System Verilog Coding Standards

Benjamin Davis, allrisc.dev@gmail.com

2023-02-28

Contents

Chapter 1

Purpose

Robert C. Martin, author of *Clean Code: A Handbook of Agile Software Craftsmanship*, famously said:

> Indeed, the ratio of time spent reading versus writing is well over 10 to 1. We are constantly reading old code as part of the effort to write new code. ... [Therefore,] making it easy to read makes it easier to write.

This quote certainly holds true whether the code is traditional software, RTL, or Verification frameworks. Thus this document puts forward a set of rules and guidelines for writing easy to understand and consistent SystemVerilog.

A vast majority of these rules apply to SystemVerilog for synthesis and for verification. When this is not the case it will be noted and further guidance shall be provided.

While these are referred to as rules, they may not be the correct solution for all code-bases. When an instance arises where a developer feels that there is a method of writing the code which is clearer they should be empowered to do so. But, in general these moments are few and far-between, so the developer is asked to think critically about whether this approach is truly better.