SystemVerilog-like Assertions in SystemC

Muhammad Hassan
German Research Center for Artificial Intelligence (DFKI), Bremen, Germany
University of Bremen, Germany
muhammad.hassan@dfki.de

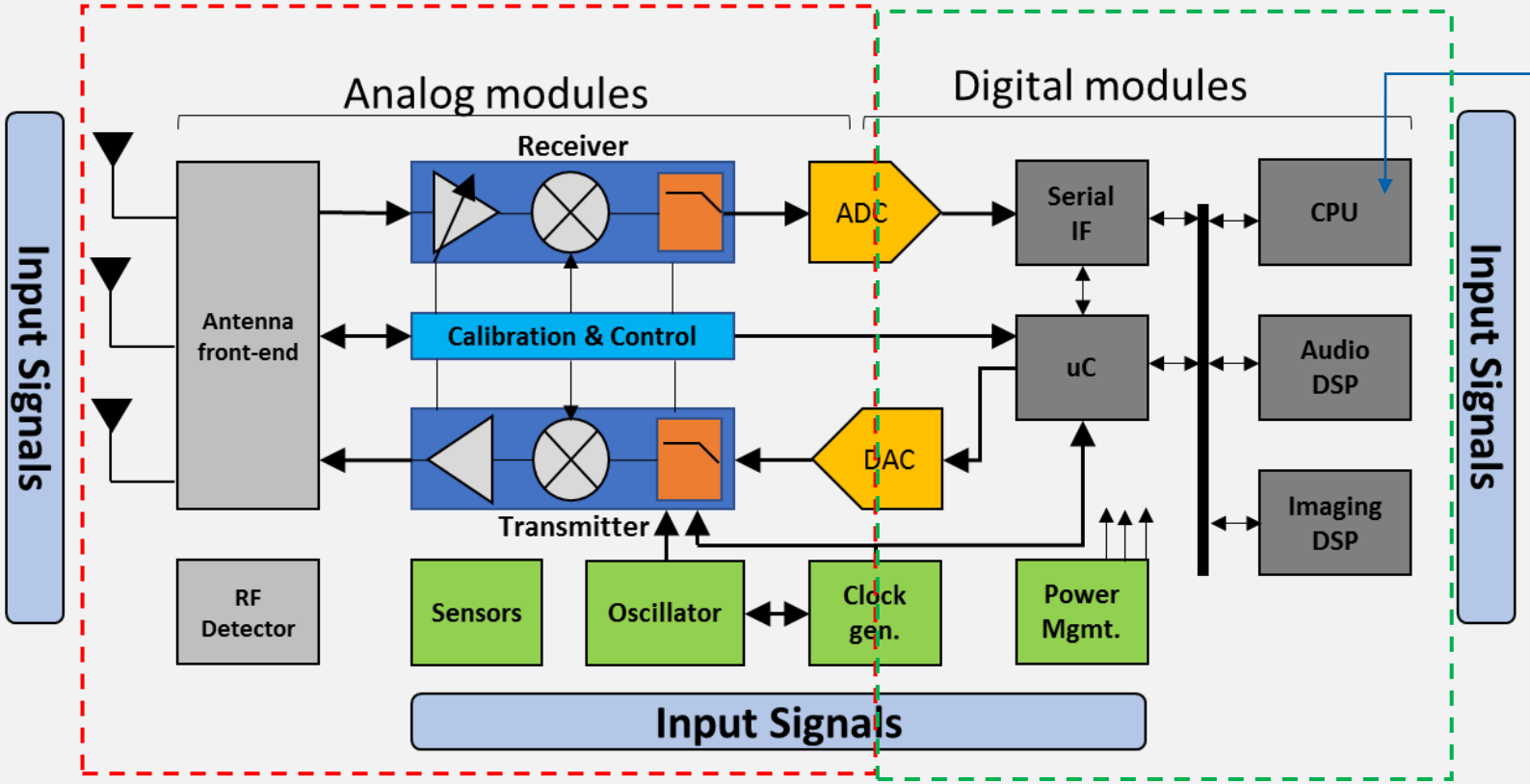
AUTOASSERT
Bundesministerium für Bildung und Forschung
16ME0117
COSEDA technologies

Agenda

- Motivation
- Assertions in SystemVerilog
- Challenges
- Assertions in SystemC
- Outlook

Complex Systems

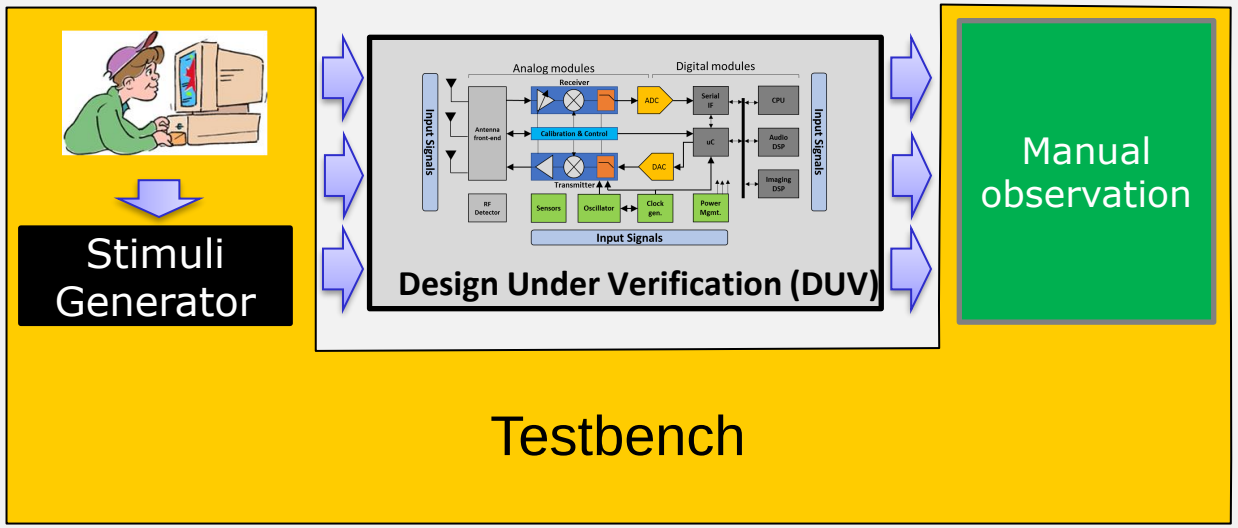
- Typical heterogeneous system



Functional correctness?

