



**Documento de Casos de Uso**

MUSA

Fazemos Qualquer Negócio Inc.

**Compilação 2.0**

## Histórico de Revisões

Date	Descrição	Autor(s)
03/10/2014	Document conception	manuellemacedo
07/10/2014	Adição do caso de uso da unidade lógica e aritmética	manuellemacedo
09/10/2014	Retirado alguns atores do documento. Modificado os Casos de uso da Unidade de Controle e da UL. Adicionado algumas definições de siglas do Diagrama de Caso de Uso.	Lucas e Anderson
09/10/2014	Formatação e ajuste do documento	santana22
09/10/2014	Adição do caso de uso [UC 005]	tarleswalker
09/10/2014	Adição dos casos de usos [UC 002] e [UC 003]	manuellemacedo
09/10/2014	Adição do caso de uso [UC 004]	weverson
16/10/2014	Refatoração da capa,	manuellemacedo
24/10/2014	Edição do caso de uso unidade de controle	mirelarios e patrickecomp
24/10/2014	Ajustes de sintaxe	manuellemacedo
29/10/2014	Edição do caso de uso Extensor de Bits	mirelarios e patrickecomp
14/12/2014	Refatoração do documento	santana22
14/12/2014	Refatoração do documento	santana22 e tarleswalker

## SUMÁRIO

<b>1</b>	<b>Introdução</b>	<b>4</b>
1.1	Objetivo . . . . .	4
1.2	Visão Geral do Documento . . . . .	4
1.3	Representação Simbólica . . . . .	4
1.4	Definições, Acrônimos e Abreviações . . . . .	5
<b>2</b>	<b>Atores do Sistema</b>	<b>5</b>
<b>3</b>	<b>Casos de Usos</b>	<b>5</b>
3.1	[UC 001] Instrução ADD . . . . .	5
3.2	[UC 002] Instrução SUB . . . . .	6
3.3	[UC 003] Instrução MUL . . . . .	7
3.4	[UC 004] Instrução DIV . . . . .	8
3.5	[UC 005] Instrução ADDI . . . . .	8
3.6	[UC 006] Instrução SUBI . . . . .	9
3.7	[UC 007] Instrução MULI . . . . .	10
3.8	[UC 008] Instrução AND . . . . .	11
3.9	[UC 009] Instrução OR . . . . .	11
3.10	[UC 010] Instrução NOT . . . . .	12
3.11	[UC 011] Instrução ANDI . . . . .	12
3.12	[UC 012] Instrução ORI . . . . .	13
3.13	[UC 013] Instrução CMP . . . . .	14
3.14	[UC 014] Instrução JR . . . . .	15
3.15	[UC 015] Instrução JPC . . . . .	16
3.16	[UC 016] Instrução BRFL . . . . .	16

3.17	[UC 017] Instrução CALL . . . . .	17
3.18	[UC 018] Instrução RET . . . . .	17
3.19	[UC 019] Instrução HALT . . . . .	18

## 1. Introdução

### 1.1. Objetivo

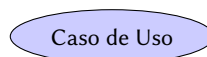
O objetivo desse documento é especificar os casos de uso do projeto MUSA. O documento contempla as seguintes informações: descrição dos Atores envolvidos no processo; definição dos fluxos de eventos principal e secundário; lista de requisitos essenciais, funcionais e não funcionais; estabelecimento de pré-condições e pós-condições.

### 1.2. Visão Geral do Documento

- Sessão 2: lista todos os possíveis atores do sistema.
- Sessão 3: relata a lista dos casos de uso do projeto.

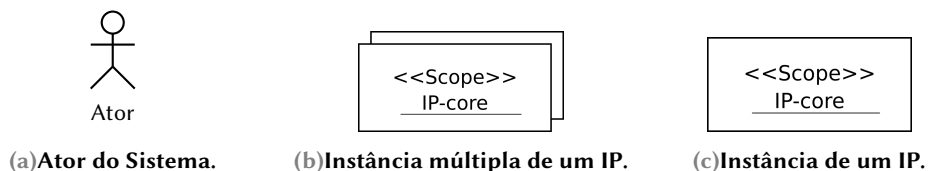
### 1.3. Representação Simbólica

A Figura 1 ilustra a simbologia utilizada para representar operações que devem ser realizadas pelo sistema. A Figura 2 ilustra as duas simbologias utilizadas para representar os Atores do sistema. Um ator, dentro do escopo desta descrição, pode ser identificado como um módulo *top level*, ou como um elemento de entrada e saída (botões, sensores, displays, etc).



