



Documento de Arquitetura

Core-MUSA

Universidade Estadual de Feira de Santana

Build 2.0a

Histórico de Revisões

Date	Descrição	Autor(s)
20/10/2014	Concepção do Documento	fmbboaventura
23/10/2014	Revisão Inicial	jadsonfirmo
29/10/2014	Adcionada Breve Descrição dos Componentes	fmbboaventura
29/10/2014	Stakeholders	jadsonfirmo
30/10/2014	Ajustes estruturais	fmbboaventura e jadsonfirmo

SUMÁRIO

1	Introdução	3
1	Propósito do Documento	3
2	Stakeholders	3
3	Visão Geral do Documento	3
4	Definições	3
5	Acrônimos e Abreviações	4
2	Visão Geral da Arquitetura	5
1	Descrição dos Componentes	5
2	Intruções	5
3	Detalhamento das Intruções	7
3	Descrição da Arquitetura	8
1	Nome do Módulo	8
1.1	Diagrama de Classe	8
1.2	Definições de Entrada e Saída	8
1.3	Datapath Interno	8
1.4	Máquina de Estados	8
1.5	Temporização	8

1 | Introdução

1. Propósito do Documento

Este documento descreve a arquitetura do projeto Core-MUSA, incluindo especificações dos circuitos internos e máquinas de estados de cada componente. Ele também apresenta diagramas de classe, definições de entrada e saída e diagramas de temporização. O principal objetivo deste documento é definir as especificações do projeto Core-MUSA e prover uma visão geral completa do mesmo.

2. Stakeholders

Nome	Papel/Responsabilidades
Diego Leite e Lucas Moraes	Gerencia
Victor Figueiredo, Matheus Castro, Odivio Caio Santos e Kelvin Carmo	Desenvolvimento
Filipe Boaventura e Wagner Bittencourt	Implementação
Jadson Firmo	Análise e Refatoração

3. Visão Geral do Documento

O presente documento é apresentado como segue:

- **Capítulo 2** – Este capítulo apresenta uma visão geral da arquitetura, com foco em entrada e saída do sistema e arquitetura geral do mesmo.
- **Capítulo 3** – Este capítulo apresenta a descrição detalhada da arquitetura bem como seus módulos e componentes.

4. Definições

Termo	Descrição

5. Acrônimos e Abreviações

Sigla	Descrição
PC	Contador de Programa (Program Counter)
ULA	Unidade Lógica e Aritmética
OPCODE	Código da Operação

2 | Visão Geral da Arquitetura

1. Descrição dos Componentes

A unidade de processamento a ser desenvolvida é composta a partir dos seguintes componentes:

- **PC** – Registrador que guarda o endereço da próxima instrução a ser executada.
- **Memória de Dados** – A Memória de dados é endereçada com 33 bits, que comporta no máximo 2^{33} palavras de instrução no total, que guarda dados com tamanho de 32 bits que foram manipulados pelo programa ou processador.
- **Memória de Instrução** – A Memória de instrução é endereçada com 18 bits, que comporta no máximo 2^{18} palavras de instrução no total, tem tamanho de 1048576 bytes, justamente pelo fato da palavra de instrução ter 32 bits. Por fim é a memória que guarda o programa codificado em linguagem assembly.
- **ULA** – É responsável por todo o processamento realizado no processador, pois esta unidade executa as instruções lógicas e aritméticas.
- **Unidade de Controle** – Esta unidade decodifica a instrução e define sinais de controle como, sinais de leitura, escrita de memória e de registradores de armazenamento temporário interno e sinais de liberação de barramentos para endereço e dados e unidades funcionais. As unidades funcionais internas do processador são controladas por esta unidade de temporização e controle. Os determinados sinais de controle são enviados para as demais unidades após a decodificação de uma determinada instrução que partem do registrador de instrução (IR).
- **Banco de Registradores** – Contém os 32 registradores de propósito geral do processador.
- **Pilha** – Memória destinada para armazenamento dos endereços de retorno de chamadas de funções. (Detalhar mais, qual tamanho dos endereçamento e tal ..)

2. Instruções

A unidade de processamento possui 21 instruções essenciais para o processamento das operações. Elas são desmembradas em quatro formatos: R-type, I-type, Load/Store e Jump.

- **R-type** – Operações lógicas e aritméticas.

- **ADD:** Soma de dois valores.
- **SUB:** Subtração de dois valores.
- **MUL:** Multiplicação de dois valores.
- **DIV:** Divisão de dois valores.
- **AND:** Operação lógica AND entre dois valores.
- **OR:** Operação lógica OR entre dois valores.
- **NOT:** Operação lógica NOT.
- **CMP:** Comparação de dois valores.
- **I-type** – Operações imediatas.
 - **ADDi:** Soma de dois valores, sendo um destes imediato.
 - **SUBi:** Subtração de dois valores, sendo um destes imediato.
 - **ANDi:** Operação lógica AND entre dois valores, sendo um destes imediato.
 - **ORi:** Operação lógica OR entre dois valores, sendo um destes imediato.
- **Load/Store** – Operações de carregamento e armazenamento.
 - **LW:** Operação de leitura na memória de dados.
 - **SW:** Operação de armazenamento na memória de dados.
- **Jump** – Operações de desvio.
 - **JR:** Desvia o programa para um endereço de destino.
 - **JPC:** Desvia o programa para um endereço relativo ao PC.
 - **BRFL:** Desvia o programa para um endereço de destino, atendendo uma condição de flag.
 - **CALL:** Desvia um programa em execução para uma sub-rotina.
 - **RET:** Retorna de uma sub-rotina.
 - **HALT:** Para a execução de um programa.
 - **NOP:** Não realiza operação.

3. Detalhamento das Instruções

- ADD, SUB, MUL, DIV, AND, OR, NOT, CMP, ADDi, SUBi, ANDi e ORi:

OPCODE	FUNCTION	RD	RS	RT	IM
04	04	05	05	05	09

Tabela 2.1: Layout das Operações Aritméticas, Lógicas e Imediatas

- LW e SW:

OPCODE	RD	RS	I	
04	04	05	16	02

Tabela 2.2: Layout das Operações de Leitura e Escrita

- JR, BRFL, CALL, RET, HALT e NOP:

OPCODE	RF	CST	
04	05	05	18

Tabela 2.3: Layout das Operações de Salto

- JPC:

OPCODE	I
04	28

Tabela 2.4: Layout da Operação JPC

3 | Descrição da Arquitetura

1. Nome do Módulo

1.1. Diagrama de Classe

1.2. Definições de Entrada e Saída

1.3. Datapath Interno

1.4. Máquina de Estados

1.5. Temporização