Accellera SystemC Standards Update

SystemC Evolution Day 2022

Martin Barnasconi

Accellera Technical Committee Chair

accellera.org

accellera SYSTEMS INITIATIVE™

SYSTEMC™ EVOLUTION DAY

DEC 8, 2022 | MUNICH | GERMANY

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 SYSTEMS INITIATIVE™

SYSTEMC™ EVOLUTION DAY
DEC 8, 2022 | MUNICH | GERMANY

Accellera Working Groups

Verification-centric Working Groups

- Portable Stimulus
- Multi-Language
- SV-AMS
- UVM
- UVM-AMS

SystemC Working Groups

- Language
  - DT
  - CPS
- AMS
- CCI
- Verification
- Synthesis

Working Groups in other domains

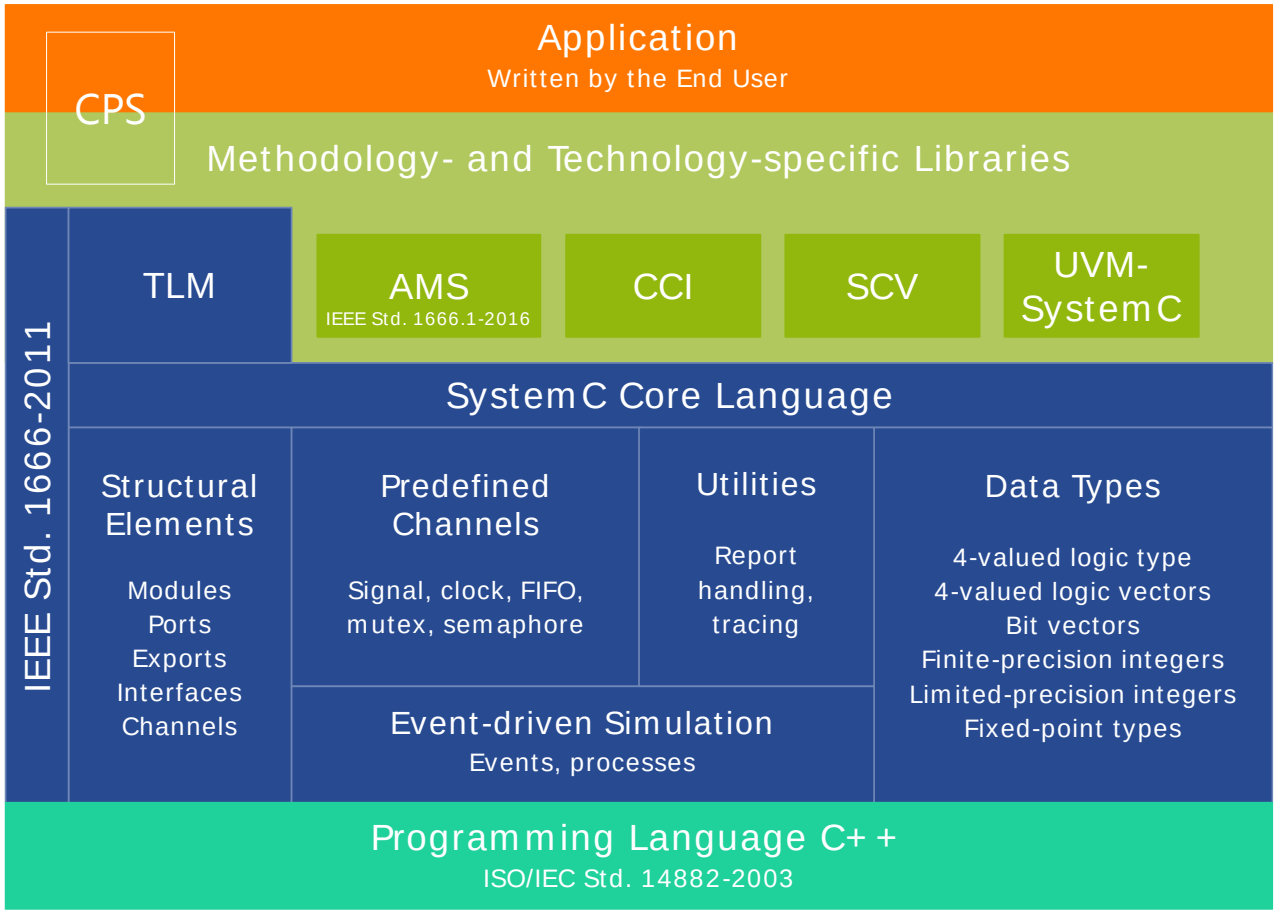
- Functional Safety
- IP-XACT
- IP Security Assurance
- CDC / RDC

accellera SYSTEMS INITIATIVE™

SYSTEMC™ EVOLUTION DAY
DEC 8, 2022 | MUNICH | GERMANY

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. 1666-2011 (SystemC)
  - IEEE Std. 1666.1-2016 (SystemC AMS)



