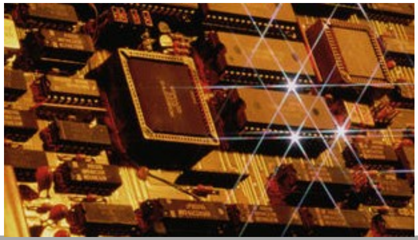
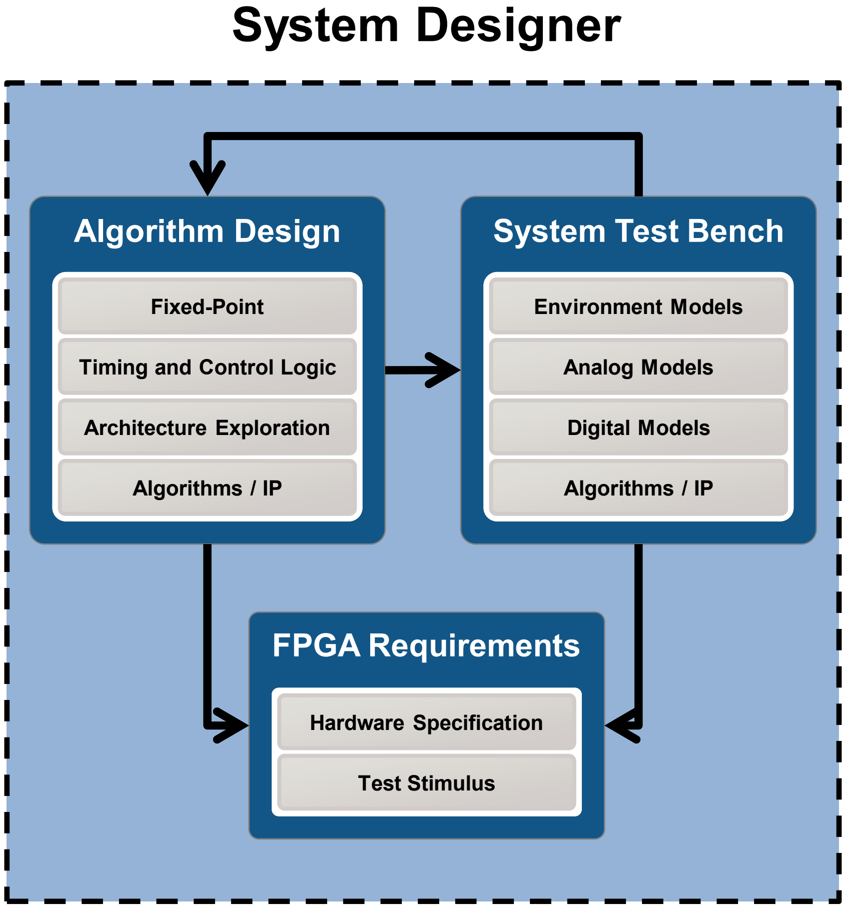
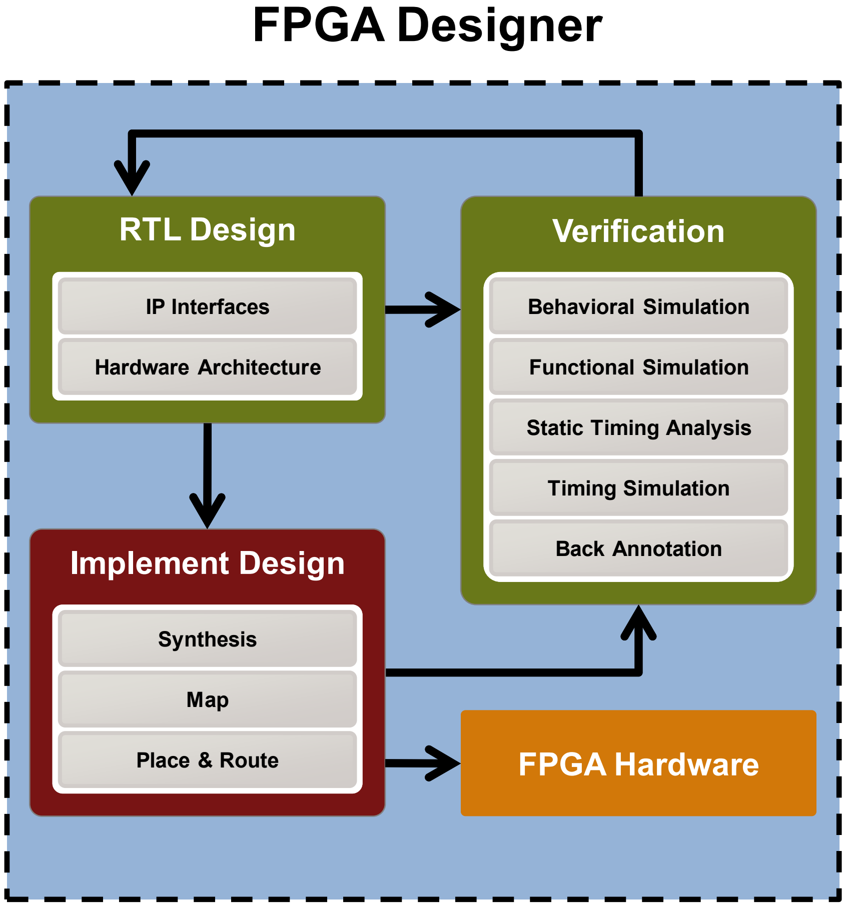
Model-Based Design for Altera FPGAs Using HDL Code Generation



