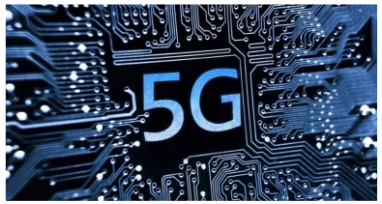
High-level Design and Verification in C++/SystemC with the COSIDE Environment

SystemC AMS & COSEDA User Group Meeting 2023

SIEMENS

New Markets Bring New Competitive Pressures

- Key markets have significant requirements for new designs



Reduced Time To Market with good QoR

Require FPGA Prototype & SoC/ASIC

Handle late changing specifications

Reduce Verification and Debug Cost/Time

Computer Vision & Neural Computing

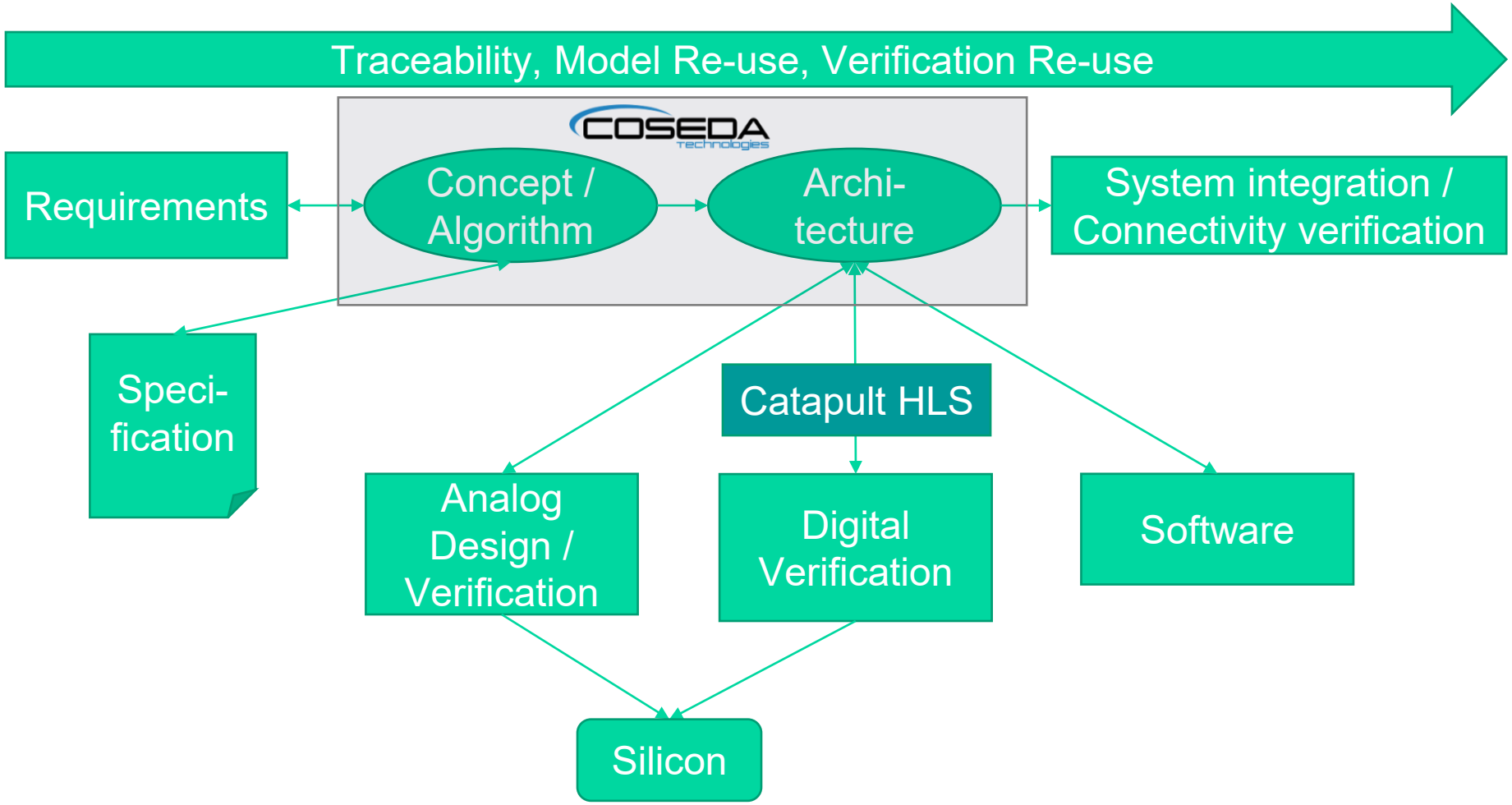
High Bandwidth & Cellular Communication

Image Processing, Video & Compression

Complex Algorithms

SIEMENS

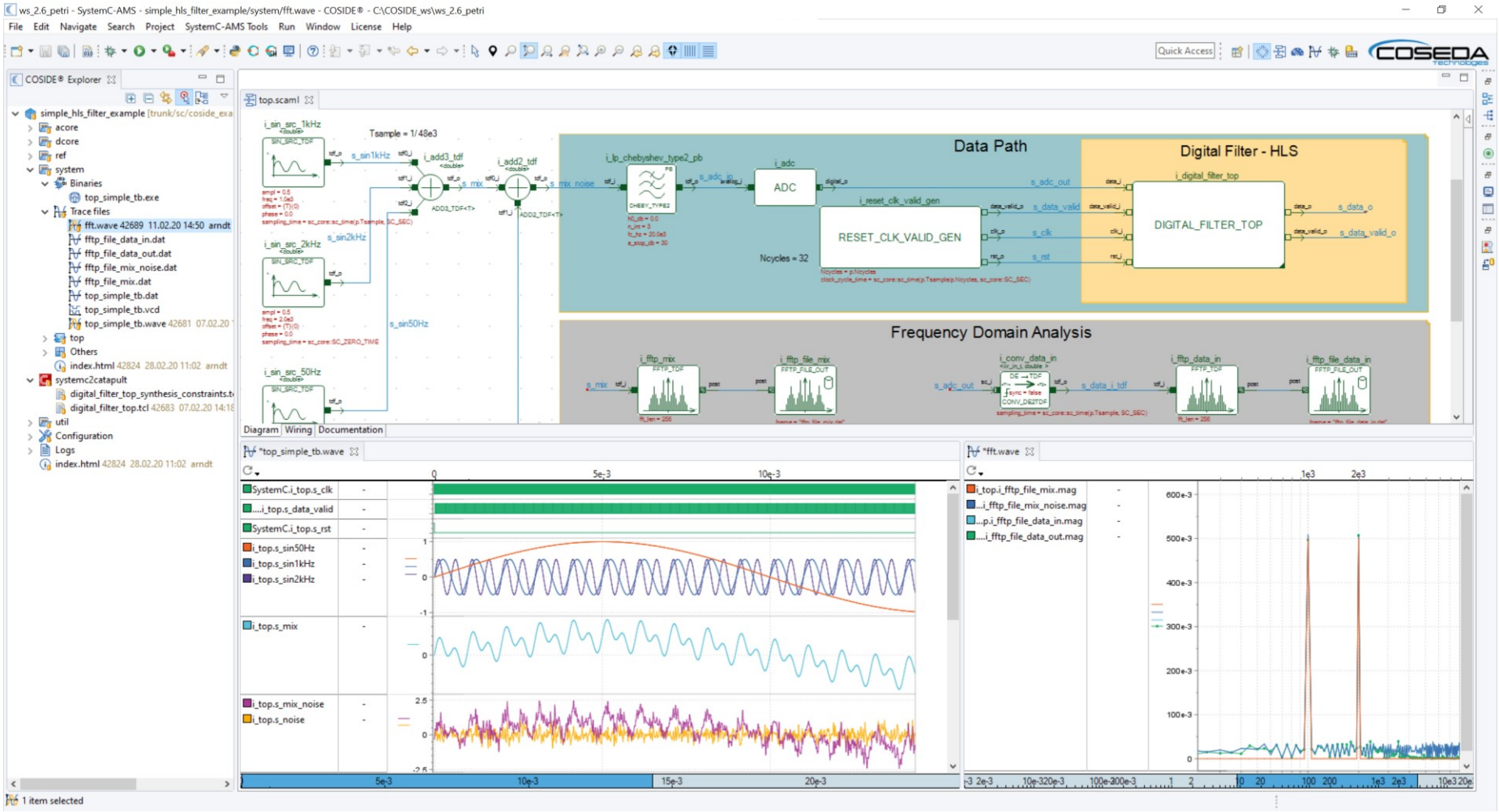
System-level Flow with COSIDE® and Catapult® HLS



