

Accellera SystemC Standards Update

December 2023

Accellera Systems Initiative

accellera.org



Outline

- Accellera Systems Initiative & Working Groups
- SystemC ecosystem
- SystemC Working Groups updates
- Public Repositories
- systemc.org updates
- How to join us

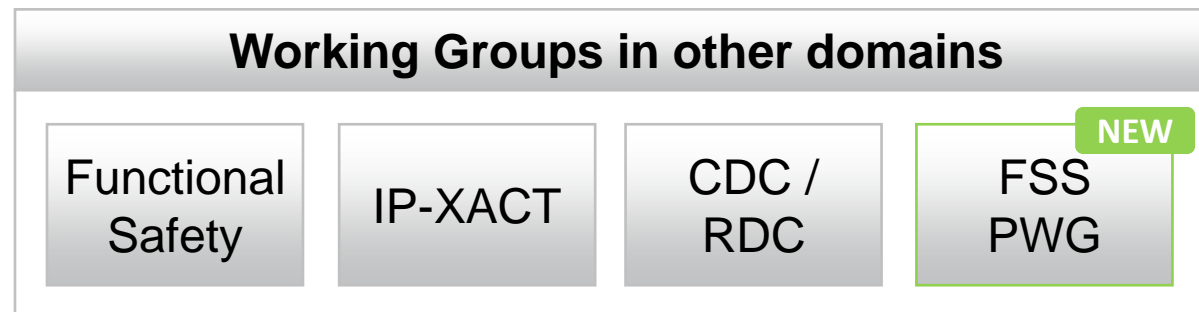
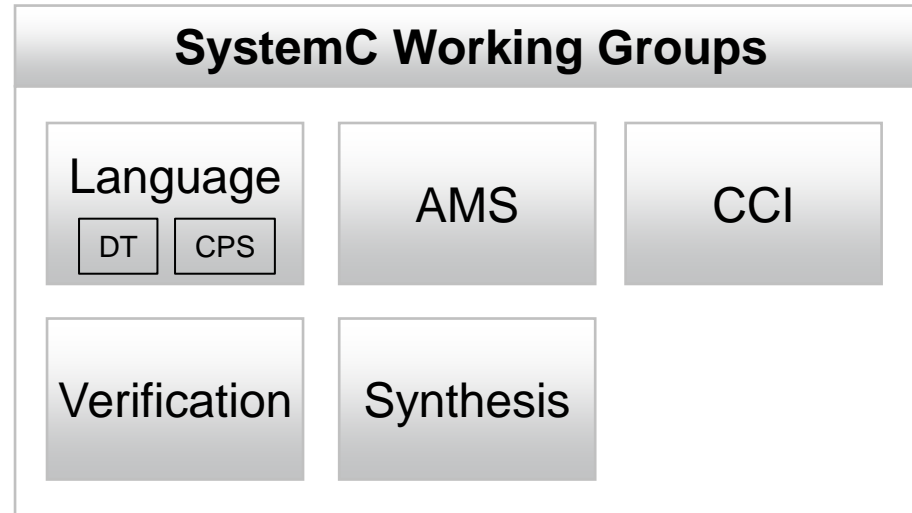
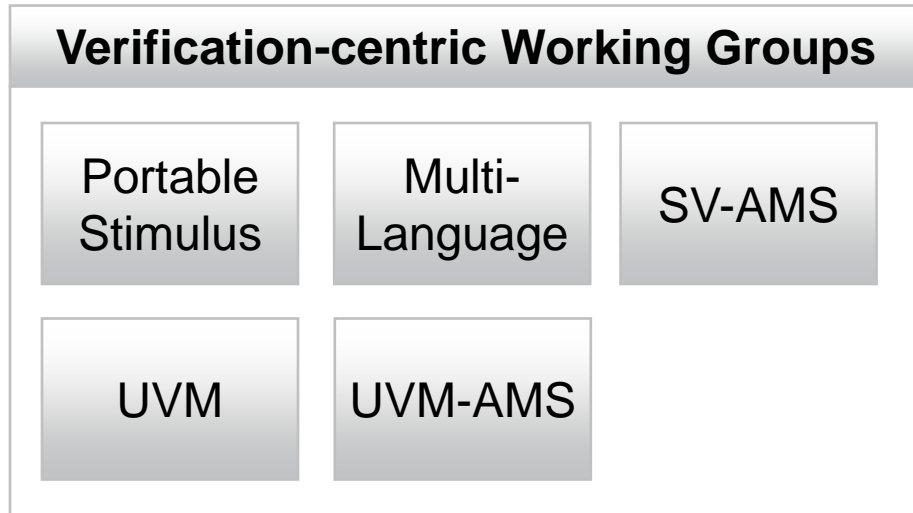
Accellera Systems Initiative

