rtl Documentation

Release v1.7

Marko Kosunen

Nov 04, 2021

CONTENTS:

1 RTL IOfile module 5

2 Testbench 7

3 Module 9

4 Entity 11

5 Indices and tables 13

Python Module Index 15

Index 17

Simulation interface package for The System Development Kit

Provides utilities to import verilog modules and VHDL entities to python environment and sutomatically generate testbenches for the most common simulation cases.

Initially written by Marko Kosunen, 2017

class rtl.**rtl**

Adding this class as a superclass enforces the definitions for rtl simulations in the subclasses.

connect_inputs()

Assigns all IOS.Members[name].Data to self.iofile_bundle.Members[ioname].Data

connect_outputs()

Connects the ouput data from files to corresponding output IOs

create_connectors()

Cretes verilog connector definitions from 1) From a iofile that is provided in the Data attribute of an IO. 2) IOS of the verilog DUT

execute_rtl_sim()

Runs the rtl simulation in external simulator

format_ios()

Verilog module does not contain information if the bus is signed or not Prior to writing output file, the type of the connecting wire defines how the bus values are interpreted.

property interactive_control_contents

Content of the interactive rtl control file (.do -file).

If this property is set, a new dofile gets written to the simulation path. This takes precedence over the file pointed by *interactive_controlfile*.

For example, the contents can be defined in the top testbench as:

```
self.interactive_control_contents="""
    add wave -position insertpoint \
    sim/:tb_inverter:A \
    sim/:tb_inverter:clock \
    sim/:tb_inverter:Z
    run -all
    wave zoom full
"""
```

property interactive_controlfile

Path to interactive rtl control file (.do -file).

The content of the file can be defined in *interactive_control*. If the content is not set in *interactive_control* -property, the do-file is read from this file path. Default path is *./dofile.do*.

property interactive_rtl

True | False (default)

Launch simulator in local machine with GUI.

property iofile_bundle

Property of type thesdk.Bundle. This property utilises iofile class to maintain list of IO-files that are automatically handled by simulator specific commands when verilog.rtl_iofile.rtl_iofile(name='<filename>,...) is used to define an IO-file, created file object is automatically appended to this Bundle property as a member. Accessible with self.iofile_bundle.Members['<filename>']

property name
Name of the entity Extracted from the _classfile attribute

property preserve_iofiles
True | False (default)

If True, do not delete file IO files after simulations. Useful for debugging the file IO

read_outfile()
Reads the oputput files

property rtlcmd
Command used for simulation invocation Compiled from various parameters. See source for details.

property rtlmisc
List<String>

List of manual commands to be pasted to the testbench. The strings are pasted to their own lines (no linebreaks needed), and the syntax is unchanged.

Example: creating a custm clock:

```
self.rtlmisc = []
self.rtlmisc.append('reg clock2;')
self.rtlmisc.append('initial clock2='b0;')
self.rtlmisc.append('always #(c_Ts2/2.0) clock2 = !clock2;')
```

property rtlparameters
Dictionary of parameters passed to the simulator during the simulation invocation

property rtlworkpath
Work library directory for rtl compilations self.simpath +'/work'

Returns

Return type self.simpath +'/work'

run_rtl()

1) Creates testbench
2) Defines the contens of the testbench
3) Creates connectors
4) Connects inputs
5) Defines IO conditions
6) Defines IO formats in testbench
7) Generates testbench contents
8) Exports the testbench to file
9) Writes input files
10) Executes the simulation
11) Read outputfiles
12) Connects the outputs

You should overload this method while creating the simulation and debugging the testbench.

property verilog_submission
Defines verilog submioddion prefix from thesdk.GLOBALS['LSFSUBMISSION']

Usually something like 'bsub -K'

property vhdlentityfiles
List of VHDL entity files to be compiled in addiotion to DUT

property vhdlsrc
VHDL source file self.vhdlsrcpath/self.name.sv'

Returns

Return type self.vhdlsrcpath + '/' + self.name + '.vhd'

property vhdlsrcpath
VHDL search path self.entitypath/vhdl

Returns

Return type self.entitypath/vhdl

property vlogmodulefiles
List of verilog modules to be compiled in addition of DUT

property vlogsrc
Verilog source file self.vlogsrcpath/self.name.sv'

Returns

Return type self.vlogsrcpath + '/' + self.name + '.sv'

property vlogsrcpath
Search path for the verilogfiles self.entitypath/sv

Returns

Return type self.entitypath/sv

property vlogtbsrc
Verilog testbench source file

write_infile()
Writes the input files

CHAPTER

ONE

RTL IOFILE MODULE

Provides verilog- file-io related attributes and methods for TheSyDeKick RTL intereface.

