

MIPS16bits

Generated by Doxygen 1.8.11

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

MIPS::ALUFlags	??
MIPS::ControlUnit	??
MIPS::CPU	??
MIPS::Encoder	??
MIPS::FormatIEncoder	??
MIPS::FormatIIEncoder	??
MIPS::FormatIIIEncoder	??
MIPS::FormatIVEncoder	??
MIPS::FormatVEncoder	??
MIPS::FormatVIEncoder	??
MIPS::FormatVIIEncoder	??
MIPS::EncoderFactory	??
MIPS::Event	??
MIPS::EventDispatcher	??
MIPS::EventListener	??
MIPS::FileReader	??
MIPS::Filter	??
MIPS::SpaceFilter	??
MIPS::FullAdder	??
MIPS::Instruction	??
MIPS::InstructionI	??
MIPS::AddIncInstruction	??
MIPS::AddInstruction	??
MIPS::AndInstruction	??
MIPS::AndnotInstruction	??
MIPS::AslInstruction	??
MIPS::AsrInstruction	??
MIPS::DecalInstruction	??
MIPS::IncInstruction	??
MIPS::NandInstruction	??
MIPS::NorInstruction	??
MIPS::OnesInstruction	??
MIPS::OrInstruction	??
MIPS::OrnotbInstruction	??

MIPS::PassaInstruction	??
MIPS::PassNotAInstruction	??
MIPS::SubdeclInstruction	??
MIPS::SubInstruction	??
MIPS::XnorInstruction	??
MIPS::XorInstruction	??
MIPS::ZeroInstruction	??
MIPS::InstructionII	??
MIPS::LoadlitInstruction	??
MIPS::InstructionIII	??
MIPS::LchInstruction	??
MIPS::LclInstruction	??
MIPS::InstructionIV	??
MIPS::JtZeroInstruction	??
MIPS::InstructionV	??
MIPS::InstructionVI	??
MIPS::InstructionVII	??
MIPS::InstructionDecoder	??
MIPS::InstructionFinder	??
MIPS::Interpreter	??
MIPS::Label	??
MIPS::EventDispatcher::ListenerMap	??
MIPS::Memory	??
MIPS::Queue< T >::Node	??
MIPS::Queue< T >	??
MIPS::Queue< MIPS::EventListener * >	??
MIPS::Register	??
MIPS::RegisterBank	??
runtime_error	
MIPS::InterpreterException	??
MIPS::MemoryException	??
MIPS::SignalExtender	??
MIPS::SignalInversor	??
MIPS::Tokenizer	??

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

MIPS::AddIncInstruction	??
MIPS::AddInstruction	??
MIPS::ALUFlags	??
MIPS::AndInstruction	??
MIPS::AndnotInstruction	??
MIPS::AslInstruction	??
MIPS::AsrInstruction	??
MIPS::ControlUnit	??
MIPS::CPU	??
MIPS::DecalInstruction	??
MIPS::Encoder	??
MIPS::EncoderFactory	??
MIPS::Event	??
MIPS::EventDispatcher	??
MIPS::EventListener	??
MIPS::FileReader	??
MIPS::Filter	??
MIPS::FormatIEncoder	??
MIPS::FormatIIEncoder	??
MIPS::FormatIIIEncoder	??
MIPS::FormatIVEncoder	??
MIPS::FormatVEncoder	??
MIPS::FormatVIEncoder	??
MIPS::FormatVIIEncoder	??
MIPS::FullAdder	??
MIPS::IncInstruction	??
MIPS::Instruction	??
MIPS::InstructionDecoder	??
MIPS::InstructionFinder	??
MIPS::InstructionI	??
MIPS::InstructionII	??
MIPS::InstructionIII	??
MIPS::InstructionIV	??
MIPS::InstructionV	??
MIPS::InstructionVI	??

MIPS::InstructionVII	??
MIPS::Interpreter	??
MIPS::InterpreterException	??
MIPS::JtZeroInstruction	??
MIPS::Label	??
MIPS::LchInstruction	??
MIPS::LclInstruction	??
MIPS::EventDispatcher::ListenerMap	??
MIPS::LoadlitInstruction	??
MIPS::Memory	??
MIPS::MemoryException	??
MIPS::NandInstruction	??
MIPS::Queue< T >::Node	??
MIPS::NorInstruction	??
MIPS::OnesInstruction	??
MIPS::OrInstruction	??
MIPS::OrnotbInstruction	??
MIPS::PassaInstruction	??
MIPS::PassNotAInstruction	??
MIPS::Queue< T >	??
MIPS::Register	??
MIPS::RegisterBank	??
MIPS::SignalExtender	??
MIPS::SignalInversor	??
MIPS::SpaceFilter	??
MIPS::SubdecInstruction	??
MIPS::SubInstruction	??
MIPS::Tokenizer	??
MIPS::XnorInstruction	??
MIPS::XorInstruction	??
MIPS::ZeroInstruction	??

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

include/mips/core.hpp	??
include/mips/cpu.hpp	??
include/mips/debug.hpp	??
include/mips/flag.hpp	??
include/mips/circuits/full_adder.hpp	??
include/mips/circuits/signal_extender.hpp	??
include/mips/circuits/signal_inversor.hpp	??
include/mips/decoder/instruction_decoder.hpp	??
include/mips/instructions/instruction.hpp	??
include/mips/instructions/instruction_I.hpp	??
include/mips/instructions/instruction_II.hpp	??
include/mips/instructions/instruction_III.hpp	??
include/mips/instructions/instruction_IV.hpp	??
include/mips/instructions/instruction_V.hpp	??
include/mips/instructions/instruction_VI.hpp	??
include/mips/instructions/instruction_VII.hpp	??
include/mips/instructions/format_I/add.hpp	??
include/mips/instructions/format_I/addinc.hpp	??
include/mips/instructions/format_I/and.hpp	??
include/mips/instructions/format_I/andnota.hpp	??
include/mips/instructions/format_I/asl.hpp	??
include/mips/instructions/format_I/asr.hpp	??
include/mips/instructions/format_I/deca.hpp	??
include/mips/instructions/format_I/inca.hpp	??
include/mips/instructions/format_I/nand.hpp	??
include/mips/instructions/format_I/nor.hpp	??
include/mips/instructions/format_I/ones.hpp	??
include/mips/instructions/format_I/or.hpp	??
include/mips/instructions/format_I/ornotb.hpp	??
include/mips/instructions/format_I/passa.hpp	??
include/mips/instructions/format_I/passnota.hpp	??
include/mips/instructions/format_I/sub.hpp	??
include/mips/instructions/format_I/subdec.hpp	??
include/mips/instructions/format_I/xnor.hpp	??
include/mips/instructions/format_I/xor.hpp	??

include/mips/instructions/format_I/zero.hpp	??
include/mips/instructions/format_II/loadlit.hpp	??
include/mips/instructions/format_III/lch.hpp	??
include/mips/instructions/format_III/lcl.hpp	??
include/mips/instructions/format_IV/jt_zero.hpp	??
include/mips/interpreter/interpreter.hpp	??
include/mips/interpreter/label.hpp	??
include/mips/interpreter/encoder/encoder.hpp	??
include/mips/interpreter/encoder/encoder_factory.hpp	??
include/mips/interpreter/encoder/format_I_encoder.hpp	??
include/mips/interpreter/encoder/format_II_encoder.hpp	??
include/mips/interpreter/encoder/format_III_encoder.hpp	??
include/mips/interpreter/encoder/format_IV_encoder.hpp	??
include/mips/interpreter/encoder/format_V_encoder.hpp	??
include/mips/interpreter/encoder/format_VI_encoder.hpp	??
include/mips/interpreter/encoder/format_VII_encoder.hpp	??
include/mips/interpreter/exception/interpreter_exception.hpp	??
include/mips/interpreter/parser/tokenizer.hpp	??
include/mips/memory/memory.hpp	??
include/mips/memory/memory_exception.hpp	??
include/mips/memory/register.hpp	??
include/mips/memory/register_bank.hpp	??
include/mips/units/control.hpp	??
include/mips/units/instruction_finder.hpp	??
include/mips/util/file_reader.hpp	??
include/mips/util/event/event.hpp	??
include/mips/util/event/event_dispatcher.hpp	??
include/mips/util/event/event_listener.hpp	??
include/mips/util/filter/filter.hpp	??
include/mips/util/filter/space_filter.hpp	??
include/mips/util/structure/queue.hpp	??

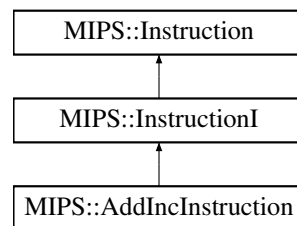
Chapter 4

Class Documentation

4.1 MIPS::AddIncInstruction Class Reference

```
#include <addinc.hpp>
```

Inheritance diagram for MIPS::AddIncInstruction:



Public Member Functions

- [AddIncInstruction](#) ([bit8_t opcode](#), [Register *rs](#), [Register *rt](#), [bit8_t shamt](#), [bit8_t funct](#))
- [bit16_t execute](#) ()

Additional Inherited Members

4.1.1 Detailed Description

Classe que faz a operação de ADDINC no processador.

Author

Felipe Dias

4.1.2 Constructor & Destructor Documentation

4.1.2.1 MIPS::AddIncInstruction::AddIncInstruction ([bit8_t opcode](#), [Register * rs](#), [Register * rt](#), [bit8_t shamt](#), [bit8_t funct](#)) [\[inline\]](#)

Constroi uma nova instrução.

4.1.3 Member Function Documentation

4.1.3.1 `bit16_t MIPS::AddIncInstruction::execute () [virtual]`

Função que executa a operação de soma.

Returns

resultado da operação

Implements [MIPS::InstructionI](#).

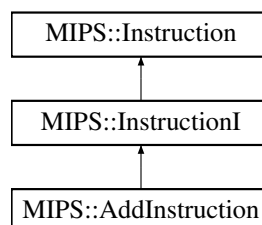
The documentation for this class was generated from the following file:

- [include/mips/instructions/format_l/addinc.hpp](#)

4.2 MIPS::AddInstruction Class Reference

```
#include <add.hpp>
```

Inheritance diagram for MIPS::AddInstruction:



Public Member Functions

- [AddInstruction](#) ([bit8_t opcode](#), [Register *rs](#), [Register *rt](#), [bit8_t shamt](#), [bit8_t funct](#))
- [bit16_t execute](#) ()

Additional Inherited Members

4.2.1 Detailed Description

Classe que faz a operação de ADD no processador.

Author

Felipe Dias

4.2.2 Constructor & Destructor Documentation

4.2.2.1 MIPS::AddInstruction::AddInstruction (bit8_t opcode, Register * rs, Register * rt, bit8_t shamt, bit8_t funct)
[inline]

Constroi uma nova instrução.

4.2.3 Member Function Documentation

4.2.3.1 bit16_t MIPS::AddInstruction::execute () [virtual]

Função que executa a operação de soma.

Returns

resultado da operação

Implements [MIPS::InstructionI](#).

The documentation for this class was generated from the following file:

- [include/mips/instructions/format_l/add.hpp](#)

4.3 MIPS::ALUFlags Struct Reference

```
#include <flag.hpp>
```

Public Attributes

- [bit8_t zero](#)
Flag de zero.
- [bit8_t carry](#)
Flag de carry.
- [bit8_t neg](#)
Flag de negativo.
- [bit8_t overflow](#)
Flag de overflow.

4.3.1 Detailed Description

Estrutura que armazena as informações das flags de saída da ALU.

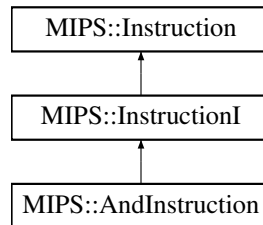
The documentation for this struct was generated from the following file:

- [include/mips/flag.hpp](#)

4.4 MIPS::AndInstruction Class Reference

```
#include <and.hpp>
```

Inheritance diagram for MIPS::AndInstruction:



Public Member Functions

- [AndInstruction](#) ([bit8_t opcode](#), [Register *rs](#), [Register *rt](#), [bit8_t shamt](#), [bit8_t funct](#))
- [bit16_t execute](#) ()

Additional Inherited Members

4.4.1 Detailed Description

Classe que faz a operação de and no processador.

Author

Lucas Fonseca dos Santos

4.4.2 Constructor & Destructor Documentation

4.4.2.1 `MIPS::AndInstruction::AndInstruction (bit8_t opcode, Register * rs, Register * rt, bit8_t shamt, bit8_t funct)`
`[inline]`

Constroi uma nova instrução.

4.4.3 Member Function Documentation

4.4.3.1 `bit16_t MIPS::AndInstruction::execute ()` `[virtual]`

Função que executa a operação de [AndInstruction](#).

Implements [MIPS::InstructionI](#).

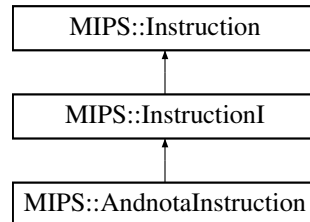
The documentation for this class was generated from the following file:

- `include/mips/instructions/format_l/and.hpp`

4.5 MIPS::AndnotaInstruction Class Reference

```
#include <andnota.hpp>
```

Inheritance diagram for MIPS::AndnotaInstruction:



Public Member Functions

- [AndnotaInstruction](#) ([bit8_t opcode](#), [Register *rs](#), [Register *rt](#), [bit8_t shamt](#), [bit8_t funct](#))
- [bit16_t execute](#) ()

Additional Inherited Members

4.5.1 Detailed Description

Classe que faz a operação de andnota no processador.

Author

Lucas Pereira

4.5.2 Constructor & Destructor Documentation

4.5.2.1 MIPS::AndnotaInstruction::AndnotaInstruction ([bit8_t opcode](#), [Register * rs](#), [Register * rt](#), [bit8_t shamt](#), [bit8_t funct](#)) [\[inline\]](#)

Constroi uma nova instrução.

4.5.3 Member Function Documentation

4.5.3.1 [bit16_t](#) MIPS::AndnotaInstruction::execute () [\[virtual\]](#)

Função que executa a operação de [AndnotaInstruction](#).

Returns

resultado da operação

Implements [MIPS::InstructionI](#).

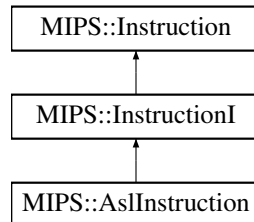
The documentation for this class was generated from the following file:

- `include/mips/instructions/format_1/andnota.hpp`

4.6 MIPS::AslInstruction Class Reference

```
#include <asl.hpp>
```

Inheritance diagram for MIPS::AslInstruction:



Public Member Functions

- [AslInstruction](#) ([bit8_t opcode](#), [Register *rs](#), [Register *rt](#), [bit8_t shamt](#), [bit8_t funct](#))
- [bit16_t execute](#) ()

Additional Inherited Members

4.6.1 Detailed Description

Classe que faz a operação de ASR no processador.

Author

Felipe Dias

4.6.2 Constructor & Destructor Documentation

4.6.2.1 `MIPS::AslInstruction::AslInstruction (bit8_t opcode, Register * rs, Register * rt, bit8_t shamt, bit8_t funct)`
`[inline]`

Constroi uma nova instrução.

4.6.3 Member Function Documentation

4.6.3.1 `bit16_t MIPS::AslInstruction::execute ()` `[virtual]`

Função que executa a operação de shift a esquerda.

Returns

resultado da operação

Implements [MIPS::InstructionI](#).

