Accelerating FPGA/ASIC Design and Verification

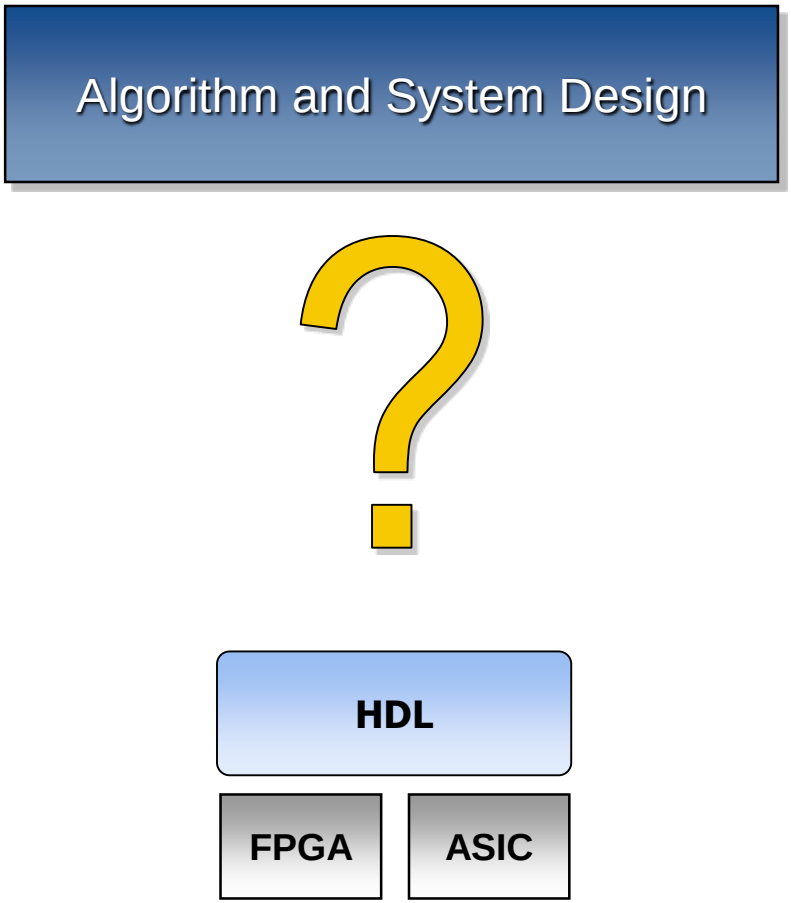
Puneet Kumar
Application Engineering Team

Agenda

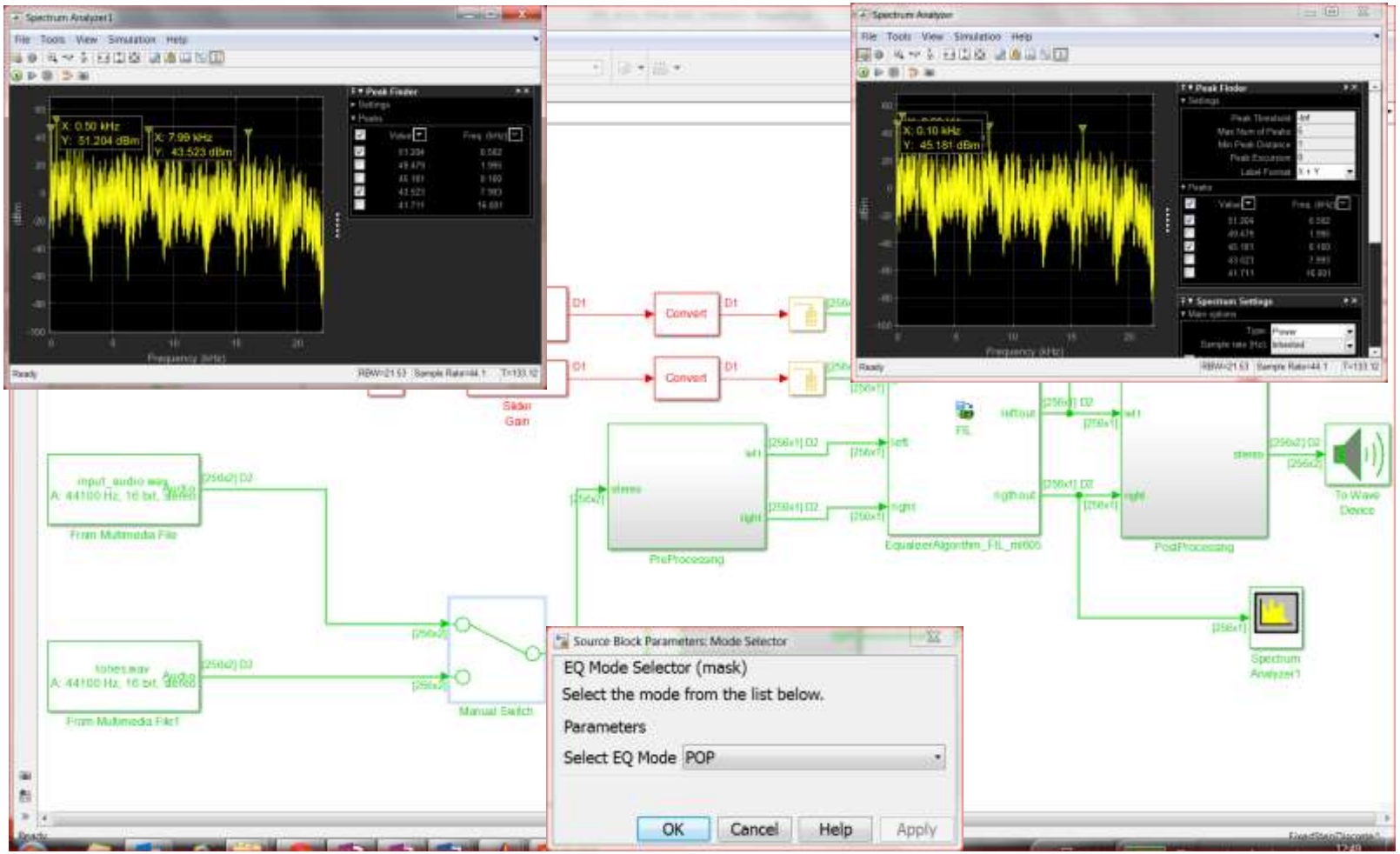
- **HDL code generation from MATLAB, Simulink, and Stateflow®**
- **Integrated RTL verification with EDA tools**
- **Advanced Techniques to Optimize the Generated HDL**
  - Workflow to explore Area and Speed Optimization
- **FPGA-in-the-loop Verification**
- **Next Steps, Q & A**

The System Design Challenge

- How can we:
  - Verify our hardware implementation matches system specification?
  - Iterate our designs faster?

