XHAL

1

Generated by Doxygen 1.6.1

Thu Feb 9 10:50:49 2017

Contents

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

File Index

2.1 File List

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

Chapter 3

Class Documentation

3.1 wisc::RPCMsg::BadKeyException Class Reference

Public Member Functions

- **BadKeyException** (std::string key)

Public Attributes

- const std::string **key**

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

- /home/mdalchen/work/xhal/include/xhal/rpc/wiscRPCMsg.h

3.2 wisc::RPCMsg::BufferTooSmallException Class Reference

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

- /home/mdalchen/work/xhal/include/xhal/rpc/wiscRPCMsg.h

3.3 wisc::RPCMsg::CorruptMessageException Class Reference

Public Member Functions

- **CorruptMessageException** (std::string reason)

Public Attributes

- const std::string **reason**

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

- /home/mdalchen/work/xhal/include/xhal/rpc/wiscRPCMsg.h

3.4 xhal::utils::Node Class Reference

stores single XML node with its attributes

```
#include <XHALXMLNode.h>
```

Public Member Functions

- Node ()

  Default constructor. Creates empty Node.

- void addChild (Node child)

  Adds child node.

- std::string getVhdlName ()

  Returns VHDL node name.

- void output ()

  Not implemented.

- void getAllChildren (Node node, std::vector< Node > kids)

  Returns all hierarchy of chlid nodes.

Public Attributes

- std::string **name**
- std::string **description**
- std::string **vhdlname**
- uint32_t **address**
- uint32_t **real_address**
- std::string **permission**
- uint32_t **mask**
- bool **isModule**
- Node ∗ **parent**
- std::vector< Node > **children**
- int **level**
- int **warn_min_value**
- int **error_min_value**

3.4.1 Detailed Description

stores single XML node with its attributes Note that all the class members are public in order to avoid extra ambiguitization of the code with getters and setters

3.4.2 Member Function Documentation

3.4.2.1 void xhal::utils::Node::getAllChildren (Node *node*, std::vector< Node > *kids*) `[inline]`

Returns all hierarchy of chlid nodes.

Parameters:

node parent node

kids vector of nodes, must be empty when function called and will be updated with node childrem

3.4.2.2 void xhal::utils::Node::output () `[inline]`

Not implemented.

TODO

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

- /home/mdalchen/work/xhal/include/xhal/utils/XHALXMLNode.h

3.5 wisc::RPCSvc::RPCException Class Reference

Public Member Functions

- **RPCException** (std::string message)

Public Attributes

- std::string **message**

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

- /home/mdalchen/work/xhal/include/xhal/rpc/wiscrpcsvc.h

3.6 wisc::RPCMsg Class Reference

Classes

- class BadKeyException
- class BufferTooSmallException
- class CorruptMessageException
- class TypeException

Public Member Functions

- **RPCMsg** (std::string method_name)
- **RPCMsg** (void ∗serial_data, uint32_t datalen)
- **RPCMsg** (const RPCMsg &msg)
- RPCMsg & **operator=** (const RPCMsg &other)
- std::string **serialize** () const
- std::string **get_method** () const
- RPCMsg & **set_method** (std::string value)
- bool **get_key_exists** (std::string key) const
- std::string **get_string** (std::string key) const
- RPCMsg & **set_string** (std::string key, std::string value)
- uint32_t **get_string_array_size** (std::string key) const
- std::vector< std::string > **get_string_array** (std::string key) const
- RPCMsg & **set_string_array** (std::string key, std::vector< std::string > value)
- uint32_t **get_word** (std::string key) const
- RPCMsg & **set_word** (std::string key, uint32_t value)
- uint32_t **get_word_array_size** (std::string key) const
- void **get_word_array** (std::string key, uint32_t ∗data) const
- RPCMsg & **set_word_array** (std::string key, uint32_t ∗data, int count)
- std::vector< uint32_t > **get_word_array** (std::string key) const
- RPCMsg & **set_word_array** (std::string key, const std::vector< uint32_t > &data)
- uint32_t **get_binarydata_size** (std::string key) const
- void **get_binarydata** (std::string key, void ∗data, uint32_t bufsize) const
- RPCMsg & **set_binarydata** (std::string key, const void ∗data, uint32_t bufsize)

Static Public Attributes

- static const char **key_characters** [ ]