Our Mission

To provide a platform in which the electronics industry can collaborate to innovate and deliver global standards that improve design and verification productivity for electronics products.

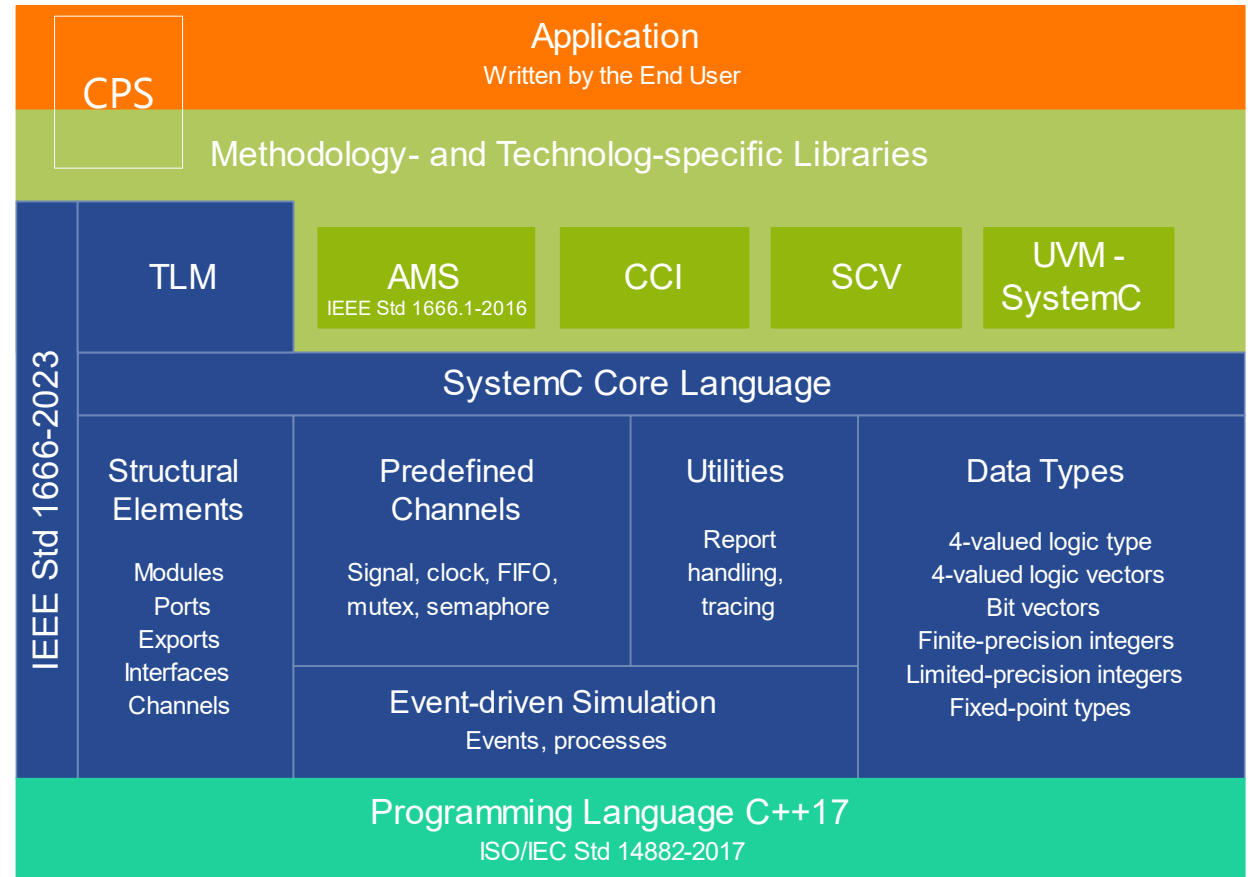


Accellera Working Groups



SystemC ecosystem

- SystemC is a C++-based language standard, widely used for
 - System-level modeling, design and verification
 - Architectural exploration, performance modeling
 - Analog/mixed signal modeling
 - High-level Synthesis
 - Software development
- Released as IEEE standards
 - IEEE Std. 1666-2023 (SystemC)
 - IEEE Std. 1666.1-2016 (SystemC AMS)