Verification Environment

- DUV: analog HW, digital HW + control SW
- Directed testing
- Stimuli generation
  - Manual

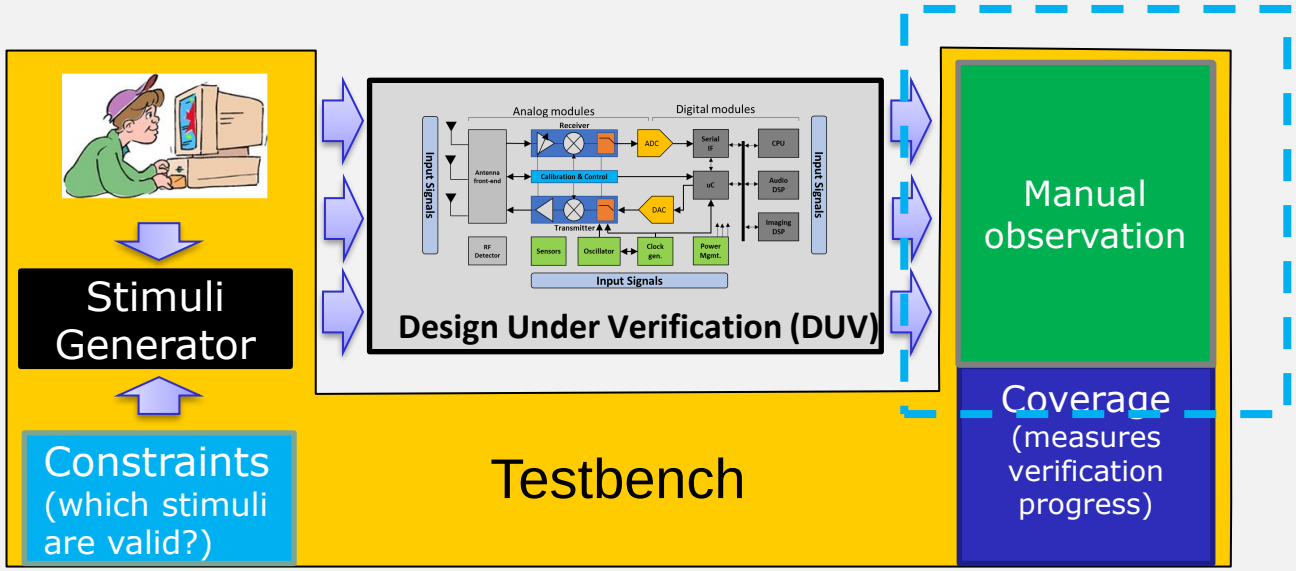


Not practical!

DUV too complex!!

Assertion Based Verification Environment

- DUV: analog HW, digital HW + control SW
- Stimuli generation
  - Manual
  - Constrained random
- Monitoring
  - Assertions
  - Coverage



What are assertions?

Assertions

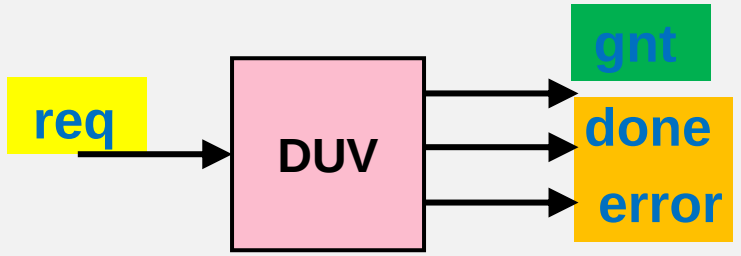
- Capture the intended behavior of DUV
  - Validate the behavior of a design — Is it working correctly?
- Also referred as properties — Executable comments
- Automate validation of the design against the specification.

