

Rule Checker Features

ZamiaCad

Table of distribution

CNES	Name and function	Contact
	Gabriel LIABEUF Project Leader	gabriel.liabeuf@cnes.fr
	Florent MANNI Technical Responsible	florent.manni@cnes.fr
ALTRAN	Name and function	Contact
	Christophe NIESNER Consultant	christophe.niesner@altran.com
	Arnaud Daniel Project Leader	arnaud.daniel@altran.com +33 6 29 68 27 52

Revision History

Date	Version	Description	Author
03 august 2015	1.0	Document creation	Arnaud DANIEL Christophe NIESNER
03 september 2015	1.1	update with latest rules checker GUI	Arnaud DANIEL Christophe NIESNER
11 september 2015	1.2	Version number aligned on delivered Rules Checker version. No modification done inside document.	Arnaud DANIEL
21 september 2015	1.3	Document renaming in Features Description Addition of a glossary to explain log parameters Move of SRS_REQ_STD_01800 rule from Tools section to Rules Selector section Addition of “Detailed Description” and “Limitations” for each tool and rule.	Arnaud DANIEL

04 december 2015	2.1	Update with WP-CLK and WP-RST.	Arnaud DANIEL Christophe NIESNER

Table of contents

Contenu

Table of distribution	2
Revision History	2
Table of contents	4
A. Scope	6
1. Identification.....	6
B. Mentioned documents	7
1. Reference documents	7
C. Project Objectives	8
D. Preamble.....	9
E. Detailed description of features	9
1. Tools	9
a. Clocks Identification (SRS_REQ_FEAT_FN15).....	9
b. Reset Identification (SRS_REQ_FEAT_FN18).....	13
c. Package/Library Identification (SRS_REQ_FEAT_FN19)	16
d. Line Counter (SRS_REQ_FEAT_FN20)	18
e. Clock Mix Edge (SRS_REQ_FEAT_AR6)	20
f. Reset mix level (SRS_REQ_FEAT_AR7).....	22
g. Clock per project (SRS_REQ_FEAT_CLK_PRJ).....	24
h. Reset per project (SRS_REQ_FEAT_RST_PRJ)	26
2. Rules Selector.....	28
a. Help	28
(1) Primitive Isolation (SRS_REQ_STD_01800)	28
(2) Reset assertion and deassertion (SRS_REQ_STD_03700).....	30
b. Algo.....	32
(1) Use of clock signal (SRS_REQ_CNE_04900).....	32
(2) Preservation of clock name (SRS_REQ_CNE_02300).....	34
(3) Preservation of reset name (SRS_REQ_CNE_02400).....	36

(4) Name of clock signal (SRS_REQ_STD_00200)	38
(5) Name of reset signal (SRS_REQ_STD_00300)	40
(6) Reset sensitivity level (SRS_REQ_STD_03600).....	42
(7) Clock edge sensitivity (SRS_REQ_STD_04800).....	44
(8) Clock reassignment (SRS_REQ_STD_04500)	46
F. Synthesis	48
G. Acronyms and abbreviations.....	49
H. Glossary.....	49

A. Scope

1. Identification

This document aims to describe the features of the Rule Checker add-on of ZamiaCad tool.

B. Mentioned documents

1. Reference documents

Index	Title	Reference	Date
[R1]	ZamiaCad_Rule_Checker_User_Guide_v2.1		04 december 2015

C. Project Objectives

ZamiaCad is a software tool used to help VHDL language users.

Rule Checker add-on has been developed in order to improve the way VHDL code is written and to reduce the time spent while performing code review.

Priority is given to code review.

Most VHDL editors are often not free and just provide syntactic highlighting.

The purpose of ZamiaCad Rule Checker is to cover these limitations and to go further by providing a tool that can be helpful for several VHDL user profiles: project managers, quality supervisors, reviewers, peers, designers.

ZamiaCad Rule Checker is licensed under the GNU Global Public License v3. The GNU General Public License is a free, copyleft license for software and other kinds of works.¹

¹ See <http://opensource.org/licenses/gpl-3.0.html>

D.Preamble

In the following sections, we consider the project “plasma” with default configuration for “XML Files Selection” as described in User Guide [R1].

All report files are created in directories:

- “\$PROJECT_ROOT\rule_checker\reporting\reporting\Algo\rc_report_rule_RuleID_RuleName\” for rules algo
- “\$PROJECT_ROOT\reporting\reporting\Help\rc_report_rule_RuleID_RuleName\” for rules help
- “\$PROJECT_ROOT\reporting\reporting\tool\rc_report_tool_RuleID_RuleName\” for tools.

They are named rc_report_tool_RuleID_RuleName.xml for tools selector execution and rc_report_rule_RuleID_RuleName.xml for rules (algo and help) selector execution.

E. Detailed description of features

1. Tools

a. Clocks Identification (SRS_REQ_FEAT_FN15)

Rationale: The objective is to identify all clock signals in a VLSI project.

Detailed Description: Clocks are detected according to the definition given by the IEEE standard. This covers the following use cases:

The criteria for detecting clock are:

- Clock signal detection is done inside “process”
- The following expressions shall represent a rising edge clock
 - o RISING_EDGE(*clk_signal_name*)
 - o *clk_signal_name* = ‘1’ and *clk_signal_name*’EVENT
 - o *clk_signal_name*’EVENT and *clk_signal_name* = ‘1’
 - o *clk_signal_name* = ‘1’ and not *clk_signal_name*’STABLE
 - o not *clk_signal_name*’STABLE and *clk_signal_name* = ‘1’
- The following expressions shall represent a falling edge clock

- FALLING_EDGE(*clk_signal_name*)
- *clk_signal_name* = '0' and *clk_signal_name*'EVENT
- *clk_signal_name*'EVENT and *clk_signal_name* = '0'
- *clk_signal_name* = '0' and not *clk_signal_name*'STABLE
- not *clk_signal_name*'STABLE and *clk_signal_name* = '0'

The VHDL TOOL shall be able to detect different clocks per process.

Limitations: The VHDL TOOL is only capable to detect clocks inside process. At this stage of progress, it is not possible to detect sources (*e.g* PLL) and sinks (*e.g* gated clock) of those clocks.

Furthermore, the VHDL TOOL does not detect clocks when they are combined with additional combinatory.

In this example the clock signal *clk_A* isn't detected.

```
P_TEST2 : process(clk_A, A_reset, B_reset)
begin
    if A_reset='0' then
        clk_selA_int <= '0';

    elsif falling_edge(clk_A) and B_reset='0' then
        clk_selA_int <= '0';
    end if;
end process;
```

The clocks are attached to process not registers at this stage of progress.

Expected Result:

When we select the feature Clocks Identification, as shown in this figure, the rule checker tool detects the clock signal in corresponding VLSI project as described.

