MATLAB EXPO 2018

DO-254 Validation & Verification

Albert Ramirez Perez

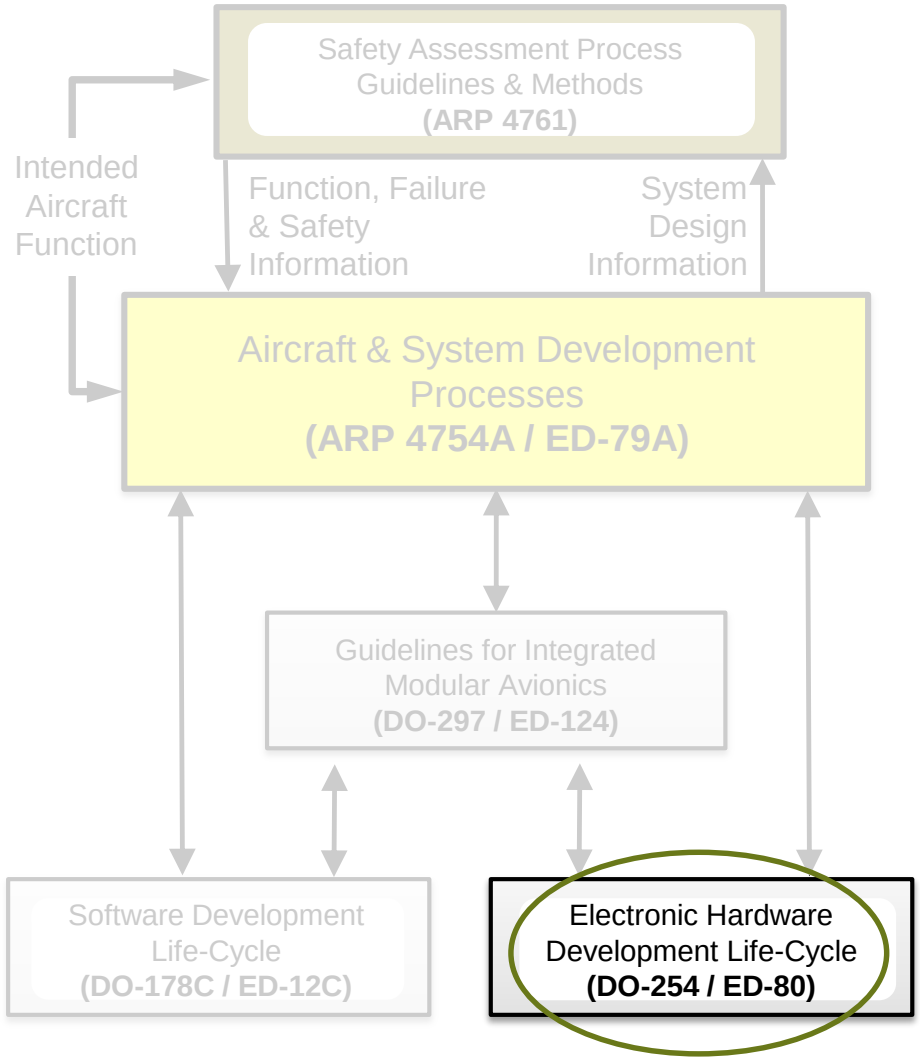
Outline

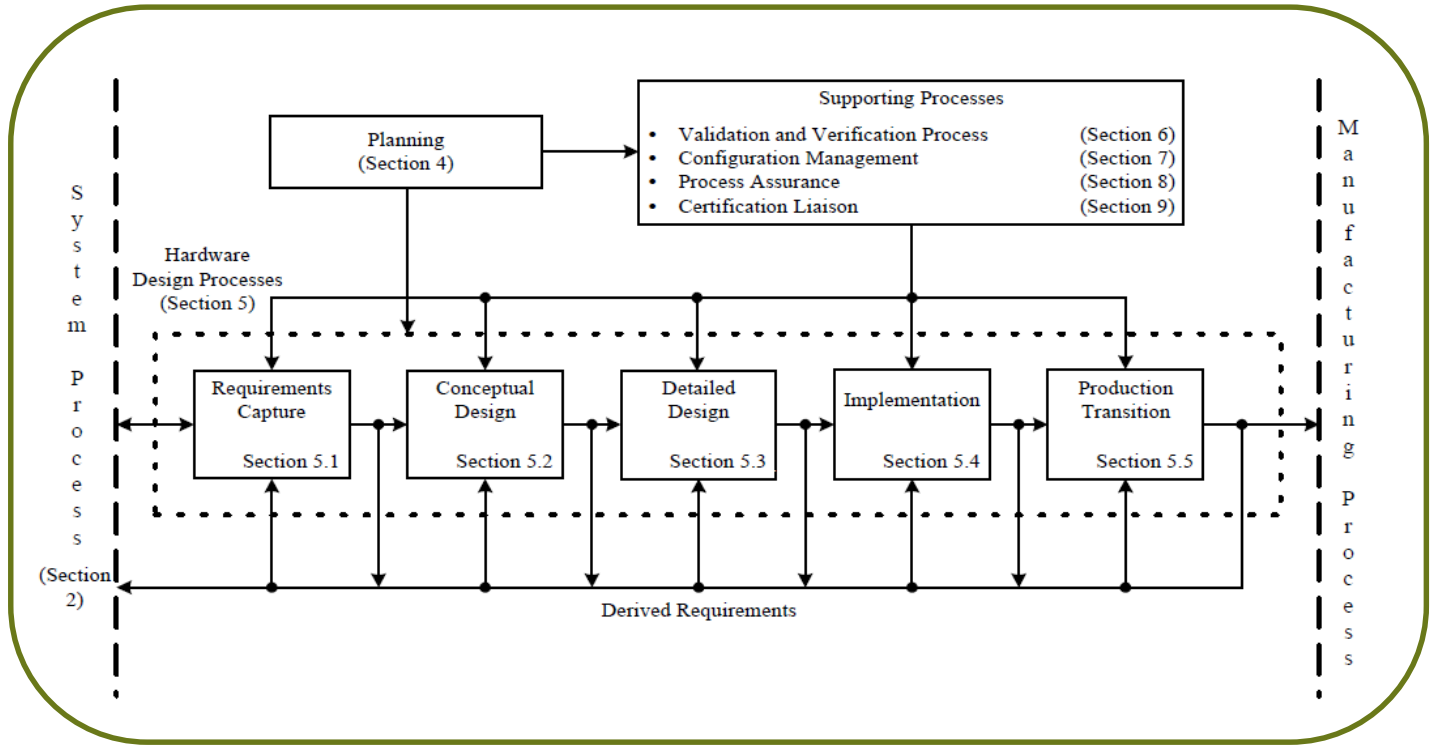
- Introduction to DO-254 – Design Guidance for Airborne Electronic HW
- MBD and other considerations for FPGA/ASIC Design
- Requirements Validation and Model Verification
- Hardware Verification
- Conclusion

Why Are Our Customers Deploying to FPGA/ASIC Hardware?

- Speed
- Power
- Latency
- Dissimilarity
- Robustness

How is the standard landscape?





Key definitions: System, Component, Item, Validation, Verification, Qualification, Certification

DO-254 Overview – history

- By **RTCA** Inc**.** (Radio Technical Commission for Aeronautics, est. 1935)
- Not for profit. Advisory committee developing standards for aviation industry.

- **DO-254** Standard provides **Design Assurance Guidance For Airborne Electronic Hardware.** DO-254 was adopted by FAA in 2005

- Airborne Electronic **Hardware** includes line replaceable units (LRUs), circuit board assemblies (PCBs), application specific integrated circuits (**ASIC**s), programmable logic devices (**PLD**s, **FPGA**s), etc.

DO-254 Overview - objectives

- DO-254 standard identifies hardware design life cycle **processes** and describes the **objectives** and activities for each process.
- Following the DO-254 standard processes and achieving their objectives is intended to meet the **design assurance level** and increase design **reliability**.
- One of the central themes in the DO-254 standard is the ability to capture, **document**, and **demonstrate** all the processes and their objectives, as identified by the standard, have been followed and met.

DO-254 Overview - emphasis

To ensure that the DO-254 guidance is considered and followed throughout the hardware life cycle, the DO-254 standard recommends and emphasizes on:

- **Planning**, **documenting** and, **reviewing** each step in the hardware life cycle
- **Capturing** and documenting **artifacts**, decisions, and action items throughout the hardware life cycle
- Showing **traceability** between the artifacts in each stage of the hardware life cycle

Outline

- Introduction to DO-254 – Design Guidance for Airborne Electronic HW
- MBD and other considerations for FPGA/ASIC Design
- Requirements Validation and Model Verification
- Hardware Verification
- Conclusion

DO-254 Reasons for a Robust HW Development Workflow

Hardware Design Assurance Levels

- A – Catastrophic (prevent continue safe flight and landing)

