



Plano de Testes

Core-MUSA

Universidade Estadual de Feira de Santana

Build 2.0a

Histórico de Revisões

Date	Descrição	Autor(s)
10/11/2014	Concepção do Documento	fmbboaventura
16/11/2014	<ul style="list-style-type: none">Plano de Teste da Memória de Instrução;	mtcastro

SUMÁRIO

1	Introdução	3
1.1	Objetivo	3
1.2	Visão Geral do Documento	3
1.3	Representação Simbólica [PARTE DO DOCUMENTO DE USE CASES!!] .	3
1.4	Definições, Acrônimos e Abreviações	4
2	Atores do Sistema [PARTE DO DOCUMENTO DE CASO DE USO!!]	4
3	Casos de Teste	4
3.1	[TC 001] ULA	4
3.1.1	Fluxo Principal de Eventos	5
3.1.2	Fluxo Secundário: Alternativo	5
3.2	[TC 002] Memória de Instrução	5
3.2.1	Fluxo Principal de Eventos	5

1. Introdução

1.1. Objetivo

Este documento tem como objetivo descrever os casos de teste dos componentes do processador de propósito geral MUSA.

1.2. Visão Geral do Documento

- Sessão 2: REVER.
- Sessão 3: relata a lista dos casos de teste do projeto.

1.3. Representação Simbólica [PARTE DO DOCUMENTO DE USE CASES!!]

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).

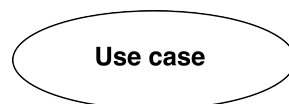


Figura 1: Exemplo de Caso de Uso.

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.

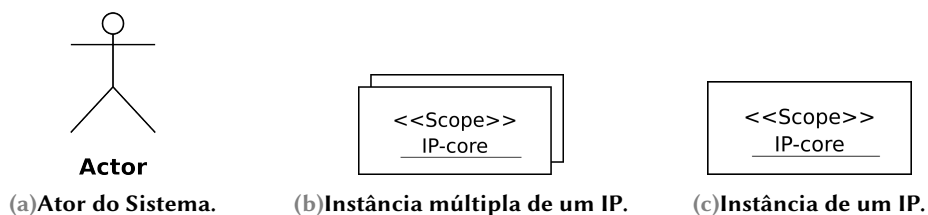


Figura 2: Simbologia utilizada na implementação dos Casos de Uso.

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
TC	Caso de Teste
SB	Sub-fluxo
FS	Fluxo Secundário
NFR	Requisito Não Funcional
FR	Requisito Funcional

2. Atores do Sistema [PARTE DO DOCUMENTO DE CASO DE USO!!]

3. Casos de Teste

Esta sessão apresenta o conjunto de TC realizados para a implementação dos testes do projeto Core-MUSA. As sessões a seguir foram divididas e nomeadas utilizando a nomenclatura abreviada [TC (NÚMERO DO TC)] seguido de uma breve descrição em forma de título.

3.1. [TC 001] ULA

A ULA tem como objetivo principal realizar operações lógicas e aritméticas, onde algumas delas estão ligadas diretamente a flags informativas ou de erros.

Entradas

- Operando 1;
- Operando 2;
- Sinal de identificação da operação;

Ações

- Realizar a operação solicitada;
- Ativar os sinais de saída de dados e de flags, caso ocorram;

Resultados Esperados

- Valor de 32 bits relativos ao resultado da operação;
- Sinal de flag, caso ocorram;

3.1.1. Fluxo Principal de Eventos

- P1. As funcionalidades serão testadas na seguinte ordem de acordo com o sinal de identificação: ADD, ADDI, SUB, SUBI, AND, ANDI, OR, ORI, MUL, DIV, CMP, NOT;
- P2. Os valores utilizados para os operandos serão adquiridos de forma aleatória;
- P3. Cada funcionalidade será testada 100 vezes;
- P4. Serão testados os seguintes casos de flags auxiliares: Equals ,Above;
- P5. Serão testados os seguintes casos de flag de erro de forma proposital: Overflow;
- P6. As flags serão testadas com a utilização de uma saída de controle temporária para os sinais;

3.1.2. Fluxo Secundário: Alternativo

[SF1] Título do Fluxo Secundário

1. Liste aqui as etapas do fluxo secundário;

[SF2] Título do Fluxo Secundário

1. Liste aqui as etapas do fluxo secundário;

3.2. [TC 002] Memória de Instrução

O objetivo deste teste é garantir que os registradores responsáveis por armazenar as instruções do programa estejam lendo as informações corretas na posição correta.

Entradas

- Endereço de Instrução;

Ações

- Busca um instrução na posição informada no endereço;

Resultados Esperados

- Confirmação da veracidade dos dados informado na saída;

3.2.1. Fluxo Principal de Eventos

- P1. A memória de instrução recebe um endereço, a instrução Relacionada é buscada e passada adiante;
- P2. A entrada será um código de endereço aleatório de 18 bits;
- P3. Será testada 20 vezes;

- P4.** A condição de testes é que o valor contido nos registradores de instrução seja igual ao valor associado ao endereço de instrução do programa de teste;
- P5.** Critério de aceitação é acertos igual ou maior 99% do total de casos de teste;
- P6.** As flags serão testadas com a utilização de uma saída de controle temporária para os sinais;