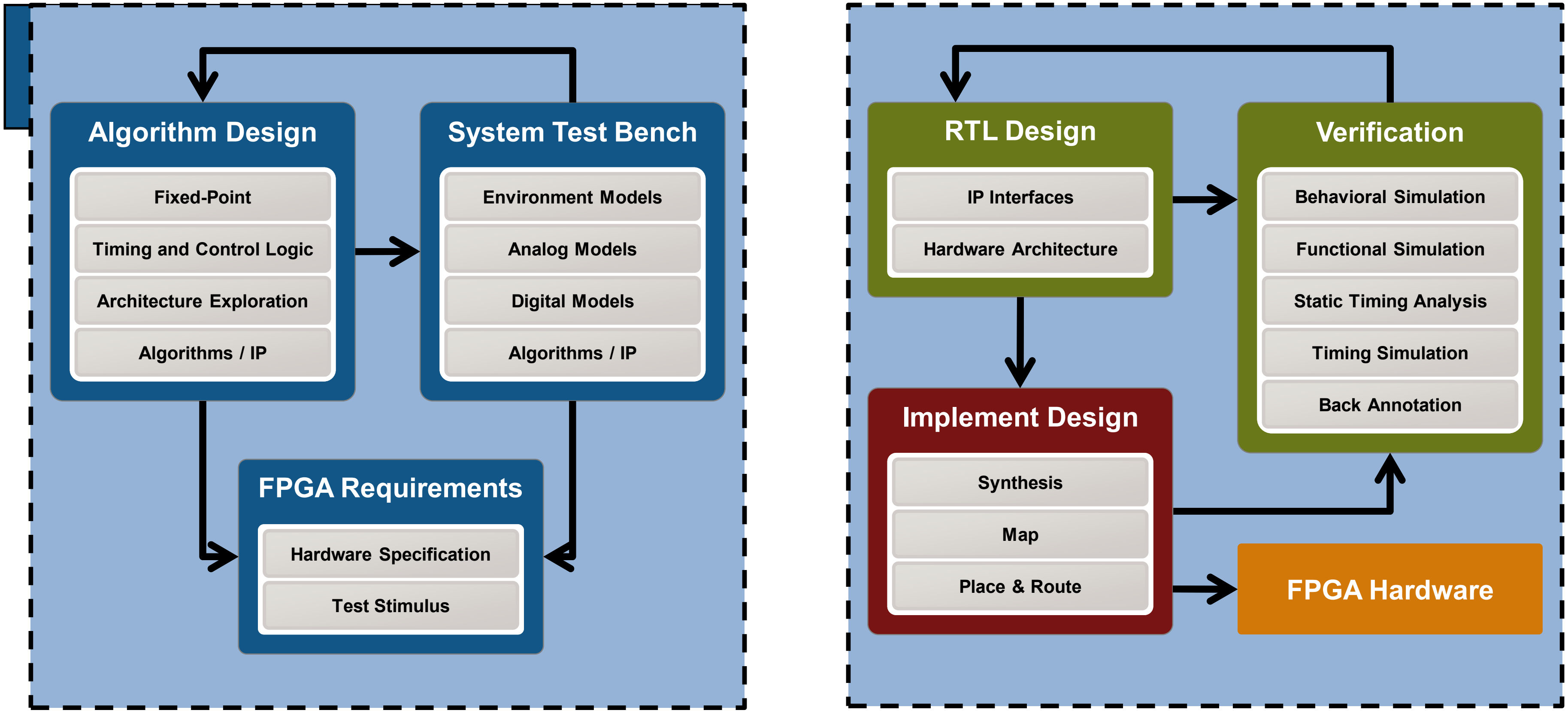
z

Separate Views of DSP Implementation





Model Based Design for Implementation



Algorithm Design

- Fixed-Point
- Timing and Control Logic
- Architecture Exploration
- Algorithms / IP

System Test Bench

- Environment Models
- Analog Models
- Digital Models
- Algorithms / IP

FPGA Requirements

- Hardware Specification
- Test Stimulus

RTL Design

- IP Interfaces
- Hardware Architecture

Verification

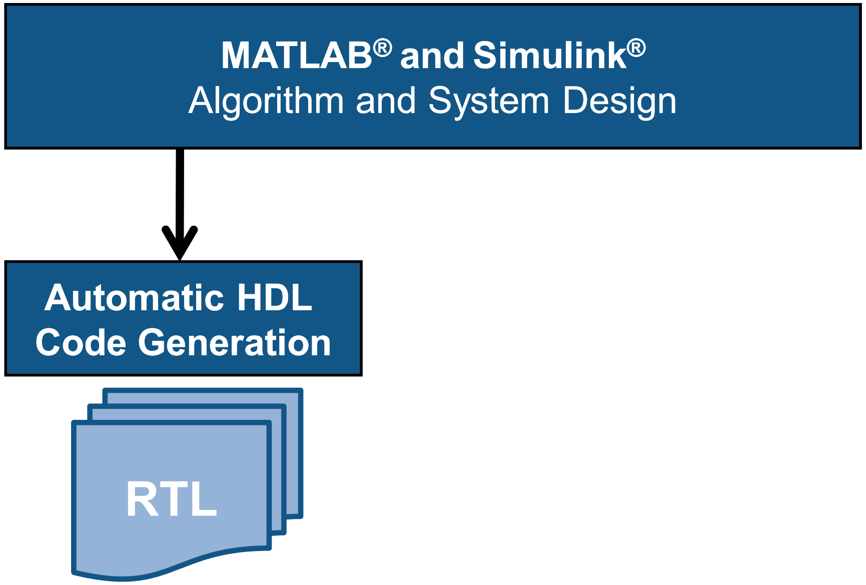
- Behavioral Simulation
- Functional Simulation
- Static Timing Analysis
- Timing Simulation
- Back Annotation

