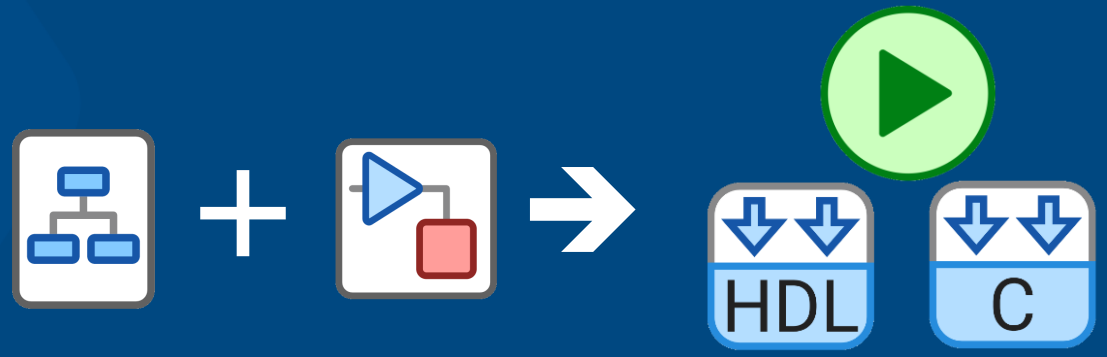
FPGA Meets Systems Engineering

Integrating Approaches for Space Applications



Stephan van Beek

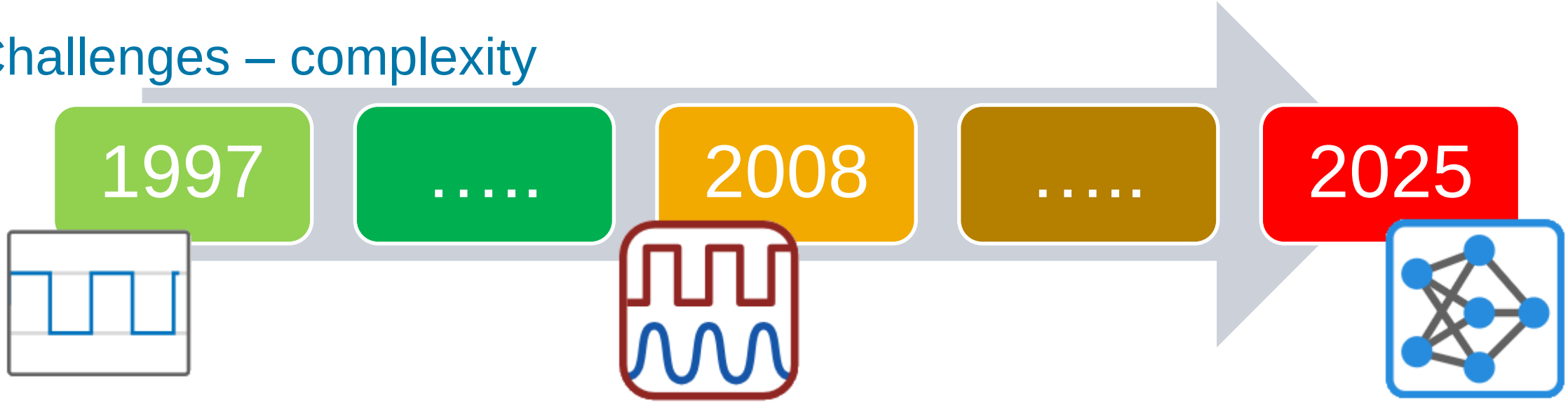
European Technical Specialist
SoC/FPGA Design Flows

The biggest problem was not the unit mismatch itself, but the failure to detect and correct this mistake

Mars Climate Orbiter

Image credit: NASA/JPL

Challenges – complexity



1 FPGA engineer,
1 board with 4 FPGAs

- Applications: logic interfaces, very simple algorithms
- *Xilinx XC4000E resources* (**DSPs 0**, Logic Cells 85K, BRAM)
- Schematic entry ➔ hybrid schematic entry + VHDL

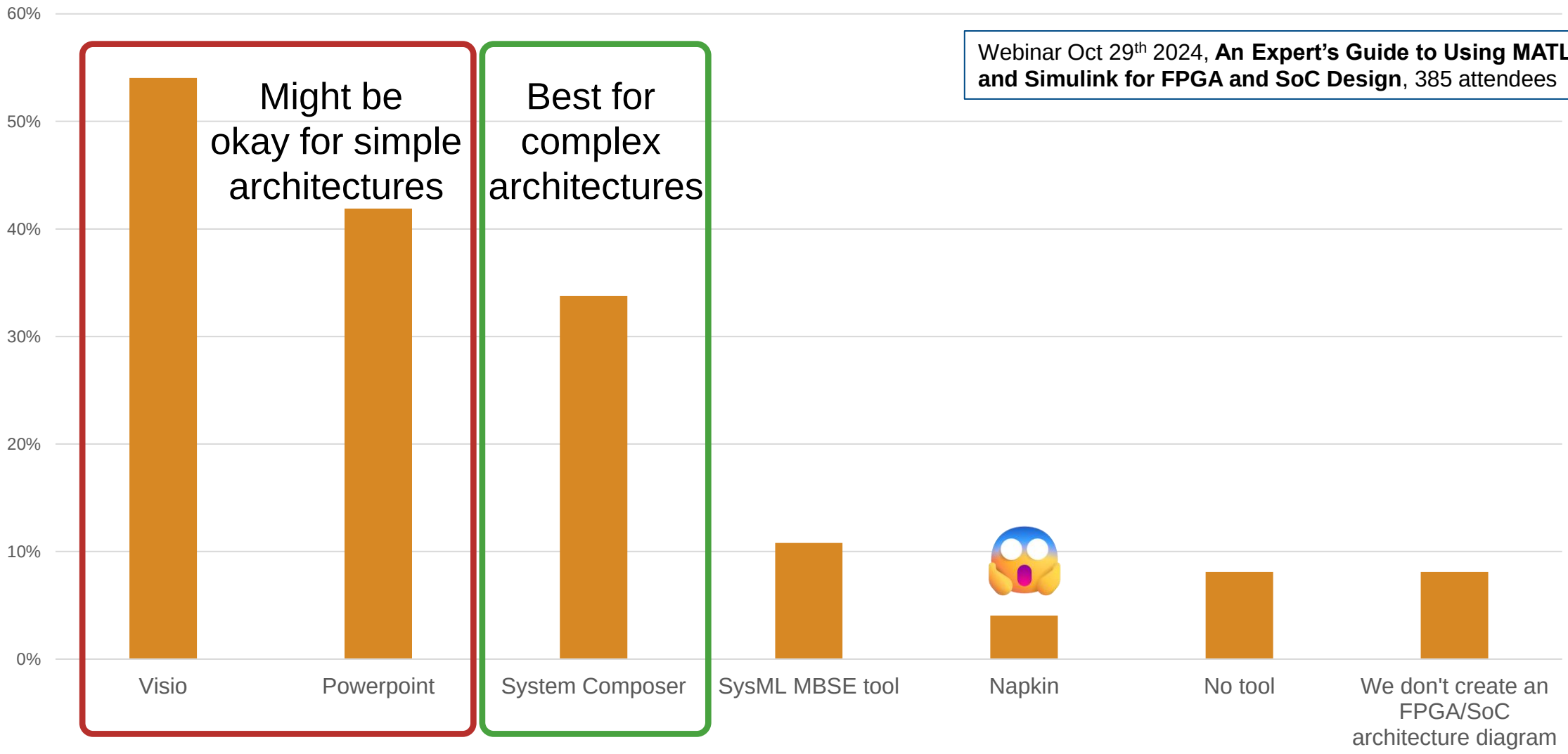
Multiple FPGA engineers,
1 board with 1 FPGA

- Applications: more complex algorithms
- *Xilinx Virtex-5 resources* (**DSPs 1K**, Logic Cells 330K, BRAM, **Embedded Processor**)
- VHDL ➔ MBD (FPGA)

Team of engineers (algorithm, architect, FPGA, software) for 1 board with 1 SoC

- Applications: very complex algorithms (artificial intelligence, wireless, signal processing, vision, motor/power, etc.)
- *AMD Versal resources* (**Logic Cells 9M**, **DSP engines 11K, AI engines 400,** BRAM, URAM, **ARM**)
- Will this be the next evolution? ➔ MBD (SoC/FPGA) + MBSE??