More information: https://systemc.org/

accellera SYSTEMS INITIATIVE™

SYSTEMC™ EVOLUTION DAY
DEC 8, 2022 | MUNICH | GERMANY

Accellera SystemC Working Groups

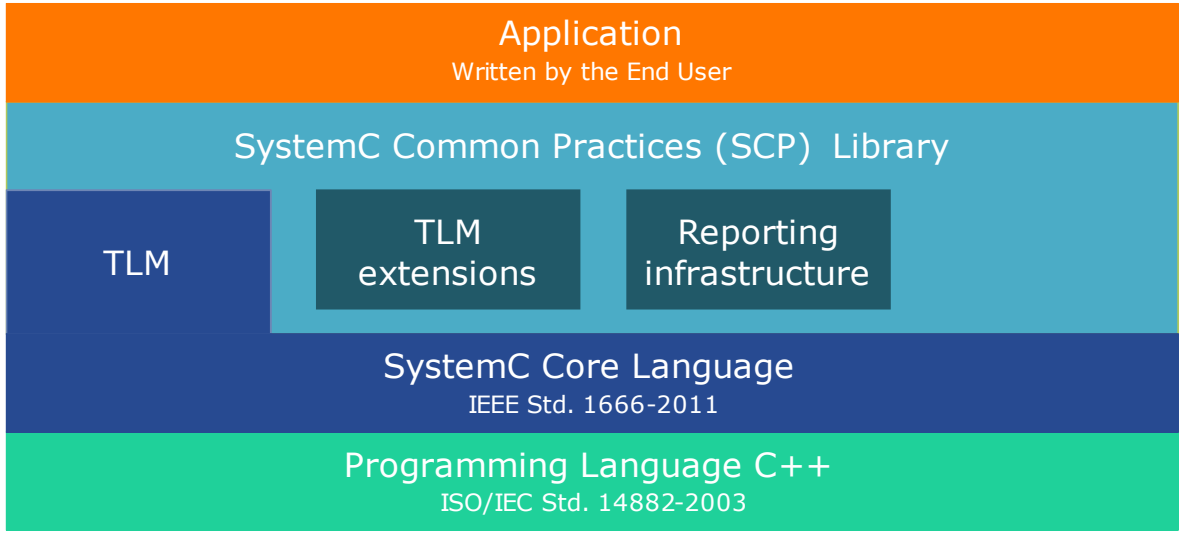
- SystemC Language Working Group (LWG)
  - Chair: Laurent Maillet-Contoz (ST)
  - Subgroups
    - Common Practices (SCP): Chair: Mark Burton (Qualcomm)
    - SystemC Datatypes (DT), Chair: Frederic Doucet (Qualcomm)
- SystemC Analog/Mixed-Signal Working Group (AMSWG)
  - Chair: Martin Barnasconi (NXP)
- SystemC Configuration, Control & Inspection Working Group (CCIWG)
  - Chair: Ola Dahl (Ericsson)
- SystemC Synthesis Working Group (SWG)
  - Chair: *Andres Takach (Mentor)*
- SystemC Verification Working Group (VWG)
  - Chair: Stephan Gerth (Bosch)

SystemC Language Working Group

- Main activities this year
  - Propose and review Language Reference Manual updates from/to IEEE-SA P1666
  - Validating new and updated API for the next revision of the SystemC standard
  - `Sanitizing` SystemC reference implementation (2.3.4) including other coding improvements focusing on stability, robustness and scalability
- SystemC 2.3.4 has been released on the Accellera public repository on GitHub!
  - https://github.com/accellera-official/systemc/tags
- Activities in the subgroups
  - Ongoing study to improve performance of updated SystemC datatypes
  - Growing collection of SystemC Common Practices thanks to community contributions
- Future plans
  - Revamp SystemC regression environment – leveraging automation in GitHub
  - Release SystemC reference implementation the next revision of the IEEE SystemC standard

