



## **Especificação de Requisitos**

Core-MUSA

Universidade Estadual de Feira de Santana

**Build 0.1a**

## **Histórico de Revisões**

## SUMÁRIO

<b>1</b>	<b>Introdução</b>	<b>3</b>
1.1	Visão Geral do Documento . . . . .	3
1.2	Definições . . . . .	3
1.3	Acrônimos e Abreviações . . . . .	4
1.4	Prioridades dos Requisitos . . . . .	4
<b>2</b>	<b>Requisitos Funcionais</b>	<b>4</b>
2.1	Instruções de Movimentação de Dados . . . . .	4
2.2	Computacionais . . . . .	5
2.3	Instruções de Desvio . . . . .	10
2.4	Outras Instruções . . . . .	11
2.5	Flags . . . . .	11
2.6	Outros Requisitos Funcionais . . . . .	12
2.7	Modos de Endereçamento . . . . .	13
<b>3</b>	<b>Requisitos não Funcionais</b>	<b>13</b>
<b>4</b>	<b>Dependências</b>	<b>14</b>

## 1. Introdução

Este documento apresenta de forma detalhada os requisitos do sistema, sendo estes divididos em funcionais e não funcionais. Também apresenta as dependências para o funcionamento do sistema.

### 1.1. Visão Geral do Documento

- **Requisitos funcionais** - Lista de todos os requisitos funcionais.
- **Requisitos não funcionais** - Lista de todos os requisitos não funcionais.
- **Dependências** - Conjunto de dependências de *IP-cores* previstos.
- **Notas** - Lista de notas apresentadas ao longo do documento.
- **Referências** - Lista de todos os textos referenciados nesse documento.

### 1.2. Definições

<b>Data</b>	<b>Descrição</b>	<b>Autor(s)</b>
27/09/2014	Concepção do Documento	fmbboaventura
12/08/2014	<ul style="list-style-type: none"> <li>• Inclusão dos requisitos [FR2];</li> <li>• Inclusão dos requisitos [NFR3];</li> <li>• Inclusão das dependências [D4];</li> </ul>	bezourokq, di3goleite, fmb-boaventura, gordinh, jadson-firmo, KelCarmo, mtcastro e wsbittencourt
30/09/2014	Inclusão dos requisitos não funcionais	gordinh, Figueiredovr
30/09/2014	Revisão do documento	gordinh
05/10/2014	Atualização dos requisitos	gordinh
07/10/2014	Atualização dos requisitos	gordinh, di3goleite
07/10/2014	Atualização dos requisitos	Figueiredovr
07/10/2014	Atualização dos requisitos não-funcionais	mtcastro
09/10/2014	Revisão do documento	gordinh, di3goleite
09/10/2014	Revisão do documento	Figueiredovr, mtcastro, di3goleite
20/10/2014	Refatoração do documento	jadsonfirmo
23/10/2014	Revisão	jadsonfirmo
28/10/2014	Atualização de Requisitos	jadsonfirmo
30/10/2014	Revisão Final	jadsonfirmo
11/12/2014	Refatoração do Documento: Revisão, correções e ajustes.	jadsonfirmo
11/12/2014	Modificação do requisito da instrução BRFL	KelCarmo
17/12/2014	Atualização das instruções	gordinh
17/12/2014	Revisão do documento	gordinh

Termo	Descrição
Requisitos Funcionais	Requisitos de <i>hardware</i> que compõem os módulos, descrevendo as ações que o mesmo deve estar apto a executar. Estas informações são capturadas a partir do desenvolvimento dos casos de uso, que documentam as entradas, os processos e as saídas geradas.
Requisitos Não Funcionais	Requisitos de <i>hardware</i> que compõem os módulos, representando as características que o mesmo deve ter, ou restrições que o mesmo deve operar. Estas características referem-se técnicas, algoritmos, tecnologias e especificidades do sistema como um todo.
Dependências	Requisitos de reuso de <i>IP-cores</i> , descrevendo as funções que cada um deve exercer.

### 1.3. Acrônimos e Abreviações

Sigla	Descrição
FR	Requisito Funcional
NFR	Requisito Não Funcional
D	Dependência
PC	<i>Program Counter</i> (Contador de Programa)
RD	Registrador de Destino
RS	<i>Register Source</i> Registrador Fonte
Bit	<i>Binary Digit</i> (Digito Binário)
Byte	<i>Binary Term</i> (Conjunto de 8 <i>bits</i> )

### 1.4. Prioridades dos Requisitos

Prioridade	Característica
Importante	Requisito sem o qual o sistema funciona, porém não como deveria.
Essencial	Requisito que deve ser implementado para que o sistema funcione.
Desejável	Requisito que não compromete o funcionamento do sistema.

