST               |
|                      | CALL      | Chamada de subrotina                            |
|                      | RET       | Retorno de Subrotina                            |
|                      | HALT      | Parada do sistema                               |
|                      | NOPE      | Refresh no módulo                               |

**Tabela 2.2: Tabela de descrição das instruções**

O formato R está relacionado as instruções lógicas e aritméticas.

| OPCODE | RS    | RT    | RD    | SHAMT | FUNCT |
|--------|-------|-------|-------|-------|-------|
| 31:26  | 25:21 | 20:16 | 15:11 | 10:6  | 5:0   |

**Figura 2.1: Formato R**



Seus respectivos campos são:

- **OPCODE** - Código da operação básica da instrução.
- **RS** - Registrador do primeiro operando de origem.
- **RT** - Registrador do segundo operando de origem.
- **RD** - Registrador destino.
- **SHAMT** - *Shift amount*; Quantidade de deslocamento.
- **FUNCT** - Função; Esse campo seleciona a variante específica da operação no campo opcode, e as vezes, é chamado de código de função.

| OPCODE | INSTRUCTION | FUNCTION |
|--------|-------------|----------|
| 01     | ADD         | 00       |
| 01     | SUB         | 01       |
| 01     | MUL         | 02       |
| 01     | DIV         | 03       |
| 02     | AND         | 04       |
| 02     | OR          | 05       |
| 02     | CMP         | 06       |
| 02     | NOT         | 07       |

**Tabela 2.3: Definição dos OPCODES do formato R**

Um segundo tipo de formato de instrução é chamado de formato I, utilizado pelas instruções imediatas e de transferência de dados.

| OPCODE | RS    | RT    | ADDRESS OR IMMEDIATE |
|--------|-------|-------|----------------------|
| 31:26  | 25:21 | 20:16 | 15:0                 |

**Figura 2.2: Formato I**

Seus respectivos campos são:

- **OPCODE** - Código da operação básica da instrução.
- **RS** - Registrador do operando de origem.
- **RT** - Registrador destino.
- **ADDRESS OR IMMEDIATE** - Endereço de memória ou constante numérica.

| OPCODE | INSTRUCTION |
|--------|-------------|
| 03     | ADDI        |
| 04     | SUBI        |
| 05     | ANDI        |
| 06     | ORI         |

**Tabela 2.4: Definição dos OPCODES do formato I**

O terceiro formato de instrução são para instruções de leitura e escrita de dados na memória de dados.

| OPCODE<br>31:26 | RS<br>25:21 | ADDRESS<br>20:0 |
|-----------------|-------------|-----------------|
|-----------------|-------------|-----------------|

**Figura 2.3: Formato Load/Store**

Seus respectivos campos são:

- **OPCODE** - Código da operação básica da instrução.
- **RS** - Registrador onde está o dado.
- **ADDRESS** - Endereço da memória de dados.

| OPCODE | INSTRUCTION |
|--------|-------------|
| 07     | LW          |
| 08     | SW          |

**Tabela 2.5: Definição dos OPCODES do formato Load/Store**

O formato Jump servem para as instruções de desvio incondicional.

| OPCODE<br>31:26 | ADDRESS<br>25:0 |
|-----------------|-----------------|
|-----------------|-----------------|

**Figura 2.4: Formato Jump**

Seus respectivos campos são:

- **OPCODE** - Código da operação básica da instrução.
- **ADDRESS** - Endereço de memória ou constante numérica.

| OPCODE | INSTRUCTION |
|--------|-------------|
| 09     | JP          |
| 0A     | JPC         |
| 0B     | BRFL        |
| 0C     | CALL        |
| 0D     | RET         |
| 0E     | HALT        |
| 00     | NOPE        |

**Tabela 2.6: Definição dos OPCODES do formato Jump**

### 3. Descrição dos Componentes

A unidade de processamento a ser desenvolvida é composta a partir dos seguintes componentes:

- **Serial Controller** – Controlador para comunicação com módulo de transmissão serial através do protocolo RS232.
- **Interface Control** – Interface de controle, responsável por fazer a leitura correta das informações da serial e transmiti-las para a unidade de processamento.
- **Processing Unit** – Unidade responsável pela realização das operações e armazenamento do resultado.