SystemC Common Practices Working Group

- The SystemC Common Practices Working Group ‘grew’ from SystemC Evolution Day – please be proud! And PLEASE contribute.
- The repository is public: https://github.com/accellera-official/systemc-common-practices
- There are a growing range of “common practice” items already in and more on the way
- A big block this year is an extensive, efficient and easy to use reporting library



SystemC Common Practices - New this year

- Initial extensions and CCI parameter definitions – WIP
- Reporting library

```
scp_debug(SCMOD) << "You text stream";
```

  - Support fatal, error, warning, info, debug and trace.
  - Extremely efficient (optimized so you can use everywhere in your code).
  - Controllable by CCI parameters (e.g. top.my.module.log_level=4)
    - CCI params 'hierarchical' (e.g. top.log_level = 1 applies to lower level modules)
  - Uses sc_report_handler::report under the hood, so as to 'play nicely' with existing SystemC code, and with the standard.
  - Many formatting options (implemented through the SC_REPORT_ mechanisms).
  - (optionally) Uses spdlog which also handles async logging

SystemC Common Practices - Next year…

- Registers …
  - The CCI WG is currently looking into the interface, our plan is to provide an implementation.

- Multi-threading and Multi-process
  - See talk later today !!!!
  - Will include ‘multi-threaded’ quantum keepers, and multi-process bridges.

Anything else you would like to see?

What could you contribute?

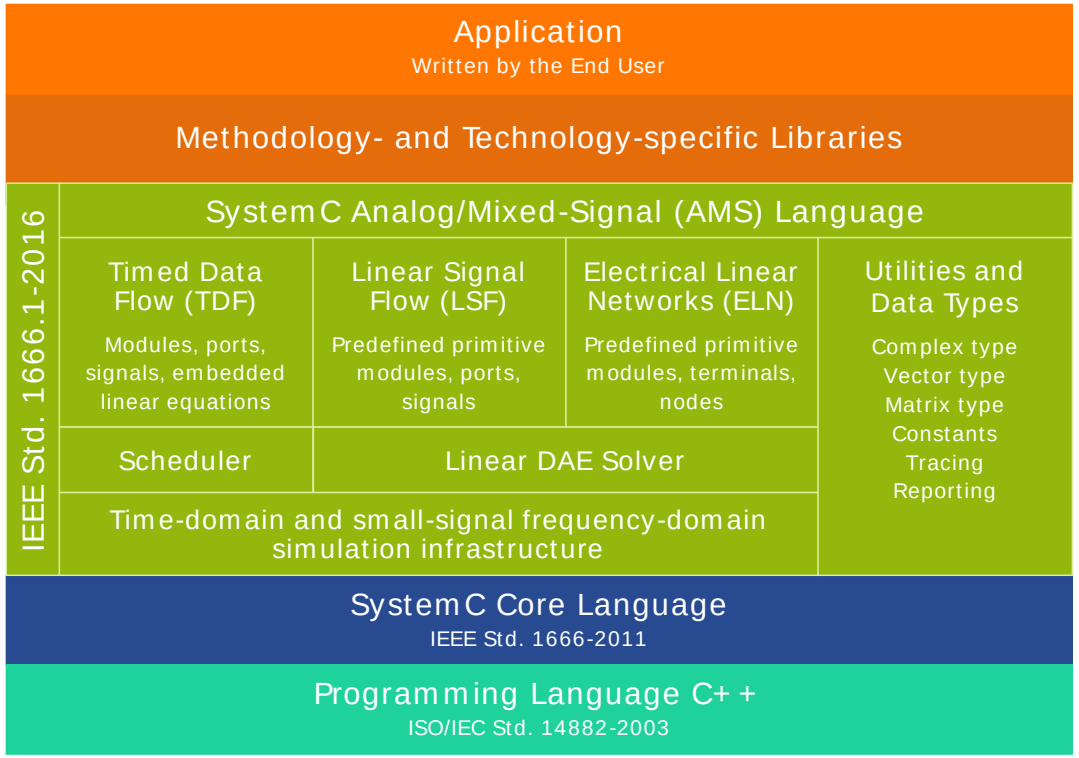
Data types Working Group

- Simulation performance improvements for sc_bigint and sc_biguint
  - implemented and verified in branch
  - will be merged into mainline after 2.3.4 release
- Future work (2023)
  - Future improvements to datatypes speed (sc_int)
  - Header only include
  - Type traits (constexpr access to number of bits, rounding mode, etc)
  - Derived types (“grow by 1 bits”, “remove two bits and round”, “add saturation” etc)
  - New data types being considered: sc_complex and sc_float