## 2. Requisitos Funcionais

### 2.1. Instruções de Movimentação de Dados

#### [FR1] Instrução LW

---

**Descrição:** O processador deve ser capaz de ler valores da memória de dados. A instrução LW (*Load Word*) está compreendida da seguinte forma:

- Registrador de Destino (RD) - registrador onde será armazenado o valor que será carregado da memória.
- Registrador Fonte (RS) - registrador fonte onde contém o dado a ser lido.
- Deslocamento (I16) - endereço, de 16 *bits*, o qual haverá o deslocamento de *bits* a partir do RS.

**Nível de Prioridade:** Essencial

#### [FR2] Instrução SW

---

**Descrição:** O processador deve ser capaz de inserir valores na memória de dados. A instrução SW (*Store Word*) está compreendida da seguinte forma:

- Registrador Fonte (RS) - registrador fonte onde contém o dado a ser salvo.
- Registrador de Destino (RD) - registrador onde será armazenado o valor proveniente do RS.
- Deslocamento (I16) - endereço, de 16 *bits*, o qual haverá o deslocamento de *bits* a partir do RD.

**Nível de Prioridade:** Essencial

### 2.2. Computacionais

#### [FR3] Instrução ADD

---

**Descrição:** O processador deve ser capaz de realizar a soma de dois valores, levando em consideração o sinal. A instrução ADD está compreendida da seguinte forma:

- Registrador Fonte 1 (RS1) - registrador fonte representando o operando 1.
- Registrador Fonte 2 (RS2) - registrador fonte representando o operando 2.
- Registrador de Destino (RD) - registrador onde será armazenado o valor da soma entre o operando 1 e operando 2 ( $RS1 + RS2$ ).

**Nível de Prioridade: Essencial**

#### **[FR4] Instrução SUB**

---

**Descrição:** O processador deve ser capaz de realizar a subtração de dois valores, levando em consideração o sinal. A instrução SUB está compreendida da seguinte forma:

- Registrador Fonte 1 (RS1) - registrador fonte representando o operando 1.
- Registrador Fonte 2 (RS2) - registrador fonte representando o operando 2.
- Registrador de Destino (RD) - registrador onde será armazenado o valor da subtração entre o operando 1 e operando 2 ( $RS1 - RS2$ ).

**Nível de Prioridade: Essencial**

#### **[FR5] Instrução MUL**

---

**Descrição:** O processador deve ser capaz de realizar a multiplicação de dois valores, levando em consideração o sinal. A instrução MUL está compreendida da seguinte forma:

- Registrador Fonte 1 (RS1) - registrador fonte representando o operando 1.
- Registrador Fonte 2 (RS2) - registrador fonte representando o operando 2.
- Registrador de Destino (RD) - registrador onde será armazenado o valor da multiplicação entre o operando 1 e operando 2 ( $RS1 * RS2$ ).

**Nível de Prioridade: Essencial**



#### **[FR6] Instrução DIV**

---

**Descrição:** O processador deve ser capaz de realizar a divisão de dois valores, levando em consideração o sinal. A instrução DIV está compreendida da seguinte forma:

- Registrador Fonte 1 (RS1) - registrador fonte representando o operando 1.
- Registrador Fonte 2 (RS2) - registrador fonte representando o operando 2.
- Registrador de Destino (RD) - registrador onde será armazenado o valor da divisão entre o operando 1 e operando 2 ( $RS1 / RS2$ ).

**Nível de Prioridade: Essencial**

#### **[FR7] Instrução ADDi**

---

**Descrição:** O processador deve ser capaz de realizar a soma de dois valores (levando em consideração o sinal), de modo que um desses valores é um valor imediato. A instrução ADDi está compreendida da seguinte forma:

- Registrador Fonte 1 (RS1) - registrador fonte representando o operando 1.
- Imediato (I)- valor imediato, de 16 bits, representando o operando 2.
- Registrador de Destino (RD) - registrador onde será armazenado o valor da soma entre o operando 1 e o imediato ( $RS1 + I$ ).

**Nível de Prioridade: Essencial**

#### **[FR8] Instrução SUBi**

---

**Descrição:** O processador deve ser capaz de realizar a subtração de dois valores (levando em consideração o sinal), de modo que um desses valores é um valor imediato. A instrução SUBi está compreendida da seguinte forma:

- Registrador Fonte 1 (RS1) - registrador fonte representando o operando 1.

- Imediato (I)- valor imediato, de 16 bits, representando o operando 2.
- Registrador de Destino (RD) - registrador onde será armazenado o valor da subtração entre o operando 1 e o imediato ( $RS1 - I$ ).

