Re-Envisioning CCI Inspection

Bill Bunton Intel
Philipp Hartmann Intel
Michael Lebert Ericsson
Ola Dahl Ericsson

Presentation Copyright Permission

– A non-exclusive, irrevocable, royalty-free copyright permission is granted by Intel & Ericsson to use this material in developing all future revisions and editions of the resulting draft and approved Accellera Systems Initiative **SystemC** standard, and in derivative works based on this standard.

What Should Be Inspected

- Memory
- Control and Status Registers (CSR)
- Model Variables
  - Architectural registers, pointer, etc.
  - Simulation-only variables
- Address space exploration is beyond the scope of this discussion

What is Inspection

- Inspect (read)
- Modify (write)
- Change Notification (callback)
- Retrieve Metadata
  - Module name (sc_module name)
  - Description
  - CSR Field Descriptions

Inspectable Memory

- Modeled Read/Write storage only
- RAM, ROM, Flash, etc.
- Provides storage only
- Accessed using an memory local offset (address)
- No side effects (w)
- Not required to be part of model address space
- May not have a TLM interface

Inspectable Control and Status Registers

- Typically provide processor to hardware interface
  - Memory Mapped read/write
  - Direct bit and field access by hardware
- Typically organized as multi-byte registers
  - 16, 32, 64, 128 and larger
  - May contain multiple bit-level fields
- Accessed using an CSR bank local offset (address)
- Not required to be part of model address space
- May not have a TLM interface

Inspectable Simulation & Architectural Variables

- C++ Data Type
- SystemC Data Type
- Common Templated Data Structures
- Arbitrary Structures (CCI value)

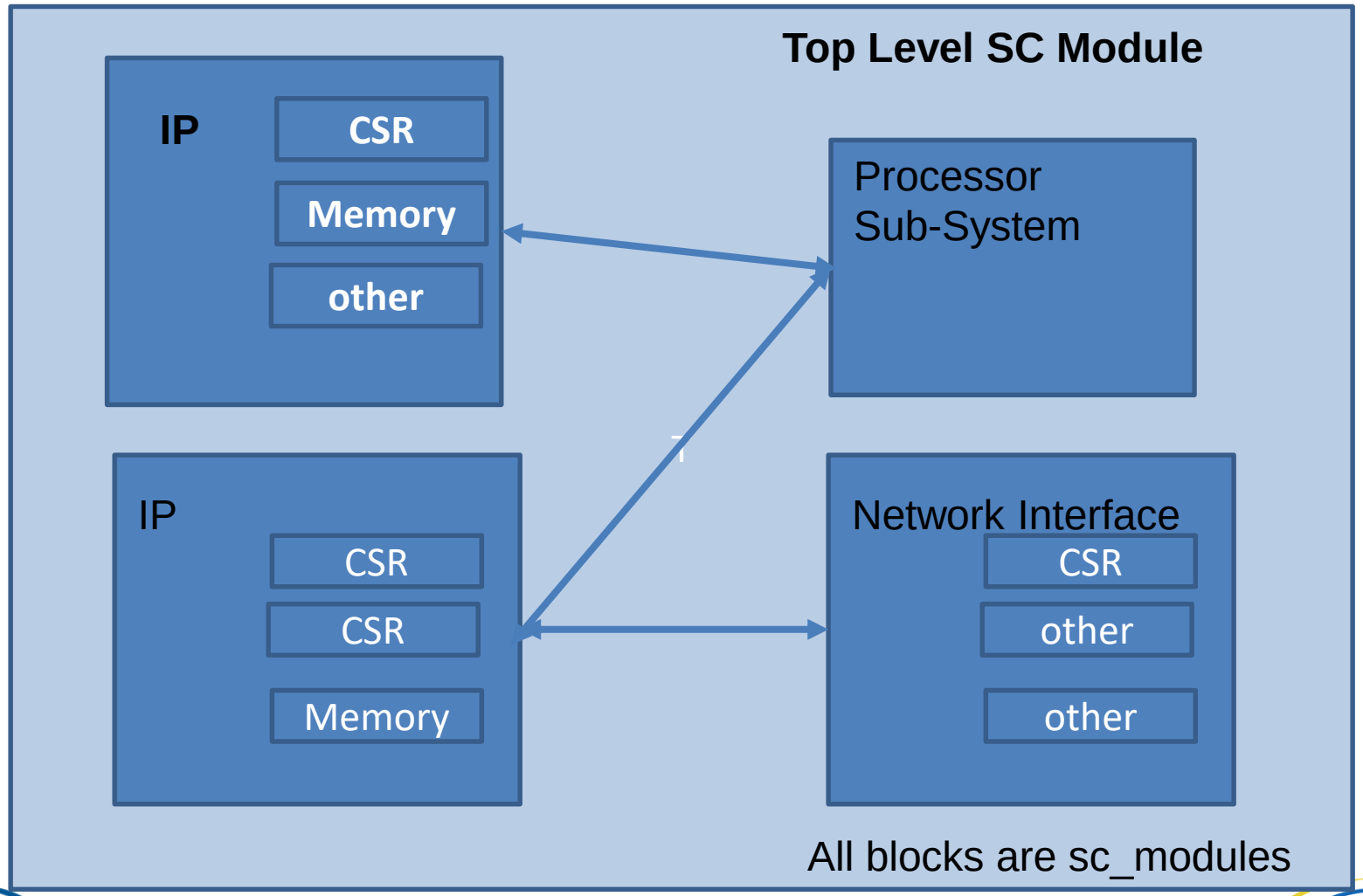
CCI Inspection Development Goals

- Define an implementable standard which provides the features described above
- Minimize needed meta-data storage in simulator code
- Rely on the external tool when practical
- Leverage CCI Configuration concepts and code
- Implement CCI Inspection as part of the delivery structure