Implement Design

- Synthesis
- Map
- Place & Route

FPGA Hardware

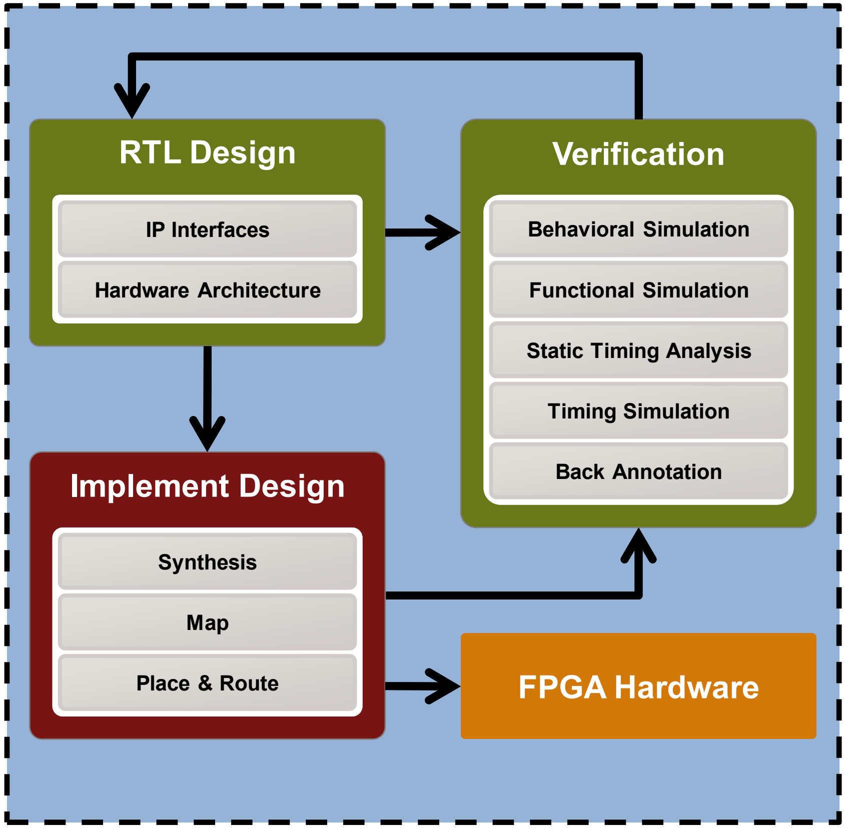
Model Based Design for Implementation



MATLAB® and Simulink®
Algorithm and System Design

Automatic HDL Code Generation

RTL



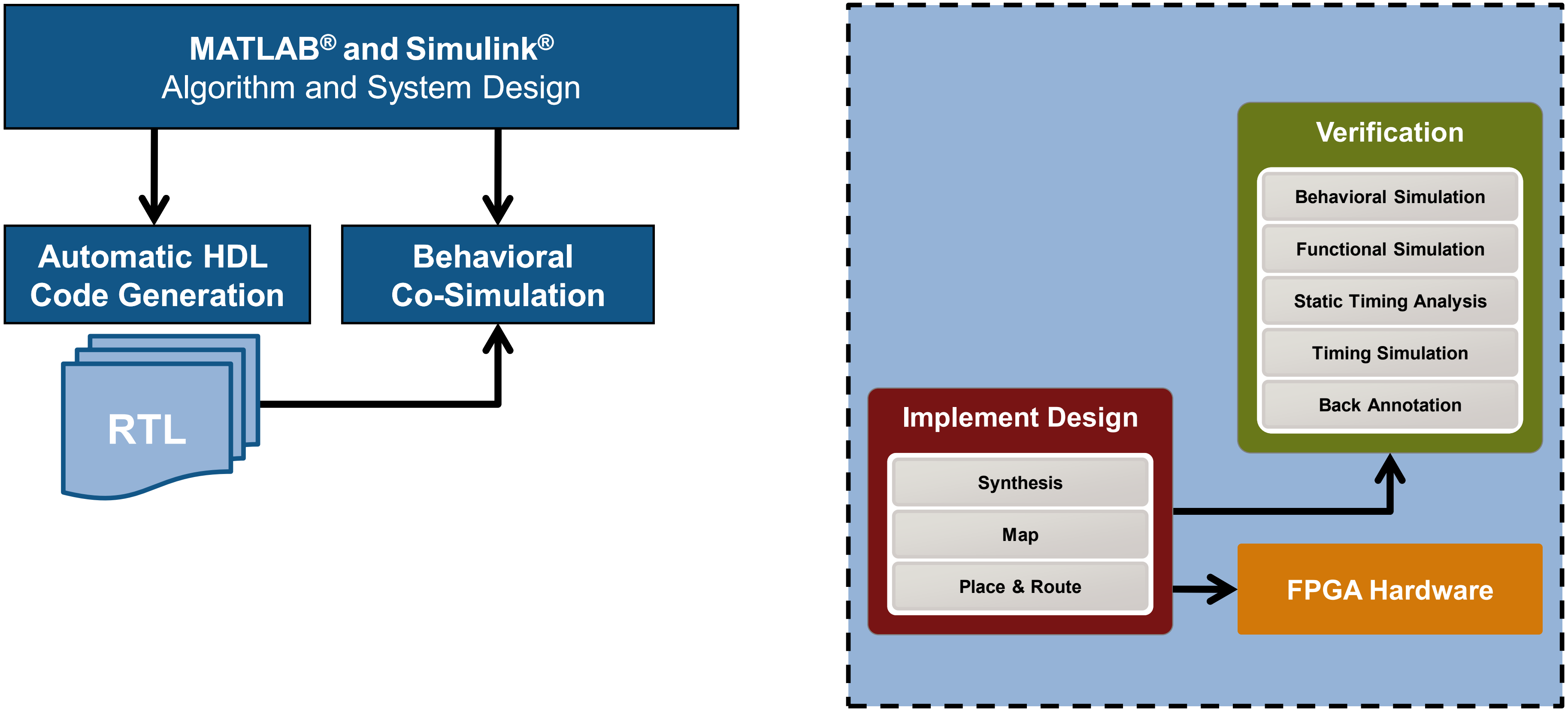
RTL Design
- IP Interfaces
- Hardware Architecture

Verification
- Behavioral Simulation
- Functional Simulation
- Static Timing Analysis
- Timing Simulation
- Back Annotation

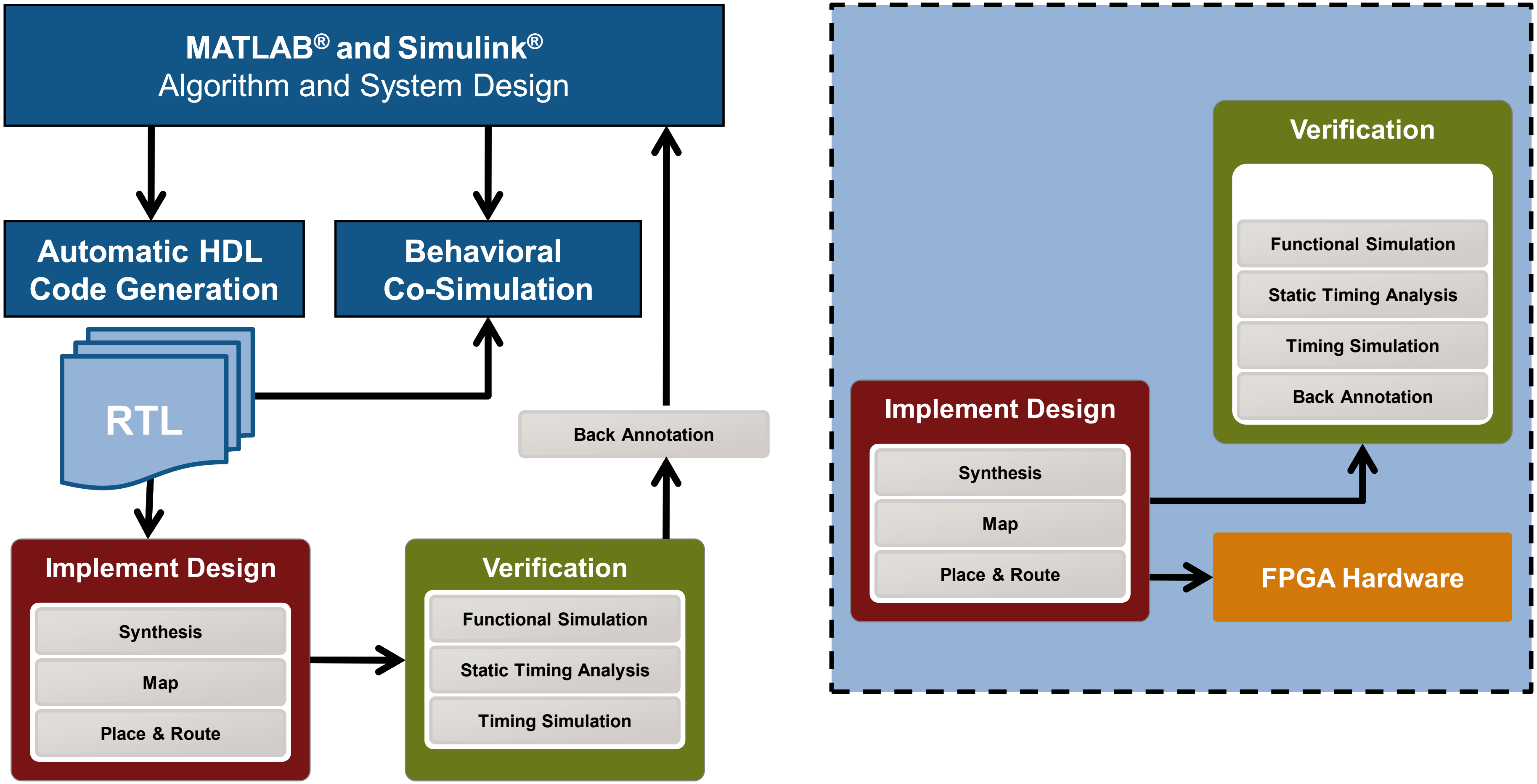
Implement Design
- Synthesis
- Map
- Place & Route

