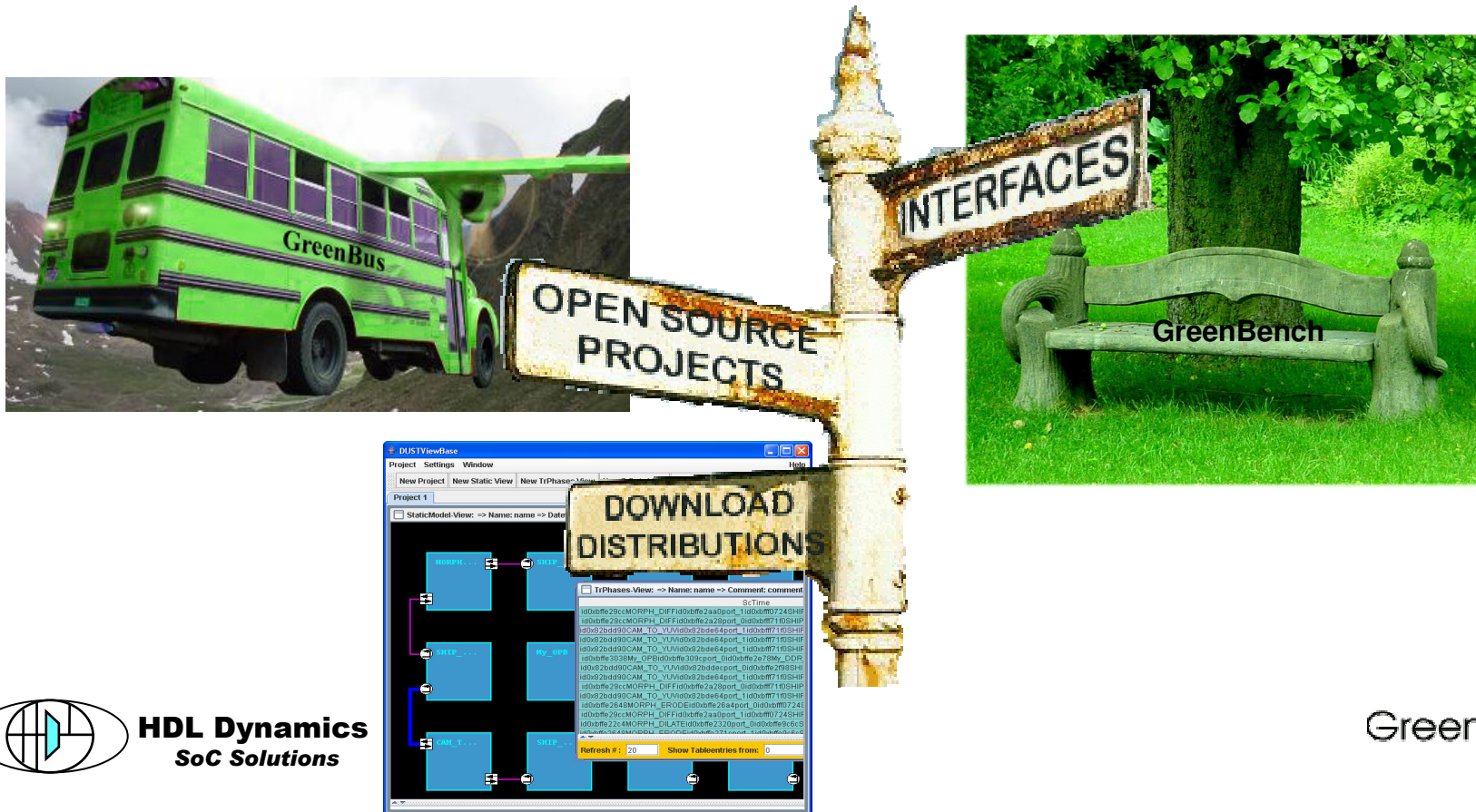


GreenSocs Overview

www.greensocs.com

ESL = Models + GreenSocs + Tools



Who is GreenSocs

GreenSocs wears 2 hats

- **Developer, supplier, contractor of SystemC infrastructure solutions**
 - partner for value added SystemC development
 - Infrastructure/tools/IP solution development
 - Customized solutions to ESL problems
 - Manage, facilitate, and productize ESL research
- **Maintains Open Source based SystemC infrastructure project**
 - Distribution of tool packages to SystemC community
 - Maintained and funded by subscriptions and contracts with GreenSocs
 - Open to all to contribute/join/use



HDL Dynamics
SoC Solutions



GreenSocs Business Model

- **SystemC environment differs from other EDA tools**
 - Based on standard C++ environment
 - Tool infrastructure based on Open Source Development
 - Proprietary info is typically company specific models
- **However Open Source does not mean simple or integrated**
 - Still many missing pieces, interfaces, tools, documentation
 - Like Linux, there is need for supported package solutions
- **GreenSoCs develops and productizes SystemC infrastructure**
 - Ensures quality tools and IP releases to reduce in-house support burden
 - Peer reviewed for quality, supported Wiki, SourceForge repository
 - Customization and enhancements
 - Subscriber (tool, IP, silicon distributor) fee or end user license
 - Ongoing support for developers and users.