How do you create your FPGA architectural block diagrams?

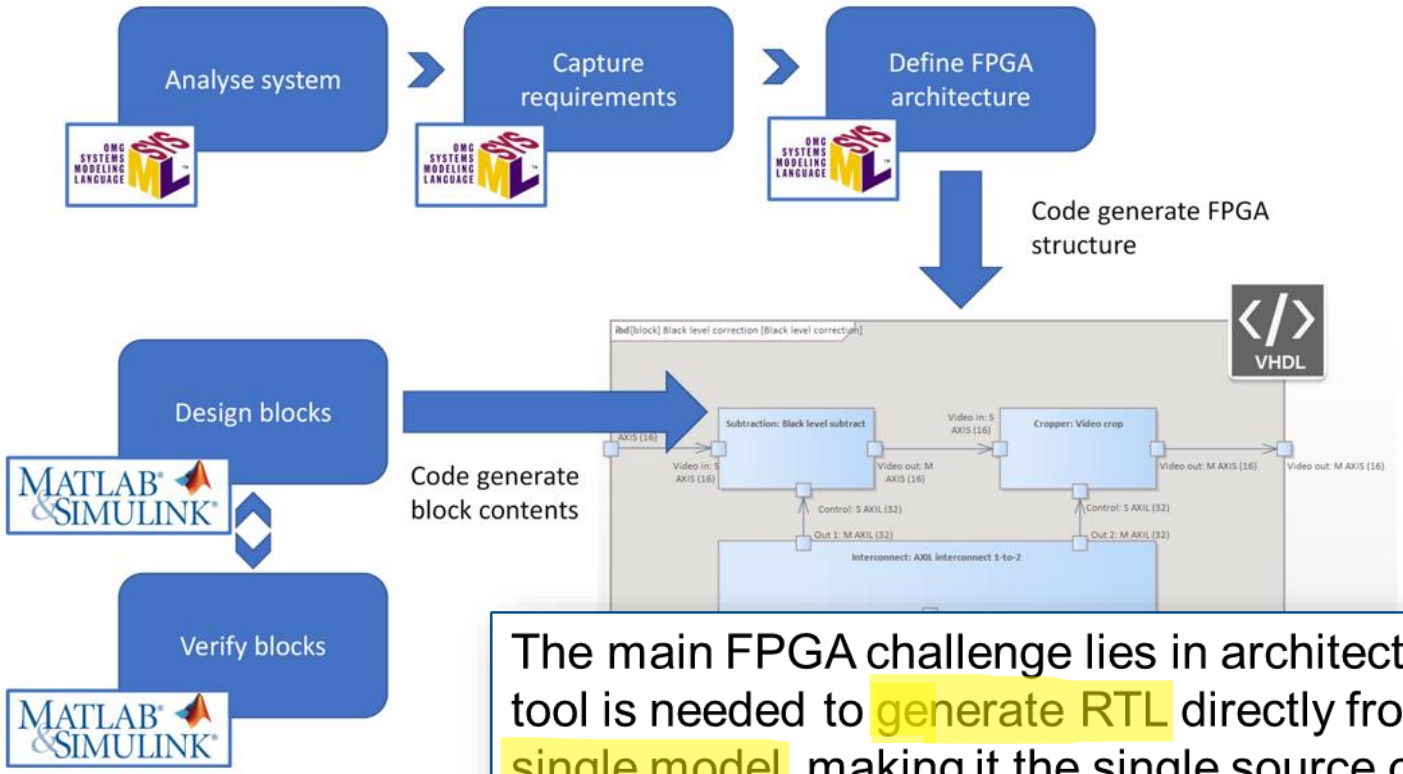


Webinar Oct 29th 2024, **An Expert's Guide to Using MATLAB and Simulink for FPGA and SoC Design**, 385 attendees

Adam Taylor is on a good path here ……

ADIUVO ENGINEERING AND TRAINING, LTD.

Adiuvo Model Based Flow



The main FPGA challenge lies in architecture, often informal and overlooked. A tool is needed to generate RTL directly from the architecture + design in a single model, making it the single source of truth.
- Adam Taylor, Adiuvo

Challenges – complexity ➔ needs

- Top-down design processes
  - Functional decomposition
  - No simulation needed early, but ….. later you will need simulation
- Go Beyond Textual Requirements
  - Use expressiveness of requirement models and trade studies
  - Use views to put stakeholder discussions in context
- Validate compliance to requirements through simulation
- Deployment
  - Generate RTL code from **architecture + design models**