#### Nível de Prioridade: I

mediato Essencial

### [FR9] Instrução CMP

---

**Descrição:** O processador deve ser capaz de comparar dois registradores e ativar ou desativar a *flag* equals para sinalizar igualdade caso os operandos sejam iguais, ou a *flag* above caso o operando 1 seja maior que o operando 2. A instrução CMP está compreendida da seguinte forma:

- Registrador 1 (RS1) - registrador representando um valor que será comparado com o RS2.
- Registrador 2 (RS2) - registrador representando um valor que será comparado com o RS1.

#### Nível de Prioridade: Essencial

### [FR10] Instrução AND

---

**Descrição:** O processador deve ser capaz de realizar a operação lógica AND, *bit a bit*, de dois valores. A instrução AND está compreendida da seguinte forma:

- Registrador Fonte 1 (RS1) - registrador fonte representando o operando 1.
- Registrador Fonte 2 (RS2) - registrador fonte representando o operando 2.
- Registrador de Destino (RD) - registrador onde será armazenado o valor da operação AND entre o operando 1 e operando 2 ( $RS1 \& RS2$ ).

#### Nível de Prioridade: Essencial

### [FR11] Instrução OR

---

**Descrição:** O processador deve ser capaz de realizar a operação lógica OR, *bit a bit*, de dois valores. A instrução OR está compreendida da seguinte forma:

- Registrador Fonte 1 (RS1) - registrador fonte representando o operando 1.
- Registrador Fonte 2 (RS2) - registrador fonte representando o operando 2.
- Registrador de Destino (RD) - registrador onde será armazenado o valor da operação OR entre o operando 1 e operando 2 (RS1 || RS2).

**Nível de Prioridade: Essencial**

#### [FR12] Instrução NOT

---

**Descrição:** O processador deve ser capaz de realizar a operação lógica NOT, de inversão, *bit a bit*. A instrução NOT está compreendida da seguinte forma:

- Registrador Fonte (RS) - registrador fonte representando o operando que será negado.
- Registrador de Destino (RD) - registrador onde será armazenado o valor a ser negado.

**Nível de Prioridade: Essencial**

#### [FR13] Instrução ANDi

---

**Descrição:** O processador deve ser capaz de realizar a operação lógica AND, *bit a bit*, de dois valores, de modo que um desses valores é um valor imediato. A instrução ANDi está compreendida da seguinte forma:

- Registrador Fonte 1 (RS1) - registrador fonte representando o operando 1.
- Imediato (I)- valor imediato, de 16 bits, representando o operando 2.
- Registrador de Destino (RD) - registrador onde será armazenado o valor da operação AND entre o operando 1 e o imediato (RS1 & I).

**Nível de Prioridade: Essencial**

#### [FR14] Instrução ORi

---

**Descrição:** O processador deve ser capaz de realizar a operação lógica OR, *bit a bit*, de dois valores, de modo que um desses valores é um valor imediato. A instrução ORi está compreendida da seguinte forma:

- Registrador Fonte 1 (RS1) - registrador fonte representando o operando 1.
- Imediato (I)- valor imediato, de 16 bits, representando o operando 2.
- Registrador de Destino (RD) - registrador onde será armazenado o valor da operação OR entre o operando 1 e o imediato (RS1 || I).

**Nível de Prioridade: Essencial**

### 2.3. Instruções de Desvio

#### [FR15] Instrução JR

---

**Descrição:** O processador deve ser capaz de desviar um programa em execução para um endereço de destino. A instrução JR está compreendida da seguinte forma:

- Registrador Endereço (R) - registrador onde contém o endereço para onde o programa deverá ir.

**Nível de Prioridade: Essencial**

#### [FR16] Instrução JPC

---

**Descrição:** O processador deve ser capaz de desviar um programa em execução para um endereço relativo ao PC. A instrução JPC está compreendida da seguinte forma:

- Registrador de Endereço (RD) - registrador que contém o valor, de 28 bits, relativo ao PC, para onde o programa deverá ser desviado.

**Nível de Prioridade: Essencial**

#### [FR17] Instrução BRFL

---

**Descrição:** O processador deve ser capaz de desviar um programa em execução para um endereço de destino atendendo a uma condição de *flag*. A instrução BRFL está compreendida da seguinte forma:

- Registrador (R) - registrador que contém a constante que será comprada com o valor de *flag*.
- Imediato - valor do endereço destino, no qual será tomado se a condição for verdadeira.

**Nível de Prioridade: Essencial**

#### [FR18] Instrução CALL

---

**Descrição:** O processador deve ser capaz de desviar um programa em execução para uma sub-rotina. A instrução CALL está compreendida da seguinte forma:

- Registrador de Destino (RD) - registrador onde contém o valor para onde o programa deverá ir. Essa instrução salva o endereço atual do PC, armazenado-o na pilha.