Please join and contribute! ☺

SystemC Analog/Mixed-Signal (AMS) WG

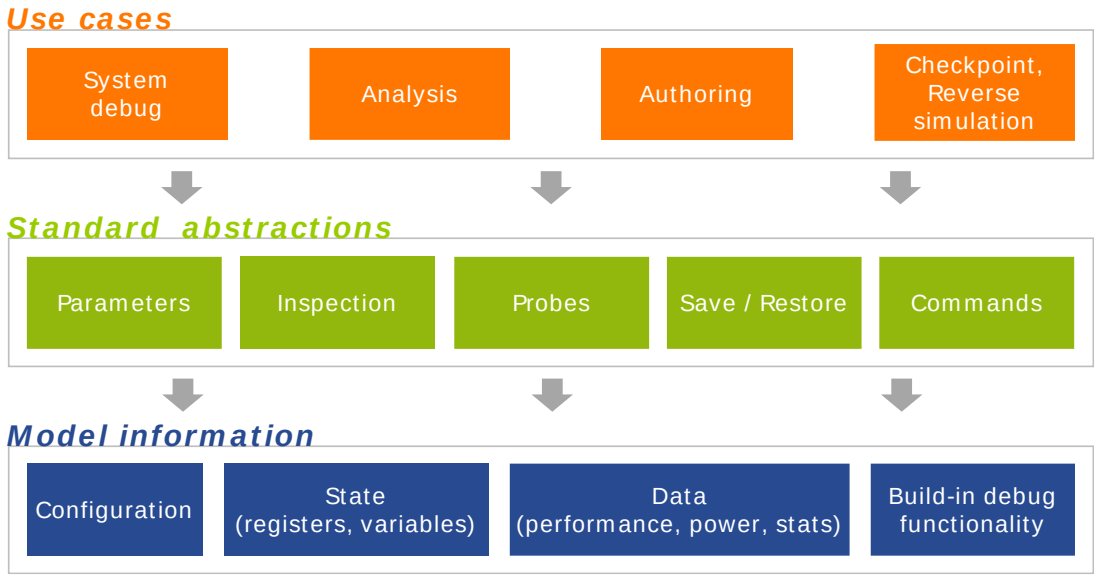
- Main activities this year
  - Complete the SystemC AMS regression suite
  - Developing extensions and new features for the next revision of IEEE 1666.1 (SystemC-AMS)
- SystemC AMS regression suite will be released soon!
  - Covering **more than 700** tests, covering unit-level tests, application-level tests and examples
  - Final testing being conducted with SystemC 2.3.4 and SystemC-AMS Proof-of-Concept library 2.4 delivered by Accellera member company



More information
https://systemc.org/overview/systemc-ams/

SystemC Configuration, Control & Inspection WG

- Main activities this year
  - Alignment with SystemC LWG on the use of `cci_value` vs. `sc_any_value`
  - Clean-up of CCI reference implementation with improved build infrastructure (automake and cmake) - CCI 1.0.1 being released soon!
- CCI repository is now public:
  - https://github.com/accellera-official/cci
- Proposal reviewed for a Register / Memory Inspection API
  - Collaboration with SystemC Common Practices WG to coordinate implementation and testing