**Figura 1: Exemplo de Caso de Uso.**

A simbologia usual para representação de um Ator é apresentada na Figura 2a, no entanto, para representar módulos incorporados que outrora deveriam utilizar a mesma simbologia, utiliza-se a representação ilustrada nas Figuras 2b e 2c, definida por convenção. Este elemento, em geral, está associado aos módulos do sistema, ou IP-cores que de terceiros incorporados ao mesmo. Esta simbologia ainda foi dividida, tendo em vista representar instâncias únicas (Figura 2c), ou múltiplas (Figura 2b) de um determinado componente.



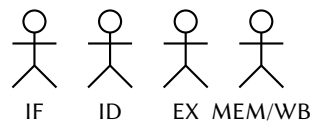
**Figura 2: Simbologia utilizada na implementação dos Casos de Uso.**

O projetista responsável por interpretar os diagramas não deve confundir-se no momento de interpretar as simbologias de atores. A representação alternativa, não implica que o módulo será instanciado no subsistema em questão, mas sim que os recursos providos por este *core* são necessários para garantir o seu funcionamento.

## 1.4. Definições, Acrônimos e Abreviações

Termo	Descrição
UC	Caso de Uso
IF	Busca de Instrução
ID	Decodificação de Instrução
EX	Execução
MEM/WB	Acesso a Memória/Escrita

## 2. Atores do Sistema



**IF** – Módulo responsável por buscar a instrução a ser executada.

**ID** – Módulo responsável por decodificar a instrução encontrada e armazenar alguns dados.

**EX** – Módulo responsável pela execução das operações.

**MEM/WB** – Módulo onde se realiza o acesso a memória e armazenamento no módulo ID.

## 3. Casos de Usos

### 3.1. [UC 001] Instrução ADD

Este caso de uso especifica a ação de execução de uma instrução de soma. O objetivo é realizar uma soma entre dois valores.

*Atores*

**IF, ID, EX, MEM/WB**

*Pré-condições*

- O módulo **IF** precisa buscar uma instrução ADD.

#### *Pós-condições*

- Permitir a execução da próxima instrução.

#### *Fluxo Básico*

- O módulo **IF** busca a instrução ADD a ser executada.
- O módulo **ID** decodifica o tipo da instrução, emite sinais de controle para os demais módulos e disponibiliza os valores para o módulo **EX**.
- O módulo **EX** realiza a operação de soma dos dois valores e atualiza para o endereço da próxima instrução.
- O módulo **MEM/WB** armazena o resultado no módulo **ID**.

#### *Fluxos Alternativos*

- O resultado da operação pode não estar na faixa de valores suportados:  
O módulo **EX** detecta e registra a ocorrência.

### **3.2. [UC 002] Instrução SUB**

Este caso de uso especifica a ação da instrução de subtração. O objetivo é realizar a subtração entre dois valores.

#### *Atores*

**IF, ID, EX, MEM/WB**

#### *Pré-condições*

- O módulo **IF** precisa buscar uma instrução SUB.

#### *Pós-condições*

- Permitir a execução da próxima instrução.

#### *Fluxo Básico*

- O módulo **IF** busca a instrução SUB a ser executada.
- O módulo **ID** decodifica o tipo da instrução, emite sinais de controle para os demais módulos e disponibiliza os valores para o módulo **EX**.

- O módulo **EX** realiza a operação de subtração dos dois valores e atualiza para o endereço da próxima instrução.
- O módulo **MEM/WB** armazena o resultado da subtração no módulo **ID**.

#### *Fluxos Alternativos*

- O resultado da operação pode não estar na faixa de valores suportados:  
O módulo **EX** detecta e registra a ocorrência.

### **3.3. [UC 003] Instrução MUL**

Este caso de uso especifica a ação da instrução de multiplicação. Seu objetivo é realizar a multiplicação entre dois valores.