Requirement ID	Requirement Name	Implemented / Not Implemented	Type	Parameter	<input type="checkbox"/> Select All	status	Log file
REQ_FEAT_FN15	Clock Identification	Implemented	Ide	No	<input checked="" type="checkbox"/>	Not executed	
REQ_FEAT_FN18	Reset Identification	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_FN19	Package Library Identification	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_FN20	Line Counter	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_AR6	Clock Mix Edges	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_AR7	Reset Mix Levels	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_CLK_PRJ	Clock Per Project	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_RST_PRJ	Reset Per Project	Implemented	Ide	No	<input type="checkbox"/>	Not executed	

Find Launch Cancel

We can see the result of this detection in rc_report_tool_REQ_FEAT_FN15_ClockIdentification.xml file. In this example, the clock signal “CLK” is detected in file \pc_next.vhd” entity “PC_NEXT” architecture “LOGIC” process “PC_NEXT”.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<REQ_FEAT_FN15>
  <author>rule checker</author>
  <version>V1.3 date 21/09/2015</version>
  <automaticGeneration>YES</automaticGeneration>
  <description>report for rule REQ_FEAT_FN15</description>
  <creationDate>Tue Sep 22 13:45:55 CEST 2015</creationDate>
  <file>
    <fileName>\pc_next.vhd</fileName>
    <nbLine>68</nbLine>
    <entity>
      <entityName>PC_NEXT</entityName>
      <entityLoc>16</entityLoc>
      <architecture>
        <architectureName>LOGIC</architectureName>
        <architectureLoc>28</architectureLoc>
        <process>
          <processName>PC_NEXT</processName>
          <processLoc>34</processLoc>
          <processIsSynchronous>yes</processIsSynchronous>
          <clockSignal>
            <clockSignalName>CLK</clockSignalName>
            <clockSignalLoc>59</clockSignalLoc>
          </clockSignal>
        </process>
      </architecture>
    </entity>
  </file>
</REQ_FEAT_FN15>
```

In this other example, no clock signal are detected in file “\alu.vhd” entity “ALU” architecture “LOGIC” process “ALU_PROC”, we can see “no” to “processIsSynchronous”.

```
<file>
  <fileName>\alu.vhd</fileName>
  <nbLine>71</nbLine>
  <entity>
    <entityName>ALU</entityName>
    <entityLoc>16</entityLoc>
    <architecture>
      <architectureName>LOGIC</architectureName>
      <architectureLoc>23</architectureLoc>
      <process>
        <processName>ALU_PROC</processName>
        <processLoc>31</processLoc>
        <processIsSynchronous>no</processIsSynchronous>
      </process>
    </architecture>
  </entity>
</file>
</REQ_FEAT_FN15>
```

b. Reset Identification (SRS_REQ_FEAT_FN18)

Rationale: The objective is to identify all asynchronous reset signals in a VLSI project.

Detailed Description: Asynchronous reset detection is done according to the following sequence:

- Detect process of type synchronous. Such a process involves a clock signal.
- Identify asynchronous reset by searching any signal declared in condition statement out of the “if...end if” statement involving the clock signal.

Limitations: The resets are attached to process not registers at this stage of progress.

Expected Result:

When we select the feature Reset Identification, as shown in this figure, the rule checker tool detects the reset signals in corresponding VLSI project.

The screenshot shows a software interface titled "Tools Selector" with a sub-section "Ruleset". At the top, there are summary statistics: "nb total 8" and "nb not executed 8". Below this, a table lists eight requirements, each with a status column indicating "Not executed" except for "REQ_FEAT_FN18" which has a checked checkbox. The table includes columns for Requirement ID, Name, Implementation status, Type, Parameter, and Log file.

Requirement ID	Requirement Name	Implemented /Not Implemented	Type	Parameter	Select All	status	Log file
REQ_FEAT_FN15	Clock Identification	Implemented	lde	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_FN18	Reset Identification	Implemented	lde	No	<input checked="" type="checkbox"/>	Not executed	
REQ_FEAT_FN19	Package Library Identification	Implemented	lde	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_FN20	Line Counter	Implemented	lde	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_AR6	Clock Mix Edges	Implemented	lde	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_AR7	Reset Mix Levels	Implemented	lde	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_CLK_PRJ	Clock Per Project	Implemented	lde	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_RST_PRJ	Reset Per Project	Implemented	lde	No	<input type="checkbox"/>	Not executed	

At the bottom of the window, there are "Find" and "Launch" buttons.

We can see the result of this detection in rc_report_tool_REQ_FEAT_FN18_Reset Identification.xml file. In this example, the reset signals “PAUSE_IN” and “RESET_IN”

are detected in file “\pc_next.vhd” entity “PC_NEXT” architecture “LOGIC” process “PC_NEXT” associate to clock signal “CLK”.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<REQ_FEAT_FN18>
  <author>rule checker</author>
  <version>V1.3 date 21/09/2015</version>
  <automaticGeneration>YES</automaticGeneration>
  <description>report for rule REQ_FEAT_FN18</description>
  <creationDate>Tue Sep 22 13:55:27 CEST 2015</creationDate>
  <file>
    <fileName>\pc_next.vhd</fileName>
    <nbLine>68</nbLine>
    <entity>
      <entityName>PC_NEXT</entityName>
      <entityLoc>16</entityLoc>
      <architecture>
        <architectureName>LOGIC</architectureName>
        <architectureLoc>28</architectureLoc>
        <process>
          <processName>PC_NEXT</processName>
          <processLoc>34</processLoc>
          <processIsSynchronous>yes</processIsSynchronous>
          <clockSignal>
            <clockSignalName>CLK</clockSignalName>
            <clockSignalLoc>59</clockSignalLoc>
            <clockSignalHasAsynchronousReset>yes</clockSignalHasAsynchronousReset>
            <resetSignal>
              <resetSignalName>PAUSE_IN</resetSignalName>
              <resetSignalLoc>53</resetSignalLoc>
            </resetSignal>
            <resetSignal>
              <resetSignalName>RESET_IN</resetSignalName>
              <resetSignalLoc>57</resetSignalLoc>
            </resetSignal>
          </clockSignal>
        </process>
      </architecture>
    </entity>
  </file>
```

In this other example, no reset signal are detected in file \mlite2sram.vhd” entity “MLITE2SRAM” architecture “LOGIC” process “SET_STATE” associate to clock signal “CLK”, we can see “no” to “clockSignalHasAsynchronousReset”.

```
<file>
  <fileName>\mlite2sram.vhd</fileName>
  <nbLine>145</nbLine>
  <entity>
    <entityName>MLITE2SRAM</entityName>
    <entityLoc>10</entityLoc>
    <architecture>
      <architectureName>LOGIC</architectureName>
      <architectureLoc>32</architectureLoc>
      <process>
        <processName>SET_STATE</processName>
        <processLoc>46</processLoc>
        <processIsSynchronous>yes</processIsSynchronous>
        <clockSignal>
          <clockSignalName>CLK</clockSignalName>
          <clockSignalLoc>48</clockSignalLoc>
          <clockSignalHasAsynchronousReset>no</clockSignalHasAsynchronousReset>
        </clockSignal>
      </process>
      <process>
        <processName>WORK</processName>
        <processLoc>53</processLoc>
        <processIsSynchronous>no</processIsSynchronous>
      </process>
    </architecture>
  </entity>
</file>
</REQ_FEAT_FN18>
```

c. Package/Library Identification (SRS_REQ_FEAT_FN19)

Rationale: The objective is to make an exhaustive list of declared libraries so that to identify which libraries are standard ones (IEEE) and which are custom ones developed especially for the project.

Detailed Description: The VHDL TOOL shall report the libraries declared in all VHDL files of the VLSI project.

Limitations: None.

Expected Result:

When we select the feature Package/library Identification, as shown in this figure, the rule checker tool detects the libraries used in all vhdl files.

