# Free Open-Source Software (FOSS) in Engineering ©

**By**

**Raymond L. Barrett, Jr., PhD, PE**
**CEO, American Research and Development, LLC**

## 1.0 Introduction to "Free Open-Source Software" (FOSS)

This course introduces the engineer to the subject of "Free Open-Source Software" (FOSS). Quotes from various sources are included widely throughout the course. Each quote is indicated by the use of the *italic* fonts to set it off, as well as the quote marks used. Each quote is obtained, along with a "screen shot" of its source from the internet, as are the tools themselves. The spelling errors within the quotes are kept as in the source and not corrected or otherwise edited in any way. Although there is a rich history of free distribution of software, some commercial ventures have striven to eliminate the practice, providing proprietary commercial and often copyrighted software products as an alternative. Commercial software has its advantages with deep and wide support that is sometimes lacking in the open-source community. However, the price of commercial software is often a barrier-to-entry for emerging small and medium enterprise (SME). As a particular case exemplified in this course, the "Silicon Renaissance Initiative" was formed to address the high-cost of Engineering Design Automation (EDA) software tools and the design flow for that use is presented. As an alternative, the commercial tools often require hundreds of thousands of dollars investment to support the design effort. In the USA, with its copyright enforcement, the SME faces competition from foreign entities with "illegal" copies of commercial software and thus cannot compete. This course illustrates both the general tools available and useful for any computer user and professional engineer with any practice specialty, as well as the professional engineer involved in designing electronic circuits.

## 2.0 Free Open-Source Software History

In the following long quote from Wikipedia, we see that FOSS has been available from the early days of computing.

*"In the 1950s, 1960s, and 1970s, it was normal for computer users to have the freedoms that are provided by free software. Software was commonly shared by individuals who used computers and most companies were so concerned with selling their hardware devices, they provided the software for free.[8] Organizations of users and suppliers were formed to facilitate the exchange of software; see, for example, SHARE and DECUS. By the late 1960s change was inevitable: software costs were dramatically increasing, a growing software industry was competing with the hardware manufacturer's bundled software products (free in that the cost was included in the hardware cost), leased machines required software support*

*while providing no revenue for software, and some customers able to better meet their own needs did not want the costs of "free" software bundled with hardware product costs. In United States vs. IBM, filed 17 January 1969, the government charged that bundled software was anticompetitive.[9] While some software might always be free, there would be a growing amount of software that was for sale only. In the 1970s and early 1980s, the software industry began using technical measures (such as only distributing binary copies of computer programs) to prevent computer users from being able to use reverse engineering techniques to study and customize software they had bought. In 1980, the copyright law (Pub. L. No. 96-517, 94 Stat. 3015, 3028) was extended to computer programs in the United States[10]*

*In 1983, Richard Stallman, longtime member of the hacker community at the MIT Artificial Intelligence Laboratory, announced the GNU project, saying that he had become frustrated with the effects of the change in culture of the computer industry and its users.[11] Software development for the GNU operating system began in January 1984, and the Free Software Foundation (FSF) was founded in October 1985. An article outlining the project and its goals was published in March 1985 titled the GNU Manifesto. The manifesto also focused heavily on the philosophy of free software. He developed The Free Software Definition and the concept of "copyleft", designed to ensure software freedom for all.*

*The Linux kernel, started by Linus Torvalds, was released as freely modifiable source code in 1991. The licence wasn't exactly a free software licence, but with version 0.12 in February 1992, he relicensed the project under the GNU General Public License.[12] Much like Unix, Torvalds' kernel attracted the attention of volunteer programmers.*

*In 1997, Eric Raymond published The Cathedral and the Bazaar, a reflective analysis of the hacker community and free software principles. The paper received significant attention in early 1998, and was one factor in motivating Netscape Communications Corporation to release their popular Netscape Communicator Internet suite as free software. This code is today better known as Mozilla Firefox and Thunderbird.*

*Netscape's act prompted Raymond and others to look into how to bring free software principles and benefits to the commercial software industry. They concluded that FSF's social activism was not appealing to companies like Netscape, and looked for a way to rebrand the free software movement to emphasize the business potential of the sharing of source code. The new name they chose was "open source", and quickly Bruce Perens, publisher Tim O'Reilly, Linus Torvalds, and others signed on to the rebranding. The Open*

*Source Initiative was founded in February 1998 to encourage use of the new term and evangelize open source principles.*[13]

*While the Open Source Initiative sought to encourage the use of the new term and evangelize the principles it adhered to, corporations found themselves increasingly threatened by the concept of freely distributed software and universal access to an application's source code. A Microsoft executive publicly stated in 2001 that "open source is an intellectual property destroyer. I can't imagine something that could be worse than this for the software business and the intellectual-property business."* [14] *This view perfectly summarizes the initial response to FOSS by the majority of big business. However, while FOSS has historically played a role outside of the mainstream of private software development, companies as large as Microsoft have begun to develop official open source presences on the Internet. Corporations ranging from IBM, Oracle, Google and State Farm are just a few of the big names with a serious public stake in today's competitive open source market signalling a shift in the corporate philosophy concerning the development of free to access software.*[15]*"*

Again, as previously stated, the source of the quoted material above is from Wikipedia, located at:
http://en.wikipedia.org/wiki/Free_and_open-source_software

**3.0 The GNU Project**

The GNU Project, started by Richard Stallman is still thriving today and offers an extensive list of FOSS tools through their website. The following quote from the GNU website explains the background and details including a repeat of the history and present mission.

*"The GNU Project was launched in 1984 to develop the GNU system. The name "GNU" is a recursive acronym for "GNU's Not Unix!". "GNU" is pronounced* g'noo*, as one syllable, like saying "grew" but replacing the* r *with* n.

*A Unix-like operating system is a software collection of applications, libraries, and developer tools, plus a program to allocate resources and talk to the hardware, known as a kernel.*

*"Free software" is a matter of liberty, not price. To understand the concept, you should think of "free" as in "free speech", not as in "free beer".*

***Free software is a matter of the users' freedom to run, copy, distribute, study, change and improve the software.*** *More precisely, it refers to four kinds of freedom, for the users of the software:*

*The freedom to run the program, for any purpose (freedom 0).*

*The freedom to study how the program works, and adapt it to your needs (freedom 1). Access to the source code is a precondition for this.*

*The freedom to redistribute copies so you can help your neighbor (freedom 2).*

*The freedom to improve the program, and release your improvements to the public, so that the whole community benefits (freedom 3). Access to the source code is a precondition for this."*

The GNU Operating System - Mozilla Firefox
File Edit View History Bookmarks Tools Help
The GNU Operating System
www.gnu.org
AVG Secure Search
AVG Search... Search Safe Weather Facebook Speedtest
Translations of this page | Accessibility
GNU Operating System
Philosophy Licenses Education Downloads Documentation Help GNU Join the FSF!
Why GNU/Linux? Search
Ok
Download GNU now

Sign up for the Free Software Supporter
A monthly email newsletter about GNU and Free Software
Enter your email address (e.g. richard@example.com)

English [en] العربية [ar] български [bg] català [ca] česky [cs] Deutsch [de] ελληνικά [el] español [es] فارسی [fa] français [fr] Bahasa Indonesia [id] italiano [it] 日本語 [ja] ಕನ್ನಡ [kn] 한국어 [ko] norsk (bokmål) [nb] Nederlands [nl] polski [pl] português do Brasil [pt-br] română [ro] русский [ru] српски [sr] Türkçe [tr] українська [uk] 简体中文 [zh-cn]

What is GNU?

GNU is a Unix-like operating system that is free software—it respects your freedom. You can install Linux-based versions of GNU which are entirely free software.

Planet GNU

Compliance Lab in the news: Joshua Gay and I were recently interviewed about our new roles here at the Free Software Foundation. Joshua Gay and I were recently interviewed for an article by Bru... more

Show your support for the FSF with our new knife and cybertool: Gnu Press announces the availability of an FSF logo knife/cybertool including twelve tools. The Free Software Foundation shop is... more

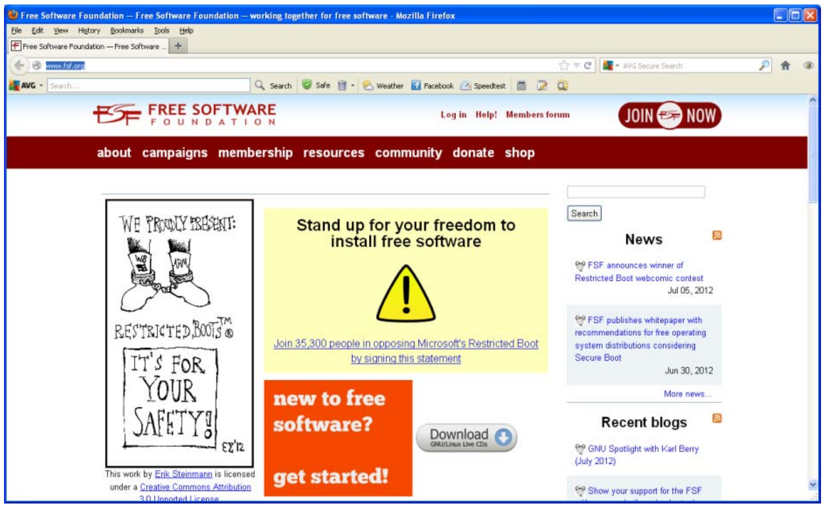
FSF announces winner of Restricted Boot webcomic contest: With over 30,000 signatures to our statement and over a dozen high-quality submissions to our contest, I'm confident our message that ... more

**Screenshot 3.0 - The GNU Project Website**

Again, as previously stated, the source of the quoted material above is from the GNU Project website, located at:
[http://www.gnu.org/](http://www.gnu.org/)

## 4.0 The Free Software Foundation

The Free Software Foundation is an advocacy group that promotes the use of FOSS and defends its use.


Free Software Foundation — Free Software Foundation — working together for free software - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Free Software Foundation — Free Software ...
www.fsf.org
AVG Secure Search
AVG Search... Search Safe Weather Facebook Speedtest
FREE SOFTWARE FOUNDATION
Log in Help! Members forum
JOIN NOW
about campaigns membership resources community donate shop
WE PROUDLY PRESENT:
RESTRICTED BOOTS™®
IT'S FOR YOUR SAFETY!
EZ'12
Search
News
Recent blogs
Download GNU/Linux Live CDs
new to free software?
get started!

Stand up for your freedom to install free software

Join 35,300 people in opposing Microsoft's Restricted Boot by signing this statement

This work by Erik Steinmann is licensed under a Creative Commons Attribution 3.0 Unported License

FSF announces winner of Restricted Boot webcomic contest Jul 05, 2012

FSF publishes whitepaper with recommendations for free operating system distributions considering Secure Boot Jun 30, 2012

More news...

GNU Spotlight with Karl Berry (July 2012)

Show your support for the FSF

**Screenshot 4.0 - The Free Software Foundation Website**

The following quote from the Free Software Foundation website details their mission.

*"The FSF advocates for free software ideals as outlined in the Free Software Definition, works for adoption of free software and free media formats, and organizes activist campaigns against threats to user freedom like Windows 7, Apple's iPhone and OS X, DRM on music, ebooks and movies, and software patents.*

*We promote completely free software distributions of GNU/Linux, and advocate that users of the GNU/Linux operating system switch to a distribution which respects their freedom.*

*We drive development of the GNU operating system and maintain a list of high-priority free software projects to promote replacements for common proprietary applications.*

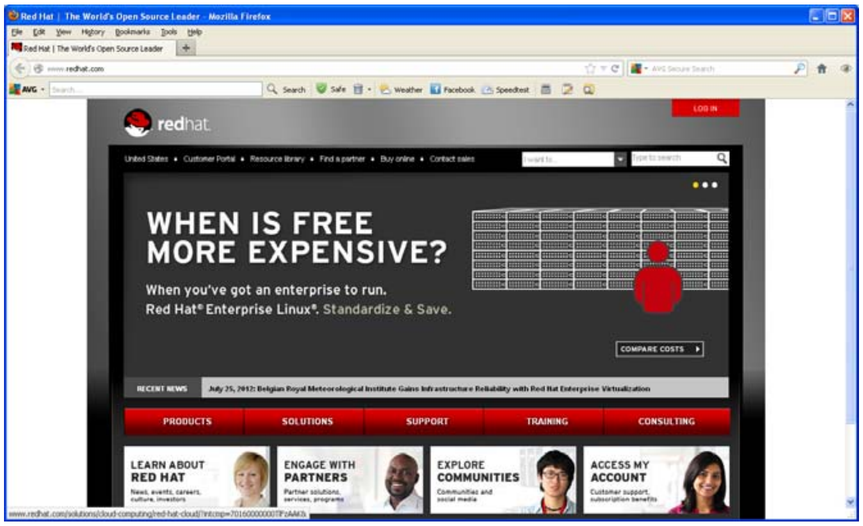
*We build and update resources useful for the free software community like the Free Software Directory, and the free software jobs board. We also provide licenses for free software developers to share their code, including the GNU General Public License."*

The Free Software Foundation website is located at:
http://www.fsf.org/

## 5.0 The Linux Operating System

On October 5, 1991, Linus Torvalds released the FOSS operating system kernel named "Linux" as a combination of his name Linus and the predecessor UNIX operating system. It was first released for the X86 computing platform, but has since been ported to numerous other platforms. Since that time, there have been multiple versions, including the GNU version above, that have been released to the public. Each version is associated with a different group of supporters, each with a different "business" model for supporting their activities. A few different versions of Linux that can be obtained as FOSS packages are listed below including a discussion of the associated business model.

5.1 Red Hat Linux – One version of Linux is distributed and supported by the commercial enterprise under the Red Hat logo. Although the Red Hat Linux software is fundamentally FOSS and can be downloaded and copied freely, the Red Hat organization offers training, consulting, and various forms of support to its client base on a fee basis.


Red Hat | The World's Open Source Leader - Mozilla Firefox
redhat.
WHEN IS FREE MORE EXPENSIVE?
When you've got an enterprise to run.
Red Hat® Enterprise Linux®. Standardize & Save.
COMPARE COSTS
PRODUCTS
SOLUTIONS
SUPPORT
TRAINING
CONSULTING
LEARN ABOUT RED HAT
ENGAGE WITH PARTNERS
EXPLORE COMMUNITIES
ACCESS MY ACCOUNT

**Screenshot 5.1 - The Red Hat Linux Website**

The Red Hat website is located at:
http://www.redhat.com/

The Red Hat value proposition is based on support as is evidenced by the following quotes from their website:

*"Red Hat® recognizes that IT managers need more than just software or support to be cost-efficient and in control of their IT infrastructures. An infrastructure based on Red Hat open source solutions not only meets customer application needs today, but will be an ideal platform well into the future."*

The support is obtained by paying a fee or subscription to Red Hat:

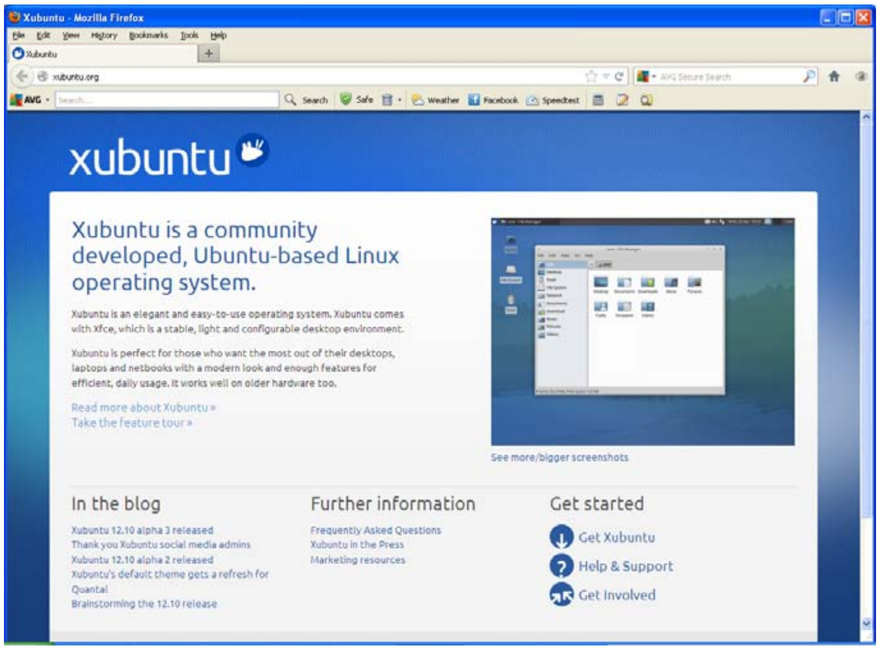
*"In addition to high-quality software and tools, a Red Hat subscription includes access to an around-the-clock global network of the most experienced, motivated, and knowledgeable Linux® and middleware support engineers in the industry. Virtually extending your in-house expertise to ensure you manage your IT with confidence. "*

The quotes were obtained from:
http://www.redhat.com/support/

5.2 Xubuntu Linux – Xubuntu is another well maintained FOSS version of Linux that does not offer the extensive support services..The mission and philosophy of the developers and maintainers of Xubuntu is expressed in the quotes below from their website.