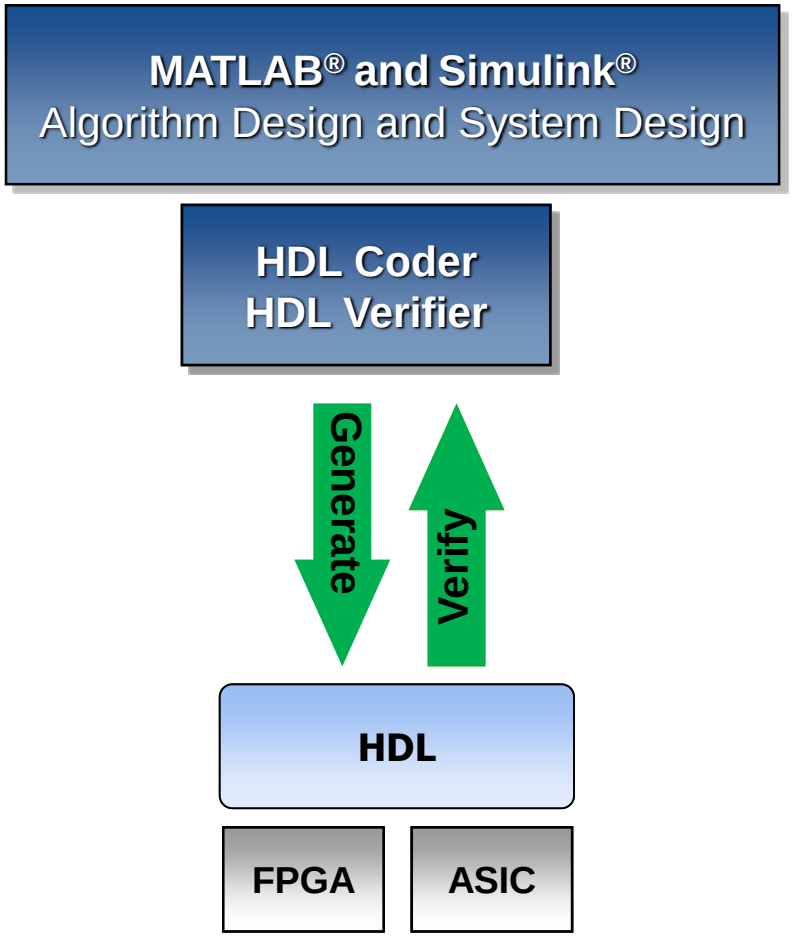


Audio Equalizer Demo



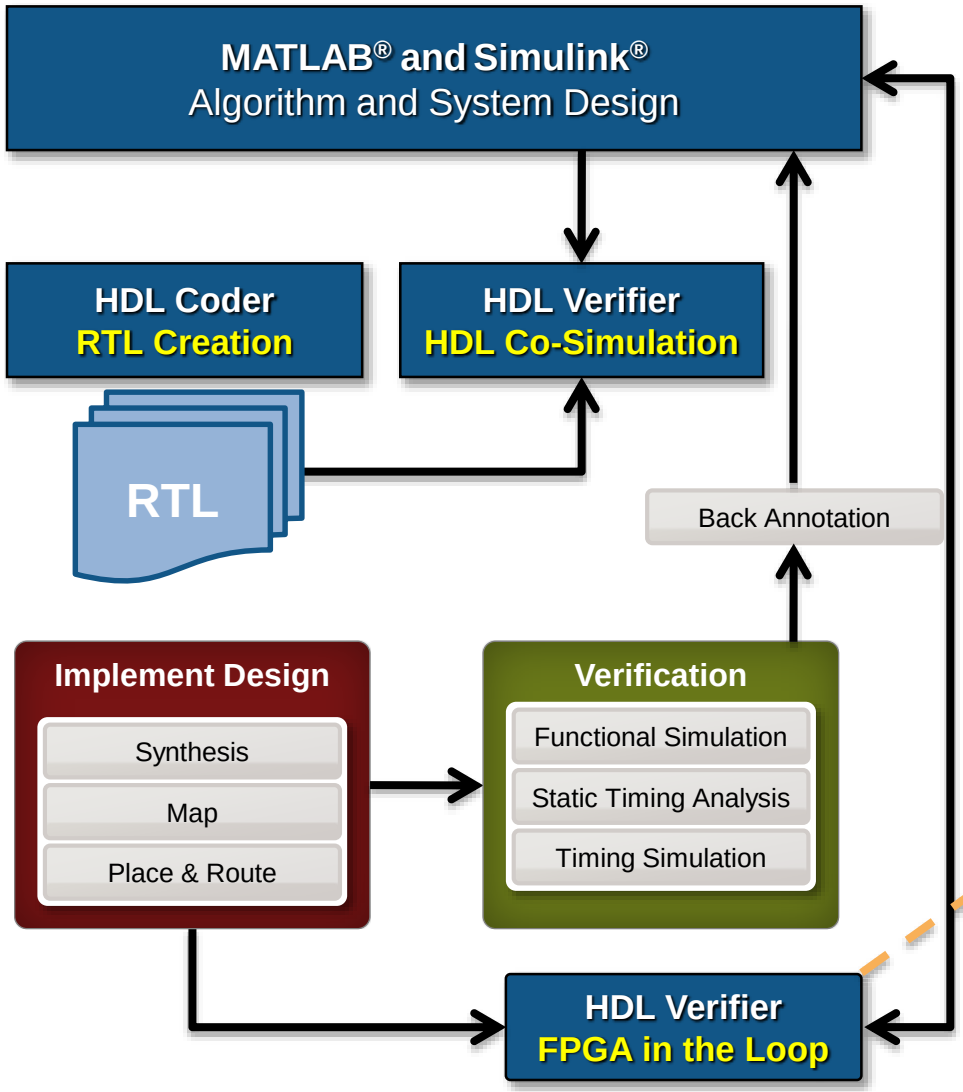
Solution: Model-Based Design

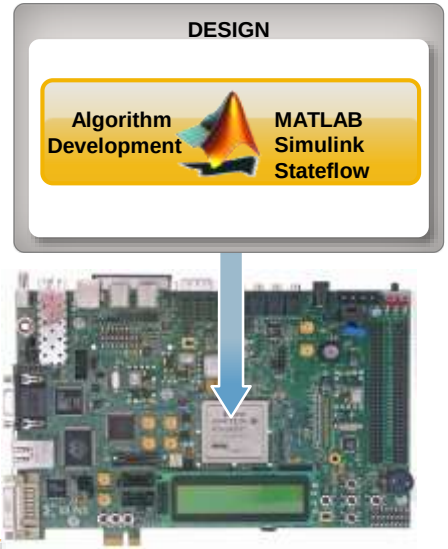
- Design, simulate, and validate algorithms and system models in MATLAB and Simulink
- Automatically generate HDL code
- Verify the hardware implementation against the system model



Model-Based Design flow using MATLAB/Simulink

from Algorithm to FPGA Implementation





Agenda

- **HDL code generation from MATLAB, Simulink, and Stateflow®**
- Integrated RTL verification with EDA tools
- Advanced Techniques to Optimize the Generated HDL
  - Workflow to explore Area and Speed Optimization
- FPGA-in-the-loop Verification
- Next Steps, Q & A

HDL Coder: It's all about the *Workflow*

MATLAB

SIMULINK

HDL Coder

VHDL or Verilog

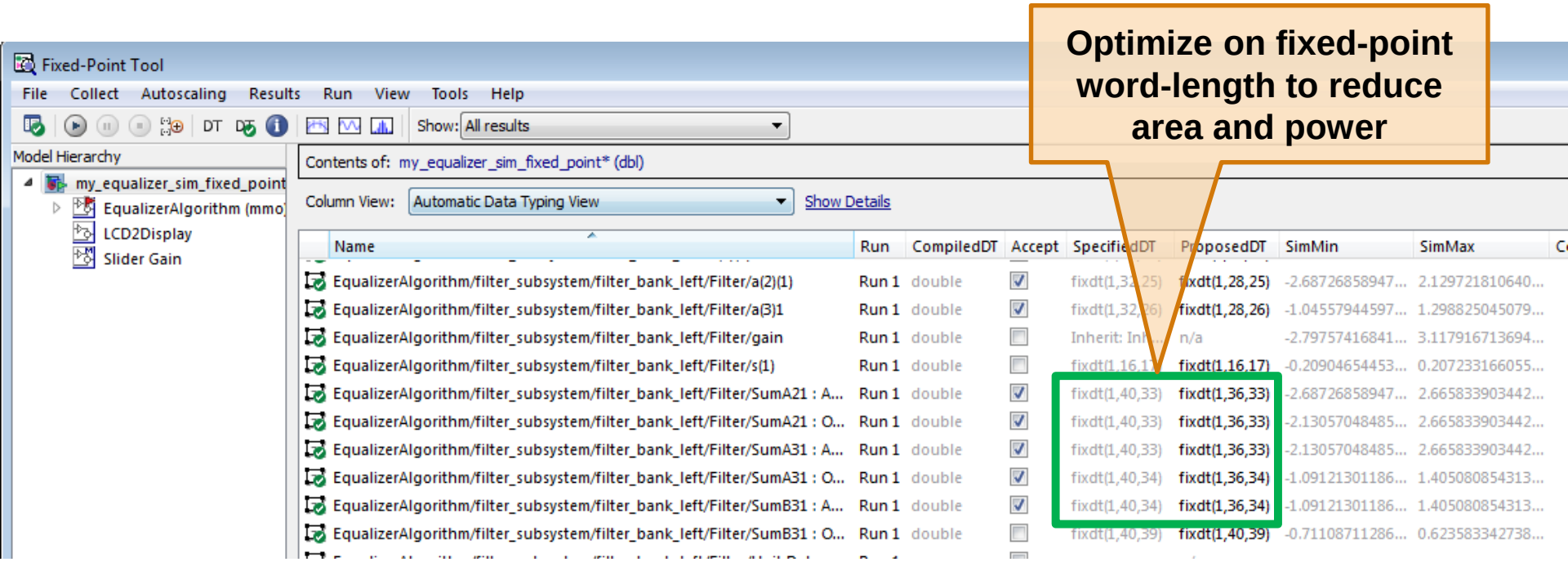
ASIC

FPGA Boards

HW-SW Co-design

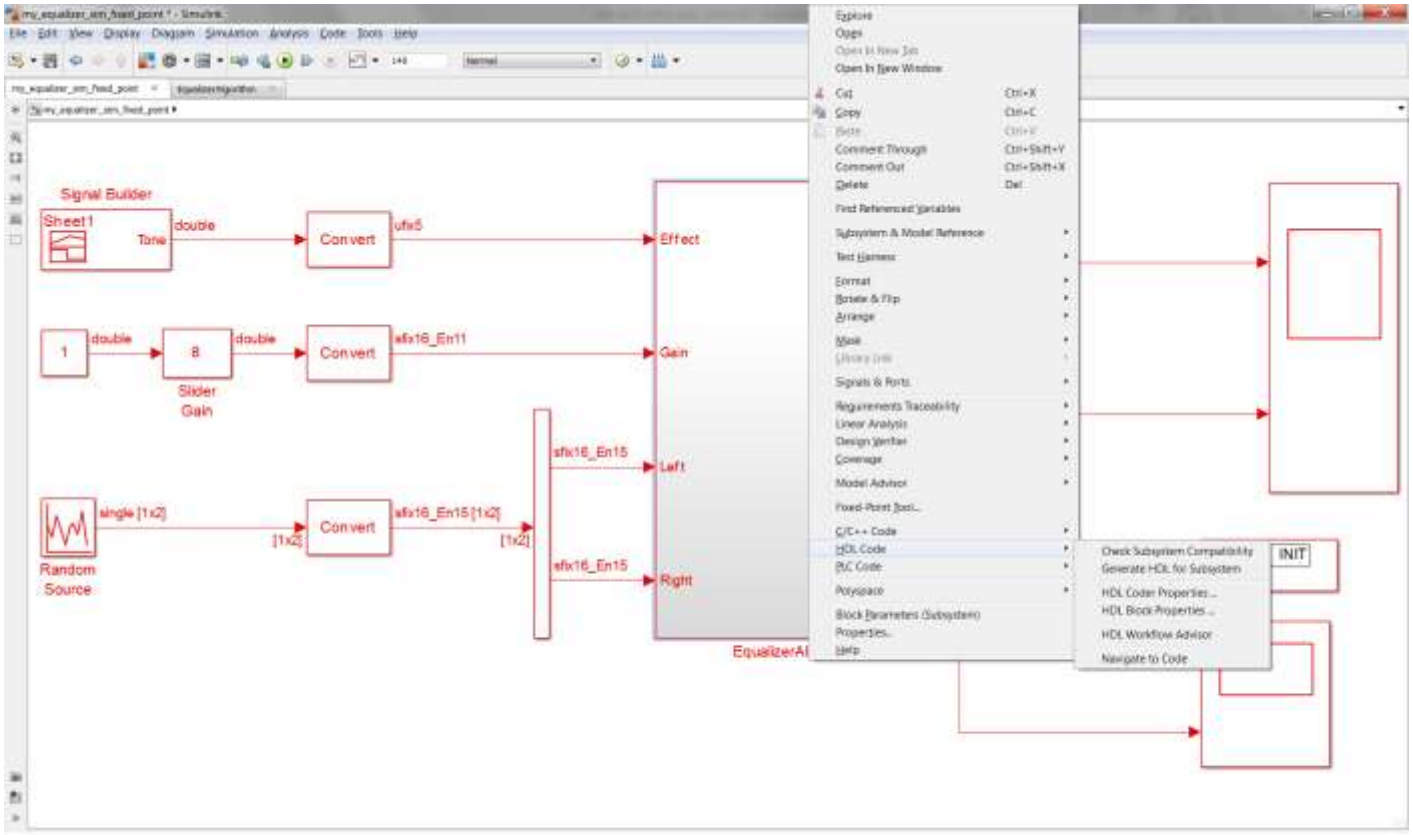
- Prepare model for code generation
- Generate HDL code
- Integrated RTL Verification
- Critical timing info in Simulink
- Area and Speed Optimization
- FPGA-in-Loop Verification

Fixed Point optimization at the system level



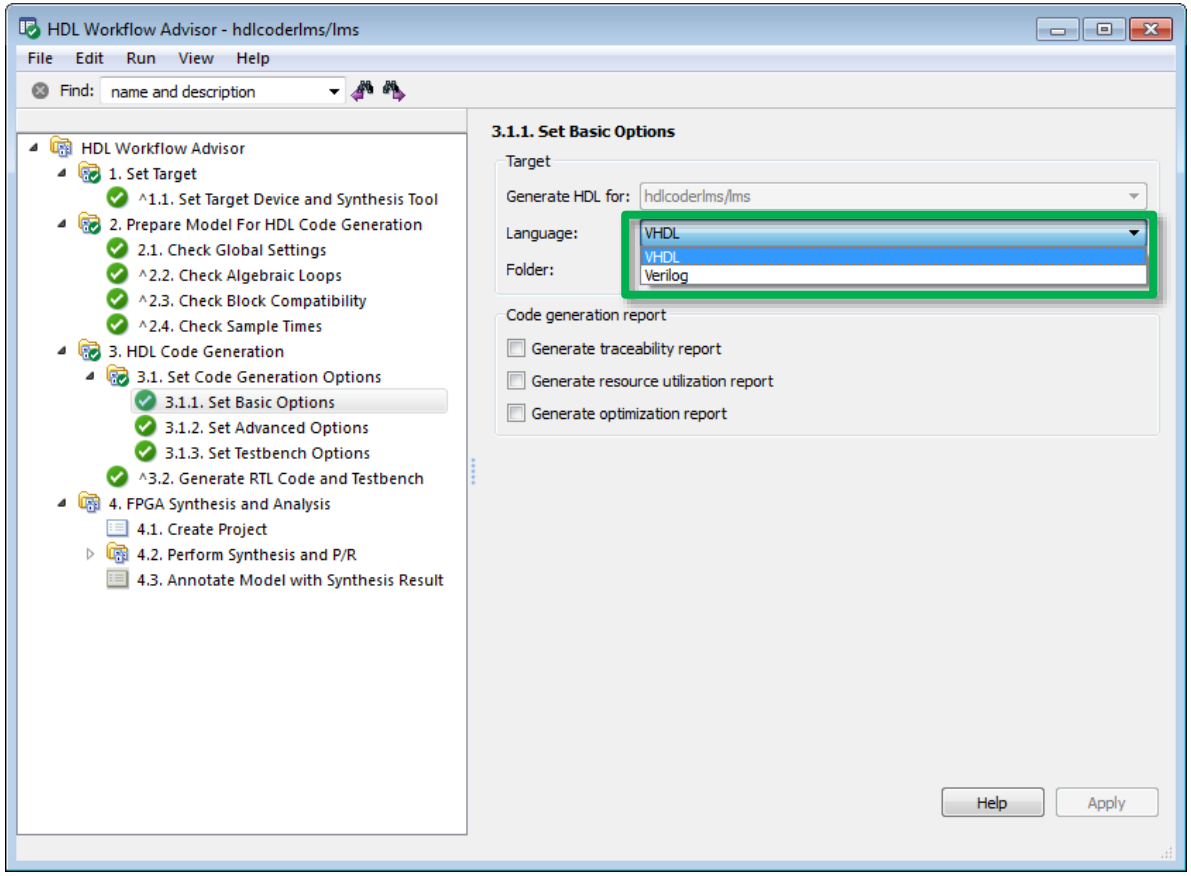
- Convert floating point to **optimized** fixed-point models
  - Automatic tracking of signal range (also intermediate quantities)
  - Word / Fraction lengths recommendation
- Bit-true models in the same environment

