HW/SW Codesign [LU] – WS2023

VHDL Coding and Design Guidelines

Florian Huemer
fhuemer@ecs.tuwien.ac.at
Department of Computer Engineering
TU Wien

Vienna, October 8, 2023

1 Overview

This document specifies the VHDL coding and design guidelines used in the HW/SW Codesign lab course. The use of these rules is mandatory. Disregarding the rules may lead 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. However, be sure to *not* include a reset for memories that should be implemented using the FPGA's block RAM. Doing so forces the synthesis tool to implement the memory using registers in the logic cells in the FPGA, since the dedicated block RAM does not support a reset.

2.2 Active Clock Edge

This restriction 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, there should be no screnario where the use `falling_edge` is applicable (if you still use it, you must have a well-founded justification). Moreover, `rising_edge` (`falling_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.

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!

2.6 Entities and Architectures

Entities and the respective architectures should be put into the same file. A single file should 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 be 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!

3 Quartus Warnings

Although your design might be correct, Quartus still outputs some warnings during the compilation process. Table 3.1 lists all warnings that your design is allowed to contain. There is no need in trying to fix these warnings from your side, since they won't have any negative impact on your grade. However, all other warnings indicate problems with your design and will hence reduce the total number of points you get for the assignment.

If you get a warning that is obviously not caused by your code, but rather originates from modules generated by the Platform Designer (e.g., the Nios 2 processor), you may also ignore it. However, this does not apply to timing violations! Your design is expected to meet all timing requirements during Timing Analysis in Quartus, no matter where the critical path lies.

The last two warnings in Table 3.1 may still indicate problems with your design. So thoroughly check which signals these warnings are reported for! If you have for example an input button that should trigger some action in your design and Quartus reports that it does not drive any logic, then there is certainly a problem. On the other hand, if you intentionally drive some output with a certain constant logical level (for example an unused seven segment display), then the "stuck at VCC or GND" warning is fine.

ID	Description
18236	Number of processors has not been specified which may cause overloading on shared machines. Set the global assignment NUM_PARALLEL_PROCESSORS in your QSF to an appropriate value for best performance.
13009	TRI or OPNDRN buffers permanently enabled.
276020	Inferred RAM node [...] from synchronous design logic. Pass-through logic has been added to match the read-during-write behavior of the original design.
15064	PLL [...]|altpll:altpll_component|pll_altpll:auto_generated|pll1" output port clk[0] feeds output pin "nclk~output" via non-dedicated routing -- jitter performance depends on switching rate of other design elements. Use PLL dedicated clock outputs to ensure jitter performance
169177	[...] pins must meet Intel FPGA requirements for 3.3-, 3.0-, and 2.5-V interfaces. For more information, refer to AN 447: Interfacing Cyclone IV E Devices with 3.3/3.0/2.5-V LVTTL/LVCMOS I/O Systems.
171167	Found invalid Fitter assignments. See the Ignored Assignments panel in the Fitter Compilation Report for more information.
15705	Ignored locations or region assignments to the following nodes
15714	Some pins have incomplete I/O assignments. Refer to the I/O Assignment Warnings report for details
12240	Synthesis found one or more imported partitions that will be treated as black boxes for timing analysis during synthesis
13024	Output pins are stuck at VCC or GND
21074	Design contains [...] input pin(s) that do not drive logic
176250	Ignoring invalid fast I/O register assignments. See the Ignored Assignments panel in the Fitter Compilation Report for more information.

Table 3.1: Allowed warnings

Revision History

Revision	Date	Author(s)	Description
23.1	09.10.2023	FH	Initial version

Author Abbreviations:

FH Florian Huemer