*"Xubuntu is a community developed operating system that is well-suited for laptops and desktops. Whether you use it at home, at school or at work Xubuntu contains all the applications you'll ever need, from word processing and email applications, to web server software and programming tools.*

*Xubuntu is and always will be free of charge. You do not pay any licensing fees. You can download, use and share Xubuntu with your friends, family, school or business for absolutely nothing."*


Xubuntu - Mozilla Firefox
File Edit View History Bookmarks Tools Help
xubuntu.org
xubuntu

Xubuntu is a community developed, Ubuntu-based Linux operating system.

Xubuntu is an elegant and easy-to-use operating system. Xubuntu comes with Xfce, which is a stable, light and configurable desktop environment.

Xubuntu is perfect for those who want the most out of their desktops, laptops and netbooks with a modern look and enough features for efficient, daily usage. It works well on older hardware too.

Read more about Xubuntu »
Take the feature tour »

See more/bigger screenshots

In the blog
Xubuntu 12.10 alpha 3 released
Thank you Xubuntu social media admins
Xubuntu 12.10 alpha 2 released
Xubuntu's default theme gets a refresh for Quantal
Brainstorming the 12.10 release

Further information
Frequently Asked Questions
Xubuntu in the Press
Marketing resources

Get started
Get Xubuntu
Help & Support
Get Involved

**Screenshot 5.2 - The Xubuntu Linux Website**

The quotes and illustration were obtained from:
http://xubuntu.org/

5.3 Various Operating System Distributions – It should be noted that the Red Hat website has the .com suffix while the Xubuntu website has the .org suffix. The difference indicates that one organization intends to make a profit from distributing and supporting the FOSS effort while the other does not. There are a sufficient number of versions of Linux that a user may have reservations about selecting any one of them for a particular application. Indeed, the user may find that a particular application recommends one version of Linux to the exclusion of others. In addition, there are FOSS packages that are configured to run only on commercial operating system software such as a particular set of Microsoft's Windows versions.

## 6.0 FOSS VirtualBox Virtual OS Support

Because various software packages may each require a different operating system, solutions have been developed that provide virtual hardware environments to support multiple concurrent operating systems on one set of hardware. One such FOSS solution is VirtualBox.

*"Innotek initially offered the application under a proprietary software license. One version of the product was available at no cost for personal or evaluation use, subject to the VirtualBox Personal Use and Evaluation License (PUEL).[7] In January 2007, VirtualBox Open Source Edition (OSE) was released as free software, subject to the requirements of the GNU General Public License (GPL), version 2.[8]*

*The original developer, innotek, also contributed to the development of OS/2 and Linux support in virtualization[9] and OS/2 ports[10] of products from Connectix which were later acquired by Microsoft. Specifically, innotek developed the "additions" code in both Microsoft Virtual PC and Microsoft Virtual Server, which enables various host-guest OS interactions like shared clipboards or dynamic viewport resizing.*

*Sun Microsystems acquired innotek in February 2008.[11][12][13]*

*Oracle Corporation acquired Sun in January 2010 and re-branded the product as "Oracle VM VirtualBox".[14][15][16]"*

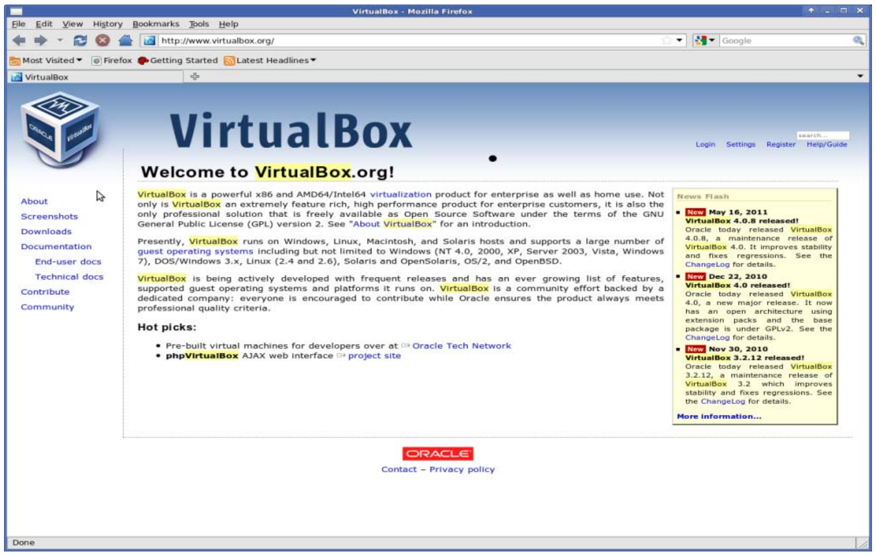
The quotes above about the history of VirtualBox were obtained from Wikipedia at:
http://en.wikipedia.org/wiki/VirtualBox

VirtualBox can still be obtained as FOSS from the VirtualBox.org website at:
https://www.virtualbox.org/

VirtualBox is described as:

*"VirtualBox is a general-purpose full virtualizer for x86 hardware, targeted at server, desktop and embedded use."*

A virtualizer is a software package that runs on one of several different "host" operating system software packages. VirtualBox lists several different Microsoft OS versions, Sun Solaris versions, Linux versions, as well as Unix itself. The present owner of VirtualBox, the Oracle corporation, offers several different pre-built versions of "guest" operating systems that can be loaded into VirtualBox. As an alternative, the user can load or compile their own choices as desired according to the instructions associated with a specific guest OS candidate.


VirtualBox - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.virtualbox.org/
Google
Most Visited Firefox Getting Started Latest Headlines
VirtualBox
VirtualBox
search...
Login Settings Register Help/Guide
About
Screenshots
Downloads
Documentation
End-user docs
Technical docs
Contribute
Community
ORACLE
Contact – Privacy policy
Done

Welcome to VirtualBox.org!

VirtualBox is a powerful x86 and AMD64/Intel64 virtualization product for enterprise as well as home use. Not only is VirtualBox an extremely feature rich, high performance product for enterprise customers, it is also the only professional solution that is freely available as Open Source Software under the terms of the GNU General Public License (GPL) version 2. See "About VirtualBox" for an introduction.

Presently, VirtualBox runs on Windows, Linux, Macintosh, and Solaris hosts and supports a large number of guest operating systems including but not limited to Windows (NT 4.0, 2000, XP, Server 2003, Vista, Windows 7), DOS/Windows 3.x, Linux (2.4 and 2.6), Solaris and OpenSolaris, OS/2, and OpenBSD.

VirtualBox is being actively developed with frequent releases and has an ever growing list of features, supported guest operating systems and platforms it runs on. VirtualBox is a community effort backed by a dedicated company: everyone is encouraged to contribute while Oracle ensures the product always meets professional quality criteria.

Hot picks:

- Pre-built virtual machines for developers over at Oracle Tech Network
- **phpVirtualBox** AJAX web interface project site

News Flash

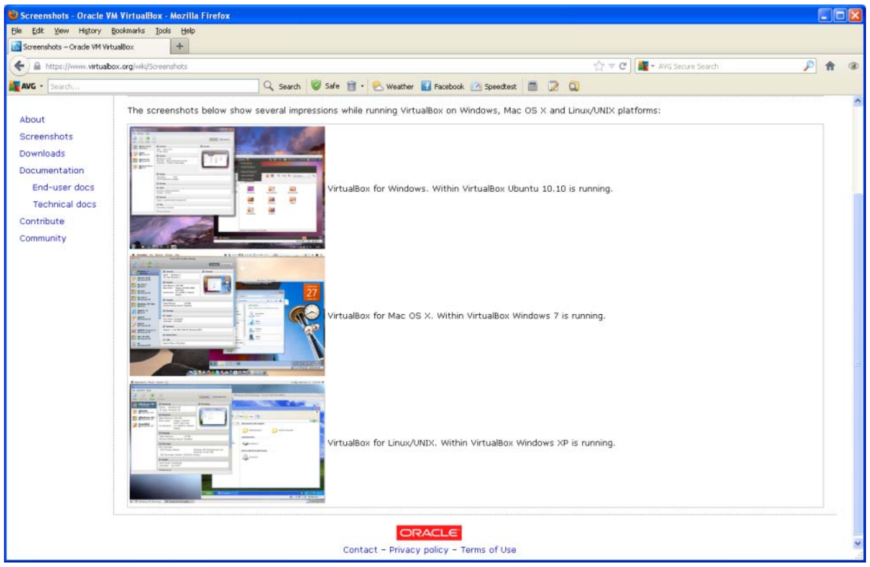
- New **May 16, 2011**
  **VirtualBox 4.0.8 released!**
  Oracle today released VirtualBox 4.0.8, a maintenance release of VirtualBox 4.0. It improves stability and fixes regressions. See the ChangeLog for details.
- New **Dec 22, 2010**
  **VirtualBox 4.0 released!**
  Oracle today released VirtualBox 4.0, a new major release. It now has an open architecture using extension packs and the base package is under GPLv2. See the ChangeLog for details.
- New **Nov 30, 2010**
  **VirtualBox 3.2.12 released!**
  Oracle today released VirtualBox 3.2.12, a maintenance release of VirtualBox 3.2 which improves stability and fixes regressions. See the ChangeLog for details.

**More information...**

**Screenshot 6.1 - The VirtualBox Website**

The quote describing VirtualBox and the illustration were obtained from:
https://www.virtualbox.org/wiki/VirtualBox

Several example deployments using various host OS examples with different guest examples is shown above in the illustration below obtained from the site at:
https://www.virtualbox.org/wiki/Screenshots


Screenshots - Oracle VM VirtualBox - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://www.virtualbox.org/wiki/Screenshots
About
Screenshots
Downloads
Documentation
End-user docs
Technical docs
Contribute
Community
ORACLE
Contact – Privacy policy – Terms of Use

The screenshots below show several impressions while running VirtualBox on Windows, Mac OS X and Linux/UNIX platforms:

VirtualBox for Windows. Within VirtualBox Ubuntu 10.10 is running.

VirtualBox for Mac OS X. Within VirtualBox Windows 7 is running.

VirtualBox for Linux/UNIX. Within VirtualBox Windows XP is running.

**Screenshot 6.2 – Several VirtualBox Deployment Examples**

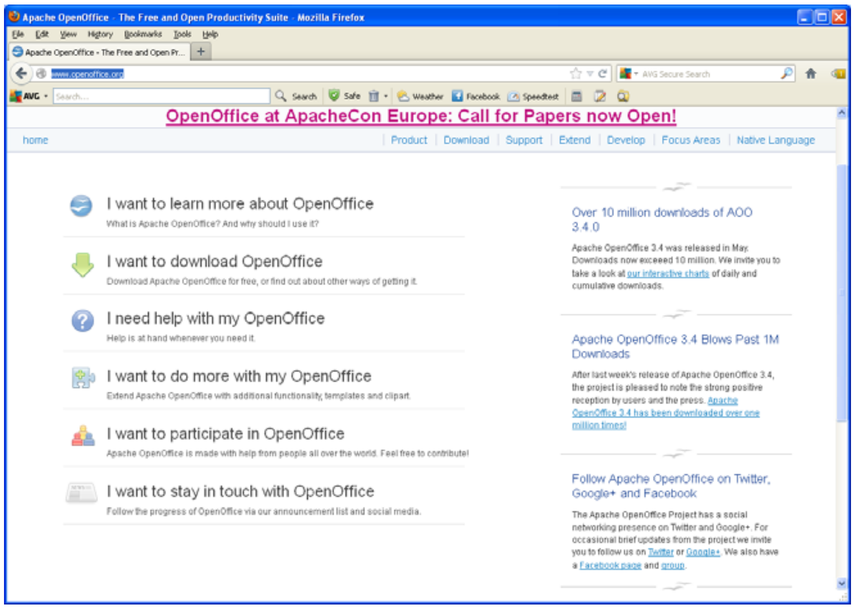
Two advantages of using a "virtualizer" software package are:

1) "Drag & Drop" features are often supported between differing OS guests, and
2) Several OS software copies can be guests concurrently to isolate new applications.

## 7.0 OpenOffice Office Software Suite

The OpenOffice software suite is comprised of a word processing application, a spreadsheet application, a presentation application, a graphics application, and a database application, as well as a number of smaller tools. It provides nominal file exchange with multiple other tools, including corresponding popular products from Microsoft and others.

Deployed on different guest OS instances, it supports drag-and-drop file exchanges between different guest OS windows in VirtualBox. The suite of applications are useful for documentation and design of engineering projects despite the fact that they are not provided specifically for engineering activities.


Apache OpenOffice - The Free and Open Productivity Suite - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Apache OpenOffice - The Free and Open Pr...
www.openoffice.org
AVG Secure Search
AVG Search... Search Safe Weather Facebook Speedtest
home Product Download Support Extend Develop Focus Areas Native Language

OpenOffice at ApacheCon Europe: Call for Papers now Open!

I want to learn more about OpenOffice
What is Apache OpenOffice? And why should I use it?

I want to download OpenOffice
Download Apache OpenOffice for free, or find out about other ways of getting it.

I need help with my OpenOffice
Help is at hand whenever you need it.

I want to do more with my OpenOffice
Extend Apache OpenOffice with additional functionality, templates and clipart.

I want to participate in OpenOffice
Apache OpenOffice is made with help from people all over the world. Feel free to contribute!

I want to stay in touch with OpenOffice
Follow the progress of OpenOffice via our announcement list and social media.

Over 10 million downloads of AOO 3.4.0
Apache OpenOffice 3.4 was released in May. Downloads now exceeed 10 million. We invite you to take a look at our interactive charts of daily and cumulative downloads.

Apache OpenOffice 3.4 Blows Past 1M Downloads
After last week's release of Apache OpenOffice 3.4, the project is pleased to note the strong positive reception by users and the press. Apache OpenOffice 3.4 has been downloaded over one million times!

Follow Apache OpenOffice on Twitter, Google+ and Facebook
The Apache OpenOffice Project has a social networking presence on Twitter and Google+. For occasional brief updates from the project we invite you to follow us on Twitter or Google+. We also have a Facebook page and group.

**Screenshot 7.0 – Apache's OpenOffice Suite**

Apache describes the OpenOffice suite in the following quote:

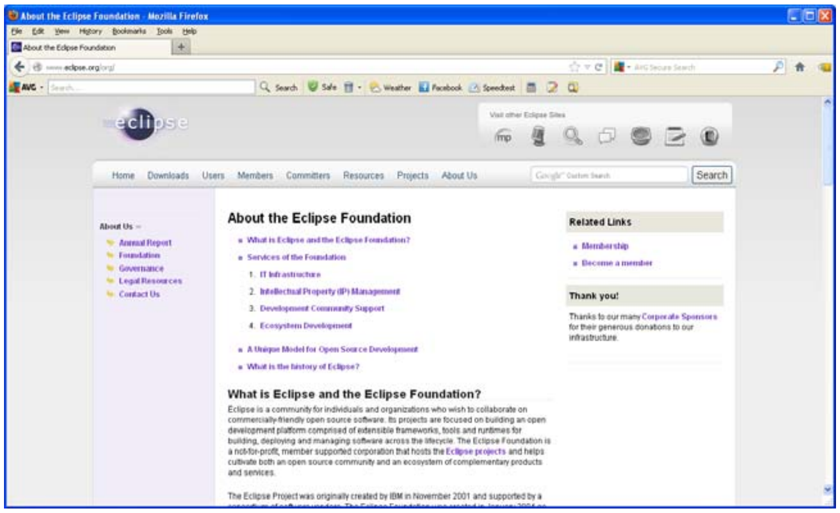
*"Apache OpenOffice is the leading* ***open-source office software suite*** *for* ***word processing****,* ***spreadsheets****,* ***presentations****,* ***graphics****,* ***databases*** *and more. It is available in* ***many languages*** *and works on all* ***common computers****. It stores all your data in an* ***international open standard format*** *and can also read and write files from other common office software packages. It can be downloaded and used completely* ***free of charge*** *for* ***any purpose****."*

The quote and illustration were obtained from the website at:
http://www.openoffice.org/

## 8.0 Eclipse Software Development Tools

The Eclipse set of software development tools are useful to the user needing to enter and debug programming code, compile that code, and document a project involving computer code. Such tools are directly applicable to the newly recognized "Software Engineer" Professional Engineering discipline, as well as the general computer-user community. The

Eclipse tools can be obtained directly from the Eclipse Foundation, or may be found included in many FOSS tools that require some element of computer program development.


About the Eclipse Foundation - Mozilla Firefox
File Edit View History Bookmarks Tools Help
About the Eclipse Foundation
www.eclipse.org/org/
AVG Search Safe Weather Facebook Speedtest
AVG Secure Search
eclipse
Visit other Eclipse Sites
Home Downloads Users Members Committers Resources Projects About Us
Search

About Us

- Annual Report
- Foundation
- Governance
- Legal Resources
- Contact Us

About the Eclipse Foundation

- What is Eclipse and the Eclipse Foundation?
- Services of the Foundation
  1. IT Infrastructure
  2. Intellectual Property (IP) Management
  3. Development Community Support
  4. Ecosystem Development
- A Unique Model for Open Source Development
- What is the history of Eclipse?