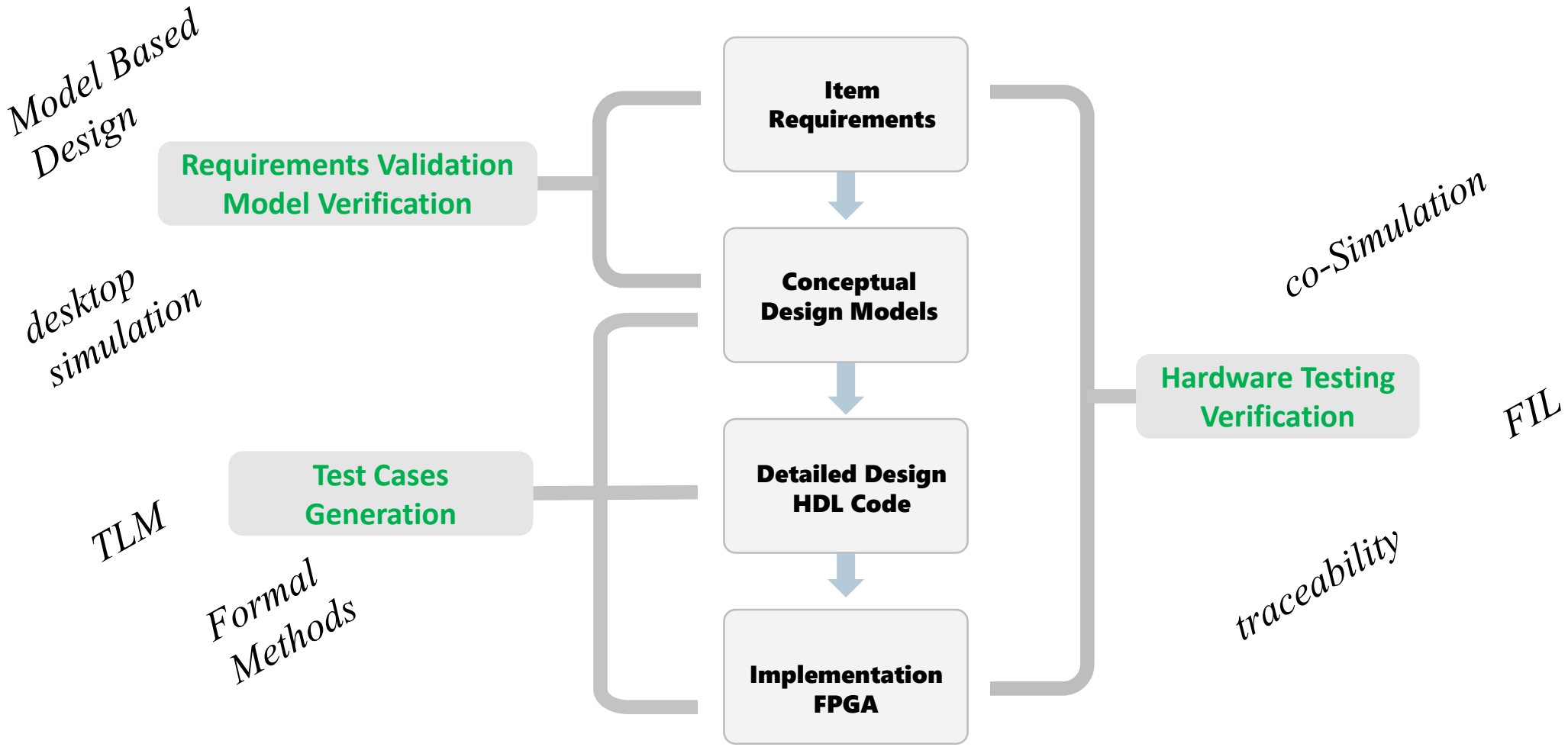
.....

- D – Minor (no significant effect on aircraft safety)

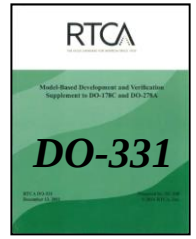
Hardware Design Processes

2. System and Hardware Safety Assessment
3. Hardware Design Life Cycle Processes
4. Planning Processes
5. Hardware Design Processes
6. Validation and Verification Processes
7. Configuration Management Processes

MathWorks DO-254 Model-Based Design Workflow



DO-331 Model-Based Development applied to HW



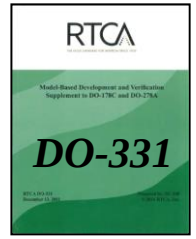
- Includes artifacts expressed using models or verification evidence derived from models

- Introduces new modeling activities

> Model simulation – The activity of exercising the behavior of a model using a model simulator.

> Model coverage analysis – An analysis that determines which requirements expressed by the Design Model were not exercised by verification based on the requirements from which the Design Model was developed. The purpose of this analysis is to support the detection of unintended function in the Design Model, where coverage of the requirements from which the model was developed has been achieved by the verification cases.

DO-331 Model-Based Development applied to HW



Defines 2 Models Types in the Glossary

Specification Model is a model representing high-level requirements that provides an abstract representation of functional, performance, interface, or safety characteristics of the software components. A Specification Model does not define item design details such as internal data structures, internal data flow, or internal control flow.

Design Model is a model that defines any item design such as low-level requirements, item architecture, algorithms, component internal data strictures, data flow and/or control flow. A model used to generate Code is a Design Model.

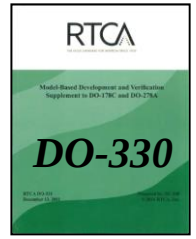
Introduce Model Usages Examples

Process that generates the life-cycle data	MB Example 3	MB Example 1	MB Example 2	MB Example 4	MB Example 5
System Requirement and System Design Processes	Requirements from which the Model is developed	Requirements allocated to software	Requirements from which the Model is developed	Requirements from which the Model is developed	Requirements from which the Model is developed
Item Requirement and Design Processes	Specification Model	Requirements from which the Model is developed	Specification Model	Design Model	Design Model
	Textual description	Design Model	Design Model		
Item Coding Process	Code	Code	Code	Code	Code

DO-331 Table MB.1-1 Model Usage Examples

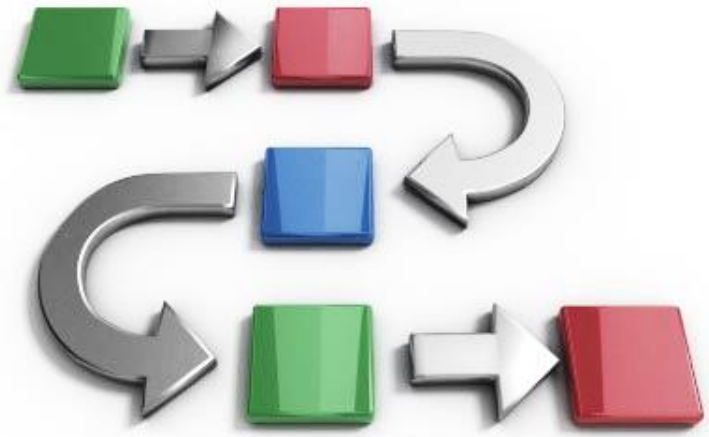
*In this slide, original text mentioning "software" was changed by "item" to be SW and HW applicable

Automating Tasks - Tool Qualification Kit



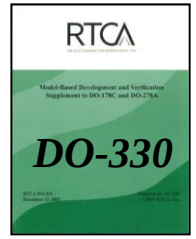
Tool Requirements
User Manuals

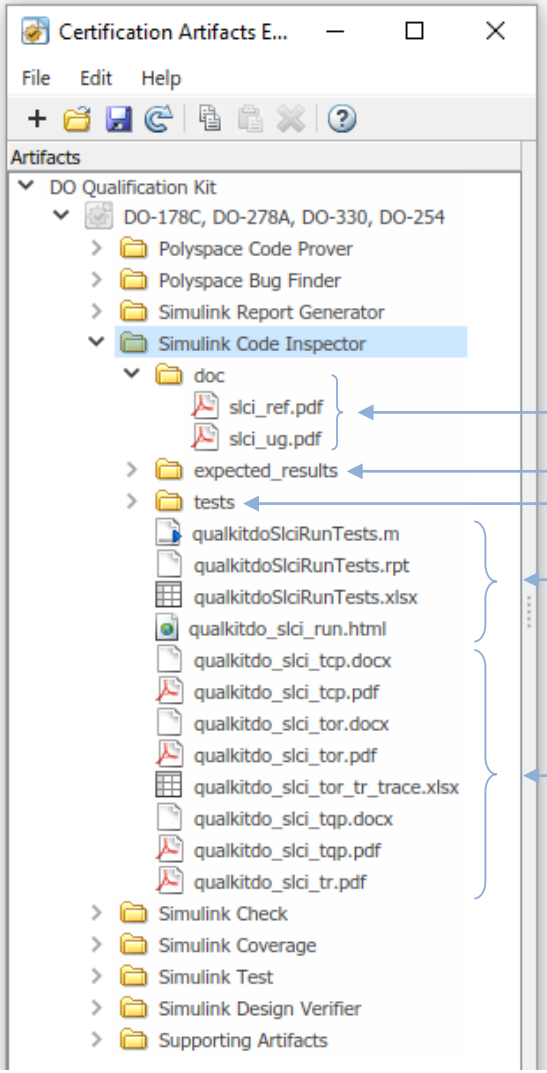
Test Cases Definition
Expected Results



Workflow Documentation
Templates for Plans

Automating Tasks - Tool Qualification Kit





Tool Documentation

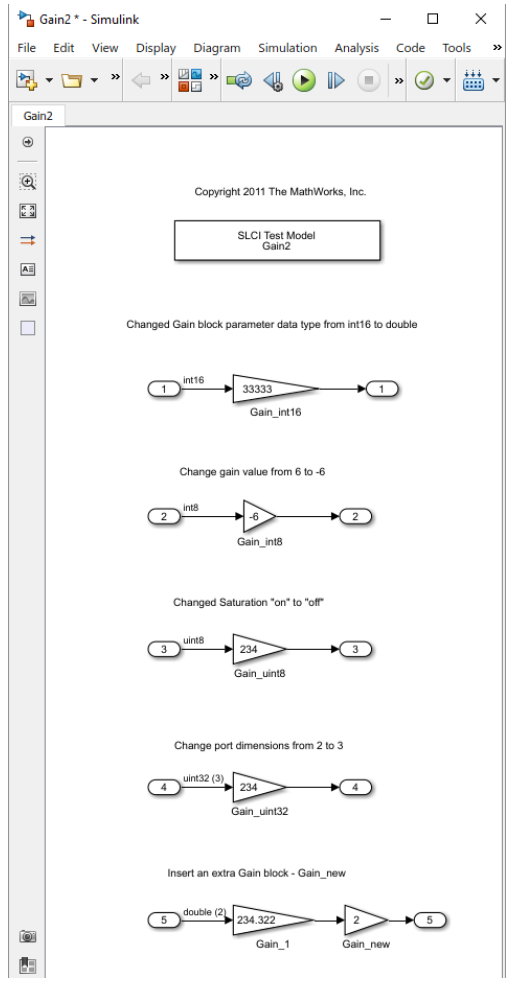
Expected Results

Test cases

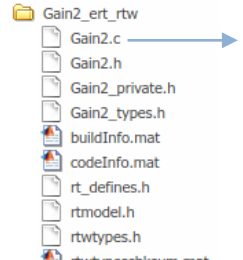
Scripts to generate the testing results reports