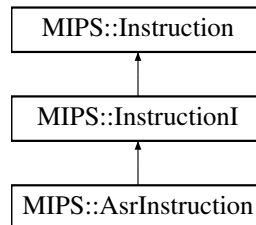
The documentation for this class was generated from the following file:

- `include/mips/instructions/format_l/asl.hpp`

4.7 MIPS::AsrInstruction Class Reference

```
#include <asr.hpp>
```

Inheritance diagram for MIPS::AsrInstruction:



Public Member Functions

- [AsrInstruction](#) ([bit8_t opcode](#), [Register *rs](#), [Register *rt](#), [bit8_t shamt](#), [bit8_t funct](#))
- [bit16_t execute](#) ()

Additional Inherited Members

4.7.1 Detailed Description

Classe que faz a operação de ASR no processador.

Author

Felipe Dias

4.7.2 Constructor & Destructor Documentation

4.7.2.1 `MIPS::AsrInstruction::AsrInstruction (bit8_t opcode, Register * rs, Register * rt, bit8_t shamt, bit8_t funct)`
`[inline]`

Constroi uma nova instrução.

4.7.3 Member Function Documentation

4.7.3.1 `bit16_t MIPS::AsrInstruction::execute ()` `[virtual]`

Função que executa a operação de shift a direita.

Returns

resultado da operação

Implements [MIPS::InstructionI](#).

The documentation for this class was generated from the following file:

- `include/mips/instructions/format_l/asr.hpp`

4.8 MIPS::ControlUnit Class Reference

```
#include <control.hpp>
```

Public Member Functions

- [ControlUnit](#) ()
- [~ControlUnit](#) ()

Public Attributes

- bool [regwrite](#)
- bool [branch](#)
- bool [jump](#)

4.8.1 Detailed Description

Unidade de controle do processador. Essa unidade é responsável por setar todas as flags que serão utilizadas para executar uma instrução.

Author

Matheus Nogueira

4.8.2 Constructor & Destructor Documentation

4.8.2.1 MIPS::ControlUnit::ControlUnit ()

Cria uma nova unidade de controle.

4.8.2.2 MIPS::ControlUnit::~~ControlUnit ()

Destroi a unidade de controle.

4.8.3 Member Data Documentation

4.8.3.1 bool MIPS::ControlUnit::branch

Flag de branch.

4.8.3.2 bool MIPS::ControlUnit::jump

Flag de jump

4.8.3.3 bool MIPS::ControlUnit::regwrite

Flag de escrita em banco de registradores.

The documentation for this class was generated from the following file:

- include/mips/units/[control.hpp](#)

4.9 MIPS::CPU Class Reference

```
#include <cpu.hpp>
```

Public Member Functions

- [CPU](#) ()
- [~CPU](#) ()
- void [loadProgram](#) (const char *program)
- void [execute](#) ()

4.9.1 Detailed Description

Classe que representa o processador m-RISC, esta é responsável por gerenciar toda a execução das instruções no processador.

Author

Matheus Nogueira

4.9.2 Constructor & Destructor Documentation

4.9.2.1 MIPS::CPU::CPU ()

Cria uma nova [CPU](#).

4.9.2.2 MIPS::CPU::~~CPU ()

Destroi a [CPU](#).

4.9.3 Member Function Documentation

4.9.3.1 void MIPS::CPU::execute ()

Executa as instruções carregadas previamente pelo método loadProgram.

4.9.3.2 void MIPS::CPU::loadProgram (const char * *program*)

Carrega um programa na memória de instruções do processador.

Parameters

<i>program</i>	caminho para o arquivo contendo o programa.
----------------	---------------------------------------------

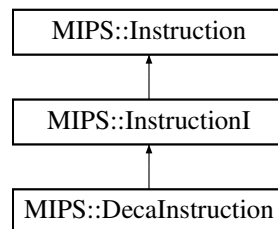
The documentation for this class was generated from the following file:

- [include/mips/cpu.hpp](#)

4.10 MIPS::DecalInstruction Class Reference

```
#include <deca.hpp>
```

Inheritance diagram for MIPS::DecalInstruction:

**Public Member Functions**

- [DecalInstruction](#) ([bit8_t opcode](#), [Register *rs](#), [Register *rt](#), [bit8_t shamt](#), [bit8_t funct](#))
- [bit16_t execute](#) ()

Additional Inherited Members

4.10.1 Detailed Description

Classe que faz a operação de DECA no processador.

Author

Felipe Dias

4.10.2 Constructor & Destructor Documentation

4.10.2.1 MIPS::DecalInstruction::DecalInstruction ([bit8_t opcode](#), [Register * rs](#), [Register * rt](#), [bit8_t shamt](#), [bit8_t funct](#)) [\[inline\]](#)

Constroi uma nova instrução.

4.10.3 Member Function Documentation

4.10.3.1 `bit16_t MIPS::DecalInstruction::execute ()` [virtual]

Função que executa a operação de decremento.

Returns

resultado da operação

Implements [MIPS::InstructionI](#).

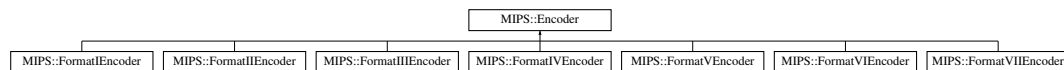
The documentation for this class was generated from the following file:

- `include/mips/instructions/format_l/deca.hpp`

4.11 MIPS::Encoder Class Reference

```
#include <encoder.hpp>
```

Inheritance diagram for MIPS::Encoder:



Public Member Functions

- [Encoder \(\)](#)
- virtual [~Encoder \(\)](#)
- virtual [instruction_t encode \(\)=0](#)
- virtual void [parse](#) (std::vector< char * > ¶ms)=0

Protected Member Functions

- [bit8_t getRegisterNumber](#) (const char *name)

Protected Attributes

- [bit8_t opcode](#)

4.11.1 Detailed Description

Classe abstrata que cria uma interface para todos os codificadores de instruções MIPS 32.

Author

Matheus Nogueira

4.11.2 Constructor & Destructor Documentation

4.11.2.1 MIPS::Encoder::Encoder ()

Cria um novo codificador.

4.11.2.2 virtual MIPS::Encoder::~~Encoder () [inline],[virtual]

Destroi o codificador.

4.11.3 Member Function Documentation

4.11.3.1 virtual instruction_t MIPS::Encoder::encode () [pure virtual]

Codifica a ultima instrução que foi analisada pelo parser do codificador.

Returns

instrução 16 bits.

Implemented in [MIPS::FormatIEncoder](#), [MIPS::FormatIIEncoder](#), [MIPS::FormatIIIEncoder](#), [MIPS::FormatIVEncoder](#), [MIPS::FormatVEncoder](#), [MIPS::FormatVIEncoder](#), and [MIPS::FormatVIIEncoder](#).

4.11.3.2 bit8_t MIPS::Encoder::getRegisterNumber (const char * *name*) [protected]

Retorna o número do registrador solicitado.

Parameters

<i>name</i>	nome do registrador.
-------------	----------------------

Returns

número do registrador

4.11.3.3 virtual void MIPS::Encoder::parse (std::vector< char *> & *params*) [pure virtual]

Percorre a instrução em assembly e extraí os dados dela para poder montar uma instrução binária.

Parameters

<i>params</i>	parametros da instrução.
---------------	--------------------------

Implemented in [MIPS::FormatIEncoder](#), [MIPS::FormatIIEncoder](#), [MIPS::FormatIIIEncoder](#), [MIPS::FormatIVEncoder](#), [MIPS::FormatVEncoder](#), [MIPS::FormatVIEncoder](#), and [MIPS::FormatVIIEncoder](#).

[Encoder](#), [MIPS::FormatVEncoder](#), [MIPS::FormatVIEncoder](#), and [MIPS::FormatVIEncoder](#).

4.11.4 Member Data Documentation

4.11.4.1 `bit8_t MIPS::Encoder::opcode` `[protected]`

Opcode da instrução.

The documentation for this class was generated from the following file:

- `include/mips/interpreter/encoder/encoder.hpp`

4.12 MIPS::EncoderFactory Class Reference

```
#include <encoder_factory.hpp>
```

Public Member Functions

- [EncoderFactory](#) ()
- [Encoder](#) * [produce](#) (const char *instruction)

4.12.1 Detailed Description

Fábrica responsável por referenciar os codificadores corretos para cada instrução da linguagem de montagem do micro-RISC.

Author

Matheus Nogueira

4.12.2 Constructor & Destructor Documentation

4.12.2.1 `MIPS::EncoderFactory::EncoderFactory ()` `[inline]`

Cria uma nova fábrica de codificadores.

4.12.3 Member Function Documentation

4.12.3.1 `Encoder*` `MIPS::EncoderFactory::produce (const char * instruction)`

Produz um codificador para determinada instrução.

Parameters

<i>instruction</i>	nome da instrução que deve ser codificada.
--------------------	--------------------------------------------

The documentation for this class was generated from the following file:

- `include/mips/interpreter/encoder/encoder_factory.hpp`

4.13 MIPS::Event Class Reference

```
#include <event.hpp>
```

Public Member Functions

- [Event](#) ([EventType](#) type, void *data, bool autodestroy=false)
- virtual [~Event](#) ()
- bool [shouldDestroy](#) ()

Public Attributes

- const [EventType](#) [event_type](#)
- void * [data_ptr](#)

4.13.1 Detailed Description

Classe responsável por representar um evento que pode ser despachado pelo despachante de eventos do emulador.

Author

Matheus Nogueira

4.13.2 Constructor & Destructor Documentation

4.13.2.1 MIPS::Event::Event ([EventType](#) type, void * data, bool autodestroy = false)

Cria um novo evento.

Parameters

<i>type</i>	tipo do evento
<i>data</i>	dados associados ao evento. Esses dados associados devem ser alocados de forma dinâmica pelo programador, utilizando a função malloc. O despachante de eventos é responsável por destruir esses dados após seu uso caso a flag autodestroy seja ativada.
<i>autodestroy</i>	indica que os dados do evento devem ser destruídos após que o evento seja despachado para todos seus ouvintes.

4.13.2.2 virtual MIPS::Event::~~Event () [virtual]

Destroi o evento e os dados relacionados ao mesmo.

4.13.3 Member Function Documentation

4.13.3.1 bool MIPS::Event::shouldDestroy ()

Verifica se o evento deve se auto destruir.

Returns

true se o evento deve se auto destruir.

4.13.4 Member Data Documentation

4.13.4.1 void* MIPS::Event::data_ptr

Ponteiro para os dados do evento.

4.13.4.2 const EventType MIPS::Event::event_type

Tipo de evento.

The documentation for this class was generated from the following file:

- include/mips/util/event/[event.hpp](#)

4.14 MIPS::EventDispatcher Class Reference

```
#include <event_dispatcher.hpp>
```

Classes

- struct [ListenerMap](#)

Public Member Functions

- [EventDispatcher](#) ()
- virtual [~EventDispatcher](#) ()
- void [dispatch](#) ([Event](#) &event)
- void [addEventListener](#) ([EventListener](#) *listener, [EventType](#) event)

4.14.1 Detailed Description

Classe responsável por permitir que componentes do emulador possam se comunicar por troca de mensagens transmitidas por eventos.

Author

Matheus Nogueira

4.14.2 Constructor & Destructor Documentation

4.14.2.1 MIPS::EventDispatcher::EventDispatcher ()

Cria um novo despachante de eventos.

4.14.2.2 virtual MIPS::EventDispatcher::~~EventDispatcher () [virtual]

Destroi o despachante de eventos.

4.14.3 Member Function Documentation

4.14.3.1 void MIPS::EventDispatcher::addEventListener (EventListener * *listener*, EventType *event*)

Adiciona um ouvinte de eventos nesse despachante.

Parameters

<i>listener</i>	ouvinte de eventos
<i>event</i>	tipo de evento que o ouvinte irá escutar.

4.14.3.2 void MIPS::EventDispatcher::dispatch (Event & *event*)

Despacha um evento para todos os seus ouvintes. Caso o evento esteja marcado com a flag autodestroy, o evento será destruído logo após a chamada desse método.

Parameters

<i>event</i>	evento a ser despachado.
--------------	--------------------------

The documentation for this class was generated from the following file:

- include/mips/util/event/[event_dispatcher.hpp](#)

4.15 MIPS::EventListener Class Reference

```
#include <event_listener.hpp>
```

Public Member Functions

- virtual void [notify](#) ([Event](#) &event)

4.15.1 Detailed Description

Classe abstrata que permite um objeto ouvir eventos vindos de outra classe.

Author

Matheus Nogueira

4.15.2 Member Function Documentation

4.15.2.1 virtual void MIPS::EventListener::notify ([Event](#) & *event*) [virtual]

Função utilizada para notificar o objeto que algum evento o qual ele está esperando, ocorreu.

Parameters

<i>event</i>	evento que ocorreu.
--------------	---------------------

The documentation for this class was generated from the following file:

- include/mips/util/event/[event_listener.hpp](#)

4.16 MIPS::FileReader Class Reference

```
#include <file_reader.hpp>
```

Public Member Functions

- [FileReader](#) (const char *filename)
- [FileReader](#) (const char *filename, [Filter](#) &filter)
- [~FileReader](#) ()
- char * [next](#) ()
- void [rewind](#) ()
- bool [hasNext](#) ()

4.16.1 Detailed Description

Classe responsável por ler o conteúdo de um arquivo e salvá-lo internamente, fazendo com que seja simples para obter as informações do mesmo.

Author

Matheus Nogueira

4.16.2 Constructor & Destructor Documentation

4.16.2.1 MIPS::FileReader::FileReader (const char * *filename*)

Cria um novo leitor de arquivo.

Parameters

<i>filename</i>	nome do arquivo a ser lido.
-----------------	-----------------------------

4.16.2.2 MIPS::FileReader::FileReader (const char * *filename*, Filter & *filter*)

Cria um novo leitor de arquivo que filtra determinadas entradas.

Parameters

<i>filename</i>	nome do arquivo a ser lido.
<i>filter</i>	filtro a ser utilizado.

4.16.2.3 MIPS::FileReader::~~FileReader ()

Destroi o leitor de arquivo e o conteúdo do buffer interno.

4.16.3 Member Function Documentation

4.16.3.1 bool MIPS::FileReader::hasNext ()

Checa se o arquivo ainda tem conteúdo para ser lido.

Returns

booleano que se true, indica que ainda existe linhas a ser lidas.

4.16.3.2 char* MIPS::FileReader::next ()

Retorna a proxima linha do arquivo.

Returns

proxima linha do arquivo.

4.16.3.3 void MIPS::FileReader::rewind ()

Retorna o ponteiro da linha atual para o inicio do arquivo.

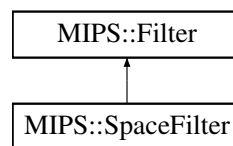
The documentation for this class was generated from the following file:

- [include/mips/util/file_reader.hpp](#)

4.17 MIPS::Filter Class Reference

```
#include <filter.hpp>
```

Inheritance diagram for MIPS::Filter:



Public Member Functions

- virtual std::string [filter](#) (std::string &input)=0

4.17.1 Detailed Description

Classe abstrata que representa um filtro de texto.

Author

Matheus Nogueira

4.17.2 Member Function Documentation

4.17.2.1 virtual std::string MIPS::Filter::filter (std::string & input) [pure virtual]

Filtra uma string e retorna uma nova instância da mesma.

Parameters

<i>input</i>	string a ser filtrada.
--------------	------------------------

Returns

string filtrada.

Implemented in [MIPS::SpaceFilter](#).

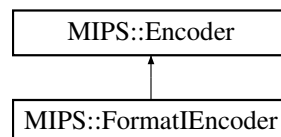
The documentation for this class was generated from the following file:

- [include/mips/util/filter/filter.hpp](#)

4.18 MIPS::FormatLEncoder Class Reference

```
#include <format_I_encoder.hpp>
```

Inheritance diagram for MIPS::FormatLEncoder:

**Public Member Functions**

- [instruction_t encode](#) ()
- void [parse](#) (std::vector< char * > ¶ms)

Additional Inherited Members

4.18.1 Detailed Description

Codificador responsável por transformar uma instrução assembly em uma instrução binária, seguindo o formato de instrução no formato I.