What is Eclipse and the Eclipse Foundation?

Eclipse is a community for individuals and organizations who wish to collaborate on commercially-friendly open source software. Its projects are focused on building an open development platform comprised of extensible frameworks, tools and runtimes for building, deploying and managing software across the lifecycle. The Eclipse Foundation is a not-for-profit, member supported corporation that hosts the Eclipse projects and helps cultivate both an open source community and an ecosystem of complementary products and services.

The Eclipse Project was originally created by IBM in November 2001 and supported by a

Related Links

- Membership
- Become a member

Thank you!

Thanks to our many Corporate Sponsors for their generous donations to our infrastructure.

**Screenshot 8.0 – Eclipse Foundation Web Page**

The Eclipse Foundation describes its mission and projects in the following quote:

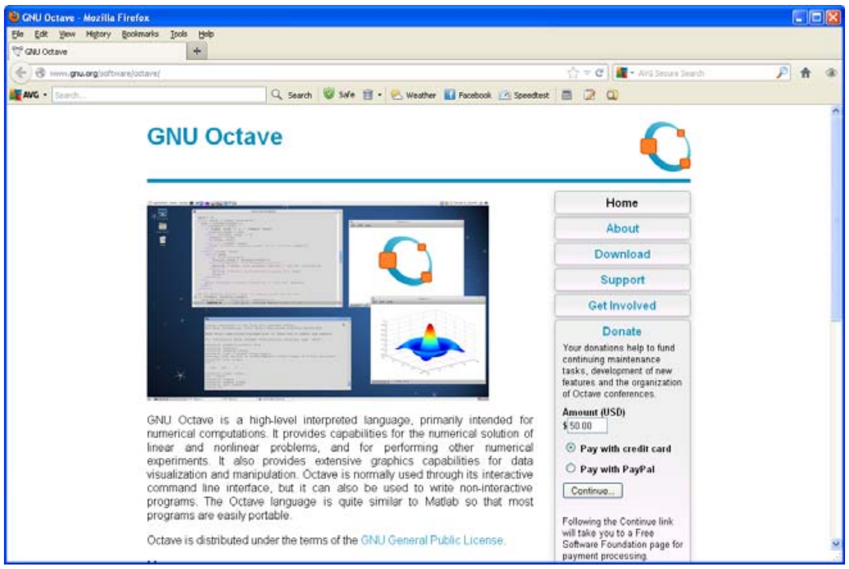
*"Eclipse is a community for individuals and organizations who wish to collaborate on commercially-friendly open source software. Its projects are focused on building an open development platform comprised of extensible frameworks, tools and runtimes for building, deploying and managing software across the lifecycle. The Eclipse Foundation is a not-for-profit, member supported corporation that hosts the Eclipse projects and helps cultivate both an open source community and an ecosystem of complementary products and services.*

*The Eclipse Project was originally created by IBM in November 2001 and supported by a consortium of software vendors. The Eclipse Foundation was created in January 2004 as an independent not-for-profit corporation to act as the steward of the Eclipse community. The independent not-for-profit corporation was created to allow a vendor neutral and open, transparent community to be established around Eclipse. Today, the Eclipse community consists of individuals and organizations from a cross section of the software industry."*

The Eclipse quote and illustration were obtained from the website:
http://www.eclipse.org/org/

## 9.0 GNU Octave Mathematical Modeling and Graphing Tools

The GNU Octave Tools are built around a specialized programming language. The tools are not written specifically for engineering applications but are extremely useful to many engineering disciplines involving numerical calculations and display as evidenced by the citation that it was originally developed for chemical engineering problem solving.


GNU Octave - Mozilla Firefox
File Edit View History Bookmarks Tools Help
GNU Octave
www.gnu.org/software/octave/
AVG Secure Search
AVG Search Safe Weather Facebook Speedtest
Home
About
Download
Support
Get Involved

GNU Octave

GNU Octave is a high-level interpreted language, primarily intended for numerical computations. It provides capabilities for the numerical solution of linear and nonlinear problems, and for performing other numerical experiments. It also provides extensive graphics capabilities for data visualization and manipulation. Octave is normally used through its interactive command line interface, but it can also be used to write non-interactive programs. The Octave language is quite similar to Matlab so that most programs are easily portable.

Octave is distributed under the terms of the GNU General Public License.

Donate

Your donations help to fund continuing maintenance tasks, development of new features and the organization of Octave conferences.

Amount (USD)
$ 50.00

[✓ Pay with credit card]
Pay with PayPal

Continue...

Following the Continue link will take you to a Free Software Foundation page for payment processing.

**Screenshot 9.0 – GNU Octave Web Page**

The GNU Octave web page can be found at:
http://www.gnu.org/software/octave/

Octave is one of many GNU Project applications but is specifically described in the quotes below.

*"GNU Octave is a high-level language, primarily intended for numerical computations. It provides a convenient command line interface for solving linear and nonlinear problems numerically, and for performing other numerical experiments using a language that is mostly compatible with Matlab. It may also be used as a batch-oriented language.*

*Octave has extensive tools for solving common numerical linear algebra problems, finding the roots of nonlinear equations, integrating ordinary functions, manipulating polynomials, and integrating ordinary differential and differential-algebraic equations. It is easily extensible and customizable via user-defined functions written in Octave's own language, or using dynamically loaded modules written in C++, C, Fortran, or other languages.*

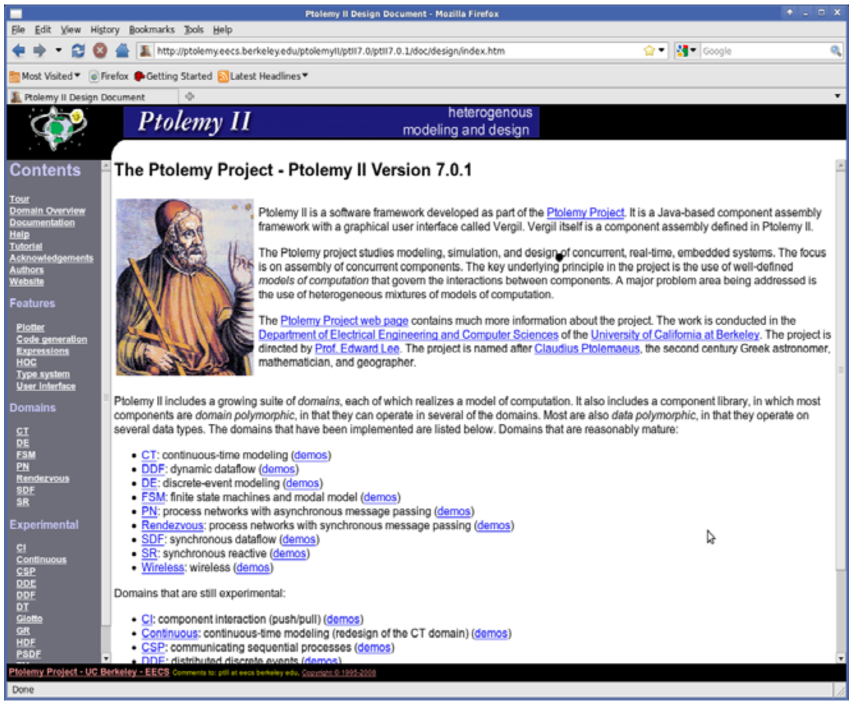
*GNU Octave is also freely redistributable software. You may redistribute it and/or modify it under the terms of the GNU General Public License (GPL) as published by the Free Software Foundation."*

The quote describing Octave was obtained from the website:
http://www.gnu.org/software/octave/about.html

## 10.0 Ptolemy II Modeling, Simulation, and System Development Tools

Ptolemy II is a general-purpose system-level modeling and simulation tool developed by a group at the University of California at Berkeley.

The Ptolemy II software system is the extensive product of many years of research and development into concurrent models of computation used to describe systems of very diverse components. It was originally written in the C language and evolved through generations including SystemC implementations into its present form written in the Java language. In that regard, Ptolemy II is platform independent, requiring Java support by the operating system. Fortunately, Java is available as a virtual machine and ordinarily does not require compilation but can be install simply if obtained from the website:
http://www.java.com/en/download/index.jsp


Ptolemy II Design Document - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://ptolemy.eecs.berkeley.edu/ptolemyII/ptII7.0/ptII7.0.1/doc/design/index.htm
Google
Most Visited Firefox Getting Started Latest Headlines
Ptolemy II Design Document
Ptolemy II
heterogenous modeling and design
Contents
Tour
Domain Overview
Documentation
Help
Tutorial
Acknowledgements
Authors
Website
Features
Plotter
Code generation
Expressions
HOC
Type system
User Interface
Domains
CT
DE
FSM
PN
Rendezvous
SDF
SR
Experimental
CI
Continuous
CSP
DDE
DDF
DT
Giotto
GR
HDF
PSDF
Ptolemy Project - UC Berkeley - EECS
Comments to: ptII at eecs berkeley edu. Copyright © 1995-2008
Done

The Ptolemy Project - Ptolemy II Version 7.0.1

Ptolemy II is a software framework developed as part of the Ptolemy Project. It is a Java-based component assembly framework with a graphical user interface called Vergil. Vergil itself is a component assembly defined in Ptolemy II.

The Ptolemy project studies modeling, simulation, and design of concurrent, real-time, embedded systems. The focus is on assembly of concurrent components. The key underlying principle in the project is the use of well-defined *models of computation* that govern the interactions between components. A major problem area being addressed is the use of heterogeneous mixtures of models of computation.

The Ptolemy Project web page contains much more information about the project. The work is conducted in the Department of Electrical Engineering and Computer Sciences of the University of California at Berkeley. The project is directed by Prof. Edward Lee. The project is named after Claudius Ptolemaeus, the second century Greek astronomer, mathematician, and geographer.

Ptolemy II includes a growing suite of *domains*, each of which realizes a model of computation. It also includes a component library, in which most components are *domain polymorphic*, in that they can operate in several of the domains. Most are also *data polymorphic*, in that they operate on several data types. The domains that have been implemented are listed below. Domains that are reasonably mature:

- CT: continuous-time modeling (demos)
- DDF: dynamic dataflow (demos)
- DE: discrete-event modeling (demos)
- FSM: finite state machines and modal model (demos)
- PN: process networks with asynchronous message passing (demos)
- Rendezvous: process networks with synchronous message passing (demos)
- SDF: synchronous dataflow (demos)
- SR: synchronous reactive (demos)
- Wireless: wireless (demos)

Domains that are still experimental:

- CI: component interaction (push/pull) (demos)
- Continuous: continuous-time modeling (redesign of the CT domain) (demos)
- CSP: communicating sequential processes (demos)
- DDE: distributed discrete events (demos)

**Screenshot 10.0 – Ptolemy II Web Page**

The Ptolemy II Project web page is found at:
http://ptolemy.eecs.berkeley.edu/

As for Ptolemy II, the tools are available in several generations including development versions by following the links from the Ptolemy II web page designated above.

*"The Ptolemy project studies modeling, simulation, and design of concurrent, real-time, embedded systems. The focus is on assembly of concurrent components. The key underlying principle in the project is the use of well-defined models of computation that govern the interaction between components. A major problem area being addressed is the use of heterogeneous mixtures of models of computation. A software system called Ptolemy II is being constructed in Java.*

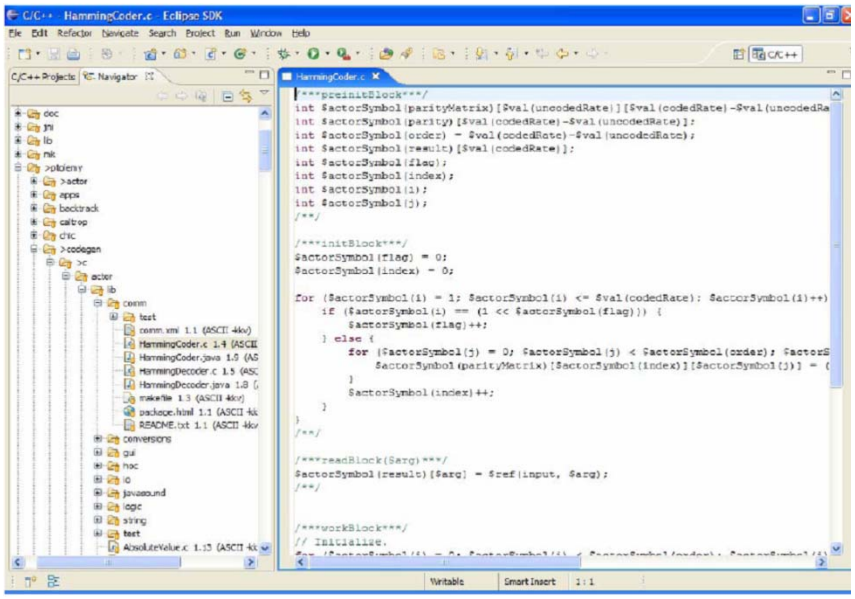
*The class of systems addressed by the project is sometimes called reactive systems. Reactive systems are those that interact with their environment at the speed of the environment. They* *are often embedded systems, and have been contrasted with interactive systems, which react with the environment at their own speed, and transformational systems, which process a body of input data to produce a body of output data. Reactive systems typically include elements of signal processing, communications, and real-time control. They are typically implemented with mixed technologies, possibly including embedded software, custom digital hardware, configurable hardware, analog circuits, microwave circuits, and microelectromechanical systems (MEMS).*

*A key principle in the Ptolemy project is the use of multiple models of computation in a hierarchical heterogeneous design environment. A premise in this work is that no single general purpose model of computation is likely to emerge in the near future that will deliver what designers need. Modeling the diverse implementation technologies and their interaction is not reasonable within a homogeneous environment. Consider embedded DSP software controlling a MEMS device, for example. The MEMS device requires physical modeling using differential equations, a modeling technique that is entirely inappropriate for the DSP software. Moreover, the ability to validate designs and to synthesize high-quality implementations from high-level abstract models are both compromised by generality in the modeling framework.*

*The project aims to develop techniques supporting heterogeneous modeling, including both formal "meta-models" and a software laboratory for experimenting with heterogeneous modeling. In this context, it has explored methods based on dataflow and process networks, discrete-event systems, synchronous/reactive languages, finite-state machines, and communicating sequential processes. It has made contributions ranging from fundamental semantics to synthesis of embedded software and custom hardware."*

The Ptolemy II Project descriptive quotes are found at the web page:
http://ptolemy.eecs.berkeley.edu/

Embedded within the Ptolemy II system, and typical of many FOSS tools is a re-use of the Eclipse SDK tool for compiling C/C++ programming code. The Eclipse tools provide such a useful programming environment that it is natural to re-use it as a common platform. In this, as in many environments, the Eclipse tools are directed to use a specific compiler for the language of the program. In that context, it is also common for the Eclipse tools to utilize the GNU Project compilers.


C/C++ - HammingCoder.c - Eclipse SDK
File Edit Refactor Navigate Search Project Run Window Help
C/C++ Projects
Navigator
HammingCoder.c
C/C++
Writable
Smart Insert
1 : 1

```
/***preinitBlock***/
int $actorSymbol(parityMatrix)[$val(uncodedRate)][$val(codedRate)-$val(uncodedRa
int $actorSymbol(parity)[$val(codedRate)-$val(uncodedRate)];
int $actorSymbol(order) = $val(codedRate)-$val(uncodedRate);
int $actorSymbol(result)[$val(codedRate)];
int $actorSymbol(flag);
int $actorSymbol(index);
int $actorSymbol(i);
int $actorSymbol(j);
/**/

/***initBlock***/
$actorSymbol(flag) = 0;
$actorSymbol(index) = 0;

for ($actorSymbol(i) = 1; $actorSymbol(i) <= $val(codedRate); $actorSymbol(i)++)
    if ($actorSymbol(i) == (1 << $actorSymbol(flag))) {
        $actorSymbol(flag)++;
    } else {
        for ($actorSymbol(j) = 0; $actorSymbol(j) < $actorSymbol(order); $actorS
            $actorSymbol(parityMatrix)[$actorSymbol(index)][$actorSymbol(j)] = (
        }
        $actorSymbol(index)++;
    }
}
/**/

/***readBlock($arg)***/
$actorSymbol(result)[$arg] = $ref(input, $arg);
/**/


/***workBlock***/
// Initialize.
```

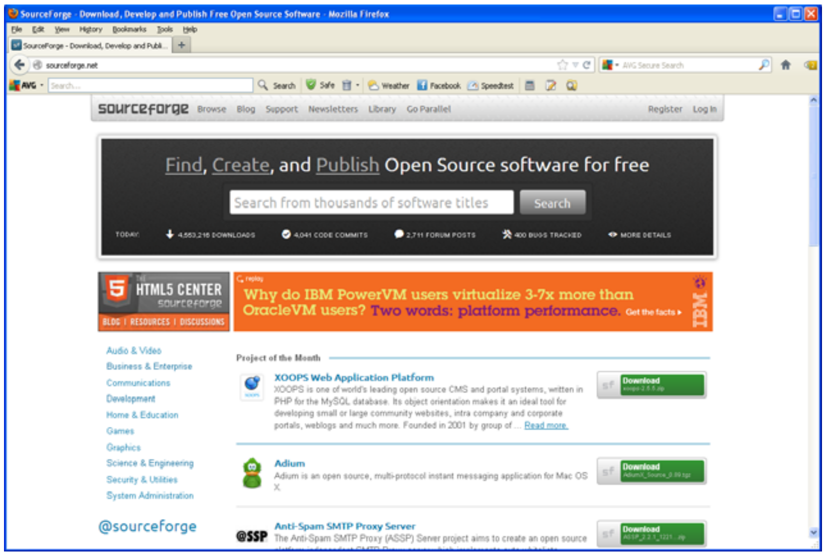
**Screenshot 10.1 – Ptolemy II Hamming Coder Showing the Eclipse SDK**

As shown, within the Ptolemy II system, we found the Eclipse tools embedded to support the entry of the C/C++ source code for the Hamming Code example. Typically, the GNU compiler tools are employed with Eclipse in Ptolemy II to compile the source code to the target system’s object code, whether that information is disclosed explicitly or not.