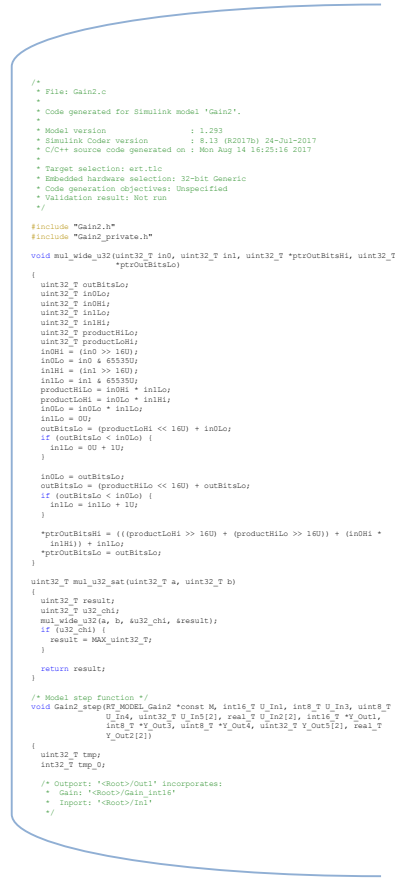
Plans templates plans to define the intended use and the desired certification credit

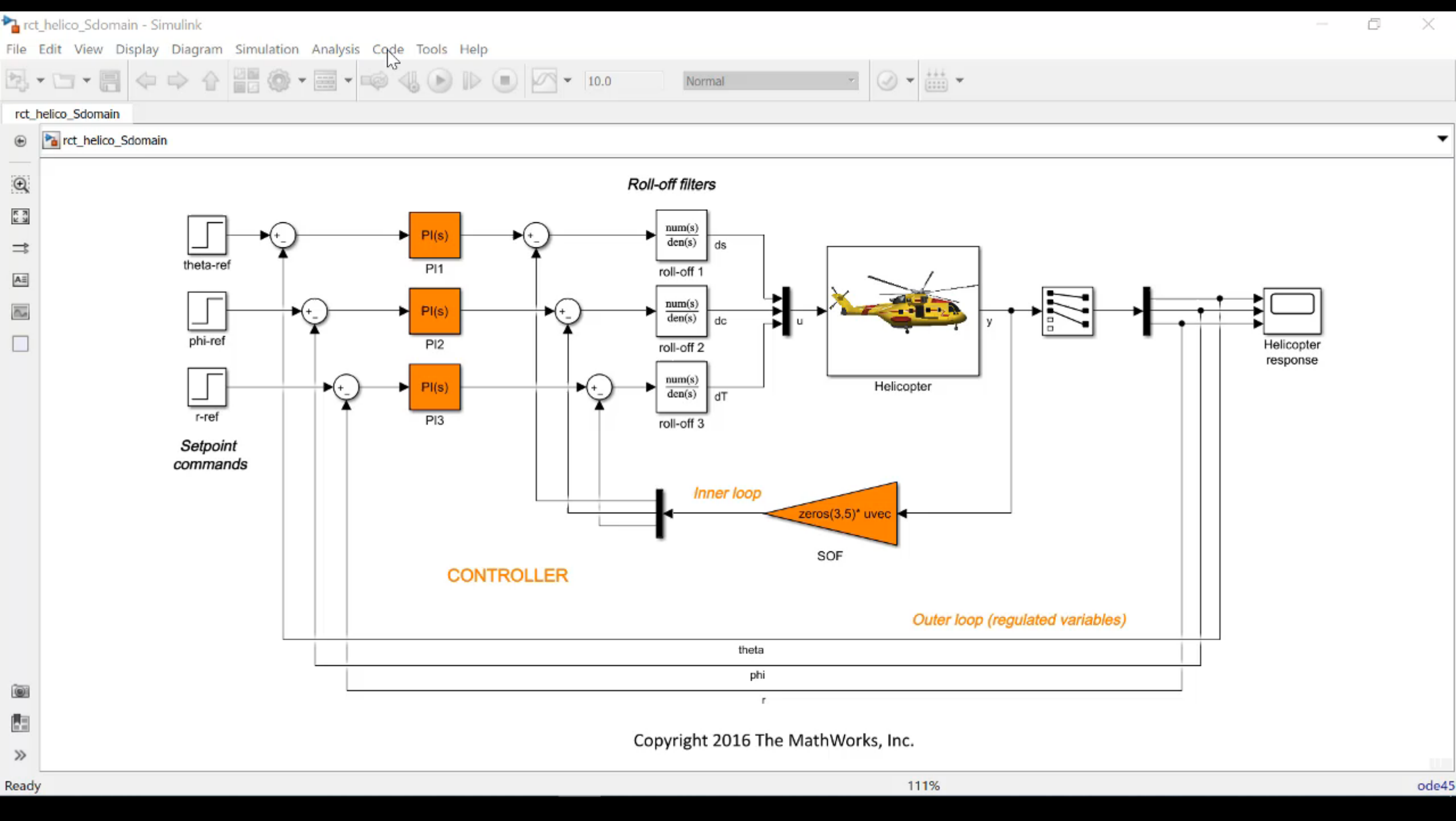
Test cases



Expected Results



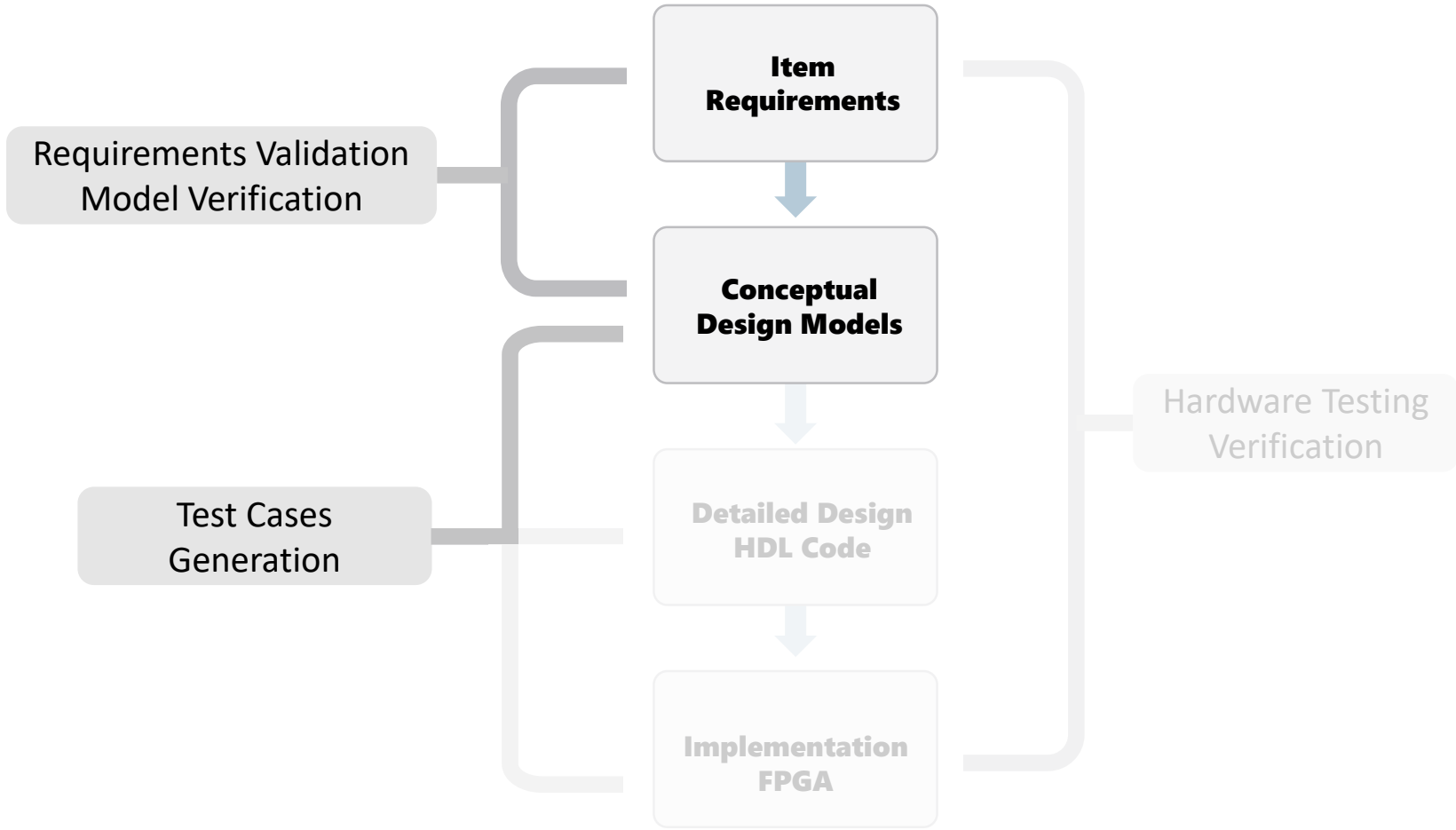



rct_helico_Sdomain - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools Help
10.0
Normal
rct_helico_Sdomain
rct_helico_Sdomain
Roll-off filters
theta-ref
phi-ref
r-ref
Setpoint commands
PI(s)
PI1
PI2
PI3
num(s) den(s)
roll-off 1
roll-off 2
roll-off 3
ds
dc
dT
u
Helicopter
y
Helicopter response
Inner loop
zeros(3,5)* uvec
SOF
CONTROLLER
Outer loop (regulated variables)
theta
phi
r
Copyright 2016 The MathWorks, Inc.
Ready
111%
ode45

