



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

SUMÁRIO

1	Introdução	3
1.1	Objetivo	3
1.2	Visão Geral do Documento	3
1.3	Representação Simbólica	3
1.4	Definições, Acrônimos e Abreviações	4
2	Atores do Sistema	4
3	Casos de Usos	4
3.1	[UC 001] Instrução ADD	4
3.2	[UC 002] Instrução SUB	5
3.3	[UC 003] Instrução MUL	6
3.4	[UC 004] Instrução DIV	7
3.5	[UC 005] Instrução AND	7
3.6	[UC 006] Instrução OR	8
3.7	[UC 007] Instrução NOT	9
3.8	[UC 008] Instrução NOP	9
3.9	[UC 009] Instrução JR	11
3.10	[UC 010] Instrução JPC	11
3.11	[UC 011] Instrução BRFL	12
3.12	[UC 012] Instrução CALL	12
3.13	[UC 013] Instrução RET	13
3.14	[UC 014] Instrução HALT	13

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).

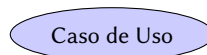


Figura 1: Exemplo de Caso de Uso.

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.

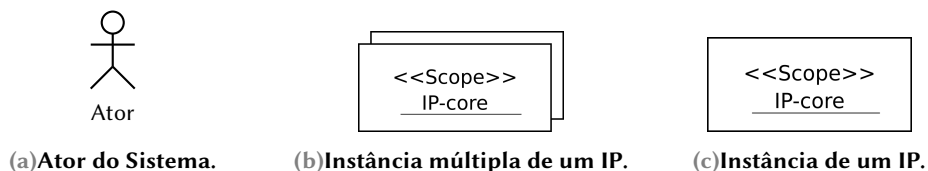


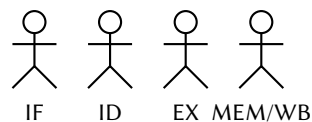
Figura 2: Simbologia utilizada na implementação dos Casos de Uso.

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
FR	Requisito Funcional
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, principal objetivo é realizar uma soma entre dois valores de 32 bits.

Atores

IF, ID, EX, MEM/WB

Pré-condições

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

Pós-condições

- Permitir execução de futuras instruções.

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 a próxima instrução.
- O módulo **MEM/WB** armazena o resultado no módulo **ID**.

Fluxos Alternativos

- O resultado da operação pode ser maior ou menor do que o suportado:
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, principal objetivo é realizar a subtração entre dois valores de 32 bits.

Atores

IF, ID, EX, MEM/WB

Pré-condições

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

Pós-condições

- Permitir execução de futuras instruções.

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 a próxima instrução.
- O módulo **MEM/WB** armazena o resultado da subtração no módulo **ID**.

Fluxos Alternativos

- O resultado da operação pode ser maior ou menor do que o suportado:
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 execução de futuras instruções.

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 a próxima instrução.
- O módulo **MEM/WB** armazena o resultado da multiplicação no módulo **ID**.

Fluxos Alternativos

- O resultado da operação pode ser maior ou menor do que o suportado:
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, seu 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 execução de futuras instruções.

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 a próxima instrução.
- O módulo **MEM/WB** armazena o resultado da divisão no módulo **ID**.

Fluxos Alternativos

- 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 AND

Este caso de uso especifica a ação da instrução AND (também chamada de conjunção lógica), que realiza a operação lógica entre dois valores.

Atores

IF, ID, EX, MEM/WB

Pré-condições

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

Pós-condições

- Permitir execução de futuras instruções.

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**.

Fluxos Alternativos

- Esta instrução não possui fluxos alternativos.

3.6. [UC 006] Instrução OR

Este caso de uso especifica a ação da instrução OR (também chamada de disjunção lógica), que realiza a operação lógica entre dois valores.

Atores

IF, ID, EX, MEM/WB

Pré-condições

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

Pós-condições

- Permitir execução de futuras instruções.

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**.

Fluxos Alternativos

- Esta instrução não possui fluxos alternativos.

3.7. [UC 007] Instrução NOT

Este caso de uso especifica a ação da instrução NOT (também chamada de inversora), pois realiza a negação (inversão) lógica de 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**.

Fluxos Alternativos

- Esta instrução não possui fluxos alternativos.