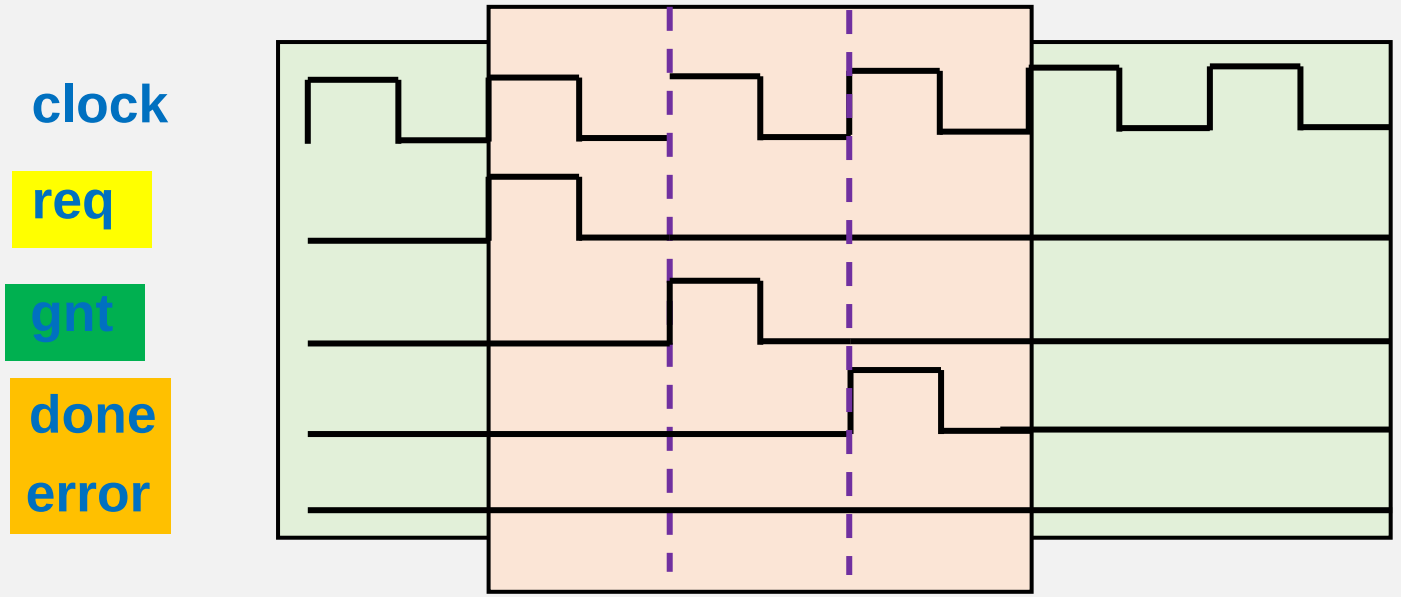
- Advantages
  - Support better testing and verification
  - Reusable
  - Portability
  - Make debugging easier

Assertions - Example

- Required behavior

  If **req** is high,
  then **gnt** should be high in next cycle
  AND **done** is high AND **error** is low
  one cycle later.

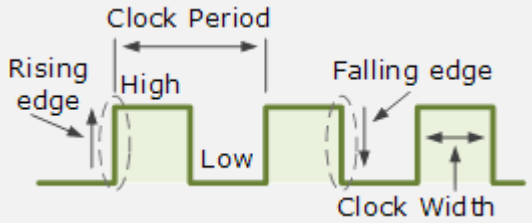


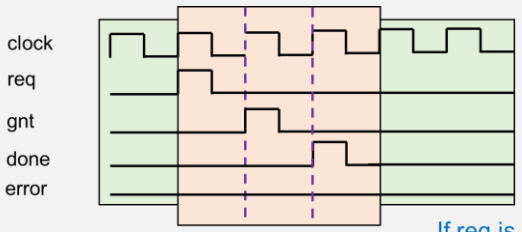


SVA Assertions – Example

SystemVerilog Assertions (SVA)

```
assert property (@posedge clock) req |-> gnt ##1 (done && !error)
```





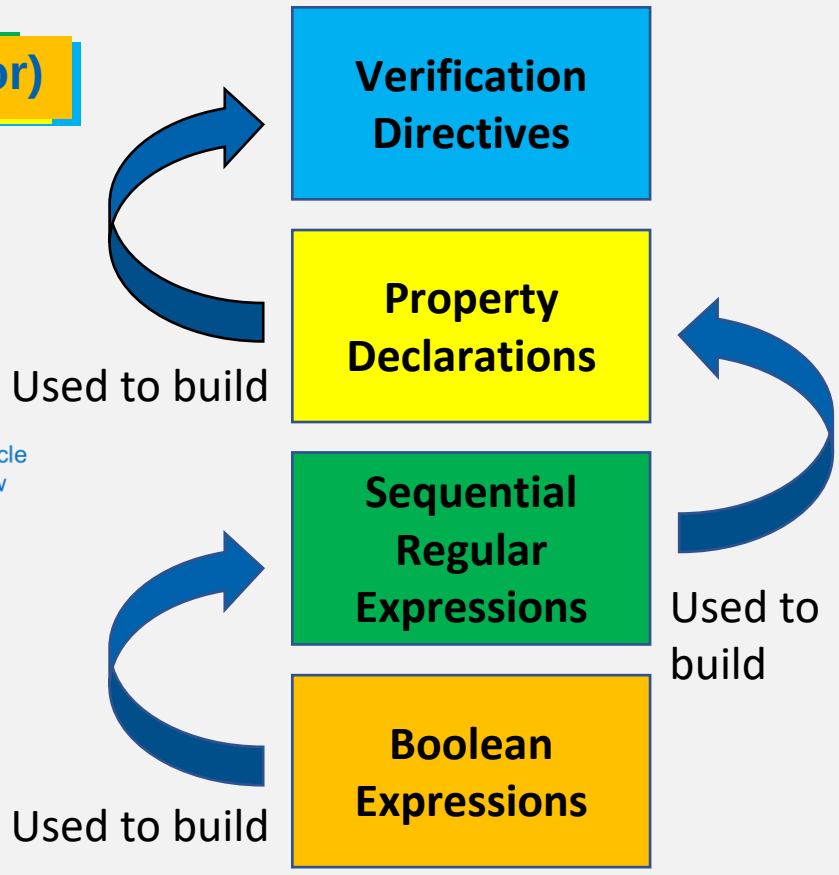
If req is high,
then gnt should be high in next cycle
AND done is high AND error is low
one cycle later.

Operators
Pre-conditions (antecedent)
Post-conditions (consequent)
Mainly for digital systems!

What about analog and digital systems controlled by SW?

