Documento de Casos de Uso

MUSA

Fazemos Qualquer Negócio Inc.

Build 1.1a

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

SUMÁRIO

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 **??** ilustra a simbologia utilizada para representar operações que devem ser realizadas pelo sistema. A Figura **??** 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).

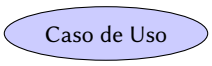
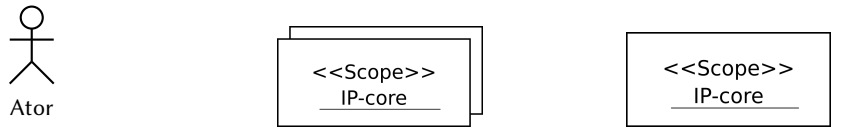


Figura 1: Exemplo de Caso de Uso.

A simbologia usual para representação de um Ator é apresentada na Figura **??**, no entanto, para representar módulos incorporados que outrora deveriam utilizar a mesma simbologia, utiliza-se a representação ilustrada nas Figuras **??** e **??**, 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 divida, tendo em vista representar instâncias únicas (Figura **??**), ou múltiplas (Figura **??**) de um determinado componente.



(a)Ator do Sistema. **(b)Instância múltipla de um IP.** **(c)Instância de um IP.**

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
SF	Fluxo Secundário
FR	Requisito Funcional
OLA	Operação Lógica Aritmética
DM	Memória de Dados

2. Atores do Sistema

UC – Unidade que controla a execução das operações.

ULA – Unidade que realiza as operações lógicas e aritméticas.

3. Casos de Usos

3.1. [UC 001] Unidade de Controle

A **Unidade de Controle** é responsável por definir, a partir de uma instrução decodificada, o comportamento de outros módulos.

Atores

UC – Unidade que controla a execução das operações.