FPGA Hardware

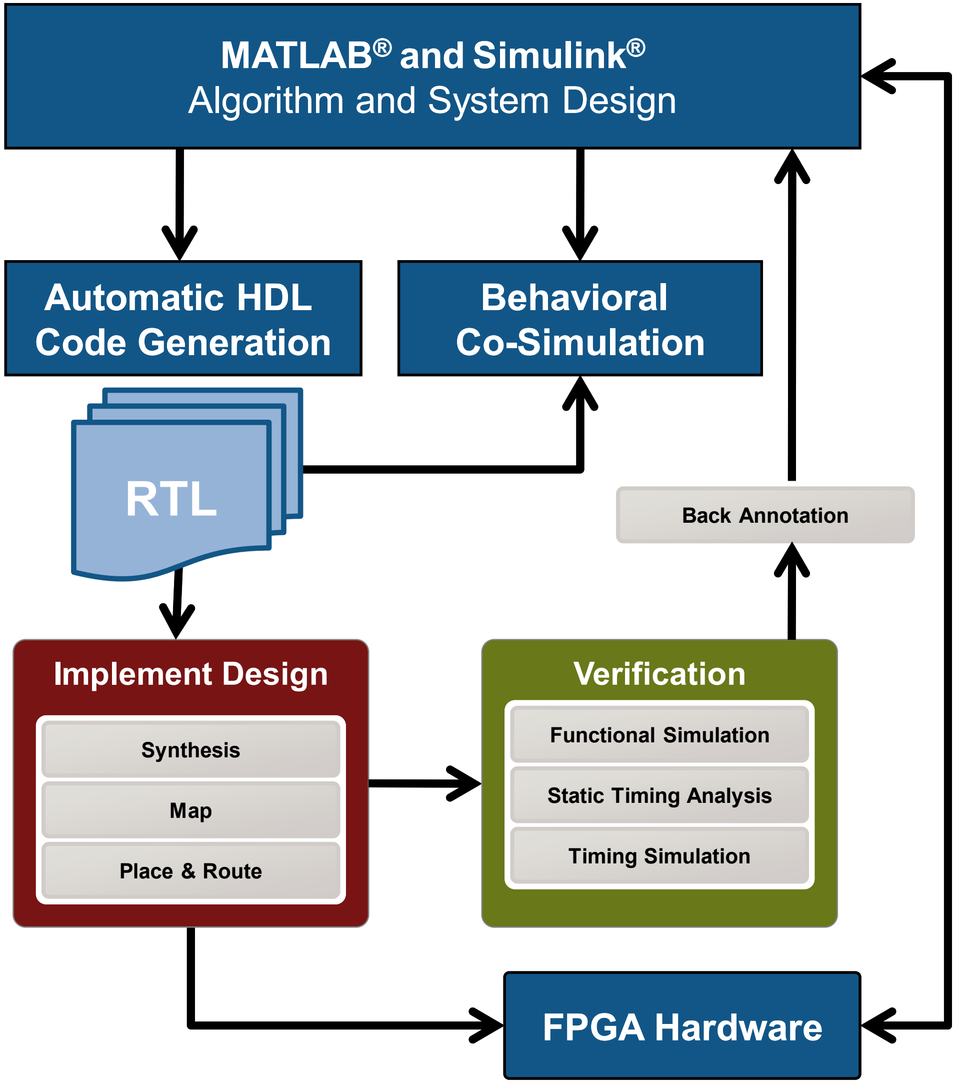
Model Based Design for Implementation



Model Based Design for Implementation

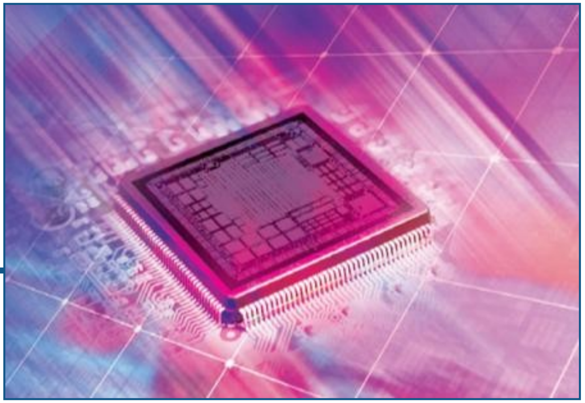


Model Based Design for Implementation



FPGA Hardware

Faraday Accelerates SIP Development and Shrinks NAND Flash Controller ECC Engine Gate Count by 57%



Faraday's silicon IP on an SoC.

Challenge

Accelerate the development of SoCs and ASICs

Solution

Use MathWorks tools for Model-Based Design to speed up system-level simulations, improve system performance, and shorten time-to-market

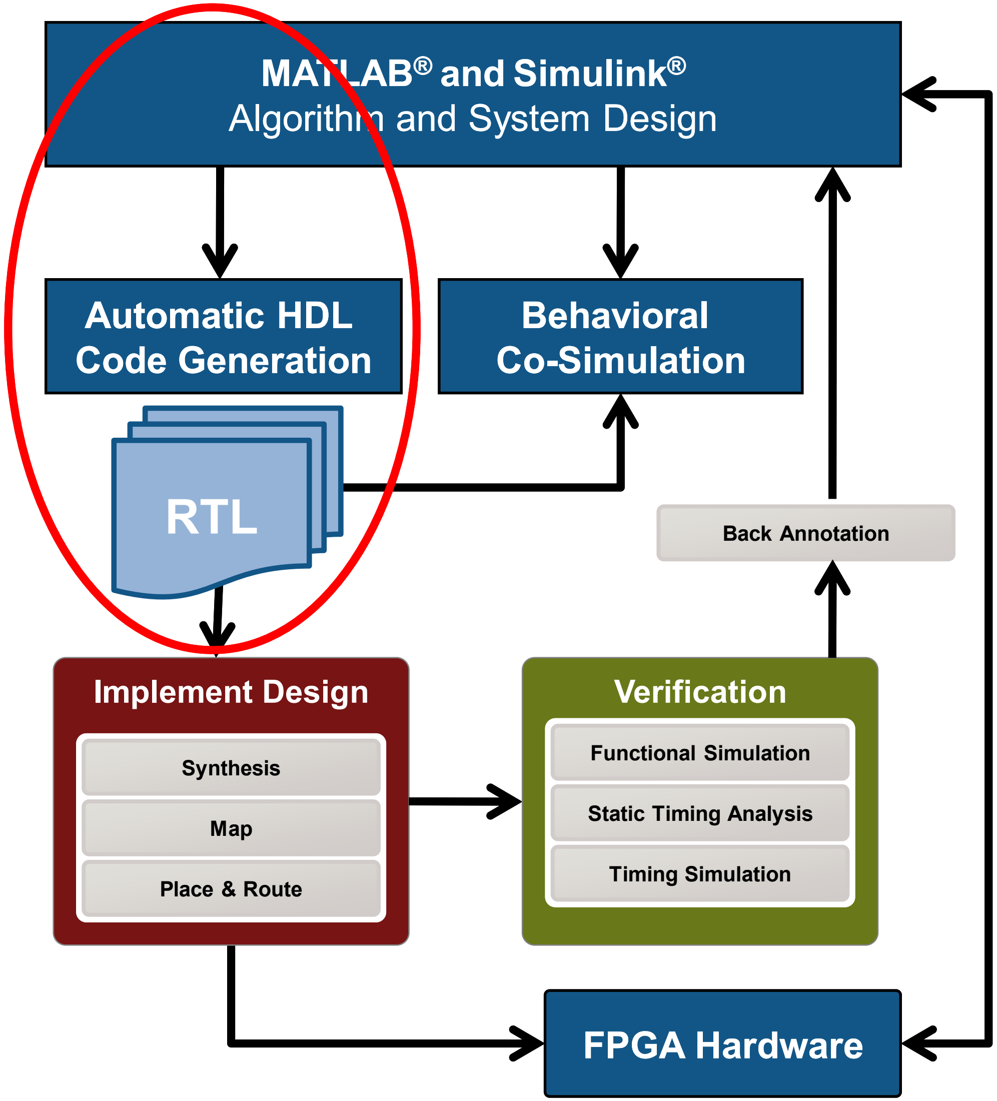
Results

- Simulations 200 times faster
- Throughput performance increased by 15%
- Gate count cut by 57%

> "The Simulink environment is ideal for system-level architecture exploration. The simulations are 200 times faster than they were in our previous workflow — and Simulink models can be easily converted to C as well as to HDL code, which enables high scalability and reusability."
>
> Ken Chen
> Faraday

