Continuous VHDL inspection with open source software : feedbacks

Florent Manni

florent.manni@cnes.fr

Octobre 2019 | COMET : Comment valoriser par l'Open Source? | Toulouse

cnes
CENTRE NATIONAL D'ÉTUDES SPATIALES

VHDLTOOL

Project developed by the French space Agency (CNES) to improve VHDL development within universities and sub-contractors

- **Started in 2014 – Still on going**
- **For Linux and Windows**

Phase 1:
Create a versatile set of VHDL rules

Phase 2:
Create a software for Rules verification

Phase 3:
GUI and project deployment

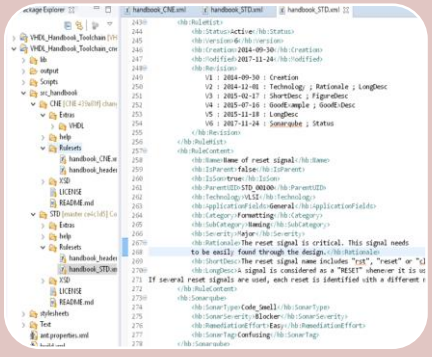
Everything is opensource!!

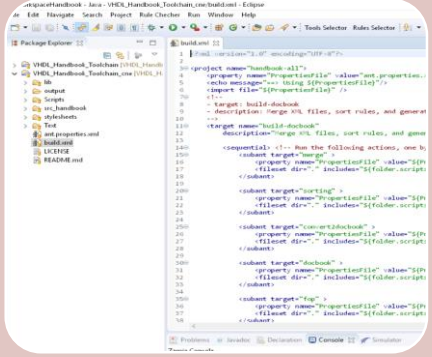
(hosted on Github)

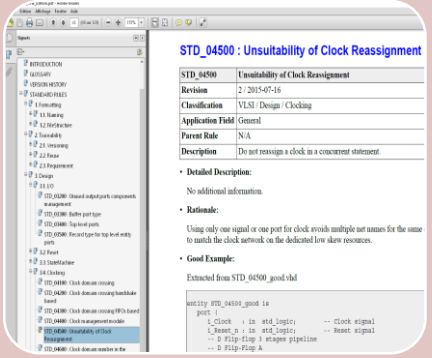
https://github.com/VHDLTool

cnes

Phase 1:
Create a versatile set of VHDL rules







Unique XML database including:

- VHDL rules
- images
- examples

Eclipse toolchain to convert XML to PDF

VHDL Rules:

- 74 standard
- 55 CNES custom

eclipse

VHDLTool / VHDL_Handbook_STD

VHDLTool / VHDL_Handbook_CNE

XML

SAXONICA
XSLT AND XQUERY PROCESSING

PDF

FOP

DocBook

https://github.com/VHDLTool/VHDL_Handbook_CNE/releases/download/V1.1RC2u1/handbook_CNE_Edition.pdf

Feedbacks

- Doing open source project at CNES need to be **anticipated from the call for tender**
  => need a **transfer of intellectual property** written in the contract (“régime de cession”)

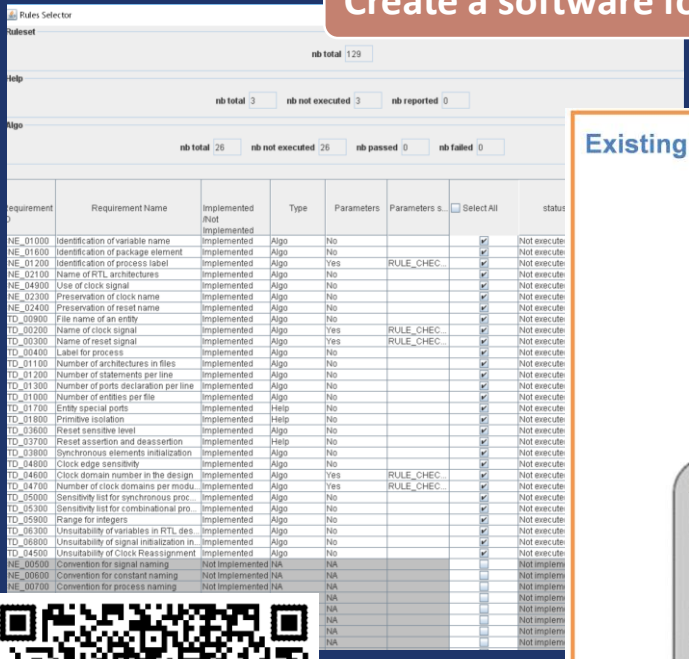
- Finding companies “opensource friendly” in “politique achat” is difficult :
  => the referenced companies do not feel confident with **opensource software** and its **business model.**