More information: <https://systemc.org/>

Accellera SystemC Working Groups

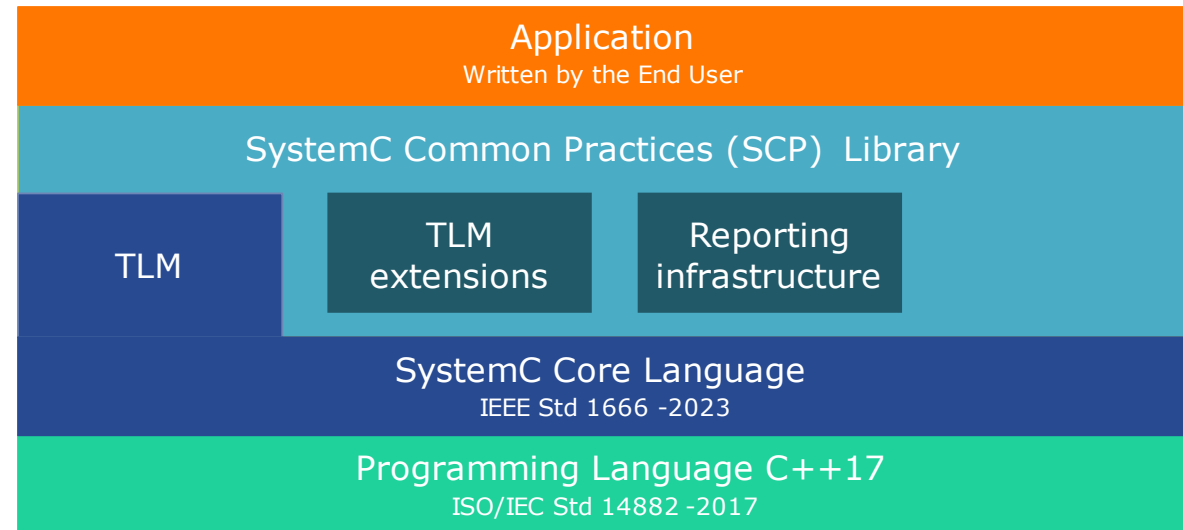
- SystemC Language Working Group (LWG)
 - Chair: Laurent Maillet-Contoz (ST)
 - Subgroups
 - Common Practices (SCP): Chair: Mark Burton (Qualcomm)
 - SystemC Datatypes (DT), Chair: Frederic Doucet (Qualcomm)
- SystemC Analog/Mixed-Signal Working Group (AMSWG)
 - Chair: Martin Barnasconi (NXP)
- SystemC Configuration, Control & Inspection Working Group (CCIWG)
 - Chair: Lukas Jünger (MachineWare)
- SystemC Synthesis Working Group (SWG)
 - Chair: Mike Meredith (Cadence) - acting
- SystemC Verification Working Group (VWG)
 - Chair: Stephan Gerth (Bosch)

