Digital Design and Computer Architecture LU

VHDL Coding and Design Guidelines

Florian Huemer, Florian Kriebel
{fhuemer,fkriebel}@ecs.tuwien.ac.at
Department of Computer Engineering
TU Wien

Vienna, March 6, 2022

1 Overview

This document specifies the VHDL coding guidelines used in the DDCA lab course. The use of these rules is mandatory and ignoring them leads to point reductions (please note that common conventions regarding naming, comments, etc. still apply but are not listed here explicitly).

2 Rules

2.1 Reset

All registers must be set to a defined value during reset. This means that every signal that is written in a synchronous process must also get an appropriate reset value.

2.2 Active Clock Edge

This section only applies to *synthesizeable* code, i.e., it does not apply to test code (testbenches). In the lab course only the *rising clock edge* is used as the active clock edge. Hence, you are not allowed to use `falling_edge` anywhere for your design. Moreover, `rising_edge` must only be used on clock signals.

2.3 Packages

Package files should always be named with the suffix `_pkg.vhd`, the package name itself should end in the suffix `_pkg`. The *math* package provided by our library is called `math_pkg` and stored in the file `math_pkg.vhd`.

If you need a package body, put it in the *same* file as the actual package.

2.4 Sensitivity Lists

If you use explicit sensitivity lists, they must contain all required signals and must not contain any spurious signals. Check the Hardware Modeling course material for detailed information about sensitivity lists. You can also use the `all` keyword. However, be aware that some tasks may disallow this method and require explicit lists!

2.5 Testbenches

Testbench files should always be named with the suffix `_tb.vhd`. The testbench itself should be named after the module it tests, extended with the suffix `_tb`. For example the testbench of the synchronizer module `sync` is named `sync_tb` and placed in the file `sync_tb.vhd`. A testbench must always contain a correct reset operation, i.e., in the beginning of the simulation, a reset pulse must be applied to the unit under test (UUT) in order to reset all internal registers and output signals of the component. The simulation must not show any undefined signals (i.e. red signal traces) except for a brief period before the reset. This means that at the beginning of a simulation *all* input signals to the UUT have to be initialized!

In simulation screenshots a clock signal, signal names and time axis must *always* be visible. Time intervals must always be measured from a signal edge to another edge! Use the alignment cursor features in QuestaSim to place the cursors exactly on the edges.

Simulations in Questa/Modelsim should also not trigger **any** warnings (except for the 0 ps time step or during reset).

2.6 Entities and Architectures

Entities and the respective architectures must always be put into the same file (this rule does not apply to files that come with the template). A single file may only contain a single entity. The file name should be the name of the entity.

2.7 Instances

To avoid the introduction of bugs associated with wrong signal mappings, **only** the “named mapping” style for connecting wires to an instantiated module must be used. The use of “positional mapping” is hence not allowed. Unused instance outputs must always bei explicitly marked with the `open` keyword.

2.8 Indentation

Use *either* tabs or spaces to properly indent and format your code. **Don’t mix** indentation styles in one document!

Revision History

Revision	**Date**	**Author(s)**	**Description**
1.0	06.03.2022	FH,FK	Initial version

Author Abbreviations:

FH	Florian Huemer
FK	Florian Kriebel