Requirement ID	Requirement Name	Implemented / Not Implemented	Type	Parameter	Select All	status	Log file
REQ_FEAT_FN15	Clock Identification	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_FN18	Reset Identification	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_FN19	Package Library Identification	Implemented	Ide	No	<input checked="" type="checkbox"/>	Not executed	
REQ_FEAT_FN20	Line Counter	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_AR6	Clock Mix Edges	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_AR7	Reset Mix Levels	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_CLK_PRJ	Clock Per Project	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_RST_PRJ	Reset Per Project	Implemented	Ide	No	<input type="checkbox"/>	Not executed	

We can see the result of this detection in rc_report_tool_REQ_FEAT_FN19_Package Library Identification.xml file. In this example, the libraries “IEEE.STD_LOGIC_1164.ALL” and “WORK.MLITE_PACK.ALL” are used in file “\pc_next.vhd”.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<REQ_FEAT_FN19>
  <author>rule checker</author>
  <version>V1.3 date 21/09/2015</version>
  <automaticGeneration>YES</automaticGeneration>
  <description>report for rule REQ_FEAT_FN19</description>
  <creationDate>Tue Sep 22 13:58:49 CEST 2015</creationDate>
  <file>
    <fileName>\pc_next.vhd</fileName>
    <nbLine>68</nbLine>
    <library>
      <libraryName>IEEE.STD_LOGIC_1164.ALL</libraryName>
      <libraryLoc>13</libraryLoc>
    </library>
    <library>
      <libraryName>WORK.MLITE_PACK.ALL</libraryName>
      <libraryLoc>14</libraryLoc>
    </library>
  </file>
</REQ_FEAT_FN19>
```

d. Line Counter (SRS_REQ_FEAT_FN20)

Rationale: The objective is to have information about code complexity by providing the number of lines per VHDL files.

Detailed Description: The VHDL TOOL shall report the number of lines for all VHDL files of the VLSI project. The line number reports the number of lines in the VHDL file until the End Of File (EOF) special character is met.

Limitations: None.

Expected Result:

When we select the feature line Counter, as shown in this figure, the rule checker tool count the lines in all vhdl files.

The screenshot shows a Windows application window titled "Tools Selector". Under the "Ruleset" tab, there is a section for "lde" which includes three counters: "nb total 8", "nb not executed 8", and "nb passed 0". Below this is a detailed table of requirements:

Requirement ID	Requirement Name	Implemented /Not Implemented	Type	Parameter	Select All	status	Log file
REQ_FEAT_FN15	Clock Identification	Implemented	lde	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_FN18	Reset Identification	Implemented	lde	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_FN19	Package Library Identification	Implemented	lde	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_FN20	Line Counter	Implemented	lde	No	<input checked="" type="checkbox"/>	Not executed	
REQ_FEAT_AR6	Clock Mix Edges	Implemented	lde	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_AR7	Reset Mix Levels	Implemented	lde	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_CLK_PRJ	Clock Per Project	Implemented	lde	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_RST_PRJ	Reset Per Project	Implemented	lde	No	<input type="checkbox"/>	Not executed	

At the bottom of the window are buttons for "Find" (with a search bar), "Launch", and "Cancel".

We can see the result in rc_report_tool_REQ_FEAT_FN20_line Counter.xml file. In this example, the file "\pc_next.vhd" has 68 lines.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<REQ_FEAT_FN20>
  <author>rule checker</author>
  <version>V1.3 date 21/09/2015</version>
  <automaticGeneration>YES</automaticGeneration>
  <description>report for rule REQ_FEAT_FN20</description>
  <creationDate>Tue Sep 22 14:00:00 CEST 2015</creationDate>
  <file>
    <fileName>\pc_next.vhd</fileName>
    <nbLine>68</nbLine>
  </file>
  <file>
    <fileName>\mem_ctrl.vhd</fileName>
    <nbLine>243</nbLine>
  </file>
</REQ_FEAT_FN20>
```

e. Clock Mix Edge (SRS_REQ_FEAT_AR6)

Rationale: The objective is to check that there is only one edge detection mechanism used for each clock source.

Detailed Description: Search Clock VHDL elements in VLSI project with “Clock identification” function, and verify that clock are always used with the same edge (falling edge or rising edge).

3 reports are created:

- The most detailed report contains all clocks per entity per process. Each clock is connected to a specific clock source as obtained in SRS_REQ_FEAT_CLK_PRJ. Then, for each clock used in process, the edge is indicated, so it permits to identify on which process a particular clock source is used either on rising or falling edge.
- The second report says for each entity of the VLSI project and for each clock source if it is used on rising edge, falling edge or both edges. It permits to identify in which entity a clock is used on both edges.
- The third report says for every clock source in a VLSI project if it is used exclusively on rising edge, falling edge or both edges.

Limitations: none at this level.

Expected Result:

When we select the feature Clock Mix Edge, as shown in this figure, the rule checker tool says when a clock is used on both edges in all vhdl files.

Tools Selector

Ruleset

nb total 8

Ide

nb total 8 nb not executed 8 nb passed 0

Requirement ID	Requirement Name	Implemented /Not Implemented	Type	Parameter	<input type="checkbox"/> Select All	status	Log file
REQ_FEAT_FN15	Clock Identification	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_FN18	Reset Identification	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_FN19	Package Library Identification	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_FN20	Line Counter	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_AR6	Clock Mix Edges	Implemented	Ide	No	<input checked="" type="checkbox"/>	Not executed	
REQ_FEAT_AR7	Reset Mix Levels	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_CLK_PRJ	Clock Per Project	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_RST_PRJ	Reset Per Project	Implemented	Ide	No	<input type="checkbox"/>	Not executed	

f. Reset mix level (SRS_REQ_FEAT_AR7)

Rationale: The objective is to check that reset signal uses two distinct activation level.

Detailed Description: Search Reset VHDL elements in VLSI project with “Reset identification” function, and verify that the reset signal is used with two distinct activation levels.

3 reports are created:

- The most detailed report contains all reset per entity per process. Each reset is connected to a specific reset source as obtained in SRS_REQ_FEAT_RST_PRJ. Then, for each reset used in process, the level is indicated, so it permits to identify on which process a particular reset source is used either on high or low level.
- The second report says for each entity of the VLSI project and for each reset source if it is used on high level, low level or both levels. It permits to identify in which entity a reset is used on both levels.
- The third report says for every reset source in a VLSI project if it is used exclusively on high level, low level or both levels.

Limitations: none at this level.

Expected Result:

When we select the feature Clock Mix Edge, as shown in this figure, the rule checker tool says when a clock is used on both edges in all vhdl files.

Tools Selector

Ruleset

nb total 8

Ide

nb total 8 nb not executed 8 nb passed 0

Requirement ID	Requirement Name	Implemented /Not Implemented	Type	Parameter	<input type="checkbox"/> Select All	status	Log file
REQ_FEAT_FN15	Clock Identification	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_FN18	Reset Identification	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_FN19	Package Library Identification	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_FN20	Line Counter	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_AR6	Clock Mix Edges	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_AR7	Reset Mix Levels	Implemented	Ide	No	<input checked="" type="checkbox"/>	Not executed	
REQ_FEAT_CLK_PRJ	Clock Per Project	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_RST_PRJ	Reset Per Project	Implemented	Ide	No	<input type="checkbox"/>	Not executed	

Find Launch Cancel

g. Clock per project (SRS_REQ_FEAT_CLK_PRJ)

Rationale: The objective is to list all clocks sources in the project.