3.8. [UC 008] Instrução NOP

Este caso de uso especifica a ação da instrução NOP, essa instrução não realiza nenhuma operação.

Atores

IF, ID, EX

Pré-condições

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

Pós-condições

- Permitir execução de futuras instruções.

Fluxo Básico

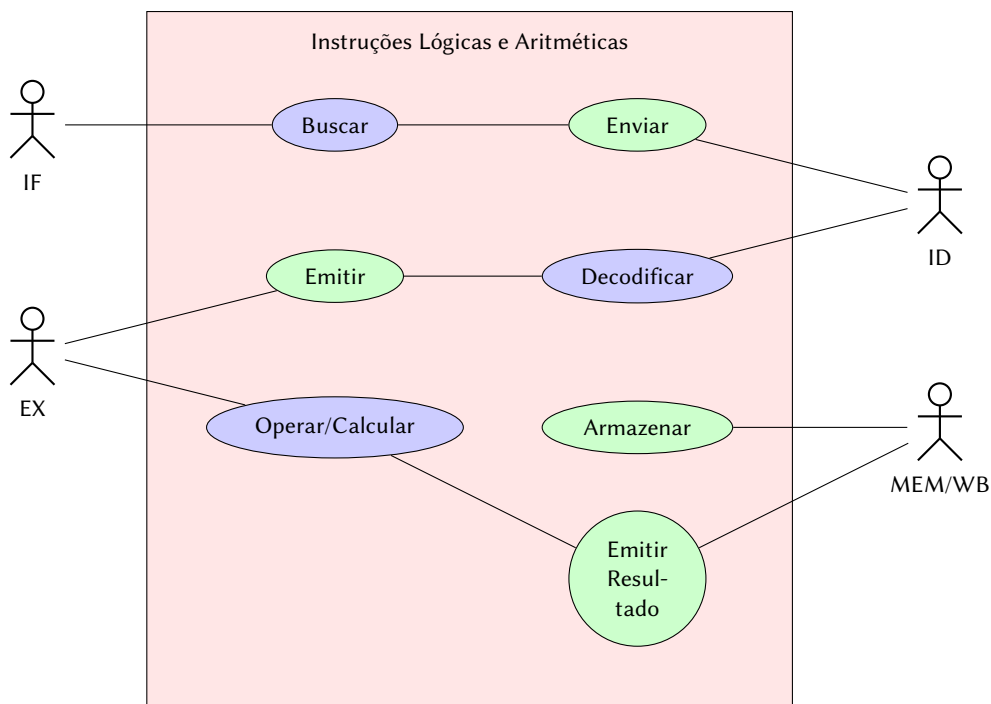
- O módulo **IF** busca a instrução NOP a ser executada.
- O módulo **ID** decodifica o tipo da instrução, emite sinais de controle para os demais módulos.
- O módulo **EX** atualiza para a próxima instrução.

Fluxos Alternativos

- Esta instrução não possui fluxos alternativos.

Basicamente as instruções lógicas e Aritméticas seguem um mesmo fluxo de execução.

Diagrama de Caso de Uso



3.9. [UC 009] Instrução JR

Este caso de uso especifica a ação da instrução de desvio incondicional JR. O principal 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.

3.10. [UC 010] Instrução JPC

Este caso de uso especifica a ação da instrução de desvio incondicional JPC. O principal 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.

3.11. [UC 011] Instrução BRFL

Este caso de uso especifica a ação da instrução de desvio condicional BRFL. O principal 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(is) foi(ram) a(s) ocorrência(s). Com base nisso, modifica o endereço da próxima instrução.

3.12. [UC 012] Instrução CALL

Este caso de uso especifica a ação da instrução de desvio incondicional CALL. O principal 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.

3.13. [UC 013] Instrução RET

Este caso de uso especifica a ação da instrução de desvio incondicional RET. O principal 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.

3.14. [UC 014] Instrução HALT

Este caso de uso especifica a ação da instrução HALT. O principal objetivo é finalizar a execução de novas instruções, parar o sistema.

Atores

IF, ID, EX

Pré-condições

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

Pós-condições

- Não ser executado mais nenhuma instrução.

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** não atualiza o valor para próxima instrução, ficando direcionado para a mesma instrução HALT (loop infinito).

Diagrama de Caso de Uso