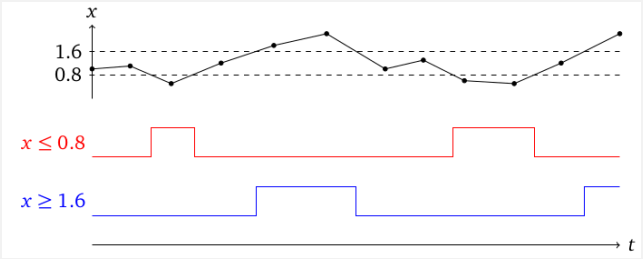
mixed assertions

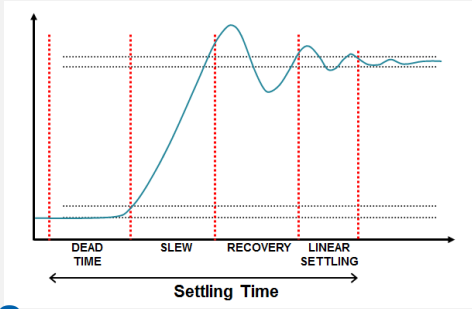
Building blocks

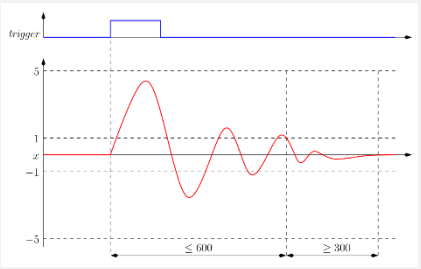


Challenges for Mixed Assertions (1)

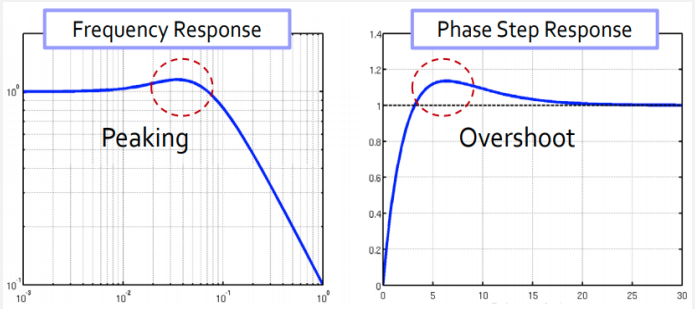
How to define time continuous properties?

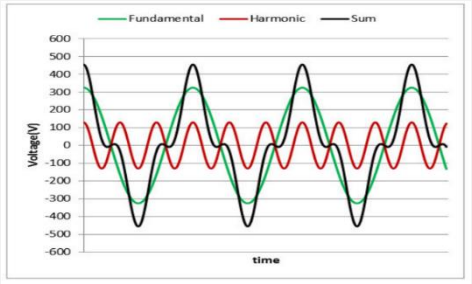


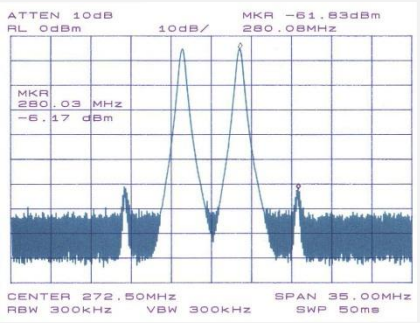




How to define frequency response properties?