SIEMENS

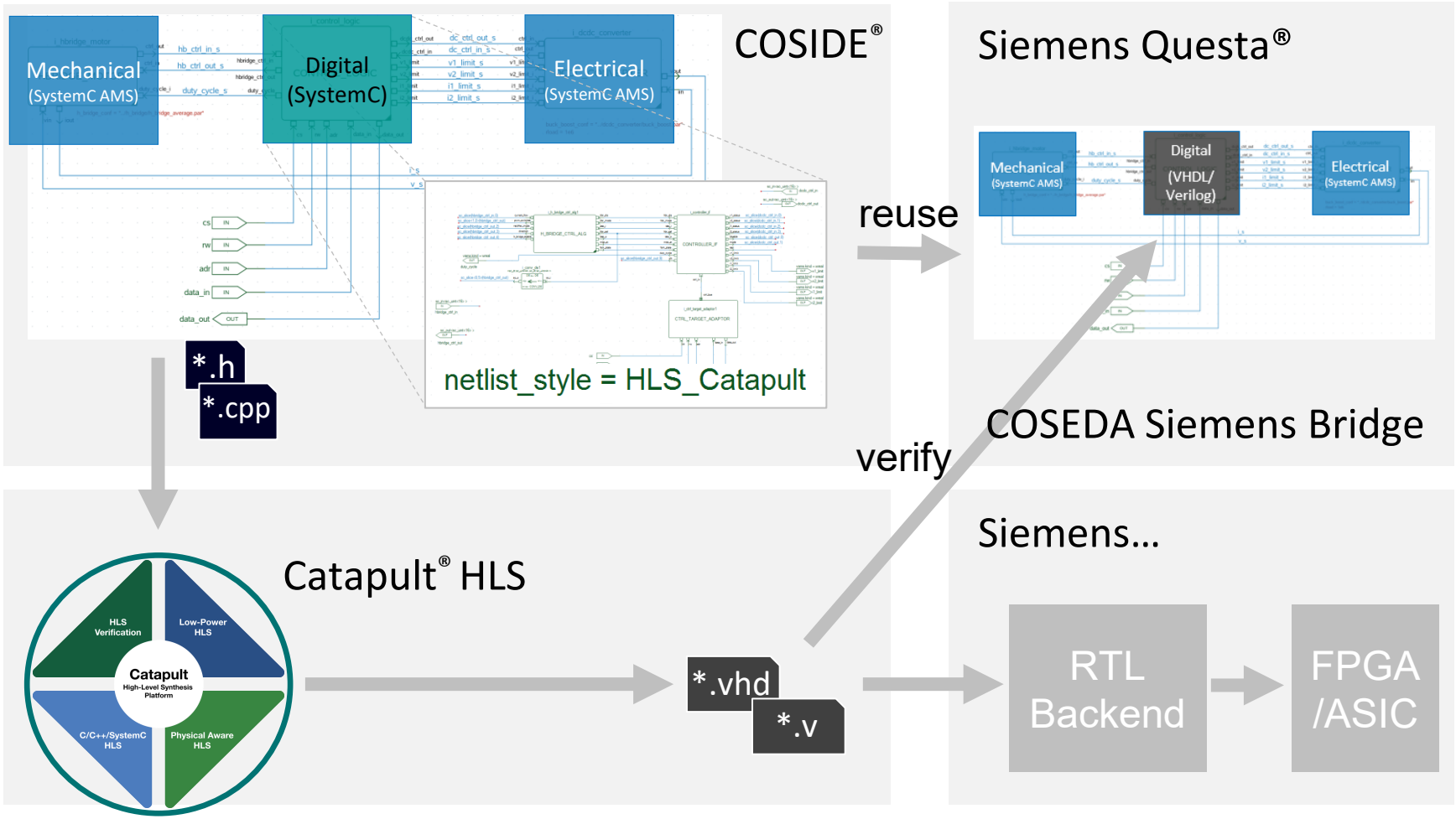
Catapult®/ COSIDE® HLS flow

Mixed signal system level modelling



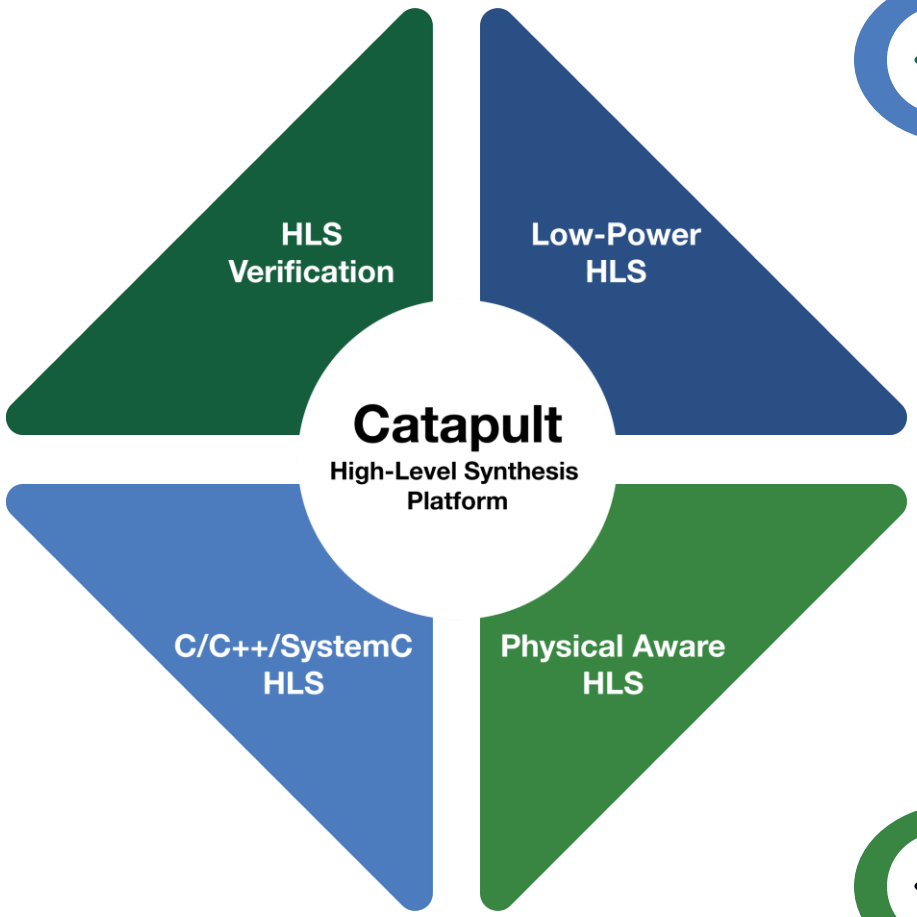
SIEMENS

COSIDE® HLS Flow Overview



SIEMENS

Introducing the Siemens Catapult HLS Platform



- High Quality RTL Synthesized from C++ and SystemC
- Comprehensive Verification for C++/SystemC and Flow into RTL
- PowerPro under-the-hood for Analysis and Best Power Optimized RTL
- Physically-Aware using RTL Synthesis/Floorplanning Engine

SIEMENS

Catapult High-Level Synthesis Benefits

Faster design