Author

Matheus Nogueira

4.18.2 Member Function Documentation

4.18.2.1 instruction_t MIPS::FormatLEncoder::encode () [virtual]

Codifica uma instrução assembly do formato I para uma instrução binária.

Returns

instrução binária de 16 bits

Implements [MIPS::Encoder](#).

4.18.2.2 void MIPS::FormatLEncoder::parse (std::vector< char * > ¶ms) [virtual]

Realiza uma varredura na instrução assembly e define seus campos binários.

Parameters

<i>params</i>	vector de parâmetros da instrução
---------------	-----------------------------------

Implements [MIPS::Encoder](#).

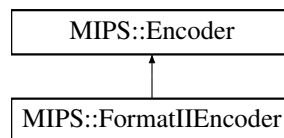
The documentation for this class was generated from the following file:

- [include/mips/interpreter/encoder/format_I_encoder.hpp](#)

4.19 MIPS::FormatIIEncoder Class Reference

```
#include <format_II_encoder.hpp>
```

Inheritance diagram for MIPS::FormatIIEncoder:



Public Member Functions

- [instruction_t encode](#) ()
- void [parse](#) (std::vector< char * > ¶ms)

Additional Inherited Members

4.19.1 Detailed Description

Codificador responsável por transformar uma instrução assembly em uma instrução binária, seguindo o formato de instrução no formato II.

Author

Matheus Nogueira

4.19.2 Member Function Documentation

4.19.2.1 instruction_t MIPS::FormatIIEncoder::encode () [virtual]

Codifica uma instrução assembly do formato II para uma instrução binária.

Returns

instrução binária de 16 bits

Implements [MIPS::Encoder](#).

4.19.2.2 void MIPS::FormatIIEncoder::parse (std::vector< char * > ¶ms) [virtual]

Realiza uma varredura na instrução assembly e define seus campos binários.

Parameters

<i>params</i>	vector de parâmetros da instrução
---------------	-----------------------------------

Implements [MIPS::Encoder](#).

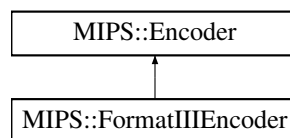
The documentation for this class was generated from the following file:

- [include/mips/interpreter/encoder/format_II_encoder.hpp](#)

4.20 MIPS::FormatIIIEncoder Class Reference

```
#include <format_III_encoder.hpp>
```

Inheritance diagram for MIPS::FormatIIIEncoder:



Public Member Functions

- [instruction_t encode](#) ()
- void [parse](#) (std::vector< char * > ¶ms)

Additional Inherited Members

4.20.1 Detailed Description

Codificador responsável por transformar uma instrução assembly em uma instrução binária, seguindo o formato de instrução no formato III.

Author

Matheus Nogueira

4.20.2 Member Function Documentation

4.20.2.1 instruction_t MIPS::FormatIIIEncoder::encode () [virtual]

Codifica uma instrução assembly do formato III para uma instrução binária.

Returns

instrução binária de 16 bits

Implements [MIPS::Encoder](#).

4.20.2.2 void MIPS::FormatIIIEncoder::parse (std::vector< char * > ¶ms) [virtual]

Realiza uma varredura na instrução assembly e define seus campos binários.

Parameters

<i>params</i>	vector de parâmetros da instrução
---------------	-----------------------------------

Implements [MIPS::Encoder](#).

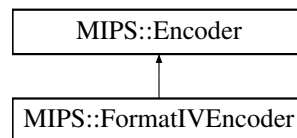
The documentation for this class was generated from the following file:

- `include/mips/interpreter/encoder/format_III_encoder.hpp`

4.21 MIPS::FormatIVEncoder Class Reference

```
#include <format_IV_encoder.hpp>
```

Inheritance diagram for MIPS::FormatIVEncoder:



Public Member Functions

- [instruction_t encode](#) ()
- void [parse](#) (std::vector< char * > ¶ms)

Additional Inherited Members

4.21.1 Detailed Description

Codificador responsável por transformar uma instrução assembly em uma instrução binária, seguindo o formato de instrução no formato IV.

Author

Matheus Nogueira

4.21.2 Member Function Documentation

4.21.2.1 instruction_t MIPS::FormatIVEncoder::encode () [virtual]

Codifica uma instrução assembly do formato IV para uma instrução binária.

Returns

instrução binária de 16 bits

Implements [MIPS::Encoder](#).

4.21.2.2 void MIPS::FormatIVEncoder::parse (std::vector< char * > ¶ms) [virtual]

Realiza uma varredura na instrução assembly e define seus campos binários.

Parameters

<i>params</i>	vector de parâmetros da instrução
---------------	-----------------------------------

Implements [MIPS::Encoder](#).

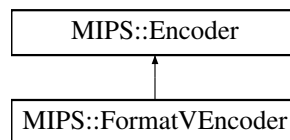
The documentation for this class was generated from the following file:

- [include/mips/interpreter/encoder/format_IV_encoder.hpp](#)

4.22 MIPS::FormatVEncoder Class Reference

```
#include <format_V_encoder.hpp>
```

Inheritance diagram for MIPS::FormatVEncoder:



Public Member Functions

- [instruction_t encode](#) ()
- void [parse](#) (std::vector< char * > ¶ms)

Additional Inherited Members

4.22.1 Detailed Description

Codificador responsável por transformar uma instrução assembly em uma instrução binária, seguindo o formato de instrução no formato V.

Author

Matheus Nogueira

4.22.2 Member Function Documentation

4.22.2.1 instruction_t MIPS::FormatVEncoder::encode () [virtual]

Codifica uma instrução assembly do formato V para uma instrução binária.

Returns

instrução binária de 16 bits

Implements [MIPS::Encoder](#).

4.22.2.2 void MIPS::FormatVEncoder::parse (std::vector< char * > ¶ms) [virtual]

Realiza uma varredura na instrução assembly e define seus campos binários.

Parameters

<i>params</i>	vector de parâmetros da instrução
---------------	-----------------------------------

Implements [MIPS::Encoder](#).

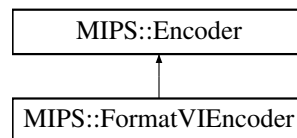
The documentation for this class was generated from the following file:

- [include/mips/interpreter/encoder/format_V_encoder.hpp](#)

4.23 MIPS::FormatVIEncoder Class Reference

```
#include <format_VI_encoder.hpp>
```

Inheritance diagram for MIPS::FormatVIEncoder:



Public Member Functions

- [instruction_t encode](#) ()
- void [parse](#) (std::vector< char * > ¶ms)

Additional Inherited Members

4.23.1 Detailed Description

Codificador responsável por transformar uma instrução assembly em uma instrução binária, seguindo o formato de instrução no formato VI.

Author

Matheus Nogueira

4.23.2 Member Function Documentation

4.23.2.1 `instruction_t MIPS::FormatVIEncoder::encode ()` `[virtual]`

Codifica uma instrução assembly do formato V para uma instrução binária.

Returns

instrução binária de 16 bits

Implements [MIPS::Encoder](#).

4.23.2.2 `void MIPS::FormatVIEncoder::parse (std::vector< char * > ¶ms)` `[virtual]`

Realiza uma varredura na instrução assembly e define seus campos binários.

Parameters

<i>params</i>	vector de parâmetros da instrução
---------------	-----------------------------------

Implements [MIPS::Encoder](#).

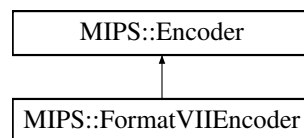
The documentation for this class was generated from the following file:

- [include/mips/interpreter/encoder/format_VI_encoder.hpp](#)

4.24 MIPS::FormatVIIEncoder Class Reference

```
#include <format_VII_encoder.hpp>
```

Inheritance diagram for MIPS::FormatVIIEncoder:



Public Member Functions

- [instruction_t encode](#) ()
- void [parse](#) (std::vector< char * > ¶ms)

Additional Inherited Members

4.24.1 Detailed Description

Codificador responsável por transformar uma instrução assembly em uma instrução binária, seguindo o formato de instrução no formato VII.

Author

Matheus Nogueira

4.24.2 Member Function Documentation

4.24.2.1 instruction_t MIPS::FormatVIIEncoder::encode () [virtual]

Codifica uma instrução assembly do formato V para uma instrução binária.

Returns

instrução binária de 16 bits

Implements [MIPS::Encoder](#).

4.24.2.2 void MIPS::FormatVIIEncoder::parse (std::vector< char * > ¶ms) [virtual]

Realiza uma varredura na instrução assembly e define seus campos binários.

Parameters

<i>params</i>	vector de parâmetros da instrução
---------------	-----------------------------------

Implements [MIPS::Encoder](#).

The documentation for this class was generated from the following file:

- [include/mips/interpreter/encoder/format_VII_encoder.hpp](#)

4.25 MIPS::FullAdder Class Reference

```
#include <full_adder.hpp>
```

Public Member Functions

- [FullAdder](#) ()
- [bit16_t add](#) ([bit16_t](#) a, [bit16_t](#) b, [bit8_t](#) c=0)
- [bool overflow](#) ()

4.25.1 Detailed Description

Classe responsável por realizar as operações de um somador de 16 bits.

Author

Matheus Nogueira

4.25.2 Constructor & Destructor Documentation

4.25.2.1 MIPS::FullAdder::FullAdder ()

Cria um novo somador.

4.25.3 Member Function Documentation

4.25.3.1 [bit16_t](#) MIPS::FullAdder::add ([bit16_t](#) a, [bit16_t](#) b, [bit8_t](#) c = 0)

Soma dois números de 16 bits.

Parameters

<i>a</i>	primeiro parametro da soma
<i>b</i>	segundo parametro da soma
<i>c</i>	carry de entrada (padrão: 0)

Returns

resultado da soma entre a e b

4.25.3.2 bool MIPS::FullAdder::overflow ()

Verifica se houve overflow na operação de adição.

Returns

true se houve overflow.

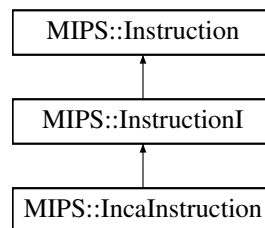
The documentation for this class was generated from the following file:

- include/mips/circuits/[full_adder.hpp](#)

4.26 MIPS::IncaInstruction Class Reference

```
#include <inca.hpp>
```

Inheritance diagram for MIPS::IncaInstruction:

**Public Member Functions**

- [IncaInstruction](#) (bit8_t opcode, Register *rs, Register *rt, bit8_t shamt, bit8_t funct)
- [bit16_t execute](#) ()

Additional Inherited Members**4.26.1 Detailed Description**

Classe que faz a operação de INCA no processador.

Author

Felipe Dias

4.26.2 Constructor & Destructor Documentation

4.26.2.1 `MIPS::IncalInstruction::IncalInstruction (bit8_t opcode, Register * rs, Register * rt, bit8_t shamt, bit8_t funct)`
`[inline]`

Constroi uma nova instrução.

4.26.3 Member Function Documentation

4.26.3.1 `bit16_t MIPS::IncalInstruction::execute ()` `[virtual]`

Função que executa a operação de incremento.

Returns

resultado da operação

Implements [MIPS::InstructionI](#).

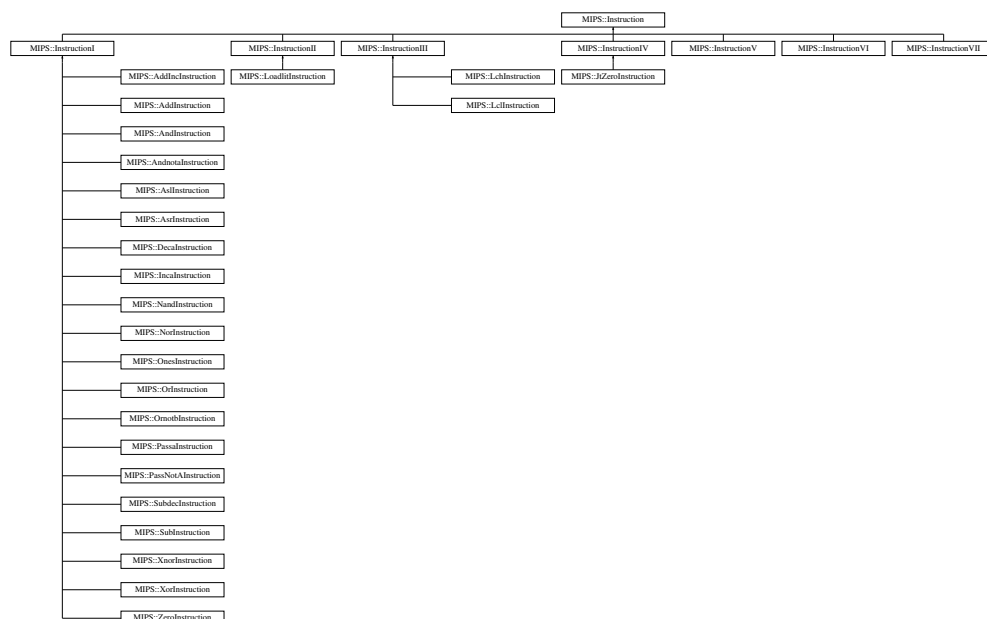
The documentation for this class was generated from the following file:

- `include/mips/instructions/format_l/inca.hpp`

4.27 MIPS::Instruction Class Reference

```
#include <instruction.hpp>
```

Inheritance diagram for MIPS::Instruction:



Public Member Functions

- [~Instruction](#) ()
- virtual [bit16_t execute](#) ()=0
- virtual void [updateControl](#) ([ControlUnit](#) &control)

Protected Attributes

- [bit8_t opcode](#)
- [bit8_t zero](#)
- [bit8_t neg](#)
- [bit8_t carry](#)
- [bit8_t overflow](#)

4.27.1 Detailed Description

Classe abstrata responsável por representar qualquer instrução em uma arquitetura de 16 bits.

Author

Matheus Nogueira

4.27.2 Constructor & Destructor Documentation

4.27.2.1 MIPS::Instruction::~~Instruction () [inline]

Destroi a instrução.

4.27.3 Member Function Documentation

4.27.3.1 virtual bit16_t MIPS::Instruction::execute () [pure virtual]

Método abstrato que deverá ser invocado para que uma instrução seja executada pelo emulador.

Returns

resultado de saída da instrução.

Implemented in [MIPS::InstructionI](#), [MIPS::InstructionVII](#), [MIPS::InstructionVI](#), [MIPS::InstructionII](#), [MIPS::InstructionIII](#), [MIPS::InstructionIV](#), [MIPS::InstructionV](#), [MIPS::LchInstruction](#), [MIPS::LclInstruction](#), [MIPS::LoadlitInstruction](#), [MIPS::AddInstruction](#), [MIPS::AddIncInstruction](#), [MIPS::AndnotInstruction](#), [MIPS::AslInstruction](#), [MIPS::AsrInstruction](#), [MIPS::DecalInstruction](#), [MIPS::IncalInstruction](#), [MIPS::NandInstruction](#), [MIPS::NorInstruction](#), [MIPS::OnesInstruction](#), [MIPS::PassalInstruction](#), [MIPS::PassNotAlInstruction](#), [MIPS::SubInstruction](#), [MIPS::XnorInstruction](#), [MIPS::XorInstruction](#), [MIPS::ZeroInstruction](#), [MIPS::JtZeroInstruction](#), [MIPS::SubdecInstruction](#), [MIPS::AndInstruction](#), [MIPS::OrInstruction](#), and [MIPS::OrnotbInstruction](#).