Model-Based Systems Engineering + Model-Based Design

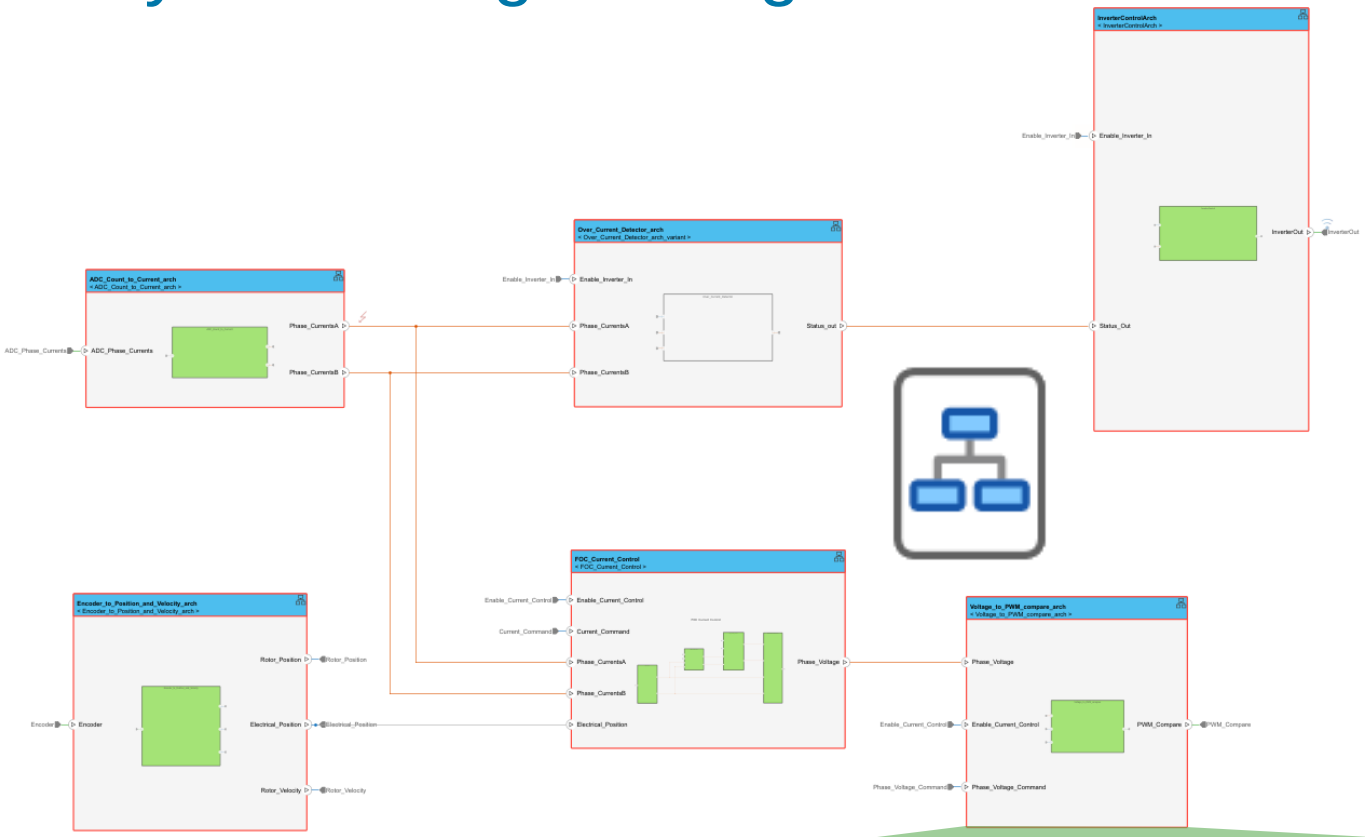
Top-down

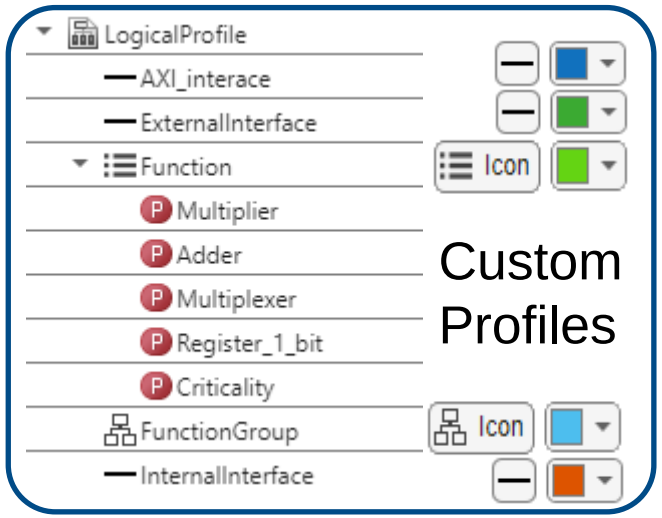
Requirements

Architecture

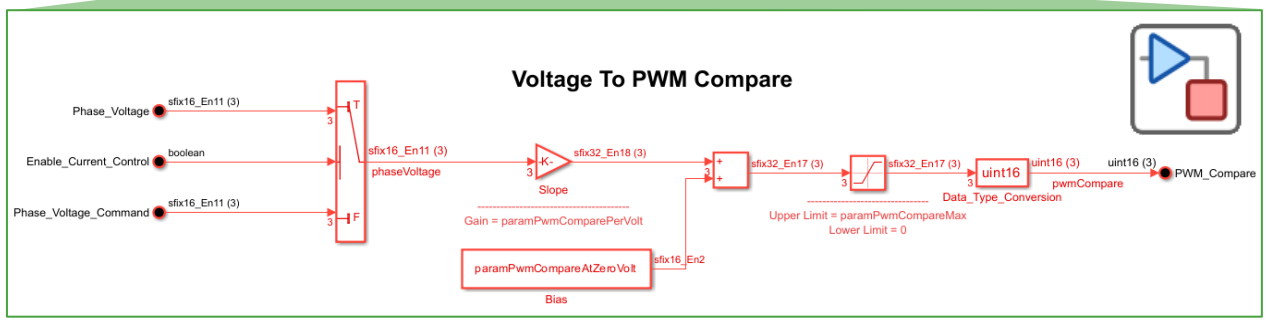
Bottom-up

Design

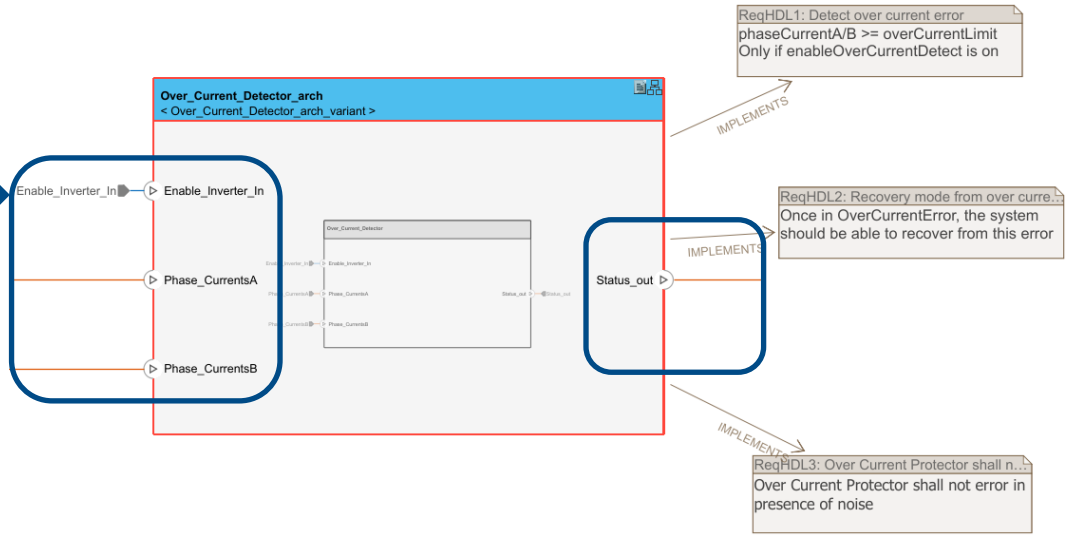




Integrate MBD with MBSE

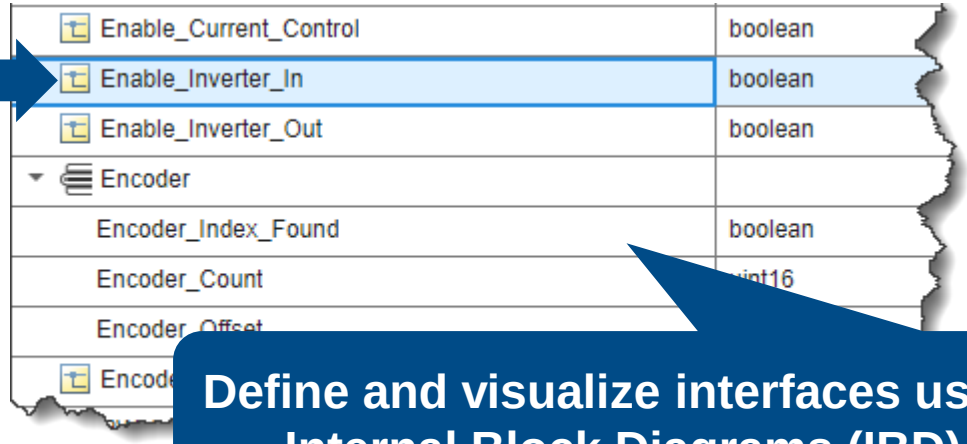