Initially written by Marko Kosunen, marko.kosunen@aalto.fi, Yue Dai, 2018

class rtl.rtl_iofile.**rtl_iofile**(*parent=None*, ***kwargs*)

Class to provide file IO for verilog simulations. When created, adds a rtl_iofile object to the parents iofile_bundle attribute. Accessible as self.iofile_bundle.Members['name'].

Provides methods and attributes that can be used to construct sections in Verilog testbenches, like file io routines, file open and close routines, file io routines, file io format strings and read/write conditions.

Example

Initiated in parent as: _=rtl_iofile(self,name='foobar')

Parameters

- **parent** (*object*) – The parent object initializing the rtl_iofile instance. Default None
- ****kwargs** –

  name [str] Name of the file. Appended with random string during the simulation.

  param [str, -g *'g_file_'*] The string defining the testbench parameter to be be passed to the simulator at command line. Sets the paramname attribute.

  ioformat [str, %d] sets the ioformat attribute.

property Data

Data value of this IO

property DictData

connector_datamap(***kwargs*)

Verilog_connectors is an ordered list. Order defines the assumed order of columns in the file to be read or written. This datamap provides {'name' : index } dictionary to assing data to correct columns. Less use for data files, more for controls

property file

Name of the IO file to be read or written.

property ioformat

Formatting string for verilog file reading Default %d, i.e. content of the file is single column of integers.

property rtlparam

Extracts the parameter name and value from simparam attribute. Used to construct the parameter definitions for Verilog testbench.

Default {'g_file_<self.name>', self.file }

set_control_data(***kwargs*)
Method to define event based data value with name, time, and value. Uses a python dictionary instead of a numpy array for more efficient insertions. The 'time' column acts as the dictionary key, the remaining columns are stored as the value.

Parameters ****kwargs** – time: int, 0 name: str val: type undefined init: int, 0 vector of values to initialize the data. lenght should correpond to *self.verilog_connectors+1*

property simparam
String definition for parameter to be passed to the simulator as a command line argument

property verilog_connectors
List for verilog connectors. These are the verilog signals/regs associated with this file

property verilog_ctstamp
Current time stamp variable name to be used in verilog testbench. Used in event type file IO.

property verilog_fclose
Verilog file close routine sting.

property verilog_fopen
Verilog file open routine string.

property verilog_fptr
Verilog file pointer name.

property verilog_io
Verilog write/read construct for file IO depending on the direction and file type (event/sample).

Returns Verilog code for file IO to read/write the IO file.

Return type str

property verilog_io_condition
Verilog condition string that must be true in ordedr to file IO read/write to occur.

Default for output file: *~$isunknown(connector.name)* for all connectors of the file. Default for input file: *'1'* file is always read with rising edge of the clock or in the time of an event defined in the file.

verilog_io_condition_append(***kwargs*)
Append new condition string to *verilog_io_condition*

Parameters ****kwargs** – cond : str

property verilog_io_sync
File io synchronization condition for sample type input. Default: *@(posedge clock)*

property verilog_ptstamp
Past time stamp variable for verilog testbench. Used in event type file IO.

property verilog_stat
Status variable name to be used in verilog testbench.

property verilog_statdef
Verilog file read status integer variable definitions and initializations strings.

property verilog_tdiff
Verilog time differencec variable. Used in event based file IO. '

CHAPTER

TWO

TESTBENCH

Verilog testbench utility mmodule for TheSyDeKick. Contains attributes annd methods to import DUT as *verilog_module* instance, parse its IO and parameter definitions and construct a structured testbench with clock and file IO.

Utilizes logging method from thesdk. Extendsd *verilog_module* with additional properties.

Initially written by Marko Kosunen 20190108, marko.kosunen@aalto.fi

class rtl.testbench.**testbench**(*parent=None*, ***kwargs*)

Testbench class. Extends *verilog_module*

Parameters

- **parent** (*object, None (mandatory to define) TheSyDeKick parent entity object for this testbench.*) –
- ****kwargs** – None

assignments(***kwargs*)

Wire assingment strings

property clock_definition

Clock definition string

Todo Create append mechanism to add more clocks.

connect_inputs()

Define connections to DUT inputs.

property connector_definitions

Verilog register and wire definition strings

define_testbench()

Defines the tb connectivity, creates reset and clock, and initializes them to zero

property dut_instance

RTL module parsed from the verilog or VHDL file of the parent depending on *parent.model*

property file

Path to the testbench file

Default: *self.parent.vlogsrcpath + '/tb_' + self.parent.name + '.sv'*

generate_contents()