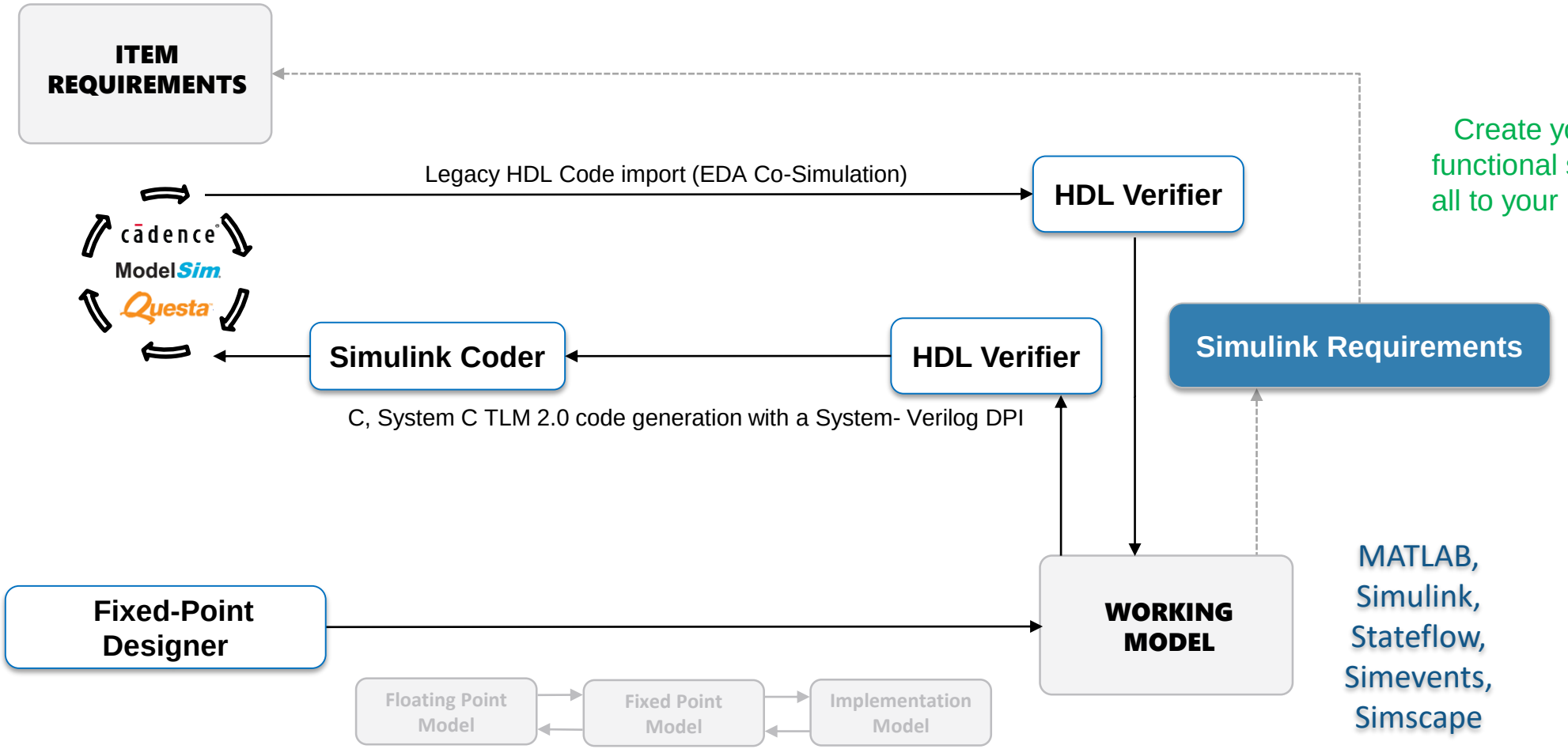
Outline

- Introduction to DO-254 – Design Guidance for Airborne Electronic HW
- MBD and other considerations for FPGA/ASIC Design
- Requirements Validation and Model Verification
- Hardware Verification
- Conclusion

MathWorks DO-254 Model-Based Design Workflow



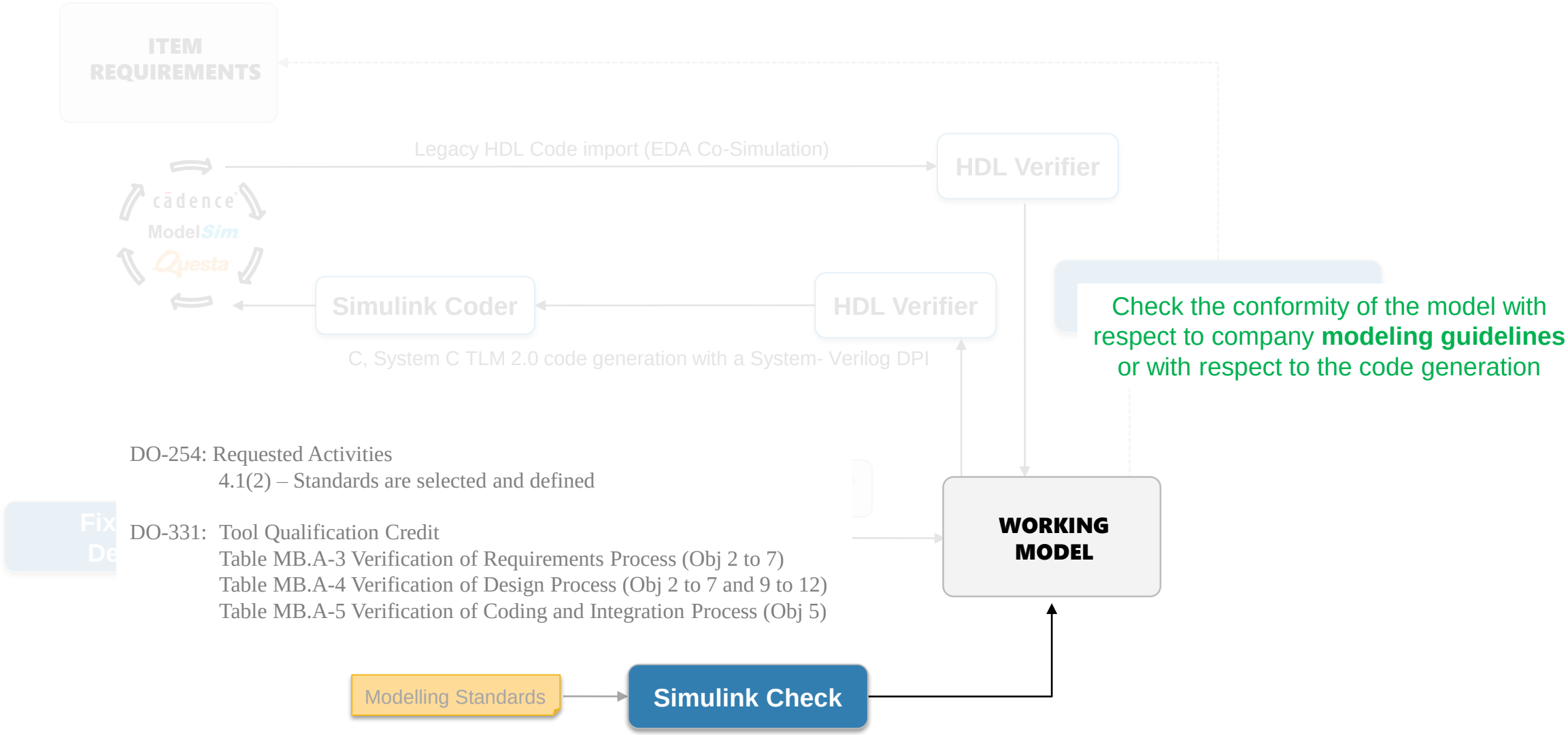
Meeting Requirements using Simulations



Create your design model, run functional simulations and **trace** it all to your high level requirements

DO-254: 6.3.3.1(5) – The requirements should be compatible with the relevant HW design standards