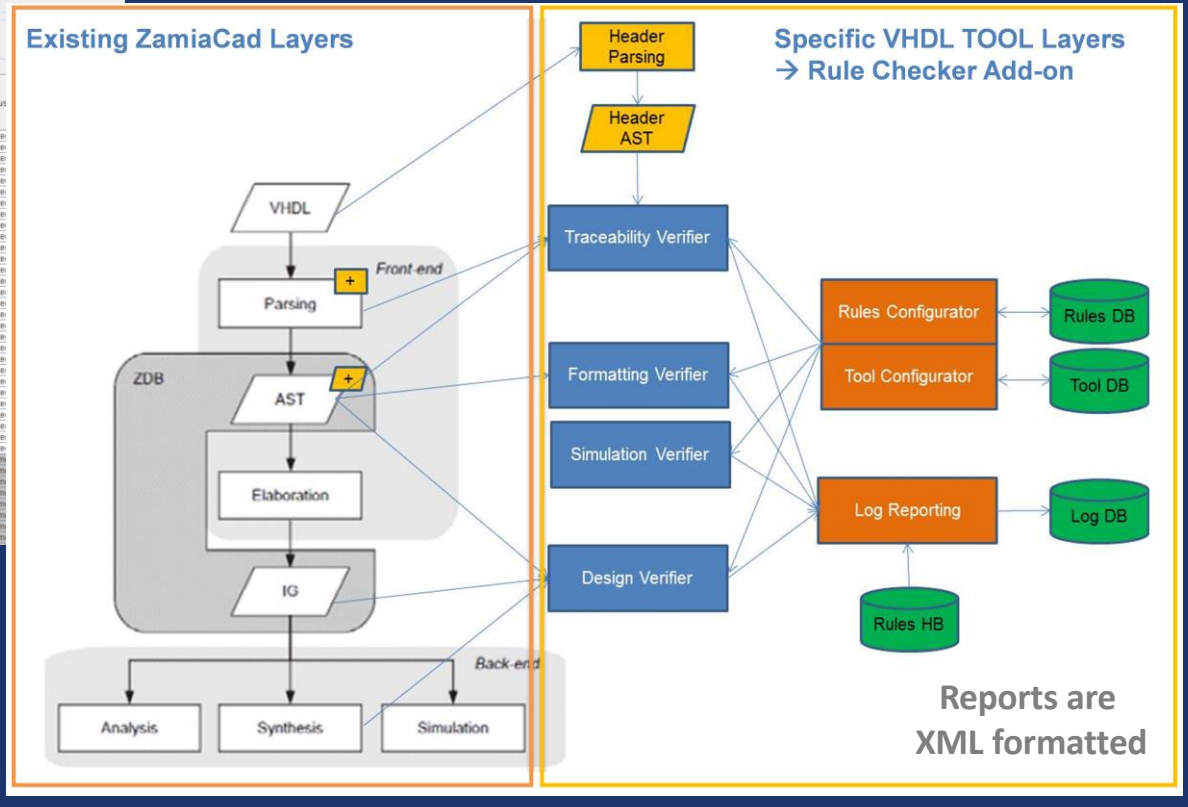
- Code release was not easy :
  => CNES personnel was **not allowed to create a Github** repository by himself
  => the “Commission Brevets et Logiciels” had to be attended to allow release
  => release need to be at first done on website https://logiciels.cnes.fr/ prior to Github

Phase 2:

Create a software for Rules verification

http://zamiacad.sourceforge.net/





ZamiaCAD

eclipse

VHDLTool / Zamiacad-Rulechecker

Built on top existing Eclipse plugin Zamiacad

Creation of a new Zamiacad-RC plugin

https://github.com/VHDLTool/Zamiacad-Rulechecker/releases/download/v7.0.0/Zamia-plugin-update-7.0.0-20190227.zip

Feedbacks

Phase 2:
Create a software for rules validation

http://zamiacad.sourceforge.net/

- **Finding a VHDL opensource solution was not easy:**
  => GHDL and Vhdleditor were considered but Zamiacad
  (http://zamiacad.sourceforge.net/web/) was selected even though it is not perfect.