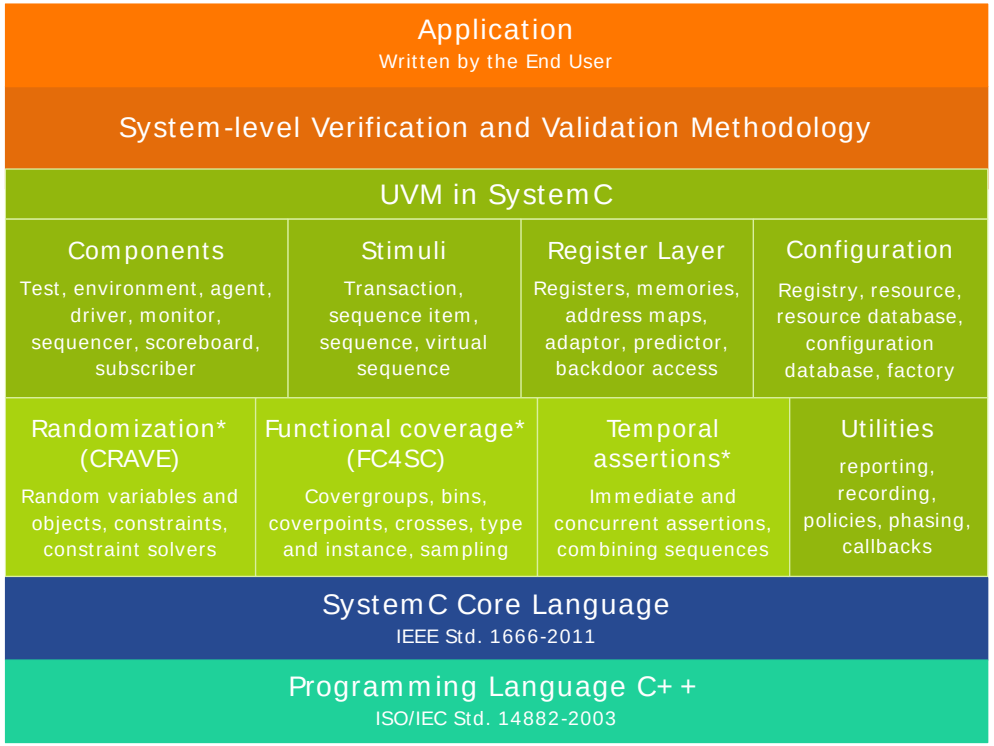
More information
https://systemc.org/overview/systemc-cci/

SystemC Synthesis WG

- The SystemC Synthesis Working Group is responsible for the SystemC synthesizable subset, to enable synthesis of digital hardware from high-level specifications
- **Current status**
  - Released the SystemC Synthesis Subset Language Reference Manual version 1.4.7 in 2017
- **Developments and future plans**
  - Working Group defining next revision of the SystemC Synthesizable Subset, including:
  - Alignment and consolidation on SystemC Datatypes to enhance HLS flows
  - Update and finalize support of modern C++ language features defined in C++11/14/17

SystemC Verification Working Group

- Main activities this year
  - Development of UVM in SystemC standard and reference implementation
  - Standardization of Constrained Randomization API
  - Extending UVM-SystemC “UBUS example” in with Constrained Randomization using CRAVE
- Release and public review of UVM-SystemC library 1.0beta4 earlier this year
- Build-flow improvements and other enhancements in CRAVE implementation