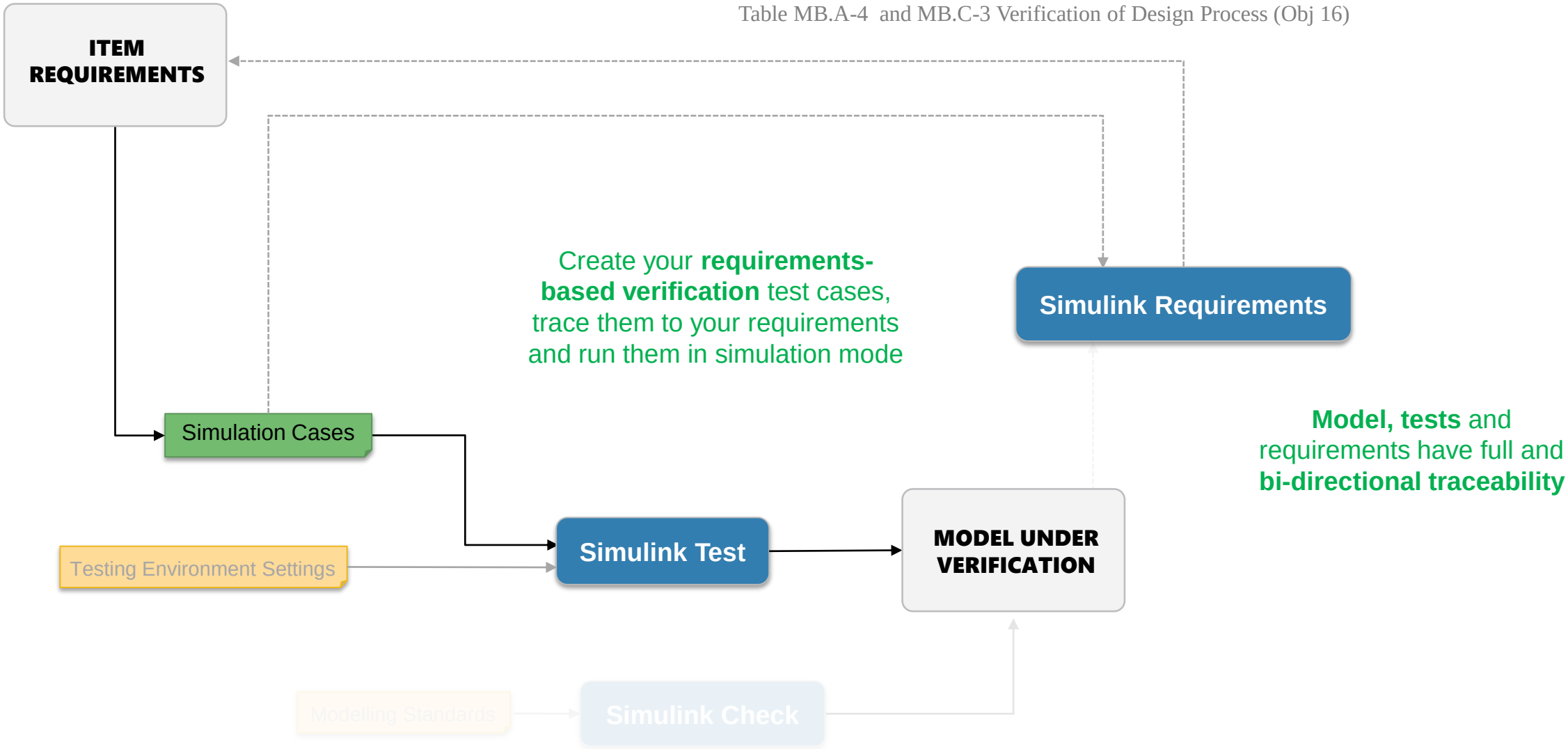
Compliance to Modelling Guidelines



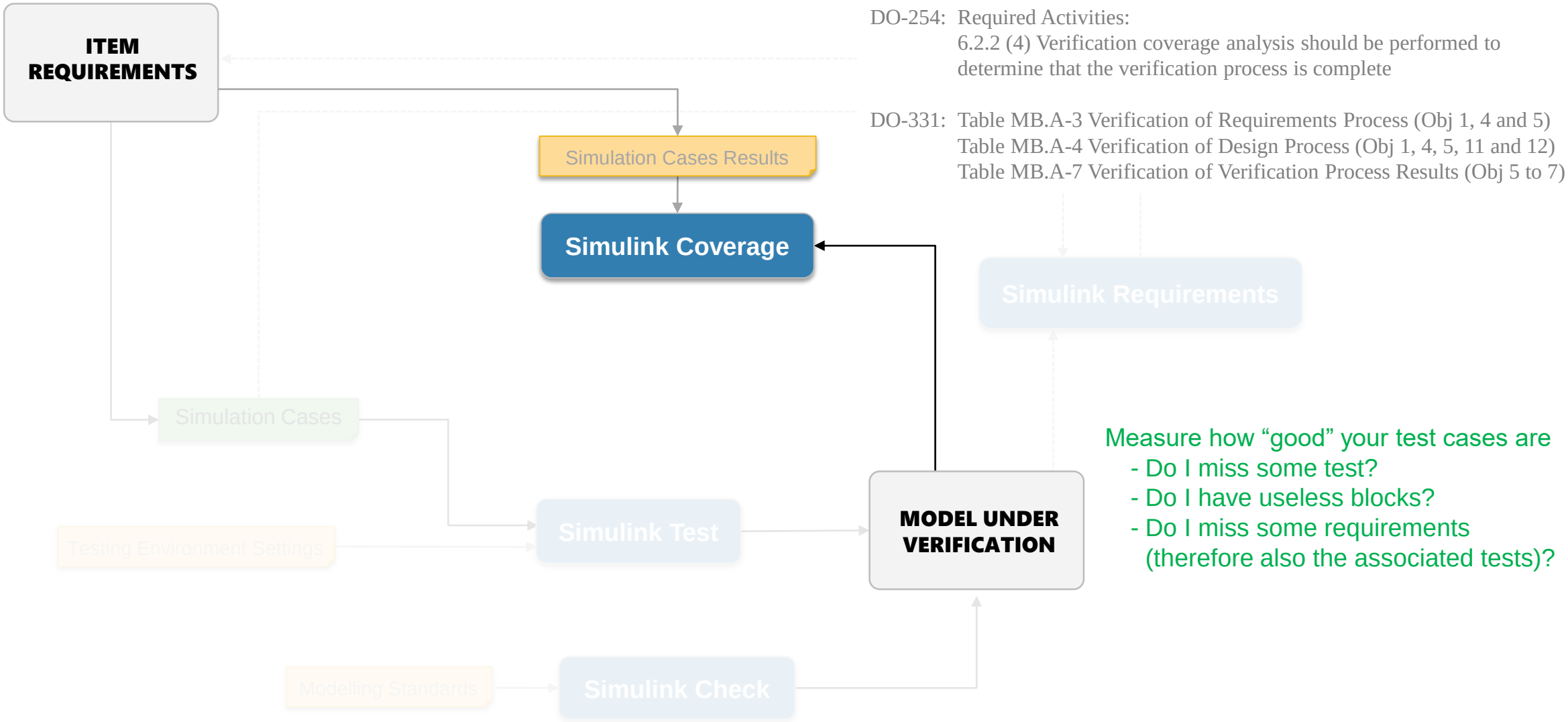
2018B : DO-254 Added to the Qualkit

Requirement-Based Test Cases Generation

DO-331: Table MB.A-3 and MB.C-3 Verification of Requirements Process (Obj 10)
Table MB.A-4 and MB.C-3 Verification of Design Process (Obj 16)



Coverage Metrics from your Requirement-Based Test Cases

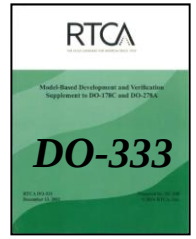


DO-254: Required Activities:
6.2.2 (4) Verification coverage analysis should be performed to determine that the verification process is complete

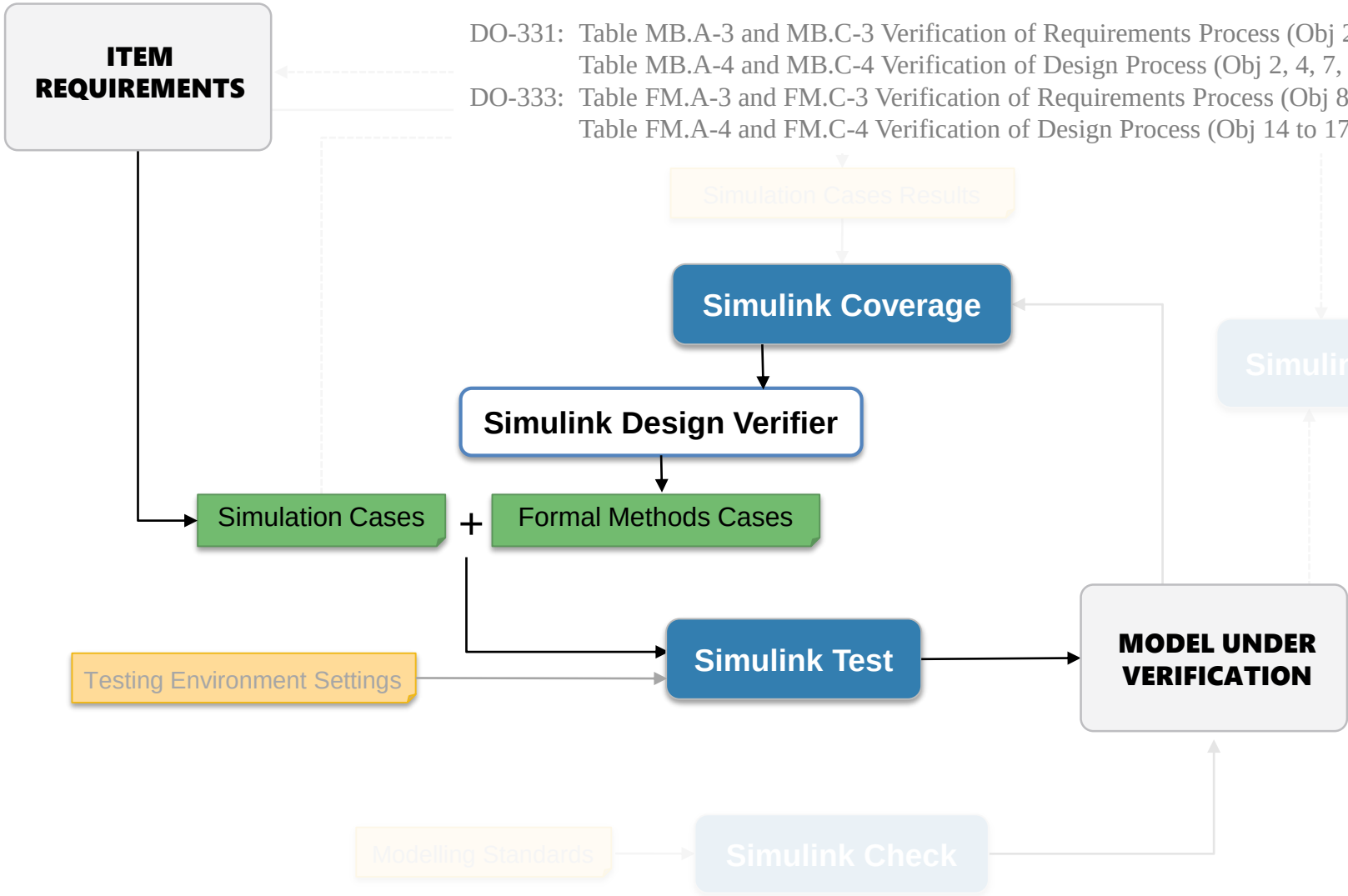
DO-331: Table MB.A-3 Verification of Requirements Process (Obj 1, 4 and 5)
Table MB.A-4 Verification of Design Process (Obj 1, 4, 5, 11 and 12)
Table MB.A-7 Verification of Verification Process Results (Obj 5 to 7)