HDL Dynamics
SoC Solutions



GreenSocs Business Model

- **Key Differentiator for tools development**
 - Getting customized rather than off the shelf solutions
 - Customers get access to customized source code
 - Raw Packages may be released back to Open Source community
 - **GreenSoCs maintains and augments customized solutions**
- **Why ? open source = flexibility in evolving solutions**
 - Simpler to update and customize tools for future needs
 - Allows choices - what do you support internally vs. what is maintained through GreenSoCs
 - Facilitates research efforts for improved SystemC tool infrastructure
- **GreenSoCs is involved and advocates users to standards groups**
 - **OSCI, OCP-IP, ECLIPSE, SPIRIT,...**

So why not just use the Open Source Code

You can, but consider the alternatives and tradeoffs

- **In House development and integration**
 - SystemC tools are complex Open Source Solutions
 - Support is needed, key pieces is missing
- **Ongoing in-house costs for specific tools and infrastructure?**
 - Learning curve in using/supporting raw 3rd party code?
 - Costs/risks associated with in-house development
 - Diverted resources better used for company IP value added models
- **ROI in GreenSoCs contracted turnkey SystemC development**
 - Debugged, documented “Working out of the box” working solutions
 - Developer value – Access to best in class SystemC developers
 - Value to Users - packaged and supported user friendly solutions



HDL Dynamics
SoC Solutions



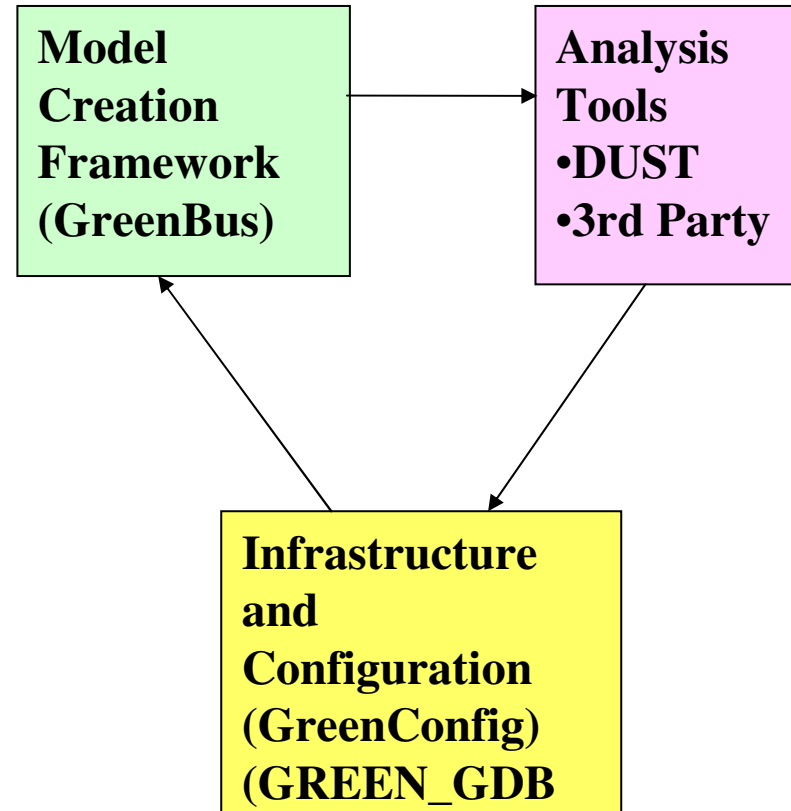
Available today from the GreenSocs repository