## 11.0 SourceForge Repository for FOSS Software

A wide range of FOSS software is available from the sourceforge repository. The categories are clear, but the very large number of applications packages leaves the casual investigator often bewildered as to the value of each entry. We see, however, that many of the applications shown in this course are listed on their website as available from sourceforge.

Sourceforge does not provide any guarantees of quality or suitability, but it is reasonably well monitored to prevent distribution of “malware” that is harmful to the user.


SourceForge - Download, Develop and Publish Free Open Source Software - Mozilla Firefox
sourceforge.net
sourceforge Browse Blog Support Newsletters Library Go Parallel Register Log In
Find, Create, and Publish Open Source software for free
Search from thousands of software titles
Search
TODAY: 4,553,216 DOWNLOADS 4,041 CODE COMMITS 2,711 FORUM POSTS 400 BUGS TRACKED MORE DETAILS
HTML5 CENTER sourceforge BLOG | RESOURCES | DISCUSSIONS
Why do IBM PowerVM users virtualize 3-7x more than OracleVM users? Two words: platform performance. Get the facts IBM
Audio & Video
Business & Enterprise
Communications
Development
Home & Education
Games
Graphics
Science & Engineering
Security & Utilities
System Administration
@sourceforge
Project of the Month
Download
Download
Download

XOOPS Web Application Platform
XOOPS is one of world's leading open source CMS and portal systems, written in PHP for the MySQL database. Its object orientation makes it an ideal tool for developing small or large community websites, intra company and corporate portals, weblogs and much more. Founded in 2001 by group of ... Read more.

Adium
Adium is an open source, multi-protocol instant messaging application for Mac OS X.

Anti-Spam SMTP Proxy Server
The Anti-Spam SMTP Proxy (ASSP) Server project aims to create an open source

**Screenshot 11.0 – Sourceforge FOSS Repository Web Page**

The illustration above was obtained from the website at:
http://sourceforge.net/

## 12.0 Silicon Renaissance Initiative: FOSS Specific to IC Design Engineering

The "Silicon Renaissance Initiative" is an informal, grass-roots effort of professionals and academics concerned with the decline of small and medium enterprise innovation in the engineering of Integrated Circuits (ICs) in the USA.

Certainly, the engineering community is aware of the great strides in computers, cell-phones, and other forms of electronics that has been made possible by the continued evolution of IC technology. There is a famous observation by Gordon Moore of Intel that advances in IC technology enable a "shrinking" in the lithographic technology associated with the fabrication of advanced IC devices to the tune of a halving of critical dimensions that can be constructed every 18 months. The observation has gotten the nickname of "Moore's Law" of IC technology. The "Law" has held true for more than five decades and continues still.

Associated with the shrinking of dimensions, there are "corollaries" that are consequences of that dimensional shrinking. One corollary is that the clock speed of digital circuits essentially doubles each time the dimensions shrink by one-half. That combination of speed increase with a decrease in cost per device has enabled the rapid improvement of a number of industries dependent on IC technology. We often hear examples of today's calculator being equivalent to a mainframe computer of several generations ago.

IC computer size, in terms of digital data bus width, has doubled regularly from that of the 4004, Intel's first 4-bit microprocessor to multi-core, 64-bit machines of today. That increase in data bus bandwidth, coupled with regular increases in speed has made today's microprocessors have truly amazing performance.

Coincident with the development of the computer hardware, the size of programs has increased, albeit mostly because of the available memory space and not because the software itself has gotten much better. The IC designer uses Engineering Design Automation (EDA) tools with algorithms that were developed in the 1960's on mainframe and minicomputer hardware. The IC EDA industry as exemplified by such entities as Cadence, Mentor Graphics, and others have improved the tools slowly, depending more on the hardware for performance increases.

Along with the constant lithographic shrinking of IC dimensions is the pressure placed on the IC fabricator in terms of constantly replacing or upgrading equipment every 18 months. The need for the fabricator or "fab" to invest in new technology knowing that the essential payback is constrained to a few generations of IC dimensions before the equipment is obsolete, and coupled with the ever-increasing cost of the replacement equipment has led to the dominance of a few very large IC fabrication facilities world-wide.

London's Economist Magazine did a profile of the IC industry of 2007 and showed that there were ~40 new IC fabrication facilities under construction world-wide (The Economist, April 4th-10th, 2009, "Briefing on The Semiconductor Industry," pp 71-73). Three were in the USA, two were in Europe, and the remaining 35 were in the "Far-East." Of the 35 under construction in the Far-East, the Economist article concluded that the construction was brought about mostly by perceptions of "national interest" at the new sites. One conclusion, seemingly borne-out by experience, is that there is now an over-capacity of fabrication facilities world-wide. Another conclusion of the article was that small and medium enterprises (SME) could take advantage of the over-capacity using a fab-less design business model and specialize in IC innovation.

Unfortunately, there are three major barriers-to-entry for SME IC design enterprises:

1) the cost of EDA software tools (much in excess of $50k per designer),
2) the cost of fabrication access through a large fabrication facility, and
3) the cost of verification and production testing of advanced IC products.

In the USA, the commercial IC EDA companies have brought about a situation of "Unintended Consequences" that the Silicon Renaissance Initiative is attempting to address. EDA tools are made available to the university community for a few thousand dollars per year. The university community then uses the tools to essentially "train" the students in the use of the tools. Licensing restrictions makes the tools available for on-campus use only. The algorithms, data structures, and programs are supplied without access to the source code for students to examine. On graduation, the students find that their only source of employment is in a large enterprise that can afford the cost of the tools.

Any traveler to major cities in China and India can tell of the availability of illegal "bootleg" or "pirated" copies of the IC EDA tools for a few dollars, leading to the conclusion that SME organizations in those countries have an economic advantage relative to that cost item.

Free, Open-Source Software (FOSS) versions of EDA software substantially reduce the cost of entry to SME organizations. Also, introduction of those same tools in the university environment exposes the student to both the Computer Science aspects of the EDA tools as well as the IC engineering design issues. The student is better "educated" but less well "trained."

The issue of access to fabrication facilities can be addressed, at least for prototype quantities, using the MOSIS or CMP multi-project wafer fabrication services. Multiple designs share the cost of fabrication masks and the processing of a few wafers. Finally, the cost of verification and production testing is also addressed as part of the Silicon Renaissance Initiative with the introduction of "Open-Source" tester designs based on the acquisition of designs compatible with the standard National Instruments card cages.

Although much of this discussion is specific to the IC design engineer, it does make the case for the use of FOSS EDA tools in an engineering discipline. The development of the discussion begins with a tool suit that is useful to general design of electronic circuits through the fabrication of printed wiring boards. That subject matter is relevant to any electronics project regardless of further development of an integrated realization. The further remainder of the course discusses the set of FOSS EDA tools specific to the IC design engineer as a more specialized example. Enough other FOSS tools are available for a concerned engineer to prepare a design support tool flow for other disciplines, but there is insufficient space to cover all disciplines in this course. Components that include Micro-Electro-Mechanical Systems (MEMS) technology such as sensors and valves that are realized using IC technology are supported by these tools but are not discussed in the following sections.

## 13.0 gEDA Project Electronic Design Automation (EDA) Tool Suite

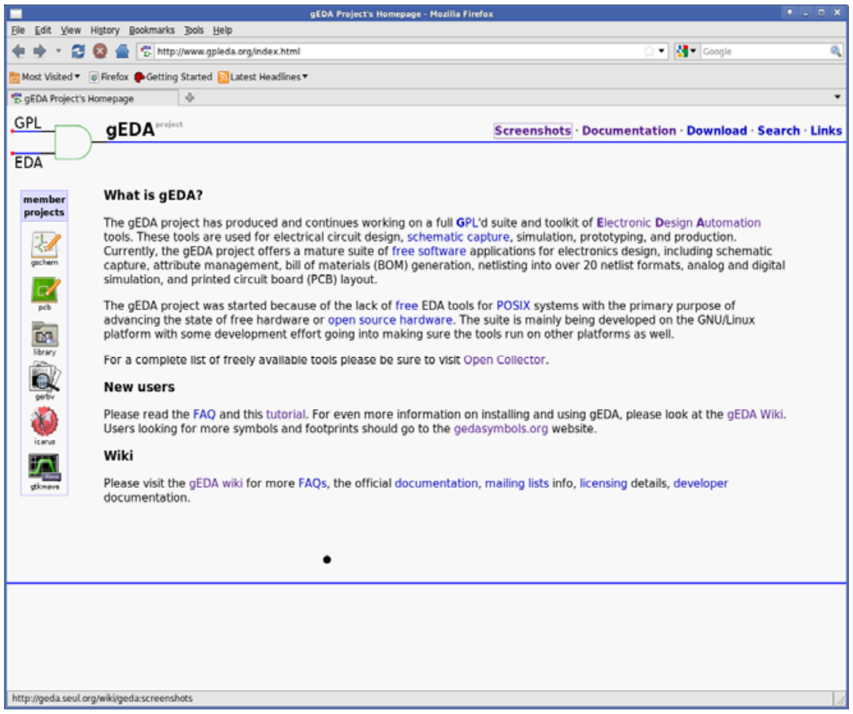
The gEDA Project was started in 1998 to support the design of electronics hardware. It has matured from its original suite of tools alone into an organization that provides the basis for user-contributed EDA software written for electronics hardware design.

*"The gEDA project is developing a full GPL'd suite and toolkit of Electronic Design Automation tools. These tools are used for electrical circuit design, schematic capture, simulation, prototyping, and production. Currently, the gEDA project offers a mature suite of free software applications for electronics design, including schematic capture, attribute management, bill of materials (BOM) generation, netlisting into over 20 netlist formats, analog and digital simulation, and printed circuit board (PCB) layout.*

*The tools involved in the suite enable you to professional-quality design of low- to mid-level complexity. Using the gEDA tools, you can create PCB of up to 8 layers (soon more) with an unlimited number of components and nets. The tools are suitable for use by students, educators, hobbyists, consultants, small businesses, and even in large corporations where an engineer might need to crank out a quick PC board (e.g. for a test stand) in a hurry."*

The quote is taken from the gEDA wiki page at:
http://wiki.geda-project.org/

The tools provided through the gEDA Project, **gschem** for schematic capture, **gnetlist** for netlist extraction from the schematic, and **gspice** for simulation are primarily intended for electronic circuit design capture and simulation prior to construction of a printed circuit board. An extensive and extensible library of components is available covering a wide range of readily obtainable active and passive devices.


gEDA Project's Homepage - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.gpleda.org/index.html
Google
Most Visited Firefox Getting Started Latest Headlines
gEDA Project's Homepage
GPL
EDA
gEDA project
Screenshots · Documentation · Download · Search · Links
member projects
gschem
pcb
library
gerbv
icarus
gtkwave
http://geda.seul.org/wiki/geda:screenshots

**What is gEDA?**

The gEDA project has produced and continues working on a full GPL'd suite and toolkit of Electronic Design Automation tools. These tools are used for electrical circuit design, schematic capture, simulation, prototyping, and production. Currently, the gEDA project offers a mature suite of free software applications for electronics design, including schematic capture, attribute management, bill of materials (BOM) generation, netlisting into over 20 netlist formats, analog and digital simulation, and printed circuit board (PCB) layout.

The gEDA project was started because of the lack of free EDA tools for POSIX systems with the primary purpose of advancing the state of free hardware or open source hardware. The suite is mainly being developed on the GNU/Linux platform with some development effort going into making sure the tools run on other platforms as well.

For a complete list of freely available tools please be sure to visit Open Collector.

**New users**

Please read the FAQ and this tutorial. For even more information on installing and using gEDA, please look at the gEDA Wiki. Users looking for more symbols and footprints should go to the gedasymbols.org website.

**Wiki**

Please visit the gEDA wiki for more FAQs, the official documentation, mailing lists info, licensing details, developer documentation.

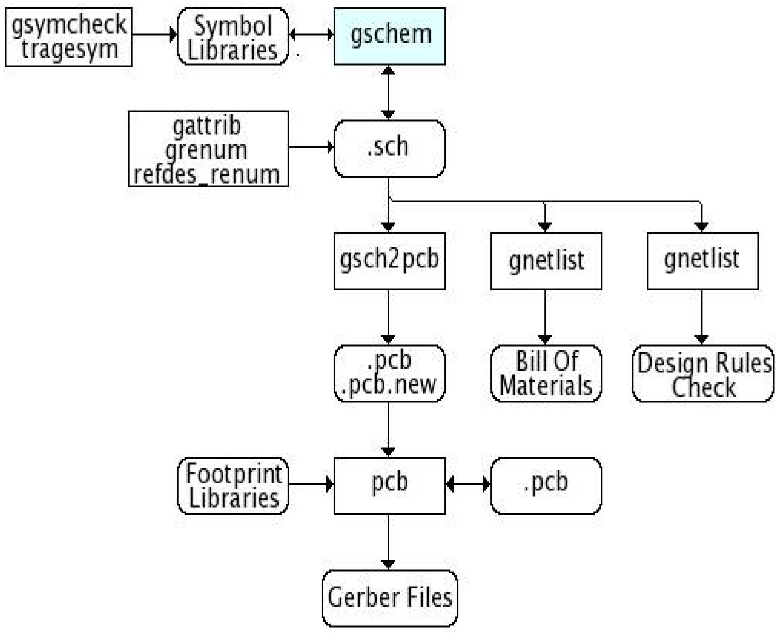
**Screenshot 13.0 – gEDA Project Web Page**

The illustration is taken from the gEDA Project web page at:
http://www.geda-project.org/

Once a design has been captured and simulated to the satisfaction of the circuit design engineer, the **pcb** tool can be used to design a printed circuit board from the mechanical data provided with the components used and the Bill of Material (BOM) generated to document the choices.

## 14.0 gEDA Project Electronic Design Printed Circuit Level Design

The design flow below illustrates the interactions of several of the tools in the process of designing a printed circuit board. The gEDA wiki stresses that essentially anyone interested in producing a printed circuit board can achieve good results with the tools.