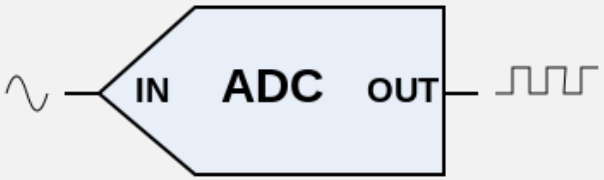


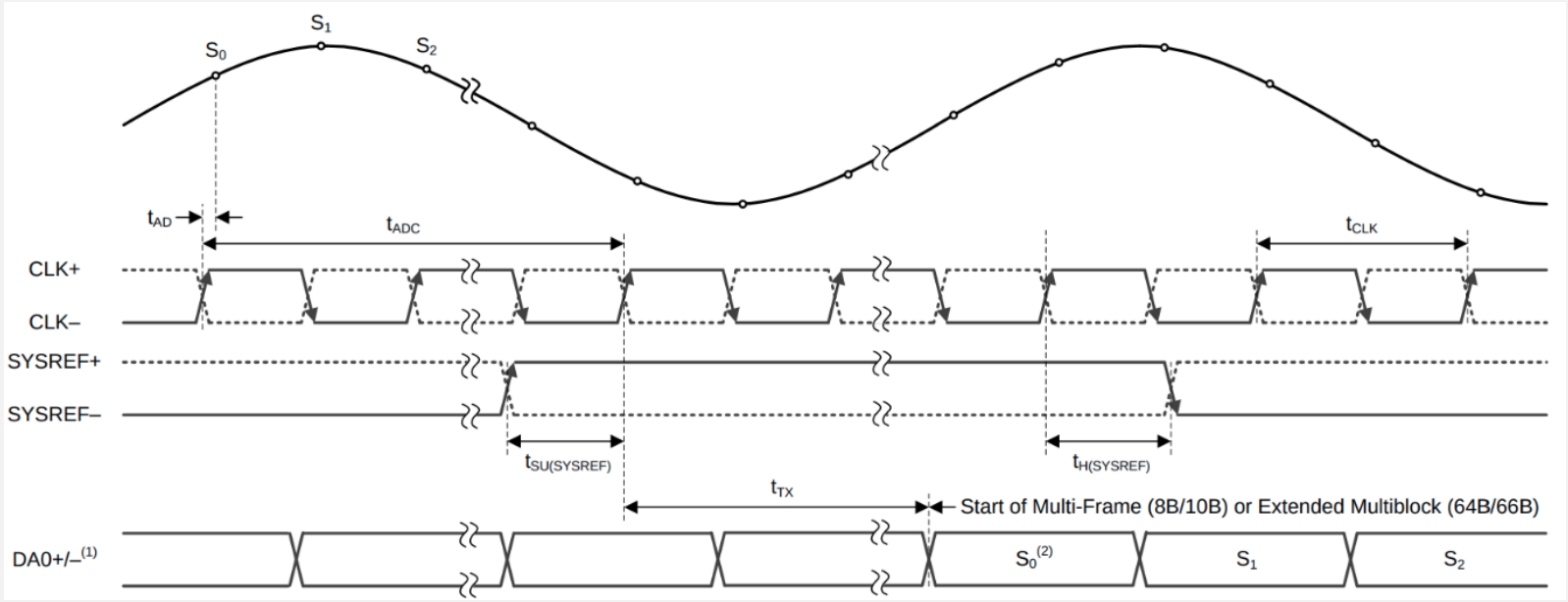




Challenges for Mixed Assertions (2)

Mixed-signal interface properties
E.g., Analog to Digital Converter

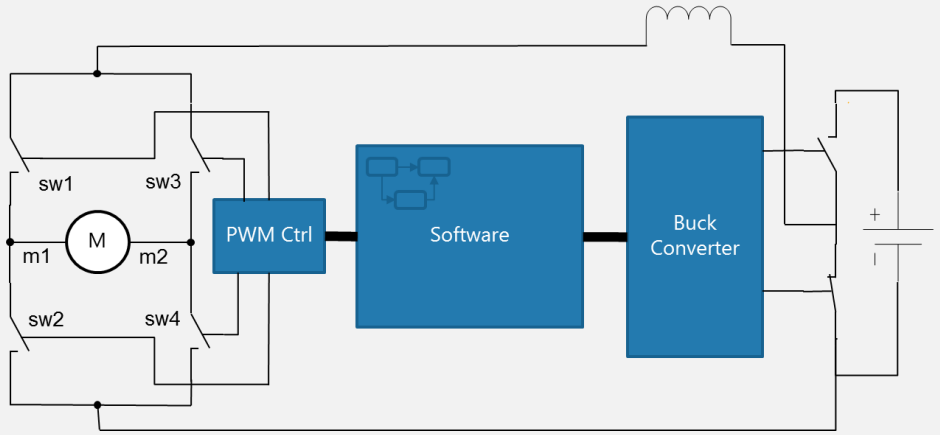




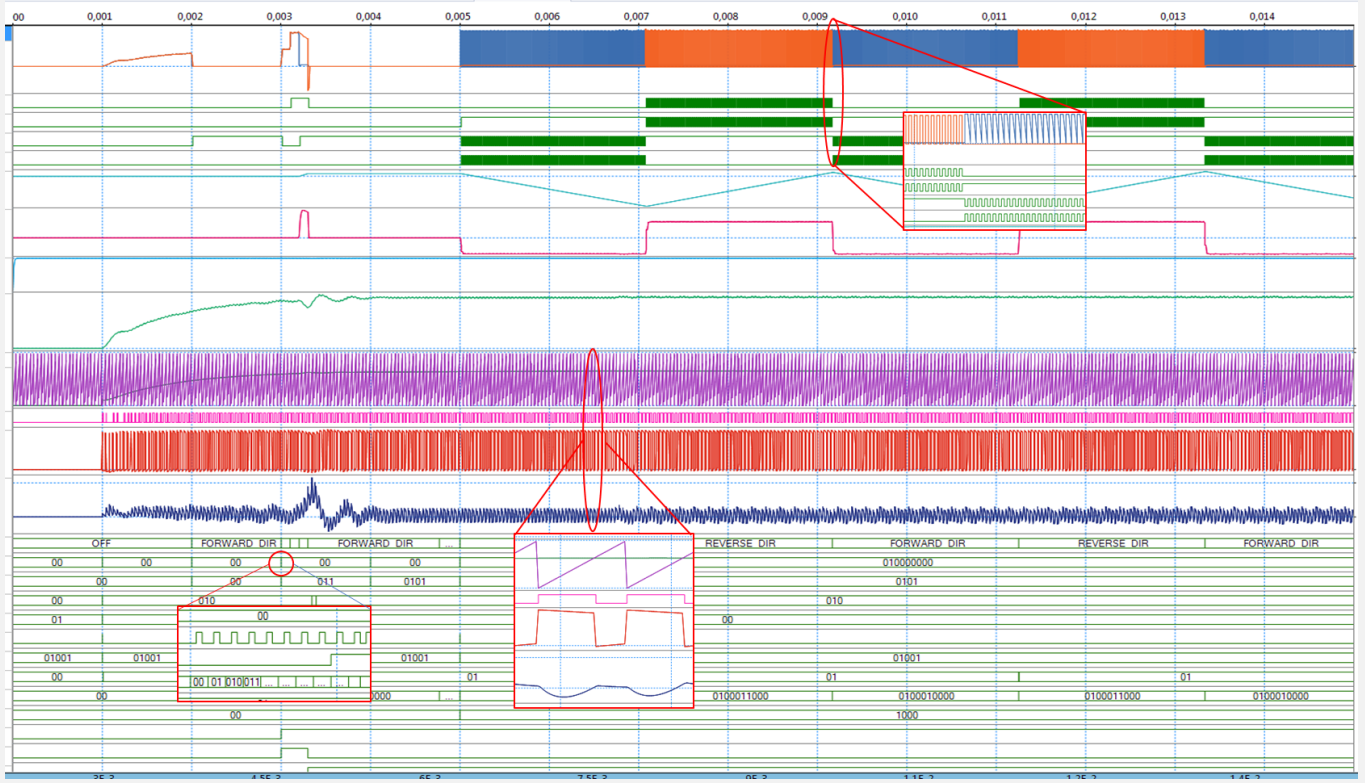
Challenges for Mixed Assertions (3)