- **Supported Architectures:**
i386, x86_64, Cygwin, Mac OS X (ppc and i386)
- **Packages:**
 - apt_0.5.28.6-gs3_i386.deb
 - archc2.0+beta3-gcc4.0.3.deb
 - cdt_3.1.0-gs3_i386.deb
 - dpkg_1.10.28-gs1_i386.deb
 - dust1.0-gcc4.0.3-systemc2.2.deb
 - eclipse_3.2-gs2_i386.deb
 - gcc4.0.3_4.0.3_i386.deb
 - jre_1.5.0-gs2_i386.deb
 - libboost-dev_1.33.1-4_i386.deb
 - modules_3.2.3-gs2_i386.deb
 - pdesigner_1.0.2-gs2_i386.deb
 - pdlibrary_1.0.1-gs1_i386.deb
 - scv1.0p2-gcc3.3.1-systemc2.1-gs2_i386.deb
 - scv1.0p2-gcc3.3.1-systemc2.2-gs2_i386.deb
 - scv1.0p2-gcc4.0.3-systemc2.1-gs2_i386.deb
 - scv1.0p2-gcc4.0.3-systemc2.2-gs2_i386.deb
 - systemc2.1-gcc3.3.1-gs2_i386.deb
 - systemc2.1-gcc4.0.3-gs2_i386.deb
 - systemc2.2-gcc3.3.1-gs2_i386.deb
 - systemc2.2-gcc4.0.3-gs2_i386.deb

GreenSocs Tools

- **GreenBus**
- **DUST analysis framework**
- **Greenbench**
- **GreenConfig**
- **GGreenControl**
- **... What else?**
 - Subscriber defined
 - Open Source contributed
 - GreenSocs Developed and Supported

Capture tools – Eclipse GUI

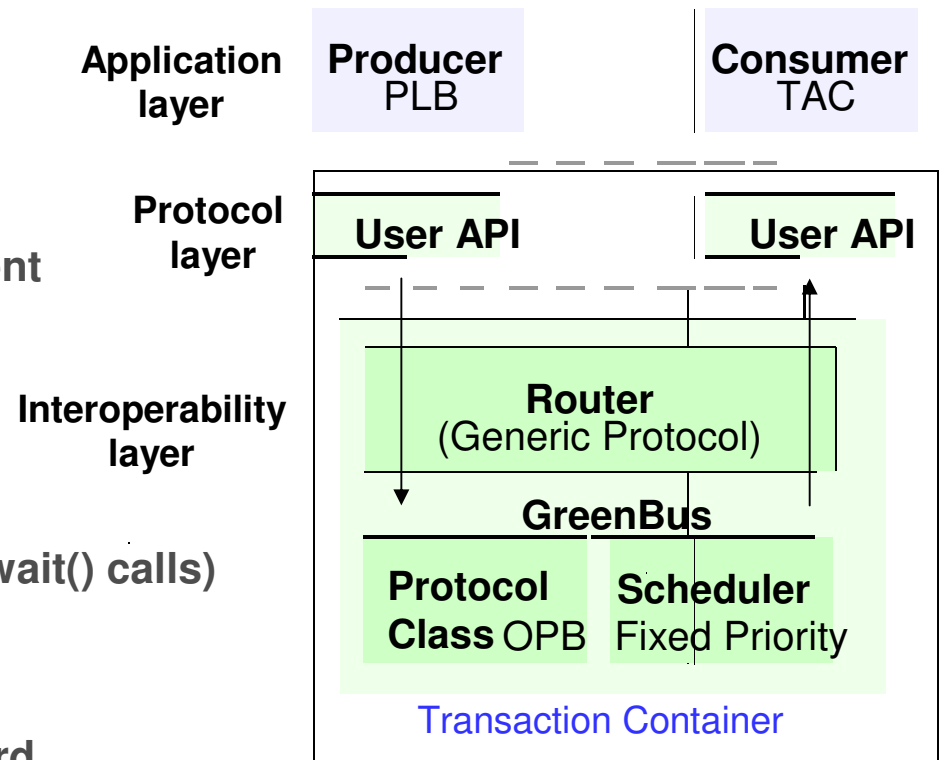


HDL Dynamics
SoC Solutions



GreenBus Overview

- **Simple System level Inter-operability**
 - Simple API between models
 - Defined Data structures
- **Safe Operation**
 - Automatic data structures management
 - Event semantics between models.
- **High Simulation Speed**
 - Data passed by Shared pointers
 - Only required events used
 - Model can use methods (rather than wait() calls)
- **Other features**
 - Built for/on proposed TLM 2.0 standard
 - Efficient (0.5 million atoms per second)
 - Multiple abstraction levels
 - Integrated analysis tools (DUST)
 - Customized support for other user API's (OCP-IP etc)



DUST Overview

Dynamic and Universal SystemC Transaction Analyzer

- Non-intrusive analysis and debugging of SystemC designs
- Easy to use (#include "utils/dust.h")
- Customizable visualization front ends
- Extensible architecture (add custom debug services, views, ...)
- Full GreenBus support today



Enhanced SystemC simulation



Analysis backend, SQL database

The screenshot shows the DUSTViewBase application window. The main area displays a SystemC diagram with a central vertical green arrow labeled 'P2S' and 'S2P' pointing up and down, and a central box labeled 'Router'. On either side of the router are several 'Master...' blocks, and one 'Slave0' block is connected to the bottom. The right-hand pane shows a 'TrData-View' table with transaction data.