Detailed Description: For each clock raised by the clock identification tool (see SRS_REQ_FEAT_FN15), find the clock source. 3 types of clock source can be retrieved:

- Input port : clock source comes from an input port of top of VLSI project

```
entity fpgaicu is
    port(
        Clk_20MHz      : in Std_Logic;      -- input port
        CLK_1HZOBC     : in Std_Logic;      -- input port
        ...
    );
```

- Instance Output : clock source comes from an output of a black box (eg PLL, clock buffer, crypted file)

```
\$1I218\ : CLKINT port map(A => CLK_B, Y => \$1N96\);
```

- Signal assignment : clock source comes from signal assignment : direct assignment, assignment with combinational logic, assignment in process

```
clk1 <= clk; -- direct assignment

clk1 <= clk and not B; -- assignment with combinational logic

process(clk, rst)
begin
    if rst = '0' then
        clk_div <= '0'; -- assignment in process
    elsif rising_edge(clk) then
        clk_div <= NOT clk_div;
    end if;
end process;
```

Each clock source is tagged with the entity name, architecture name, the filename, line number. This creates a list of clock sources.

Limitations: none

Expected Result:

When we select the feature Clock per project, as shown in this figure, the rule checker tool lists all the clock source of a project.

Tools Selector

Ruleset

nb total 8

Ide

nb total 8 nb not executed 8 nb passed 0

Requirement ID	Requirement Name	Implemented /Not Implemented	Type	Parameter	<input type="checkbox"/> Select All	status	Log file
REQ_FEAT_FN15	Clock Identification	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_FN18	Reset Identification	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_FN19	Package Library Identification	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_FN20	Line Counter	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_AR6	Clock Mix Edges	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_AR7	Reset Mix Levels	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_CLK_PRJ	Clock Per Project	Implemented	Ide	No	<input checked="" type="checkbox"/>	Not executed	
REQ_FEAT_RST_PRJ	Reset Per Project	Implemented	Ide	No	<input type="checkbox"/>	Not executed	

Find Launch Cancel

h. Reset per project (SRS_REQ_FEAT_RST_PRJ)

Rationale: The objective is to list all reset sources in the project.

Detailed Description: For each reset issue to the reset identification, find the reset source.

3 types of reset source can be retrieved:

- Input port : reset source comes from an input port of top of VLSI project

```
entity fpgaicu is
    port(
        Reset_n : in Std_Logic;      -- input port
        ...
    );
```

- Instance Output : reset source comes from an output of a black box (eg CLKINT)

```
\$1I218\ : CLKINT port map(A => RESET_B, Y => \$1N96\);
```

- Signal assignment : reset source comes from a signal assignment : direct assignment, assignment with combinational logic, assignment in process

```
reset1 <= reset; -- direct assignment

reset1 <= reset_A and not B; -- assignment with combinational logic

process(clk, rst)
begin
    if rst ='0' then
        rst_resync <= '0'; -- assignment in process
    elsif rising_edge(clk) then
        rst_resync <= '1';
    end if;
end process;
```

Each reset source is tagged with the entity name, architecture name, the filename, line number. This creates a list of reset sources.

Limitations: none at this level.

Expected Result:

When we select the feature Reset per project, as shown in this figure, the rule checker tool lists all the reset source of a project.

Tools Selector

Ruleset

nb total 8

Ide

nb total 8 nb not executed 8 nb passed 0

Requirement ID	Requirement Name	Implemented /Not Implemented	Type	Parameter	<input type="checkbox"/> Select All	status	Log file
REQ_FEAT_FN15	Clock Identification	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_FN18	Reset Identification	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_FN19	Package Library Identification	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_FN20	Line Counter	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_AR6	Clock Mix Edges	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_AR7	Reset Mix Levels	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_CLK_PRJ	Clock Per Project	Implemented	Ide	No	<input type="checkbox"/>	Not executed	
REQ_FEAT_RST_PRJ	Reset Per Project	Implemented	Ide	No	<input checked="" type="checkbox"/>	Not executed	

2. Rules Selector

a. Help

(1) Primitive Isolation (SRS_REQ_STD_01800)

Rationale: The objective is to help reviewers to identify which VHDL files declare libraries other than IEEE (eg ALTERAMF, AXCELERATOR) and verify if concerned files are correctly isolated from the rest of the project.

Detailed Description: The VHDL TOOL shall report the libraries other than IEEE declared in all VHDL files of the VLSI project.

Limitations: None.

Expected Result:

When we select the feature Primitive Isolation, as shown in this figure, the rule checker tool detects the vhdl files used a specific library.

Requirement ID	Requirement Name	Implemented /Not Implemented	Type	Parame...	Select All	status	Log file
CNE_01200	Identification of process label	Implemented	Algo	Yes	<input type="checkbox"/>	Not executed	
STD_01800	Primitive isolation	Implemented	Help	Yes	<input checked="" type="checkbox"/>	Not executed	
CNE_01000	Identification of variable name	Not Impleme...	NA	NA	<input type="checkbox"/>	Not implemented	
CNE_00500	Convention for signal naming	Not Impleme...	NA	NA	<input type="checkbox"/>	Not implemented	
CNE_00600	Convention for constant naming	Not Impleme...	NA	NA	<input type="checkbox"/>	Not implemented	
CNE_00700	Convention for process naming	Not Impleme...	NA	NA	<input type="checkbox"/>	Not implemented	