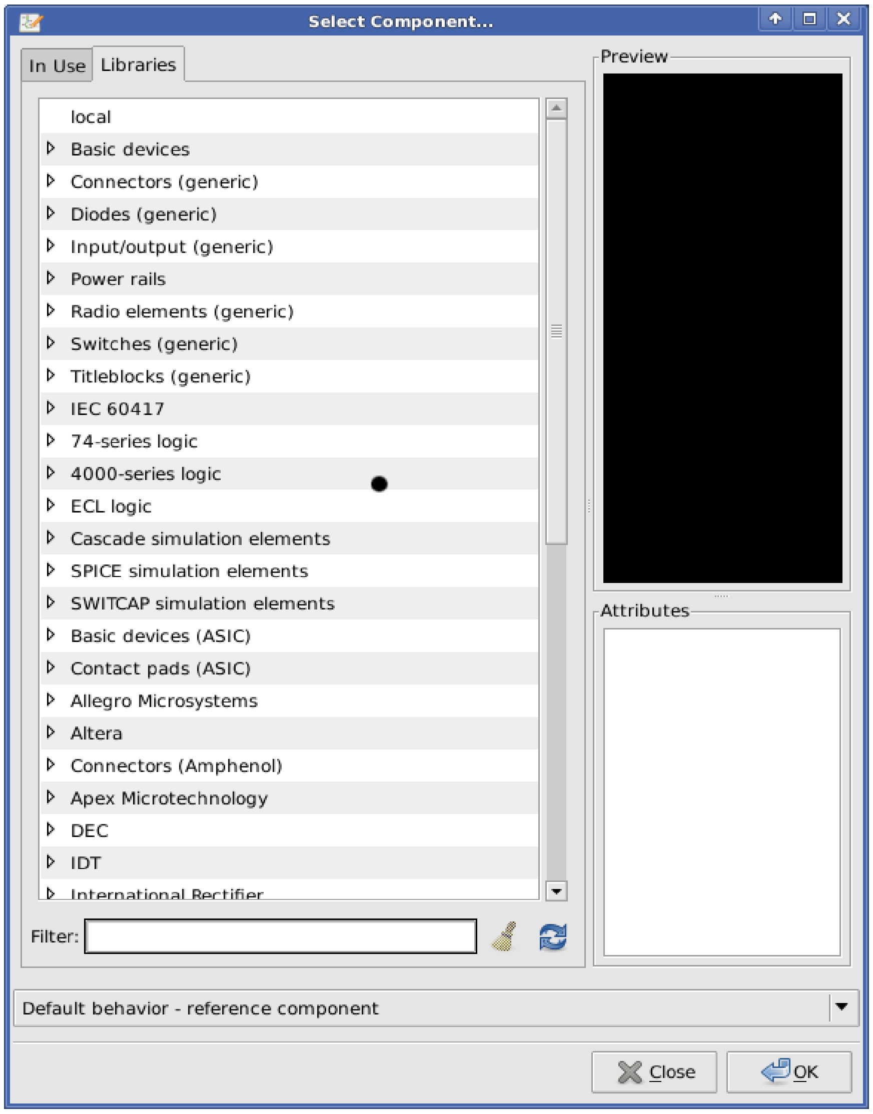
**Design Flow 14.0 – Printed Circuit Board Design Flow Using gEDA Tools**

A tutorial for the gEDA package tools for printed circuit design that dates back to 2005 is available on-line in PDF format from Cambridge University at the URL: http://www-mdp.eng.cam.ac.uk/web/CD/engapps/geda/starting_gEDA_long.pdf

It is written for the user that is already familiar with the concepts of electronic circuit design, but is complete enough to provide familiarity with the basic capabilities of the gEDA tool suite well beyond what is presented here. Each tool has tutorial material available to the user associated with the distribution of individual packages. Because of the nature of FOSS, though, the user should expect numerous updates and be prepared to accept that the tools are improved constantly.

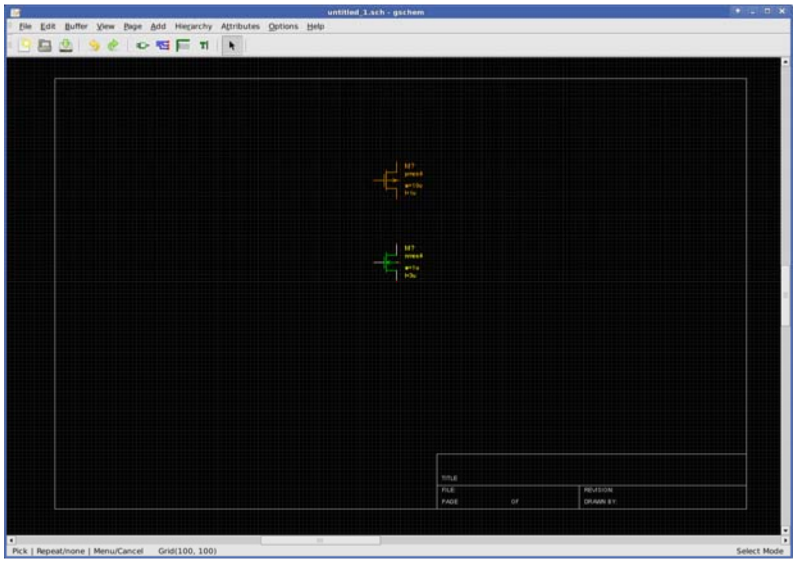
We present here a relatively short introduction to the tools to familiarize the reader with the design flow in an overview fashion.

14.1 Component Library Use to Construct a Schematic Diagram – To construct a schematic view of a circuit, the designer selects symbols from a library of components. An extensive library is provided with the tool suite and each component listed has an associated set of attributes that can be editied, copied to a new descriptor, or used as a template for extending the library.

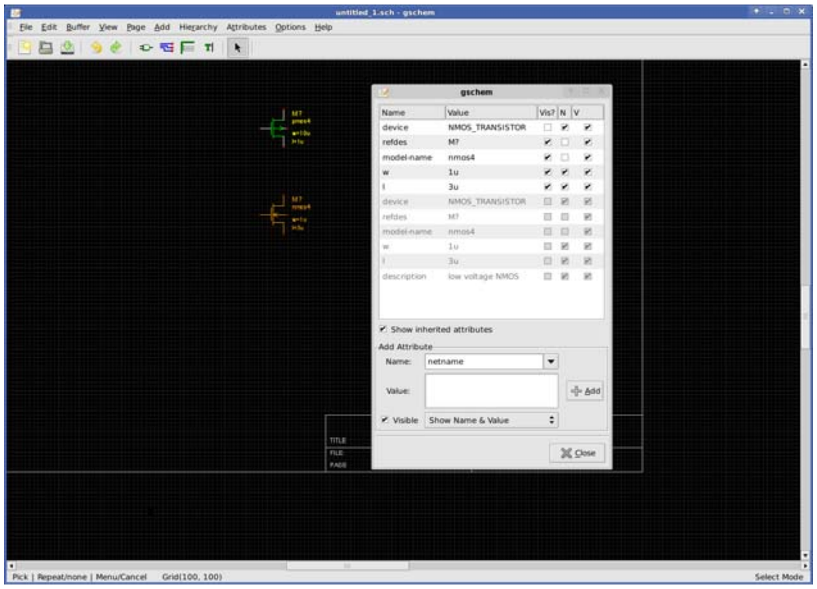


**Screen Shot 14.0 – gEDA Project Printed Circuit Board Component Library**

Components are selected from the library and placed in the gschem schematic area where each component may be selected for review of its detailed attributes as shown below:


untitled_1.sch - gschem
File Edit Buffer View Page Add Hierarchy Attributes Options Help
M?
pmos4
w=10u
l=1u
M?
nmos4
w=1u
l=3u
TITLE
FILE
PAGE
OF
REVISION
DRAWN BY
Pick | Repeat/none | Menu/Cancel Grid(100, 100)
Select Mode

**Screen Shot 14.1 – gschem with NMOS and PMOS Symbol Placement**


untitled_1.sch - gschem
File Edit Buffer View Page Add Hierarchy Attributes Options Help
gschem

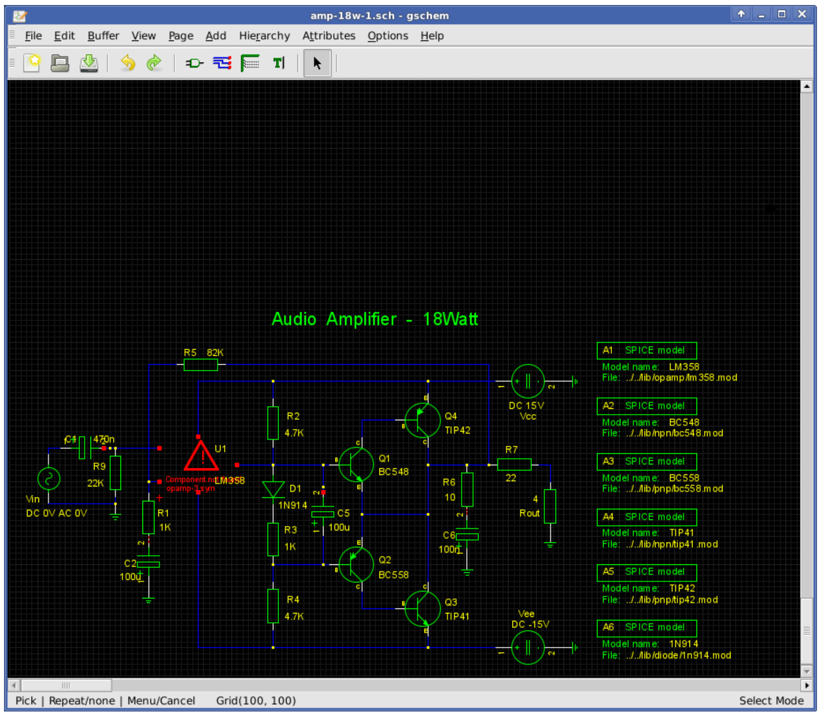
| Name | Value | Vis? | N | V |
|---|---|---|---|---|
| device | NMOS_TRANSISTOR | ☐ | ✓ | ✓ |
| refdes | M? | ✓ | ☐ | ✓ |
| model-name | nmos4 | ✓ | ☐ | ✓ |
| w | 1u | ✓ | ✓ | ✓ |
| l | 3u | ✓ | ✓ | ✓ |
| device | NMOS_TRANSISTOR | ☐ | ✓ | ✓ |
| refdes | M? | ☐ | ☐ | ✓ |
| model-name | nmos4 | ☐ | ☐ | ✓ |
| w | 1u | ☐ | ✓ | ✓ |
| l | 3u | ☐ | ✓ | ✓ |
| description | low voltage NMOS | ☐ | ✓ | ✓ |

✓ Show inherited attributes

Add Attribute
Name: netname
Value:
✓ Visible: Show Name & Value

Add
Close
Pick | Repeat/none | Menu/Cancel Grid(100, 100)
Select Mode

**Screen Shot 14.2 – gschem with NMOS Symbol Attributes Selected**



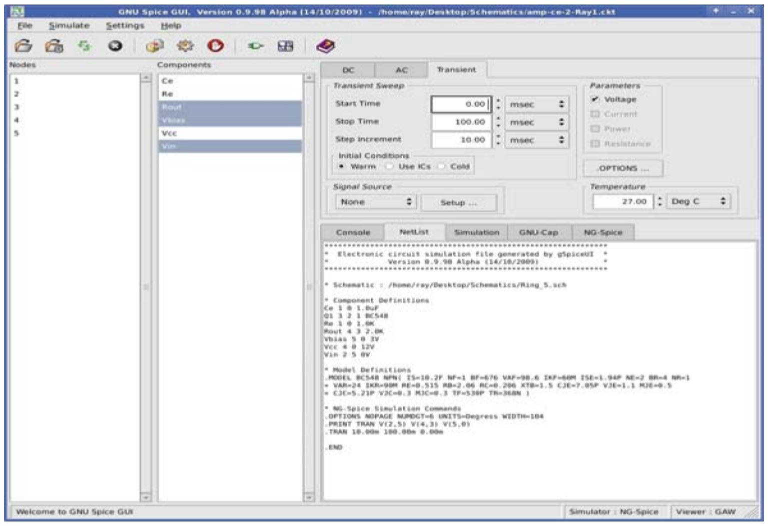
**Screen Shot 14.3 – gschem Tutorial Amplifier Example Schematic**

14.2 Component Library Use to Support Simulation – To support simulation, symbols may be added to the schematic directing the gspice simulator where to find simulation models for specific components. This arrangement allows the designer to have independent databases for symbols and for models.

The component models for printed circuit level design are generally provided by the component manufacturer and are known as “macro-models” because they provide behavioral features of the component without revealing details enough for a competitor to construct a copy of the component.

For Integrated Circuit (IC) design projects, the models are identified with the fabrication facility chosen for the project and are obtained from the fabrication organization directly, or from a multi-project wafer broker service like MOSIS or CMP. Process-specific models, however are of sufficient detail that they may reveal enough for a competitor to copy a design, or they may reveal much about the fabrication facilities processes and are usually held as confidential requiring special agreements prior to access.

14.3 **gspiceUI** Use to Support Simulation – To support simulation, gEDA provides the User Interface (**gspiceUI**) to combine access to multiple other tools and simplify simulations. The **gspiceUI** is directed to a specific **gschem** schematic through the “file” pull-down menu, automatically invokes the **gnetlist** tool to convert the interconnections on the schematic to a textual netlist required by **gspice**, finds the associated models from the model file directory, prepares, checks, and displays the netlist as shown below.



Nodes: 1 2 3 4 5
Components: Ce, Re, Rout, Vbias, Vcc, Vin

Transient Sweep
Start Time 0.00 msec
Stop Time 100.00 msec
Step Increment 10.00 msec
Initial Conditions: [✓ Warm] Use ICs Cold
Parameters: [✓ Voltage] Current Power Resistance
.OPTIONS ...
Signal Source: None  Setup ...
Temperature: 27.00 Deg C

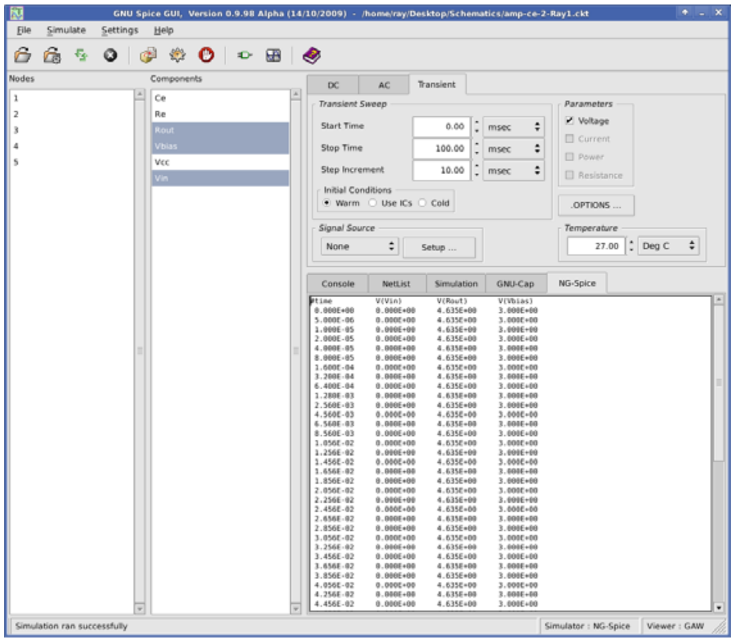
```
***************************************************************
*  Electronic circuit simulation file generated by gSpiceUI  *
*              Version 0.9.98 Alpha (14/10/2009)              *
***************************************************************

* Schematic : /home/ray/Desktop/Schematics/Ring_5.sch

* Component Definitions
Ce 1 0 1.0uF
Q1 3 2 1 BC548
Re 1 0 1.0K
Rout 4 3 2.0K
Vbias 5 0 3V
Vcc 4 0 12V
Vin 2 5 0V

* Model Definitions
.MODEL BC548 NPN( IS=10.2F NF=1 BF=676 VAF=98.6 IKF=60M ISE=1.94P NE=2 BR=4 NR=1
+ VAR=24 IKR=90M RE=0.515 RB=2.06 RC=0.206 XTB=1.5 CJE=7.05P VJE=1.1 MJE=0.5
+ CJC=5.21P VJC=0.3 MJC=0.3 TF=530P TR=368N )

* NG-Spice Simulation Commands
.OPTIONS NOPAGE NUMDGT=6 UNITS=Degress WIDTH=104
.PRINT TRAN V(2,5) V(4,3) V(5,0)
.TRAN 10.00m 100.00m 0.00m

.END
```

Simulator : NG-Spice    Viewer : GAW

**Screen Shot 14.3 – gspiceUI with Circuit Netlist Shown**

From the **gspiceUI**, simulation environmental parameters can be designated including temperature, supply voltage levels, and simulation types. DC analysis is supported to establish the bias levels of all components and allow the designer to consider if static design limits are within acceptable ranges. AC analysis is supported that permits checking of small-signal behaviors and stabilities. Transient analysis is supported to permit large-signal circuit performance and component stress levels.

We see in the illustration below, that for a transient analysis the **gspice** tool produces a file with time instances and the values of selected signals at each time step. The **gspiceUI** also allows the selection of a number of compatible display software tools to present the data in a graphical display format. Sown below are the textual form and the **gaw** tool graphics.

