FOSS - A Solution to the Challenges faced by the Closed Source world

Medha Chhillar and Neeraj Kr. Shukla

Abstract— The concept of open source and free sharing of technological information existed long before EDA or computers. FOSS (Free Open Source Software) or Open Source Software is easily available to the public. This in turn enables anyone and everyone to use it without paying royalties or fees. This paper discusses about the advantages of free and open source software over close source structures. It also discusses about the different types of open source EDA tools available on the internet; mainly focusing on the ICARUS Verilog simulator and its related problems.

Index Terms— Crowd Sourcing, EDA, Open Source, GUI, GPL, Simulation

1 INTRODUCTION

The concept of open source and free sharing of technological information existed long before EDA or computers. Open source software is software that is made available to the public, enabling anyone to use it without paying royalties or fees. Open source software evolves through community cooperation (crowd sourcing) with the sharing of experiences and ideas that can then be incorporated into future releases [1]. The Open Source industry is gaining importance these days due to the reasons mentioned below.

i. Open Source over Closed Source

These days designing a chip has become very important. This is due to the newer technologies evolving at a very high pace. So, to match up the growth rate, people require tools to design these chips. Since it takes a great deal of time to build up a tool, people seek out for the tools that are already available in the market. Now the problem is that, not all can afford to buy such tools. So SME's suffer at this point and large firms that can afford such tools have an advantage. Some of the commercial tools having a closed file structure are by MentorGraphics, Cadence, etc. In order to deal with the situation, people came up with the idea of free sourcing or crowd sourcing. Here, people from all parts of the world come together at a common platform to build a software that is available to all and for free. On the other hand, bug-fixing or adding features were difficult in case of closed file structures whereas it is not so with the open file structures. The advantages of the open source EDA tools usage and the easy availability of its various types such as Icarus Verilog, Verilator, GHDL VHDL simulator, gEDA, Veriwell, etc makes the design process comparatively more cost-effective, less time consuming and affordable for all. With Open source software the development is fast and so it is cost effective. It can be freely redistributed. Closed source on the other hand is available only after paying heavy remunerable fee to buy the software. Adding functionality to these softwares is very difficult and time consuming. This is so, because the person may have to wait for the owner of the tool to make the required changes, since the source code is not available to all. The company may have to change even the software and hence it would be more cost consuming.

ii. Future Scope of Open Source

Libraries are the basic source of information. Open source softwares are free to make use of these libraries. As discussed above, these are good for small and medium libraries. Since these are free, so these are issued in the interest of the public. So the software is issued under GPL. It encourages students for effective education. Since, these libraries are free of cost, so it is beneficial for the users. The transfer of technical know-how is also encouraged.

2 OPEN SOURCE SIMULATION TOOLS

2.1 EMACS – Text Editor

GNU Emacs is an extensible, customizable text editor. It provides very good support for both VerilogHDL and VHDL editing. At its core is an interpreter for Emacs Lisp, a dialect of the Lisp programming language with extensions to support text editing. The features of GNU Emacs include:

- Content-sensitive editing modes, including syntax colouring, for a variety of file types including plain text, source code, and HTML.
- Complete built-in documentation, including a tutorial for new users.
- Full Unicode support for nearly all human languages and their scripts.
- Highly customizable, using Emacs Lisp code or a graphical interface.
- A large number of extensions that add other functionality, including a project planner, mail and news reader, debugger interface, calendar, and more. Many of these extensions are distributed with GNU Emacs; others are available separately [2].

2.2 Fizzim

It is a FREE, open-source GUI-based FSM design tool. The GUI is written in java for portability. The backend code generation is written in perl for portability and ease of modification.
Features of GUI:

- Runs on Windows, Linux, Apple, anything with java.
- Familiar Windows look-and-feel.
- Visibility (on/off/only-non-default) and color control on data and comment fields.
- Multiple pages for complex state machines.
- "Output to clipboard" makes it easy to pull the state

diagram into your documentation.

Backend:

- Verilog code generation based on recommendations from experts in the field.
- Output code has "hand-coded" look-and-feel (no tasks, functions, etc).
- Switch between highly encoded or one hot output without changing the source.
- Registered outputs can be specified to be included as state bits, or pulled out as independent flops.
- Mealy and Moore outputs available.
- Transition priority available.
- Automatic grey coding available.
- Code and/or comments can be inserted at strategic places in the output - no need to "perl" the output to add your copyright or `include [3].

2.3 C code into Verilog

All C-to-Verilog is a free and open sourced on-line C to Verilog compiler. One could copy-and-paste an existing C code and our on-line compiler will synthesize it into optimized verilog. It also provides a free on-line service which allows users to compile their existing C code into optimized Verilog code. This code can be synthesized into an FPGA or ASIC.

C-to-Verilog.com is the result of an academic study in the field of high-level synthesis at Haifa University by Nadav Rotem. Their tools enable the development of embedded system-on-chip and integrate with design tools such as Xilinx EDK [4].

2.4 GTKWave Viewer

GTKWave is a fully featured GTK+ based waveform viewer which reads FST, LXT, LXT2, VZT, and GHW files as well as standard Verilog VCD/EVCD files and allows their viewing. GTKWave is developed for Linux, with ports for various other operating systems including Microsoft Windows (either natively as a Win32 application or via Cygwin), and Mac OS X targeting either X11 or Quartz. GTKWave is one of the applications affiliated with the open-source gEDA Project. The viewer supports both post-mortem viewing of VCD files and interactive viewing of VCD data, known as partial loading [5].