We can see the result of this detection in rc_report_rule_REQ_FEAT_STD_01800.xml file. In this example, the files “mult.vhd”, “alu_tb.vhd”, ... using “IEEE.STD_LOGIC_UNSIGNED.ALL” library.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<REQ_STD_01800>
  <author>rule checker</author>
  <version>V1.3 date 21/09/2015</version>
  <automaticGeneration>YES</automaticGeneration>
  <description>report for rule REQ_STD_01800</description>
  <creationDate>Tue Sep 22 14:01:05 CEST 2015</creationDate>
  <primitive>
    <libraryName>IEEE.STD_LOGIC_UNSIGNED.ALL</libraryName>
    <fileName>mult.vhd</fileName>
    <fileName>alu_tb.vhd</fileName>
    <fileName>mlite2uart.vhd</fileName>
    <fileName>ram.vhd</fileName>
    <fileName>mlite2sram.vhd</fileName>
    <fileName>mlite_cpu.vhd</fileName>
    <fileName>sram2mlite.vhd</fileName>
    <fileName>reg_bank.vhd</fileName>
    <fileName>cpu_testbench.vhd</fileName>
  </primitive>
```

(2) Reset assertion and deassertion (SRS_REQ_STD_03700)

Rationale: The objective is to return reset sources so that user can check if reset sources are correctly generated.

Detailed Description: with the help of the “SRS_REQ_FEAT_RST_PRJ” function, a report is built containing all reset sources of VLSI project. 3 possible cases may be identified: input port, instance output, assignment.

Limitations: At this level, this is a rule help so it is user responsibility to judge according to reset source report and actual design whether the reset are correctly generated.

Expected Result:

When we select the feature Reset assertion and deassertion, as shown in this figure, the rule checker tool.

Rules Selector							
Ruleset							
nb total 128							
Help							
nb total 2 nb not executed 2 nb reported 0							
Algo							
nb total 9 nb not executed 9 nb passed 0 nb failed 0							
Requirement ID	Requirement Name	Implemented /Not Implemented	Type	Parameter	Select All	status	Log file
STD_01800	Primitive isolation	Implemented	Help	No	<input type="checkbox"/>	Not executed	
STD_03600	Reset sensitive level	Implemented	Algo	No	<input type="checkbox"/>	Not executed	
STD_03700	Reset assertion and deassertion	Implemented	Help	No	<input checked="" type="checkbox"/>	Not executed	
STD_04800	Clock edge sensitivity	Implemented	Algo	No	<input type="checkbox"/>	Not executed	
STD_04500	Clock Reassignment	Implemented	Algo	No	<input type="checkbox"/>	Not executed	

We can see the result of this detection in rc_report_rule_REQ_STD_03700_Reset assertion and deassertion.xml file. In this example, the reset source “Y” is an input port in entity “QCLKDRIVER”.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<STD_03700>
  <author>rule checker</author>
  <automaticGeneration>YES</automaticGeneration>
  <description>report for rule STD_03700</description>
  <ruleName>Reset Assertion and Deassertion</ruleName>
  <creationDate>Fri Nov 27 16:21:28 CET 2015</creationDate>
  <resetSource>
    <fileName>FPGA_FM_Datapackage\02_Code\DPULogic\Definitions\ModuleMap_pkg.vhd</fileName>
    <entityName>QCLKDRIVER</entityName>
    <architectureName>DPULOGIC_STR</architectureName>
    <resetSourceName>Y</resetSourceName>
    <resetSourceType>Input Port</resetSourceType>
    <resetSourceLoc>444</resetSourceLoc>
  </resetSource>
```

b. Algo

(1) Use of clock signal (SRS_REQ_CNE_04900).

Rationale: The objective is to check that clock signals are not used inside combinatory function.

Detailed Description: Search Clock Source elements in VLSI project with “SRS_REQ_FEAT_CLK_PRJ” function. Then, launch for each clock source a search to know all uses of each clock source. Check that the search only returns:

- passages to lower levels of components,
- use in edge detection
- an output port of the FPGA,
- input port of black box (PLL for example)

For all the other cases (logical equation, usage in the synchronous part of a process, etc.), a violation report shall be established with following information for all violated paths:

- clock source
- entity name
- architecture name
- filename
- line number

Limitations: At this level, the reason of violation is not explained. The user has to jump to the corresponding line number and analyses origin of problem.

Expected Result:

When we select the feature Use of clock signal, as shown in this figure, the rule checker tool list clock signal are used inside combinatory function.

Rules Selector

Ruleset

nb total 128

Help

nb total 2 nb not executed 2 nb reported 0

Algo

nb total 9 nb not executed 9 nb passed 0 nb failed 0

Requirement ID	Requirement Name	Implemented /Not Implemented	Type	Parameter	Select All	status	Log file
CNE_01200	Identification of process label	Implemented	Algo	Yes	<input type="checkbox"/>	Not executed	
CNE_04900	Use of clock signal	Implemented	Algo	No	<input checked="" type="checkbox"/>	Not executed	
CNE_02300	Preservation of clock name	Implemented	Algo	No	<input type="checkbox"/>	Not executed	
CNE_02400	Preservation of reset name	Implemented	Algo	No	<input type="checkbox"/>	Not executed	
STD_00200	Name of clock signal	Implemented	Algo	Yes	<input type="checkbox"/>	Not executed	

Find Launch Cancel

We can see the result of this detection in rc_report_rule_REQ_CNE_04900_Use of clock signal.xml file. In this example, the source clock "Y" is used in a logical equation in file "EEPDecoder.vhd" in line 109.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<CNE_04900>
  <author>rule checker</author>
  <automaticGeneration>YES</automaticGeneration>
  <description>violation report for rule CNE_04900</description>
  <ruleName>Use of clock signal</ruleName>
  <creationDate>Fri Nov 27 16:27:48 CET 2015</creationDate>
  <clockSource>
    <violationType>wrong uses clock</violationType>
    <clockSourceName>Y</clockSourceName>
    <fileName>FPGA_FM_Datapackage\02_Code\DPULogic\AddressDecoder\EEPDecoder.vhd</fileName>
    <entityName>EEPDECODER</entityName>
    <architectureName>EEPDECODER_RTL</architectureName>
    <clockSignalLoc>109</clockSignalLoc>
  </clockSource>
```

(2) Preservation of clock name (SRS_REQ_CNE_02300)

Rationale: The objective is to check that clock does not change name when it is connected to components.

Detailed Description: Search Clock VHDL elements in VLSI project with “Clock per project” function, and verify that the signal name does not change when entering components at lower level.

Limitations: none at this level.

Expected Result:

When we select the feature Preservation of clock name, as shown in this figure, the rule checker tool list clock signal are used inside combinatory function.

The screenshot shows the 'Rules Selector' application window. At the top, there are three tabs: 'Ruleset', 'Help', and 'Algo'. The 'Ruleset' tab is active, displaying 'nb total 128'. Below it, under 'Help', are 'nb total 2', 'nb not executed 2', and 'nb reported 0'. Under 'Algo', it shows 'nb total 9', 'nb not executed 9', 'nb passed 0', and 'nb failed 0'. The main area is a table with columns: Requirement ID, Requirement Name, Implemented /Not Implemented, Type, Parameter, Select All, status, and Log file. The table contains the following data:

Requirement ID	Requirement Name	Implemented /Not Implemented	Type	Parameter	Select All	status	Log file
CNE_01200	Identification of process label	Implemented	Algo	Yes	<input type="checkbox"/>	Not executed	
CNE_04900	Use of clock signal	Implemented	Algo	No	<input type="checkbox"/>	Not executed	
CNE_02300	Preservation of clock name	Implemented	Algo	No	<input checked="" type="checkbox"/>	Not executed	
CNE_02400	Preservation of reset name	Implemented	Algo	No	<input type="checkbox"/>	Not executed	
STD_00200	Name of clock signal	Implemented	Algo	Yes	<input type="checkbox"/>	Not executed	
STD_00300	Name of reset signal	Implemented	Algo	Yes	<input type="checkbox"/>	Not executed	

At the bottom of the window, there is a 'Find' input field, a 'Launch' button, and a 'Cancel' button.

When a violation is detected, a report shall be established with following information:

- Entity name
- Architecture name
- File name

- Clock name before component
- Clock name after component
- Line number of name change
- Component instance name

We can see the result of this detection in

rc_report_rule_REQ_CNE_02300_Preservation of clock name.xml file. In this example, the clock signal "WR" change name for "EEP_WRITE" when the signal go in the component "EEP" in file "AddressDecoder.vhd".

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<CNE_02300>
  <author>rule checker</author>
  <automaticGeneration>YES</automaticGeneration>
  <description>violation report for rule CNE_02300</description>
  <ruleName>Preservation of Clock Name</ruleName>
  <creationDate>Fri Nov 27 16:27:48 CET 2015</creationDate>
  <clockSignalName>
    <violationType>noPreservationName</violationType>
    <entityName>ADDRESSDECODER</entityName>
    <architectureName>ADRESSDECODER_STR</architectureName>
    <fileName>FPGA_FM_Datapackage\02_Code\DPULogic\AddressDecoder\AddressDecoder.vhd</fileName>
    <clockSignalNameBefore>EEP_WRITE</clockSignalNameBefore>
    <clockSignalNameAfter>WR</clockSignalNameAfter>
    <componentName>EEP</componentName>
    <mapLoc>94</mapLoc>
  </clockSignalName>
```

(3) Preservation of reset name (SRS_REQ_CNE_02400)

Rationale: The objective is to check that reset does not change name when it is connected to components.