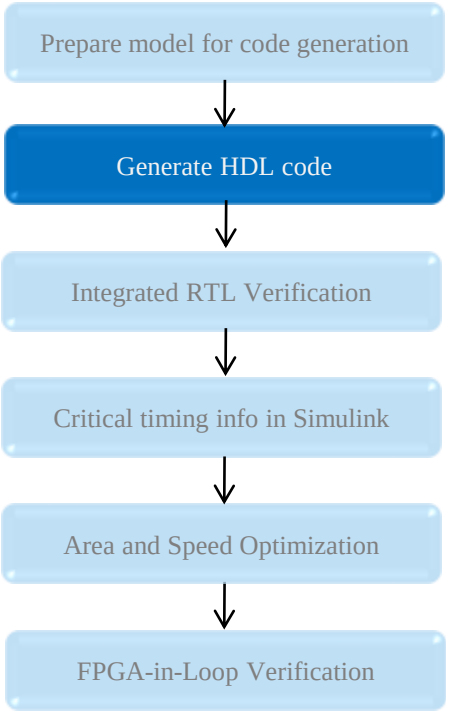
HDL Code Generation Example



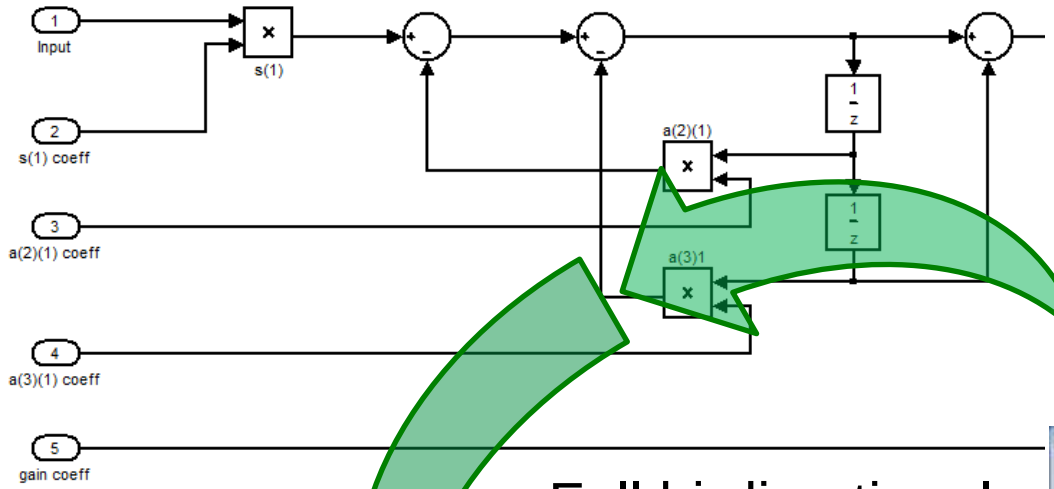
- Prepare model for code generation
- **Generate HDL code**
- Integrated RTL Verification
- Critical timing info in Simulink
- Area and Speed Optimization
- FPGA-in-Loop Verification

Generate Verilog And VHDL Code





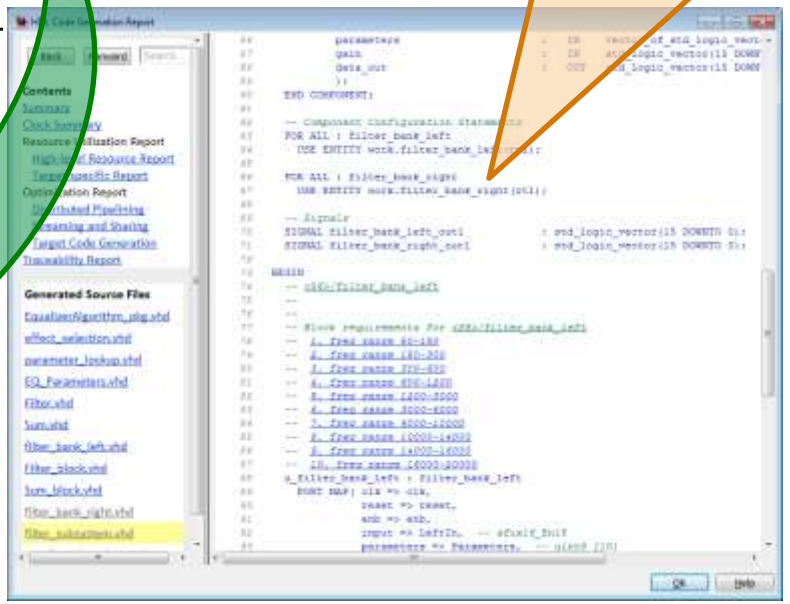
Automatically generate readable, traceable HDL code for FPGA and ASIC designs



Full bi-directional traceability!!

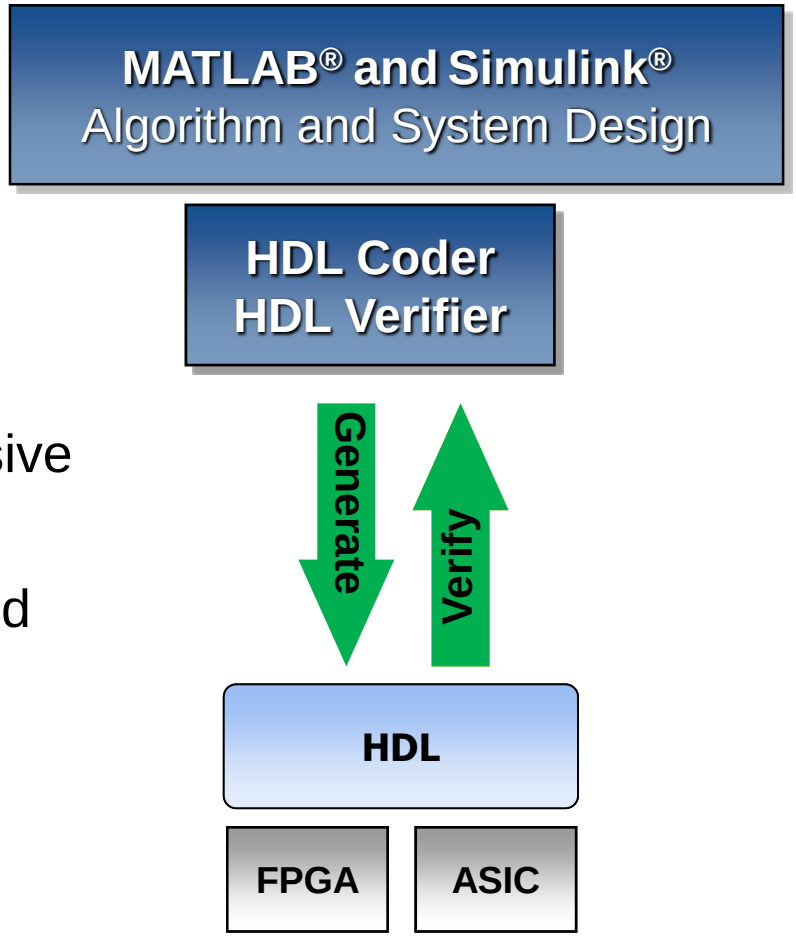
Requirements

Automatically generate bit true, cycle accurate HDL code from Simulink, MATLAB and Stateflow



HDL Coder Key Features

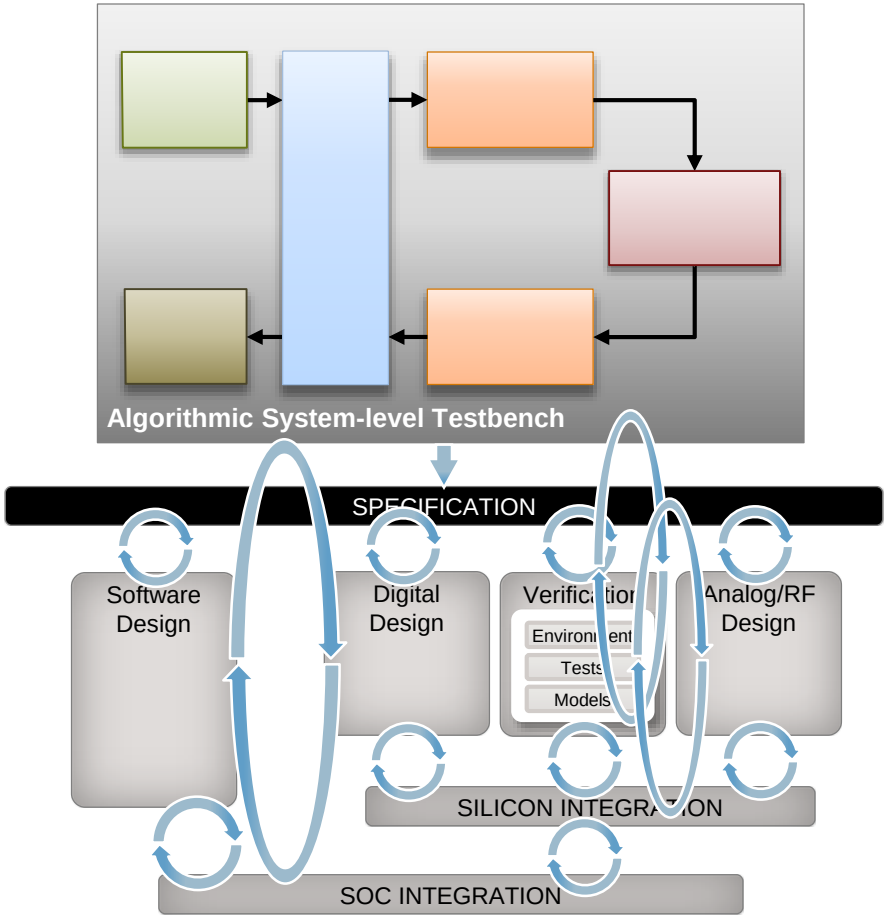
- Code Generation
  - Target-independent HDL Code
    - VHDL-1993 (IEEE® 1076-1993) or later
    - Verilog-2001 (IEEE 1364-2001) or later
- Verification
  - Generate HDL test-bench
  - Co-simulate with ModelSim and Incisive
- Design automation
  - Synthesize using integrated Xilinx and Altera synthesis tool interface
  - Optimize for area-speed
  - Program Xilinx and Altera boards