This is the method to generate testbench contents. Override if needed Contents of the testbench is constructed from attributes in the following order

```
   self.parameter_definitions
   self.connector_definitions
   self.assignments()
   self.iofile_definitions
   sefl.misccmd
   self.dut_instance.instance
   self.verilog_instance_members.items().instance (for all members)
   self.connectors.verilog_inits()
   self.iofiles.Members.items().verilog_io (for all members)
   self.iofile.close (for all members)

Addtional code may be currently injected by appending desired
strings (Verilog sytax) to the relevant string attributes.
```

property iofile_close
File close procedure for all IO files.

property iofile_definitions
IOfile definition strings

property misccmd
String

Miscellaneous command string corresponding to self.rtlmisc -list in the parent entity.

property parameter_definitions
Parameter and variable definition strins of the testbench

verilog_instance_add(***kwargs*)
Add verilog instance to the Bundle fro a file

Parameters ****kwargs** –

name [str] name of the module

file : File defining the module

property verilog_instances
Verilog instances Bundle to be added to tesbench

Todo Need to handle VHDL instance too.

CHAPTER

THREE

MODULE

Verilog module import features for RTL simulation package of The System Development Kit.

Provides utilities to import Verilog modules to python environment.

Initially written by Marko Kosunen, 2017 Last modification by Okko Järvinen, 28.10.2021 07:55

class rtl.module.**verilog_module**(***kwargs*)

Objective:

1) a) Collect IO's to database

   b) collect parameters to dict

2) Reconstruct the module definition

3) a) Implement methods provide sinal connections

   b) Implement methods to provide parameter assingments

4) Create a method to create assigned module definition, where signals are a) assigned by name b) to arbitrary name vector.

5) Add contents, if required, and include that to definition

Parameters ****kwargs** –

file: str Verilog file containing the module

name: str Name of the module

instname: str, self.name Name of the instance

property contents

Contents of the module. String containing the Verilog code after the module definition.

property definition

Module definition part extracted for the file. Contains parameters and IO definitions.

export(***kwargs*)

Method to export the module to a given file.

Parameters ****kwargs** – force: Bool

property instance

Instantioation string of the module. Can be used inside of the other modules.

property instname

Name of the instance, when instantiated inside other module.

Default: *self.name_DUT*

property io_signals
Bundle containing the signal connectors for IO connections.

property ios
Verilog connector bundle containing connectors for all module IOS. All the IOs are connected to signal connectors that have the same name than the IOs. This is due to fact the we have decided that all signals are connectors.

property name
Name of the module. Derived from the file name.

property parameters
Parameters of the verilog module. Bundle of values of type string.

CHAPTER

FOUR

ENTITY

VHDL import features for RTL simulation package of The System Development Kit

Provides utilities to import VHDL entities to python environment. Imported VHDL entities will be instantiated as verilog modules, and are intended to be simulated within verilog testbench with simulator supporting cross language compilations.

Initially written by Marko Kosunen, 2017

Transferred from VHL package in Dec 2019

Last modification by Marko Kosunen, marko.kosunen@aalto.fi, 21.01.2020 19:27

class rtl.entity.**vhdl_entity**(***kwargs*)

Executes init of verilog_module, thus having the same attributes and parameters.

Parameters ****kwargs** – See module module

property contents

Contents is extracted. We do not know what to do with it yet.

property ios

Verilog connector bundle containing connectors for all module IOS. All the IOs are connected to signal connectors that have the same name than the IOs. This is due to fact the we have decided that all signals are connectors.

property parameters

Verilog parameters extracted from generics of the VHDL entity

CHAPTER

FIVE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

r

rtl, 1
rtl.entity, 10
rtl.module, 8
rtl.rtl_iofile, 3
rtl.testbench, 6

INDEX

A

assignments() (*rtl.testbench.testbench method*), 7

C

clock_definition (*rtl.testbench.testbench property*), 7
connect_inputs() (*rtl.rtl method*), 1
connect_inputs() (*rtl.testbench.testbench method*), 7
connect_outputs() (*rtl.rtl method*), 1
connector_datamap() (*rtl.rtl_iofile.rtl_iofile method*), 5
connector_definitions (*rtl.testbench.testbench property*), 7
contents (*rtl.entity.vhdl_entity property*), 11
contents (*rtl.module.verilog_module property*), 9
create_connectors() (*rtl.rtl method*), 1