Detailed Description: Search Reset VHDL elements in VLSI project with “Reset identification” function, and verify that the signal name does not change when entering components at lower level.

Limitations: none at this level.

Expected Result:

When we select the feature Preservation of reset name, as shown in this figure, the rule checker tool list clock signal are used inside combinatory function.

Rules Selector							
Ruleset							
nb total 128							
Help							
nb total 2 nb not executed 2 nb reported 0							
Algo							
nb total 9 nb not executed 9 nb passed 0 nb failed 0							
Requirement ID	Requirement Name	Implemented /Not Implemented	Type	Parameter	<input type="checkbox"/> Select All	status	Log file
CNE_02400	Preservation of reset name	Implemented	Algo	No	<input type="checkbox"/>	Not executed	
STD_00200	Name of clock signal	Implemented	Algo	Yes	<input type="checkbox"/>	Not executed	
STD_00300	Name of reset signal	Implemented	Algo	Yes	<input checked="" type="checkbox"/>	Not executed	
STD_01800	Primitive isolation	Implemented	Help	No	<input type="checkbox"/>	Not executed	
STD_03600	Reset sensitive level	Implemented	Algo	No	<input type="checkbox"/>	Not executed	
STD_03700	Reset assertion and deassertion	Implemented	Help	No	<input type="checkbox"/>	Not executed	

When a violation is detected, a report shall be established with following information:

- Entity name
- Architecture name
- File name
- reset name before component
- reset name after component

- Line number of name change
- Component instance name

We can see the result of this detection in

rc_report_rule_REQ_CNE_02400_Preservation of reset name.xml file. In this example, the reset signal "WR_EN" change name for "RD_COMMAND" when the signal go in the component "CF" in file "SISHandler.vhd".

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<CNE_02400>
  <author>rule checker</author>
  <automaticGeneration>YES</automaticGeneration>
  <description>violation report for rule CNE_02400</description>
  <ruleName>Preservation of Reset Name</ruleName>
  <creationDate>Fri Nov 27 16:27:48 CET 2015</creationDate>
  <resetSignalName>
    <violationType>noPreservationName</violationType>
    <entityName>SISHANDLER</entityName>
    <architectureName>SISHANDLER_STR</architectureName>
    <fileName>FPGA_FM_Datapackage\02_Code\DPULogic\SISHandler\SISHandler.vhd</fileName>
    <resetSignalNameBefore>RD_COMMAND</resetSignalNameBefore>
    <resetSignalNameAfter>WR_EN</resetSignalNameAfter>
    <instanceName>CF</instanceName>
    <mapLoc>91</mapLoc>
  </resetSignalName>
</CNE_02400>
```

(4) Name of clock signal (SRS_REQ_STD_00200)

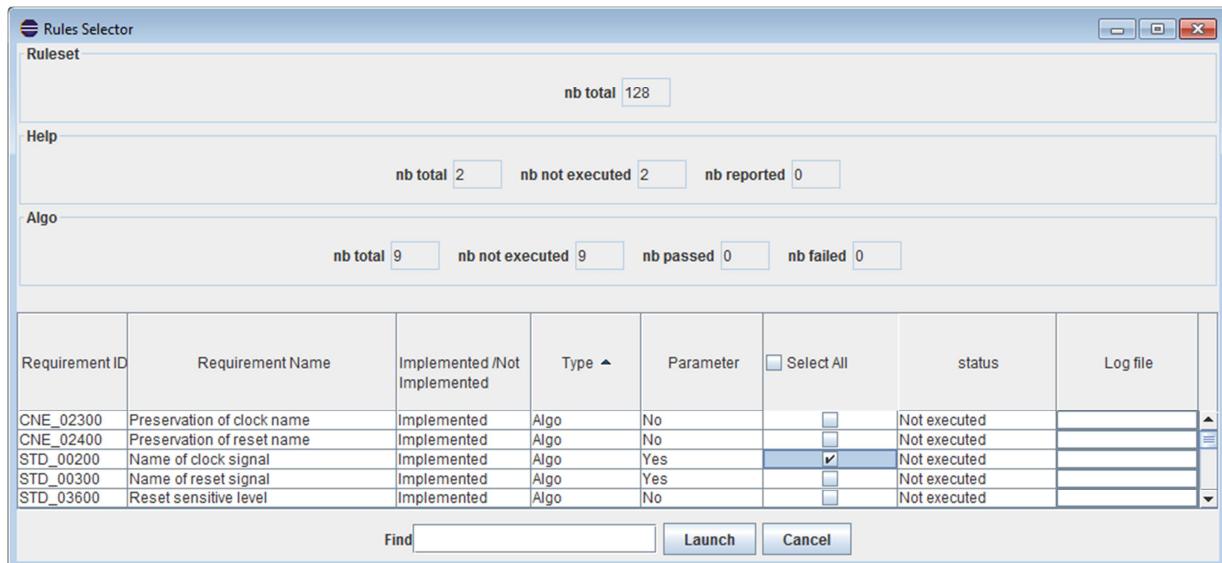
Rationale: The objective is to check that clock signal name includes “clk” or “clock” (for example).

Detailed Description: Search Clock VHDL elements in VLSI project with “Clock identification” function. Verify that the signal name contains string “Clk” or “clock”.

Limitations: none at this level

Expected Result:

When we select the feature Name of clock signal, as shown in this figure, the rule checker tool list clock signal are used inside combinatory function.



When a violation is detected, a report shall be established with following information:

- Clock name
- Entity name
- Architecture name
- File name
- Line number where clock name is badly written

We can see the result of this detection in rc_report_rule_REQ_STD_00200_Name of clock signal.xml file. In this example, the clock signal "c" do not respect the prefix "clk" or "clock" in file "ModuleMap_pkg.vhd".

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<STD_00200>
  <author>rule checker</author>
  <automaticGeneration>YES</automaticGeneration>
  <description>violation report for rule STD_00200</description>
  <ruleName>Name of clock signal</ruleName>
  <creationDate>Fri Nov 27 16:27:48 CET 2015</creationDate>
  <clockSignal>
    <violationType>nameInvalid</violationType>
    <clockSignalName>C</clockSignalName>
    <clockSignalLoc>
      <fileName>\FPGA_FM_Datapackage\02_Code\DPULogic\Definitions\ModuleMap_pkg.vhd</fileName>
      <entityName>CLKOUT</entityName>
      <architectureName>CLKOUT_RTL</architectureName>
      <processName>CLOCK</processName>
      <clockSignalLoc>271</clockSignalLoc>
    </clockSignalLoc>
  </clockSignal>
</STD_00200>
```

(5) Name of reset signal (SRS_REQ_STD_00300)

Rationale: The objective is to check that reset signal name includes “rst”, “reset” or “clr” (for example).