Requirements: Textual, Traceability, Interfaces, Interactions

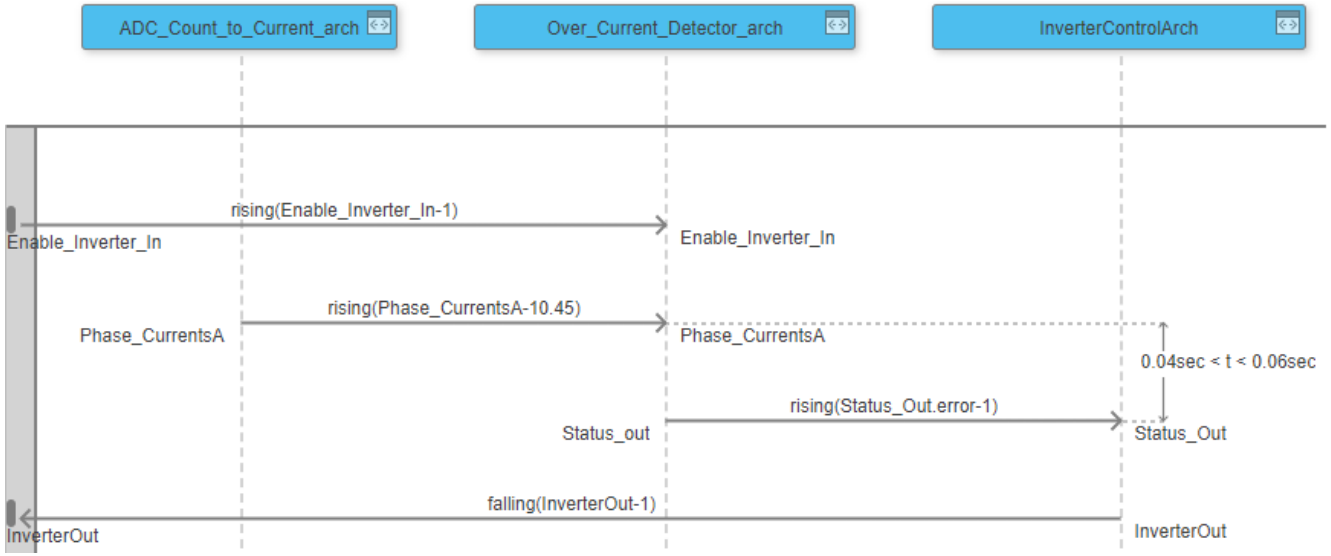


Establish traceability between architecture & design and textual requirements

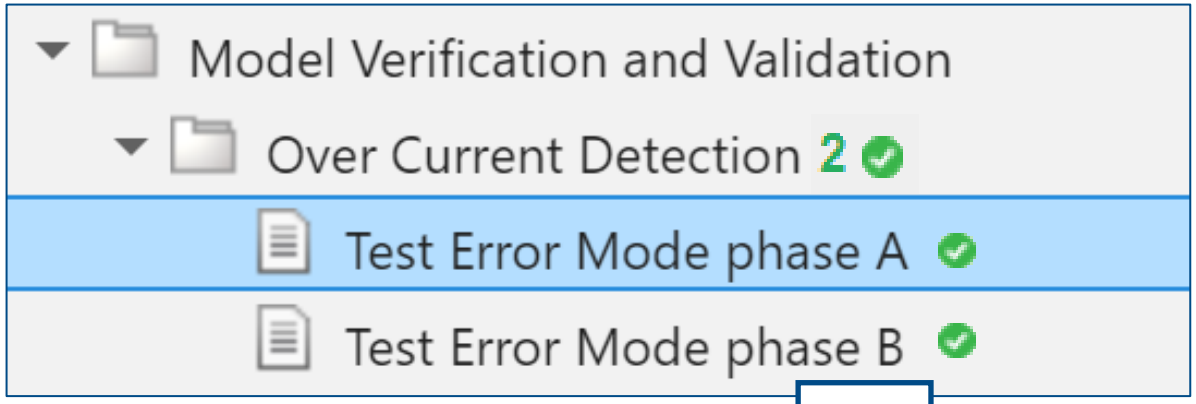
Define interface behaviors using Sequence Diagrams (SD)

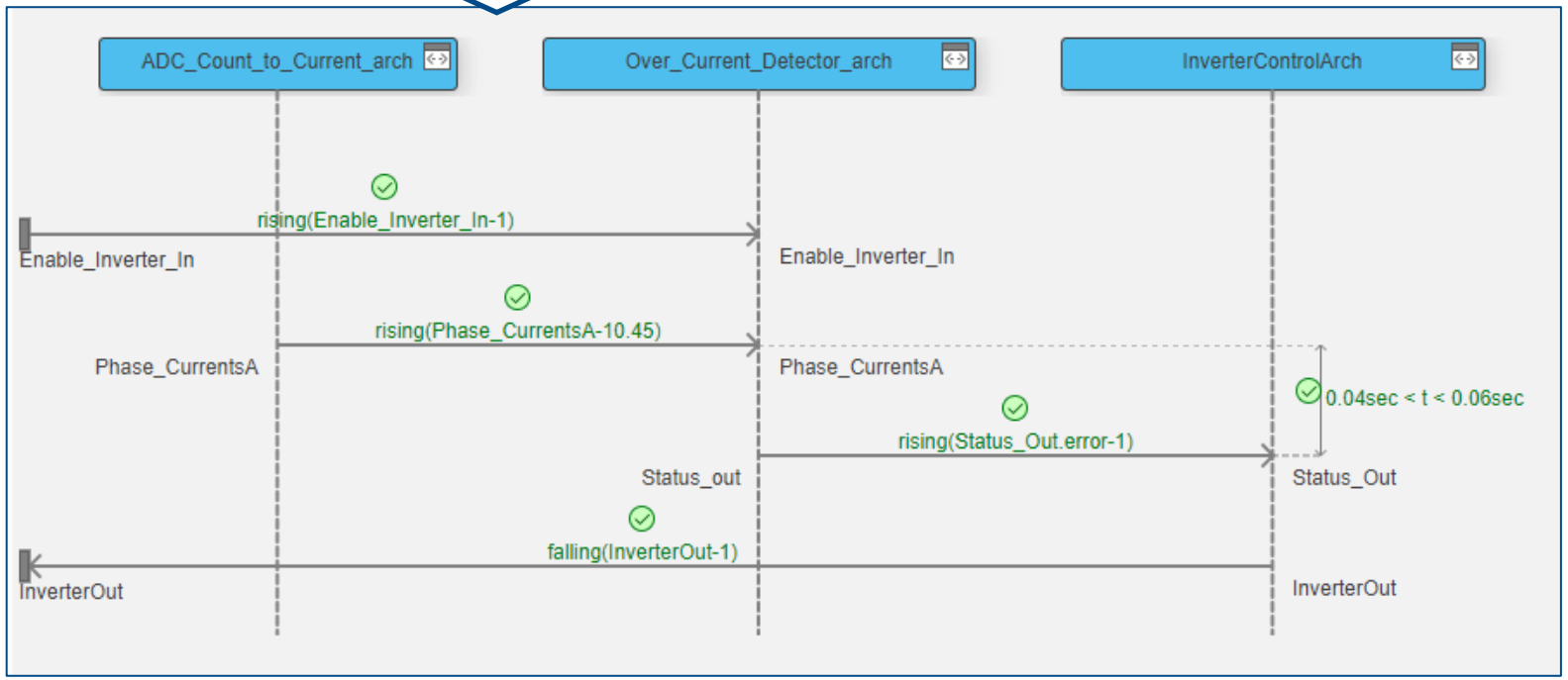


Define and visualize interfaces using Internal Block Diagrams (IBD)