4.27.3.2 virtual void MIPS::Instruction::updateControl ([ControlUnit](#) & *control*) [inline], [virtual]

Método utilizado para atualizar os sinais de controle do processador.s

Parameters

<i>control</i>	unidade de controle do processador.
----------------	-------------------------------------

4.27.4 Member Data Documentation

4.27.4.1 `bit8_t MIPS::Instruction::carry` [protected]

Flag de carry.

4.27.4.2 `bit8_t MIPS::Instruction::neg` [protected]

Flag de negativo

4.27.4.3 `bit8_t MIPS::Instruction::opcode` [protected]

Código da operação (opcode) da instrução.

4.27.4.4 `bit8_t MIPS::Instruction::overflow` [protected]

flag de overflow

4.27.4.5 `bit8_t MIPS::Instruction::zero` [protected]

Flag de zero.

The documentation for this class was generated from the following file:

- [include/mips/instructions/instruction.hpp](#)

4.28 MIPS::InstructionDecoder Class Reference

```
#include <instruction_decoder.hpp>
```

Public Member Functions

- [InstructionDecoder](#) ([RegisterBank](#) &bank)
- [~InstructionDecoder](#) ()
- [Instruction](#) * [decode](#) ([instruction_t](#) instruction)
- [bit8_t](#) [getOPCode](#) ([instruction_t](#) instruction)
- [bit8_t](#) [getRs](#) ([instruction_t](#) instruction)
- [bit8_t](#) [getRt](#) ([instruction_t](#) instruction)
- [bit8_t](#) [getRd](#) ([instruction_t](#) instruction)
- [bit8_t](#) [getFunct](#) ([instruction_t](#) instruction)
- [bit16_t](#) [getOffset](#) ([instruction_t](#) instruction, [bit8_t](#) size=8)

Protected Attributes

- [RegisterBank](#) & `registerBank`

4.28.1 Detailed Description

Classe responsável por receber uma instrução em binário e instanciar uma instrução do emulador que pode executar a instrução equivalente.

Author

Matheus Nogueira

4.28.2 Constructor & Destructor Documentation

4.28.2.1 MIPS::InstructionDecoder::InstructionDecoder (`RegisterBank` & *bank*)

Cria um novo decodificador de instruções.

Parameters

<i>bank</i>	banco de registradores usado.
-------------	-------------------------------

4.28.2.2 MIPS::InstructionDecoder::~~InstructionDecoder ()

Destroi o decodificador de instruções.

4.28.3 Member Function Documentation

4.28.3.1 `Instruction*` MIPS::InstructionDecoder::decode (`instruction_t` *instruction*)

Decodifica uma instrução em binário e cria uma instrução do emulador que realize a operação equivalente.

Parameters

<i>instruction</i>	instrução 16 bits em binário.
--------------------	-------------------------------

Returns

ponteiro para a instrução criada pelo emulador.

4.28.3.2 `bit8_t` MIPS::InstructionDecoder::getFunct (`instruction_t` *instruction*)

Função que recupera o valor do funct da instrução.

Parameters

<i>instruction</i>	instrução binária de 16 bits.
--------------------	-------------------------------

Returns

valor do funct

4.28.3.3 `bit16_t MIPS::InstructionDecoder::getOffset (instruction_t instruction, bit8_t size = 8)`

Função que recupera o valor do offset da instrução.

Parameters

<i>instruction</i>	instrução binária de 16 bits.
<i>size</i>	número de bits de offset

Returns

valor do offset.

4.28.3.4 `bit8_t MIPS::InstructionDecoder::getOPCode (instruction_t instruction)`

Método responsável por recuperar o código de operação (opcode) de uma instrução.

Parameters

<i>instruction</i>	instrução de onde o opcode deve ser extraído.
--------------------	-----------------------------------------------

Returns

opcode da instrução.

4.28.3.5 `bit8_t MIPS::InstructionDecoder::getRd (instruction_t instruction)`

Função que recupera o endereço do registrador destination (Rd) da instrução.

Parameters

<i>instruction</i>	instrução binária de 16 bits.
--------------------	-------------------------------

Returns

endereço do registrador destination.

4.28.3.6 `bit8_t MIPS::InstructionDecoder::getRs (instruction_t instruction)`

Função que recupera o endereço do registrador source (Rs) da instrução.

Parameters

<code>instruction</code>	instrução binária de 16 bits.
--------------------------	-------------------------------

Returns

endereço do registrador source.

4.28.3.7 `bit8_t MIPS::InstructionDecoder::getRt (instruction_t instruction)`

Função que recupera o endereço do registrador target (Rt) da instrução.

Parameters

<code>instruction</code>	instrução binária de 16 bits.
--------------------------	-------------------------------

Returns

endereço do registrador target.

4.28.4 Member Data Documentation

4.28.4.1 `RegisterBank& MIPS::InstructionDecoder::registerBank` `[protected]`

Banco de registradores do decodificador.

The documentation for this class was generated from the following file:

- `include/mips/decoder/instruction_decoder.hpp`

4.29 MIPS::InstructionFinder Class Reference

```
#include <instruction_finder.hpp>
```

Public Member Functions

- `InstructionFinder (Memory &memoryUnit, RegisterBank &bank)`
- `~InstructionFinder ()`
- `instruction_t getNext ()`

4.29.1 Detailed Description

Classe que representa a unidade de busca de instruções na memória.

Author

Matheus Nogueira

4.29.2 Constructor & Destructor Documentation

4.29.2.1 MIPS::InstructionFinder::InstructionFinder (*Memory* & *memoryUnit*, *RegisterBank* & *bank*)

Cria uma nova unidade para busca de instruções.

4.29.2.2 MIPS::InstructionFinder::~~InstructionFinder ()

Destroi a unidade de busca de instruções.

4.29.3 Member Function Documentation

4.29.3.1 `instruction_t` MIPS::InstructionFinder::getNext ()

Retorna a próxima instrução a ser executada, utilizando o registrador PC para identificá-la.

Returns

próxima instrução a ser executada.

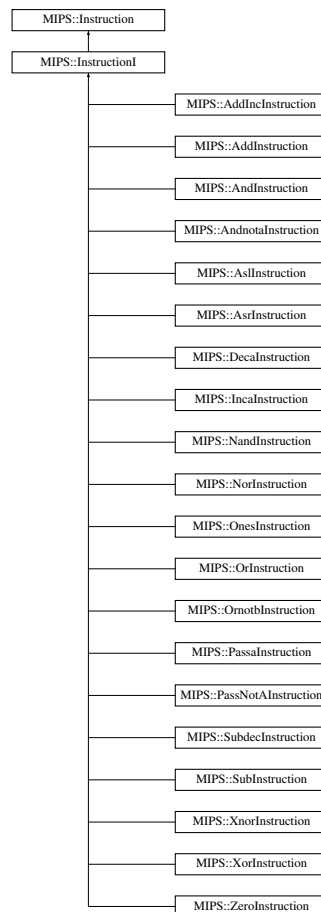
The documentation for this class was generated from the following file:

- `include/mips/units/instruction_finder.hpp`

4.30 MIPS::InstructionI Class Reference

```
#include <instruction_I.hpp>
```

Inheritance diagram for MIPS::InstructionI:



Public Member Functions

- [InstructionI](#) (bit8_t opcode, [Register](#) *rs, [Register](#) *rt, bit8_t shamt, bit8_t funct)
- virtual [~InstructionI](#) ()
- virtual [bit16_t execute](#) ()=0

Protected Attributes

- [Register](#) * rs
- [Register](#) * rt
- bit8_t shamt
- bit8_t funct

4.30.1 Detailed Description

Classe que representa uma instrução do tipo (R)egister.

Author

Matheus Nogueira

4.30.2 Constructor & Destructor Documentation

4.30.2.1 `MIPS::InstructionI::InstructionI (bit8_t opcode, Register * rs, Register * rt, bit8_t shamt, bit8_t funct)`
`[inline]`

Cria uma nova instrução do formato I.

Parameters

<i>opcode</i>	codigo da operação
<i>rs</i>	registrador source
<i>rt</i>	registrador target
<i>shamt</i>	quantidade de shift
<i>funct</i>	bits para escolha da função da instrução

4.30.2.2 `virtual MIPS::InstructionI::~~InstructionI ()` `[inline]`, `[virtual]`

Destroi a instrução.

4.30.3 Member Function Documentation

4.30.3.1 `virtual bit16_t MIPS::InstructionI::execute ()` `[pure virtual]`

Executa a instrução.

Returns

resultado da instrução

Implements [MIPS::Instruction](#).

Implemented in [MIPS::AddInstruction](#), [MIPS::AddIncInstruction](#), [MIPS::AndnotInstruction](#), [MIPS::AslInstruction](#), [MIPS::AsrInstruction](#), [MIPS::DecalInstruction](#), [MIPS::IncalInstruction](#), [MIPS::NandInstruction](#), [MIPS::NorInstruction](#), [MIPS::OnesInstruction](#), [MIPS::PassalInstruction](#), [MIPS::PassNotAInstruction](#), [MIPS::SubInstruction](#), [MIPS::XnorInstruction](#), [MIPS::XorInstruction](#), [MIPS::ZeroInstruction](#), [MIPS::SubdecInstruction](#), [MIPS::AndInstruction](#), [MIPS::OrInstruction](#), and [MIPS::OrnotbInstruction](#).

4.30.4 Member Data Documentation

4.30.4.1 `bit8_t MIPS::InstructionI::funct` `[protected]`

Valor da funct da instrução.

4.30.4.2 `Register* MIPS::InstructionI::rs` `[protected]`

Registrador source (Rs) da instrução.

4.30.4.3 Register* MIPS::InstructionI::rt [protected]

Registrador target (Rt) da instrução.

4.30.4.4 bit8_t MIPS::InstructionI::shamt [protected]

Valor do shamt (shift amount) da instrução.

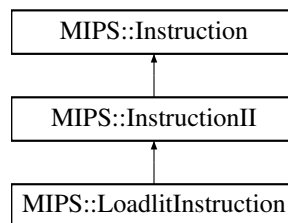
The documentation for this class was generated from the following file:

- [include/mips/instructions/instruction_I.hpp](#)

4.31 MIPS::InstructionII Class Reference

```
#include <instruction_II.hpp>
```

Inheritance diagram for MIPS::InstructionII:



Public Member Functions

- [InstructionII](#) ([bit8_t opcode](#), [Register *rd](#), [bit16_t offset](#))
- virtual [bit16_t execute](#) ()=0

Protected Attributes

- [Register * rd](#)
- [bit16_t offset](#)

4.31.1 Detailed Description

Classe que representa uma instrução do formato II do trabalho.

Author

Matheus Nogueira

4.31.2 Constructor & Destructor Documentation

4.31.2.1 MIPS::InstructionII::InstructionII ([bit8_t opcode](#), [Register * rd](#), [bit16_t offset](#)) [inline]

Cria uma nova instrução do formato II

Parameters

<i>opcode</i>	código da operação
<i>rd</i>	registrador de destino
<i>offset</i>	offset de 11 bits.

4.31.3 Member Function Documentation

4.31.3.1 `virtual bit16_t MIPS::InstructionIII::execute () [pure virtual]`

Executa a instrução.

Returns

resultado da instrução

Implements [MIPS::Instruction](#).

Implemented in [MIPS::LoadlitInstruction](#).

4.31.4 Member Data Documentation

4.31.4.1 `bit16_t MIPS::InstructionIII::offset [protected]`

Offset utilizado pela instrução.

4.31.4.2 `Register* MIPS::InstructionIII::rd [protected]`

Registrador de destino da instrução.

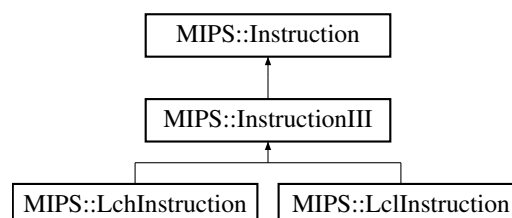
The documentation for this class was generated from the following file:

- [include/mips/instructions/instruction_II.hpp](#)

4.32 MIPS::InstructionIII Class Reference

```
#include <instruction_III.hpp>
```

Inheritance diagram for MIPS::InstructionIII:



Public Member Functions

- [InstructionIII](#) ([bit8_t opcode](#), [Register *rd](#), [bit8_t offset](#))
- virtual [bit16_t execute](#) ()=0

Protected Attributes

- [Register * rd](#)
- [bit8_t offset](#)

4.32.1 Detailed Description

Classe que representa uma instrução do formato III do trabalho.

Author

Matheus Nogueira

4.32.2 Constructor & Destructor Documentation

4.32.2.1 `MIPS::InstructionIII::InstructionIII (bit8_t opcode, Register * rd, bit8_t offset)` `[inline]`

Cria uma nova instrução do formato III

Parameters

<i>opcode</i>	código da operação
<i>rd</i>	registrador de destino
<i>offset</i>	offset de 8 bits

4.32.3 Member Function Documentation

4.32.3.1 `virtual bit16_t MIPS::InstructionIII::execute ()` `[pure virtual]`

Executa a instrução.

Returns

resultado da instrução

Implements [MIPS::Instruction](#).

Implemented in [MIPS::LchInstruction](#), and [MIPS::LclInstruction](#).

4.32.4 Member Data Documentation

4.32.4.1 `bit8_t MIPS::InstructionIII::offset` `[protected]`

Offset utilizado pela instrução.

4.32.4.2 `Register* MIPS::InstructionIII::rd` `[protected]`

Registrador de destino da instrução.

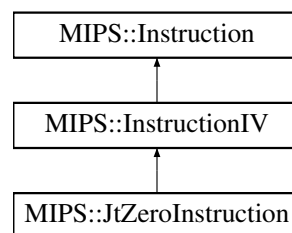
The documentation for this class was generated from the following file:

- [include/mips/instructions/instruction_III.hpp](#)

4.33 MIPS::InstructionIV Class Reference

```
#include <instruction_IV.hpp>
```

Inheritance diagram for MIPS::InstructionIV:



Public Member Functions

- [InstructionIV](#) ([bit8_t opcode](#), [struct ALUFlags &aluFlags](#), [bit8_t offset](#))
- virtual [bit16_t execute](#) ()=0

Protected Attributes

- [bit8_t offset](#)
- [struct ALUFlags & flags](#)

4.33.1 Detailed Description

Classe que representa uma instrução do formato IV do trabalho.

Author

Matheus Nogueira

4.33.2 Constructor & Destructor Documentation

4.33.2.1 `MIPS::InstructionIV::InstructionIV (bit8_t opcode, struct ALUFlags & aluFlags, bit8_t offset)` `[inline]`

Cria uma nova instrução do formato IV

Parameters

<i>opcode</i>	código da operação
<i>aluFlags</i>	Flags vindas da ALU
<i>offset</i>	offset do jump.

4.33.3 Member Function Documentation

4.33.3.1 `virtual bit16_t MIPS::InstructionIV::execute () [pure virtual]`

Executa a instrução.

Returns

0 se o desvio não é tomado, 1 se o desvio for tomado.

Implements [MIPS::Instruction](#).

Implemented in [MIPS::JtZeroInstruction](#).

4.33.4 Member Data Documentation

4.33.4.1 `struct ALUFlags& MIPS::InstructionIV::flags [protected]`

Flags de saída da ALU

4.33.4.2 `bit8_t MIPS::InstructionIV::offset [protected]`

Offset da instrução de jump

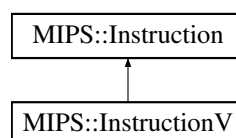
The documentation for this class was generated from the following file:

- [include/mips/instructions/instruction_IV.hpp](#)

4.34 MIPS::InstructionV Class Reference

```
#include <instruction_V.hpp>
```

Inheritance diagram for MIPS::InstructionV:



Public Member Functions

- [InstructionV](#) ([bit8_t opcode](#), [bit16_t offset](#))
- virtual [bit16_t execute](#) ()=0

Protected Attributes

- [bit16_t offset](#)

4.34.1 Detailed Description

Classe que representa uma instrução do formato V do trabalho.