Agenda

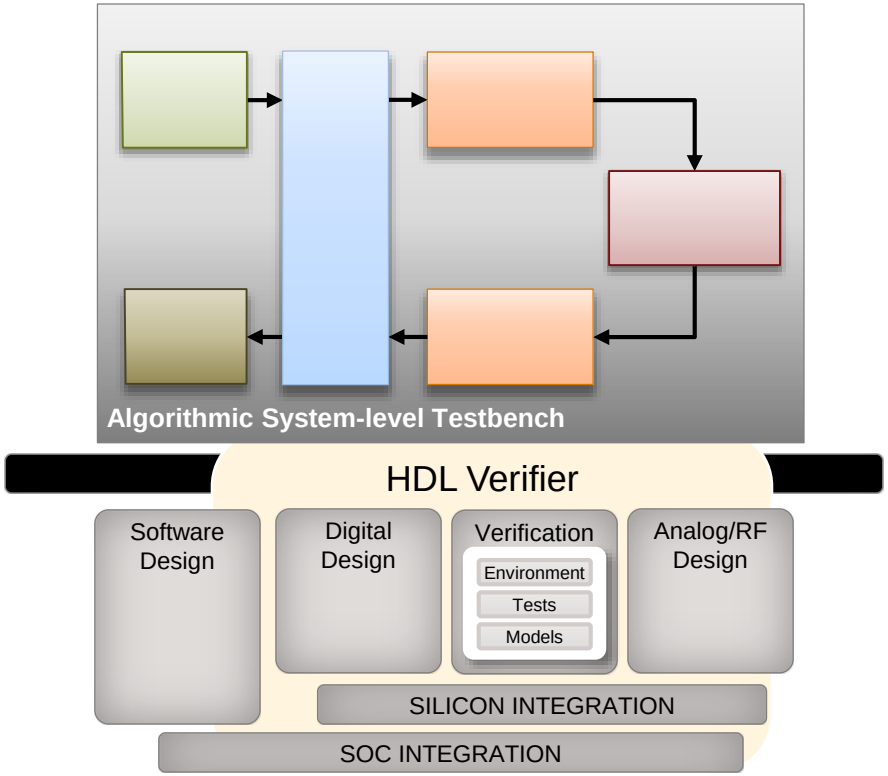
- HDL code generation from MATLAB, Simulink, and Stateflow®
- **Integrated RTL verification with EDA tools**
- Advanced Techniques to Optimize the Generated HDL
  - Workflow to explore Area and Speed Optimization
- FPGA-in-the-loop Verification
- Next Steps, Q & A

FPGA/ASIC Verification Challenges



- Spec is an over-the-wall handoff
- Success requires proper spec interpretation by everyone
  - Spec-related bugs are costly
  - How are changes handled?
- Hardware design and verification are bottom-up
  - Digital and analog isolated until silicon integration
  - Chip-level issues identified late
    - Difficult to fix and risky to the schedule
- Software isolated from hardware until SoC integration

HDL Verifier Connects Model-Based Design to FPGA and ASIC Verification



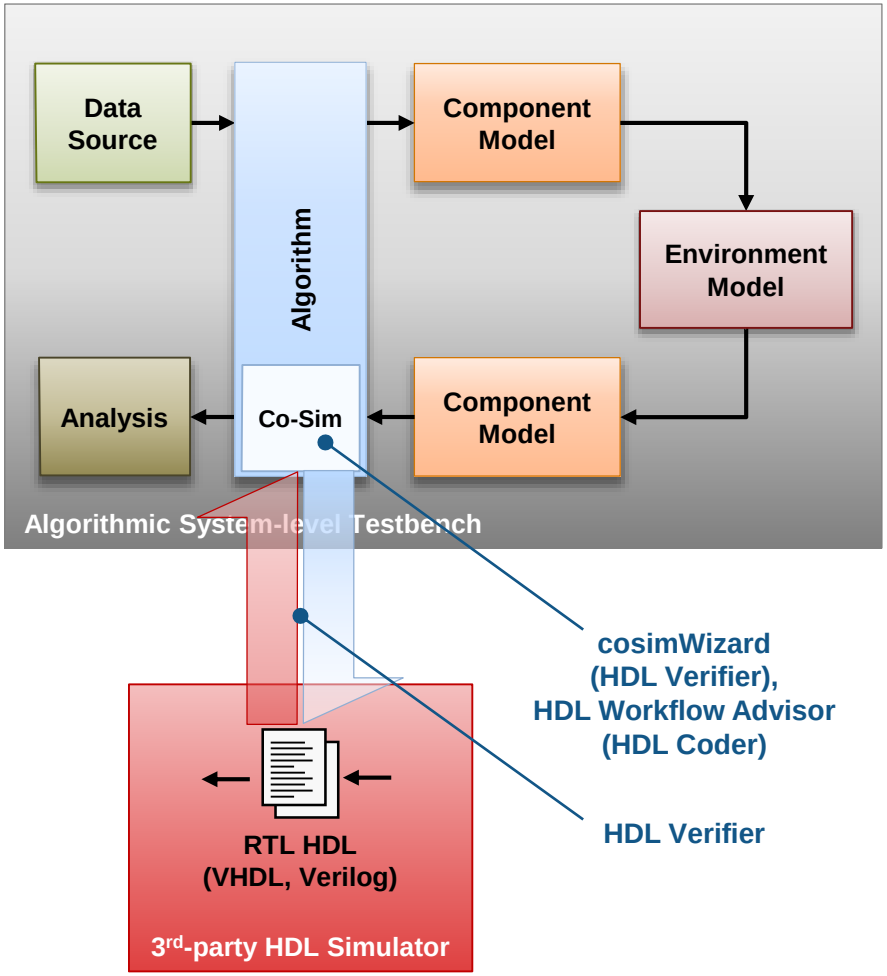
Model-Based Verification & Validation

- Trace requirements
- Check modeling standards compliance
- Analyze coverage

- Generate DPI-C models for SystemVerilog simulation
- Generate SystemC TLM-2.0 models for virtual platform

- Co-simulate MATLAB/Simulink system with 3rd-party simulator
- FPGA-in-the-loop with MATLAB/Simulink

Co-simulation for Verification of HDL Code

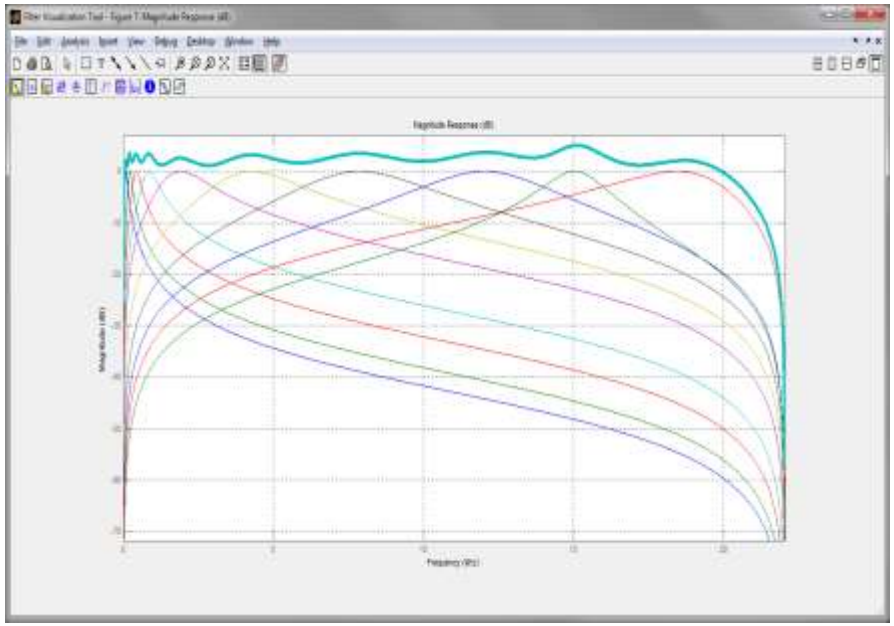


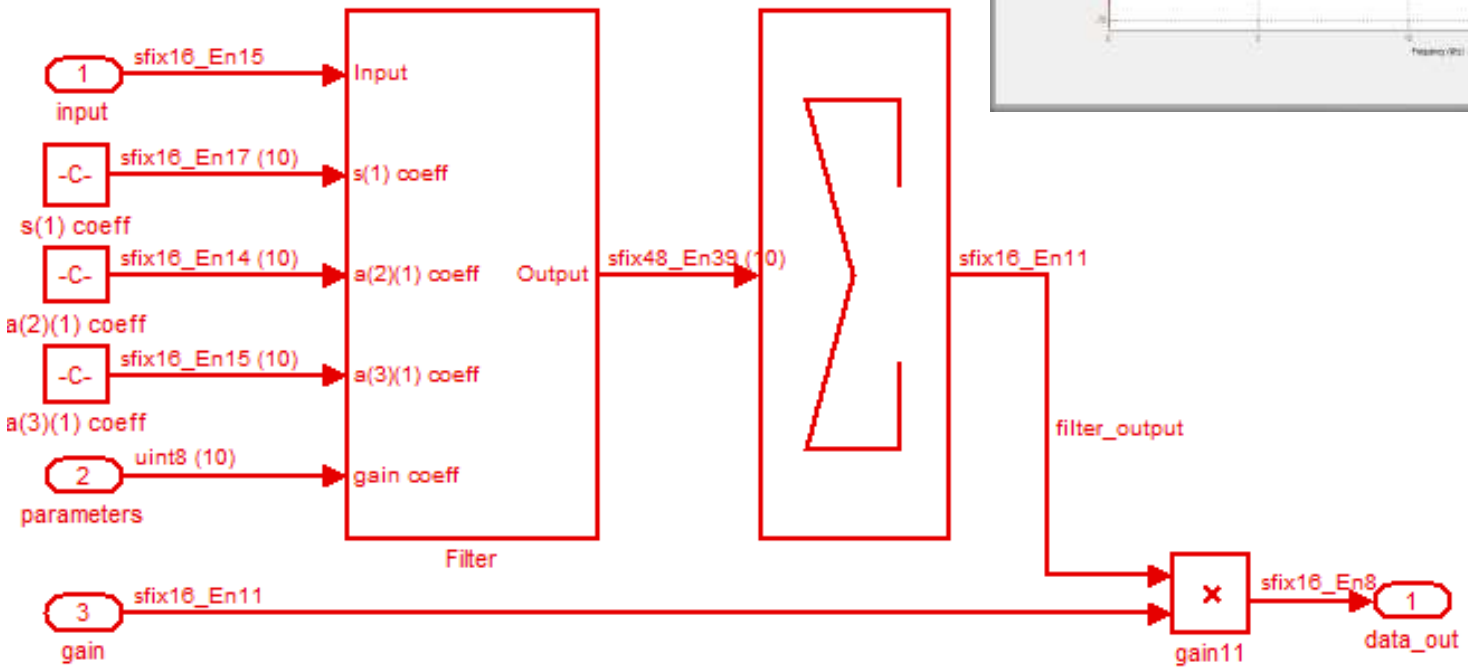
Verify your HDL using your system-level test environment