- **Getting support from original developer was tried :**
  => Audio conferences done but no real time available on there side.
  Zamiacad was a university project

VHDLTool / Zamiacad-Rulechecker

- **Keeping knowledge in the contractor (Altran) was hard to maintain**
  => founding rely only on CNES and was done for a short period of time
  => Altran tried once to use "crédit de recherche" to add some rules by itself

VHDLTool / **VHDL-RC**
forked from Linty-Services/VHDL-RC

sonarqube
Community 7.9 LTS

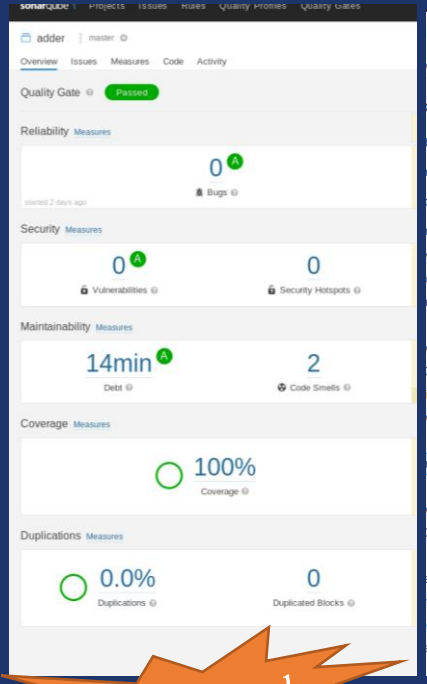
Create 3 Sonarqube plugins:

- sonar-VHDLRC
- sonar-coverage-ghdl
- sonar-modelsim-plugin

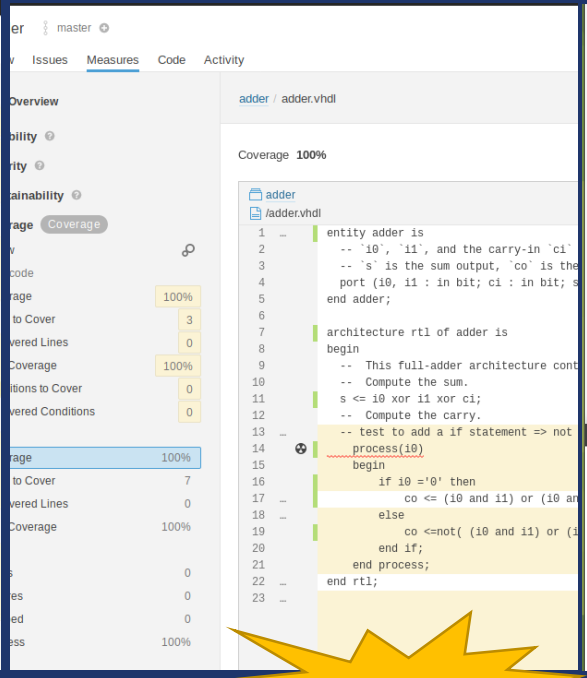
Phase 3:

GUI and project deployment

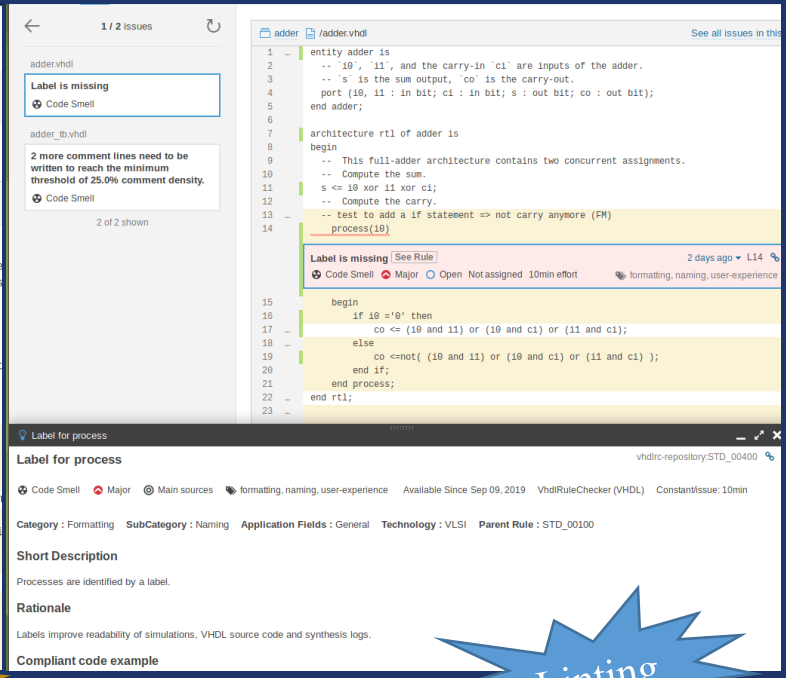
Hide Eclipse project inside a custom sonar scanner