Author

Matheus Nogueira

4.34.2 Constructor & Destructor Documentation

4.34.2.1 MIPS::InstructionV::InstructionV ([bit8_t opcode](#), [bit16_t offset](#)) [\[inline\]](#)

Cria uma nova instrução do formato V

Parameters

<i>opcode</i>	código da operação
<i>offset</i>	offset do jump.

4.34.3 Member Function Documentation

4.34.3.1 virtual [bit16_t](#) MIPS::InstructionV::execute () [\[pure virtual\]](#)

Executa a instrução.

Returns

endereço da próxima instrução.

Implements [MIPS::Instruction](#).

4.34.4 Member Data Documentation

4.34.4.1 [bit16_t](#) MIPS::InstructionV::offset [\[protected\]](#)

Offset do jump

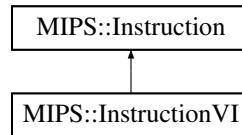
The documentation for this class was generated from the following file:

- [include/mips/instructions/instruction_V.hpp](#)

4.35 MIPS::InstructionVI Class Reference

```
#include <instruction_VI.hpp>
```

Inheritance diagram for MIPS::InstructionVI:



Public Member Functions

- [InstructionVI](#) ([bit8_t opcode](#), [Register *rt](#), [Register *pc](#))
- virtual [bit16_t execute](#) ()=0

Protected Attributes

- [Register * pc](#)
- [Register * rt](#)

4.35.1 Detailed Description

Classe que representa uma instrução do formato VI do trabalho.

Author

Matheus Nogueira

4.35.2 Constructor & Destructor Documentation

4.35.2.1 MIPS::InstructionVI::InstructionVI ([bit8_t opcode](#), [Register * rt](#), [Register * pc](#)) [\[inline\]](#)

Cria uma nova instrução do formato VI

Parameters

<i>opcode</i>	código da operação
<i>rt</i>	registrador target
<i>pc</i>	registrador de contador de programa

4.35.3 Member Function Documentation

4.35.3.1 `virtual bit16_t MIPS::InstructionVI::execute () [pure virtual]`

Executa a instrução.

Returns

resultado da instrução

Implements [MIPS::Instruction](#).

4.35.4 Member Data Documentation

4.35.4.1 `Register* MIPS::InstructionVI::pc [protected]`

Registrador de contador de programa

4.35.4.2 `Register* MIPS::InstructionVI::rt [protected]`

Registrador target.

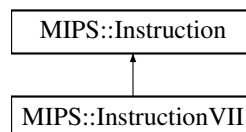
The documentation for this class was generated from the following file:

- [include/mips/instructions/instruction_VI.hpp](#)

4.36 MIPS::InstructionVII Class Reference

```
#include <instruction_VII.hpp>
```

Inheritance diagram for MIPS::InstructionVII:



Public Member Functions

- [InstructionVII](#) ([bit8_t opcode](#), [Register *rs](#), [Register *rt](#), [Memory *memory](#))
- `virtual bit16_t execute ()=0`

Protected Attributes

- [Register * rs](#)
- [Register * rt](#)
- [Memory * memory](#)

4.36.1 Detailed Description

Classe que representa uma instrução do formato VII do trabalho.

Author

Matheus Nogueira

4.36.2 Constructor & Destructor Documentation

4.36.2.1 `MIPS::InstructionVII::InstructionVII (bit8_t opcode, Register * rs, Register * rt, Memory * memory)`
[inline]

Cria uma nova instrução do formato VII

Parameters

<i>opcode</i>	código da operação
<i>rs</i>	registrador destination
<i>rt</i>	registrador target
<i>memory</i>	unidade de memória do processador

4.36.3 Member Function Documentation

4.36.3.1 `virtual bit16_t MIPS::InstructionVII::execute ()` [pure virtual]

Executa a instrução.

Returns

resultado da instrução

Implements [MIPS::Instruction](#).

4.36.4 Member Data Documentation

4.36.4.1 `Memory* MIPS::InstructionVII::memory` [protected]

Unidade de memória usada para acessar os dados.

4.36.4.2 `Register* MIPS::InstructionVII::rs` [protected]

Registrador source

4.36.4.3 Register* MIPS::InstructionVII::rt [protected]

Registrador target.

The documentation for this class was generated from the following file:

- include/mips/instructions/instruction_VII.hpp

4.37 MIPS::Interpreter Class Reference

```
#include <interpreter.hpp>
```

Public Member Functions

- [Interpreter](#) (const char *file)
- [~Interpreter](#) ()
- void [compile](#) (const char *output="out.mips")
- bool [ok](#) ()

4.37.1 Detailed Description

Classe responsável por receber um texto e criar instruções 16 bits correspondentes para a arquitetura MIPS 32.

Author

Matheus Nogueira

4.37.2 Constructor & Destructor Documentation

4.37.2.1 MIPS::Interpreter::Interpreter (const char * *file*)

Cria uma nova instância do interpretador.

Parameters

<i>file</i>	arquivo que será interpretado.
-------------	--------------------------------

4.37.2.2 MIPS::Interpreter::~~Interpreter ()

Destroi o interpretador.

4.37.3 Member Function Documentation

4.37.3.1 void MIPS::Interpreter::compile (const char * *output* = "out.mips")

Processa o arquivo de entrada para que ele possa ser interpretado.

Parameters

<i>output</i>	arquivo que será escrito com as instruções.
---------------	---------------------------------------------

4.37.3.2 bool MIPS::Interpreter::ok () [inline]

Checa se o interpretador encontrou algum erro durante a sua execução. Se sim, retorna false.

Returns

true se o interpretador realizou sem papel sem erros.

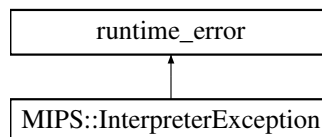
The documentation for this class was generated from the following file:

- include/mips/interpreter/interpreter.hpp

4.38 MIPS::InterpreterException Class Reference

```
#include <interpreter_exception.hpp>
```

Inheritance diagram for MIPS::InterpreterException:



Public Member Functions

- [InterpreterException](#) (const char *msg, [bit8_t](#) opcode)
- virtual const char * [what](#) () const throw ()
- [bit8_t](#) [getCode](#) ()

4.38.1 Detailed Description

Exceção que é lançada pelo interpretador quando um erro é encontrado no arquivo fonte.

Author

Matheus Nogueira

4.38.2 Constructor & Destructor Documentation

4.38.2.1 MIPS::InterpreterException::InterpreterException (const char * *msg*, [bit8_t](#) *opcode*) [inline]

Cria uma nova exceção.

Parameters

<i>msg</i>	mensagem de erro.
<i>opcode</i>	código do erro.

4.38.3 Member Function Documentation

4.38.3.1 `bit8_t MIPS::InterpreterException::getCode () [inline]`

Retorna o codigo de operação da exceção.

Returns

codigo de operação.

4.38.3.2 `virtual const char* MIPS::InterpreterException::what () const throw () [inline], [virtual]`

Retorna a mensagem de erro.

Returns

mensagem de erro.

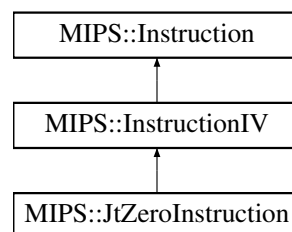
The documentation for this class was generated from the following file:

- [include/mips/interpreter/exception/interpreter_exception.hpp](#)

4.39 MIPS::JtZeroInstruction Class Reference

```
#include <jt_zero.hpp>
```

Inheritance diagram for MIPS::JtZeroInstruction:



Public Member Functions

- [JtZeroInstruction](#) ([bit8_t opcode](#), [struct ALUFlags &aluFlags](#), [bit8_t offset](#))
- [bit16_t execute](#) ()

Additional Inherited Members

4.39.1 Detailed Description

Instrução que faz o desvio se a flag zero da ALU está setada como 1.

Author

Matheus Nogueira

4.39.2 Constructor & Destructor Documentation

4.39.2.1 MIPS::JtZeroInstruction::JtZeroInstruction (*bit8_t opcode*, *struct ALUFlags & aluFlags*, *bit8_t offset*) [inline]

Cria uma instrução de jump on true.

Parameters

<i>opcode</i>	código da operação
<i>aluFlags</i>	objeto de flags da ALU
<i>offset</i>	offset de 11 bits

4.39.3 Member Function Documentation

4.39.3.1 bit16_t MIPS::JtZeroInstruction::execute () [virtual]

Executa a instrução.

Returns

retorna 0 se o desvio não for tomado, 1 se desvio for tomado.

Implements [MIPS::InstructionIV](#).

The documentation for this class was generated from the following file:

- include/mips/instructions/format_IV/jt_zero.hpp

4.40 MIPS::Label Struct Reference

```
#include <label.hpp>
```

Public Attributes

- char [label](#) [64]
Nome do label.
- unsigned long [line](#)
Linha que ele está definido.

4.40.1 Detailed Description

Estrutura que armazena o nome do label e a linha que ele se encontra.

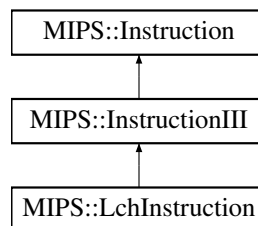
The documentation for this struct was generated from the following file:

- [include/mips/interpreter/label.hpp](#)

4.41 MIPS::LchInstruction Class Reference

```
#include <lch.hpp>
```

Inheritance diagram for MIPS::LchInstruction:



Public Member Functions

- [LchInstruction](#) ([bit8_t opcode](#), [Register *rd](#), [bit8_t offset](#))
- [bit16_t execute](#) ()

Additional Inherited Members

4.41.1 Detailed Description

Instrução utilizada para carregar 8 bits e carregá-los em um no bit mais significativo do registrador definido pelo programador.

Author

Matheus Nogueira

4.41.2 Constructor & Destructor Documentation

4.41.2.1 MIPS::LchInstruction::LchInstruction ([bit8_t opcode](#), [Register * rd](#), [bit8_t offset](#)) [\[inline\]](#)

Cria uma instrução de ldc.

Parameters

<i>opcode</i>	código da operação
<i>rd</i>	registrador rd
<i>offset</i>	offset de 11 bits

4.41.3 Member Function Documentation

4.41.3.1 `bit16_t MIPS::LchInstruction::execute ()` [virtual]

Executa a instrução.

Returns

resultado da operação

Implements [MIPS::InstructionIII](#).

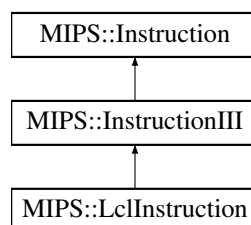
The documentation for this class was generated from the following file:

- [include/mips/instructions/format_III/lch.hpp](#)

4.42 MIPS::LclInstruction Class Reference

```
#include <lcl.hpp>
```

Inheritance diagram for MIPS::LclInstruction:



Public Member Functions

- [LclInstruction](#) ([bit8_t opcode](#), [Register *rd](#), [bit8_t offset](#))
- [bit16_t execute](#) ()

Additional Inherited Members

4.42.1 Detailed Description

Instrução utilizada para carregar 8 bits e carregá-los em um no bit menos significativo do registrador definido pelo programador.

Author

Matheus Nogueira

4.42.2 Constructor & Destructor Documentation

4.42.2.1 MIPS::LclInstruction::LclInstruction (*bit8_t opcode*, *Register * rd*, *bit8_t offset*) [inline]

Cria uma instrução de lhc.

Parameters

<i>opcode</i>	código da operação
<i>rd</i>	registrador de destino
<i>offset</i>	offset de 11 bits

4.42.3 Member Function Documentation

4.42.3.1 *bit16_t* MIPS::LclInstruction::execute () [virtual]

Executa a instrução.

Returns

resultado da operação

Implements [MIPS::InstructionIII](#).

The documentation for this class was generated from the following file:

- [include/mips/instructions/format_III/lcl.hpp](#)

4.43 MIPS::EventDispatcher::ListenerMap Struct Reference

```
#include <event_dispatcher.hpp>
```

Public Attributes

- [EventType](#) type
Tipo de evento.
- [Queue](#)< [EventListener](#) * > * [listeners](#)
Fila de ouvintes.

4.43.1 Detailed Description

Classe responsável por representar um tipo de evento ligado a uma fila de ouvintes que devem ser notificados.

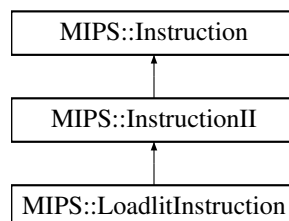
The documentation for this struct was generated from the following file:

- include/mips/util/event/[event_dispatcher.hpp](#)

4.44 MIPS::LoadlitInstruction Class Reference

```
#include <loadlit.hpp>
```

Inheritance diagram for MIPS::LoadlitInstruction:



Public Member Functions

- [LoadlitInstruction](#) ([bit8_t](#) opcode, [Register](#) *rd, [bit16_t](#) offset)
- [bit16_t](#) execute ()

Additional Inherited Members

4.44.1 Detailed Description

Instrução utilizada para carregar 11 bits e carregá-los em um registrador definido pelo programador.

Author

Matheus Nogueira

4.44.2 Constructor & Destructor Documentation

4.44.2.1 MIPS::LoadlitInstruction::LoadlitInstruction ([bit8_t](#) opcode, [Register](#) * rd, [bit16_t](#) offset) [\[inline\]](#)

Cria uma instrução de loadlit.

Parameters

<i>opcode</i>	código da operação
<i>rd</i>	registrador de destino
<i>offset</i>	offset de 11 bits

4.44.3 Member Function Documentation

4.44.3.1 `bit16_t MIPS::LoadlitInstruction::execute () [virtual]`

Executa a instrução.

Returns

resultado da operação

Implements [MIPS::InstructionII](#).

The documentation for this class was generated from the following file:

- `include/mips/instructions/format_II/loadlit.hpp`

4.45 MIPS::Memory Class Reference

```
#include <memory.hpp>
```

Public Member Functions

- [Memory](#) ()
- [~Memory](#) ()
- void [setInstructionSize](#) (size_t size)
- void [setDataSize](#) (size_t size)
- void [write](#) (bit16_t data, bit32_t offset, bit8_t iOrD=1)
- bit16_t [read](#) (bit32_t offset, bit8_t iOrD=1)

4.45.1 Detailed Description

Classe responsável por iteragir com a memória do processador, para assim, criar uma interface de maior facilidade para acessar a memória.

Author

Matheus Nogueira

4.45.2 Constructor & Destructor Documentation

4.45.2.1 MIPS::Memory::Memory ()

Cria uma nova unidade de memória.

4.45.2.2 MIPS::Memory::~~Memory ()

Destroi a unidade de memória.

4.45.3 Member Function Documentation

4.45.3.1 bit16_t MIPS::Memory::read (bit32_t offset, bit8_t iOrD = 1)

L uma palavra que está na posição de memória especificada.

Parameters

<i>offset</i>	posição da memória que será lida.
<i>iOrD</i>	tipo de dado que será lido (instrução ou dado) (padrão: dado)

Returns

palavra armazenada na posição de memória especificada.

4.45.3.2 void MIPS::Memory::setDataSize (size_t size)

Define o tamanho da memória de dados.

Parameters

<i>size</i>	tamanho da memória em número de palavras.
-------------	-------------------------------------------

4.45.3.3 void MIPS::Memory::setInstructionSize (size_t size)

Define o tamanho da memória de instruções.

Parameters

<i>size</i>	tamanho da memória em número de palavras.
-------------	-------------------------------------------

4.45.3.4 void MIPS::Memory::write (bit16_t data, bit32_t offset, bit8_t iOrD = 1)

Escreve uma palavra na posição de memória especificada.