SystemC Language Working Group

- SystemC Language Reference Manual released as IEEE Std. 1666-2023
 - Free download under the Get IEEE Program thanks to Accellera sponsorship:
<https://ieeexplore.ieee.org/document/10246125>
- SystemC 3.0.0 *public review* version to be released soon
 - Fully compliant with IEEE Std. 1666-2023
 - Will be made available via Accellera public repository on GitHub:
<https://github.com/accellera-official/systemc/tags>
 - Final version of 3.0.0 expected later this year.
- Next steps
 - Integrate SystemC tests into main SystemC repository.
 - Establishing CI/CD flow in the Accellera public repository on GitHub
 - Start collecting inputs and requirements for next standardization cycle

LWG - SystemC Common Practices Working Group

- Recent discussions focusing on addressing limitations in reporting and logging capabilities in the SystemC core language
- Different proposals available for review
 - Presented in September Fika and SystemC Evolution Day
 - Improvements considered for next standardization round
- This is a public repository, so the community is encouraged to submit proposals
 - <https://github.com/accellera-official/systemc-common-practices>



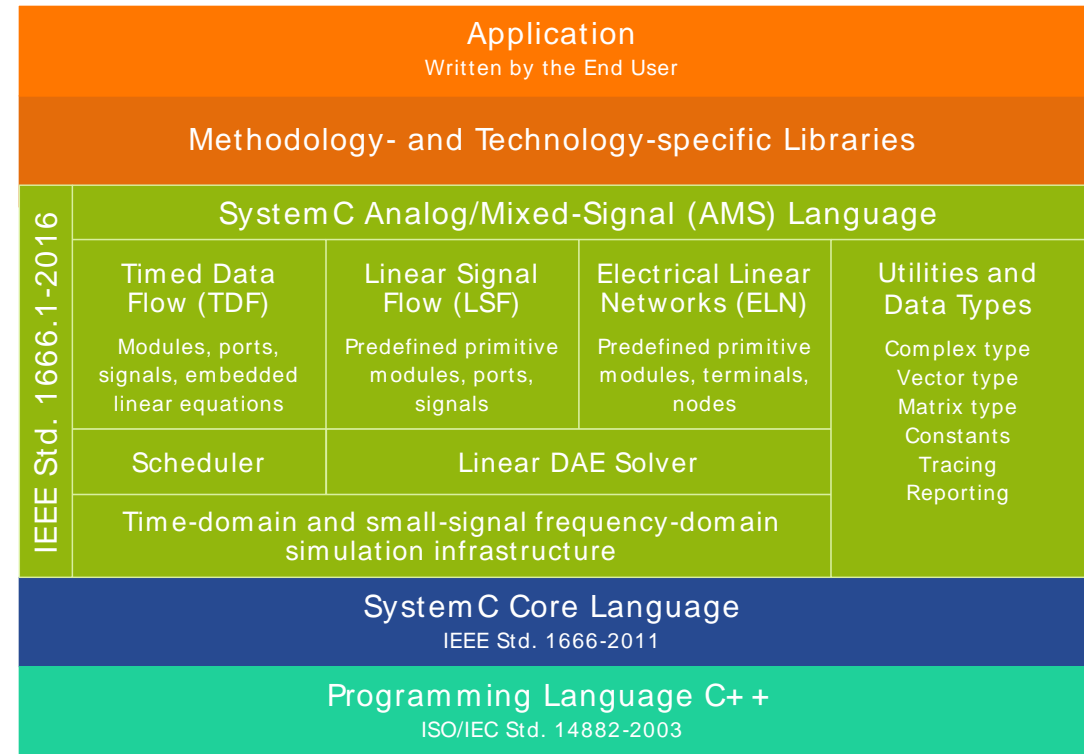
More information: <https://systemc.org/overview/systemc-scp/>

LWG - Data types Working Group

- Multi-year effort completed to address the simulation performance improvements of SystemC data types
 - Resolving many issues found in data type implementation of `sc_bigint`, `sc_biguint`, `sc_signed`, `sc_unsigned`, `sc_fixed`, and `sc_ufixed`
 - All improvements are implementation-specific, no change to the language standard / API
 - These updates are integral part of SystemC 3.0.0 public review release
- A detailed technical presentation on this data type refactoring will be shared in an upcoming SystemC Fika Event
- Special Thanks to Andy Goodrich and Fred Doucet to make this happen!