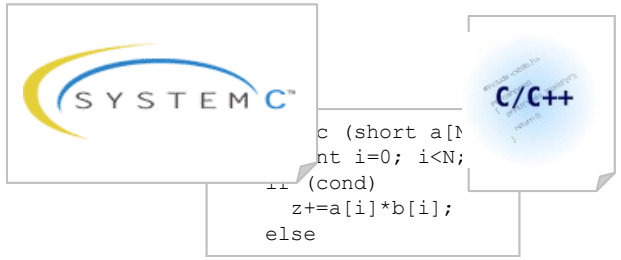
- Typically, RTL design phase is 2X faster for novice users, up to 10X for experienced users
- Project start to tape-out can be 4X faster

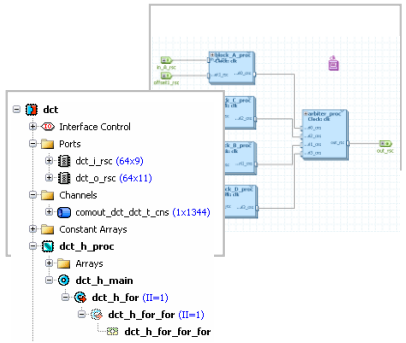
Faster verification

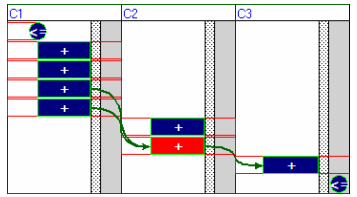
- Algorithm is verified at the abstract level
- Formal and dynamic verification can be used to prove equivalence between C++ and HDL

Easy technology retargeting, retiming

- RTL can be mapped to new technology library or clock frequency by re-synthesizing
- Simple transition between FPGA and ASIC implementation