Parameters

<i>data</i>	palavra que será escrita.
<i>offset</i>	posição da memória em que a palavra será escrita.
<i>iOrD</i>	tipo de dado que será escrito (instrução ou dados) (padrão: dado)

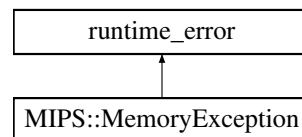
The documentation for this class was generated from the following file:

- include/mips/memory/[memory.hpp](#)

4.46 MIPS::MemoryException Class Reference

```
#include <memory_exception.hpp>
```

Inheritance diagram for MIPS::MemoryException:



Public Member Functions

- [MemoryException](#) (const char *msg)
- virtual const char * [what](#) () const throw ()

4.46.1 Detailed Description

Exceção que é lançada pelo interpretador quando um erro é encontrado durante o acesso à memória.

Author

Matheus Nogueira

4.46.2 Constructor & Destructor Documentation

4.46.2.1 MIPS::MemoryException::MemoryException (const char * msg) [inline]

Cria uma nova exceção.

Parameters

<i>msg</i>	mensagem de erro.
------------	-------------------

4.46.3 Member Function Documentation

4.46.3.1 `virtual const char* MIPS::MemoryException::what () const throw ()` `[inline], [virtual]`

Retorna a mensagem de erro.

Returns

mensagem de erro.

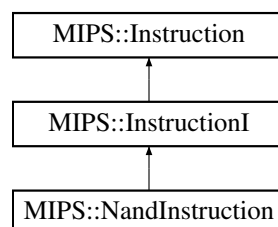
The documentation for this class was generated from the following file:

- [include/mips/memory/memory_exception.hpp](#)

4.47 MIPS::NandInstruction Class Reference

```
#include <nand.hpp>
```

Inheritance diagram for MIPS::NandInstruction:



Public Member Functions

- [NandInstruction](#) ([bit8_t opcode](#), [Register *rs](#), [Register *rt](#), [bit8_t shamt](#), [bit8_t funct](#))
- [bit16_t execute](#) ()

Additional Inherited Members

4.47.1 Detailed Description

Classe que faz a operação de NAND no processador.

Author

Matheus Nogueira

4.47.2 Constructor & Destructor Documentation

4.47.2.1 MIPS::NandInstruction::NandInstruction (bit8_t *opcode*, Register * *rs*, Register * *rt*, bit8_t *shamt*, bit8_t *funct*) [inline]

Constroi uma nova instrução.

4.47.3 Member Function Documentation

4.47.3.1 bit16_t MIPS::NandInstruction::execute () [virtual]

Função que executa a operação de soma.

Returns

resultado da operação

Implements [MIPS::InstructionI](#).

The documentation for this class was generated from the following file:

- [include/mips/instructions/format_l/nand.hpp](#)

4.48 MIPS::Queue< T >::Node Struct Reference

```
#include <queue.hpp>
```

Public Attributes

- T [content](#)
Conteúdo do nó da fila.
- struct [Node](#) * [next](#)
Próximo elemento da fila.

4.48.1 Detailed Description

```
template<typename T>
struct MIPS::Queue< T >::Node
```

Nó da fila.

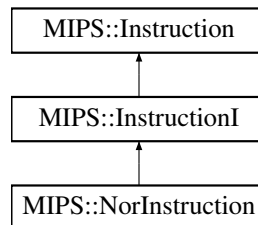
The documentation for this struct was generated from the following file:

- [include/mips/util/structure/queue.hpp](#)

4.49 MIPS::NorInstruction Class Reference

```
#include <nor.hpp>
```

Inheritance diagram for MIPS::NorInstruction:



Public Member Functions

- [NorInstruction](#) ([bit8_t opcode](#), [Register *rs](#), [Register *rt](#), [bit8_t shamt](#), [bit8_t funct](#))
- [bit16_t execute](#) ()

Additional Inherited Members

4.49.1 Detailed Description

Classe que faz a operação de NOR no processador.

Author

Felipe Dias

4.49.2 Constructor & Destructor Documentation

4.49.2.1 **MIPS::NorInstruction::NorInstruction** ([bit8_t opcode](#), [Register * rs](#), [Register * rt](#), [bit8_t shamt](#), [bit8_t funct](#))
[inline]

Constroi uma nova instrução.

4.49.3 Member Function Documentation

4.49.3.1 **bit16_t MIPS::NorInstruction::execute** () [virtual]

Função que executa a operação de soma.

Returns

resultado da operação

Implements [MIPS::InstructionI](#).

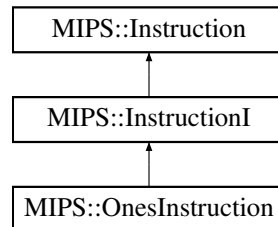
The documentation for this class was generated from the following file:

- [include/mips/instructions/format_l/nor.hpp](#)

4.50 MIPS::OnesInstruction Class Reference

```
#include <ones.hpp>
```

Inheritance diagram for MIPS::OnesInstruction:



Public Member Functions

- [OnesInstruction](#) ([bit8_t opcode](#), [Register *rs](#), [Register *rt](#), [bit8_t shamt](#), [bit8_t funct](#))
- [bit16_t execute](#) ()

Additional Inherited Members

4.50.1 Detailed Description

Classe que faz a operação de ONES no processador.

Author

Felipe Dias

4.50.2 Constructor & Destructor Documentation

4.50.2.1 `MIPS::OnesInstruction::OnesInstruction (bit8_t opcode, Register * rs, Register * rt, bit8_t shamt, bit8_t funct)` `[inline]`

Constroi uma nova instrução.

4.50.3 Member Function Documentation

4.50.3.1 `bit16_t MIPS::OnesInstruction::execute ()` `[virtual]`

Função que executa a operação de soma.

Returns

resultado da operação

Implements [MIPS::InstructionI](#).

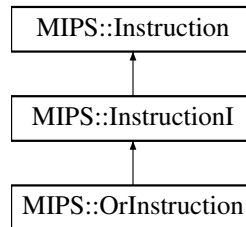
The documentation for this class was generated from the following file:

- `include/mips/instructions/format_l/ones.hpp`

4.51 MIPS::OrInstruction Class Reference

```
#include <or.hpp>
```

Inheritance diagram for MIPS::OrInstruction:



Public Member Functions

- [OrInstruction](#) ([bit8_t opcode](#), [Register *rs](#), [Register *rt](#), [bit8_t shamt](#), [bit8_t funct](#))
- [bit16_t execute](#) ()

Additional Inherited Members

4.51.1 Detailed Description

Classe que faz a operação de OR no processador.

Author

Lucas Fonseca dos Santos

4.51.2 Constructor & Destructor Documentation

4.51.2.1 `MIPS::OrInstruction::OrInstruction (bit8_t opcode, Register * rs, Register * rt, bit8_t shamt, bit8_t funct)`
`[inline]`

Constroi uma nova instrução.

4.51.3 Member Function Documentation

4.51.3.1 `bit16_t MIPS::OrInstruction::execute ()` `[virtual]`

Função que executa a operação de [OrInstruction](#).

Implements [MIPS::InstructionI](#).

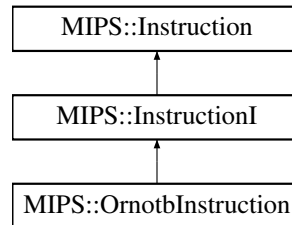
The documentation for this class was generated from the following file:

- `include/mips/instructions/format_l/or.hpp`

4.52 MIPS::OrnotbInstruction Class Reference

```
#include <ornotb.hpp>
```

Inheritance diagram for MIPS::OrnotbInstruction:



Public Member Functions

- [OrnotbInstruction](#) ([bit8_t opcode](#), [Register *rs](#), [Register *rt](#), [bit8_t shamt](#), [bit8_t funct](#))
- [bit16_t execute](#) ()

Additional Inherited Members

4.52.1 Detailed Description

Classe que faz a operação de ornotb no processador.

Author

Lucas Pereira

4.52.2 Constructor & Destructor Documentation

4.52.2.1 `MIPS::OrnotbInstruction::OrnotbInstruction (bit8_t opcode, Register * rs, Register * rt, bit8_t shamt, bit8_t funct) [inline]`

Constroi uma nova instrução.

4.52.3 Member Function Documentation

4.52.3.1 `bit16_t MIPS::OrnotbInstruction::execute () [virtual]`

Função que executa a operação de [OrnotbInstruction](#).

Implements [MIPS::InstructionI](#).

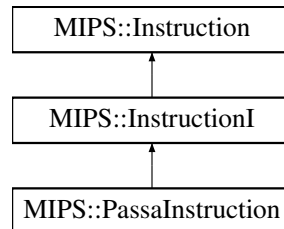
The documentation for this class was generated from the following file:

- `include/mips/instructions/format_l/ornotb.hpp`

4.53 MIPS::PassaInstruction Class Reference

```
#include <passa.hpp>
```

Inheritance diagram for MIPS::PassaInstruction:



Public Member Functions

- [PassaInstruction](#) ([bit8_t opcode](#), [Register *rs](#), [Register *rt](#), [bit8_t shamt](#), [bit8_t funct](#))
- [bit16_t execute](#) ()

Additional Inherited Members

4.53.1 Detailed Description

Classe que faz a operação de PASSA no processador.

Author

Matheus Nogueira

4.53.2 Constructor & Destructor Documentation

4.53.2.1 `MIPS::PassaInstruction::PassaInstruction (bit8_t opcode, Register * rs, Register * rt, bit8_t shamt, bit8_t funct)` `[inline]`

Constroi uma nova instrução.

4.53.3 Member Function Documentation

4.53.3.1 `bit16_t MIPS::PassaInstruction::execute ()` `[virtual]`

Função que executa a operação de soma.

Returns

resultado da operação

Implements [MIPS::InstructionI](#).

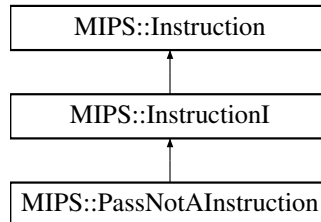
The documentation for this class was generated from the following file:

- `include/mips/instructions/format_l/passa.hpp`

4.54 MIPS::PassNotAInstruction Class Reference

```
#include <passnota.hpp>
```

Inheritance diagram for MIPS::PassNotAInstruction:



Public Member Functions

- [PassNotAInstruction](#) ([bit8_t opcode](#), [Register *rs](#), [Register *rt](#), [bit8_t shamt](#), [bit8_t funct](#))
- [bit16_t execute](#) ()

Additional Inherited Members

4.54.1 Detailed Description

Classe que faz a operação de PASSNOTA no processador.

Author

Felipe Dias

4.54.2 Constructor & Destructor Documentation

4.54.2.1 MIPS::PassNotAInstruction::PassNotAInstruction ([bit8_t opcode](#), [Register * rs](#), [Register * rt](#), [bit8_t shamt](#), [bit8_t funct](#)) [\[inline\]](#)

Constroi uma nova instrução.

4.54.3 Member Function Documentation

4.54.3.1 [bit16_t](#) MIPS::PassNotAInstruction::execute () [\[virtual\]](#)

Função que executa a operação de soma.

Returns

resultado da operação

Implements [MIPS::InstructionI](#).

The documentation for this class was generated from the following file:

- [include/mips/instructions/format_l/passnota.hpp](#)

4.55 MIPS::Queue< T > Class Template Reference

```
#include <queue.hpp>
```

Classes

- struct [Node](#)

Public Member Functions

- [Queue](#) ()
- [~Queue](#) ()
- void [push](#) (T &elem)
- T [pop](#) ()
- T [get](#) (size_t pos)
- size_t [size](#) ()

4.55.1 Detailed Description

```
template<typename T>
class MIPS::Queue< T >
```

Fila genérica que utiliza template para auxiliar em seu reuso.

Parameters

<i>Matheus</i>	Nogueira
----------------	----------

4.55.2 Constructor & Destructor Documentation

4.55.2.1 `template<typename T> MIPS::Queue< T >::Queue ()` `[inline]`

Cria uma nova fila encadeada. < Cabeça da fila

< Número de elementos da fila

4.55.2.2 `template<typename T> MIPS::Queue< T >::~~Queue ()` `[inline]`

Destroi a fila e todos os seus elementos.

4.55.3 Member Function Documentation

4.55.3.1 `template<typename T> T MIPS::Queue< T >::get (size_t pos)` `[inline]`

Recupera o elemento que está na posição especificada da fila.

Parameters

<i>pos</i>	posição do elemento na fila.
------------	------------------------------

Returns

elemento na posição especificada.

4.55.3.2 `template<typename T> T MIPS::Queue< T >::pop () [inline]`

Retira um elemento da fila.

Returns

elemento no início da fila.

4.55.3.3 `template<typename T> void MIPS::Queue< T >::push (T & elem) [inline]`

Adiciona um elemento à fila.

Parameters

<i>elem</i>	elemento que será inserido na fila.
-------------	-------------------------------------

4.55.3.4 `template<typename T> size_t MIPS::Queue< T >::size () [inline]`

Retorna o tamanho da fila.

Returns

tamanho atual da fila.

The documentation for this class was generated from the following file:

- [include/mips/util/structure/queue.hpp](#)

4.56 MIPS::Register Class Reference

```
#include <register.hpp>
```

Public Member Functions

- [Register](#) (const char *name, bool protect=false)
- [~Register](#) ()
- void [put](#) (bit16_t value)
- [bit16_t](#) [get](#) ()
- const char * [getName](#) ()

4.56.1 Detailed Description

Classe que representa um registrador de 16 bits.

Author

Matheus Nogueira

4.56.2 Constructor & Destructor Documentation

4.56.2.1 MIPS::Register::Register (const char * *name*, bool *protect* = false)

Cria um novo registrador.

Parameters

<i>name</i>	nome do registrador.
<i>protect</i>	indica se o registrador é protegido para escrita.

4.56.2.2 MIPS::Register::~~Register ()

Destroi o registrador.

4.56.3 Member Function Documentation

4.56.3.1 bit16_t MIPS::Register::get ()

Pega o valor 16 bits armazenado no registrador.

Returns

valor armazenado no registrador.

4.56.3.2 const char* MIPS::Register::getName ()

Retorna o nome do registrador.

Returns

nome do registrador.

4.56.3.3 void MIPS::Register::put (bit16_t *value*)

Define um valor que o registrador irá guardar.

Parameters

<i>value</i>	valor 16 bits que será armazenado no registrador.
--------------	---------------------------------------------------

The documentation for this class was generated from the following file:

- [include/mips/memory/register.hpp](#)

4.57 MIPS::RegisterBank Class Reference

```
#include <register_bank.hpp>
```

Public Member Functions

- [RegisterBank](#) ()
- [~RegisterBank](#) ()
- [Register](#) * [getRegister](#) ([bit8_t](#) id)
- [Register](#) * [getPC](#) ()
- void [write](#) ([bit16_t](#) result, [bit8_t](#) rd)

4.57.1 Detailed Description

Classe que representa o banco de registradores do MIPS. Esta é responsável por gerenciar os registradores do processador.

Author

Matheus Nogueira

4.57.2 Constructor & Destructor Documentation

4.57.2.1 MIPS::RegisterBank::RegisterBank ()

Cria uma instância do banco de registradores.

4.57.2.2 MIPS::RegisterBank::~~RegisterBank ()

Destroi a instância do banco de registradores.

4.57.3 Member Function Documentation

4.57.3.1 Register* MIPS::RegisterBank::getPC ()

Retorna o ponteiro para o registrador contador de programa.

Returns

ponteiro para o program counter

4.57.3.2 Register* MIPS::RegisterBank::getRegister ([bit8_t](#) id)

Retorna um ponteiro para o registrador identificado pelo código especificado.