- Co-simulation with 3rd-party HDL simulator
  - Reuse existing system-level testbench
  - HDL code execution in 3rd-party HDL simulator
  - Flexible HDL sources
    - Automatic HDL code
    - Manual / Legacy HDL code
  - Automated generation of co-simulation infrastructure
  - Automatic handshaking
    - Combined analysis and debugging in both simulators

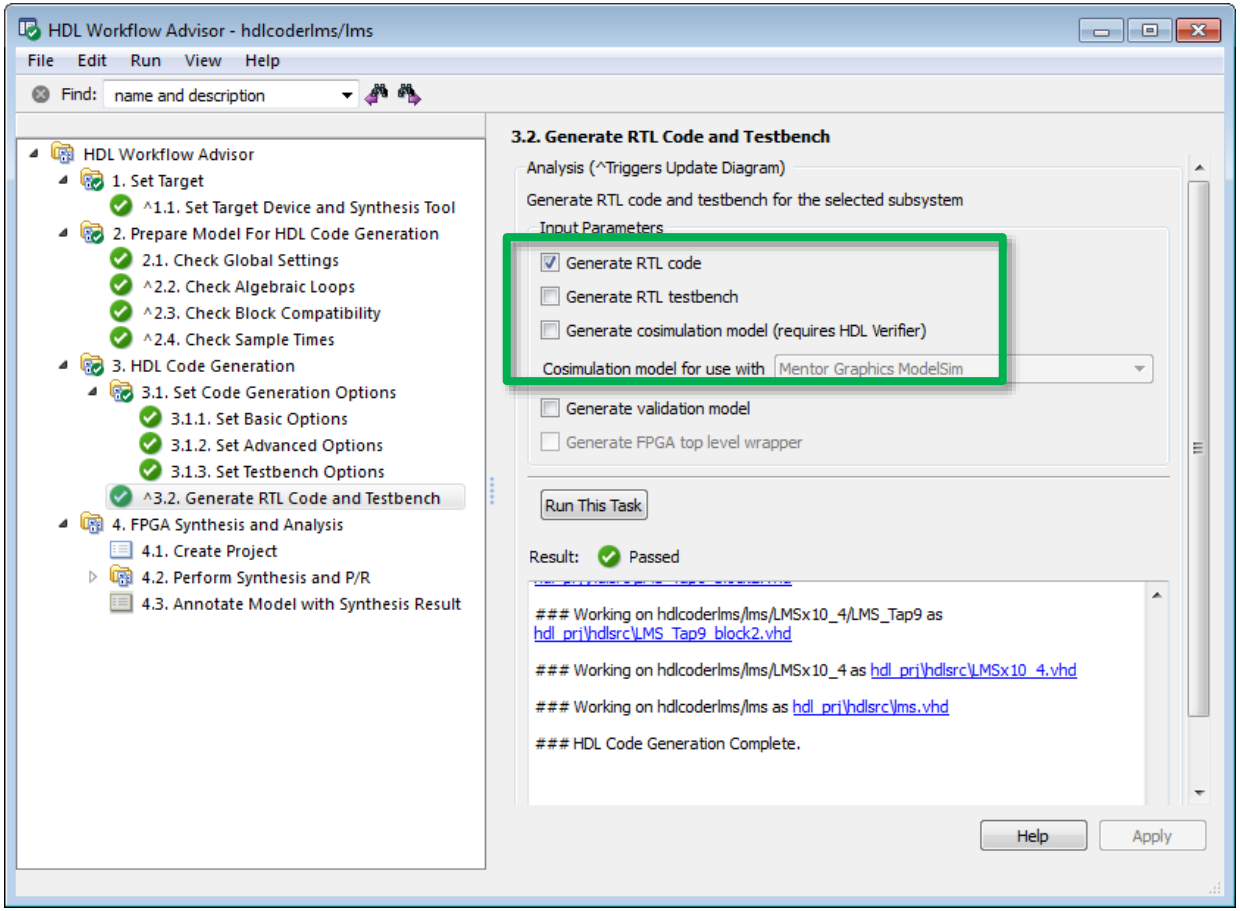
Audio Equalizer

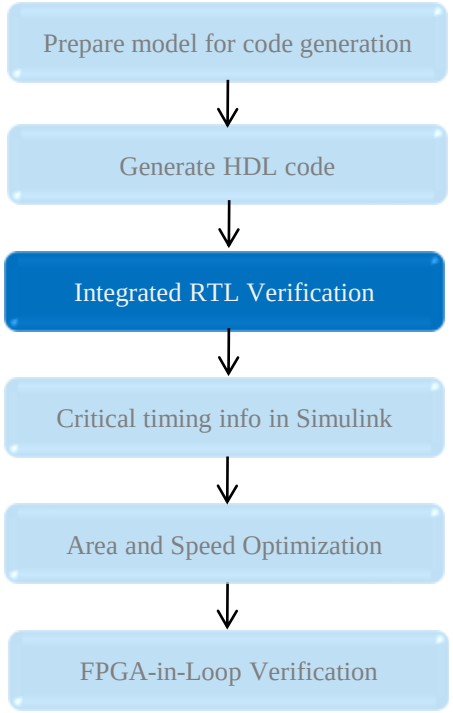
- **Bank of 10 filters**
  - Controllable by up to +/-6dB
- **5 pre-programmed user settings for**
  - Rock, Pop, Jazz, Classical, Vocal
- **Fits into available FPGA space**
- **No dead-locks or unreachable states**
- **Sounds good**





Automate RTL Verification





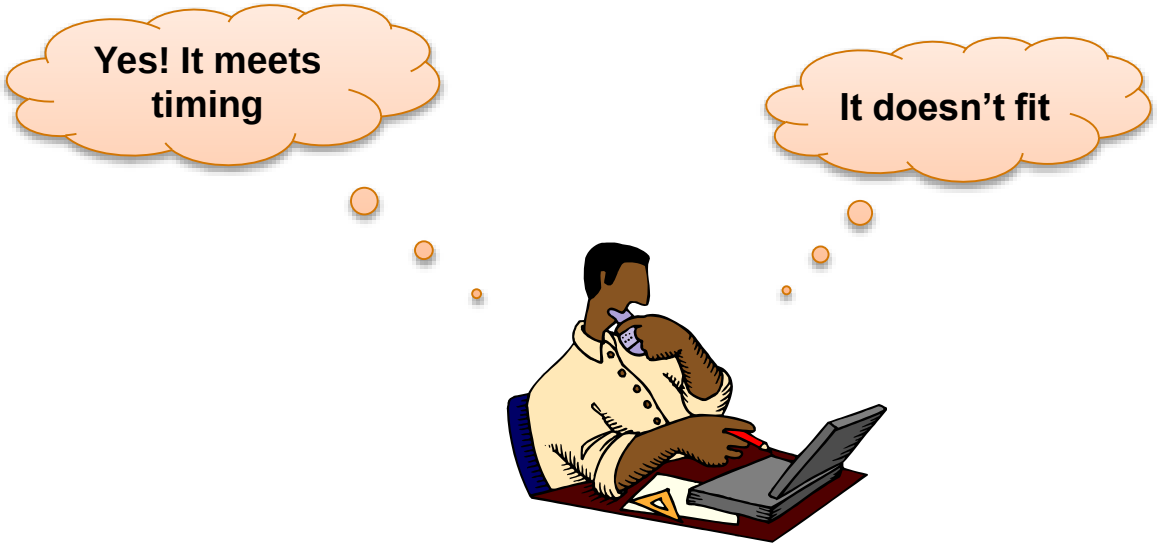
Agenda

- HDL code generation from MATLAB, Simulink, and Stateflow®
- Integrated RTL verification with EDA tools
- **Advanced Techniques to Optimize the Generated HDL**
  - Workflow to explore Area and Speed Optimization
- FPGA-in-the-loop Verification
- Next Steps, Q & A

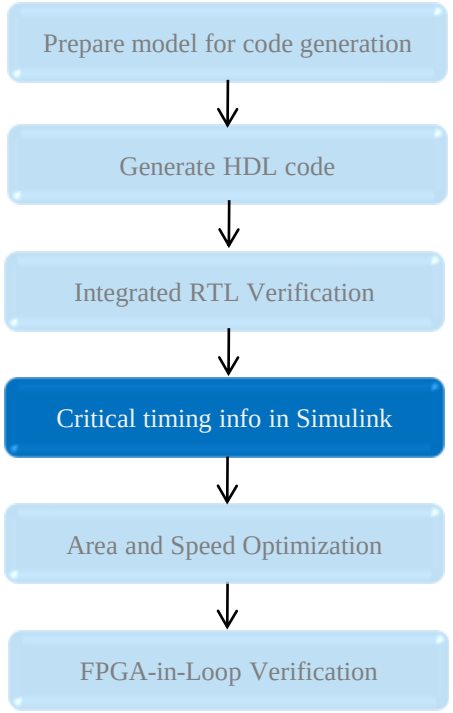
HDL Optimizations: What, How and Why

The three golden questions:

1. Speed: Does it meet timing?
2. Area: Does it fit on my FPGA?
3. Validation: Does it do the right thing?

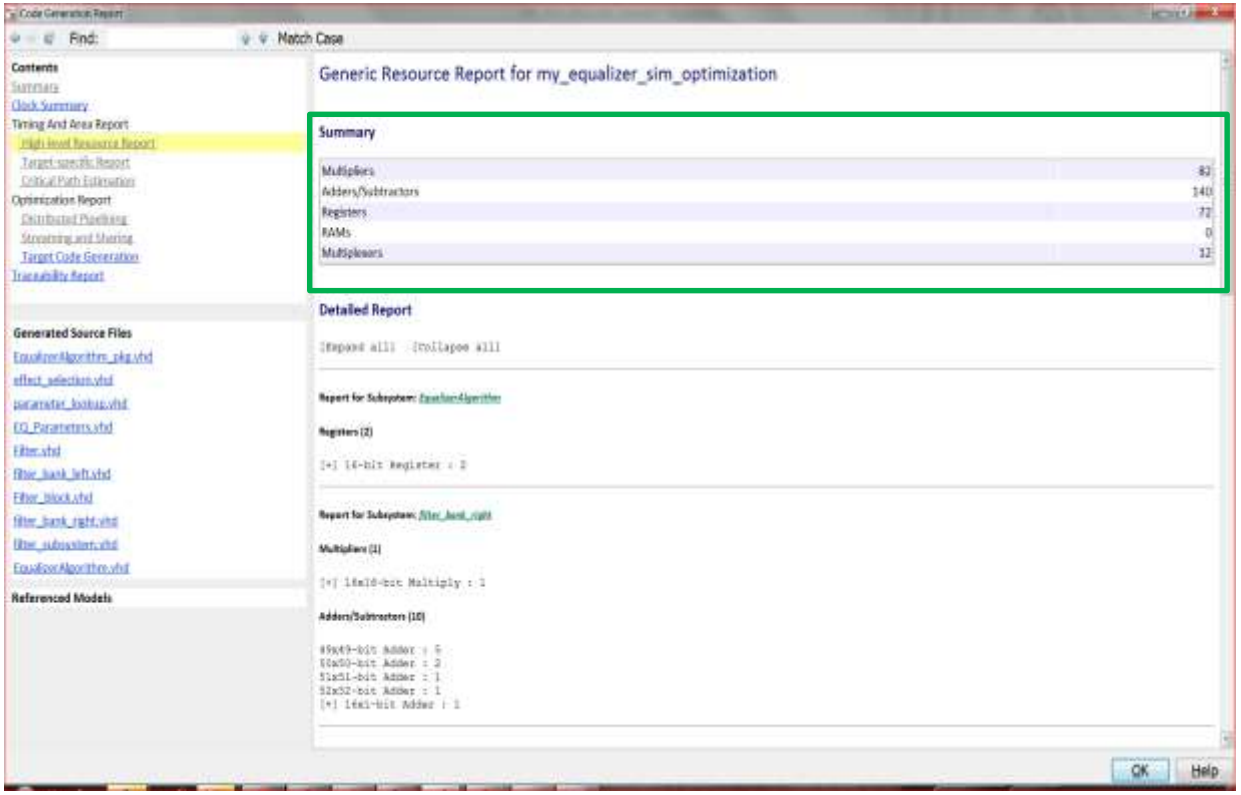


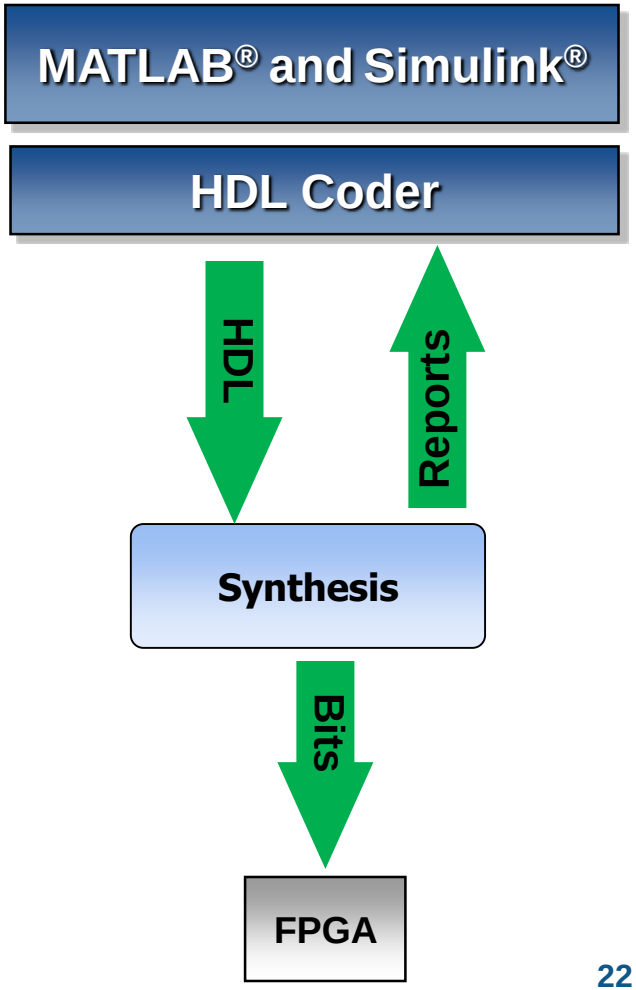
FPGA Engineer



HDL optimizations assists the engineer in meeting these constraints

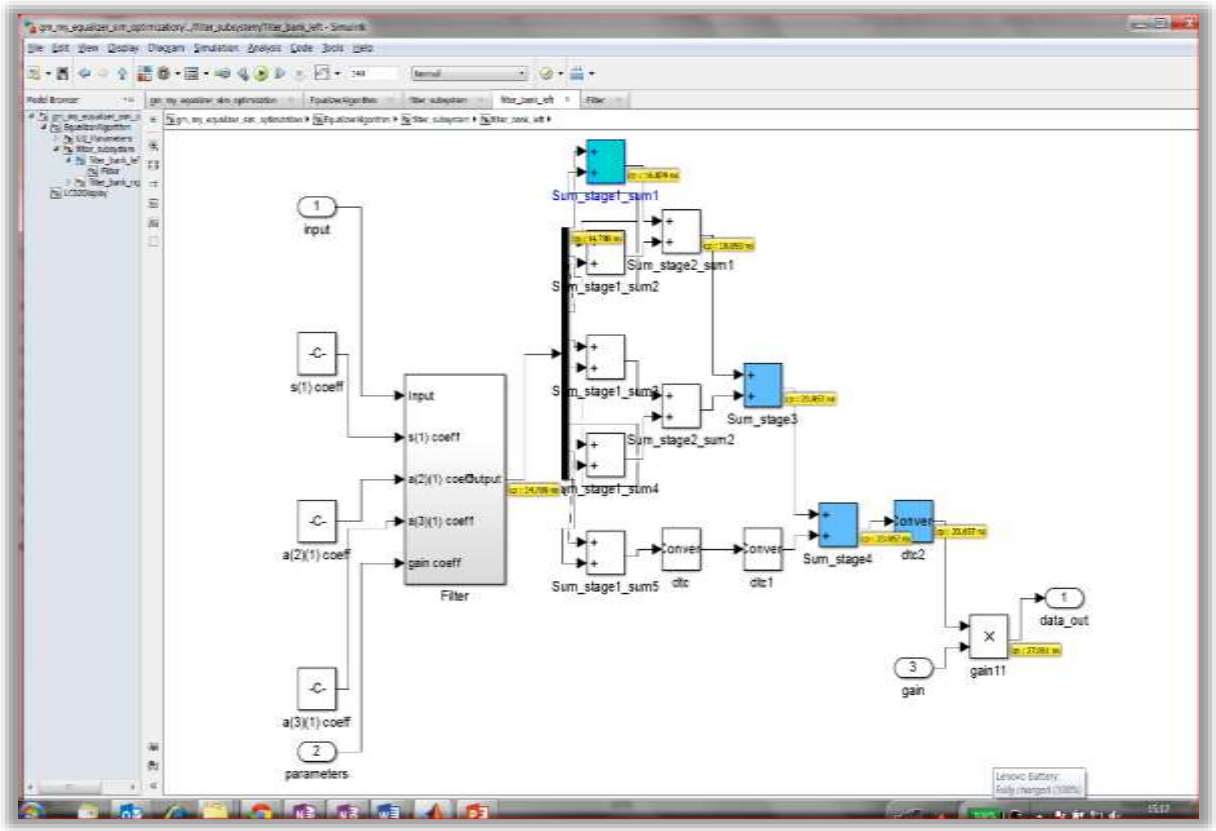
Resource Utilization Report





- ✓ Resource utilization report
- ✓ Helps you estimate design area

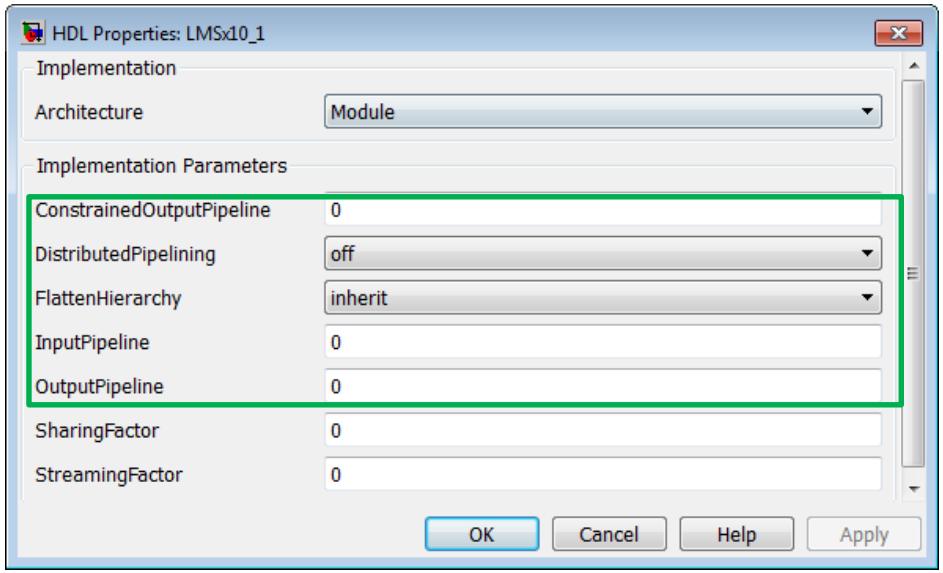
Critical Timing Path

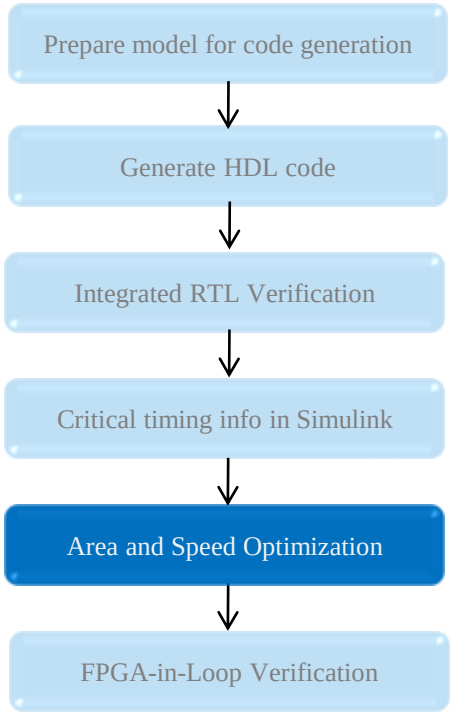


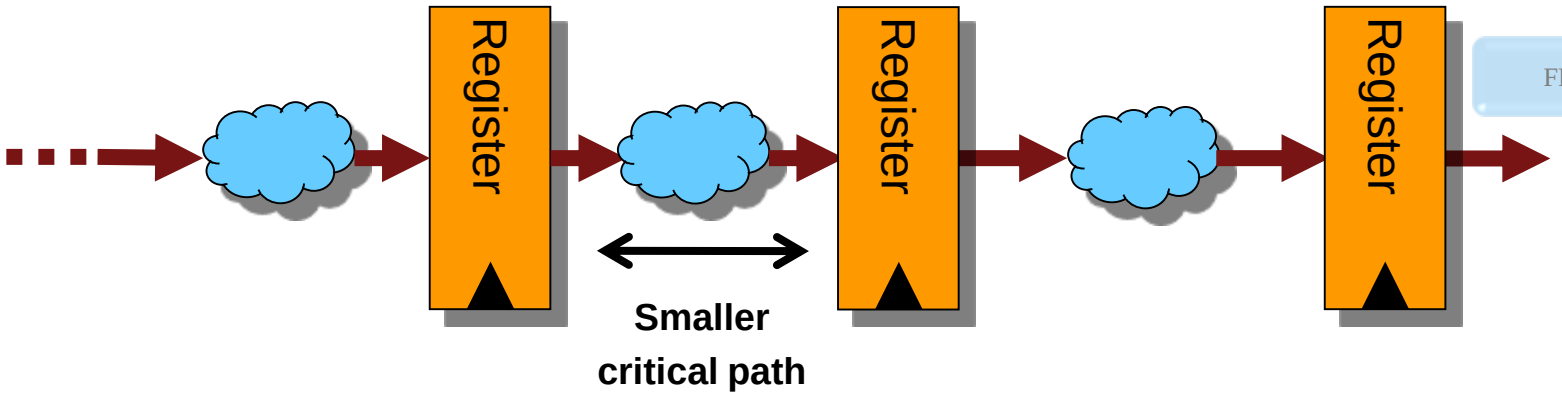
- Prepare model for code generation
- Generate HDL code
- Integrated RTL Verification
- **Critical timing info in Simulink**
- Area and Speed Optimization
- FPGA-in-Loop Verification