Id	Value	+	...	Id	Value
1	id0xbffe4da0	+	1	2	id0xbffa938
2	id0xbffe4da0	+	2	2	id0xbffa938
3	id0xbffe4f80	+	3	4	id0xbffa9c8
4	id0xbffe4da0	+	4	8	id0xbff7e98
5	id0xbffe4da0	+	5	8	id0xbff7e98
6	id0xbffe4f80	+	6	10	id0xbff7f28
7	id0xbffe4f80	+	7	14	id0xbff5428
8	id0xbffe4da0	+	8	14	id0xbff5428
9	id0xbffe4f80	+	9	16	id0xbff54b8

BeginRef	GenRef	Id	Name	Value
1	1	1	master_id	0
2	2	2	master_id	0
3	4	4	master_id	3221203520
4	8	8	master_id	0

GenId	Id	Na...	Stre...	+	BA*	BA*	BA*	EA*	EA*	EA*
2	1	+	attri...	12	string attri...	12	stri...			
4	3	+	attri...	12	string attri...	12	stri...			
6	5	+	begin	12	string end	12	stri...			

DbRef	Id	Name	StreamId	StreamKind
my_db	1	port_stream	1	transactor
my_db	3	port_stream	3	transactor
my_db	5	port_stream	5	transactor

Refresh #: 20 Show Tableentries from: 0



Visualization, Monitoring, Debugging



HDL Dynamics
SoC Solutions

DUST JAVA-based GUI for GreenBus integrated analysis



GreenSocs – GreenBench

Configuration Framework – Critical to Proper Large Systems

Operation

- **GS_PARAM** wrapper makes module parameters configurable
- Configure parameters from within models...or via configuration file
- Configure parameters or communication architecture by config file

The screenshot displays a development environment with two windows showing C++ code and a terminal window showing the execution output.

Left Window (example_gsconfigurable.cpp):

```
1 #include "gstlm/tlm.h"
2
3 using namespace tlm;
4
5 class master : public sc_module, public gs_configurable
6 public:
7
8     SC_CTOR(master) {
9         GS_TRACE("I am the master %s", name);
10
11         // register parameters
12         GS_PARAM(addr, MAddr);
13         GS_PARAM(cmd, MCmd);
14         GS_PARAM(data, char[20]);
15     }
16
17     gs_param<MAddr> addr;
18     gs_param<MCmd> cmd;
19     gs_param<char[20]> data;
20
21
```

Right Window (sc_main.cpp):

```
1 int sc_main(int argc, char *argv[]) {
2
3     master m("master");
4
5     m.setParam("addr", "0x1000");
6     m.setParam("cmd", "1");
7     m.setParam("data", "Hello,GreenBus!");
8
9     // test: set a Parameter that has not been registered
10    m.setParam("foo", "bar");
11
12    sc_start(-1);
13
14    return EXIT_SUCCESS;
15 }
16
```