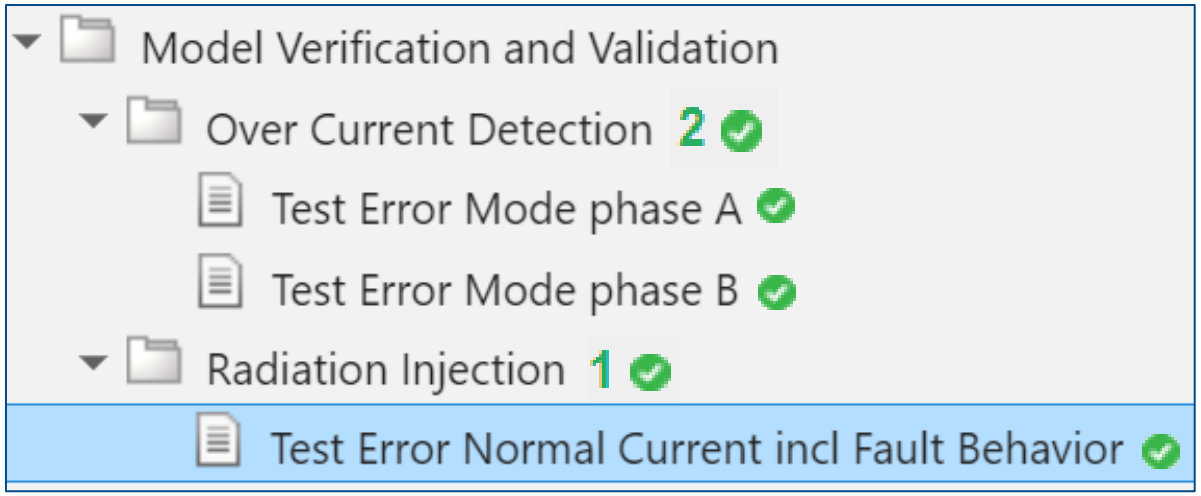


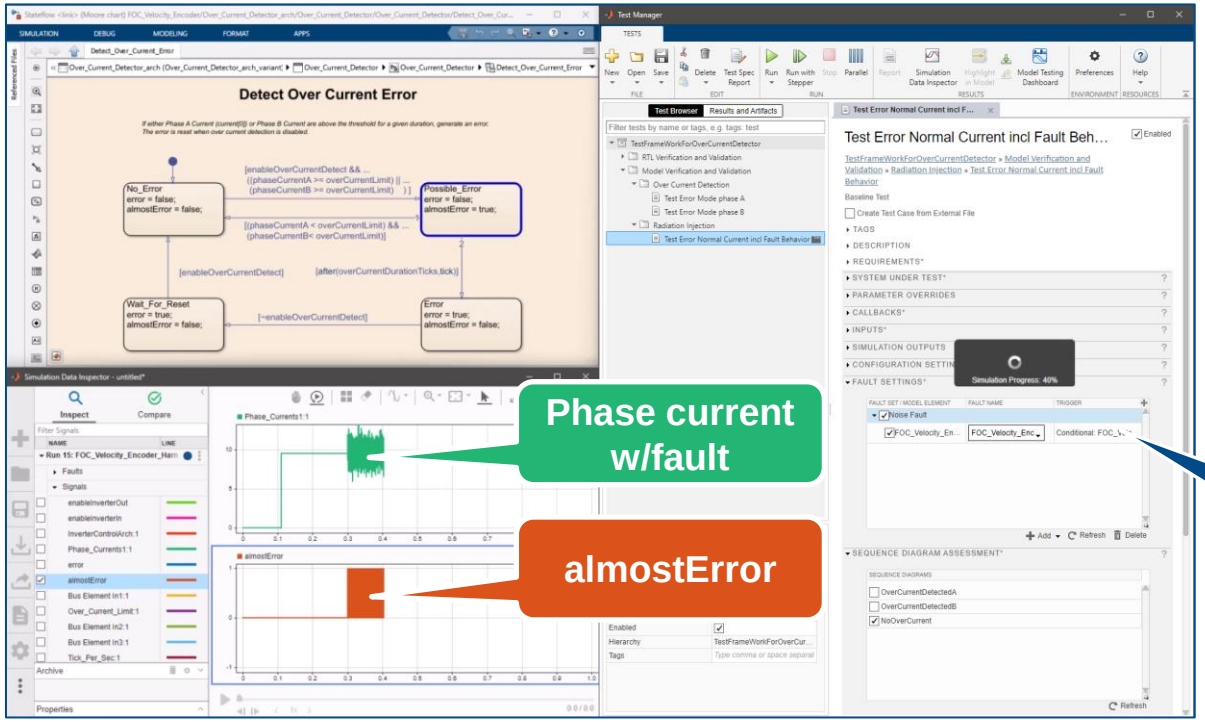
Validate Systems by Re-Using Requirements Models