✓ **Critical path highlighting**
✓ **Helps you identify speed bottlenecks**

Optimize Design For Speed And Area

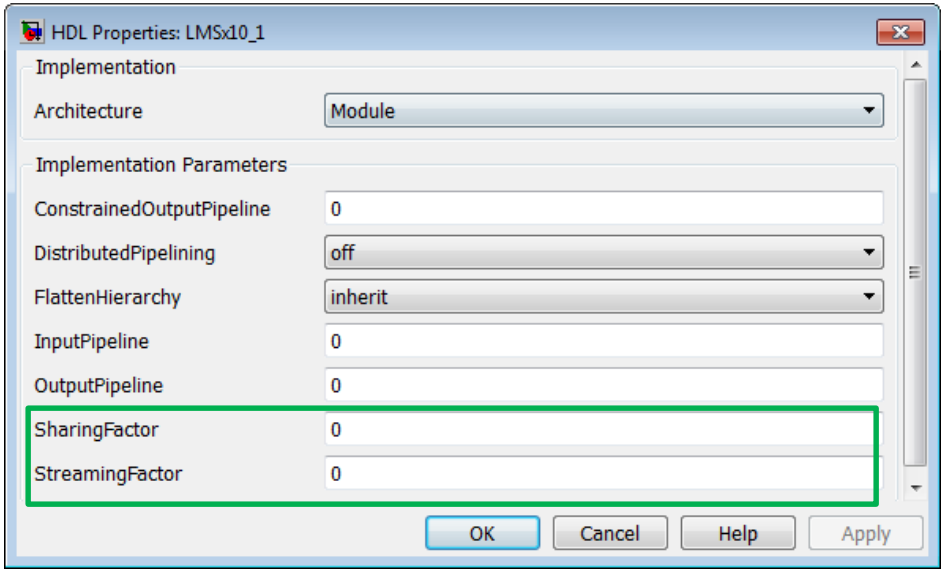


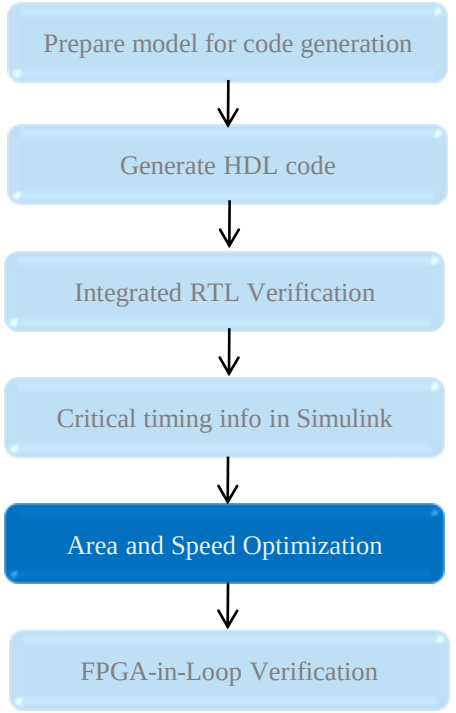




- ✓ **Automatic pipelining**
- ✓ **Helps you meet speed objectives**

Optimize Design For Speed And Area



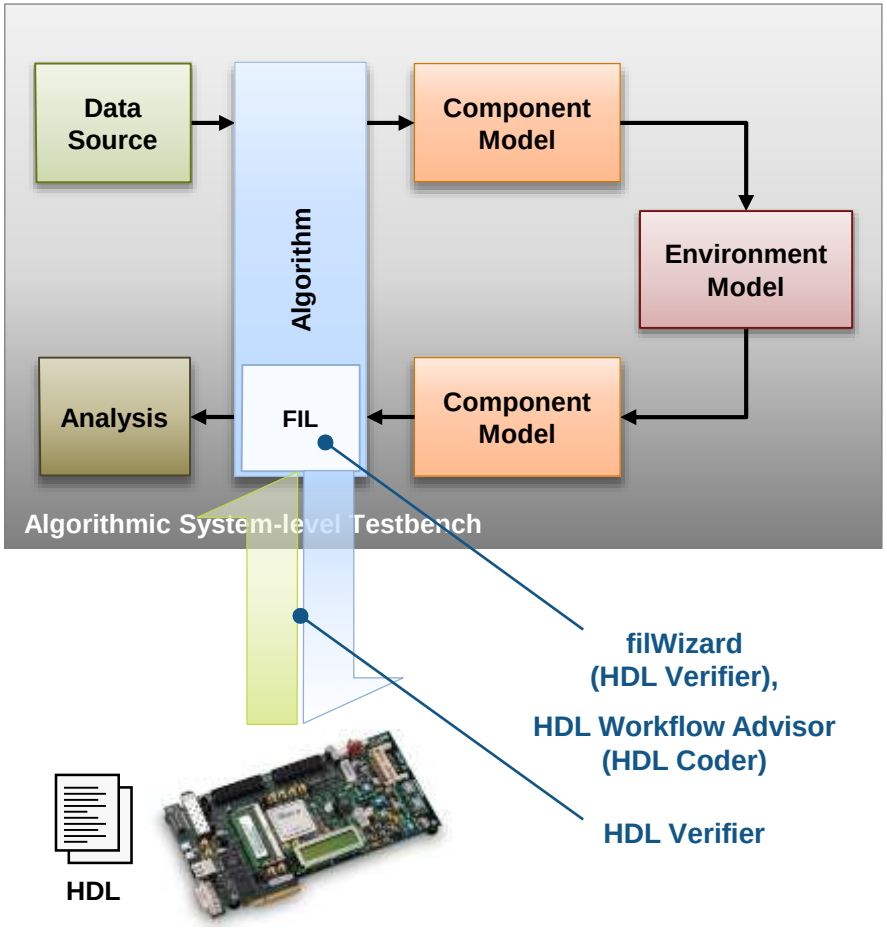


- ✓ **Streaming and resource sharing**
- ✓ **Helps you do speed-area optimization**

Agenda

- HDL code generation from MATLAB, Simulink, and Stateflow®
- Integrated RTL verification with EDA tools
- Advanced Techniques to Optimize the Generated HDL
  - Workflow to explore Area and Speed Optimization
- **FPGA-in-the-loop Verification**
- Next Steps, Q & A

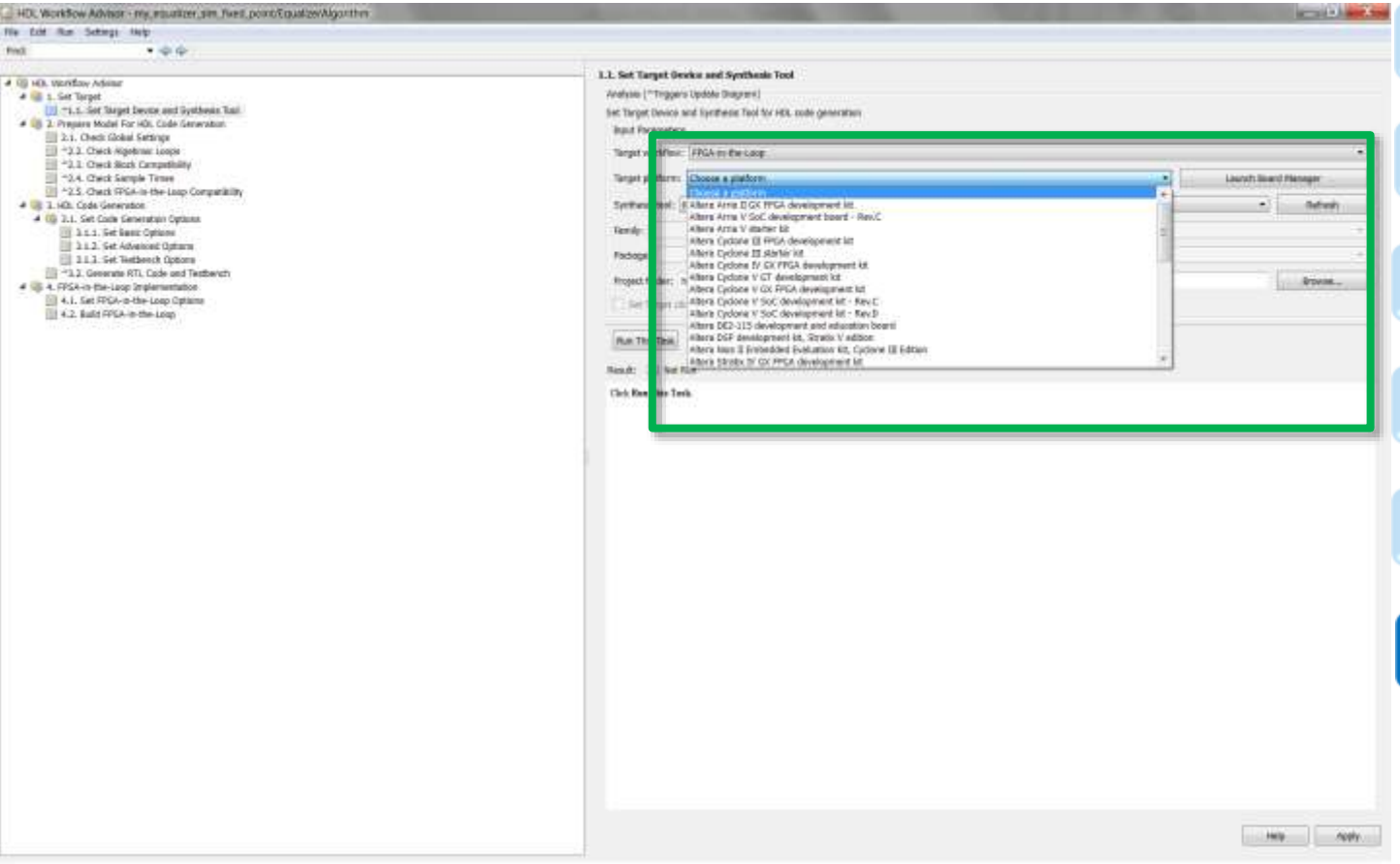
FPGA-in-the-Loop Verification of HDL



Prototype your algorithm in hardware connected to the system-level test environment

- FIL simulation with FPGA development board
  - Reuse existing testbench
  - HDL code execution on FPGA
  - Handwritten or generated HDL code
  - Automated generation of co-simulation infrastructure
    - Encapsulation of algorithm within GBit Ethernet MAC, or JTAG
  - Automatic handshaking