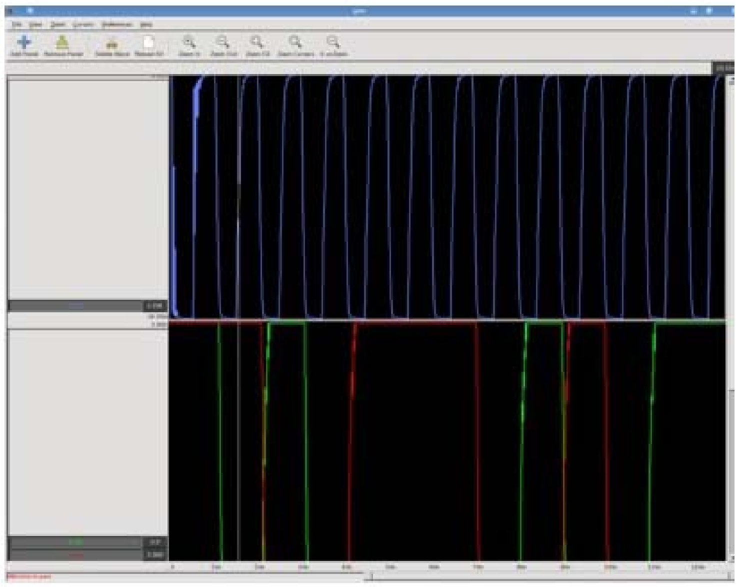


Transient Sweep
Start Time 0.00 msec
Stop Time 100.00 msec
Step Increment 10.00 msec
Initial Conditions: [✓ Warm] Use ICs Cold
Parameters: [✓ Voltage] Current Power Resistance
Signal Source: None
Temperature: 27.00 Deg C

```
#time          V(Vin)         V(Rout)        V(Vbias)
 0.000E+00     0.000E+00      4.635E+00      3.000E+00
 5.000E-06     0.000E+00      4.635E+00      3.000E+00
 1.000E-05     0.000E+00      4.635E+00      3.000E+00
 2.000E-05     0.000E+00      4.635E+00      3.000E+00
 4.000E-05     0.000E+00      4.635E+00      3.000E+00
 8.000E-05     0.000E+00      4.635E+00      3.000E+00
 1.600E-04     0.000E+00      4.635E+00      3.000E+00
 3.200E-04     0.000E+00      4.635E+00      3.000E+00
 6.400E-04     0.000E+00      4.635E+00      3.000E+00
 1.280E-03     0.000E+00      4.635E+00      3.000E+00
 2.560E-03     0.000E+00      4.635E+00      3.000E+00
 4.560E-03     0.000E+00      4.635E+00      3.000E+00
 6.560E-03     0.000E+00      4.635E+00      3.000E+00
 8.560E-03     0.000E+00      4.635E+00      3.000E+00
 1.056E-02     0.000E+00      4.635E+00      3.000E+00
 1.256E-02     0.000E+00      4.635E+00      3.000E+00
 1.456E-02     0.000E+00      4.635E+00      3.000E+00
 1.656E-02     0.000E+00      4.635E+00      3.000E+00
 1.856E-02     0.000E+00      4.635E+00      3.000E+00
 2.056E-02     0.000E+00      4.635E+00      3.000E+00
 2.256E-02     0.000E+00      4.635E+00      3.000E+00
 2.456E-02     0.000E+00      4.635E+00      3.000E+00
 2.656E-02     0.000E+00      4.635E+00      3.000E+00
 2.856E-02     0.000E+00      4.635E+00      3.000E+00
 3.056E-02     0.000E+00      4.635E+00      3.000E+00
 3.256E-02     0.000E+00      4.635E+00      3.000E+00
 3.456E-02     0.000E+00      4.635E+00      3.000E+00
 3.656E-02     0.000E+00      4.635E+00      3.000E+00
 3.856E-02     0.000E+00      4.635E+00      3.000E+00
 4.056E-02     0.000E+00      4.635E+00      3.000E+00
 4.256E-02     0.000E+00      4.635E+00      3.000E+00
 4.456E-02     0.000E+00      4.635E+00      3.000E+00
```

Simulation ran successfully | Simulator : NG-Spice | Viewer : GAW

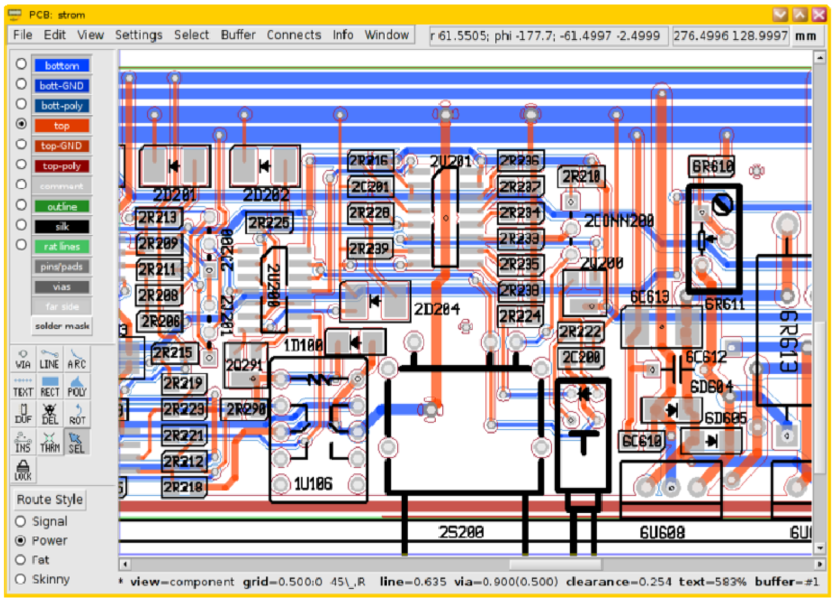
**Screen Shot 14.4 – gspiceUI with Transient Simulation Data List**



**Screen Shot 14.5 – gaw Tool with Transient Simulation Data Plot**

## 15.0 pcb Printed Circuit Design Artwork

The printed circuit board is presented in graphical layout form a shown below. It is used to present enumerated components, the wiring "traces" and other "printed" features of the board, the "silk-screen" artwork showing component enumerators and placement, and mechanical features such as component lead and mounting holes. Associated with the **pcb** artwork is a "Gerber File" used by the printed circuit manufacturer to fabricate the board.



r 61.5505; phi -177.7; -61.4997 -2.4999 276.4996 128.9997 mm

view=component grid=0.500:0 45_/R line=0.635 via=0.900(0.500) clearance=0.254 text=583% buffer=#1

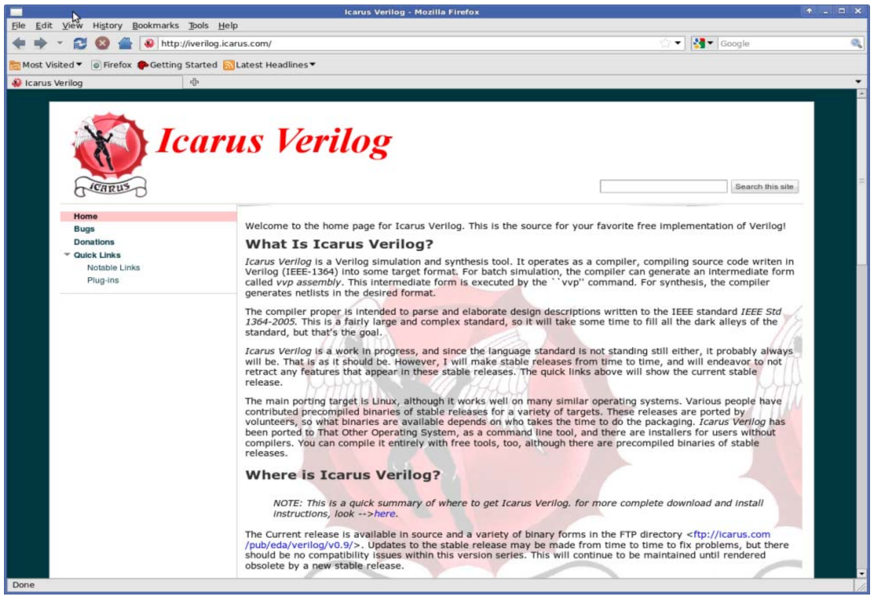
**Screen Shot 15.0 – pcb Graphic Printed Circuit Board Artwork**

## 16.0 gEDA Icarus and gtkwave Tools

Two other tools in the gEDA suite, **Icarus** and **gtkwave**, are targeted to the design and simulation of digital circuits and programmable digital components. For proprietary commercial off-the shelf (COTS) components, a proprietary additional tool specific to the component chosen is often require to complete the design but those tools are usually provided free of charge.

16.1 Icarus Verilog - Icarus Verilog supports the construction of a textual description of a digital circuit in the verilog hardware design language (HDL) that supports digital function design at several levels of abstraction. The levels of abstraction supported include transistor-level digital circuits using pmos and nmos primitives found in CMOS logic design and in IC designs. The Register Transfer Level (RTL) is also supported, as well as primitive Boolean logic functions including nand, nor, not (for inverters), exor, and many more.

The Icarus Verilog tool is a compiler for the verilog HDL and produces intermediate files that are useful for producing netlists of many logic families, programmable logic devices including Field Programmable Gate Array (FPGA) devices.


Icarus Verilog - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://iverilog.icarus.com/
Google
Most Visited Firefox Getting Started Latest Headlines
Icarus Verilog
Icarus Verilog
ICARUS
Search this site
Home
Bugs
Donations
Quick Links
Notable Links
Plug-ins
Done

Welcome to the home page for Icarus Verilog. This is the source for your favorite free implementation of Verilog!

**What Is Icarus Verilog?**

*Icarus Verilog* is a Verilog simulation and synthesis tool. It operates as a compiler, compiling source code writen in Verilog (IEEE-1364) into some target format. For batch simulation, the compiler can generate an intermediate form called *vvp assembly*. This intermediate form is executed by the ``vvp'' command. For synthesis, the compiler generates netlists in the desired format.

The compiler proper is intended to parse and elaborate design descriptions written to the IEEE standard *IEEE Std 1364-2005*. This is a fairly large and complex standard, so it will take some time to fill all the dark alleys of the standard, but that's the goal.

*Icarus Verilog* is a work in progress, and since the language standard is not standing still either, it probably always will be. That is as it should be. However, I will make stable releases from time to time, and will endeavor to not retract any features that appear in these stable releases. The quick links above will show the current stable release.

The main porting target is Linux, although it works well on many similar operating systems. Various people have contributed precompiled binaries of stable releases for a variety of targets. These releases are ported by volunteers, so what binaries are available depends on who takes the time to do the packaging. *Icarus Verilog* has been ported to That Other Operating System, as a command line tool, and there are installers for users without compilers. You can compile it entirely with free tools, too, although there are precompiled binaries of stable releases.

**Where is Icarus Verilog?**

*NOTE: This is a quick summary of where to get Icarus Verilog. for more complete download and install instructions, look -->here.*

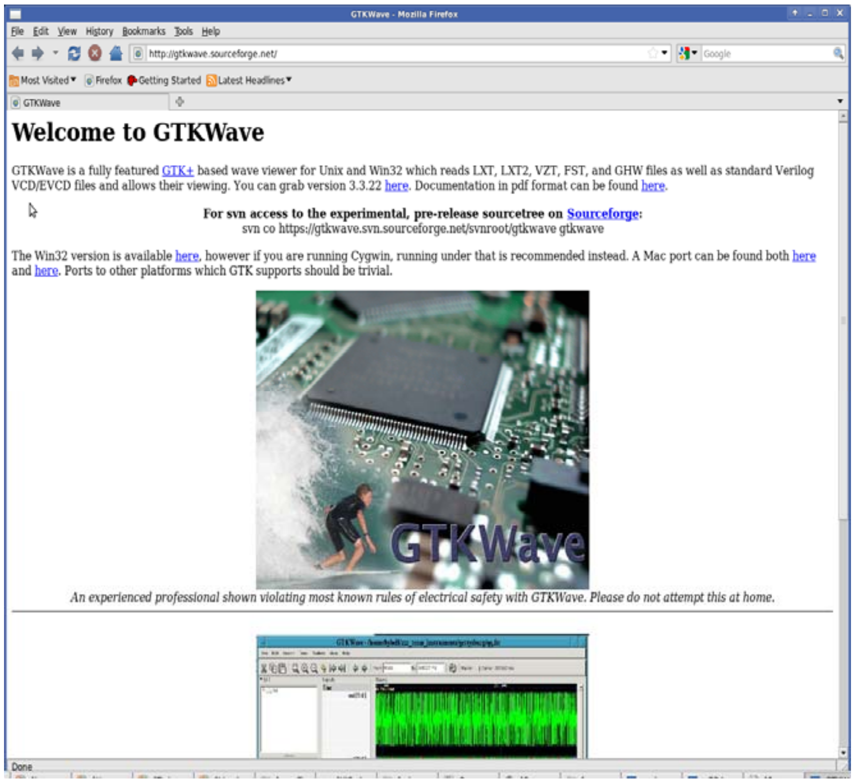
The Current release is available in source and a variety of binary forms in the FTP directory <ftp://icarus.com/pub/eda/verilog/v0.9/>. Updates to the stable release may be made from time to time to fix problems, but there should be no compatibility issues within this version series. This will continue to be maintained until rendered obsolete by a new stable release.

**Screenshot 16.0 – Icarus Verilog Web Page**

The illustration is taken from the Icarus Verilog web page at:
http://iverilog.icarus.com/

16.2 **gtkwave** Tool – **Icarus** Verilog produces a number of intermediate textual file outputs, some of which are used for the preparation of a netlist of components and another that supports the graphical display of transient simulation waveforms. The **gtkwave** tool displays the transient waveforms produced by the **Icarus** verilog simulations.

The transient response waveform is useful for deriving many timing issues but there does not seem to be a time-constraint design tool equivalent to “TimeMill,” or a power constraint design tool equivalent to “PowerMill” which were commercial tools. All timing and power issues are derived from manual designer interventions; tedious but effective.


GTKWave - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://gtkwave.sourceforge.net/
Google
Most Visited Firefox Getting Started Latest Headlines
GTKWave
GTKWave
Done

Welcome to GTKWave

GTKWave is a fully featured GTK+ based wave viewer for Unix and Win32 which reads LXT, LXT2, VZT, FST, and GHW files as well as standard Verilog VCD/EVCD files and allows their viewing. You can grab version 3.3.22 here. Documentation in pdf format can be found here.

**For svn access to the experimental, pre-release sourcetree on Sourceforge:**
svn co https://gtkwave.svn.sourceforge.net/svnroot/gtkwave gtkwave

The Win32 version is available here, however if you are running Cygwin, running under that is recommended instead. A Mac port can be found both here and here. Ports to other platforms which GTK supports should be trivial.

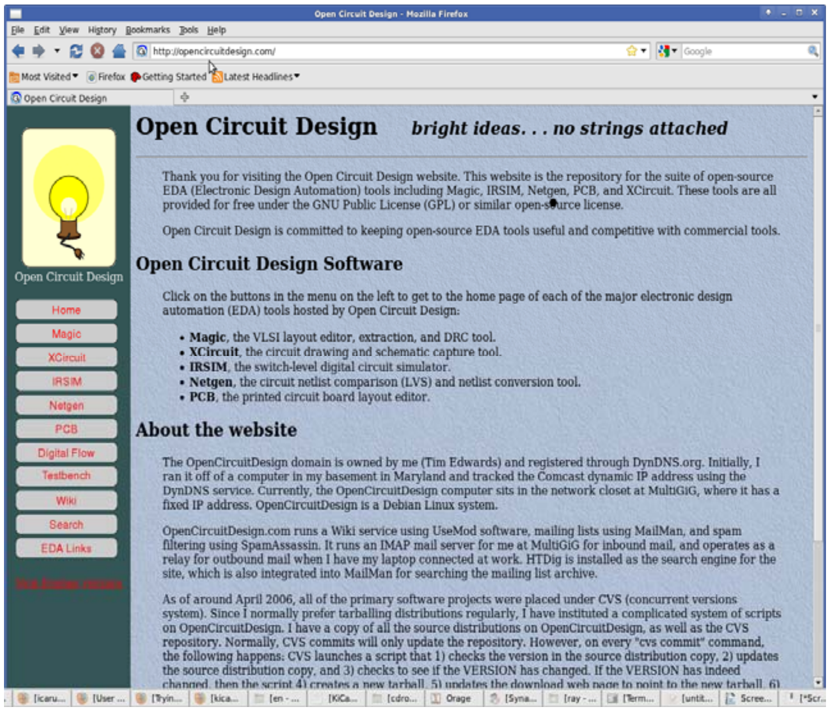
*An experienced professional shown violating most known rules of electrical safety with GTKWave. Please do not attempt this at home.*

**Screenshot 16.1 – Sourceforge gtkwave Web Page**

The illustration is taken from the sourceforge **gtkwave** web page at:
http://gtkwave.sourceforge.net/

## 17.0 Integrated Circuit (IC) Design-Specific Tools

The FOSS EDA tools that we have discussed thus far are quite effective and perform most of the tasks required to perform electronic circuit design at the Printed Circuit Board (PCB) level of construction. Mounted on the PCB, we most often find Integrated Circuit (IC) components that are modeled by and purchased from brand-name component suppliers. Occasionally, there are functions that cannot be obtained from commercially available off-the-shelf (COTS) suppliers. In those instances, a new IC design is undertaken if it is feasible from both a technical and economic viewpoint. Such special applications are most often associated with innovations in performance or economics and may be the target of product development by Small and Medium Enterprise (SME) design firms. As discussed in a prior section, the “Silicon Renaissance Initiative” was formed to recommend FOSS tools that are already available or need to be developed in order to re-invigorate the domestic IC design capabilities of the USA in SME and university settings. The “Silicon Renaissance Initiative” is not starting from a “table-rasa” or “clean-slate” environment, however, but has collected references to the existing FOSS IC EDA tools that are presented here. Most of the FOSS IC EDA tools are already collected and distributed freely by Dr. Tim Edwards at the opencircuitdesign website. There is some overlap of tools between the gEDA suite and the opencircuitdesign collection, but there is ongoing work to make the boundaries transparent between suites of tools.