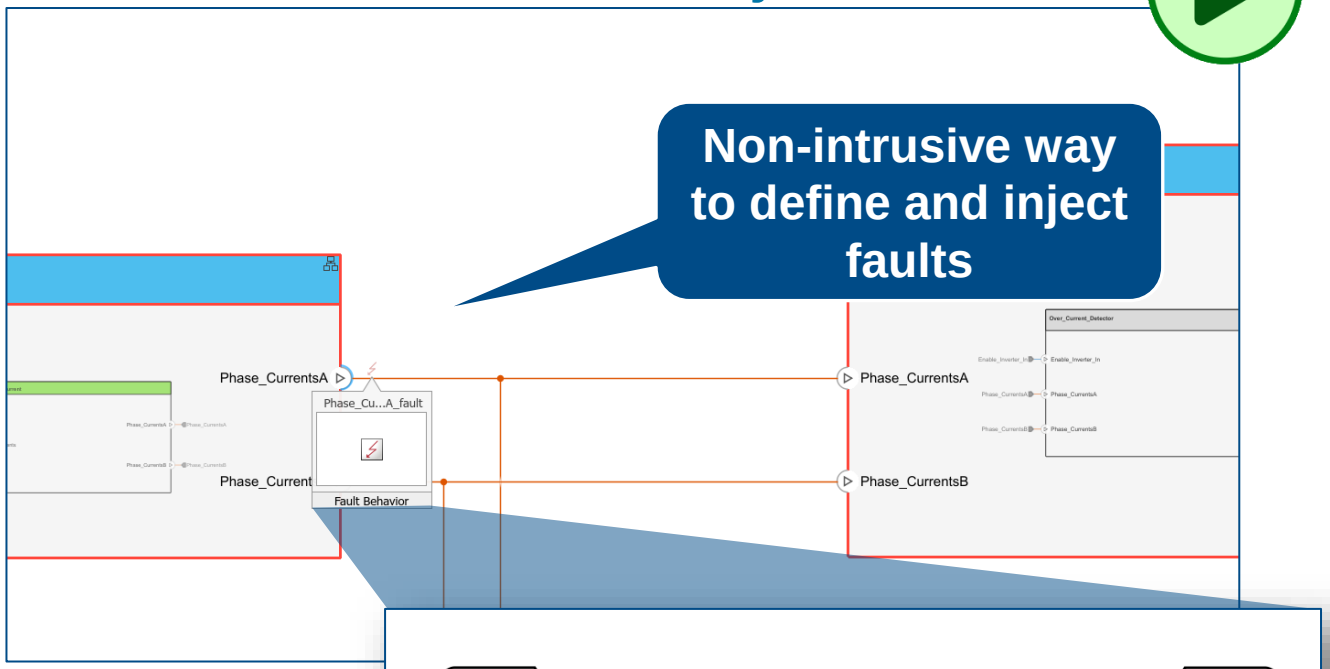


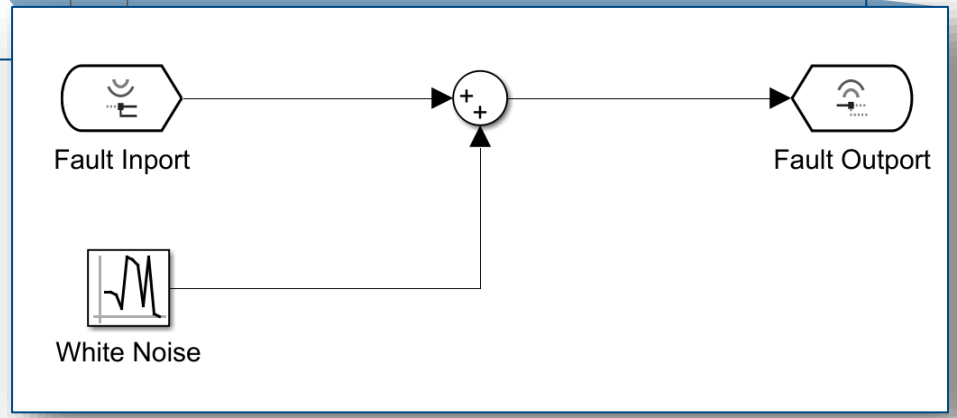


Validate System Behavior and Robustness with Fault Injection



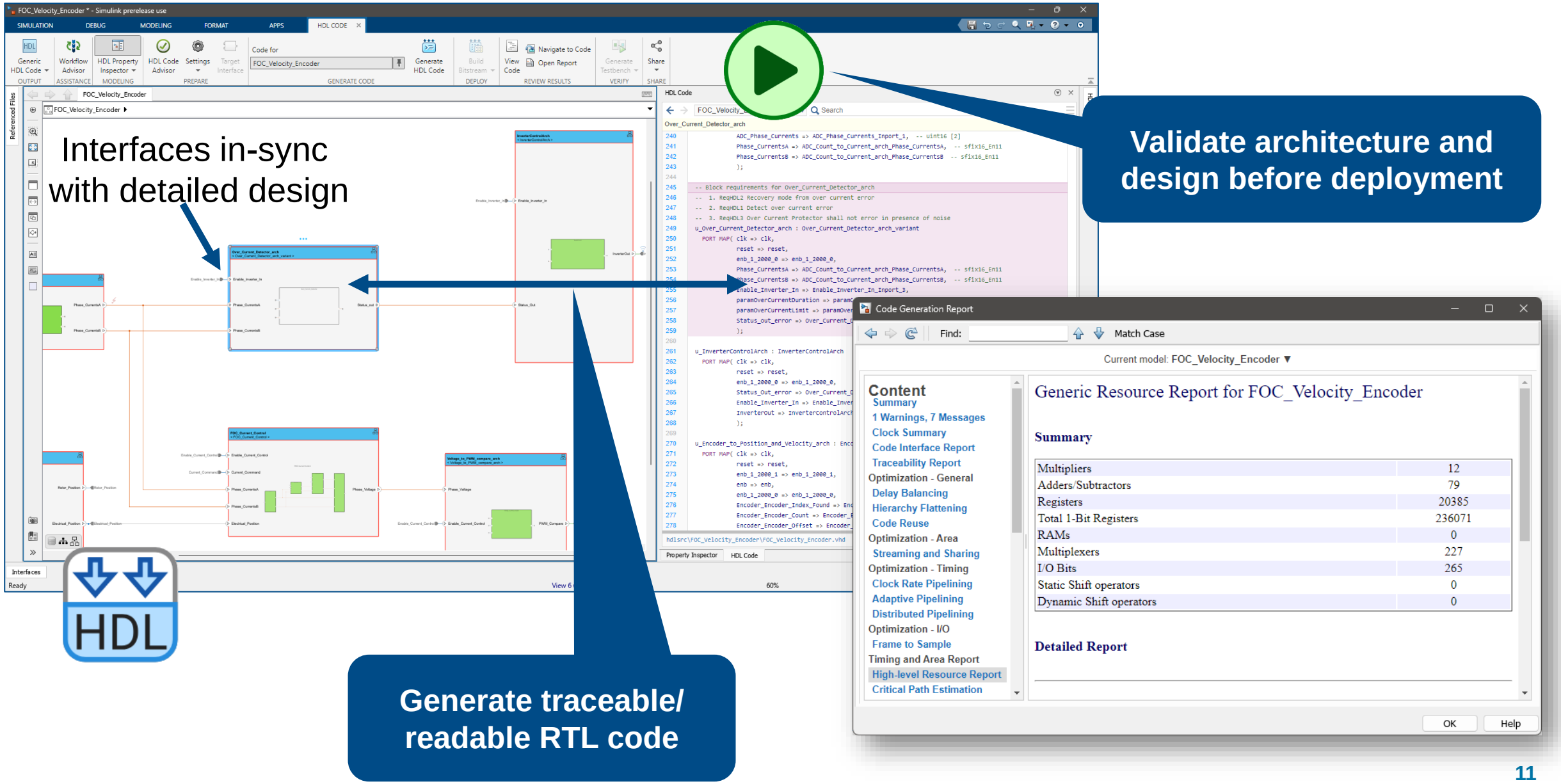






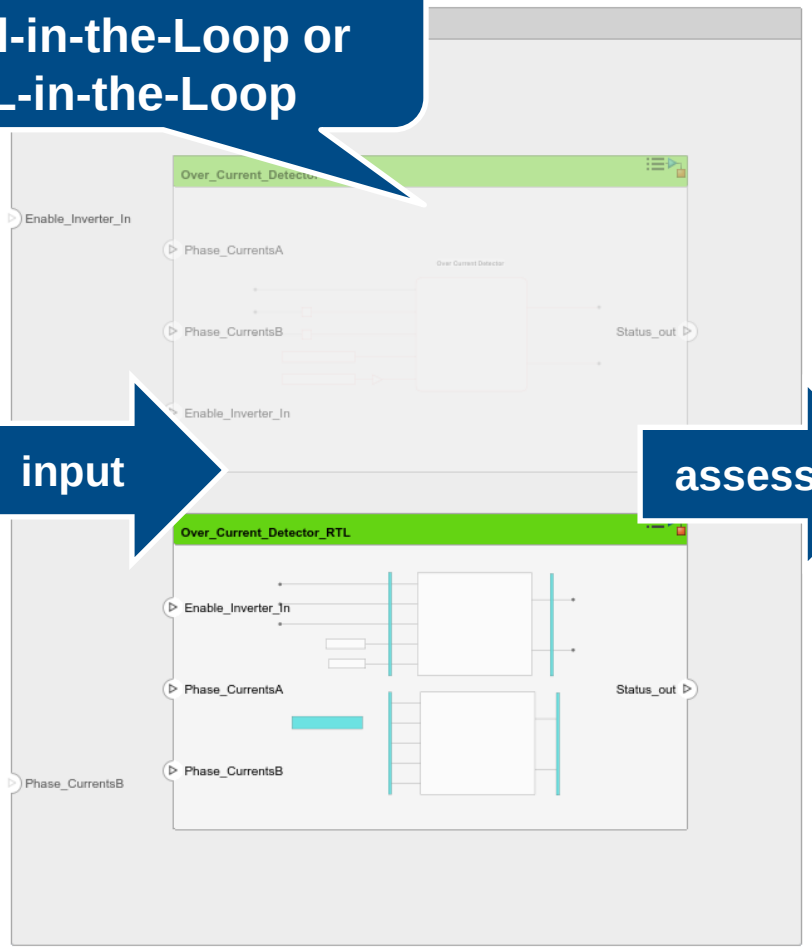
Analyze impact of faults using simulation