#### *Atores*

**IF, ID, EX, MEM/WB**

#### *Pré-condições*

- O módulo **IF** precisa buscar uma instrução **MUL**.

#### *Pós-condições*

- Permitir a execução da próxima instrução.

#### *Fluxo Básico*

- O módulo **IF** busca a instrução **MUL** a ser executada.
- O módulo **ID** decodifica o tipo da instrução, emite sinais de controle para os demais módulos e disponibiliza os valores para o módulo **EX**.
- O módulo **EX** realiza a operação de multiplicação dos dois valores e atualiza para o endereço da próxima instrução.
- O módulo **MEM/WB** armazena o resultado da multiplicação no módulo **ID**.

#### *Fluxos Alternativos*

- O resultado da operação pode não estar na faixa de valores suportados:  
O módulo **EX** detecta e registra a ocorrência.



### 3.4. [UC 004] Instrução DIV

Este caso de uso especifica a ação da instrução de divisão. O objetivo é realizar a divisão entre dois valores.

*Atores*

**IF, ID, EX, MEM/WB**

*Pré-condições*

- O módulo **IF** precisa buscar uma instrução DIV.

*Pós-condições*

- Permitir a execução da próxima instrução.

*Fluxo Básico*

- O módulo **IF** busca a instrução DIV a ser executada.
- O módulo **ID** decodifica o tipo da instrução, emite sinais de controle para os demais módulos e disponibiliza os valores para o módulo **EX**.
- O módulo **EX** realiza a operação de divisão dos valores e atualiza para o endereço da próxima instrução.
- O módulo **MEM/WB** armazena o resultado da divisão no módulo **ID**.

*Fluxos Alternativos*

- O resultado da operação pode não estar na faixa de valores suportados:  
O módulo **EX** detecta e registra a ocorrência.
- O divisor disponibilizado pelo módulo **ID** para o módulo **EX** for 0 (zero):  
O módulo **EX** detecta e registra a ocorrência.

### 3.5. [UC 005] Instrução ADDI

Este caso de uso especifica a ação de execução de uma instrução de soma imediata. O objetivo é realizar uma soma entre dois valores sendo um destes imediato.

*Atores*

**IF, ID, EX, MEM/WB**

#### *Pré-condições*

- O módulo **IF** precisa buscar uma instrução ADDI.

#### *Pós-condições*

- Permitir a execução da próxima instrução.

#### *Fluxo Básico*

- O módulo **IF** busca a instrução ADDI a ser executada.
- O módulo **ID** decodifica o tipo da instrução, emite sinais de controle para os demais módulos e disponibiliza os valores para o módulo **EX**.
- O módulo **EX** realiza a operação de soma dos dois valores e atualiza para o endereço da próxima instrução.
- O módulo **MEM/WB** armazena o resultado no módulo **ID**.

#### *Fluxos Alternativos*

- O resultado da operação pode não estar na faixa de valores suportados:  
O módulo **EX** detecta e registra a ocorrência.

### **3.6. [UC 006] Instrução SUBI**

Este caso de uso especifica a ação de execução de uma instrução de subtração imediata. O objetivo é realizar uma subtração entre dois valores sendo um destes imediato.

#### *Atores*

**IF, ID, EX, MEM/WB**

#### *Pré-condições*

- O módulo **IF** precisa buscar uma instrução SUBI.

#### *Pós-condições*

- Permitir execução da próxima instrução.

#### *Fluxo Básico*

- O módulo **IF** busca a instrução SUBI a ser executada.

- O módulo **ID** decodifica o tipo da instrução, emite sinais de controle para os demais módulos e disponibiliza os valores para o módulo **EX**.
- O módulo **EX** realiza a operação de subtração dos dois valores e atualiza para o endereço da próxima instrução.
- O módulo **MEM/WB** armazena o resultado da subtração no módulo **ID**.

#### *Fluxos Alternativos*

- O resultado da operação pode não estar na faixa de valores suportados:  
O módulo **EX** detecta e registra a ocorrência.