Parameters

<i>id</i>	código do registrador.
-----------	------------------------

Returns

ponteiro para o registrador.

4.57.3.3 void MIPS::RegisterBank::write (bit16_t result, bit8_t rd)

Escreve o valor de um registrador usando o seu índice para localizá-lo

Parameters

<i>result</i>	novo valor do registrador
<i>rd</i>	registrador de destino

The documentation for this class was generated from the following file:

- [include/mips/memory/register_bank.hpp](#)

4.58 MIPS::SignalExtender Class Reference

```
#include <signal_extender.hpp>
```

Static Public Member Functions

- static [bit16_t extend](#) (bit16_t num, bit8_t bits=8)

4.58.1 Detailed Description

Circuito responsável por estender um sinal para 16 bits.

Author

Matheus Nogueira

4.58.2 Member Function Documentation

4.58.2.1 static bit16_t MIPS::SignalExtender::extend (bit16_t num, bit8_t bits = 8) [static]

Estende o sinal de um número.

Parameters

<i>num</i>	número que será extendido.
<i>bits</i>	número de bits do número de entrada. (Padrão: 8)

Returns

número de 16 bits

The documentation for this class was generated from the following file:

- [include/mips/circuits/signal_extender.hpp](#)

4.59 MIPS::SignalInversor Class Reference

```
#include <signal_inversor.hpp>
```

Public Member Functions

- [SignalInversor](#) ()
- [bit16_t invert](#) (bit16_t num)

4.59.1 Detailed Description

Classe responsável por realizar inversão de sinal de 16 bits.

Author

Matheus Nogueira

4.59.2 Constructor & Destructor Documentation

4.59.2.1 MIPS::SignalInversor::SignalInversor ()

Cria um novo somador.

4.59.3 Member Function Documentation

4.59.3.1 bit16_t MIPS::SignalInversor::invert (bit16_t num)

Inverte o sinal de um número.

Parameters

<i>num</i>	número que terá seu sinal invertido.
------------	--------------------------------------

Returns

número com o sinal invertido.

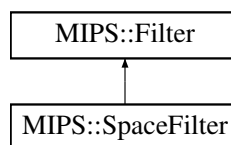
The documentation for this class was generated from the following file:

- [include/mips/circuits/signal_inversor.hpp](#)

4.60 MIPS::SpaceFilter Class Reference

```
#include <space_filter.hpp>
```

Inheritance diagram for MIPS::SpaceFilter:

**Public Member Functions**

- `std::string filter (std::string &input)`

4.60.1 Detailed Description

Classe responsável por remover espaços e tabulações de um texto.

Author

Matheus Matheus Nogueira

4.60.2 Member Function Documentation

4.60.2.1 `std::string MIPS::SpaceFilter::filter (std::string & input)` [virtual]

Remove todos os espaços e tabulações de uma string.

Parameters

<i>input</i>	string a ser filtrada.
--------------	------------------------

Returns

string filtrada.

Implements [MIPS::Filter](#).

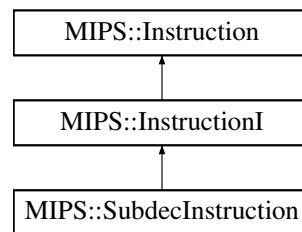
The documentation for this class was generated from the following file:

- [include/mips/util/filter/space_filter.hpp](#)

4.61 MIPS::SubdecInstruction Class Reference

```
#include <subdec.hpp>
```

Inheritance diagram for MIPS::SubdecInstruction:

**Public Member Functions**

- [SubdecInstruction](#) ([bit8_t opcode](#), [Register *rs](#), [Register *rt](#), [bit8_t shamt](#), [bit8_t funct](#))
- [bit16_t execute](#) ()

Additional Inherited Members

4.61.1 Detailed Description

Classe que faz a operação de subdec no processador.

Author

Lucas Pereira

4.61.2 Constructor & Destructor Documentation

4.61.2.1 MIPS::SubdecInstruction::SubdecInstruction ([bit8_t opcode](#), [Register * rs](#), [Register * rt](#), [bit8_t shamt](#), [bit8_t funct](#)) [\[inline\]](#)

Constroi uma nova instrução.

4.61.3 Member Function Documentation

4.61.3.1 `bit16_t MIPS::SubdecInstruction::execute () [virtual]`

Função que executa a operação de subtração.

Implements [MIPS::InstructionI](#).

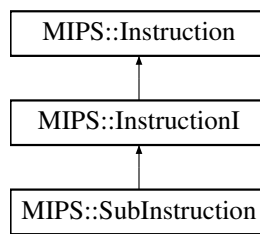
The documentation for this class was generated from the following file:

- `include/mips/instructions/format_l/subdec.hpp`

4.62 MIPS::SubInstruction Class Reference

```
#include <sub.hpp>
```

Inheritance diagram for MIPS::SubInstruction:



Public Member Functions

- [SubInstruction](#) (`bit8_t opcode`, `Register *rs`, `Register *rt`, `bit8_t shamt`, `bit8_t funct`)
- [bit16_t execute](#) ()

Additional Inherited Members

4.62.1 Detailed Description

Classe que faz a operação de ADD no processador.

Author

Felipe Dias

4.62.2 Constructor & Destructor Documentation

4.62.2.1 `MIPS::SubInstruction::SubInstruction (bit8_t opcode, Register * rs, Register * rt, bit8_t shamt, bit8_t funct) [inline]`

Constroi uma nova instrução.

4.62.3 Member Function Documentation

4.62.3.1 `bit16_t MIPS::SubInstruction::execute () [virtual]`

Função que executa a operação de soma.

Returns

resultado da operação

Implements [MIPS::InstructionI](#).

The documentation for this class was generated from the following file:

- `include/mips/instructions/format_l/sub.hpp`

4.63 MIPS::Tokenizer Class Reference

```
#include <tokenizer.hpp>
```

Public Member Functions

- `void tokenize (char *str, std::vector< char * > &vector)`

4.63.1 Detailed Description

Classe responsável por quebrar uma string em diversos tokens.

Author

Matheus Nogueira

4.63.2 Member Function Documentation

4.63.2.1 `void MIPS::Tokenizer::tokenize (char * str, std::vector< char * > & vector)`

Retira os tokens de uma string e os armazena em um vector.

Parameters

<i>str</i>	string contendo os tokens
<i>vector</i>	vector que será usado para armazenar os tokens

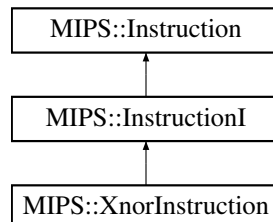
The documentation for this class was generated from the following file:

- [include/mips/interpreter/parser/tokenizer.hpp](#)

4.64 MIPS::XnorInstruction Class Reference

```
#include <xnor.hpp>
```

Inheritance diagram for MIPS::XnorInstruction:



Public Member Functions

- [XnorInstruction](#) ([bit8_t opcode](#), [Register *rs](#), [Register *rt](#), [bit8_t shamt](#), [bit8_t funct](#))
- [bit16_t execute](#) ()

Additional Inherited Members

4.64.1 Detailed Description

Classe que faz a operação de XNOR no processador.

Author

Matheus Nogueira

4.64.2 Constructor & Destructor Documentation

4.64.2.1 `MIPS::XnorInstruction::XnorInstruction (bit8_t opcode, Register * rs, Register * rt, bit8_t shamt, bit8_t funct) [inline]`

Constroi uma nova instrução.

4.64.3 Member Function Documentation

4.64.3.1 `bit16_t MIPS::XnorInstruction::execute () [virtual]`

Função que executa a operação de soma.

Returns

resultado da operação

Implements [MIPS::InstructionI](#).

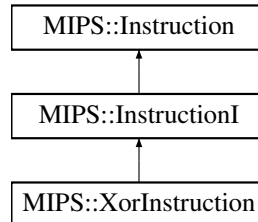
The documentation for this class was generated from the following file:

- [include/mips/instructions/format_l/xnor.hpp](#)

4.65 MIPS::XorInstruction Class Reference

```
#include <xor.hpp>
```

Inheritance diagram for MIPS::XorInstruction:



Public Member Functions

- [XorInstruction](#) ([bit8_t opcode](#), [Register *rs](#), [Register *rt](#), [bit8_t shamt](#), [bit8_t funct](#))
- [bit16_t execute](#) ()

Additional Inherited Members

4.65.1 Detailed Description

Classe que faz a operação de XOR no processador.

Author

Felipe Dias

4.65.2 Constructor & Destructor Documentation

4.65.2.1 `MIPS::XorInstruction::XorInstruction (bit8_t opcode, Register * rs, Register * rt, bit8_t shamt, bit8_t funct)`
`[inline]`

Constroi uma nova instrução.

4.65.3 Member Function Documentation

4.65.3.1 `bit16_t MIPS::XorInstruction::execute ()` `[virtual]`

Função que executa a operação de soma.

Returns

resultado da operação

Implements [MIPS::InstructionI](#).

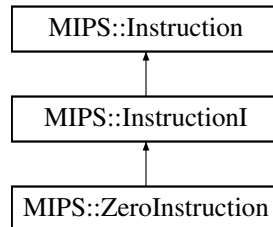
The documentation for this class was generated from the following file:

- `include/mips/instructions/format_l/xor.hpp`

4.66 MIPS::ZeroInstruction Class Reference

```
#include <zero.hpp>
```

Inheritance diagram for MIPS::ZeroInstruction:



Public Member Functions

- [ZeroInstruction](#) ([bit8_t opcode](#), [Register *rs](#), [Register *rt](#), [bit8_t shamt](#), [bit8_t funct](#))
- [bit16_t execute](#) ()

Additional Inherited Members

4.66.1 Detailed Description

Classe que faz a operação de ZERO no processador.

Author

Felipe Dias

4.66.2 Constructor & Destructor Documentation

4.66.2.1 MIPS::ZeroInstruction::ZeroInstruction ([bit8_t opcode](#), [Register * rs](#), [Register * rt](#), [bit8_t shamt](#), [bit8_t funct](#)) [inline]

Constroi uma nova instrução.

4.66.3 Member Function Documentation

4.66.3.1 [bit16_t](#) MIPS::ZeroInstruction::execute () [virtual]

Função que executa a operação de soma.

Returns

resultado da operação

Implements [MIPS::InstructionI](#).

The documentation for this class was generated from the following file:

- [include/mips/instructions/format_l/zero.hpp](#)

Chapter 5

File Documentation

5.1 include/mips/circuits/full_adder.hpp File Reference

```
#include <mips/core.hpp>
```

Classes

- class [MIPS::FullAdder](#)

5.1.1 Detailed Description

Somador de 16 bits. Referencia: <http://isweb.redwoods.edu/instruct/calderwoodd/diglogic/full.↵htm>

5.2 include/mips/circuits/signal_extender.hpp File Reference

```
#include <mips/core.hpp>
```

Classes

- class [MIPS::SignalExtender](#)

5.2.1 Detailed Description

Extensor de sinal. Converte um número para 16 bits.

5.3 include/mips/circuits/signal_inversor.hpp File Reference

```
#include <mips/core.hpp>
```

Classes

- class [MIPS::SignalInversor](#)

5.3.1 Detailed Description

Inversor de sinal em números que utilizam a representação de complemento de 2.

5.4 include/mips/core.hpp File Reference

```
#include <mips/debug.hpp>
```

Typedefs

- typedef signed char [MIPS::bit8_t](#)
- typedef signed short [MIPS::bit16_t](#)
- typedef signed int [MIPS::bit32_t](#)
- typedef bit16_t [MIPS::instruction_t](#)

5.4.1 Detailed Description

Arquivo que contém tipos utilizados pelo emulaodr.

5.4.2 Typedef Documentation

5.4.2.1 typedef signed short [MIPS::bit16_t](#)

Tipo que representa um inteiro de 16 bits.

5.4.2.2 typedef signed int [MIPS::bit32_t](#)

Tipo que representa um inteiro de 16 bits.

5.4.2.3 typedef signed char [MIPS::bit8_t](#)

Tipo que representa um inteiro de 8 bits.

5.4.2.4 typedef bit16_t MIPS::instruction_t

Tipo que representa uma instrução de 16 bits.

5.5 include/mips/cpu.hpp File Reference

```
#include <mips/memory/register_bank.hpp>
#include <mips/memory/memory.hpp>
#include <mips/units/instruction_finder.hpp>
#include <mips/units/control.hpp>
#include <mips/decoder/instruction_decoder.hpp>
```

Classes

- class [MIPS::CPU](#)

5.5.1 Detailed Description

Arquivo contendo uma estrutura que representa a CPU m-RISC.

5.6 include/mips/debug.hpp File Reference

```
#include <iostream>
#include <cstdio>
#include <mips/core.hpp>
```

Macros

- #define [MESSAGE](#)(arg)
- #define [DEBUG](#)(arg)
- #define [FORMAT_DEBUG](#)(format, ...)
- #define [PRINT_BIN](#)(num)

5.6.1 Detailed Description

Arquivo que contém funções para auxiliar o processo de DEBUG da execução do emulador.

5.6.2 Macro Definition Documentation

5.6.2.1 #define DEBUG(arg)

Value:

```
{
    std::cout << "[DEBUG] " << arg << std::endl;
}
```

Macro para escrita de mensagens de debug com o prefixo [DEBUG].

Parameters

<i>arg</i>	mensagem
------------	----------

5.6.2.2 #define FORMAT_DEBUG(*format*, ...)

Value:

```
{
    printf(format, __VA_ARGS__);
}
```

Macro para escrita de uma string formatada no modo debug.

Parameters

<i>format</i>	formato da string
...	variant args

5.6.2.3 #define MESSAGE(*arg*)

Value:

```
{
    std::cout << arg << std::endl;
}
```

Macro para escrita de mensagens em modo de debug

Parameters

<i>arg</i>	mensagem
------------	----------

5.6.2.4 #define PRINT_BIN(*num*)

Value:

```
{
    char x09878412bin = 0;
    char i = 15;
    for (; i >= 0; --i) {
        x09878412bin = (num >> i) & 1;
        printf("%d", x09878412bin);
        if (i % 4 == 0)
            printf(" ");
    }
    printf("\n");
}
```

Macro para imprimir um número no formato binário.

Parameters

<i>num</i>	número que será impresso
------------	--------------------------

5.7 include/mips/decoder/instruction_decoder.hpp File Reference

```
#include <mips/core.hpp>
#include <mips/instructions/instruction.hpp>
#include <mips/memory/register_bank.hpp>
```

Classes

- class [MIPS::InstructionDecoder](#)

5.7.1 Detailed Description

Arquivo contendo um decodificador de instruções para a arquitetura MIPS de 16 bits.

5.8 include/mips/flag.hpp File Reference

```
#include <mips/core.hpp>
```

Classes

- struct [MIPS::ALUFlags](#)

5.8.1 Detailed Description

Arquivo contendo uma estrutura de dados que armazena as flags de saída da ALU.

5.9 include/mips/instructions/format_l/add.hpp File Reference

```
#include <mips/instructions/instruction_I.hpp>
```

Classes

- class [MIPS::AddInstruction](#)

5.9.1 Detailed Description

Declaração da instrução de ADD.

Declaração da instrução de ASR.

Declaração da instrução de DECA.

Declaração da instrução de INCA.

Declaração da instrução de SUB.

5.10 `include/mips/instructions/format_l/addinc.hpp` File Reference

```
#include <mips/instructions/instruction_I.hpp>
```

Classes

- class [MIPS::AddIncInstruction](#)

5.10.1 Detailed Description

Declaração da instrução de ADDINC.

5.11 `include/mips/instructions/format_l/and.hpp` File Reference

```
#include <mips/instructions/instruction_I.hpp>
```

Classes

- class [MIPS::AndInstruction](#)

5.11.1 Detailed Description

Declaração da instrução de andnota.