Measure how “good” your test cases are
- Do I miss some test?
- Do I have useless blocks?
- Do I miss some requirements (therefore also the associated tests)?

Generate Automatically Missing Test Cases



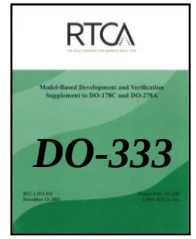
DO-331: Table MB.A-3 and MB.C-3 Verification of Requirements Process (Obj 2, 4 and 7)
Table MB.A-4 and MB.C-4 Verification of Design Process (Obj 2, 4, 7, 9,11)
DO-333: Table FM.A-3 and FM.C-3 Verification of Requirements Process (Obj 8 to 11)
Table FM.A-4 and FM.C-4 Verification of Design Process (Obj 14 to 17)



Leverage Formal Methods to generate automatically the missing test cases.

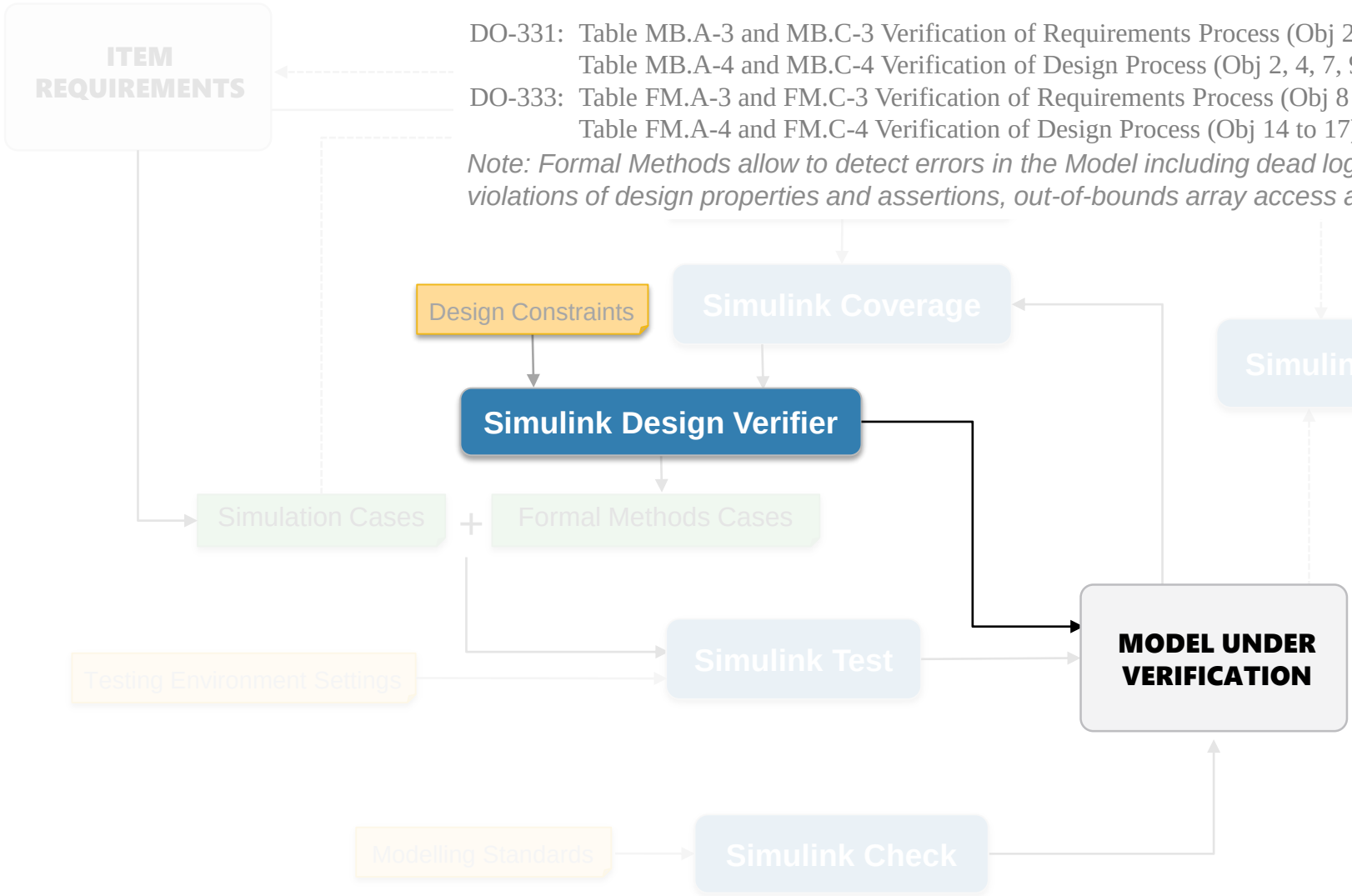
Run all test cases which covers 100% of your logical model

Model Design Error Detection and Property Proving



DO-331: Table MB.A-3 and MB.C-3 Verification of Requirements Process (Obj 2, 4 and 7)
Table MB.A-4 and MB.C-4 Verification of Design Process (Obj 2, 4, 7, 9,11)
DO-333: Table FM.A-3 and FM.C-3 Verification of Requirements Process (Obj 8 to 11)
Table FM.A-4 and FM.C-4 Verification of Design Process (Obj 14 to 17)

Note: Formal Methods allow to detect errors in the Model including dead logic, integer overflow, division by zero, and violations of design properties and assertions, out-of-bounds array access and certain other run-time errors in source code



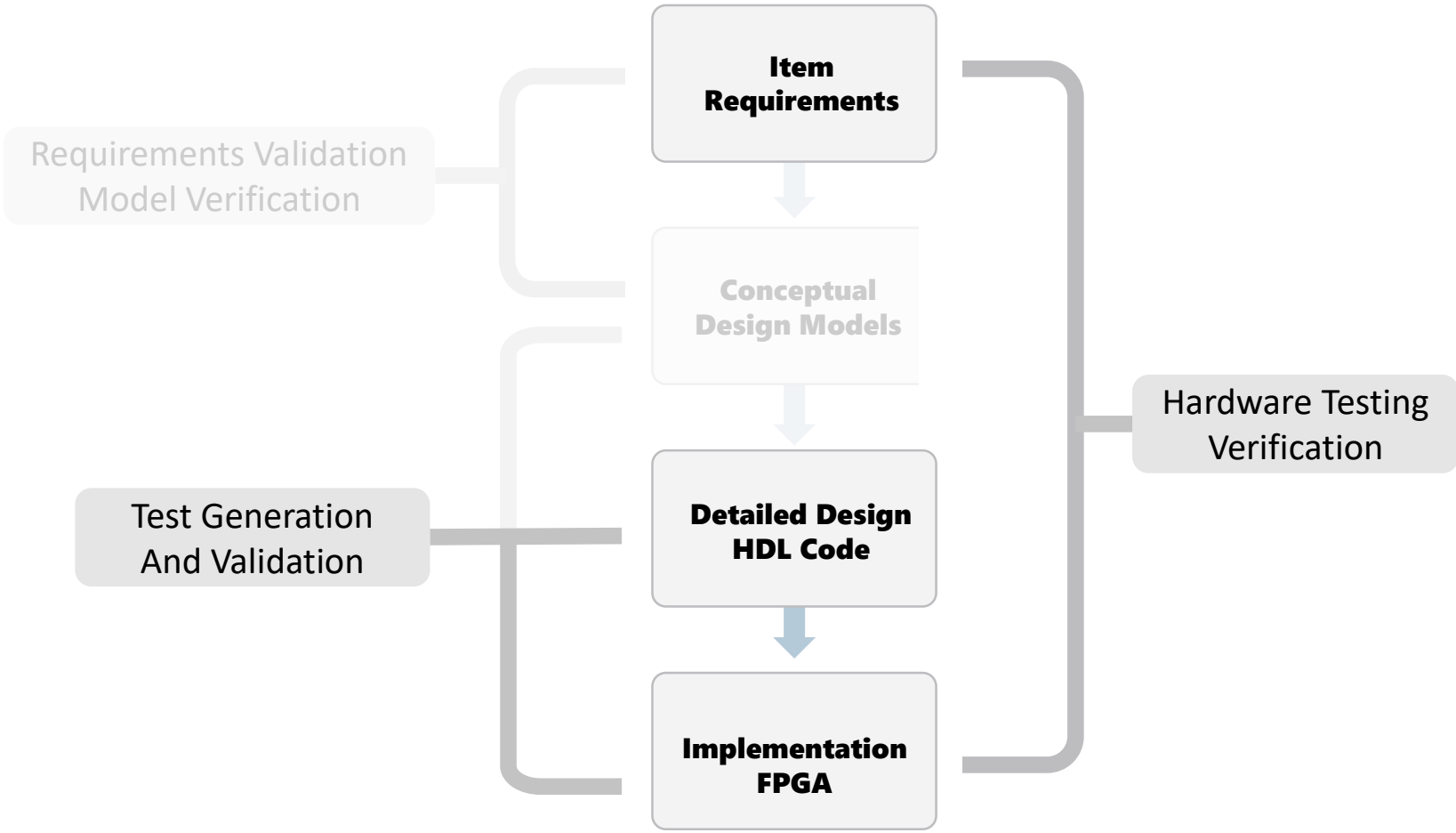
Leverage Formal Methods to Detect Design Errors like Divisions by zeros or out of ranges, etc.