Dashboard



Coverage



Linting

https://github.com/VHDLTool/sonar-VHDLRC/releases

VHDLTool / **VHDL-RC**
forked from Linty-Services/VHDL-RC

sonarqube
Community 7.9 LTS

Phase 2 : GUI and project deployment Feedbacks

- Partnership was made with DGA spinoff Linty Service for the GUI into **Sonarqube**
  => the company is more “OpenSource friendly”
- Usage of **XML** “open structured” language allow quick importation into Sonarqube
  => a proof a concept was made during the call for tender allowing **a mockup demo**
- **VHDLTool** is now the free first step in the continuous VHDL quality assessment
- **Bug correction** for Zamiacad is still not secured

- Doing opensource at CNES is quite difficult and need a important commitment from the project engineer.

- Opensource is virtuous:
  the handbook is now used by universities and laboratories
  and sonar-VHDL-RC will improve VHDL development

- The opensource project need a community or a company using it to “survive”. To do so, doing conferences (like ORCONF) is important

- Usage of Opensource technology (like for GUI and VHDL parsing) allowed us to focus on core knowledge (VHDL rules) application and to do it quickly

- VHDLTOOL is now vector for continuous quality assessment for new VHDL development

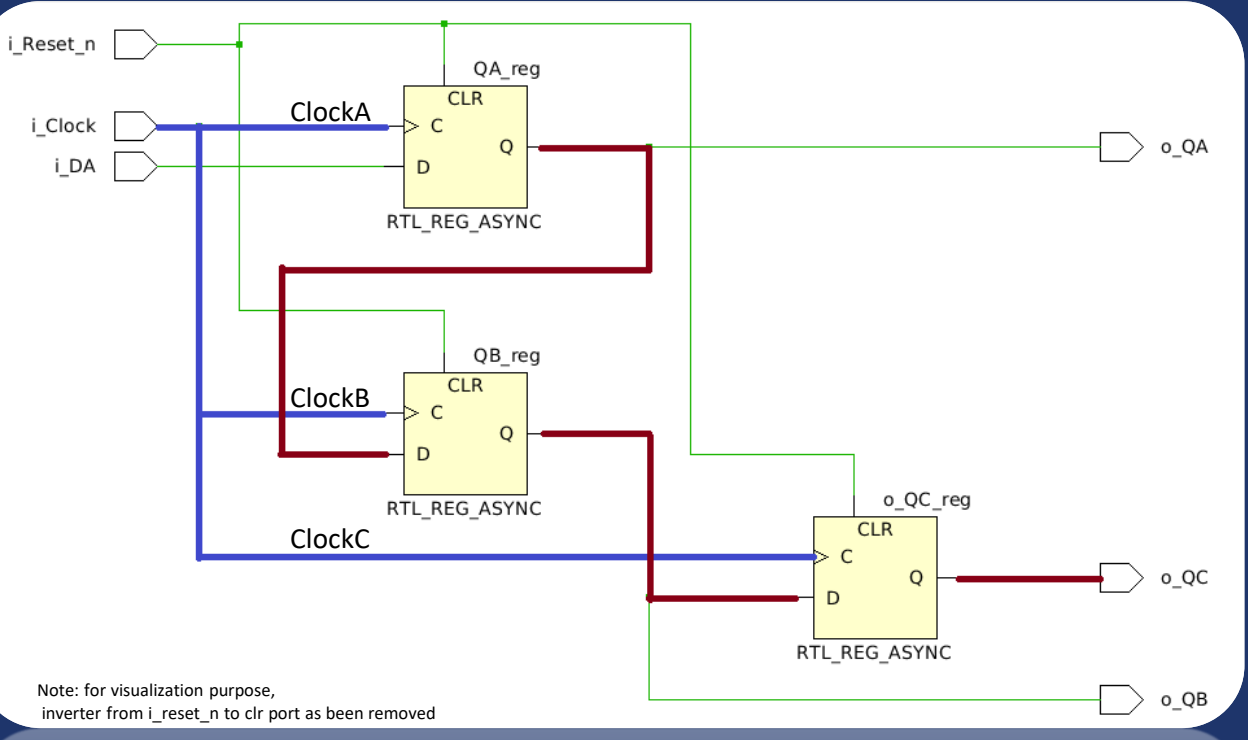
Try the demo…

Thank you !