Terminal Window (Konsole):

```
klingauf@sauron:~/svndev/greenbus/trunk/gstlm/usercode/example_configurabl - Befehlsfenster Nr. 3 - Konsole
klingauf@sauron ~/svndev/greenbus/trunk/gstlm/usercode/example_configurable $ ls
example_configurable3.cpp  example_gsconfigurable2.cpp~  example_gsconfigurable.cpp  example_primchannel.cpp  test1.cfg
example_configurable.v2p  example_gsconfigurable2.o    example_gsconfigurable.o    log                       test2.cfg
example_gsconfigurable2.cpp  example_gsconfigurable2.x    example_gsconfigurable.x    Makefile                  test3.cfg
klingauf@sauron ~/svndev/greenbus/trunk/gstlm/usercode/example_configurable $ ./example_gsconfigurable2.x

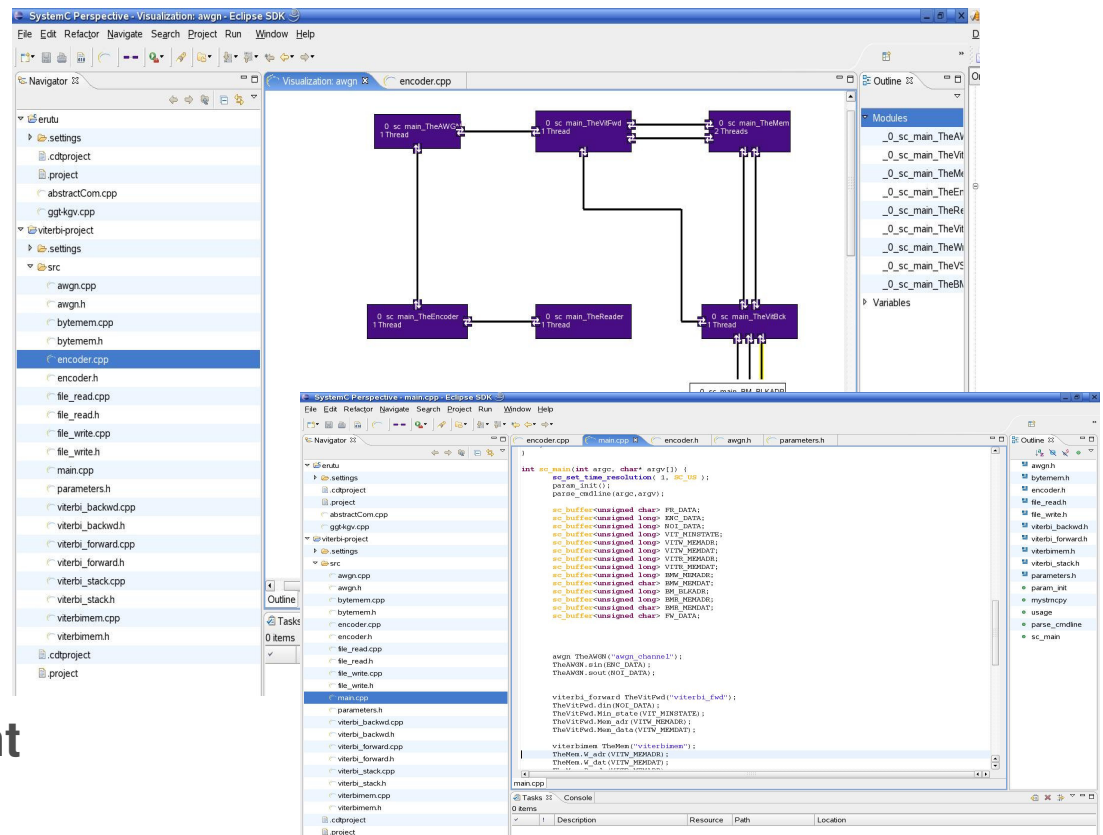
SystemC 2.2.05jun06_beta --- Feb 23 2007 10:06:17
Copyright (c) 1996-2006 by all Contributors
ALL RIGHTS RESERVED

GS_CONFIG: Reading configuration from file test3.cfg
GS_CONFIG: Configuring value of param [addr] in module [master] to [0x1000]
GS_CONFIG: Configuring value of param [cmd] in module [master] to [1]
GS_CONFIG: Configuring value of param [addr_base] in module [slave] to [0x800]
GS_CONFIG: Configuring value of param [addr_high] in module [slave] to [0x1200]
GS_CONFIG: Configuring value of param [type] in module [slave.erik] to [girl]
GS_CONFIG: Configuring value of param [color] in module [slave.erik] to [red]
```



Eclipse Environment

- **Modular and extensible SystemC designs IDE**
- **SystemC-specific features and extensions**
 - Project management
 - Configuration
 - Visualization
 - Model execution
 - Inherited CDT functionality
 - Syntax highlighting
 - Module tree viewer
 - Structural representation
 - GSPM integration
- **Availability**
 - Under subscriber agreement
 - Flexible Licensing
 - Customer-specific extensions and integration



GreenSocs SystemC Sandbox – Come Play With Us

Requirements
identification

Open Source
Community code

Subscriber funded
development

GreenSocs
Engineering



De-Facto
Standards
Support

Passed to WG
(e.g.)
OSCI, OCP-IP

Released
Packages to
users

GreenSoCs Package
Management, Repository, Wiki,

in Americas - Neal Stollon
in Europe - Mark Burton

neal.stollon@greensocs.com

mark.burton@greensocs.com



HDL Dynamics
SoC Solutions