SystemC Analog/Mixed-Signal (AMS) WG

- SystemC AMS regression suite released
 - Covering more than 700 tests, covering unit-level tests, application-level tests and examples
 - <https://www.accellera.org/images/downloads/standards/systemc/systemc-ams-regressions-1.0.0.tar.gz>
- Developing extensions and enhancements as preparation for the next IEEE update (~2026)
 - Analog solver API
 - Converter primitives between LSF and ELN MoC
 - Interactive tracing and debug interface, tracing customization
 - Analog event detection
 - ...



More information

<https://systemc.org/overview/systemc-ams/>

SystemC Configuration, Control & Inspection WG

- CCI 1.0.1 reference implementation released
 - <https://github.com/accellera-official/cci/releases/tag/v1.0.1>
 - Improved build infra (automake, cmake, msvc)
 - Established basic CI/CD flow
 - Documentation updates for examples
- Proposal available for Register / Memory Inspection API
 - Ongoing discussion on implementation strategy – topic later today!

Use cases



Standard abstractions



Model information



More information

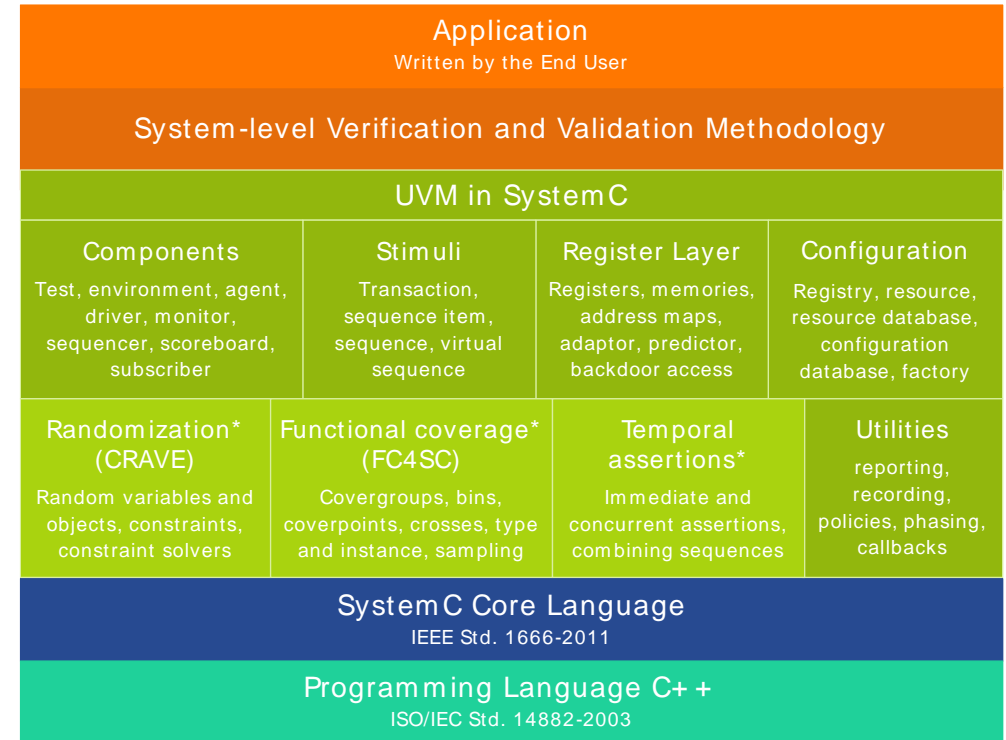
<https://systemc.org/overview/systemc-cci/>

SystemC Synthesis WG

- SystemC Synthesis Working Group is restarting its standardization alignments
 - Mike Meredith is coordinating this effort
- Considered activities by the team (not finalized / prioritized list)
 - Study impact/benefits of SystemC IEEE Std. 1666-2023 and its C++17 baseline on the Synthesis Subset Language Reference Manual
 - Next revision of the SystemC Synthesizable Subset
 - Discuss latest technologies and developments in HLS flow/tools and opportunities for standardization
- Sign-up to the Synthesis WG if you are interested to participate and contribute!