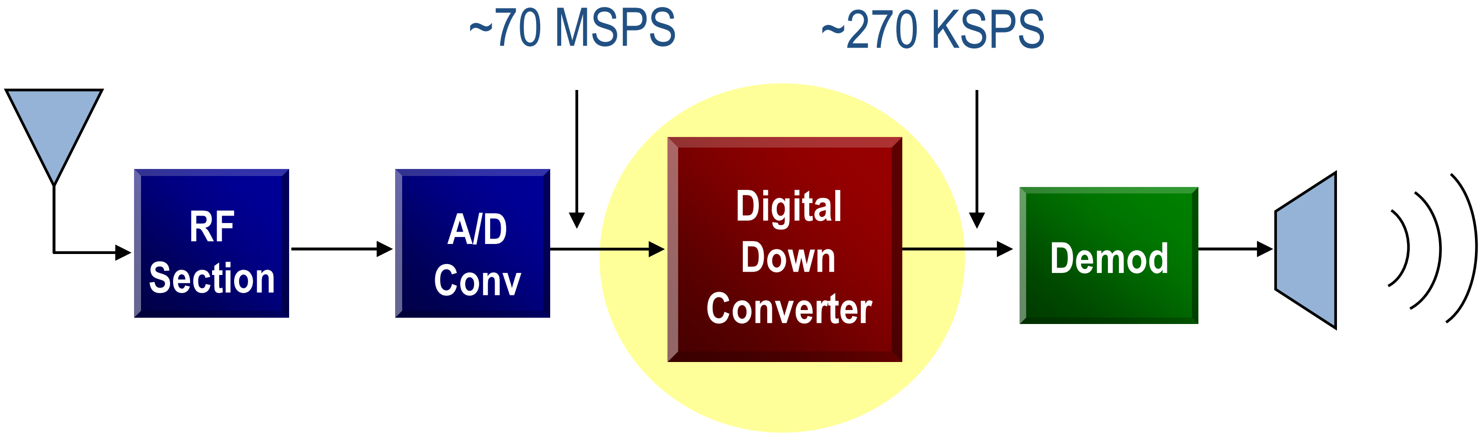
Link to user story

From Algorithm to Synthesizable RTL



Digital Down Converter

- DDC accepts
  - A high sample-rate passband signal (may be 50 to 100 Msps)
- DDC produces
  - A low sample-rate baseband signal ready for demodulation

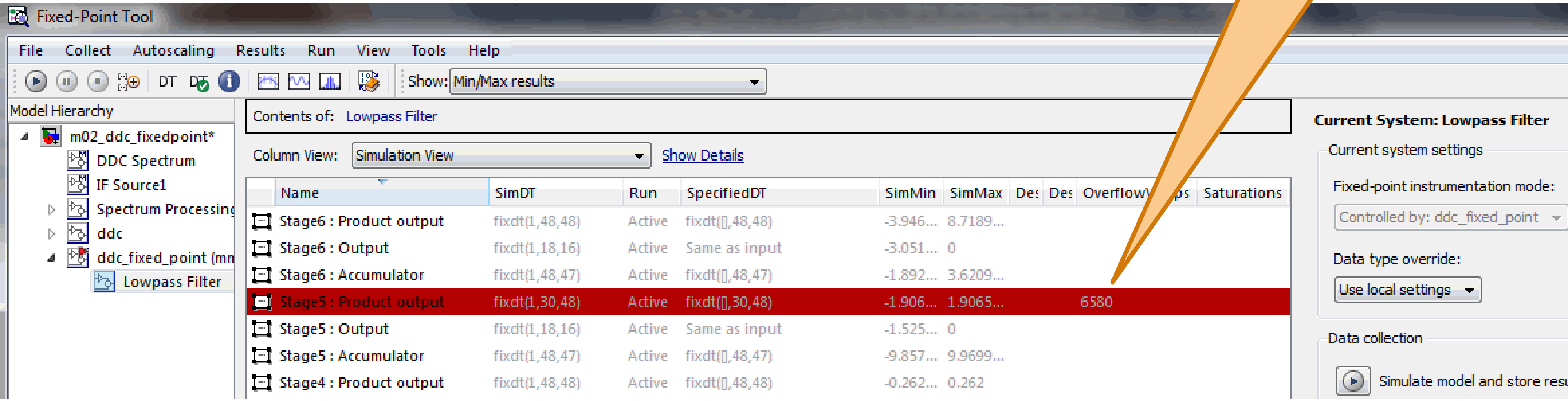


Fixed Point Analysis

Digital Down Converter

- Convert floating point to fixed point models
  - Automatic tracking of signal range (also intermediate quantities)
  - Fraction lengths recommendation
- Bit-true models in the same environment
  - Quantify the impact of fixed point quantization

Find and fix issues with fixed point easily



Show: Min/Max results

Model Hierarchy

- m02_ddc_fixedpoint*
  - DDC Spectrum
  - IF Source1
  - Spectrum Processing
  - ddc
  - ddc_fixed_point (mn
    - Lowpass Filter

Contents of: Lowpass Filter

Column View: Simulation View Show Details

Name	SimDT	Run	SpecifiedDT	SimMin	SimMax	Des	Des	OverflowW ps	Saturations
Stage6 : Product output	fixdt(1,48,48)	Active	fixdt([],48,48)	-3.946...	8.7189...				
Stage6 : Output	fixdt(1,18,16)	Active	Same as input	-3.051...	0				
Stage6 : Accumulator	fixdt(1,48,47)	Active	fixdt([],48,47)	-1.892...	3.6209...				
Stage5 : Product output	fixdt(1,30,48)	Active	fixdt([],30,48)	-1.906...	1.9065...			6580	
Stage5 : Output	fixdt(1,18,16)	Active	Same as input	-1.525...	0				
Stage5 : Accumulator	fixdt(1,48,47)	Active	fixdt([],48,47)	-9.857...	9.9699...				
Stage4 : Product output	fixdt(1,48,48)	Active	fixdt([],48,48)	-0.262...	0.262				