3 Icarus Verilog Simulator

Icarus Verilog is an open source synthesis and simulation tool that works as a compiler. It eventually compiles the source code which we write in Verilog (according to the IEEE-1364 standard) into a target format. The creator of this tool is Stephen Williams who is still working on enhancing its functionality. This tool also supports various operating systems such as Mac OS, Microsoft Windows, Linux, etc and is compliant with the IEEE 1364-2005 standard. This tool has been released in various versions such as version 0.9.0, 0.9.1, 0.9.2 etc. I will be focusing on the latest of all versions 0.9.6. These versions contain some released notes to keep track of the bugs associated with it. It also has a graphical support from GTKWave viewer that is being used for my study [6].

Characteristics or features of the tool:

- Simulation engine is efficient
- Portable compiler
- Challenge for commercial tools
- Supported graphics tool like GTKwave
- New compatibility with de facto standards such as library formats and command files

List of providers who offer commercial support for Icarus Verilog and/or related products are Dolly Software Private Limited, Embecosm, OCLogic Limited [6].

3.1 Version Control

A version control system which may also be known as a Revision Control System is a repository of files. These are the files used for the source code with monitored access. These codes are for computer programs. This is in general, for a good management of keeping track of the latest codes. It has a record of as to who made the changes to the code, when wre the changes made, why were these changes made. It also consists of the problems that have been fixed or the improvements that were made. Version control systems are essential for any form of distributed, collaborative development [7]. Git is one of the most widely used version control tool.

3.2 Types of Bugs in ICARUS Verilog Simulator

There is a searchable reporting system to track the problems [8]. The bugs present can have either of the two status; closed-fixed or open. If it is closed-fixed, then it means that these bugs have been looked after and are no more present in the tool. On the other hand, if it is an open bug, this means that it is still being dealt with and has not yet been fixed. Some of the open bugs in ICARUS Verilog v0.9.6 [9] are:

- No support for passing arrays (SystemVerilog),
- Can't set arrays of reals using VPI,
- No support for annotating specparams from SDF file,
- Module path delays do not properly filter pulses,
- Various attributes not parsed,
- VVP error when trying to $dumpvars real array, etc

Here, we will be discussing about one of the ICARUS Verilog bugs that say the tool does not support tri-reg nets.

4 Functionality for Tri-Reg Nets in Icarus Verilog V0.9.6

The Icarus Verilog V0.9.6 doesnot support the functionality for trireg nets. There are two different kinds of net types: built-in and user-defined. The *net* types can represent physical connections between structural entities, such as gates. A net shall not store a value (except for the ***trireg*** net). Instead, its value shall be determined by the values of its drivers, such as a continuous assignment or a gate. If no driver is connected to a net, its value shall be high-impedance (z) unless the net is a trireg, in which case it shall hold the previously driven value. The trireg net stores a value and is used to model charge storage nodes.

- *Medha Chhillar is currently pursuing masters degree program inVLSI Design in ITM University, India. E-mail: medha.chhillar@gmail.com*
- *Neeraj Kr. Shukla (IETE, IE, IACSIT, IAENG, CSI, ISTE, VSI-India), an Associate Professor in the Department of Electrical, Electronics & Communication Engineering, and Project Manager – VLSI Design at ITM University,India, E-mail: neerajkshukla@itmindia.edu*

A trireg net can be in one of two states:
i. ***Driven state*** When at least one driver of a trireg net has a value of 1, 0, or x, the resolved value propagates into the trireg net and is the trireg net's driven value.
ii. ***Capacitive state*** When all the drivers of a trireg net are at the high impedance value (z), the trireg net retains its last driven value; the high impedance value does not propagate from the driver to the trireg [10].
The basic idea to proceed in order to solve this issue is use the trireg net type along with its charge value, decay time, etc. and parse, elaborate and pass these values to the back end of the compiler (ivl interface).

5 CONCLUSION

Open Source eda tools have been studied along with their advantages over the closed source structures. It has also been justified as to why the open source is gaining more popularity. Eventually, we have discussed a few of the open source eda tools that are very popular among the industries. Also, main focus has been laid on the ICARUS Verilog simulation tool with some of the functional bugs appearing in the tool. We have also discussed one of the functional bugs and have suggested a possible solution to resolve the problem. These bugs are open to all and anyone can contribute to these tools.

ACKNOWLEDGMENT

I wish to thank Dr. Neeraj Kr. Shukla who has been a constant source of inspiration and a great guide to me. I also wish to thank Mr. Varad Gupta for inculcating in me the importance of open source in today’s scenario.

REFERENCES

[1] URL: http://www.semiwiki.com/forum/showwiki.php?title=Semi+Wiki:EDA+Open+Source+Tools+Wiki
[2] URL: http://www.gnu.org/software/emacs/
[3] URL: http://www.fizzim.com/
[4] URL: http://c-to-verilog.com/
[5] URL: http://6004.csail.mit.edu/6.371/cadtools/gtkwave/wave.html
[6] URL: http://iverilog.icarus.com/
[7] URL: http://oss-watch.ac.uk/resources/versioncontrol
[8] URL: http://iverilog.icarus.com/support/bugs
[9] URL: http://sourceforge.net/p/iverilog/bugs/
[10] IEEE Standard for SystemVerilog—Unified Hardware Design, Specification, and Verification Language, IEEE Std 1800™-2012. Revision of IEEE Std 1800-2009, IEEE Computer Society and the IEEE Standards Association Corporate Advisory Group, NewYork, USA, Feburary 21st 2013