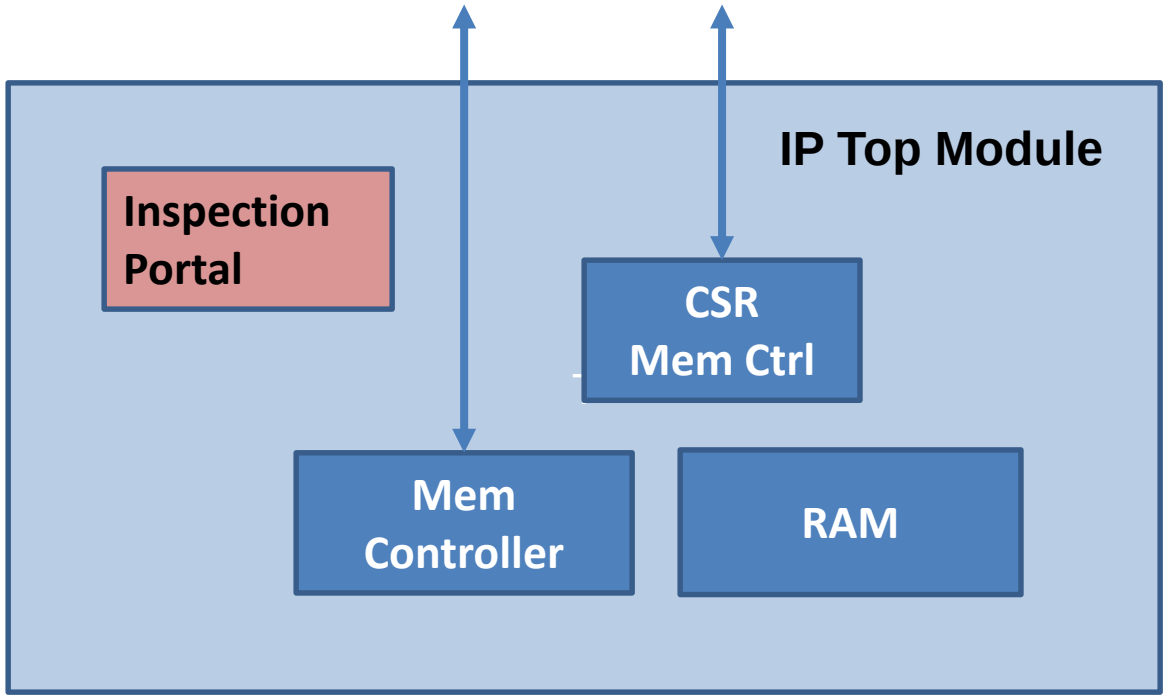
CCI Inspection
Development Non-Goals

- Extract model's Address map from TLM structure
- Standardize a tool that generates SystemC code from register description.
- Standardize any other SystemC code generator tool.
- Prefer a standard register description language or proprietary register description over others.

Simulation Model Structure



IP Model with Inspection Portal



All block are sc_module

Tools View
Operational Overview

- Get list of inspection portals
- Get portal handle
- Get list of (memories, register banks, or variables)
- Use handle to
  - Inspect
  - Modify
  - Register Call-backs

Inspection Interfaces (1of3)

- There are five interface definitions
  - Portal interface
  - Memory inspection interface
  - Register Bank inspection interface
  - Register inspection interface
  - Variable inspection interface
- All five share two methods
  - name()
    - returns sc_module hierarchical name
  - get_description returns
    - text description

Inspection Interfaces (2of3)

- The portal interface has methods to:
  - List inspectable memories, register blocks & variables
  - Get a handle to any of the inspectable items

- Inspectable memories and register banks have methods for:
  - Bus Read & Write operations (offset and char* for data)
  - Call-back for read or write operations

Inspection Interfaces (3of3)

- Register banks also have:
  - Call-back for non-write register modification
  - Get register handle

- Variables are type specific and have methods
  - Typed read & write operations
  - Call-backs for variable accessed and modified

Next Steps
Formula 1 Race
Circuit of The Americas (COTA)

Bill Bunton image creator

Next Steps
CCI Inspection

- Align terminology with existing standards (IP-XACT)
- Complete internal prototypes
- Get corporate signoff for contribution to Accellera
- Restart the CCI working group
- Make the prototype code base available to the CCI working group

Discussion?