Detailed Description: Search Reset VHDL elements in VLSI project with “Reset identification” function. Verify that the signal name contains string “rst”, “reset” or “clr”.

Limitations: none at this level

Expected Result:

When we select the feature Name of reset signal, as shown in this figure, the rule checker tool list clock signal are used inside combinatory function.

Rules Selector							
Ruleset							
nb total 128							
Help							
nb total 2 nb not executed 2 nb reported 0							
Algo							
nb total 9 nb not executed 9 nb passed 0 nb failed 0							
Requirement ID	Requirement Name	Implemented /Not Implemented	Type ▾	Parameter	<input type="checkbox"/> Select All	status	Log file
CNE_02300	Preservation of clock name	Implemented	Algo	No	<input type="checkbox"/>	Not executed	
CNE_02400	Preservation of reset name	Implemented	Algo	No	<input type="checkbox"/>	Not executed	
STD_00200	Name of clock signal	Implemented	Algo	Yes	<input type="checkbox"/>	Not executed	
STD_00300	Name of reset signal	Implemented	Algo	Yes	<input checked="" type="checkbox"/>	Not executed	
STD_03600	Reset sensitive level	Implemented	Algo	No	<input type="checkbox"/>	Not executed	

When a violation is detected, a report shall be established with following information:

- Reset name
- Entity name
- Architecture name
- File name
- Line number where reset name is badly written

We can see the result of this detection in rc_report_rule_REQ_STD_00300_Name of reset signal.xml file. In this example, the reste signal “DIR_CLK” do not respect the prefix “rst” or “reset” in file “DCFIFO.vhd”.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<STD_00300>
  <author>rule checker</author>
  <automaticGeneration>YES</automaticGeneration>
  <description>violation report for rule STD_00300</description>
  <ruleName>Name of reset signal</ruleName>
  <creationDate>Fri Nov 27 16:27:48 CET 2015</creationDate>
  <resetSignal>
    <violationType>nameInvalid</violationType>
    <resetSignalName>WR_SEQ.WR_HOLD</resetSignalName>
    <resetSignalLoc>
      <fileName>\FPGA_FM_Datapackage\02_Code\DPULogic\SISHandler\DCFIFO.vhd</fileName>
      <entityName>DCFIFO</entityName>
      <architectureName>DCFIFO_RTL</architectureName>
      <processName>SET</processName>
      <resetSignalName>DIR_CLK</resetSignalName>
      <resetSignalLoc>201</resetSignalLoc>
    </resetSignalLoc>
  </resetSignal>
</STD_00300>
```

(6) Reset sensitivity level (SRS_REQ_STD_03600)

Rationale: The objective is to check that each reset source uses only one of activation level. The detection must be done per entity and per VLSI project.

Detailed Description: Search Reset VHDL elements in VLSI project with “Reset identification” function, and verify that the reset signal is always used with the same activation level in each file. When a reset source is used on 2 different levels, a violation report is built showing the different usage of the concerned reset source in the file.

The same detection mechanism is done at VLSI project level. When a reset source is used on different levels between entities, then a violation report is done and all files using the concerned reset source are reported with the corresponding level.

Limitations: none at this level

Expected Result:

When we select the feature Reset sensitivity level, as shown in this figure, the rule checker tool list clock signal are used inside combinatory function.

The screenshot shows the 'Rules Selector' application window. At the top, there are three sections: 'Ruleset' (nb total 128), 'Help' (nb total 2, nb not executed 2, nb reported 0), and 'Algo' (nb total 9, nb not executed 9, nb passed 0, nb failed 0). Below these sections is a large table titled 'Requirement ID' with columns for Requirement ID, Requirement Name, Implemented /Not Implemented, Type, Parameter, Select All, status, and Log file. The table contains the following data:

Requirement ID	Requirement Name	Implemented /Not Implemented	Type	Parameter	Select All	status	Log file
CNE_02400	Preservation of reset name	Implemented	Algo	No	<input type="checkbox"/>	Not executed	
STD_00200	Name of clock signal	Implemented	Algo	Yes	<input type="checkbox"/>	Not executed	
STD_00300	Name of reset signal	Implemented	Algo	Yes	<input type="checkbox"/>	Not executed	
STD_01800	Primitive isolation	Implemented	Help	No	<input type="checkbox"/>	Not executed	
STD_03600	Reset sensitive level	Implemented	Algo	No	<input checked="" type="checkbox"/>	Not executed	
STD_03700	Reset assertion and deassertion	Implemented	Help	No	<input type="checkbox"/>	Not executed	

At the bottom of the table are buttons for 'Find' (with a search bar), 'Launch', and 'Cancel'.

We can see the result of this detection in rc_report_rule_REQ_STD_03600_Reset sensitivity level.xml file. In this example, the reset source tag “resetSourceF” is used in both level in file “DC FIFO.vhd”.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<STD_03600>
  <author>rule checker</author>
  <automaticGeneration>YES</automaticGeneration>
  <description>violation report for rule STD_03600</description>
  <ruleName>Reset Sensitive Level</ruleName>
  <creationDate>Fri Nov 27 16:27:48 CET 2015</creationDate>
  <entity>
    <violationType>mixLevelPerProject</violationType>
    <fileName>\FPGA_FM_Datapackage\02_Code\DPULogic\SISHandler\DC FIFO.vhd</fileName>
    <entityName>DC FIFO</entityName>
    <resetSourceATag>resetSourceA</resetSourceATag>
    <resetSourceALevel>low</resetSourceALevel>
    <resetSourceBTag>resetSourceB</resetSourceBTag>
    <resetSourceBLevel/>
    <resetSourceCTag>resetSourceC</resetSourceCTag>
    <resetSourceCLevel/>
    <resetSourceDTag>resetSourceD</resetSourceDTag>
    <resetSourceDLevel/>
    <resetSourceETag>resetSourceE</resetSourceETag>
    <resetSourceELevel>high</resetSourceELevel>
    <resetSourceFTag>resetSourceF</resetSourceFTag>
    <resetSourceFLevel>both</resetSourceFLevel>
    <resetSourceGTag>resetSourceG</resetSourceGTag>
    <resetSourceGLevel>low</resetSourceGLevel>
    <resetSourceHTag>resetSourceH</resetSourceHTag>
    <resetSourceHLevel>low</resetSourceHLevel>
  </entity>
```

(7) Clock edge sensitivity (SRS_REQ_STD_04800)

Rationale: The objective is to check that for each clock source that there is only one edge detection mechanism used. The detection must be done per entity and per VLSI project.

Detailed Description: Search Clock VHDL elements in VLSI project with “Clock identification” function, and verify that clock are always used with the same edge (falling edge or rising edge) in each file. When a clock source is used on 2 different edges, a violation report is built showing the different usage of the concerned clock source in the file.

The same detection mechanism is done at VLSI project level. When a clock source is used on different edges between entities, then a violation report is done and all files using the concerned clock source are reported with the corresponding edge.

Limitations: none at this level.

Expected Result:

When we select the feature Clock edge sensitivity, as shown in this figure, the rule checker tool list clock signal are used inside combinatory function.