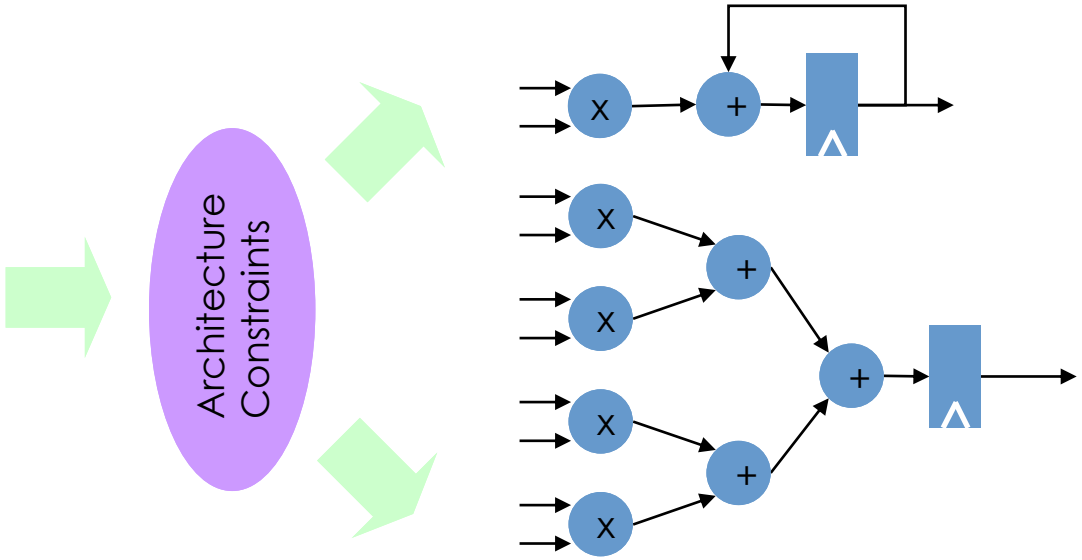
SIEMENS

Micro-Architecture Control

- User control over the micro-architecture implementation
  - Parallelism, Throughput, Area, Latency (loop unrolling & pipelining)
  - Memories (DPRAM/SPRAM/split/bank) vs Registers (Resource allocation)

- Exploration is accomplished by applying constraints
  - Not by changing the source code

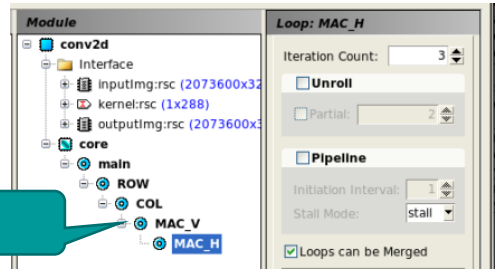
```
int mac(
  char data[N],
  char coef[N]
) {
  int accum=0;
  for (int i=0; i<N; i++)
    accum += data[i] * coef[i];
  return accum;
}
```



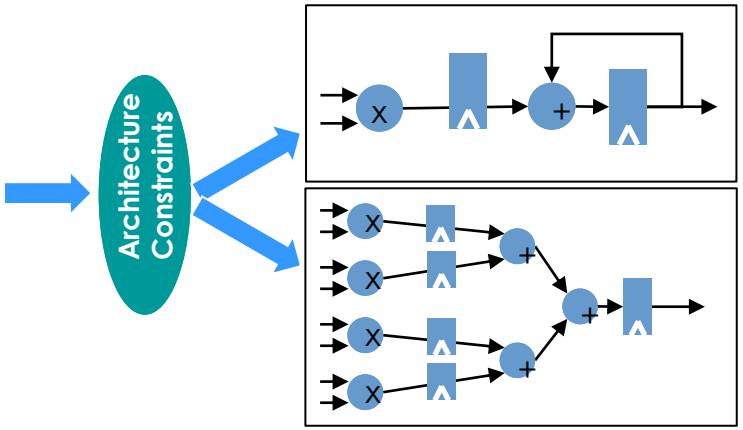
Typical Hand-coded RTL Optimizations Are Done Automatically by HLS

Loop optimizations

- Unrolling
- Pipelining
- Automatic merging

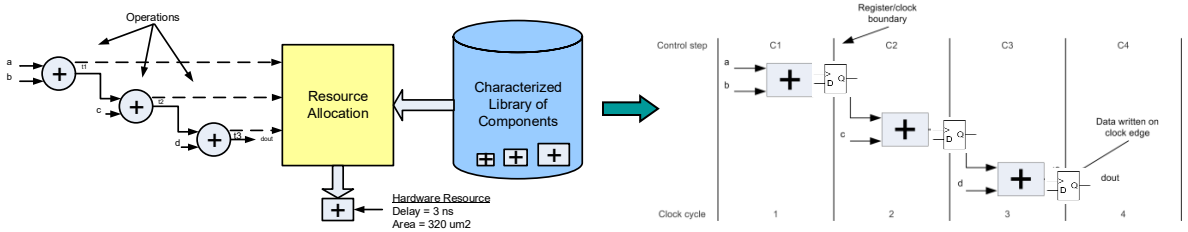



```
for (int i=0;i<4;i++) {
   acc += data_in[i] * coef_in[i] ;
}
```