FPGA-in-the-Loop Verification of HDL



Prepare model for code generation

↓

Generate HDL code

↓

Integrated RTL Verification

↓

Critical timing info in Simulink

↓

Area and Speed Optimization

↓

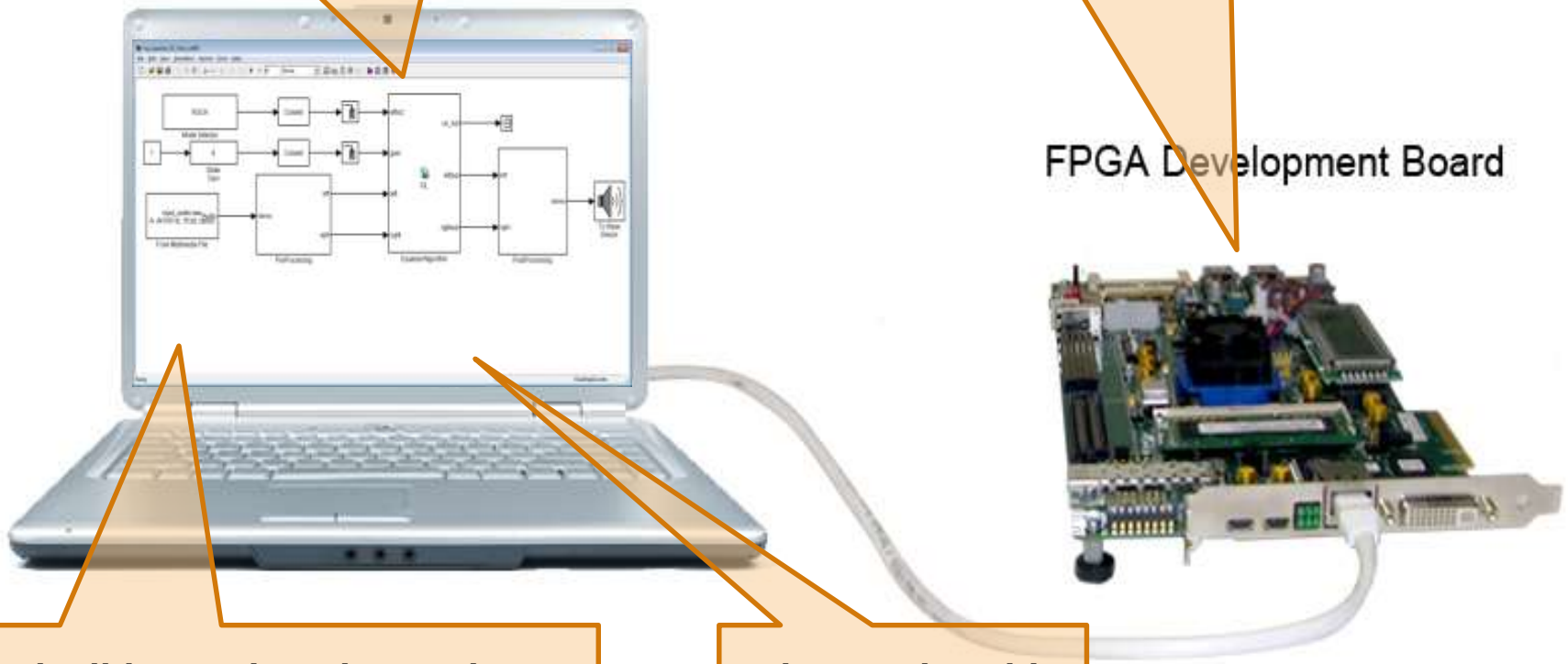
FPGA-in-Loop Verification

FPGA-in-the-loop

Enable regression testing with FPGA-in-the-loop simulation

Re-use test benches for regression testing

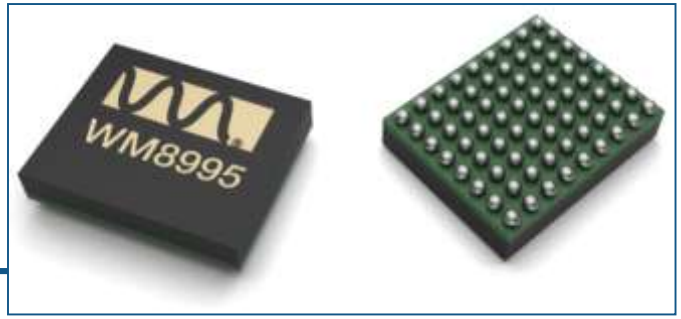
Integrate with Altera / Xilinx FPGA Development Boards



Flexible test bench creation: closed loop, multi domain

Also works with handwritten code

Wolfson Microelectronics Accelerates Audio Hub Design Verification



Wolfson Microelectronics digital audio hub.

Challenge

Develop a multipath, multichannel audio hub for smartphones

Solution

Use Simulink to model and simulate the DSP design and use HDL Coder to generate bit-true Verilog models for verification of the digital implementation

Results

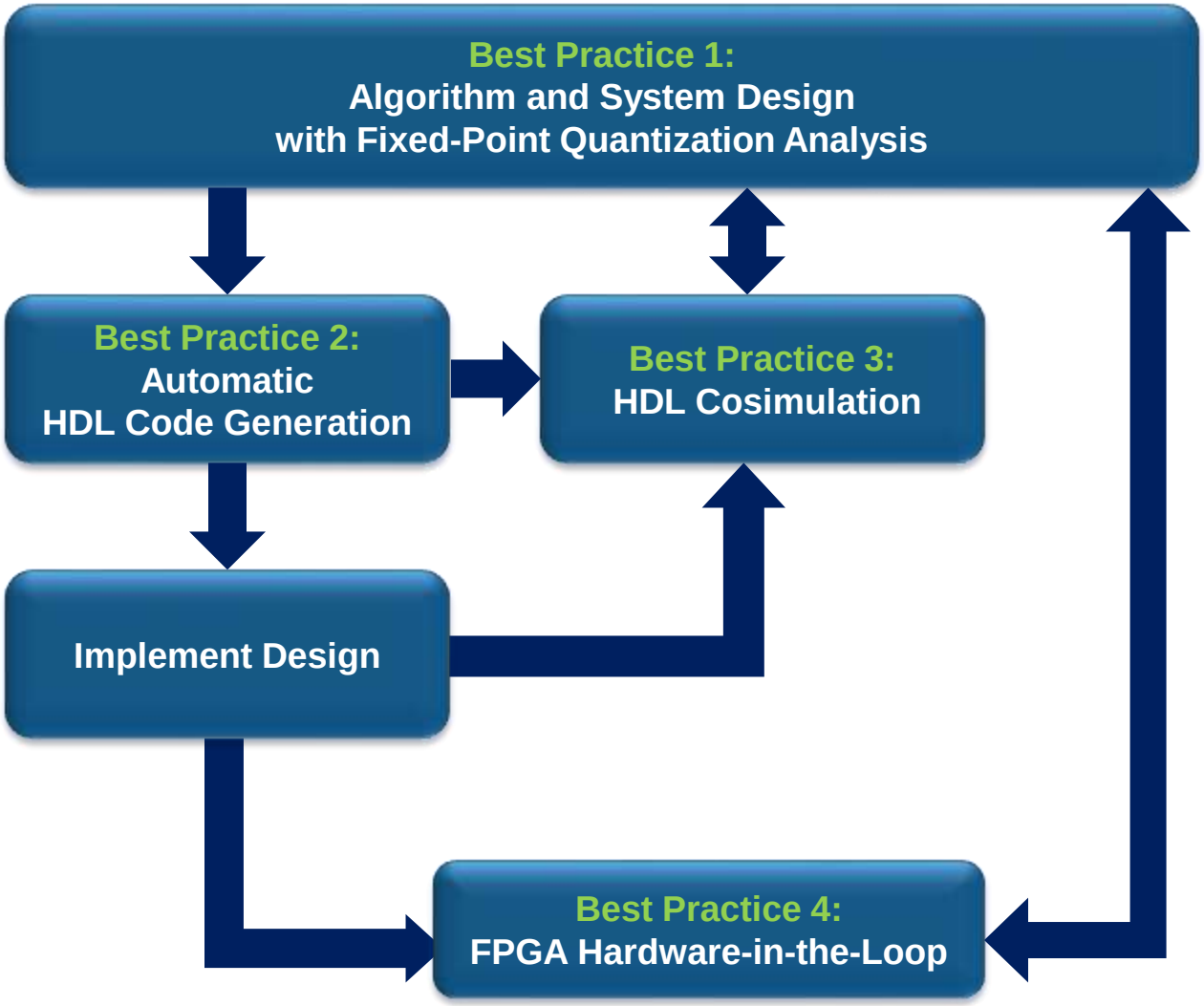
- Months of hand-coding eliminated
- Datapath verification coverage increased to 100%
- Debugging process accelerated by 20%

"For development of the world's first highly optimized digital audio hub solution, Simulink and HDL Coder were the best options. The design and verification flow we applied using MathWorks tools scales well and provides the route to build more complex DSP and signal mixing paths."

Brian Paisley
Wolfson Microelectronics

Link to user story

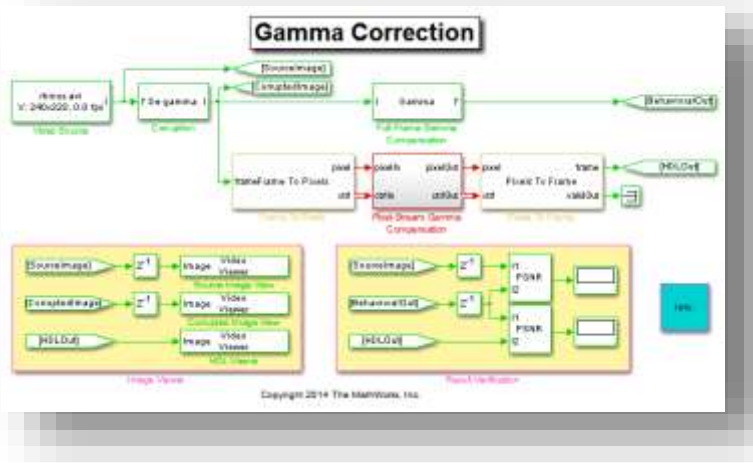
Key Takeaway



Vision HDL Toolbox

Design and prototype video/image processing systems

- Modeling hardware behavior of the algorithms
  - Pixel-based functions and blocks
  - Conversion between frames and pixels
  - Standard and custom frame sizes



- Prototyping algorithms on hardware
  - (**With HDL Coder**) Efficient and readable HDL code
  - (**With HDL Verifier**) FPGA-in-the-loop testing and acceleration

Agenda

- HDL code generation from MATLAB, Simulink, and Stateflow®
- Integrated RTL verification with EDA tools
- Advanced Techniques to Optimize the Generated HDL
  - Workflow to explore Area and Speed Optimization
- FPGA-in-the-loop Verification
- **Next Steps, Q & A**

MathWorks Training

Self-Paced Courses	Dates	Location
Signal Processing with MATLAB	16 June	Bangalore
Image Processing with MATLAB	18 June	Bangalore
Generating HDL Code from Simulink	7th – 8th Sep	Bangalore
Programming Xilinx Zynq SoCs with MATLAB and Simulink	10th – 11th Sep	Bangalore
MathWorks Certification Exams	**Dates**	**Location**
MathWorks Certified MATLAB Associate Exam	29 July	Bangalore

New

Domain Specific Trainings

Signal Processing and Communication

- Signal Processing with MATLAB
- Signal Processing with Simulink
- Image Processing with MATLAB
- Generating HDL Code from Simulink
- Communication Systems Design with MATLAB
- Communication Systems Modeling with Simulink
- Programming Xilinx Zynq SoCs with MATLAB and Simulink

**Training courses are also available on MATLAB based Optimization and Statistics techniques*

Modeling, Simulation and Control

- Simulink for System and Algorithm Modeling
- Stateflow for Logic Driven System Modeling
- MATLAB and Simulink for Control Design Acceleration
- Fundamentals of Code Generation for Real-Time Design and Testing.
- Physical Modeling of Multi-Domain Systems using Simscape
- Physical Modeling of Mechanical Systems with SimMechanics
- Physical Modeling of Electric Power Systems with SimPowerSystems

Contact MathWorks India

URL: http://www.mathworks.in

E-mail: info@mathworks.in

Technical Support: www.mathworks.in/myservicerequests

Tel: +91-80-6632 6000

Fax: +91-80-6632 6010

- **MathWorks India Private Limited**
  9th Floor, 'B' Wing, Etamin Block
  Prestige Technology Park II
  Marathahalli – Sarjapur Ring Road
  Bangalore – 560103, Karnataka
  India

Thank You for Attending

Talk to Us – We are Happy to Support You