The screenshot shows the 'Rules Selector' window with the following details:

- Ruleset:** nb total 128
- Help:** nb total 2, nb not executed 2, nb reported 0
- Algo:** nb total 9, nb not executed 9, nb passed 0, nb failed 0

Requirement ID	Requirement Name	Implemented /Not Implemented	Type	Parameter	Select All	status	Log file
STD_01800	Primitive isolation	Implemented	Help	No	<input type="checkbox"/>	Not executed	
STD_03600	Reset sensitive level	Implemented	Algo	No	<input type="checkbox"/>	Not executed	
STD_03700	Reset assertion and deassertion	Implemented	Help	No	<input type="checkbox"/>	Not executed	
STD_04800	Clock edge sensitivity	Implemented	Algo	No	<input checked="" type="checkbox"/>	Not executed	
STD_04500	Clock Reassignment	Implemented	Algo	No	<input type="checkbox"/>	Not executed	
CNE_01000	Identification of variable name	Not Implemented	NA	NA	<input type="checkbox"/>	Not implemented	

Buttons at the bottom: Find, Launch, Cancel.

We can see the result of this detection in rc_report_rule_REQ_STD_04800_Clock edge sensitivity level.xml file. In this example, the clock source tag “clockSourceB” is used in both edge in file “ModuleMap_pkg.vhd”.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<STD_04800>
  <author>rule checker</author>
  <automaticGeneration>YES</automaticGeneration>
  <description>violation report for rule STD_04800</description>
  <ruleName>Clock Edge Sensitivity</ruleName>
  <creationDate>Fri Nov 27 16:27:48 CET 2015</creationDate>
  <entity>
    <violationType>mixEdgesPerProject</violationType>
    <fileName>\FPGA_FM_Datapackage\02_Code\DPULogic\Definitions\ModuleMap_pkg.vhd</fileName>
    <entityName>CLKOUT</entityName>
    <clockSourceATag>clockSourceA</clockSourceATag>
    <clockSourceAEdge/>
    <clockSourceBTag>clockSourceB</clockSourceBTag>
    <clockSourceBEdge>both</clockSourceBEdge>
    <clockSourceCTag>clockSourceC</clockSourceCTag>
    <clockSourceCEdge/>
    <clockSourceDTag>clockSourceD</clockSourceDTag>
    <clockSourceDEdge/>
    <clockSourceETag>clockSourceE</clockSourceETag>
    <clockSourceEEdge/>
    <clockSourceFTag>clockSourceF</clockSourceFTag>
    <clockSourceFEdge/>
  </entity>
```

(8) Clock reassignment (SRS_REQ_STD_04500)

Rationale: The objective is to check that clock signal does not reassign to another signal.

Detailed Description: Search Clock Source elements in VLSI project with “SRS_REQ_FEAT_CLK_PRJ” function, and verify that clock sources are not issued of an assignment. This is symbolized with statement “<=“.

Limitations: none at this level.

Expected Result:

When we select the feature Clock reassignment, as shown in this figure, the rule checker tool list clock signal are used inside combinatory function.

Rules Selector							
Ruleset							
nb total 128							
Help							
nb total 2 nb not executed 2 nb reported 0							
Algo							
nb total 9 nb not executed 9 nb passed 0 nb failed 0							
Requirement ID	Requirement Name	Implemented /Not Implemented	Type	Parameter	<input type="checkbox"/> Select All	status	Log file
STD_04800	Clock edge sensitivity	Implemented	Algo	No	<input type="checkbox"/>	Not executed	
STD_04500	Clock Reassignment	Implemented	Algo	No	<input checked="" type="checkbox"/>	Not executed	
CNE_01000	Identification of variable name	Not Implemented	NA	NA	<input type="checkbox"/>	Not implemented	
CNE_00500	Convention for signal naming	Not Implemented	NA	NA	<input type="checkbox"/>	Not implemented	
CNE_00600	Convention for constant naming	Not Implemented	NA	NA	<input type="checkbox"/>	Not implemented	

When a violation is detected, a report shall be generated with following information:

- Clock source name
- Entity name
- Architecture name
- File name
- Line number

We can see the result of this detection in rc_report_rule_REQ_STD_04500_clock_reassignment.xml file. In this example, the clock source "Y" is assign to '1' in file "DCFIFO.vhd".

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<STD_04500>
  <author>rule checker</author>
  <automaticGeneration>YES</automaticGeneration>
  <description>violation report for rule STD_04500</description>
  <ruleName>Clock Reassignment</ruleName>
  <creationDate>Fri Nov 27 16:27:48 CET 2015</creationDate>
  <clockSource>
    <violationType>ClockReassignment</violationType>
    <fileName>FPGA_FM_Datapackage\02_Code\DPULogic\SISHandler\DCFIFO.vhd</fileName>
    <entityName>DCFIFO</entityName>
    <architectureName>DCFIFO_RTL</architectureName>
    <clockSourceName>DIR_CLK</clockSourceName>
    <clockSourceLoc>155</clockSourceLoc>
  </clockSource>
</STD_04500>
```

F. Synthesis

The following file gives a list of the tools and rules supported by the Rule Checker. The parent references, the log report items and the limitations for each tool/rule are described.



G.Acronyms and abbreviations

Acronym and abbreviation	Meaning
CNES	Centre National d'Etude Spatial (French Space Agency)
FPGA	Field Programmable Gate Array
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuit
VLSI	Very Large Scale Integration

H.Glossary

The table below lists the definition of each item used in log reports.

Items	Description
author	Name of the author of the report <u>or</u> Name of the tool used for report generation.
automaticGeneration	Tells if the report has been generated automatically.
description	Name of the test done.
ruleName	Name of the rule
creationDate	Date of report generation.
fileName	Name of VLSI file.
nbLine	Number of lines of the VLSI file.
entityName	Name of the VHDL entity. When no entity is present in VLSI file, the field is left empty.
entityLoc	Location (expressed in line number) in the VLSI file of the declaration of the entityName.
architectureName	Name of the VHDL architecture. When no architecture is present in VLSI file, the field is left empty.
architectureLoc	Location (expressed in line number) in the VLSI file of the declaration of the architectureName.
processName	Name of the VHDL process. When no process is present in VLSI file, the field is left empty. When the process has no label, then the field returns "unnamed".
processLoc	Location (expressed in line number) in the VLSI file of the declaration of the processName.
isSynchronous	Tells if the process is a synchronous process.

clockSignalName	Name of clock signal used In synchronous process. When process is not synchronous, the field is left empty.
clockSignalNameBefore	Name of clock signal before changed
clockSignalNameAfter	Name of clock signal after changed
clockSignalLoc	Location (expressed in line number) in the VLSI file of the usage of the clockSignalName.
hasAsynchronousReset	Tells if the synchronous process uses an asynchronous reset.
resetSignalName	Name of asynchronous reset signal used In synchronous process. When process has no asynchronous reset, the field is left empty.
resetSignalLoc	Location (expressed in line number) in the VLSI file of the usage of the resetSignalName.
libraryName	Name of library declared In VLSI file.
libraryNameLoc	Location (expressed in line number) in the VLSI file of the declaration of the libraryName.
violationType	Name of violation for rule Algo
instanceName	Name of the component when is instantiate
mapLoc	Location (expressed in line number) in the VLSI file of the declaration of the signal mapping.
clockSourceName	Name of clock source represent the origin of the clock signal.
clockSourceTag	Tag of clock source, tag is used to avoid mismatch.
clockSourceLoc	Location (expressed in line number) in the VLSI file of the usage of the clockSourceName.