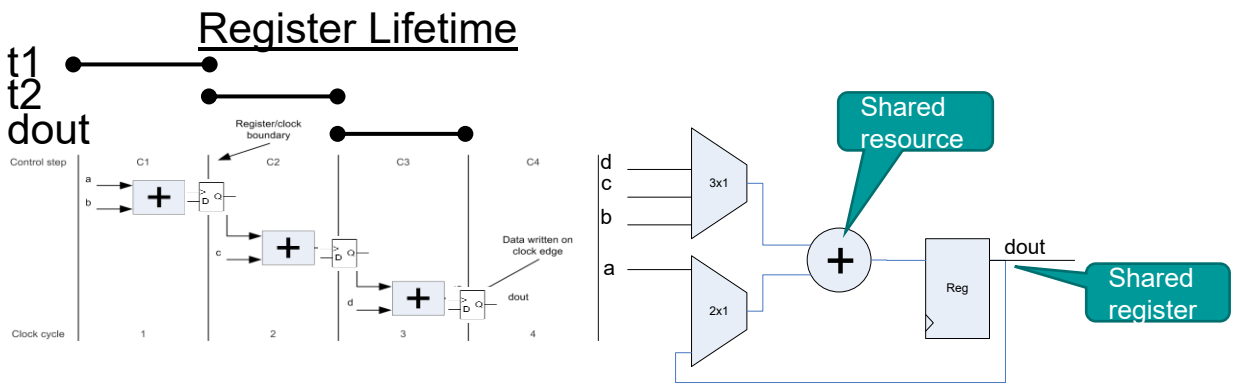
Scheduling

- Automatic timing closure based on target technology



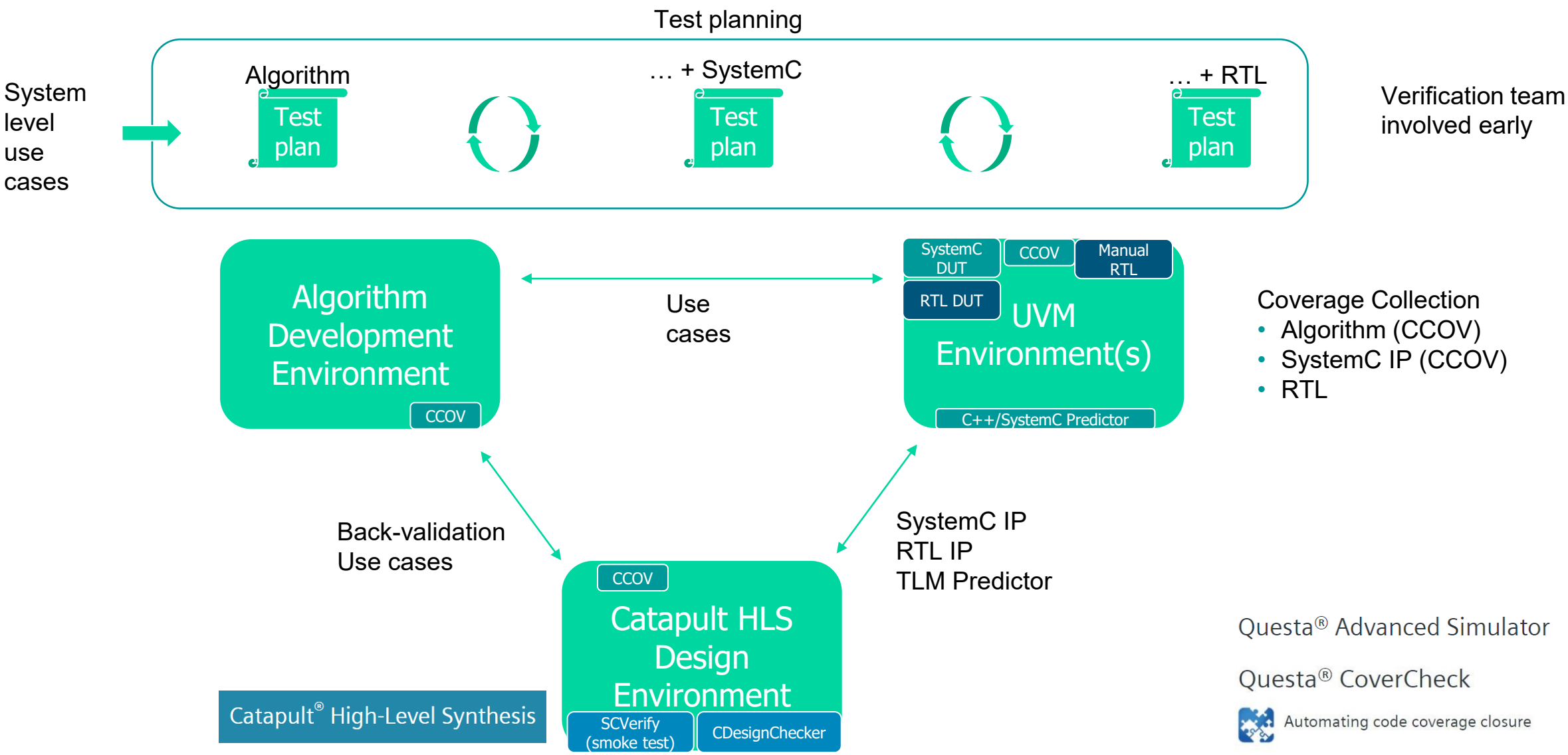
Register and Resource sharing

- Automatic lifetime and mutual exclusivity analysis and optimization



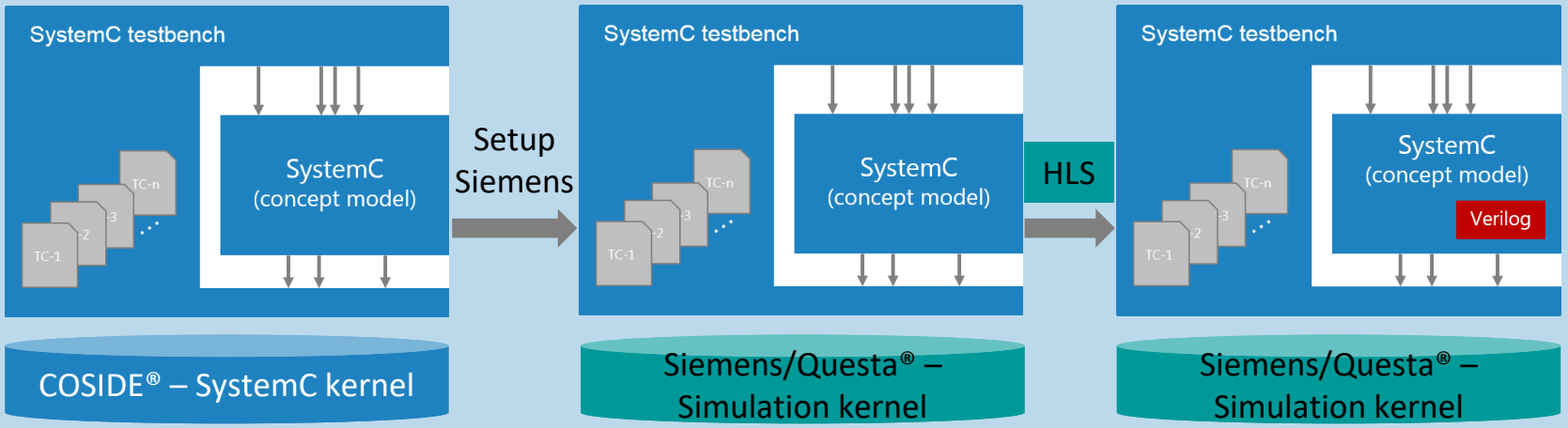
SIEMENS

High-Level Verification Flow Overview



SIEMENS

Reusable Verification Environment with Analog Functionality



- Reuse of existing system level test bench, reuseable verification environment with analog functionality
- Simulation runs immediatelly, since only simulation kernel changed
- Model hierarchy kept

SIEMENS

Catapult HLV Brings Known & Trusted Methods to HLS Verification

- Checks for Synthesizable code
  - Automatic Formal Property checking of HLS source
  - Static “lint” and formal methods to ensure correctness