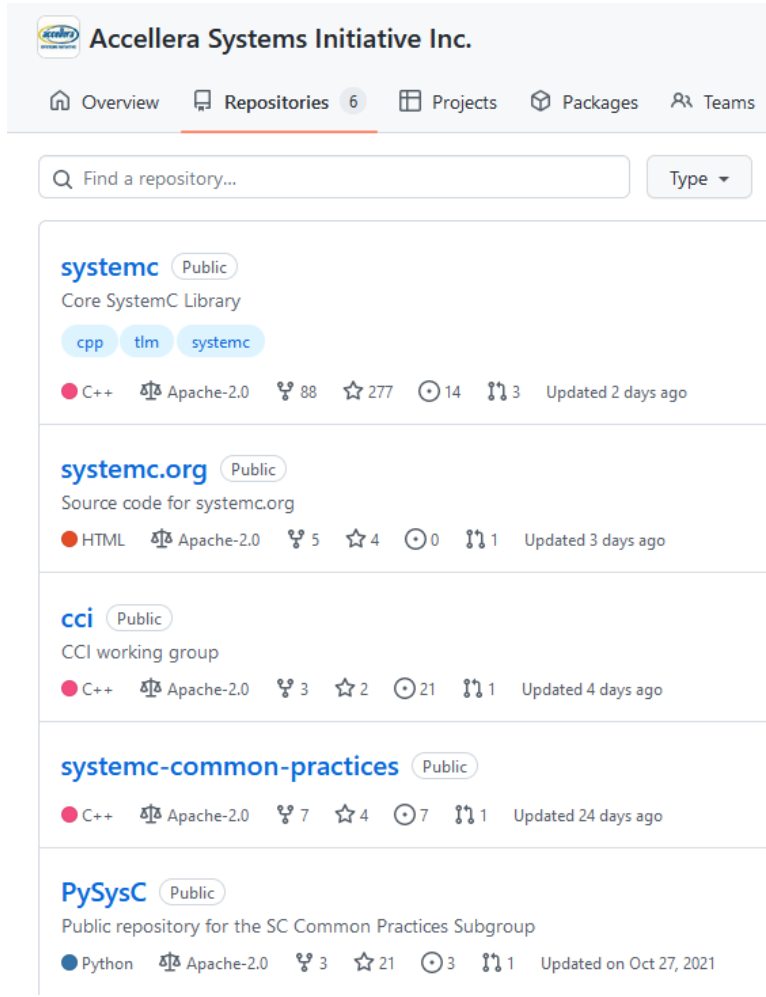


* Integration on Roadmap

More information
https://systemc.org/overview/systemc-verification/

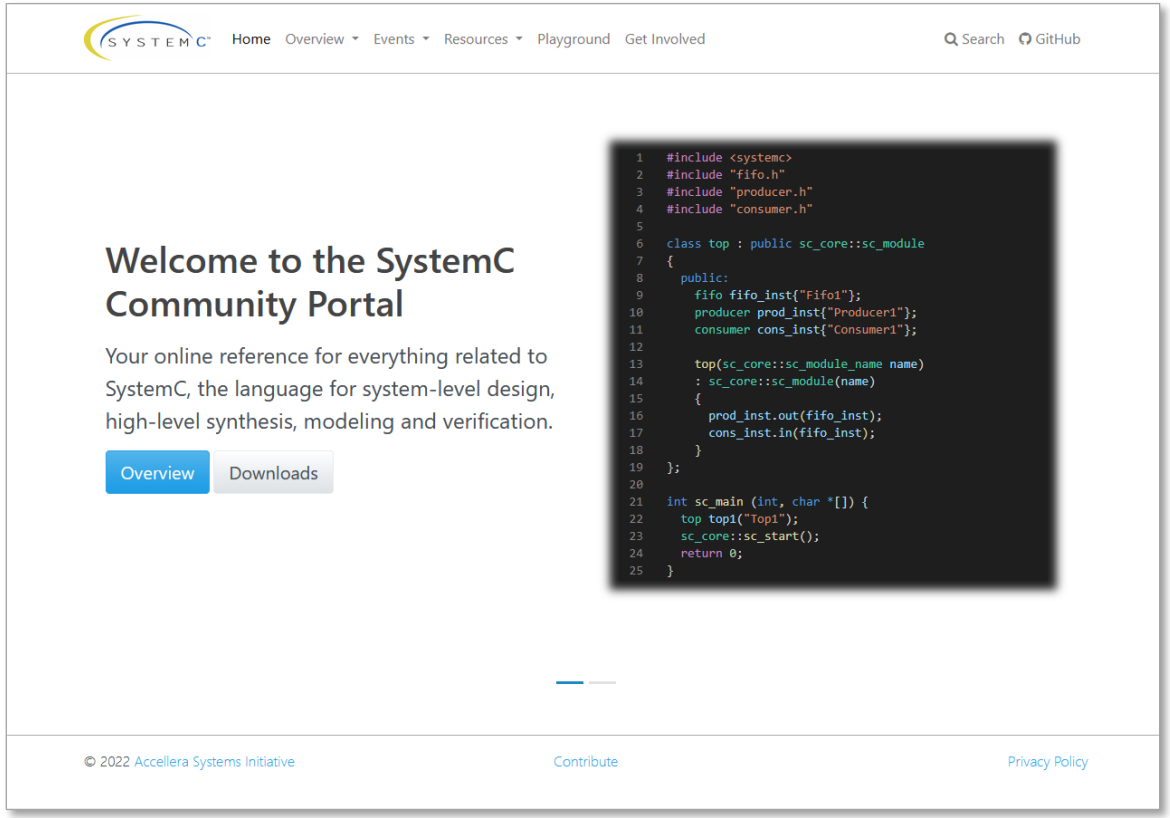
Accellera Public Repositories

- Accellera Public Repositories: https://github.com/accellera-official/
- SystemC Working Groups with public repositories
  - SystemC: https://github.com/accellera-official/systemc
  - CCI: https://github.com/accellera-official/cci
  - SystemC Common Practices: https://github.com/accellera-official/systemc-common-practices https://github.com/accellera-official/PySysC
  - SystemC.org website https://github.com/accellera-official/systemc.org
- Objective to make more working group repositories public



systemc.org Updates

- New content added in 2022
  - 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



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.accellera.org and join the community forums at forums.accellera.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