### 3.7. [UC 007] Instrução MULI

Este caso de uso especifica a ação de execução de uma instrução de multiplicação imediata. O objetivo é realizar uma multiplicação entre dois valores sendo um destes imediato.

#### *Atores*

**IF, ID, EX, MEM/WB**

#### *Pré-condições*

- O módulo **IF** precisa buscar uma instrução MULI.

#### *Pós-condições*

- Permitir a execução da próxima instrução.

#### *Fluxo Básico*

- O módulo **IF** busca a instrução MULI a ser executada.
- O módulo **ID** decodifica o tipo da instrução, emite sinais de controle para os demais módulos e disponibiliza os valores para o módulo **EX**.
- O módulo **EX** realiza a operação de multiplicação dos dois valores e atualiza para o endereço da próxima instrução.
- O módulo **MEM/WB** armazena o resultado da multiplicação no módulo **ID**.

#### *Fluxos Alternativos*

- O resultado da operação pode não estar na faixa de valores suportados:  
O módulo **EX** detecta e registra a ocorrência.

### 3.8. [UC 008] Instrução AND

Este caso de uso especifica a ação da instrução AND. O objetivo é realizar uma a operação lógica AND entre dois valores.

*Atores*

**IF, ID, EX, MEM/WB**

*Pré-condições*

- O módulo **IF** precisa buscar uma instrução AND.

*Pós-condições*

- Permitir execução da próxima instrução.

*Fluxo Básico*

- O módulo **IF** busca a instrução AND a ser executada.
- O módulo **ID** decodifica o tipo da instrução, emite sinais de controle para os demais módulos e disponibiliza os valores para o módulo **EX**.
- O módulo **EX** realiza a operação lógica determinada dos valores e atualiza para a próxima instrução.
- O módulo **MEM/WB** armazena o resultado da operação no módulo **ID**.

### 3.9. [UC 009] Instrução OR

Este caso de uso especifica a ação da instrução OR. O objetivo é realizar a operação lógica OR entre dois valores.

*Atores*

**IF, ID, EX, MEM/WB**

*Pré-condições*

- O módulo **IF** precisa buscar uma instrução OR.

*Pós-condições*

- Permitir execução da próxima instrução.

### *Fluxo Básico*

- O módulo **IF** busca a instrução OR a ser executada.
- O módulo **ID** decodifica o tipo da instrução, emite sinais de controle para os demais módulos e disponibiliza os valores para o módulo **EX**.
- O módulo **EX** realiza a operação lógica determinada dos valores e atualiza para a próxima instrução.
- O módulo **MEM/WB** armazena o resultado da operação no módulo **ID**.

### **3.10. [UC 010] Instrução NOT**

Este caso de uso especifica a ação da instrução NOT. O objetivo é realizar a operação lógica NOT sobre um valor.

#### *Atores*

**IF, ID, EX, MEM/WB**

#### *Pré-condições*

- O módulo **IF** precisa buscar uma instrução NOT.

#### *Pós-condições*

- Permitir execução de futuras instruções.

### *Fluxo Básico*

- O módulo **IF** busca a instrução NOT a ser executada.
- O módulo **ID** decodifica o tipo da instrução, emite sinais de controle para os demais módulos e disponibiliza o valor para o módulo **EX**.
- O módulo **EX** realiza a operação lógica determinada de negação do valor e atualiza para a próxima instrução.
- O módulo **MEM/WB** armazena o resultado da operação no módulo **ID**.

### **3.11. [UC 011] Instrução ANDI**

Este caso de uso especifica a ação da instrução ANDI. O objetivo é realizar uma a operação lógica AND entre dois valores sendo um destes imediato.

*Atores*

**IF, ID, EX, MEM/WB**

*Pré-condições*

- O módulo **IF** precisa buscar uma instrução ANDI.

*Pós-condições*

- Permitir execução da próxima instrução.

*Fluxo Básico*

- O módulo **IF** busca a instrução ANDI a ser executada.
- O módulo **ID** decodifica o tipo da instrução, emite sinais de controle para os demais módulos e disponibiliza os valores para o módulo **EX**.
- O módulo **EX** realiza a operação lógica determinada dos valores e atualiza para a próxima instrução.
- O módulo **MEM/WB** armazena o resultado da operação no módulo **ID**.