You can also leverage Formal Methods to prove certain properties that are mission critical (don't deploy the parachute unless certain properties are meet)

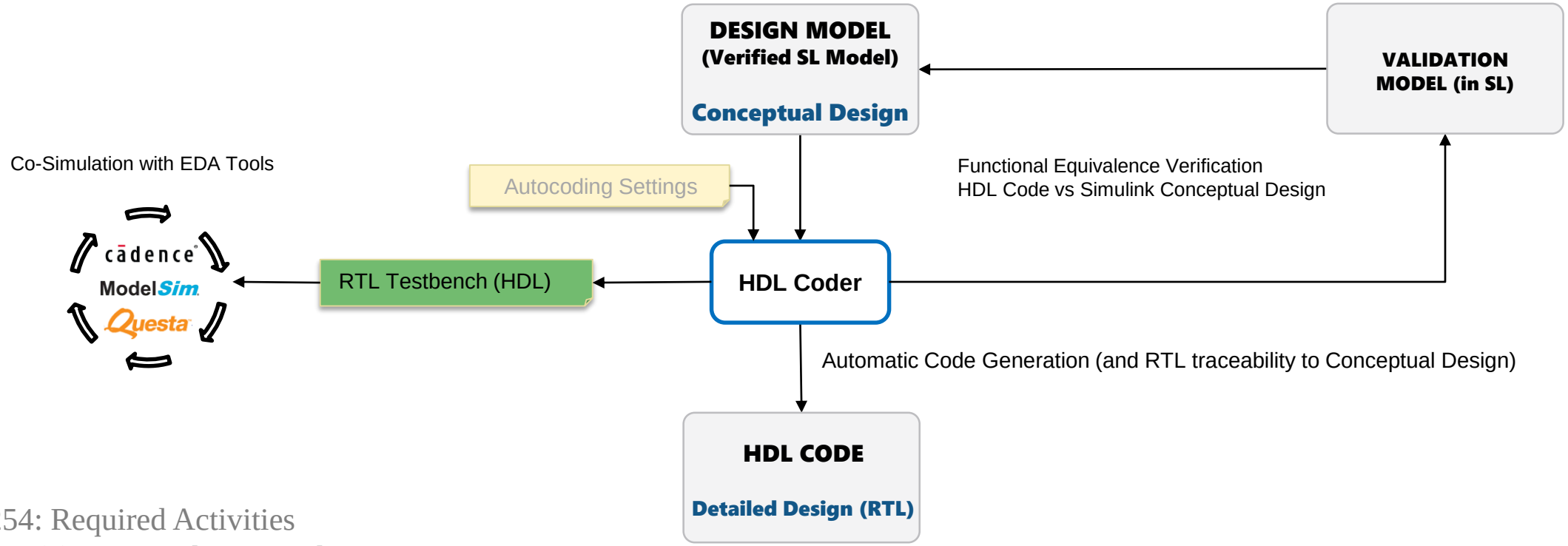
Outline

- Introduction to DO-254 – Design Guidance for Airborne Electronic HW
- MBD and other considerations for FPGA/ASIC Design
- Requirements Validation and Model Verification
- Hardware Verification
- Conclusion

MathWorks DO-254 Model-Based Design Workflow



Autogenerated Optimized HDL Functional Equivalence



DO-254: Required Activities

6.3.2(6) – A simulation analysis compares the simulation results to the expected results

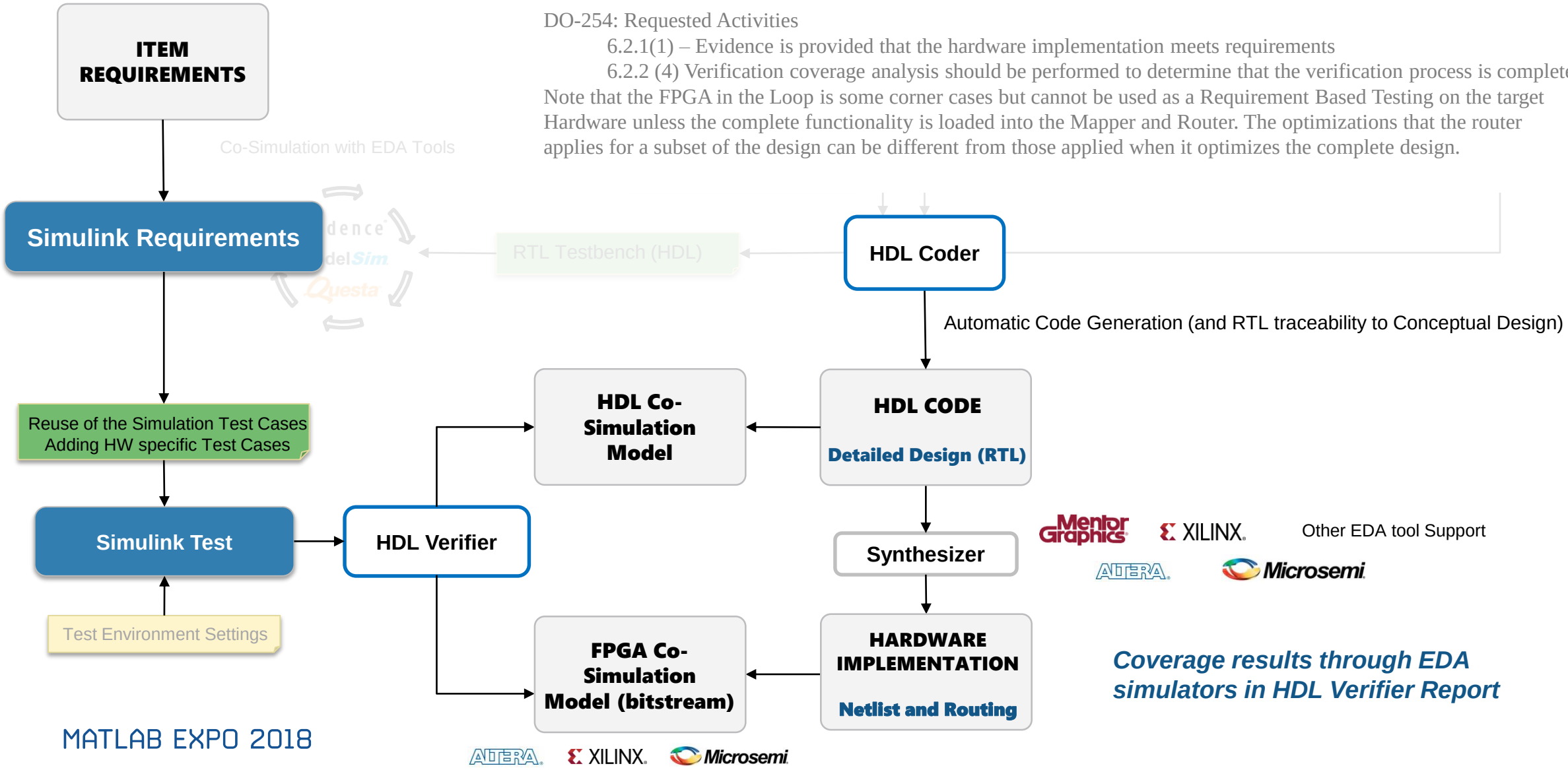
Functional and Hardware Performance Testing

DO-254: Requested Activities

6.2.1(1) – Evidence is provided that the hardware implementation meets requirements

6.2.2 (4) Verification coverage analysis should be performed to determine that the verification process is complete

Note that the FPGA in the Loop is some corner cases but cannot be used as a Requirement Based Testing on the target Hardware unless the complete functionality is loaded into the Mapper and Router. The optimizations that the router applies for a subset of the design can be different from those applied when it optimizes the complete design.

