SystemC Fika May 2024
CCI Working Group Update

Lukas Jünger, Chair of the CCI WG
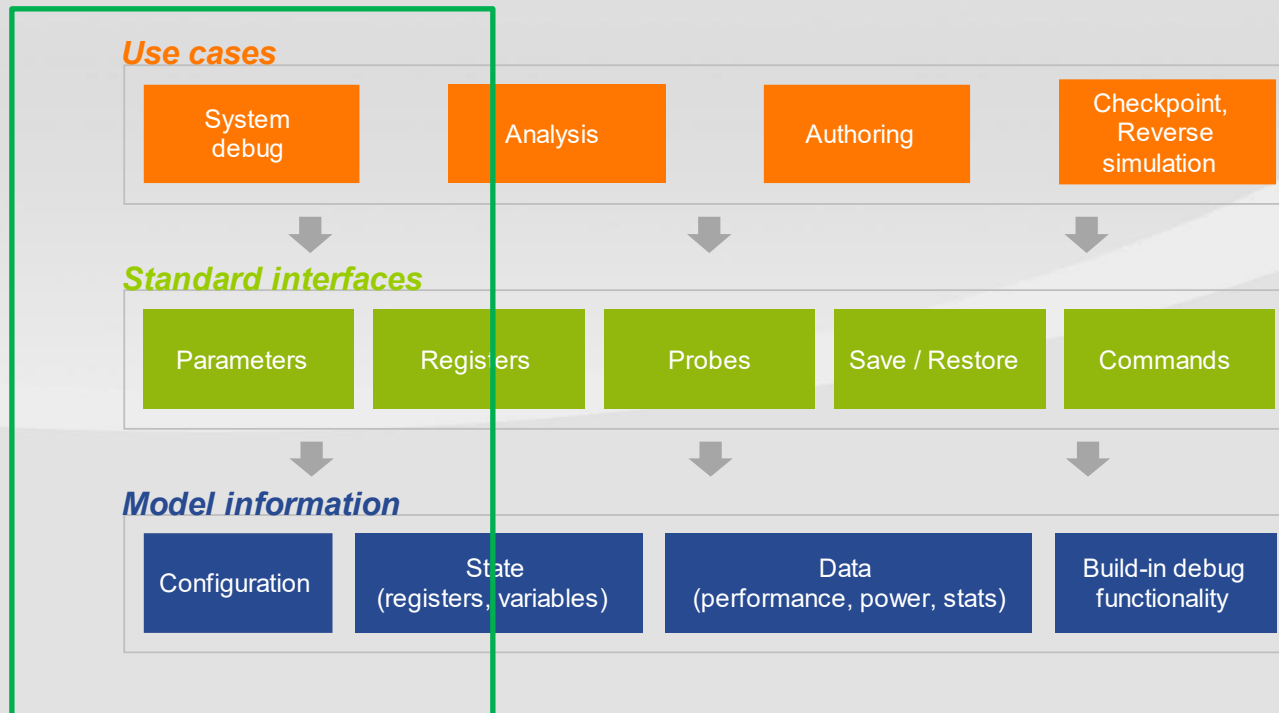
Presentation Copyright Permission

What is CCI?

- **Accellera Standard SystemC Configuration, Control and Inspection**

- **Goal: Make SystemC models more interoperable**
  - E.g. by providing a common way for model parameter specification

What's going on in the CCI WG?

- **Continous Integration Improvements**

- **Build Flow Improvements**
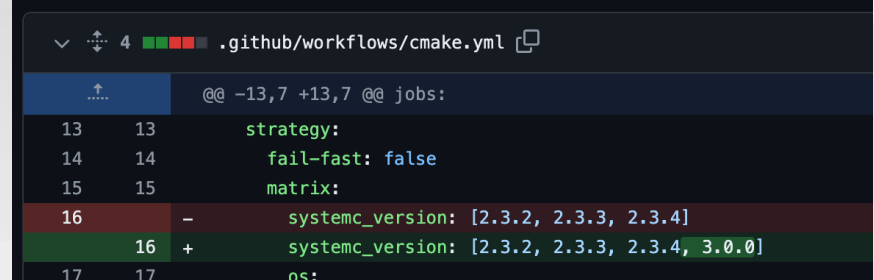
- **On-Going Discussion: Memory Inspection API**

CCI CI

- **CCI WG employing a GitHub Actions-based CI flow**
  - Automatic CI builds for: Linux, Windows, MacOS with Clang, GCC and MSVC
  - SystemC 2.3.2, 2.3.3, 2.3.4, 3.0.0 included
  - Working well overall, issues remain for Windows GCC and Clang builds
  - Similar flow upstreamed to SystemC kernel

- **Added SystemC 3.0.0 to CI**
  - 1 line change (+ minor cmake fix)
  - CCI supports SystemC 3.0.0 😊


```
   ∨  ⊕  4 ■■■■  .github/workflows/cmake.yml  ⧉

   ⬆            @@ −13,7 +13,7 @@ jobs:

   13    13         strategy:
   14    14             fail-fast: false
   15    15             matrix:
   16         −            systemc_version: [2.3.2, 2.3.3, 2.3.4]
         16   +            systemc_version: [2.3.2, 2.3.3, 2.3.4, 3.0.0]
   17    17             os:
```

- **Looking for Windows contributions**
  - Windows build for GCC and Clang succeed, but tests fail in CI
  - "Good first issue"
  - Community contributions very welcome!

Build Flow Improvements

- **Automake flow is deprecated and will be removed soon**

- **Cmake flow improvements around SystemC discovery**
  - add_subdirectory, CMake package, SYSTEMC_HOME

Getting started:
https://mwa.re/cci-example

Memory Inspection API

- **So far CCI only addresses (C)onfiguration**
  - Extend to (C)ontrol and (I)nspection

- **Proposal: Memory Inspection**
  - Original proposal available
  - POC implementation exists

- **Discussion topics**
  - Minimze effort for library providers
  - Keep it optional
  - Reuse existing CCI infrastructure

- **Looking for interested parties to join the discussion**

MOTIVATION

- Model
  - Addressable memory, registers
  - Behavior (reset, r/w actions, …)

- Tool
  - Inspect, monitor, modify
  - Visualize, …

memory

register block

| 0x10 |
| 0x1C |
| 0x08 |
| 0x04 |
| 0x00 |

CCI Mem API

- Tool independent
- Memory base-class independent

System

Your Memory

Your Registers

| 32 |
| 16 |
| Status OK |
| "cci" |
| 0x04 |