Software + Mixed Signal Properties

DC-DC converter (Buck mode)



All together: mixed assertions



Mixed ABV

Requirements

a formal declarative language for expressing properties

a mechanism for checking that the DUV satisfies the properties

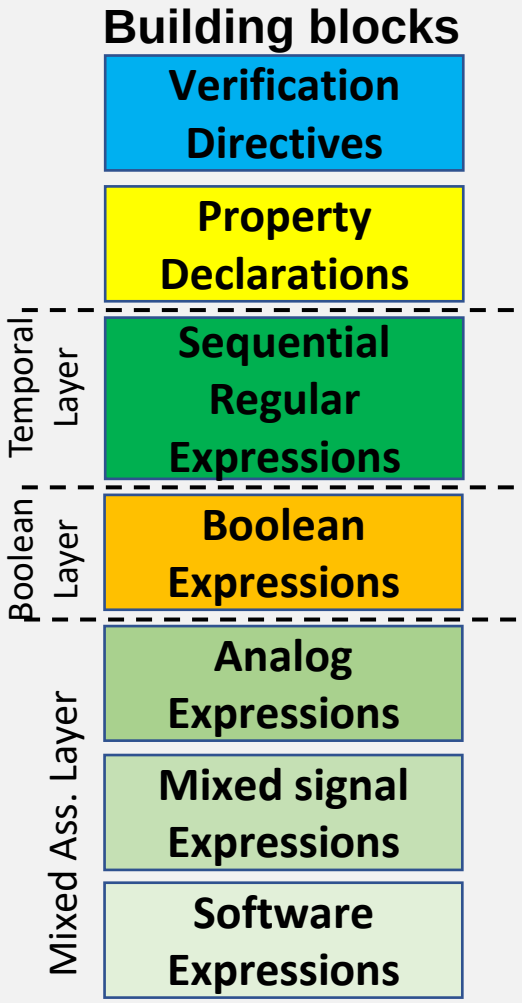