Protected Attributes

- RPCMsgProto::RPCMsg ∗ **buf**

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

- /home/mdalchen/work/xhal/include/xhal/rpc/wiscRPCMsg.h

3.7 wisc::RPCSvc Class Reference

Classes

- class RPCException

Public Member Functions

- **__LIBRPCSVC_EXCEPTION** (NotConnectedException, RPCException)
- **__LIBRPCSVC_EXCEPTION** (ConnectionFailedException, RPCException)
- **__LIBRPCSVC_EXCEPTION** (RPCErrorException, RPCException)
- void **connect** (std::string host, uint16_t port)
- void **connect** (std::string host)
- void **disconnect** ()
- bool **load_module** (std::string module, std::string module_version_key)
- RPCMsg **call_method** (const RPCMsg &reqmsg)

Protected Attributes

- int **fd**

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

- /home/mdalchen/work/xhal/include/xhal/rpc/wiscrpcsvc.h

3.8 wisc::RPCMsg::TypeException Class Reference

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

- /home/mdalchen/work/xhal/include/xhal/rpc/wiscRPCMsg.h

3.9 xhal::XHALInterface Class Reference

provide interface to call remote procedures at Zynq CPU and basic FW registers manipulation

`#include <XHALInterface.h>`

Public Member Functions

- XHALInterface (const std::string &board_domain_name, const std::string &address_table_filename)

  Default constructor.

- void init ()

  Initialize interface and establish RPC service connection with CTP7.

- void loadModule (const std::string &module_name, const std::string &module_version)

  load remote module

- void setLogLevel (int loglevel)

  sets amount of logging/debugging information to display

- uint32_t readReg (std::string regName)

  read FW register by its name applies reading mask if any

- uint32_t readReg (uint32_t address)

  read FW register by its address

3.9.1 Detailed Description

provide interface to call remote procedures at Zynq CPU and basic FW registers manipulation

3.9.2 Constructor & Destructor Documentation

3.9.2.1 xhal::XHALInterface::XHALInterface (const std::string & *board_domain_name*, const std::string & *address_table_filename*)

Default constructor.

Parameters:

board_domain_name domain name of CTP7

address_table_filename XML address table file name

3.9.3 Member Function Documentation

3.9.3.1 void xhal::XHALInterface::init ()

Initialize interface and establish RPC service connection with CTP7. Parses XML file and starts the RPCSvc connection

3.9.3.2 void xhal::XHALInterface::setLogLevel (int *loglevel*)

sets amount of logging/debugging information to display

Parameters:

loglevel,: 0 - ERROR 1 - WARN 2 - INFO 3 - DEBUG 4 - TRACE

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

- /home/mdalchen/work/xhal/include/xhal/XHALInterface.h

3.10 xhal::utils::XHALXMLParser Class Reference

provide parsing interface and search through flattened node tree

```
#include <XHALXMLParser.h>
```

Public Member Functions

- XHALXMLParser (const std::string &xmlFile)

  Default constructor.

- void setLogLevel (int loglevel)

  sets amount of logging/debugging information to display

- void parseXML ()

  parses XML file and creates flattened nodes tree

- std::experimental::optional< xhal::utils::Node > getNode (const char ∗nodeName)

  returns node object by its name or nothing if name is not found

- std::experimental::optional< xhal::utils::Node > getNodeFromAddress (const uint32_t nodeAddress)

  not implemented

3.10.1 Detailed Description

provide parsing interface and search through flattened node tree

3.10.2 Constructor & Destructor Documentation

3.10.2.1 xhal::utils::XHALXMLParser::XHALXMLParser (const std::string & *xmlFile*)

Default constructor.

Parameters:

xmlFile address table file name

3.10.3 Member Function Documentation

3.10.3.1 void xhal::utils::XHALXMLParser::setLogLevel (int *loglevel*)

sets amount of logging/debugging information to display

Parameters:

loglevel,: 0 - ERROR 1 - WARN 2 - INFO 3 - DEBUG 4 - TRACE

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

- /home/mdalchen/work/xhal/include/xhal/utils/XHALXMLParser.h

Chapter 4

File Documentation

4.1 /home/mdalchen/work/xhal/include/xhal/utils/Exception.h File Reference

`#include <string>`

`#include <exception>`

Defines

- #define **XHAL_UTILS_DEFINE_EXCEPTION**(EXCEPTION_NAME)

4.1.1 Detailed Description

XHAL exception base class

Author:

Mykhailo Dalchenko

Version:

1.0

4.1.2 Define Documentation

4.1.2.1 #define XHAL_UTILS_DEFINE_EXCEPTION(EXCEPTION_NAME)

Value:

```
namespace xhal {                                                        \
  namespace utils {                                                     \
    class EXCEPTION_NAME : public std::exception {                      \
      public:                                                           \
        EXCEPTION_NAME(const char* message) : m(message) {              \
                                                                        \
        }                                                               \
                                                                        \
        virtual ~EXCEPTION_NAME() throw() {                              \
                                                                         \
        }                                                                \
                                                                         \
        virtual const char* what() const throw() {                       \
            return m.c_str();                                            \
        }                                                                \
                                                                         \
      protected:                                                         \
        std::string m;                                                   \
                                                                         \
      private:                                                           \
        EXCEPTION_NAME();                                                \
    };                                                                   \
  } /* namespace xhal::utils */                                          \
}
```