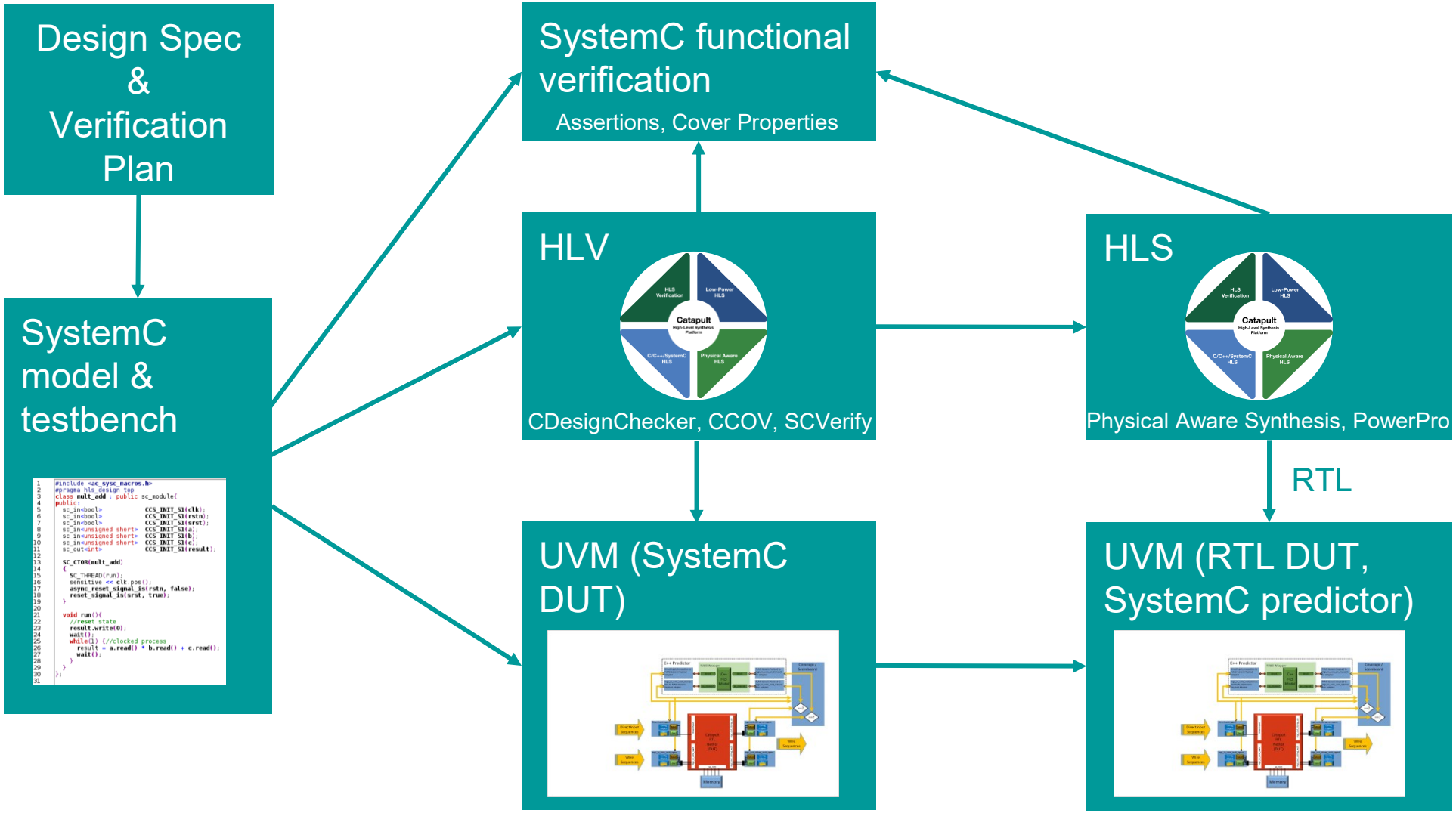
- Functional Correctness of the C++/SystemC
  - Does it work?
  - Metrics driven verification coverage and formal analysis

- Confidence that the RTL is correct
  - RTL co-simulation
  - RTL Code Coverage using the same metrics
  - Formal apps including equivalence for design assurance

- Verification & Integration
  - Verification and Debug aids for post-HLS RTL integration
  - Visibility and Cross-probing to HLS source

SIEMENS

Catapult Verification Infrastructure



SIEMENS

Key benefits of Catapult High-Level Verification

All the benefits of C++ simulation for functional verification. Enable faster verification starting earlier

Metrics-driven verification of the HLS design source with several orders of magnitude in speed

Use known techniques to implement a trusted verification methodology

Make system integration testing & debug as easy as possible

Thank You

Published by Siemens EDA

Dr. Werner Bachhuber
Senior Field Application Engineer
DI SW EDAGS TSS CLD EU
Arnulfstraße 201
80634 München
Germany

Phone +49 (0) 89 5709 6375

Mobile +49 (0) 172 8545995

E-mail werner.bachhuber@siemens.com

SIEMENS