5.12 `include/mips/instructions/format_l/andnota.hpp` File Reference

```
#include <mips/instructions/instruction_I.hpp>
```

Classes

- class [MIPS::AndnotaInstruction](#)

5.12.1 Detailed Description

Declaração da instrução de andnota.

5.13 include/mips/instructions/format_l/nand.hpp File Reference

```
#include <mips/instructions/instruction_I.hpp>
```

Classes

- class [MIPS::NandInstruction](#)

5.13.1 Detailed Description

Declaração da instrução de NAND.

5.14 include/mips/instructions/format_l/nor.hpp File Reference

```
#include <mips/instructions/instruction_I.hpp>
```

Classes

- class [MIPS::NorInstruction](#)

5.14.1 Detailed Description

Declaração da instrução de NOR.

5.15 include/mips/instructions/format_l/ones.hpp File Reference

```
#include <mips/instructions/instruction_I.hpp>
```

Classes

- class [MIPS::OnesInstruction](#)

5.15.1 Detailed Description

Declaração da instrução de ONES.

5.16 include/mips/instructions/format_l/or.hpp File Reference

```
#include <mips/instructions/instruction_I.hpp>
```

Classes

- class [MIPS::OrInstruction](#)

5.16.1 Detailed Description

Declaração da instrução de or.

5.17 include/mips/instructions/format_l/ornotb.hpp File Reference

```
#include <mips/instructions/instruction_I.hpp>
```

Classes

- class [MIPS::OrnotbInstruction](#)

5.17.1 Detailed Description

Declaração da instrução de ornotb.

5.18 include/mips/instructions/format_l/passa.hpp File Reference

```
#include <mips/instructions/instruction_I.hpp>
```


Classes

- class [MIPS::PassaInstruction](#)

5.18.1 Detailed Description

Declaração da instrução de PASSA.

5.19 include/mips/instructions/format_l/passnota.hpp File Reference

```
#include <mips/instructions/instruction_I.hpp>
```

Classes

- class [MIPS::PassNotAInstruction](#)

5.19.1 Detailed Description

Declaração da instrução de PASSNOTA.

5.20 include/mips/instructions/format_l/subdec.hpp File Reference

```
#include <mips/instructions/instruction_I.hpp>
```

Classes

- class [MIPS::SubdeclInstruction](#)

5.20.1 Detailed Description

Declaração da instrução de SubDec.

5.21 include/mips/instructions/format_l/xnor.hpp File Reference

```
#include <mips/instructions/instruction_I.hpp>
```

Classes

- class [MIPS::XnorInstruction](#)

5.21.1 Detailed Description

Declaração da instrução de XNOR.

5.22 `include/mips/instructions/format_I/xor.hpp` File Reference

```
#include <mips/instructions/instruction_I.hpp>
```

Classes

- class [MIPS::XorInstruction](#)

5.22.1 Detailed Description

Declaração da instrução de XOR.

5.23 `include/mips/instructions/format_I/zero.hpp` File Reference

```
#include <mips/instructions/instruction_I.hpp>
```

Classes

- class [MIPS::ZeroInstruction](#)

5.23.1 Detailed Description

Declaração da instrução de ZERO.

5.24 `include/mips/instructions/format_II/loadlit.hpp` File Reference

```
#include <mips/instructions/instruction_II.hpp>
```

Classes

- class [MIPS::LoadlitInstruction](#)

5.24.1 Detailed Description

Instrução que carrega uma constante com sinal em um registrador.

5.25 include/mips/instructions/format_III/lch.hpp File Reference

```
#include <mips/instructions/instruction_III.hpp>
```

Classes

- class [MIPS::LchInstruction](#)

5.25.1 Detailed Description

Instrução que carrega uma constante com sinal no bit mais significativo do registrador.

Instrução que faz o desvio quando a flag zero da alu é igual a 1.

5.26 include/mips/instructions/format_III/lcl.hpp File Reference

```
#include <mips/instructions/instruction_III.hpp>
```

Classes

- class [MIPS::LclInstruction](#)

5.26.1 Detailed Description

Instrução que carrega uma constante com sinal no bit menos significativo do registrador.

5.27 include/mips/instructions/instruction.hpp File Reference

```
#include <mips/core.hpp>
#include <mips/units/control.hpp>
```

Classes

- class [MIPS::Instruction](#)

5.27.1 Detailed Description

Arquivo contendo a estrutura abstrata que representa uma instrução qualquer em uma arquitetura de 16 bits.

5.28 include/mips/instructions/instruction_I.hpp File Reference

```
#include <mips/core.hpp>
#include <mips/instructions/instruction.hpp>
#include <mips/memory/register.hpp>
```

Classes

- class [MIPS::InstructionI](#)

5.28.1 Detailed Description

Arquivo contendo uma classe que representa uma instrução do tipo (R)egister.

5.29 include/mips/instructions/instruction_II.hpp File Reference

```
#include <mips/core.hpp>
#include <mips/instructions/instruction.hpp>
#include <mips/memory/register.hpp>
```

Classes

- class [MIPS::InstructionII](#)

5.29.1 Detailed Description

Arquivo descrevendo o formato de uma instrução do formato II.

5.30 include/mips/instructions/instruction_III.hpp File Reference

```
#include <mips/core.hpp>
#include <mips/instructions/instruction.hpp>
#include <mips/memory/register.hpp>
```

Classes

- class [MIPS::InstructionIII](#)

5.30.1 Detailed Description

Arquivo descrevendo o formato de uma instrução do formato III.

5.31 include/mips/instructions/instruction_IV.hpp File Reference

```
#include <mips/core.hpp>
#include <mips/flag.hpp>
#include <mips/instructions/instruction.hpp>
#include <mips/memory/register.hpp>
```

Classes

- class [MIPS::InstructionIV](#)

5.31.1 Detailed Description

Arquivo descrevendo o formato de uma instrução do formato IV.

5.32 include/mips/instructions/instruction_V.hpp File Reference

```
#include <mips/core.hpp>
#include <mips/flag.hpp>
#include <mips/instructions/instruction.hpp>
#include <mips/memory/register.hpp>
```

Classes

- class [MIPS::InstructionV](#)

5.32.1 Detailed Description

Arquivo descrevendo o formato de uma instrução do formato V.

5.33 include/mips/instructions/instruction_VI.hpp File Reference

```
#include <mips/core.hpp>
#include <mips/flag.hpp>
#include <mips/instructions/instruction.hpp>
#include <mips/memory/register.hpp>
```

Classes

- class [MIPS::InstructionVI](#)

5.33.1 Detailed Description

Arquivo descrevendo o formato de uma instrução do formato VI.

5.34 include/mips/instructions/instruction_VII.hpp File Reference

```
#include <mips/core.hpp>
#include <mips/flag.hpp>
#include <mips/instructions/instruction.hpp>
#include <mips/memory/memory.hpp>
#include <mips/memory/register.hpp>
```

Classes

- class [MIPS::InstructionVII](#)

5.34.1 Detailed Description

Arquivo descrevendo o formato de uma instrução do formato VII.

5.35 include/mips/interpreter/encoder/encoder.hpp File Reference

```
#include <mips/core.hpp>
#include <mips/interpreter/label.hpp>
#include <map>
#include <vector>
```

Classes

- class [MIPS::Encoder](#)

5.35.1 Detailed Description

Arquivo contendo um codificador genérico para instruções MIPS 16 bits.

5.36 include/mips/interpreter/encoder/encoder_factory.hpp File Reference

```
#include <mips/interpreter/encoder/encoder.hpp>
#include <mips/interpreter/encoder/format_I_encoder.hpp>
#include <mips/interpreter/encoder/format_II_encoder.hpp>
#include <mips/interpreter/encoder/format_III_encoder.hpp>
#include <mips/interpreter/encoder/format_IV_encoder.hpp>
#include <mips/interpreter/encoder/format_V_encoder.hpp>
#include <mips/interpreter/encoder/format_VI_encoder.hpp>
#include <mips/interpreter/encoder/format_VII_encoder.hpp>
```

Classes

- class [MIPS::EncoderFactory](#)

5.36.1 Detailed Description

Arquivo contendo uma fábrica de codificadores de instruções.

5.37 include/mips/interpreter/encoder/format_I_encoder.hpp File Reference

```
#include <mips/core.hpp>
#include <mips/interpreter/encoder/encoder.hpp>
#include <vector>
```

Classes

- class [MIPS::FormatIEncoder](#)

5.37.1 Detailed Description

Arquivo contendo o codificador de instruções do formato I.

5.38 include/mips/interpreter/encoder/format_II_encoder.hpp File Reference

```
#include <mips/interpreter/encoder/encoder.hpp>
#include <vector>
```

Classes

- class [MIPS::FormatIIEncoder](#)

5.38.1 Detailed Description

Arquivo contendo o codificador de instruções do formato II.

5.39 `include/mips/interpreter/encoder/format_III_encoder.hpp` File Reference

```
#include <mips/interpreter/encoder/encoder.hpp>
#include <vector>
```

Classes

- class [MIPS::FormatIIIEncoder](#)

5.39.1 Detailed Description

Arquivo contendo o codificador de instruções do formato III.

5.40 `include/mips/interpreter/encoder/format_IV_encoder.hpp` File Reference

```
#include <mips/interpreter/encoder/encoder.hpp>
#include <vector>
```

Classes

- class [MIPS::FormatIVEncoder](#)

5.40.1 Detailed Description

Arquivo contendo o codificador de instruções do formato IV.

5.41 `include/mips/interpreter/encoder/format_V_encoder.hpp` File Reference

```
#include <mips/interpreter/encoder/encoder.hpp>
#include <vector>
```


Classes

- class [MIPS::FormatVEncoder](#)

5.41.1 Detailed Description

Arquivo contendo o codificador de instruções do formato V.

5.42 include/mips/interpreter/encoder/format_VI_encoder.hpp File Reference

```
#include <mips/interpreter/encoder/encoder.hpp>
#include <vector>
```

Classes

- class [MIPS::FormatVIEncoder](#)

5.42.1 Detailed Description

Arquivo contendo o codificador de instruções do formato VI.

5.43 include/mips/interpreter/encoder/format_VII_encoder.hpp File Reference

```
#include <mips/interpreter/encoder/encoder.hpp>
#include <vector>
```

Classes

- class [MIPS::FormatVIIEncoder](#)

5.43.1 Detailed Description

Arquivo contendo o codificador de instruções do formato VII.

5.44 include/mips/interpreter/exception/interpreter_exception.hpp File Reference

```
#include <exception>
#include <stdexcept>
#include <mips/core.hpp>
```

Classes

- class [MIPS::InterpreterException](#)

5.44.1 Detailed Description

Arquivo contendo uma exceção que é lançada pelo interpretador de assembly.

5.45 include/mips/interpreter/interpreter.hpp File Reference

```
#include <mips/core.hpp>
#include <mips/interpreter/label.hpp>
#include <mips/util/file_reader.hpp>
#include <iostream>
```

Classes

- class [MIPS::Interpreter](#)

5.45.1 Detailed Description

Arquivo contendo a declaração do interpretador de instruções MIPS 32.

5.46 include/mips/interpreter/label.hpp File Reference

Classes

- struct [MIPS::Label](#)

5.46.1 Detailed Description

Arquivo contendo uma estrutura que armazena um label do código assembly.

5.47 include/mips/interpreter/parser/tokenizer.hpp File Reference

```
#include <vector>
```

Classes

- class [MIPS::Tokenizer](#)

5.47.1 Detailed Description

Arquivo contendo a classe responsável por separar uma string em tokens.

5.48 include/mips/memory/memory.hpp File Reference

```
#include <mips/core.hpp>
```

Classes

- class [MIPS::Memory](#)

5.48.1 Detailed Description

Arquivo contendo a estrutura responsável por encapsular a memória de instruções e a memória de dados utilizada pelo processador.

5.49 include/mips/memory/memory_exception.hpp File Reference

```
#include <exception>
#include <stdexcept>
#include <mips/core.hpp>
```

Classes

- class [MIPS::MemoryException](#)

5.49.1 Detailed Description

Arquivo contendo uma exceção que é lançada pelo interpretador de assembly.

5.50 include/mips/memory/register.hpp File Reference

```
#include <mips/core.hpp>
```

Classes

- class [MIPS::Register](#)

5.50.1 Detailed Description

Arquivo contendo a definição de um registrador.

5.51 include/mips/memory/register_bank.hpp File Reference

```
#include <mips/core.hpp>
#include <mips/memory/register.hpp>
```

Classes

- class [MIPS::RegisterBank](#)

5.51.1 Detailed Description

Arquivo contendo o banco de registradores contido no processador MIPS.

5.52 include/mips/units/control.hpp File Reference

Classes

- class [MIPS::ControlUnit](#)

5.52.1 Detailed Description

Arquivo contendo a unidade de controle do processador.

5.53 include/mips/units/instruction_finder.hpp File Reference

```
#include <mips/core.hpp>
#include <mips/memory/memory.hpp>
#include <mips/memory/register_bank.hpp>
```

Classes

- class [MIPS::InstructionFinder](#)

5.53.1 Detailed Description

Arquivo contendo o buscador de instruções do processador.

5.54 include/mips/util/event/event.hpp File Reference

Classes

- class [MIPS::Event](#)

Enumerations

- enum [MIPS::EventType](#) { [MIPS::REGISTER_UPDATE](#), [TEST](#) }

5.54.1 Detailed Description

Arquivo contendo um evento base que é despachavel pelo despachante de eventos do emulador de MIPS.

5.54.2 Enumeration Type Documentation

5.54.2.1 enum MIPS::EventType

Enum utilizado para listar todos os tipos de eventos suportados pelo emulador MIPS.

Enumerator

REGISTER_UPDATE Indica que um registrador foi alterado.

5.55 include/mips/util/event/event_dispatcher.hpp File Reference

```
#include <mips/util/structure/queue.hpp>
#include <mips/util/event/event_listener.hpp>
#include <mips/util/event/event.hpp>
#include <vector>
```

Classes

- class [MIPS::EventDispatcher](#)
- struct [MIPS::EventDispatcher::ListenerMap](#)

5.55.1 Detailed Description

Arquivo contendo o despachante de eventos utilizado por diversos componentes do núcleo do emulador MIPS. Esse despachante é responsável por criar uma interface de despacho de eventos para outros componentes, dessa forma, um componente pode comunicar com outro componente usando troca de mensagens, onde estas são transmitidas via eventos.

5.56 include/mips/util/event/event_listener.hpp File Reference

```
#include <mips/util/event/event.hpp>
```

Classes

- class [MIPS::EventListener](#)

5.56.1 Detailed Description

Arquivo contendo a classe abstrata a qual deve ser pai de todos os objetos que devem ouvir um evento de outro objeto.

5.57 include/mips/util/file_reader.hpp File Reference

```
#include <mips/util/filter/filter.hpp>
#include <fstream>
#include <vector>
```

Classes

- class [MIPS::FileReader](#)

5.57.1 Detailed Description

Arquivo contendo uma classe responsável por ler e limpar o conteúdo de um arquivo do disco.

5.58 include/mips/util/filter/filter.hpp File Reference

```
#include <string>
```

Classes

- class [MIPS::Filter](#)

5.58.1 Detailed Description

Arquivo contendo um filtro de string abstrato.

5.59 include/mips/util/filter/space_filter.hpp File Reference

```
#include <mips/util/filter/filter.hpp>
```

Classes

- class [MIPS::SpaceFilter](#)

5.59.1 Detailed Description

Arquivo contendo um filtro que retira tabulações e espaços em branco desnecessários de um texto.

5.60 include/mips/util/structure/queue.hpp File Reference

```
#include <cstdlib>
```

Classes

- class [MIPS::Queue< T >](#)
- struct [MIPS::Queue< T >::Node](#)

5.60.1 Detailed Description

Arquivo contendo a implementação de uma fila encadeada.