## 3 | Descrição da Arquitetura

### 1. Leitura da Instrução

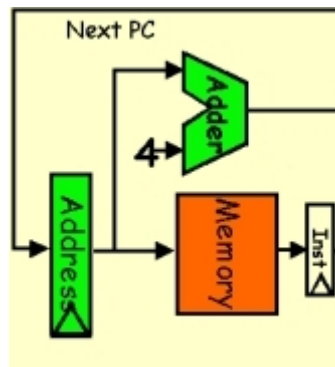
#### 1.1. Diagrama de Classe

| Instruction Fetch  |
|--|
| + clock : input bit<br>+ pcInput : input bit[32]<br>+ pcWrite : input bit<br>+ pcOutput : output bit[32]<br>+ instruction : output bit[32] |
|  |

#### 1.2. Definições de entrada e saída

| Nome        | Tamanho | Direção | Descrição  |
|-------------|---------|---------|--|
| pcInput     | 32      | entrada | Valor do PC atual.   |
| pcWrite     | 1       | entrada | Sinal proveniente da UC que habilita a modificação do valor de PC. |
| pcOutput    | 32      | saída   | Valor do PC atual.   |
| instruction | 32      | saída   | Instrução a ser executada.   |

#### 1.3. Datapath Interno



**Datapath preliminar**

## 2. Decodificação da Instrução

### 2.1. Diagrama de Classe

| Instruction Decode   |
|--|
| <ul style="list-style-type: none"> <li>+ clock : input bit</li> <li>+ pcInput : input bit[32]</li> <li>+ pcWrite : input bit</li> <li>+ pcOutput : output bit[32]</li> <li>+ instruction : output bit[32]</li> </ul> |

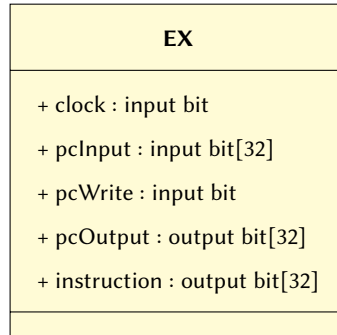
### 2.2. Definições de entrada e saída

| Nome        | Tamanho | Direção | Descrição  |
|-------------|---------|---------|--|
| pcInput     | 32      | entrada | Valor do PC atual.   |
| pcWrite     | 1       | entrada | Sinal proveniente da UC que habilita a modificação do valor de PC. |
| pcOutput    | 32      | saída   | Valor do PC atual.   |
| instruction | 32      | saída   | Instrução a ser executada.   |

### 2.3. Datapath Interno

### 3. Calculos

#### 3.1. Diagrama de Classe



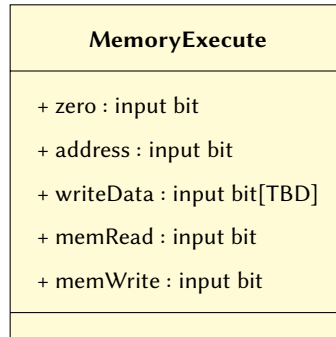
#### 3.2. Definições de entrada e saída

| Nome        | Tamanho | Direção | Descrição  |
|-------------|---------|---------|--|
| pcInput     | 32      | entrada | Valor do PC atual.   |
| pcWrite     | 1       | entrada | Sinal proveniente da UC que habilita a modificação do valor de PC. |
| pcOutput    | 32      | saída   | Valor do PC atual.   |
| instruction | 32      | saída   | Instrução a ser executada.   |

#### 3.3. Datapath Interno

## 4. Acesso à memória

### 4.1. Diagrama de Classe



### 4.2. Definições de entrada e saída

| Nome            | Tamanho | Direção | Descrição                                      |
|-----------------|---------|---------|--|
| zero            | 1       | entrada | Executa branch quando é zero.                  |
| address         | TBD     | entrada | Endereço no qual o dado deve ser escrito.      |
| memRead         | 1       | entrada | Sinal proveniente da UC que habilita leitura.  |
| memWrite        | 1       | entrada | Sinal proveniente da UC que habilita escrita.  |
| writeData       | 1       | entrada | O dado a ser escrito na memória.               |
| readData        | TBD     | saída   | Dado a ser utilizado pelo MUX do "Write Back". |
| writeToRegister | TBD     | saída   | Dado do segundo operando.                      |