Current System: Lowpass Filter

Current system settings

Fixed-point instrumentation mode: Controlled by: ddc_fixed_point

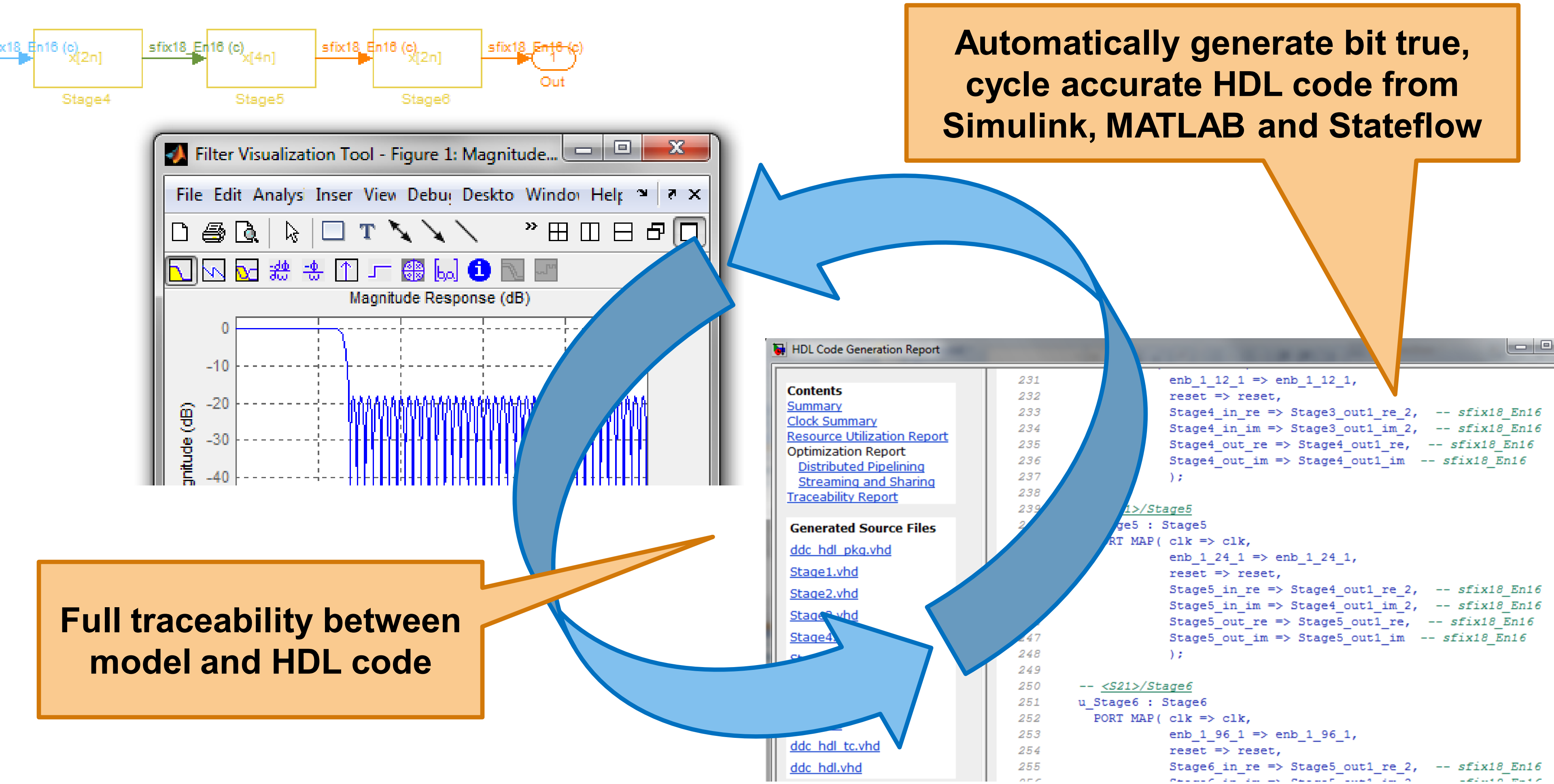
Data type override: Use local settings

Data collection

Simulate model and store resu

Automatic HDL Code Generation

Digital Down Converter



Contents
Summary
Clock Summary
Resource Utilization Report
Optimization Report
Distributed Pipelining
Streaming and Sharing
Traceability Report

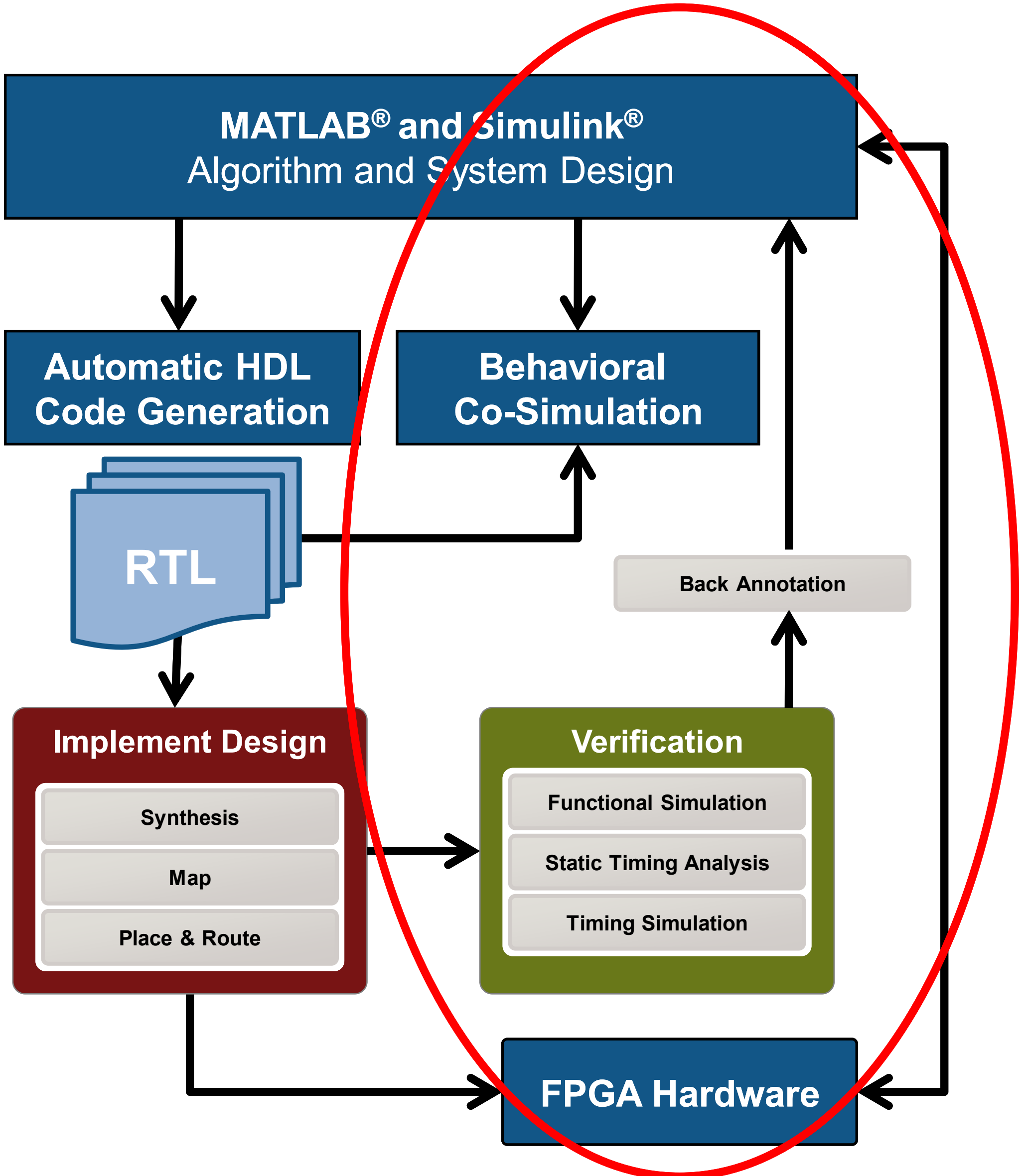
Generated Source Files
ddc_hdl_pkg.vhd
Stage1.vhd
Stage2.vhd
Stage3.vhd
Stage4.
ddc_hdl_tc.vhd
ddc_hdl.vhd

```
                enb_1_12_1 => enb_1_12_1,
                reset => reset,
                Stage4_in_re => Stage3_out1_re_2,  -- sfix18_En16
                Stage4_in_im => Stage3_out1_im_2,  -- sfix18_En16
                Stage4_out_re => Stage4_out1_re,  -- sfix18_En16
                Stage4_out_im => Stage4_out1_im  -- sfix18_En16
                );

     1>/Stage5
     ge5 : Stage5
     RT MAP( clk => clk,
                enb_1_24_1 => enb_1_24_1,
                reset => reset,
                Stage5_in_re => Stage4_out1_re_2,  -- sfix18_En16
                Stage5_in_im => Stage4_out1_im_2,  -- sfix18_En16
                Stage5_out_re => Stage5_out1_re,  -- sfix18_En16
                Stage5_out_im => Stage5_out1_im  -- sfix18_En16
                );

  -- <S21>/Stage6
  u_Stage6 : Stage6
    PORT MAP( clk => clk,
                enb_1_96_1 => enb_1_96_1,
                reset => reset,
                Stage6_in_re => Stage5_out1_re_2,  -- sfix18_En16
```