Generate HDL code from System Architecture incl Detailed Models



Re-use System Architecture and Design Models for RTL Verification & Validation

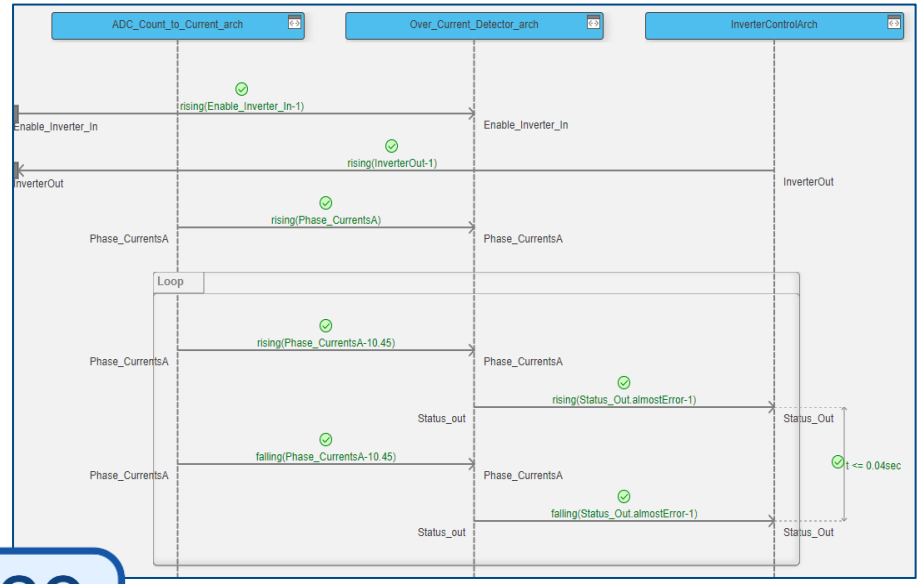
Variants to select Model-in-the-Loop or RTL-in-the-Loop



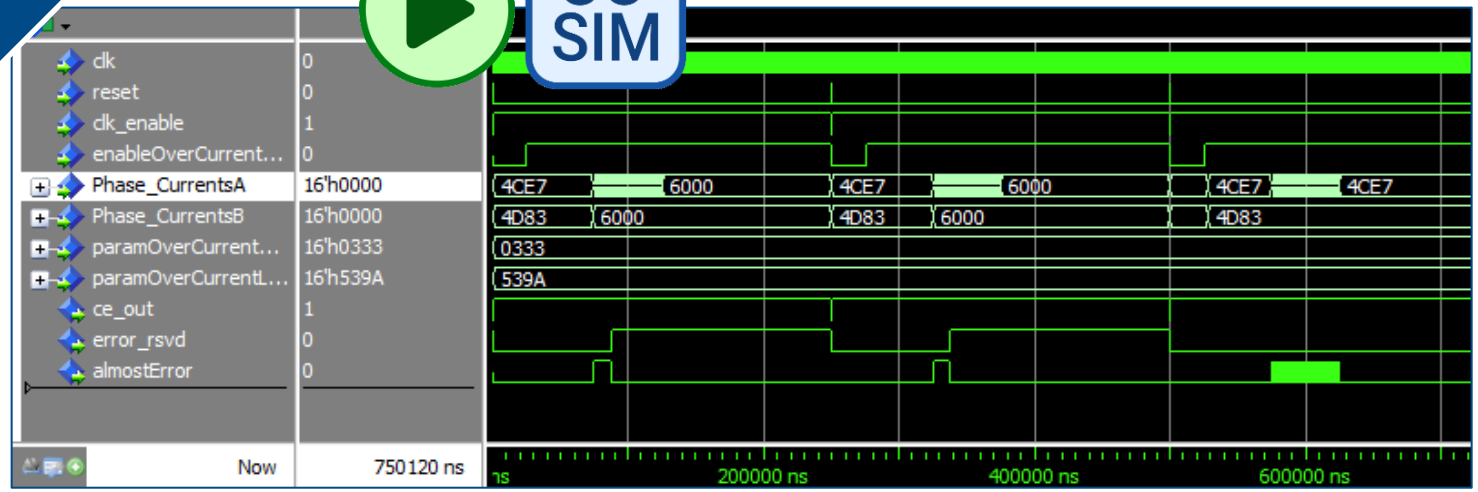
Excel file (input)