SystemC Verification Working Group

- UVM-SystemC library 1.0beta5 was released early this year
 - <https://www.accellera.org/images/downloads/draft-s-review/uvm-systemc-1.0-beta5.tar.gz>
 - Various bugfixes and enhancements to uvm_sequencer classes
- The class libraries for Functional Coverage (FC4SC) and Constrained Randomization (CRAVE) are now available via the Accellera public repositories
 - <https://github.com/accellera-official/fc4sc>
 - <https://github.com/accellera-official/crave>
- Current focus on supporting SystemC 3.0.0



* Integration on Roadmap

More information

<https://systemc.org/overview/systemc-verification/>

Accellera Public Repositories

- The number of Accellera Public Repositories is growing!
- More information: <https://github.com/accellera-official/>

cci Public
SystemC Configuration, Control and Inspection (CCI)
C++ 4 stars Apache-2.0 license 7 forks 16 issues 1 pull request Updated last week

systemc.org Public
Source code for systemc.org
HTML 5 stars Apache-2.0 license 7 forks 2 issues 0 pull requests Updated last week

systemc Public
SystemC Reference Implementation
C++ 353 stars Apache-2.0 license 122 forks 26 issues 11 pull requests Updated 3 weeks ago

fc4sc Public Added this year
Functional Coverage for SystemC (FC4SC) library which provides mechanisms for functional coverage definition, collection and reporting.
C++ 1 star Apache-2.0 license 1 fork 13 issues 0 pull requests Updated on Jul 12

uvm-core Public Added this year
SystemVerilog 7 stars Apache-2.0 license 2 forks 0 issues 0 pull requests Updated on Jul 7

crave Public Added this year
Constrained RANdom Verification Enviroment (CRAVE)
C++ 6 stars Apache-2.0 license 0 forks 11 issues 0 pull requests Updated on Jun 14

uvm-tests Public Added this year
2 stars 1 fork 0 issues 0 pull requests Updated on Jun 6

systemc-common-practices Public
SystemC Common Practices (SCP)
C++ 8 stars Apache-2.0 license 8 forks 7 issues 0 pull requests Updated on Mar 24

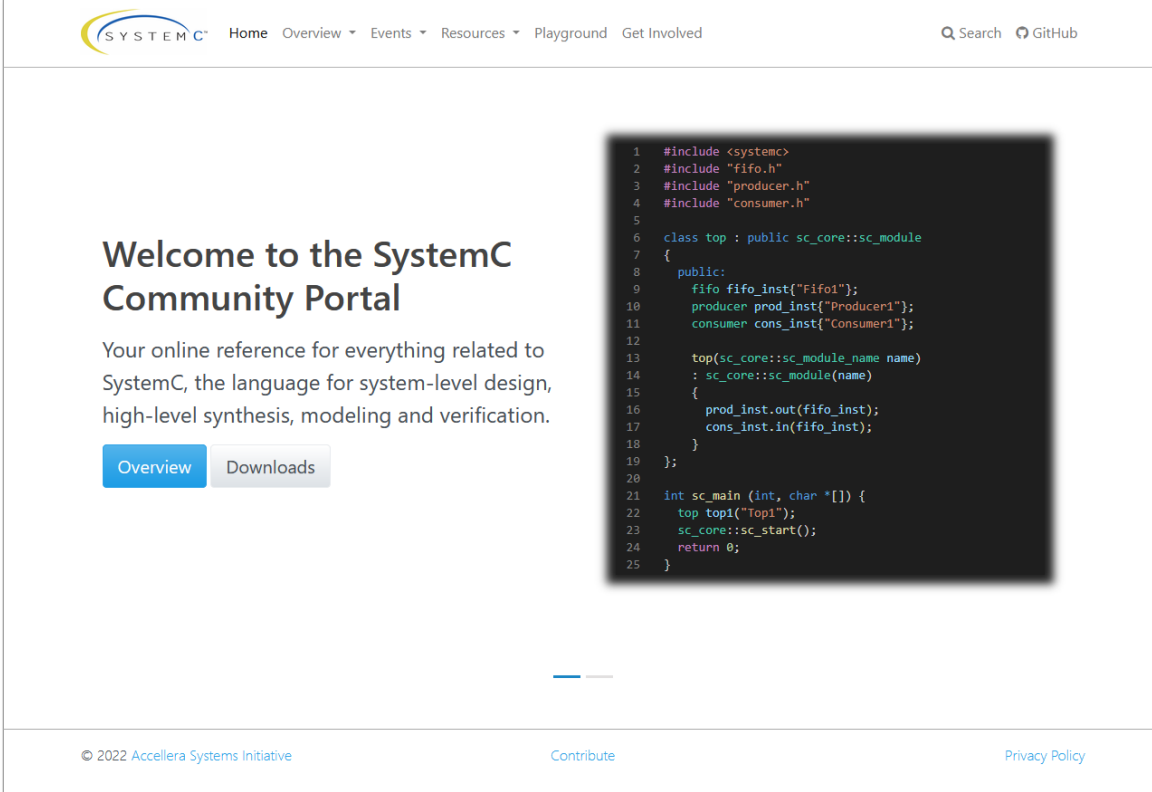
PySysC Public
Public repository for the SC Common Practices Subgroup
Python 36 stars Apache-2.0 license 5 forks 3 issues 1 pull request Updated on Oct 27, 2021