Open Circuit Design - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://opencircuitdesign.com/
Google
Most Visited Firefox Getting Started Latest Headlines
Open Circuit Design
Open Circuit Design
Home
Magic
XCircuit
IRSIM
Netgen
PCB
Digital Flow
Testbench
Wiki
Search
EDA Links
[icaru... [User ... [Tryin... [kica... [en - ... [KiCa... [cdro... Orage [Syna... [ray - ... [Term... [untit... Scree... [*Scr...

**Open Circuit Design** ***bright ideas. . . no strings attached***

Thank you for visiting the Open Circuit Design website. This website is the repository for the suite of open-source EDA (Electronic Design Automation) tools including Magic, IRSIM, Netgen, PCB, and XCircuit. These tools are all provided for free under the GNU Public License (GPL) or similar open-source license.

Open Circuit Design is committed to keeping open-source EDA tools useful and competitive with commercial tools.

**Open Circuit Design Software**

Click on the buttons in the menu on the left to get to the home page of each of the major electronic design automation (EDA) tools hosted by Open Circuit Design:

- **Magic**, the VLSI layout editor, extraction, and DRC tool.
- **XCircuit**, the circuit drawing and schematic capture tool.
- **IRSIM**, the switch-level digital circuit simulator.
- **Netgen**, the circuit netlist comparison (LVS) and netlist conversion tool.
- **PCB**, the printed circuit board layout editor.

**About the website**

The OpenCircuitDesign domain is owned by me (Tim Edwards) and registered through DynDNS.org. Initially, I ran it off of a computer in my basement in Maryland and tracked the Comcast dynamic IP address using the DynDNS service. Currently, the OpenCircuitDesign computer sits in the network closet at MultiGiG, where it has a fixed IP address. OpenCircuitDesign is a Debian Linux system.

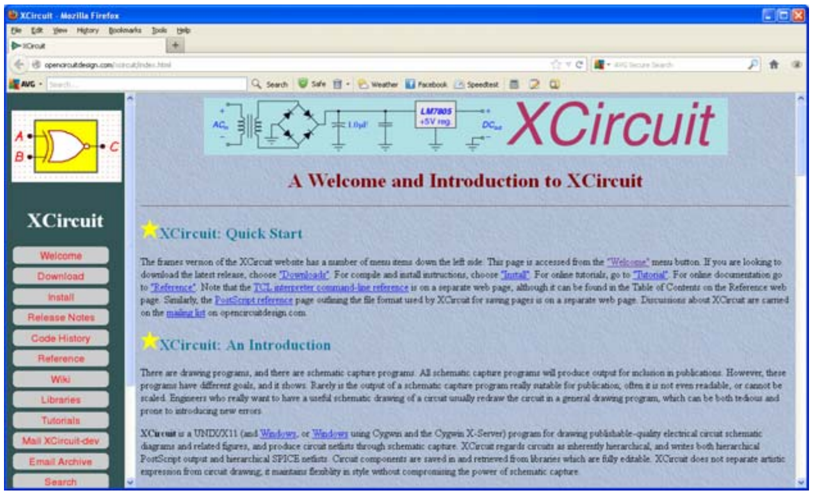
OpenCircuitDesign.com runs a Wiki service using UseMod software, mailing lists using MailMan, and spam filtering using SpamAssassin. It runs an IMAP mail server for me at MultiGiG for inbound mail, and operates as a relay for outbound mail when I have my laptop connected at work. HTDig is installed as the search engine for the site, which is also integrated into MailMan for searching the mailing list archive.

As of around April 2006, all of the primary software projects were placed under CVS (concurrent versions system). Since I normally prefer tarballing distributions regularly, I have instituted a complicated system of scripts on OpenCircuitDesign. I have a copy of all the source distributions on OpenCircuitDesign, as well as the CVS repository. Normally, CVS commits will only update the repository. However, on every "cvs commit" command, the following happens: CVS launches a script that 1) checks the version in the source distribution copy, 2) updates the source distribution copy, and 3) checks to see if the VERSION has changed. If the VERSION has indeed changed, then the script 4) creates a new tarball, 5) updates the download web page to point to the new tarball, 6)

**Screenshot 17.0 – Opencircuitdesign Web Page**

The illustration is taken from the opencircuitdesign web page at: http://opencircuitdesign.com/

17.1 **XCircuit** Schematic Capture Tool – The **XCircuit** schematic capture tool is effective but does not have the extensive library of component models that are available with the **gschem** tool. Because **gschem** supports simulation seamlessly between PCB and IC levels of abstraction, **gschem** is preferred in the Silicon Renaissance Initiative repertoire of tools. **XCircuit** is shown here for reference and may be used, perhaps with some difficulty over a mixed PCB and IC scope if desired.



A Welcome and Introduction to XCircuit

XCircuit: Quick Start

The frames version of the XCircuit website has a number of menu items down the left side. This page is accessed from the "Welcome" menu button. If you are looking to download the latest release, choose "Downloads". For compile and install instructions, choose "Install". For online tutorials, go to "Tutorial". For online documentation go to "Reference". Note that the TCL interpreter command-line reference is on a separate web page, although it can be found in the Table of Contents on the Reference web page. Similarly, the PostScript reference page outlining the file format used by XCircuit for saving pages is on a separate web page. Discussions about XCircuit are carried on the mailing list on opencircuitdesign.com.

XCircuit: An Introduction

There are drawing programs, and there are schematic capture programs. All schematic capture programs will produce output for inclusion in publications. However, these programs have different goals, and it shows. Rarely is the output of a schematic capture program really suitable for publication; often it is not even readable, or cannot be scaled. Engineers who really want to have a useful schematic drawing of a circuit usually redraw the circuit in a general drawing program, which can be both tedious and prone to introducing new errors.

**XCircuit** is a UNIX/X11 (and Windows, or Windows using Cygwin and the Cygwin X-Server) program for drawing publishable-quality electrical circuit schematic diagrams and related figures, and produce circuit netlists through schematic capture. XCircuit regards circuits as inherently hierarchical, and writes both hierarchical PostScript output and hierarchical SPICE netlists. Circuit components are saved in and retrieved from libraries which are fully editable. XCircuit does not separate artistic expression from circuit drawing; it maintains flexibility in style without compromising the power of schematic capture.

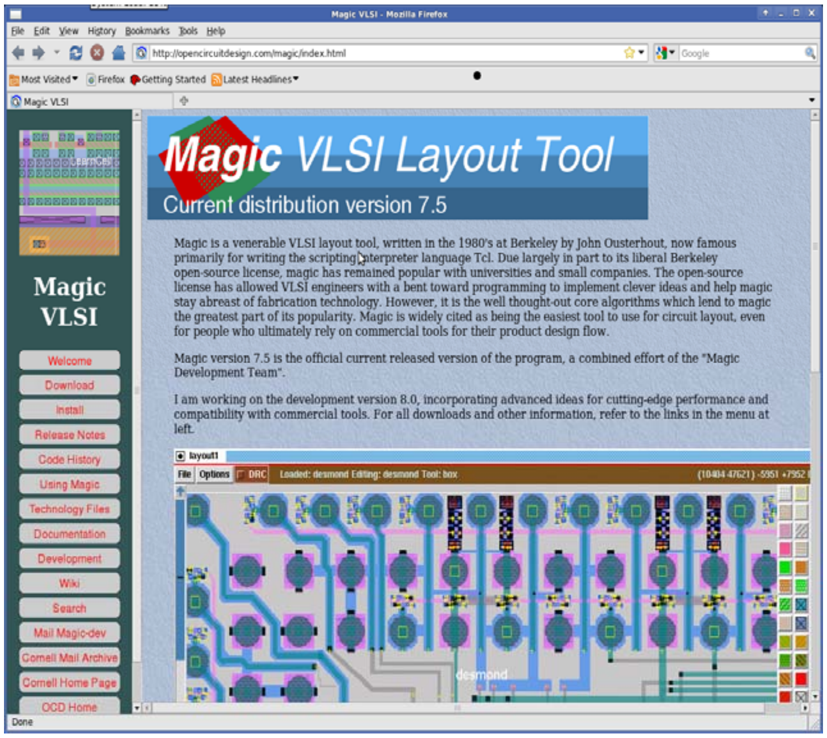
**Screenshot 17.1 – XCircuit Web Page**

The illustration is taken from the opencircuitdesign web page at: http://opencircuitdesign.com/xcircuit/index.html

17.2 **Magic** IC Layout Tool – The FOSS Magic tool has been used in university and SME environments since the mid-1980s when it was developed under a grant to support a National Science Foundation (NSF) initiative to bring the early Carver-Mead approach to IC design into widespread university and SME use.

Magic enjoyed notable success in the university environment but was nearly displaced by deep-discount EDA tool offerings from commercial EDA tool vendors and their university programs. Dr. Tim Edwards was an early adopter of the Magic tool suite, maintaining, rewriting, and extending the tool suite as FOSS through today.

The flagship Magic tool uses a dimensioning system in "lambda" units that are indirectly defined in terms of the physical dimensions used in other tools. In this regard, there is an issue in simulation that requires a script to translate dimensions from lambda units into those compatible with spice model data structures.  Dimensions are "extracted" from the layout and converted to a "**sim**" data structure that is used by the IRSIM simulator tool directly for digital simulations using the "**ext2sim**" tool supplied. The "**sim**" data structure is converted to a "**spice**" netlist using the "**sim2spice**" tool that is also supplied and analog, digital, and mixed-signal circuits can be simulated using the **gspice** tool, once the model files are converted from lambda units to the metric dimensions required by the **gspice** tool.


Magic VLSI - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://opencircuitdesign.com/magic/index.html
Most Visited Firefox Getting Started Latest Headlines
Magic VLSI
Magic VLSI
Welcome
Download
Install
Release Notes
Code History
Using Magic
Technology Files
Documentation
Development
Wiki
Search
Mail Magic-dev
Cornell Mail Archive
Cornell Home Page
OCD Home
Magic VLSI Layout Tool
Current distribution version 7.5
layout1
File Options DRC Loaded: desmond Editing: desmond Tool: box
(10404 47621) -5951 +7952
desmond
Done

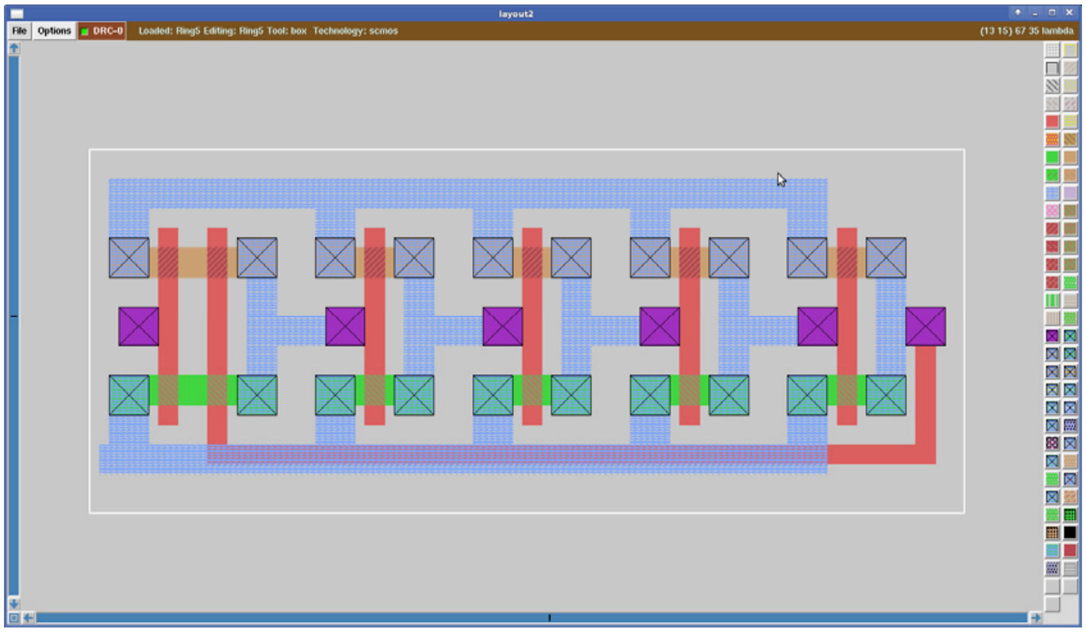
Magic is a venerable VLSI layout tool, written in the 1980's at Berkeley by John Ousterhout, now famous primarily for writing the scripting interpreter language Tcl. Due largely in part to its liberal Berkeley open-source license, magic has remained popular with universities and small companies. The open-source license has allowed VLSI engineers with a bent toward programming to implement clever ideas and help magic stay abreast of fabrication technology. However, it is the well thought-out core algorithms which lend to magic the greatest part of its popularity. Magic is widely cited as being the easiest tool to use for circuit layout, even for people who ultimately rely on commercial tools for their product design flow.

Magic version 7.5 is the official current released version of the program, a combined effort of the "Magic Development Team".

I am working on the development version 8.0, incorporating advanced ideas for cutting-edge performance and compatibility with commercial tools. For all downloads and other information, refer to the links in the menu at left.

**Screenshot 17.1 – Magic Web Page**

The illustration is taken from the opencircuitdesign web page at:
http://opencircuitdesign.com/magic/



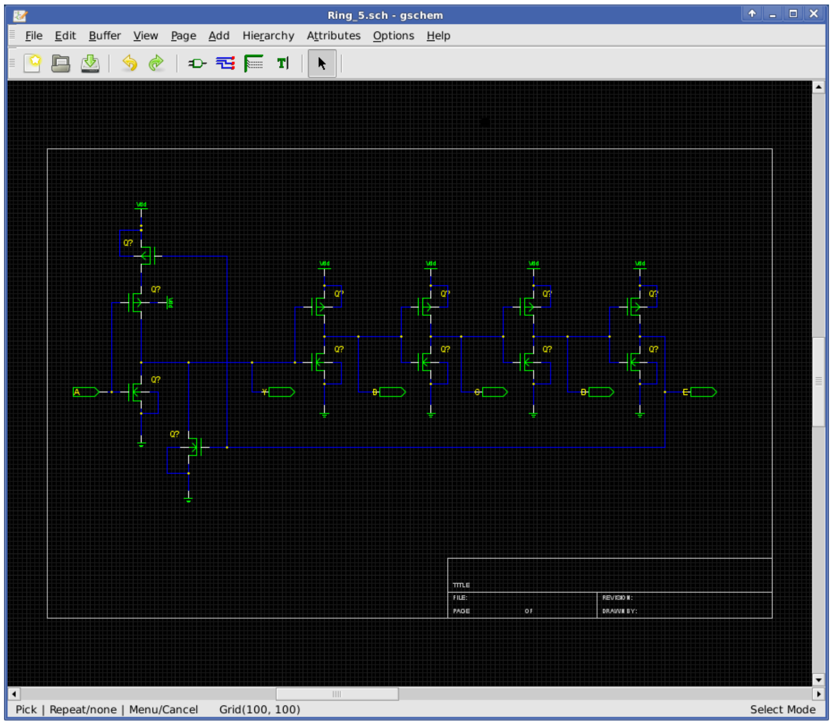
**Screenshot 17.2 – Magic Layout of 5-Stage Ring Oscillator**

17.3 **Magic** IC Layout of 5-Stage Ring Oscillator – The example above is that for 4 inverters and a 2-input NOR gate (with a connection/construction) error to show the need for the **netgen** tool discussed later. The "brown" structures represent diffusion patterns that make PMOS transistors, the "green" structures represent diffusion patterns that make NMOS transistors, the "red" structures represent deposited polysilicon patterns that make gate structures, and the "blue" structures represent deposited metallic layer patterns to connect the circuit. The remaining square patterns represent connections between layers.

Integrated circuit design makes no fundamental distinction between analog, digital, and mixed-signal circuits: they are all fundamentally somewhat analog in nature and simulated in a spice simulator much like purely analog designs.

Unfortunately, the Magic tool suite does not automatically generate schematics and that requires manual effort by the designer, In addition, the designer must reconcile the dimensions and models somewhat also. The Magic tool suite, though, does permit a path to automated generation of a netlist from a design such as that one shown above. The designer must then generate the intended schematic (if one is required) for documentation, simulate and verify that the schematic behavior and layout behaviors match.

We show below a **gschem** schematic intended to represent the **Magic** IC Layout of the 5-Stage Ring Oscillator.