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

Full traceability between model and HDL code

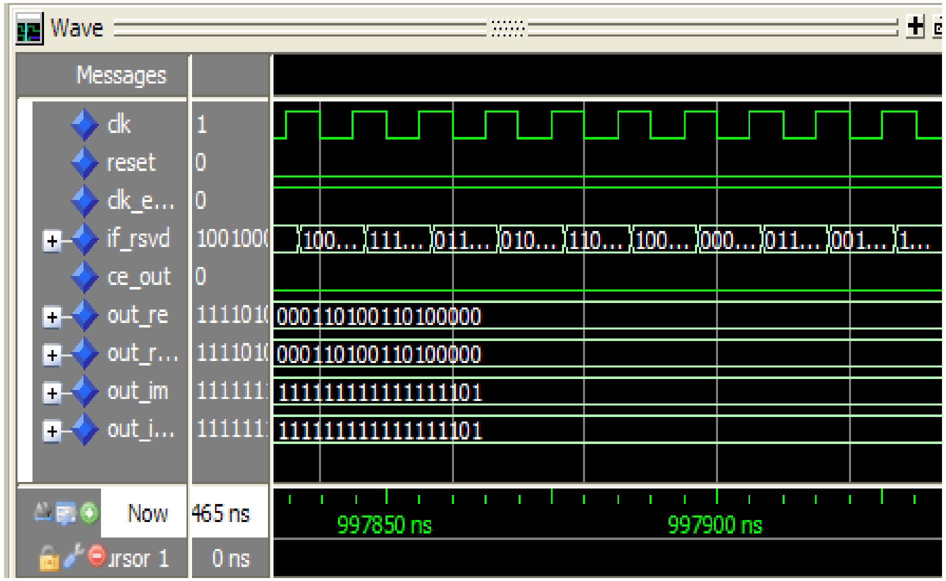
Integrated HDL Verification

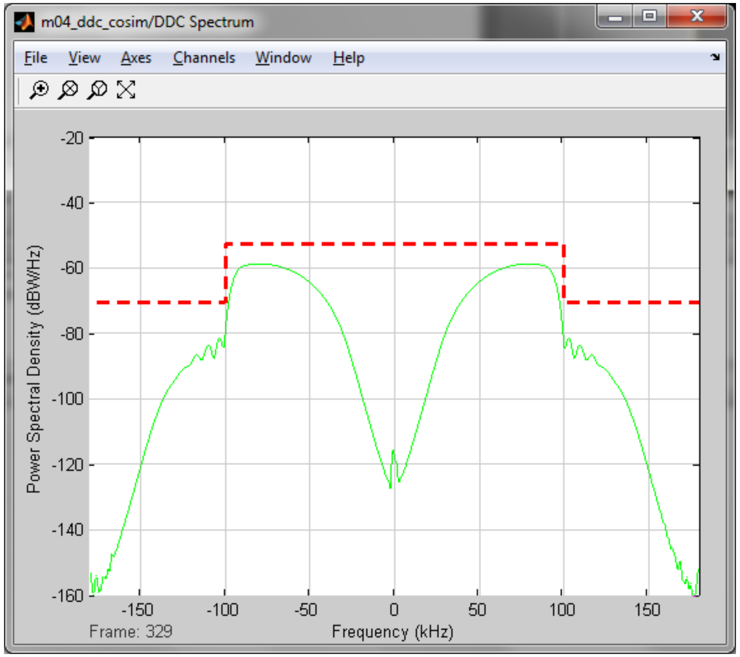


Verification Challenges:
HDL Verification

- Design the Test Bench twice
  - 10 – to – 1 ratio of Test bench LOC – to – Design LOC
- Many stimulus files from MATLAB
- These are ideal references which require pre- and post-processing
- How to analyze results?

Verification Challenges:
HDL Verification


Wave
Messages
clk 1
reset 0
clk_e... 0
if_rsvd 100100(100... 111... 011... 010... 110... 100... 000... 011... 001... 1...
ce_out 0
out_re 111101(0001101001101000000
out_r... 111101(0001101001101000000
out_im 111111 11111111111111101
out_i... 111111 11111111111111101
Now 465 ns
997850 ns
997900 ns
ırsor 1 0 ns