What we need

SystemC/AMS assertions language

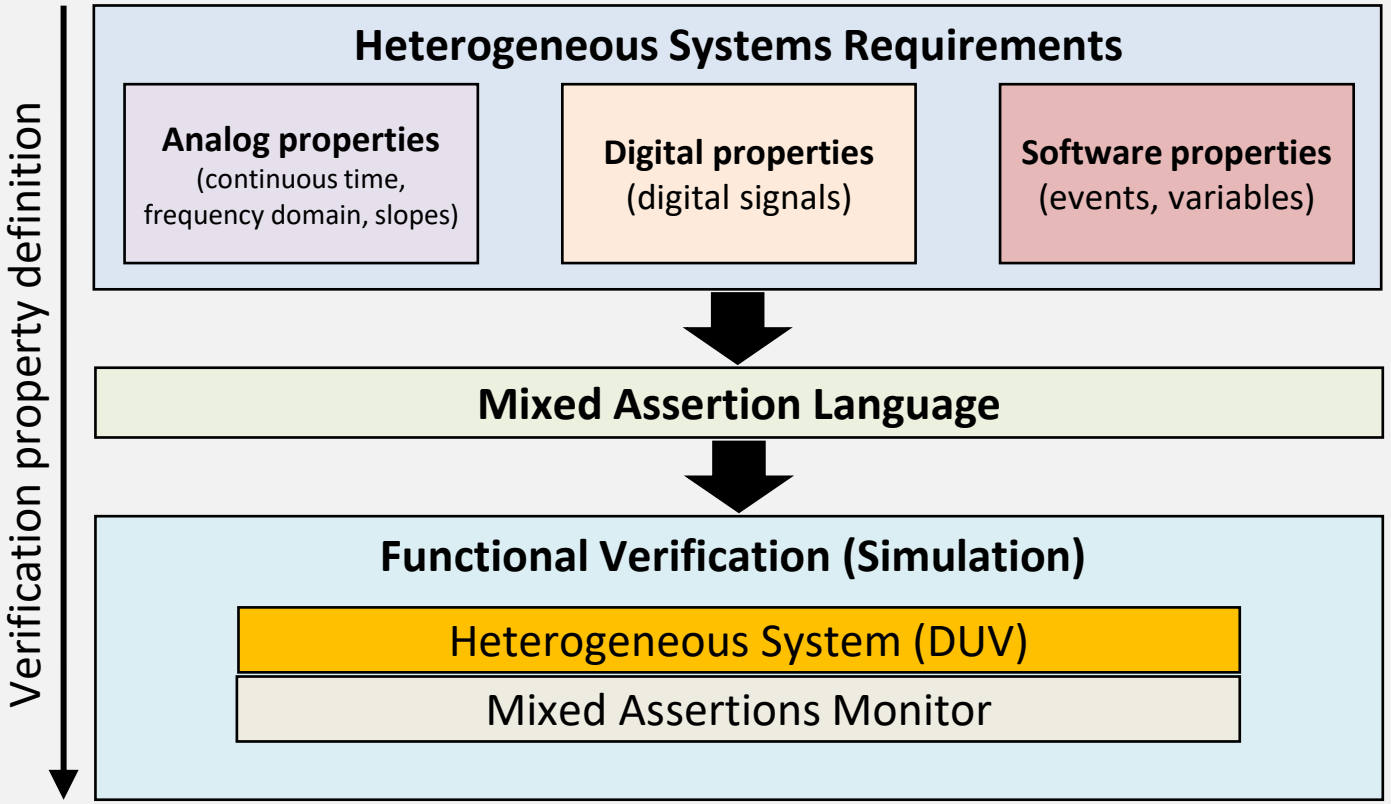
All the building blocks from digital

Mixed assertions layer

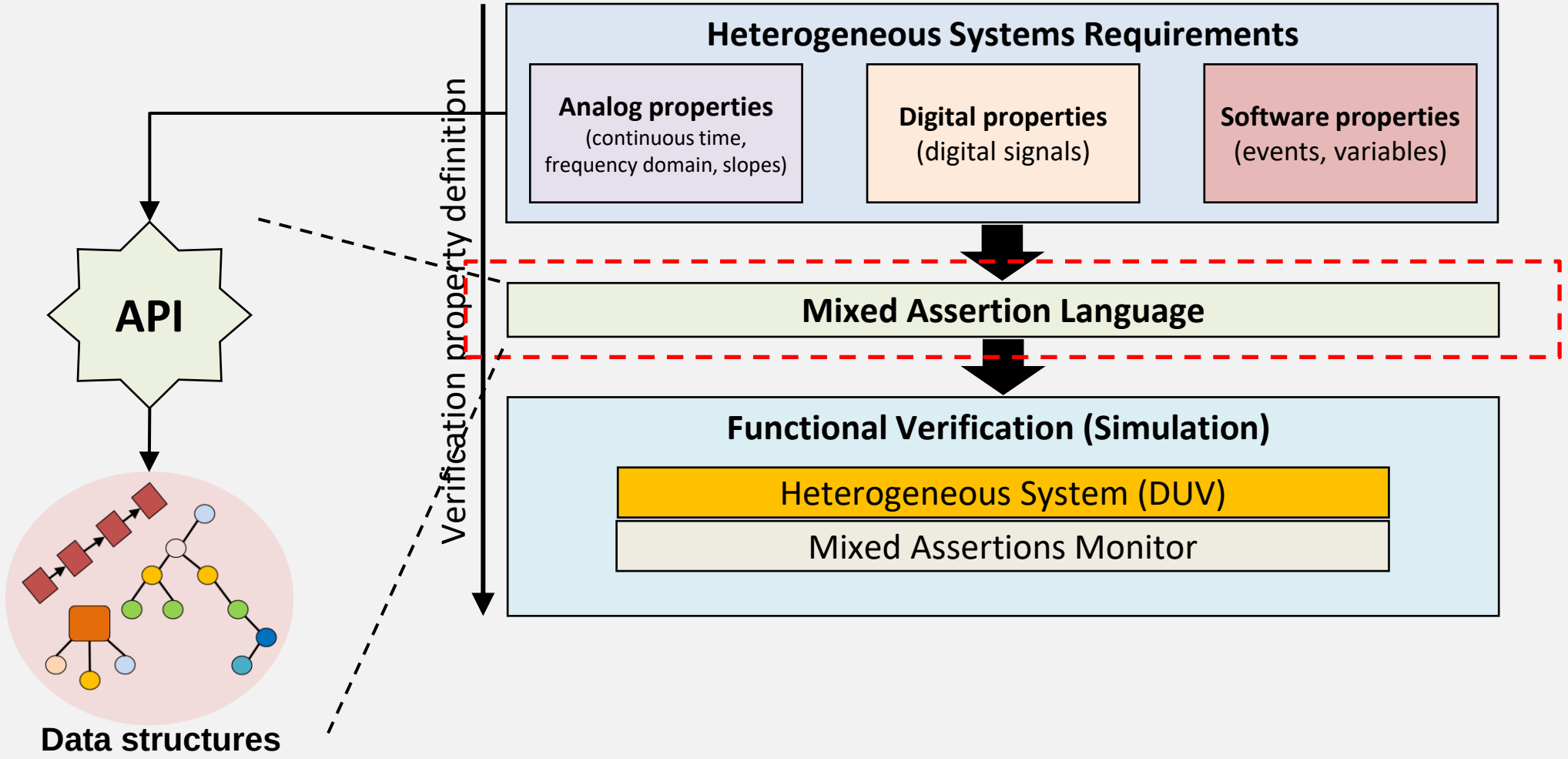
- Analog expressions
- Mixed signal expressions
- Software expressions