### 3.12. [UC 012] Instrução ORI

Este caso de uso especifica a ação da instrução ORI. O objetivo é realizar a operação lógica OR entre dois valores sendo um destes imediato.

*Atores*

**IF, ID, EX, MEM/WB**

*Pré-condições*

- O módulo **IF** precisa buscar uma instrução ORI.

*Pós-condições*

- Permitir execução da próxima instrução.

*Fluxo Básico*

- O módulo **IF** busca a instrução ORI a ser executada.

- O módulo **ID** decodifica o tipo da instrução, emite sinais de controle para os demais módulos e disponibiliza os valores para o módulo **EX**.
- O módulo **EX** realiza a operação lógica determinada dos valores e atualiza para a próxima instrução.
- O módulo **MEM/WB** armazena o resultado da operação no módulo **ID**.

### 3.13. [UC 013] Instrução CMP

Este caso de uso especifica a ação da instrução CMP. O objetivo é realizar a comparação entre dois valores.

*Atores*

**IF, ID, EX**

*Pré-condições*

- O módulo **IF** precisa buscar uma instrução CMP.

*Pós-condições*

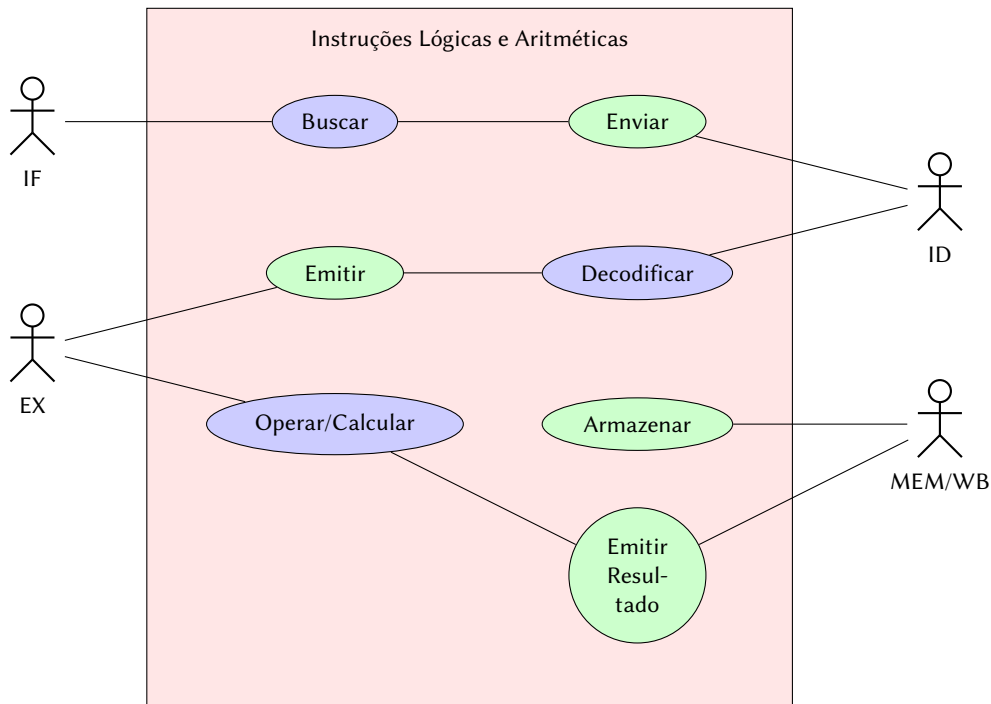
- Permitir execução da próxima instrução.

*Fluxo Básico*

- O módulo **IF** busca a instrução CMP a ser executada.
- O módulo **ID** decodifica o tipo da instrução, emite sinais de controle para os demais módulos e disponibiliza os valores para o módulo **EX**.
- O módulo **EX** realiza a comparação dos valores, atualiza o registrador de ocorrências e atualiza para a próxima instrução.

Basicamente, as instruções Lógicas e Aritméticas seguem um mesmo fluxo de execução:

## Diagrama de Caso de Uso



### 3.14. [UC 014] Instrução JR

Este caso de uso especifica a ação da instrução de desvio incondicional JR. O objetivo é realizar um salto para o endereço armazenado em um registrador.