input

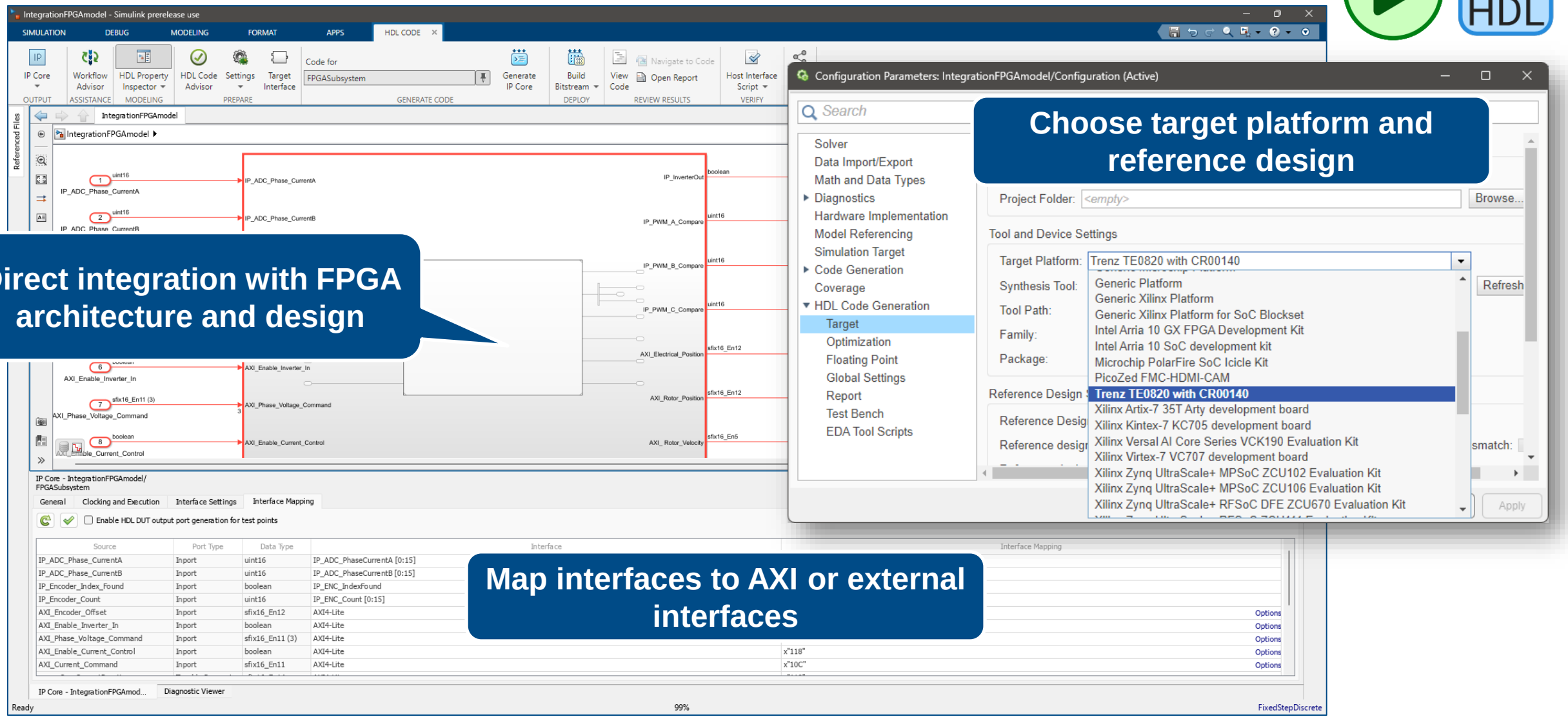
assess



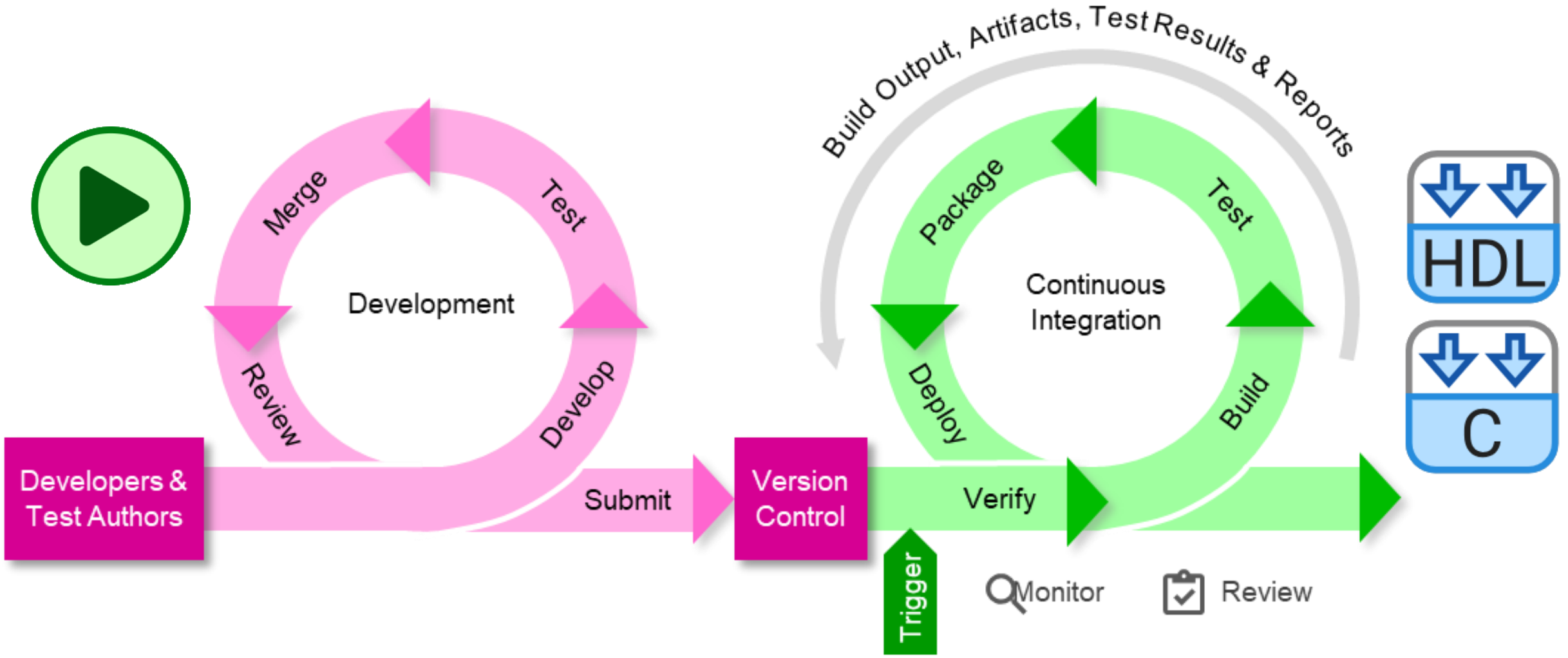
CO SIM



Integration with other IPs or Software Components

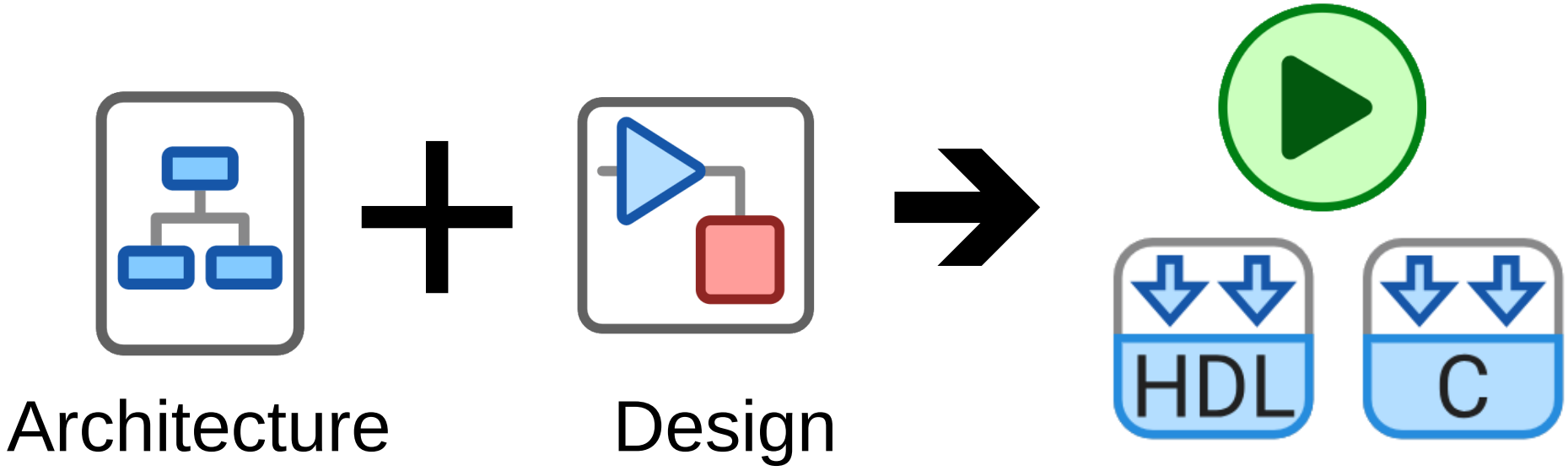


Continuous Integration for Model-Based Design



[Continuous Integration: CI/CD Automation for Model-Based Design] (support package)
[CI/CD Automation for Simulink Check] (reference book)

Concluding remark



Architecture Design

The main FPGA challenge lies in architecture, often informal and overlooked. A tool is needed to generate RTL directly from the architecture + design in a single model, making it the single source of truth.
- Adam Taylor, Adiuvo