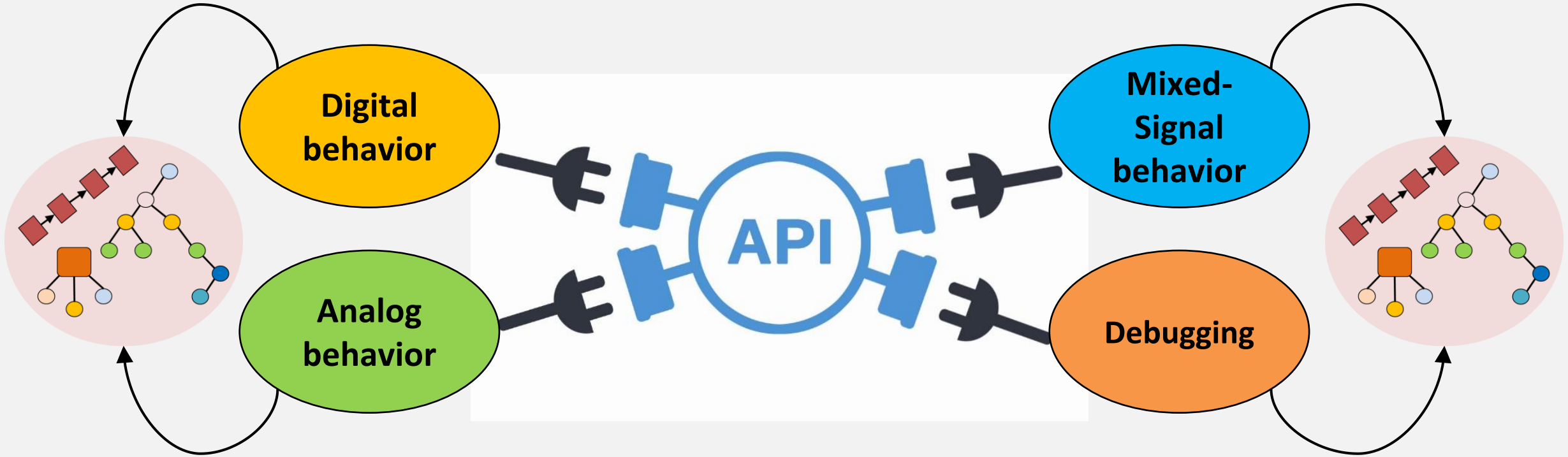
Mixed Assertions Based Flow



Mixed Assertions Based Flow - Language



Application Programming Interface (API)



Interface/ gateway to a software component
Defines how other components or systems can use it
Components/systems rely only on the API

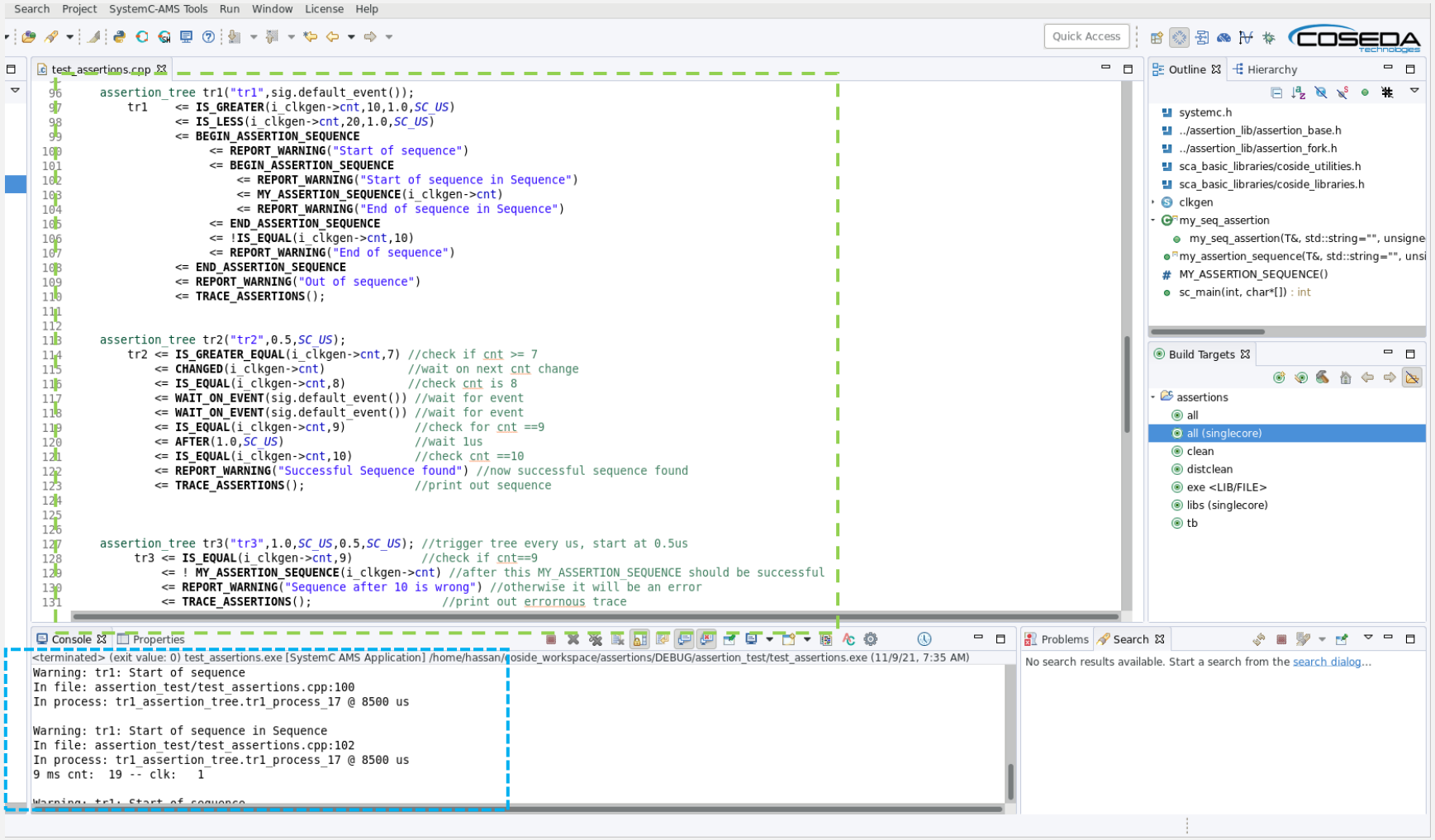
COSIDE Integration

German Research Center for Artificial Intelligence

Universität Bremen

Assertions

Assertions output



Summary and Outlook

- Assertions lib for SystemC/AMS not available
- Running prototype
- Syntax is work-in-progress
- Closely follow SystemVerilog Assertions
- Analog and mixed-signal interactions

SystemVerilog-like Assertions in SystemC

Muhammad Hassan
German Research Center for Artificial Intelligence (DFKI), Bremen, Germany
University of Bremen, Germany
muhammad.hassan@dfki.de

AUTOASSERT
Bundesministerium für Bildung und Forschung
16ME0117
COSEDA technologies