#### Atores

**IF, ID**

#### Pré-condições

- O módulo **IF** precisa buscar uma instrução JR.

#### Pós-condições

- Permitir a execução da instrução endereçada pela conteúdo do registrador.

#### Fluxo Básico

- O módulo **IF** busca a instrução JR a ser executada.
- O módulo **ID** decodifica o tipo da instrução, emite sinais de controle para os demais módulos e seleciona o conteúdo do registrador.



### 3.15. [UC 015] Instrução JPC

Este caso de uso especifica a ação da instrução de desvio incondicional JPC. O objetivo é realizar um salto para o endereço relativo ao endereço da instrução atual.

*Atores*

**IF, ID, EX**

*Pré-condições*

- O módulo **IF** precisa buscar uma instrução JPC.

*Pós-condições*

- Permitir a execução da instrução com o endereço relativo ao endereço da instrução atual.

*Fluxo Básico*

- O módulo **IF** busca a instrução JPC a ser executada.
- O módulo **ID** decodifica o tipo da instrução, emite sinais de controle para os demais módulos e seleciona o valor do desvio.
- O módulo **EX** realiza a operação de soma do endereço da instrução atual com o valor selecionado e atualiza o endereço da próxima instrução.

### 3.16. [UC 016] Instrução BRFL

Este caso de uso especifica a ação da instrução de desvio condicional BRFL. O objetivo é realizar um salto para um endereço com base nas ocorrências registradas pelo módulo **EX**.

*Atores*

**IF, ID, EX**

*Pré-condições*

- O módulo **IF** precisa buscar uma instrução BRFL.

*Pós-condições*

- Permitir a execução da próxima instrução.

- O registrador de ocorrências deve ser limpo.

#### *Fluxo Básico*

- O módulo **IF** busca a instrução BRFL a ser executada.
- O módulo **ID** decodifica o tipo da instrução, emite sinais de controle para os demais módulos.
- O módulo **EX** verifica qual foi a ocorrência. Com base nisso, modifica o endereço da próxima instrução.

### 3.17. [UC 017] Instrução CALL

Este caso de uso especifica a ação da instrução de desvio incondicional CALL. O objetivo é realizar um salto para uma sub-rotina.

#### *Atores*

**IF, ID**

#### *Pré-condições*

- O módulo **IF** precisa buscar uma instrução CALL.

#### *Pós-condições*

- Permitir a execução da sub-rotina.

#### *Fluxo Básico*

- O módulo **IF** busca a instrução CALL a ser executada.
- O módulo **ID** decodifica o tipo da instrução, emite sinais de controle para os demais módulos.

### 3.18. [UC 018] Instrução RET

Este caso de uso especifica a ação da instrução de desvio incondicional RET. O objetivo é realizar um salto de uma sub-rotina.

#### *Atores*

**IF, ID**

#### *Pré-condições*

- O módulo **IF** precisa buscar uma instrução RET.

#### *Pós-condições*

- Permitir a execução da instrução armazenada no topo da Pilha de Instruções.

#### *Fluxo Básico*

- O módulo **IF** busca a instrução RET a ser executada.
- O módulo **ID** decodifica o tipo da instrução, emite sinais de controle para os demais módulos.

### 3.19. [UC 019] Instrução HALT

Este caso de uso especifica a ação da instrução HALT. O objetivo é finalizar a execução de novas instruções.

#### *Atores*

#### **IF, ID, EX**

#### *Pré-condições*

- O módulo **IF** precisa buscar uma instrução HALT.

#### *Pós-condições*

- Nenhuma instrução deve ser executada.

#### *Fluxo Básico*

- O módulo **IF** busca a instrução HALT a ser executada.
- O módulo **ID** decodifica o tipo da instrução, emite sinais de controle para os demais módulos.
- O módulo **EX** atualiza o valor da próxima instrução para o endereço da instrução atual (*loop* infinito).

De maneira análoga as instruções Lógicas e Aritméticas, as Instruções de Jump seguem um mesmo fluxo de execução:

## Diagrama de Caso de Uso