**Nível de Prioridade: Essencial**

#### [FR19] Instrução RET

---

**Descrição:** O processador deve ser capaz de retornar de uma sub-rotina. A instrução RET está compreendida da seguinte forma:

- A instrução deverá acessar a pilha procurando pelo endereço de retorno. Ao acessar a pilha, que contém os endereços do PC salvos, o programa deve ir para o endereço que está no topo da pilha.

**Nível de Prioridade: Essencial**

### 2.4. Outras Instruções

#### [FR20] Instrução NOP

---

**Descrição:** O processador deve ser capaz de não realizar operações durante os 5 ciclos de *clock*.

**Nível de Prioridade: Essencial**

#### [FR21] Instrução HALT

---

**Descrição:** O processador deve ser capaz de parar a execução de um programa.

A instrução HALT está compreendida da seguinte forma:

- Imediato (I) - valor imediato, de 26 bits, representando o endereço da instrução HALT para fazer o desvio, interrompendo assim a execução do programa.

**Nível de Prioridade:** Essencial

### 2.5. Flags

#### [FR22] Overflow/Underflow

---

**Descrição:** O processador deve ser capaz de avisar que houve um erro (ou um "estouro") na operação aritmética através da *flag* de *overflow/underflow*.

**Nível de Prioridade:** Essencial

#### [FR23] Equals

---

**Descrição:** Esta *flag* deve ser utilizada como resultado da instrução CMP, e deve constar verdadeiro (valor 1) quando as duas palavras comparadas forem iguais.

**Nível de Prioridade:** Essencial

#### [FR24] Above

---

**Descrição:** Esta *flag* deve ser utilizada como resultado da instrução CMP, e deve constar verdadeiro quando o valor de uma palavra é maior do que o valor da outra.

**Nível de Prioridade:** Essencial

### 2.6. Outros Requisitos Funcionais

#### [FR25] Tamanho da palavra de uma instrução

---

**Descrição:** O tamanho da palavra de instrução, que é compatível com o processador, é de 32 *bits*.

**Nível de Prioridade:** Essencial

#### [FR26] Tamanho de endereço

---

**Descrição:** O processador trabalha com endereços de 18 bits, para a memória de instrução e 32 bits, para a memória de dados.

**Nível de Prioridade:** Essencial

#### [FR27] Pilha

---

**Descrição:** O processador trabalha com uma pilha separada, dedicada somente a guardar endereços do PC para o retorno de funções.

**Nível de Prioridade:** Essencial

### 2.7. Modos de Endereçamento

#### [FR28] Imediato

---

**Descrição:** O processador deve aceitar instruções onde o dado já é passado explicitamente na instrução.

**Nível de Prioridade:** Essencial

#### [FR29] Deslocamento de Base

---

**Descrição:** O processador deve aceitar instruções onde os operandos contenham o endereço da base e o valor do deslocamento.

**Nível de Prioridade:** Essencial

#### [FR30] Por Registrador

---

**Descrição:** O processador deve aceitar instruções onde o endereço do registrador seja passado como parâmetro.

**Nível de Prioridade:** Essencial

### 3. Requisitos não Funcionais

Esta seção apresenta a lista de Requisitos Não Funcionais do projeto.

#### [NFR1] Ferramenta para simulação de testes

---

**Descrição:** Será utilizado o programa ModelSim®-Altera Web Edition, para fazer a simulação dos módulos e testes dos mesmos.

**Nível de Prioridade: Importante**

---

**[NFR2] Ferramenta para prototipação**

---

**Descrição:** Será utilizada a Plataforma de Desenvolvimento FPGA (Terasic ALTERA Cyclone III (EP3C25F324)) para a execução do protótipo.

**Nível de Prioridade: Importante**

---

**[NFR3] Linguagem de Descrição**

---

**Descrição:** Tanto o projeto quanto os testes serão descritos usando Verilog-HDL.

**Nível de Prioridade: Desejável**

---

**[NFR4] Plano de Teste**

---

**Descrição:** Será desenvolvido um conjunto de programa de teste para cada bloco implementado no projeto.

**Nível de Prioridade: Desejável**

---

**[NFR5] Organização dos Dados**

---

**Descrição:** Os *bytes* são numerados da esquerda para a direita: formato *Big Endian*.

**Nível de Prioridade: Essencial**

---

**[NFR6] Ferramenta para programar o processador**

---

**Descrição:** Será utilizado o *software* Quartus para descarregar os programas que serão executados no processador.

**Nível de Prioridade: Importante**

---

**[NFR7] Tempo de Operação**

---

**Descrição:** Toda instrução deve ser executada em exatamente cinco ciclos de *clock*.

**Nível de Prioridade: Essencial**



#### 4. Dependências

##### [D1] ULA

---

Módulo da ULA implementado no projeto *Warmup*, contando com algumas adaptações e incremento de operações.