4.2 /home/mdalchen/work/xhal/include/xhal/utils/XHALXMLParser.h File Reference

```
#include <string>
#include <iostream>
#include <unordered_map>
#include <experimental/optional>
#include <boost/algorithm/string.hpp>
#include <boost/lexical_cast.hpp>
#include <boost/format.hpp>
#include <xercesc/util/PlatformUtils.hpp>
#include <xercesc/util/XMLString.hpp>
#include <xercesc/dom/DOM.hpp>
#include <xercesc/dom/DOMElement.hpp>
#include <xercesc/parsers/XercesDOMParser.hpp>
#include <xercesc/util/OutOfMemoryException.hpp>
#include <xercesc/framework/XMLFormatter.hpp>
#include <xercesc/framework/LocalFileFormatTarget.hpp>
#include <xercesc/dom/DOMDocument.hpp>
#include <xercesc/dom/DOMImplementation.hpp>
#include <xercesc/dom/DOMImplementationRegistry.hpp>
#include <xercesc/dom/DOMLSSerializer.hpp>
#include <xercesc/dom/DOMLSOutput.hpp>
#include "log4cplus/logger.h"
#include "log4cplus/loglevel.h"
#include "log4cplus/loggingmacros.h"
#include "log4cplus/consoleappender.h"
#include "xhal/utils/XHALXMLNode.h"
#include "xhal/utils/Exception.h"
```

Classes

- class xhal::utils::XHALXMLParser

  provide parsing interface and search through flattened node tree

Defines

- #define **TRANSCODE**(STR) xercesc::XMLString::transcode(STR)

- #define **TRACE**(MSG) LOG4CPLUS_TRACE(m_logger, MSG)
- #define **DEBUG**(MSG) LOG4CPLUS_DEBUG(m_logger, MSG)
- #define **INFO**(MSG) LOG4CPLUS_INFO(m_logger, MSG)
- #define **WARN**(MSG) LOG4CPLUS_WARN(m_logger, MSG)
- #define **ERROR**(MSG) LOG4CPLUS_ERROR(m_logger, MSG)
- #define **FATAL**(MSG) LOG4CPLUS_FATAL(m_logger, MSG)

4.2.1 Detailed Description

XML parser for XHAL library. Parses the XML address table and store results in unordered map of (regName, Node)

Author:

Mykhailo Dalchenko

Version:

1.0

4.3 /home/mdalchen/work/xhal/include/xhal/XHALInterface.h File Reference

```
#include <string>
#include "xhal/rpc/wiscrpcsvc.h"
#include "xhal/utils/XHALXMLParser.h"
#include "xhal/utils/Exception.h"
```

Classes

- class xhal::XHALInterface

  provide interface to call remote procedures at Zynq CPU and basic FW registers manipulation

Defines

- #define **STANDARD_CATCH**
- #define **ASSERT**(x)

4.3.1 Detailed Description

Hardware interface for XHAL

Author:

Mykhailo Dalchenko

Version:

1.0

4.3.2 Define Documentation

4.3.2.1 #define ASSERT(x)

Value:

```
do { \
                if (!(x)) { \
                        printf("Assertion Failed on line %u: %s\n", __LINE__, #x)
      ; \
      throw xhal::utils::Exception("ASSERT failure");\
                } \
        } while (0)
```

4.3.2.2 #define STANDARD_CATCH

Value:

```
catch (wisc::RPCSvc::NotConnectedException &e) { \
                ERROR("Caught NotConnectedException: " << e.message.c_str()); \
    throw xhal::utils::Exception(strcat("RPC exception: ", e.message.c_str()));\
        } \
        catch (wisc::RPCSvc::RPCErrorException &e) { \
                ERROR("Caught RPCErrorException: " << e.message.c_str()); \
    throw xhal::utils::Exception(strcat("RPC exception: ", e.message.c_str()));\
        } \
        catch (wisc::RPCSvc::RPCException &e) { \
                ERROR("Caught exception: " << e.message.c_str()); \
    throw xhal::utils::Exception(strcat("RPC exception: ", e.message.c_str()));\
        } \
  catch (wisc::RPCMsg::BadKeyException &e) { \
    ERROR("Caught exception: " << e.key.c_str()); \
    throw xhal::utils::Exception(strcat("RPC exception (most probably remote regi
      ster not accessible): ", e.key.c_str()));\
        }
```

Index