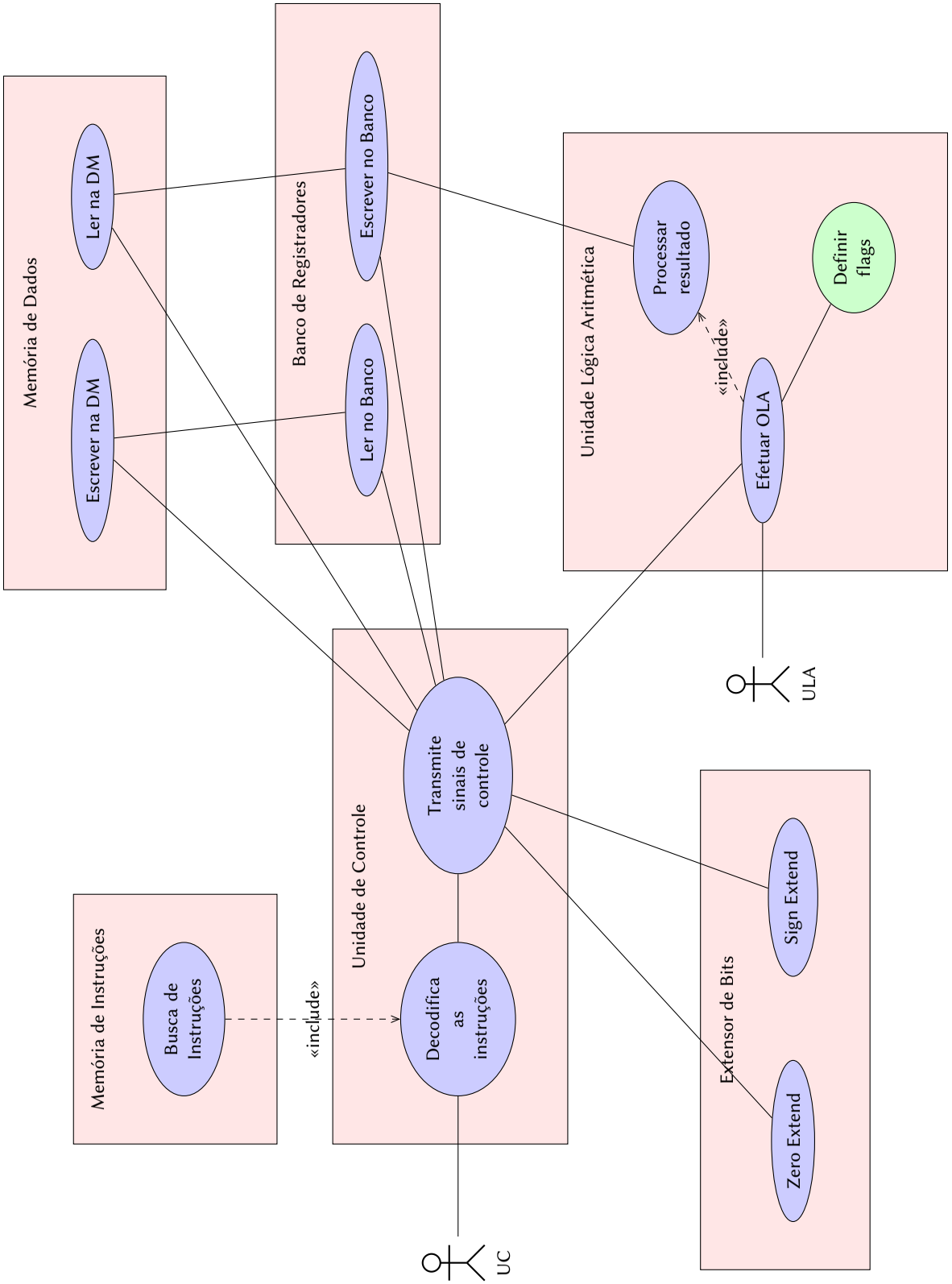
Pré-condições

- Atender ao requisito funcional [FR23];
- Receber opcode da instrução.

Pós-condições

- Ativar sinais de controle.

Diagrama de Caso de Uso



3.1.1. *Fluxo Principal de Eventos*

P1. Decodificar a instrução;

3.1.2. *Fluxo Secundário: Alternativo*

[SF1] Ativar sinais de controle para instruções do tipo R.

[SF2] Ativar sinais de controle para instruções do tipo I.

[SF3] Ativar sinais de controle para instruções do tipo J.

3.2. [UC 002] Unidade Lógica Aritmética

A **Unidade Lógica Aritmética** é responsável por realizar as operações aritméticas e lógicas, de acordo com o código da operação.

Atores

ULA – Unidade que realiza as operações lógicas e aritméticas.

Pré-condições

- Atender aos requisitos funcionais [FR1-19];
- Codificação das operações deve ser definida;

Pós-condições

- O módulo deve ser capaz de detectar overflow e underflow aritmético;
- O módulo deve ser capaz de ativar flags;
- O resultado deve estar armazenada no banco de registradores;

3.2.1. *Fluxo Principal de Eventos*

P1. Decodificar o indicador operação;

P2. Realizar operação aritmética ou lógica;

P3. Armazenar o resultado no banco de registradores;

3.2.2. *Fluxo Secundário: Alternativo*

[SF1] Valor do resultado excede o suportado

1. Habilitar sinal de overflow ou underflow;

[SF2] Operação que ativa flags

1. Habilitar um dos sinais de flags;

3.3. [UC 003] Memória de Dados

A **Mémoria de Dados** é responsável por armazenar os dados do programa.

Atores

UC – Unidade que controla a execução das operações.

Pré-condições

- Atender aos requisitos funcionais [FR25-26];
- O(s) dado(s) a serem armazenados devem estar no banco de registradores;

Pós-condições

- A memória não deve permitir sobrescrita dados quando não-autorizado;

3.3.1. *Fluxo Principal de Eventos*

P1. Recebecor o endereço de onde o dado será armazenado/lido;

P2. Realizar a leitura/escrita do dado;

3.4. [UC 004] Banco de Registradores

O **Banco de Registradores** é responsável por armazenar os dados oriundos da memória, diretamente da instrução recebida ou da ULA.

Atores

UC – Unidade que controla a execução das operações.

Pós-condições

- O banco não deve permitir sobrescrita dados quando não-autorizado;

3.4.1. *Fluxo Principal de Eventos*

P1. Realiza a leitura dos dados de acordo com o endereço contido na instrução;

P2. Realiza a escrita dos dados de acordo com o endereço contido na instrução;

3.5. [UC 005] Extensor de Bits

O **Extensor de Bit** é responsável por transformar um valor em outro de largura diferente.

Atores

UC – Unidade que controla a execução das operações.

Pré-condições

- Atender o requisito funcional [FR25];
- Receber dados de 16 bits;

Pós-condições

- O módulo deve ser capaz de fazer a réplica de sinal do bit mais significativo carregando-o nos bits restantes para dados com sinal;
- O módulo deve ser capaz de fazer preencher os bits restantes com zeros para dados sem sinal;

3.5.1. *Fluxo Principal de Eventos*

P1. Identificar se o dado de entrada a ser tratado possui sinal ou não;

P2. Realizar a extensão dos bits;

P3. Devolve o dado alterado;