VHDLTOOL in action

cnes
CENTRE NATIONAL D'ÉTUDES SPATIALES

Is it worth bothering?

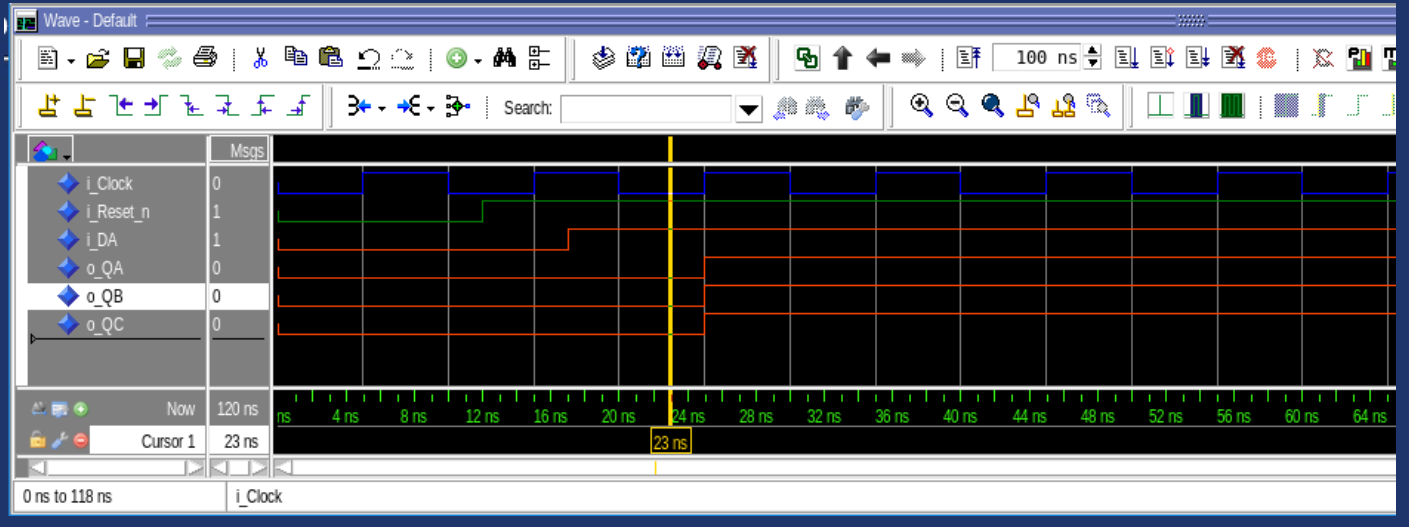



```
architecture Behavioral of STD_04500_bad is
   signal ClockA : std_logic;          -- Clock input for A Flip-Flop
   signal ClockB : std_logic;          -- Clock input for B Flip-Flop
   signal ClockC : std_logic;          -- Clock input for C Flip-Flop
   signal QA     : std_logic;
   signal QB     : std_logic;
begin

   ClockC <= ClockB;
   ClockB <= ClockA;
   ClockA <= i_Clock;


   -- First Flip-Flop
   P_FlipFlopA : process(ClockA, i_Reset_n)
   begin
      if (i_Reset_n = '0') then
         QA <= '0';
      elsif (rising_edge(ClockA)) then
         QA <= i_DA;
      end if;
   end process;

   -- Second Flip-Flop
P_FlipFlopB: process(ClockB, i_Reset_n)
   begin
      if (i_Reset_n = '0') then
         QB <= '0';
      elsif (rising_edge(ClockB)) then
         QB <= QA;
      end if;
   end process;

   -- Third Flip-Flop
P_FlipFlopC: process(ClockC, i_Reset_n)
   begin
      if (i_Reset_n = '0') then
         o_QC <= '0';
      elsif (rising_edge(ClockC)) then
         o_QC <= QB;
      end if;
   end process;

   o_QA <= QA;
   o_QB <= QB;
end Behavioral;
--CODE
```