D

Data (*rtl.rtl_iofile.rtl_iofile property*), 5
define_testbench() (*rtl.testbench.testbench method*), 7
definition (*rtl.module.verilog_module property*), 9
DictData (*rtl.rtl_iofile.rtl_iofile property*), 5
dut_instance (*rtl.testbench.testbench property*), 7

E

execute_rtl_sim() (*rtl.rtl method*), 1
export() (*rtl.module.verilog_module method*), 9

F

file (*rtl.rtl_iofile.rtl_iofile property*), 5
file (*rtl.testbench.testbench property*), 7
format_ios() (*rtl.rtl method*), 1

G

generate_contents() (*rtl.testbench.testbench method*), 7

I

instance (*rtl.module.verilog_module property*), 9
instname (*rtl.module.verilog_module property*), 9
interactive_control_contents (*rtl.rtl property*), 1
interactive_controlfile (*rtl.rtl property*), 1
interactive_rtl (*rtl.rtl property*), 1
io_signals (*rtl.module.verilog_module property*), 9
iofile_bundle (*rtl.rtl property*), 1
iofile_close (*rtl.testbench.testbench property*), 8
iofile_definitions (*rtl.testbench.testbench property*), 8
ioformat (*rtl.rtl_iofile.rtl_iofile property*), 5
ios (*rtl.entity.vhdl_entity property*), 11
ios (*rtl.module.verilog_module property*), 10

M

misccmd (*rtl.testbench.testbench property*), 8
module
    rtl, 1
    rtl.entity, 10
    rtl.module, 8
    rtl.rtl_iofile, 3
    rtl.testbench, 6

N

name (*rtl.module.verilog_module property*), 10
name (*rtl.rtl property*), 1

P

parameter_definitions (*rtl.testbench.testbench property*), 8
parameters (*rtl.entity.vhdl_entity property*), 11
parameters (*rtl.module.verilog_module property*), 10
preserve_iofiles (*rtl.rtl property*), 2

R

read_outfile() (*rtl.rtl method*), 2
rtl
    module, 1
rtl (*class in rtl*), 1
rtl.entity
    module, 10
rtl.module
    module, 8
rtl.rtl_iofile
    module, 3

rtl.testbench
    module, 6
rtl_iofile (*class in rtl.rtl_iofile*), 5
rtlcmd (*rtl.rtl property*), 2
rtlmisc (*rtl.rtl property*), 2
rtlparam (*rtl.rtl_iofile.rtl_iofile property*), 5
rtlparameters (*rtl.rtl property*), 2
rtlworkpath (*rtl.rtl property*), 2
run_rtl() (*rtl.rtl method*), 2

S

set_control_data() (*rtl.rtl_iofile.rtl_iofile method*), 6
simparam (*rtl.rtl_iofile.rtl_iofile property*), 6

T

testbench (*class in rtl.testbench*), 7

V

verilog_connectors (*rtl.rtl_iofile.rtl_iofile property*), 6
verilog_ctstamp (*rtl.rtl_iofile.rtl_iofile property*), 6
verilog_fclose (*rtl.rtl_iofile.rtl_iofile property*), 6
verilog_fopen (*rtl.rtl_iofile.rtl_iofile property*), 6
verilog_fptr (*rtl.rtl_iofile.rtl_iofile property*), 6
verilog_instance_add() (*rtl.testbench.testbench method*), 8
verilog_instances (*rtl.testbench.testbench property*), 8
verilog_io (*rtl.rtl_iofile.rtl_iofile property*), 6
verilog_io_condition (*rtl.rtl_iofile.rtl_iofile property*), 6
verilog_io_condition_append() (*rtl.rtl_iofile.rtl_iofile method*), 6
verilog_io_sync (*rtl.rtl_iofile.rtl_iofile property*), 6
verilog_module (*class in rtl.module*), 9
verilog_ptstamp (*rtl.rtl_iofile.rtl_iofile property*), 6
verilog_stat (*rtl.rtl_iofile.rtl_iofile property*), 6
verilog_statdef (*rtl.rtl_iofile.rtl_iofile property*), 6
verilog_submission (*rtl.rtl property*), 2
verilog_tdiff (*rtl.rtl_iofile.rtl_iofile property*), 6
vhdl_entity (*class in rtl.entity*), 11
vhdlentityfiles (*rtl.rtl property*), 3
vhdlsrc (*rtl.rtl property*), 3
vhdlsrcpath (*rtl.rtl property*), 3
vlogmodulefiles (*rtl.rtl property*), 3
vlogsrc (*rtl.rtl property*), 3
vlogsrcpath (*rtl.rtl property*), 3
vlogtbsrc (*rtl.rtl property*), 3

W

write_infile() (*rtl.rtl method*), 3