
Synopsis

coverage 92% coverage 92%

NVC is a VHDL compiler and simulator.

NVC supports almost all of VHDL-2008 with the exception of PSL, and it has been successfully used to simulate several real-world designs. Experimental support for VHDL-2019 is under development.

NVC has a particular emphasis on simulation performance and uses LLVM to compile VHDL to native machine code.

NVC is not a synthesizer. That is, it does not output something that could be used to program an FPGA or ASIC. It implements only the simulation behaviour of the language as described by the IEEE 1076 standard.

NVC supports popular verification frameworks including OSVVM, UVVM, VUnit and cocotb. See below for installation instructions.

Usage

Simulating a VHDL hardware design involves three steps: *analysing* the source files; *elaborating* the design; and *running* the simulation. This is analogous to compiling, linking, and executing a software program. With NVC these steps are accomplished using the `-a`, `-e`, and `-r` commands:

```
1 $ nvc -a my_design.vhd my_tb.vhd
2 $ nvc -e my_tb
3 $ nvc -r my_tb
```

Or more succinctly, as a single command:

```
1 $ nvc -a my_design.vhd my_tb.vhd -e my_tb -r
```

Where `my_tb` is the name of the top-level test-bench entity.

The full manual can be read after installation using `man nvc` or online.

License

This program is free software distributed under the terms of the GNU General Public License version 3 or later. You may use, modify, and redistribute the program as you wish but if you distribute modifications you must preserve the license text and copyright notices, and also make the modified source code available to your users.

The source files for the IEEE standard libraries are included in the repository. These were originally provided under a proprietary license that forbid distribution of modifications, but in 2019 were relicensed under Apache 2.0. Freely redistributable versions of the 1993 libraries were made by editing and removing declarations from the 2019 libraries, and so are also licensed under Apache 2.0. Certain VHDL libraries developed specifically for NVC under `lib/nvc` and `lib/std` are also licensed under Apache 2.0. See the individual files for details.

The VITAL libraries are distributed under `lib/vital`. These were derived from draft copies of the packages freely available on the internet. The license status of these is unclear as the final text is part of the VITAL standard which must be purchased from the IEEE. If you are packaging this program for a distribution with strict free software requirements you should strip these files from the tarball and configure with `--disable-vital`.

Installing

NVC is developed under GNU/Linux and is regularly tested on macOS and Windows under MSYS2.

On macOS NVC can be installed with `brew install nvc`. NVC is also packaged for FreeBSD, Gentoo, Arch Linux AUR, and several other distributions. A Windows installer is available from the releases page. Users of systems without existing packages should build from source.

NVC has both a release branch and a development master branch. The master branch should be stable enough for day-to-day use and has comprehensive regression tests, but the release branch is more suitable for third party packaging. The latest released version is 1.12.0. Significant changes since the last release are detailed in NEWS.md.

If you are building from a Git clone rather than a released tarball you first need to generate the configure script using:

```
1 ./autogen.sh
```

In-tree builds are not supported so create a separate build directory:

```
1 mkdir build && cd build
```

Finally build and install using the standard autotools steps:

```
1 ../configure
2 make
3 sudo make install
```

To use a specific version of LLVM add `--with-llvm=/path/to/llvm-config` to the configure command. The minimum supported LLVM version is 8.0. Versions between 8 and 16 have all been tested.

NVC also depends on Flex to generate the lexical analyser.

On a Debian derivative the following should be sufficient to install all required dependencies:

```
1 sudo apt-get install build-essential automake autoconf \  
2 flex check llvm-dev pkg-config zlib1g-dev libdw-dev \  
3 libffi-dev libzstd-dev
```

Only the MSYS2 environment on Windows is supported. The required dependencies can be installed with:

```
1 pacman -S base-devel mingw-w64-x86_64-{llvm,ncurses,libffi,check,pkg-  
config,zstd}
```

GtkWave can be used to view simulation waveforms. Version 3.3.79 or later is required for the default FST format.

Testing To run the regression tests:

```
1 make check
```

The unit tests require the check library.

Reporting bugs

Report bugs to nick@nickg.me.uk or using the GitHub issue tracker. Please include enough information to reproduce the problem, ideally with a small VHDL test case. Issue #412 is a good example.

Please remember that this software is provided to you with NO WARRANTY and no expectation of support, but I will do my best to help with any issues you encounter.

Contributing

Patches can be sent as either pull requests on GitHub or by email using `git -send-email`. Please note however that as this is purely a spare-time hobby project for me, I have limited time available to review patches. All code submitted must follow the guidelines in `contrib/STYLE.md`.

I will not accept patches that add new copyright owners under `src/`. This is to ensure there is clear legal ownership should, for example, the license need to be updated. Significant contributors are instead listed in `THANKS.md`.

Language Support

VHDL standard revisions are commonly referred to by the year they were published. For example IEEE 1076-2008 is known as VHDL-2008. The default standard in NVC is currently VHDL-2008 but this can be changed with the `--std` argument. For example `--std=1993` selects the VHDL-1993 standard.

The 1993, 2000, and 2002 revisions of the standard are fully supported. Please raise bugs for any missing or incorrectly implemented features you encounter. The current status of VHDL-2008 and VHDL-2019 support can be found on the features page.

VHPI The VHDL standard contains a comprehensive API called VHPI for interfacing with foreign code written in C or another language. NVC implements a subset of VHPI sufficient for running cocotb. Refer to the manual for more information.

Vendor Libraries

NVC provides scripts to compile popular verification frameworks and the simulation libraries of common FPGA vendors.

- For OSVVM use `nvc --install osvvm`
- For UVVM use `nvc --install uvvm`
- For Xilinx ISE use `nvc --install ise`
- For Xilinx Vivado use `nvc --install vivado` and additionally `nvc --install xpm_vhdl` if you require simulation models of the XPM macros
- For Altera Quartus use `nvc --install quartus`
- For Lattice iCEcube2 use `nvc --install icecube2`
- For Free Model Foundry common packages use `nvc --install fmf`

The libraries will be installed under `~/ .nvc/lib`.