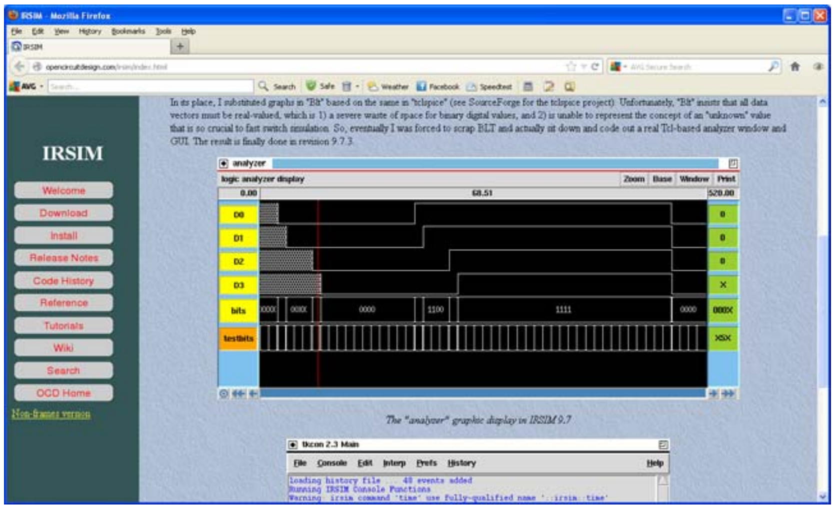

Ring_5.sch - gschem
File Edit Buffer View Page Add Hierarchy Attributes Options Help
Vdd
Q?
A
B
C
TITLE
FILE:
PAGE
REVISION:
DRAWN BY:
Pick | Repeat/none | Menu/Cancel Grid(100, 100)
Select Mode

**Screenshot 17.3 – gschem Schematic of 5-Stage Ring Oscillator**

17.4 **IRSIM** Simulation of the **Magic** 5-Stage Ring Oscillator – The Magic tool suite contains a simulator that is much simpler than a full **gspice** simulator. It does, however, extract from the design parameters that describe parasitic wiring effects, and capacitances so that it can make a straight-line approximation to the rise and fall times associated with digital signals. In this regard, and for digital circuits, it is often good practice to simulate the circuit extracted from the Magic layout using the **IRSIM** tool.

We show below the web page associated with the **IRSIM** simulator and provide a link to its location so that it can be reviewed in greater detail if desired. For the 5-Stage Ring Oscillator, **IRSIM** would provide a behavior tht does not match the expected behavior and point to an error in the layout, whereas the **gspice** simulation of the **gschem** schematic would provide more nearly correct behavior.

Once the behaviors are close to a match, we will use the **netgen** tool to compare netlists extracted from the Magic layout with the netlist derived from the **gschem** schematic.



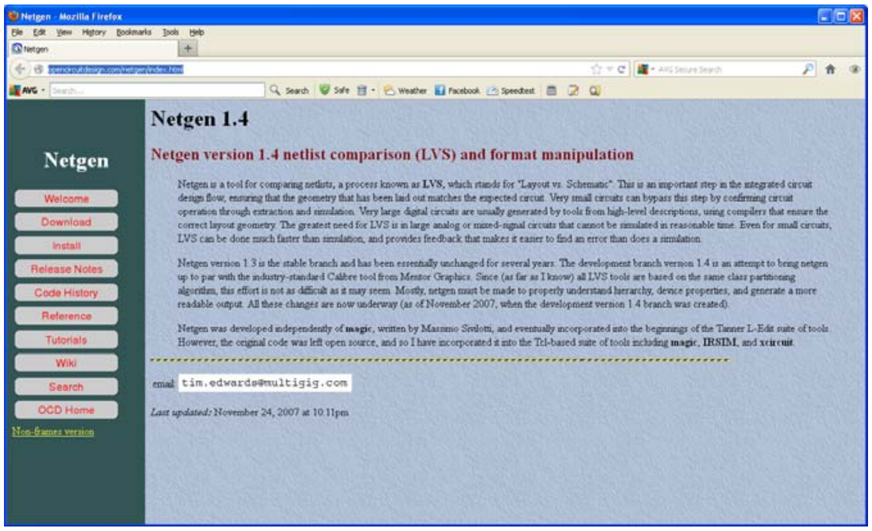
In its place, I substituted graphs in "Blt" based on the same in "tclspice" (see SourceForge for the tclspice project). Unfortunately, "Blt" insists that all data vectors must be real-valued, which is 1) a severe waste of space for binary digital values, and 2) is unable to represent the concept of an "unknown" value that is so crucial to fast switch simulation. So, eventually I was forced to scrap BLT and actually sit down and code out a real Tcl-based analyzer window and GUI. The result is finally done in revision 9.7.3.

*The "analyzer" graphic display in IRSIM 9.7*

```
loading history file ... 48 events added
Running IRSIM Console Functions
Warning: irsim command 'time' use fully-qualified name '::irsim::time'
```

**Screenshot 17.4 – IRSIM simulation of a Ring Oscillator**

The illustration is taken from the opencircuitdesign web page at:
http://opencircuitdesign.com/irsim/index.html



Netgen 1.4

Netgen version 1.4 netlist comparison (LVS) and format manipulation

Netgen is a tool for comparing netlists, a process known as **LVS**, which stands for "Layout vs. Schematic". This is an important step in the integrated circuit design flow, ensuring that the geometry that has been laid out matches the expected circuit. Very small circuits can bypass this step by confirming circuit operation through extraction and simulation. Very large digital circuits are usually generated by tools from high-level descriptions, using compilers that ensure the correct layout geometry. The greatest need for LVS is in large analog or mixed-signal circuits that cannot be simulated in reasonable time. Even for small circuits, LVS can be done much faster than simulation, and provides feedback that makes it easier to find an error than does a simulation.

Netgen version 1.3 is the stable branch and has been essentially unchanged for several years. The development branch version 1.4 is an attempt to bring netgen up to par with the industry-standard Calibre tool from Mentor Graphics. Since (as far as I know) all LVS tools are based on the same class partitioning algorithm, this effort is not as difficult as it may seem. Mostly, netgen must be made to properly understand hierarchy, device properties, and generate a more readable output. All these changes are now underway (as of November 2007, when the development version 1.4 branch was created).

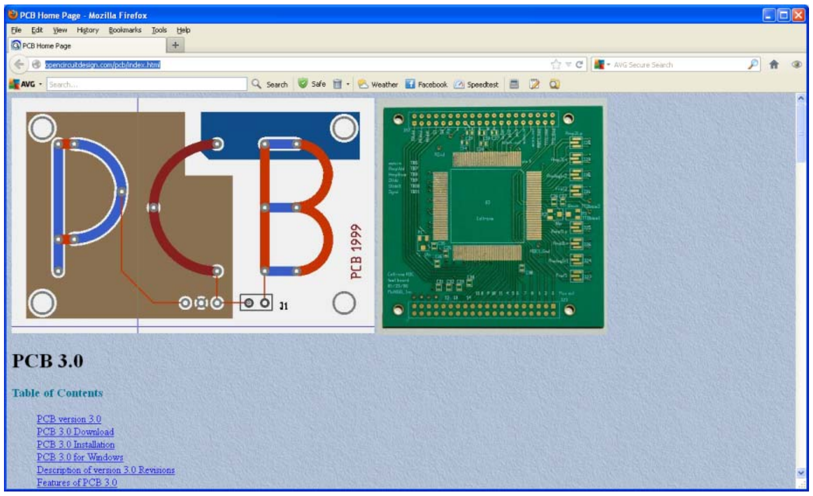
Netgen was developed independently of **magic**, written by Massimo Sivilotti, and eventually incorporated into the beginnings of the Tanner L-Edit suite of tools. However, the original code was left open source, and so I have incorporated it into the Tcl-based suite of tools including **magic**, **IRSIM**, and **xcircuit**.

email tim.edwards@multigig.com

*Last updated:* November 24, 2007 at 10:11pm

**Screenshot 17.5 – Netgen Netlist Comparison Tool**

The **netgen** illustration is taken from the opencircuitdesign web page at:
http://opencircuitdesign.com/netgen/index.html


PCB Home Page - Mozilla Firefox
File Edit View History Bookmarks Tools Help
PCB Home Page
opencircuitdesign.com/pcb/index.html
Search Safe Weather Facebook Speedtest
PCB 1999
J1

PCB 3.0

Table of Contents

PCB version 3.0
PCB 3.0 Download
PCB 3.0 Installation
PCB 3.0 for Windows
Description of version 3.0 Revisions
Features of PCB 3.0

**Screenshot 17.6 – Magic Version of PCB Tool**

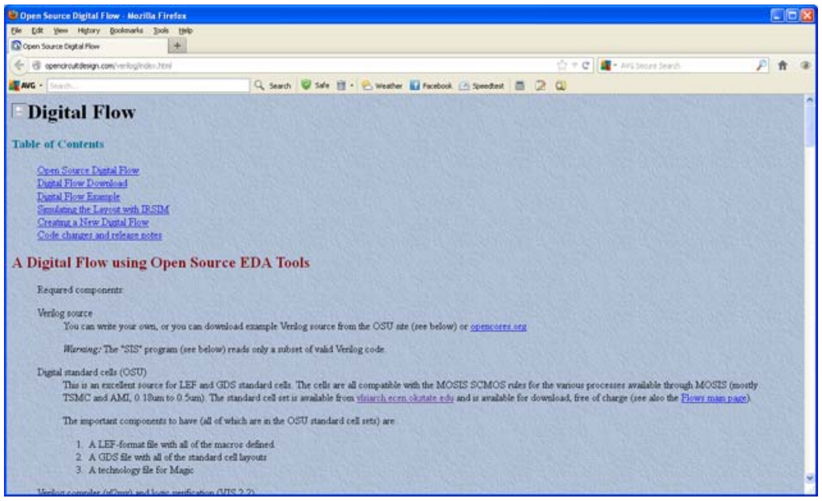
The **PCB** illustration is taken from the opencircuitdesign web page at:
http://opencircuitdesign.com/pcb/index.html

The Magic Tool suite also contains a version of the **PCB** layout tool shown above, but the coupling of the tool with a schematic is not as closely linked as in the case with the gEDA version.

## 18.0 New Magic Tools for Digital Design

The Magic Tool suite, much as is the case with the gEDA tool suite is under constant improvement and enhancement. Fundamentally, though it is driven by the layout or physical design project inputs. In gEDA, the digital design is driven by the schematic and a loosely coupled dependence on verilog through the Icarus tool.

There is some attempt to support verilog from Magic, particularly using a characterized cell library available in the university community that is suitable for automated synthesis to layout. Magic was used in the development of the cells and IRSIM in the characterization.


Open Source Digital Flow - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Open Source Digital Flow
opencircuitdesign.com/verilog/index.html
AVG Search Safe Weather Facebook Speedtest

Digital Flow

Table of Contents

Open Source Digital Flow
Digital Flow Download
Digital Flow Example
Simulating the Layout with IRSIM
Creating a New Digital Flow
Code changes and release notes

A Digital Flow using Open Source EDA Tools

Required components:

Verilog source
You can write your own, or you can download example Verilog source from the OSU site (see below) or opencores.org

*Warning:* The "SIS" program (see below) reads only a subset of valid Verilog code.

Digital standard cells (OSU)
This is an excellent source for LEF and GDS standard cells. The cells are all compatible with the MOSIS SCMOS rules for the various processes available through MOSIS (mostly TSMC and AMI, 0.18um to 0.5um). The standard cell set is available from vlsiarch.ecen.okstate.edu and is available for download, free of charge (see also the Flows main page).

The important components to have (all of which are in the OSU standard cell sets) are:

1. A LEF-format file with all of the macros defined
2. A GDS file with all of the standard cell layouts
3. A technology file for Magic

Verilog compiler (vl2mv) and logic verification (VIS 2.2)

**Screenshot 18.0 – Magic Digital Design Flow**

The Digital Design Flow illustration is taken from the opencircuitdesign web page at: http://opencircuitdesign.com/verilog/index.html

## 19.0 Open Source Verilog Designs at OpenCores

OpenCores is an organization that provides a repository of existing designs for a wide variety of project components at several levels of functionality. There are restrictions of the use of some of the files for commercial purposes, bu a lot can be learned from examples given. There are instances in the press of commercial entities that have used OpenCores designs in their products and were restrained from selling products without providing copies of all information on an open-source basis. In this case the FOSS of the work product is more of an Intellectual Property (IP) issue that the FOSS nature of the tools and the entirely original designs produced using those tools. The user is cautioned to read the licensing agreements carefully and determine if any of the FOSS elements are passed into a design or if the design work-product is totally independent of the tools used.

The following quote from OpenCores describes their mission:

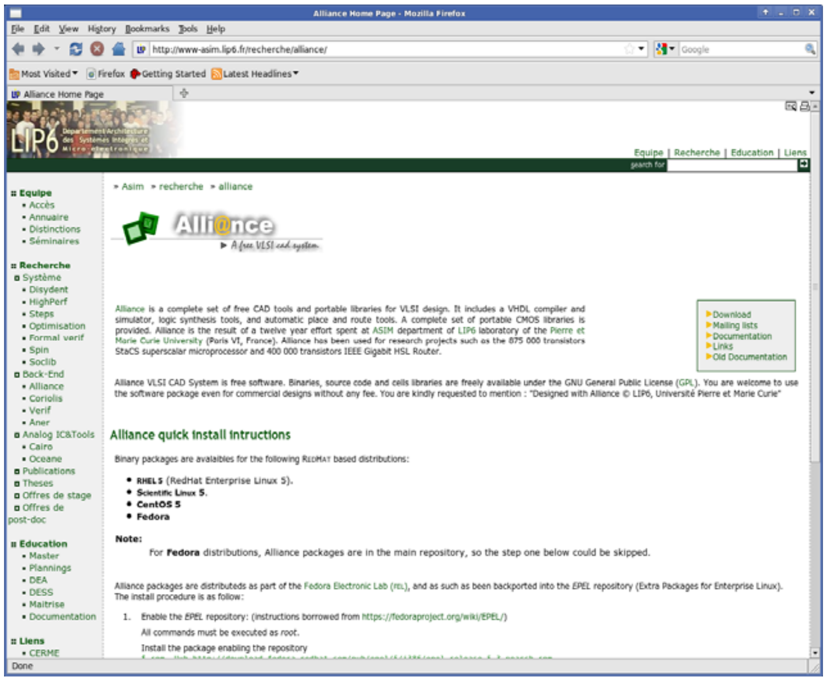
*"OpenCores is the world's largest site/community for development of hardware IP cores as open source.*

*OpenCores.org host the source code for different digital HW projects (IP-cores, SoC, boards, etc) and support the users with different tools, platforms, forums and other useful information. Please join us!"*

The quote above is taken from the OpenCores web page at:
http://opencores.org/

## 20.0 Open Source VHDL Designs

Another FOSS design flow is supported for VHDL; another older version of a hardware design language used mostly in universities and sometimes specified for use in military contracts. The Alliance VHDL tools are supported by a French university



Alliance is a complete set of free CAD tools and portable libraries for VLSI design. It includes a VHDL compiler and simulator, logic synthesis tools, and automatic place and route tools. A complete set of portable CMOS libraries is provided. Alliance is the result of a twelve year effort spent at ASIM department of LIP6 laboratory of the Pierre et Marie Curie University (Paris VI, France). Alliance has been used for research projects such as the 875 000 transistors StaCS superscalar microprocessor and 400 000 transistors IEEE Gigabit HSL Router.

Alliance VLSI CAD System is free software. Binaries, source code and cells libraries are freely available under the GNU General Public License (GPL). You are welcome to use the software package even for commercial designs without any fee. You are kindly requested to mention : "Designed with Alliance © LIP6, Université Pierre et Marie Curie"

Alliance quick install intructions

Binary packages are avalaibles for the following RedHat based distributions:

- **RHEL 5** (RedHat Enterprise Linux 5).
- **Scientific Linux 5**.
- **CentOS 5**
- **Fedora**

**Note:**
For **Fedora** distributions, Alliance packages are in the main repository, so the step one below could be skipped.

Alliance packages are distributeds as part of the Fedora Electronic Lab (FEL), and as such as been backported into the *EPEL* repository (Extra Packages for Enterprise Linux).
The install procedure is as follow:

1. Enable the *EPEL* repository: (instructions borrowed from https://fedoraproject.org/wiki/EPEL/)

   All commands must be executed as *root*.

   Install the package enabling the repository

**Screenshot 19.0 – Alliance Digital Design Flow**

The illustration above is taken from the Alliance web page at:
http://www-soc.lip6.fr/recherche/cian/alliance/

## 21.0 FOSS in Engineering Summary and Conclusions

This course introduced the engineer to the subject of “Free Open-Source Software” (FOSS). Quotes from various sources are included widely and attributed to their source throughout the course. The price of commercial software is often a barrier-to-entry for emerging small and medium enterprise (SME) and for the case of circuit designs in electronic engineering to both the printed circuit design and Integrated Circuit design levels, a design flow using only FOSS was demonstrated to exist. As the particular case exemplified in this course, the “Silicon Renaissance Initiative” tool recommendations were shown to illustrate Engineering Design Automation (EDA) software tools and the design flow for that use. As an alternative, the commercial tools often require hundreds of thousands of dollars investment to support the design effort. In the USA, with its copyright enforcement, the SME faces competition from foreign entities with “illegal” copies of commercial software and thus cannot compete. This course illustrates both the general tools available and useful for any computer user and professional engineer with any practice specialty, as well as the professional engineer involved in designing electronic circuits.