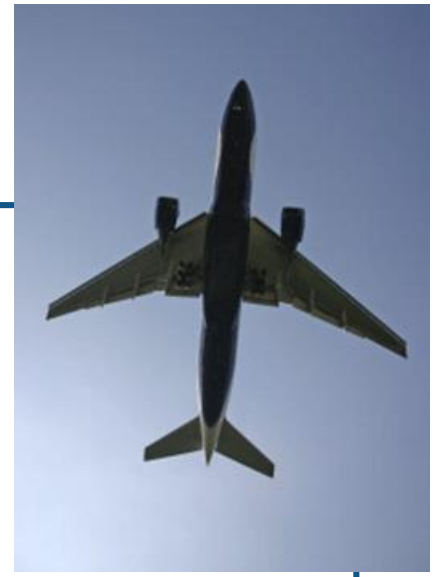


Coverage results through EDA simulators in HDL Verifier Report

Outline

- Introduction to DO-254 – Design Guidance for Airborne Electronic HW
- MBD and other considerations for FPGA/ASIC Design
- Requirements Validation and Model Verification
- Hardware Verification
- Conclusion

BAE Utilizes HDL Coder for DO-254 Level A Project



Challenge

Develop and deploy a control law to an FPGA that can be certified to **DO-254 Level A** for a commercial business jet.

Solution

Develop control laws in Simulink, link requirements to blocks using Simulink Verification and Validation, and generate HDL code using HDL Coder.

Results

- Simulink models are used to develop and verify designs, and generate HDL code
- Generated **code is platform independent, readable, and efficient**
- **Traceability** established between requirements, detailed design expressed in Simulink, and HDL implemented for DO-254 Level A certification

"HDL Coder generates readable code that is traceable to the requirements and is critical to our DO-254 Level A certification plans."

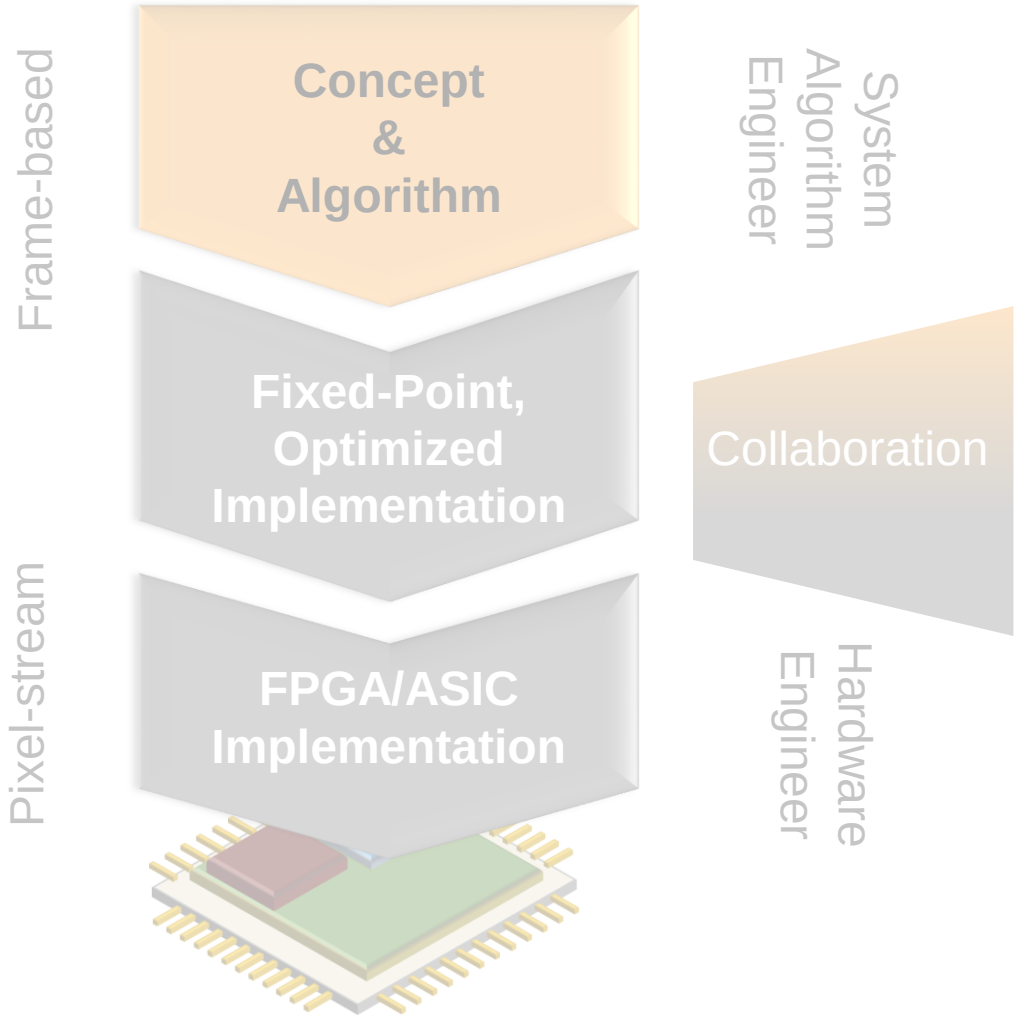
Mike Weaver

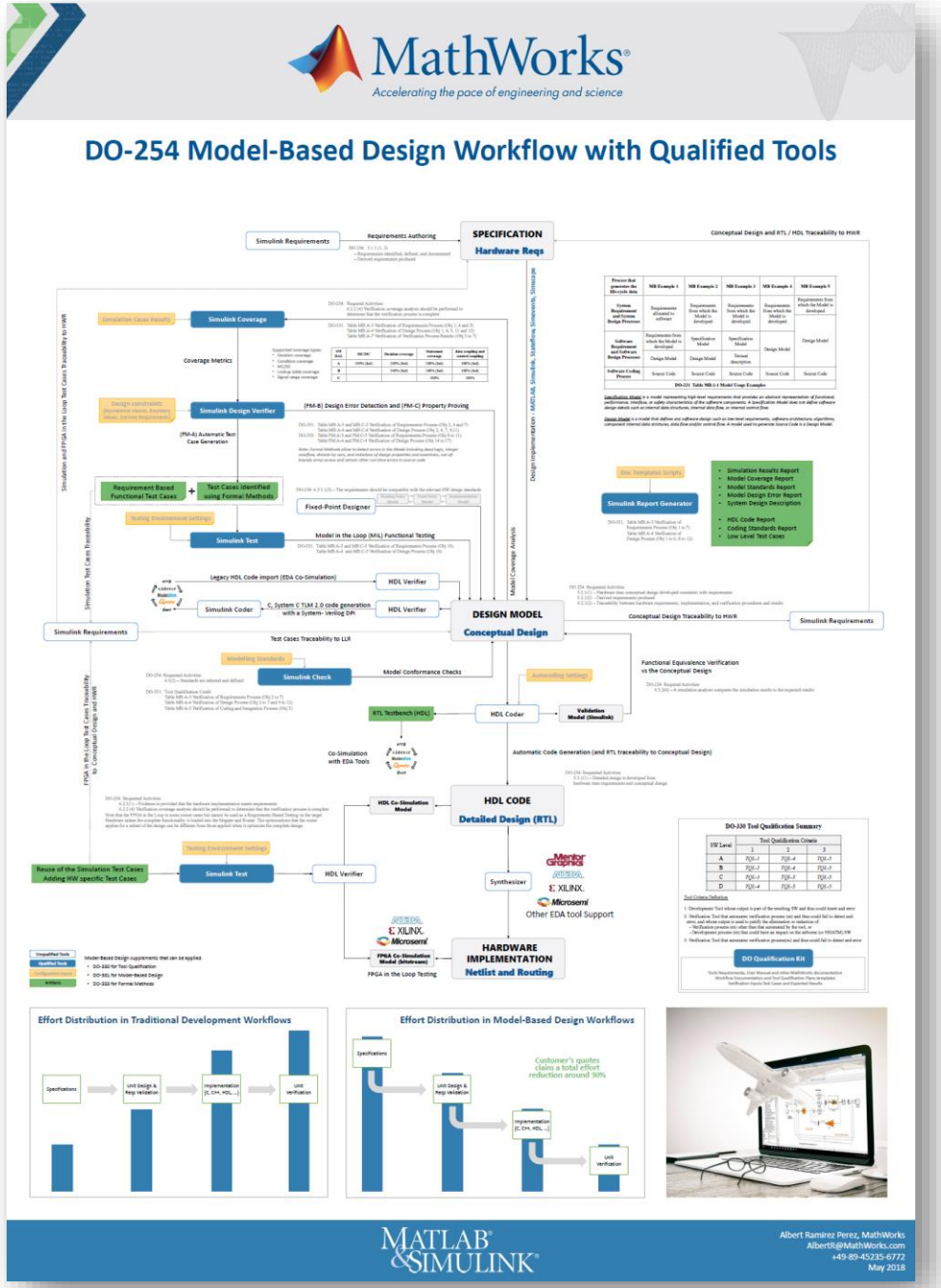
Senior Systems Engineer

MathWorks DO-254 Model-Based Design is your Right Choice

- The tools are mature and have been used in many projects today.
- DO Qualkit makes the qualification of the V&V tools easy.
- Customers and Cert Authorities acknowledge the value.
- MathWorks increasing investments in HW workflows.

Learn More


Frame-based
Concept
&
Algorithm
System
Algorithm
Engineer
Fixed-Point,
Optimized
Implementation
Collaboration
Pixel-stream
FPGA/ASIC
Implementation
Hardware
Engineer


MathWorks
Accelerating the pace of engineering and science
DO-254 Model-Based Design Workflow with Qualified Tools
SPECIFICATION
Hardware Reqs
DESIGN MODEL
Conceptual Design
HDL CODE
Detailed Design (RTL)
HARDWARE IMPLEMENTATION
Netlist and Routing
Other EDA tool Support
Effort Distribution in Traditional Development Workflows
Effort Distribution in Model-Based Design Workflows
MATLAB & SIMULINK
Albert Ramirez Perez, MathWorks
Albert.R@MathWorks.com
+49-89-45235-6772
May 2018