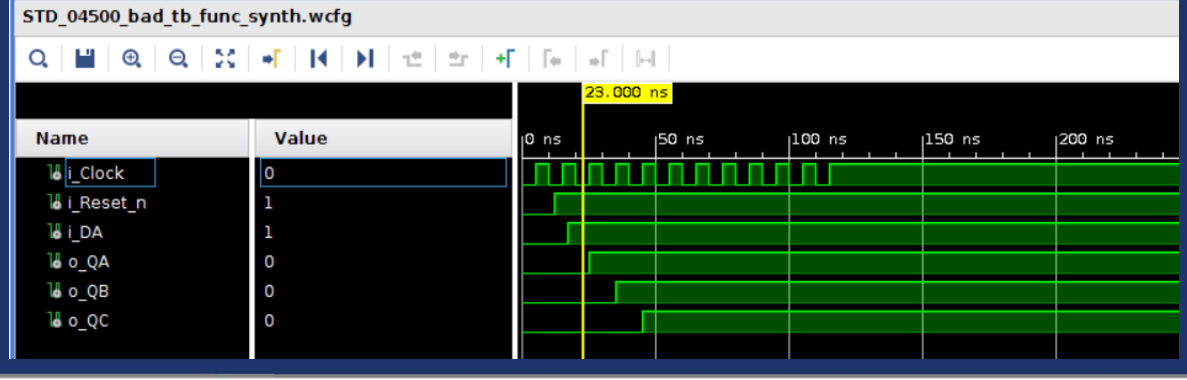
Example coming from : https://github.com/VHDLTool/VHDL_Handbook_STD/blob/master/Extras/VHDL/STD_04500_bad.vhd

Is it worth bothering?.....Really?

Functional simulation



Post-synthesis simulation



How long to fix this issue?
Minutes ?
Hours?
Days?

With Sonarqube it would have taken seconds....

STD_04500 : Unsuitability of Clock Reassignment

STD_04500	Unsuitability of Clock Reassignment
Revision	3 / 2017-11-24
Classification	VLSI / Design / Clocking
Application Field	General
Parent Rule	N/A
Description	Do not reassign a clock in a concurrent statement.

- **Detailed Description:**

  No additional information.

- **Rationale:**

  Reassignement of clock signal can lead to simulation mismatch regarding the Plac
  simulation and delta cycle topic.

- **Good Example:**

  Extracted from STD_04500_good.vhd

```
entity STD_04500_good is
   port (
      i_Clock    : in  std_logic;        -- Clock signal
      i_Reset_n : in  std_logic;        -- Reset signal
      -- D Flip-flop 3 stages pipeline
      -- D Flip-Flop A
      i_DA       : in  std_logic;        -- Input signal
      o_QA       : out std_logic;        -- Output signal
      -- D Flip-flop B
      o_QB       : out std_logic;        -- Output signal
      -- D Flip-Flop C
      o_QC       : out std_logic         -- Output signal
      );
end STD_04500_good;
```

```
66        o_QC      : out std_logic         -- Output signal
67        );
68  end STD_04500_bad;
69
70  architecture Behavioral of STD_04500_bad is
71     signal ClockA : std_logic;           -- Clock input for A Flip-Flop
72     signal ClockB : std_logic;           -- Clock input for B Flip-Flop
73     signal ClockC : std_logic;           -- Clock input for C Flip-Flop
74     signal QA     : std_logic;
75     signal QB     : std_logic;
76  begin
77  …
78     ClockC <= ClockB;
79     ClockB <= ClockA;
80  …  ClockA <= i_Clock;
```

Clock signal I_CLOCK is reassigned to CLOCKA See Rule — 5 hours ago · L80
Bug · Blocker · Open · Not assigned · 5min effort · Comment — clocking, design

Too many clock domains in the design See Rule — 5 hours ago · L80
Code Smell · Critical · Open · Not assigned · 1d effort · Comment — clocking, design, suspicious

Too many clock domains in the entity STD_04500_BAD See Rule — 4 minutes ago · L80
Code Smell · Critical · Open · Not assigned · 3h effort · Comment — clocking, design, suspicious

```
81
82  …
83  …     -- First Flip-Flop
84  …     P_FlipFlopA : process(ClockA, i_Reset_n)
85        begin
86           if (i_Reset_n = '0') then
87              QA <= '0';
88           elsif (rising_edge(ClockA)) then
89              QA <= i_DA;
90           end if;
91        end process;
92  …
```

https://github.com/VHDLTool/VHDL_Handbook_STD/blob/master/Extras/VHDL/STD_04500_bad.vhd