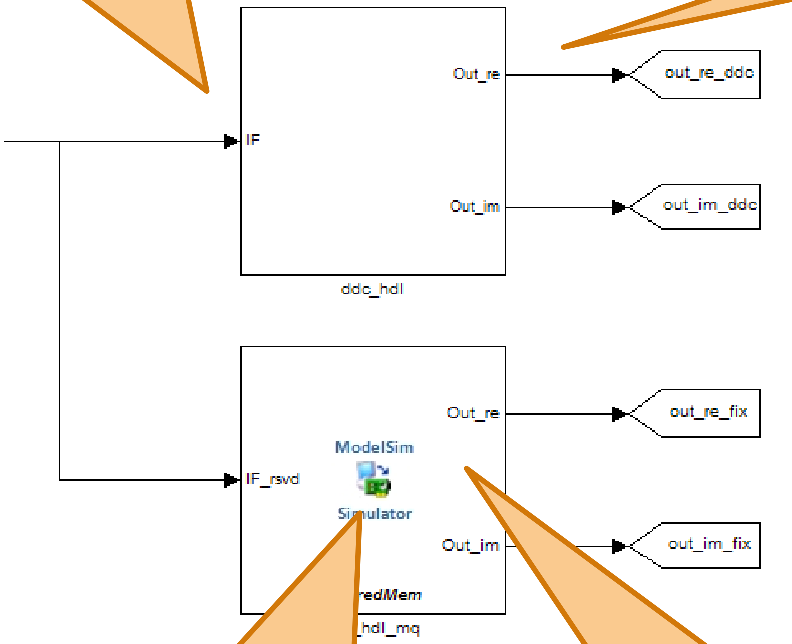
Demo: Re-Use System Level Test Bench

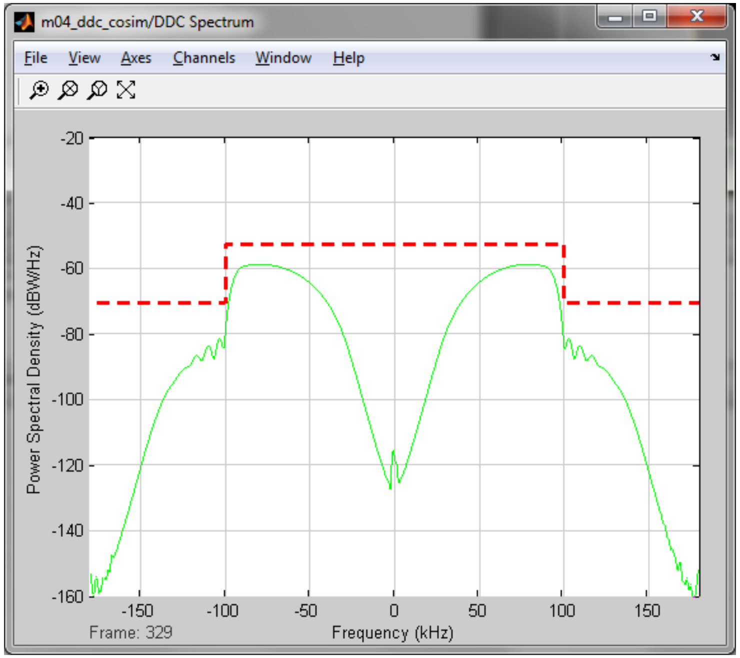
Co-Simulation with HDL simulators

Digital Down Converter

Re-use system level test bench

Flexible test bench creation in Simulink

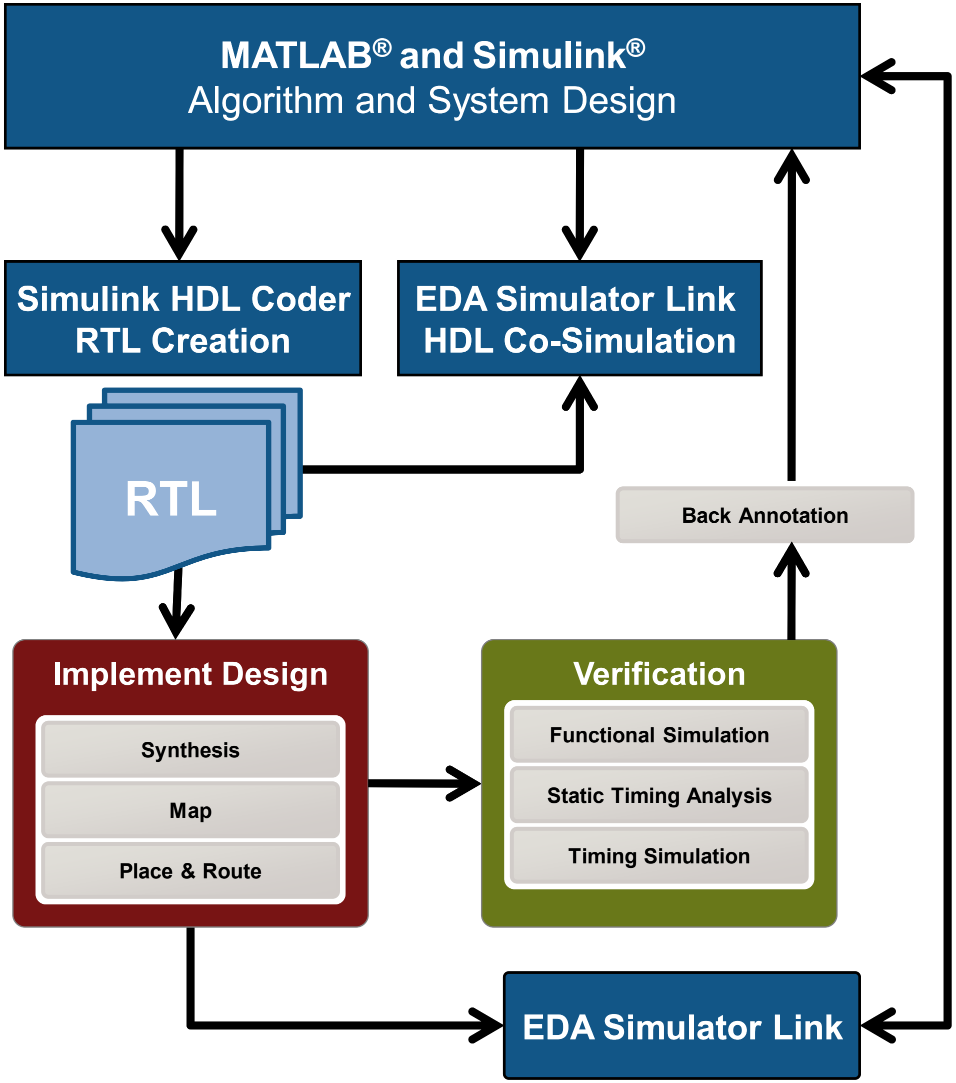




Direct simulation link to HDL Simulators

Automatically generated co-simulation models and Wizards for legacy HDL code

From Algorithm to FPGA Prototyping and Verification



MATLAB® and Simulink®
Algorithm and System Design

Simulink HDL Coder
RTL Creation

EDA Simulator Link
HDL Co-Simulation

RTL

Back Annotation

Implement Design

- Synthesis
- Map
- Place & Route

Verification

- Functional Simulation
- Static Timing Analysis
- Timing Simulation

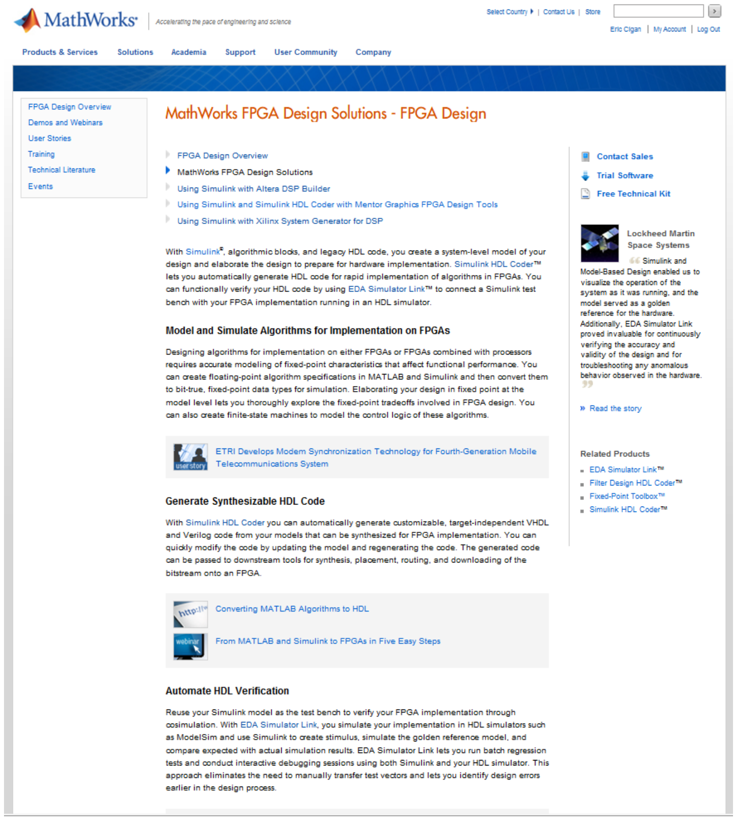
EDA Simulator Link

Next Steps …

1. Visit www.mathworks.com/fpga for more information

2. Watch our FPGA webinars: mathworks.com/company/events/webinars

3. Contact your local sales reps for a **trial** of MathWorks HDL code generation and verification products



MathWorks | Accelerating the pace of engineering and science

Select Country | Contact Us | Store

Eric Cigan | My Account | Log Out

Products & Services Solutions Academia Support User Community Company

- FPGA Design Overview
- Demos and Webinars
- User Stories
- Training
- Technical Literature
- Events

MathWorks FPGA Design Solutions - FPGA Design

- FPGA Design Overview
- MathWorks FPGA Design Solutions
- Using Simulink with Altera DSP Builder
- Using Simulink and Simulink HDL Coder with Mentor Graphics FPGA Design Tools
- Using Simulink with Xilinx System Generator for DSP

With Simulink®, algorithmic blocks, and legacy HDL code, you create a system-level model of your design and elaborate the design to prepare for hardware implementation. Simulink HDL Coder™ lets you automatically generate HDL code for rapid implementation of algorithms in FPGAs. You can functionally verify your HDL code by using EDA Simulator Link™ to connect a Simulink test bench with your FPGA implementation running in an HDL simulator.

Model and Simulate Algorithms for Implementation on FPGAs

Designing algorithms for implementation on either FPGAs or FPGAs combined with processors requires accurate modeling of fixed-point characteristics that affect functional performance. You can create floating-point algorithm specifications in MATLAB and Simulink and then convert them to bit-true, fixed-point data types for simulation. Elaborating your design in fixed point at the model level lets you thoroughly explore the fixed-point tradeoffs involved in FPGA design. You can also create finite-state machines to model the control logic of these algorithms.

user story ETRI Develops Modem Synchronization Technology for Fourth-Generation Mobile Telecommunications System

Generate Synthesizable HDL Code

With Simulink HDL Coder you can automatically generate customizable, target-independent VHDL and Verilog code from your models that can be synthesized for FPGA implementation. You can quickly modify the code by updating the model and regenerating the code. The generated code can be passed to downstream tools for synthesis, placement, routing, and downloading of the bitstream onto an FPGA.

Converting MATLAB Algorithms to HDL

webinar From MATLAB and Simulink to FPGAs in Five Easy Steps

Automate HDL Verification

Reuse your Simulink model as the test bench to verify your FPGA implementation through cosimulation. With EDA Simulator Link, you simulate your implementation in HDL simulators such as ModelSim and use Simulink to create stimulus, simulate the golden reference model, and compare expected with actual simulation results. EDA Simulator Link lets you run batch regression tests and conduct interactive debugging sessions using both Simulink and your HDL simulator. This approach eliminates the need to manually transfer test vectors and lets you identify design errors earlier in the design process.

Contact Sales

Trial Software

Free Technical Kit

Lockheed Martin Space Systems

"Simulink and Model-Based Design enabled us to visualize the operation of the system as it was running, and the model served as a golden reference for the hardware. Additionally, EDA Simulator Link proved invaluable for continuously verifying the accuracy and validity of the design and for troubleshooting any anomalous behavior observed in the hardware."

» Read the story

Related Products

- EDA Simulator Link™
- Filter Design HDL Coder™
- Fixed-Point Toolbox™
- Simulink HDL Coder™

Questions?