systemc-regressions Public
SystemC Regressions Test Suite
5 stars Apache-2.0 license 0 forks 1 issue 0 pull requests Updated on Jul 25, 2019



systemc.org Updates

- New content added
 - SystemC overview pages covering all Working Groups
 - SystemC Evolution Day Events and Fikas: all presentations and videos
 - Open Access Publications
 - Libraries and Projects
- **YOU** can help in adding content!
 - Submit your pull request to github.com/accellera-official/systemc.org



The screenshot shows the systemc.org website. The header includes the SYSTEMC logo, navigation links (Home, Overview, Events, Resources, Playground, Get Involved), a search bar, and a GitHub link. The main content area features a "Welcome to the SystemC Community Portal" message, a description of the site as an online reference for SystemC, and two buttons: "Overview" and "Downloads". A code snippet is displayed on the right side of the page, showing SystemC code for a producer-consumer example. The footer contains copyright information for the 2022 Accellera Systems Initiative, a "Contribute" link, and a "Privacy Policy" link.

```
1 #include <systemc>
2 #include "fifo.h"
3 #include "producer.h"
4 #include "consumer.h"
5
6 class top : public sc_core::sc_module
7 {
8 public:
9     fifo fifo_inst("Fifo1");
10    producer prod_inst("Producer1");
11    consumer cons_inst("Consumer1");
12
13    top(sc_core::sc_module_name name)
14    : sc_core::sc_module(name)
15    {
16        prod_inst.out(fifo_inst);
17        cons_inst.in(fifo_inst);
18    }
19 };
20
21 int sc_main(int, char *[]) {
22     top top1("Top1");
23     sc_core::sc_start();
24     return 0;
25 }
```

How to join us

- Become an Accellera Working Group member
 - [Join Accellera](#) and participate in the Accellera working groups
 - Direct access to the latest standardization proposals and development implementations
- Become a member of the IEEE Standards Association
 - Join [IEEE-SA](#) to participate in the IEEE P1666 (SystemC) working group
- Share your experiences
 - Visit www.accelera.org and join the community forums at forums.accelera.org
 - Report your issues and/or create pull requests on the public SystemC [GitHub](#) repository
- Help us to grow the SystemC ecosystem and community
 - Participate in community events such as the [SystemC Evolution Day and Fika](#)
 - Contribute to the SystemC Community Portal systemc.org
 - Promote the use of the SystemC standard in complex system simulation tasks

Thank You

Q&A

