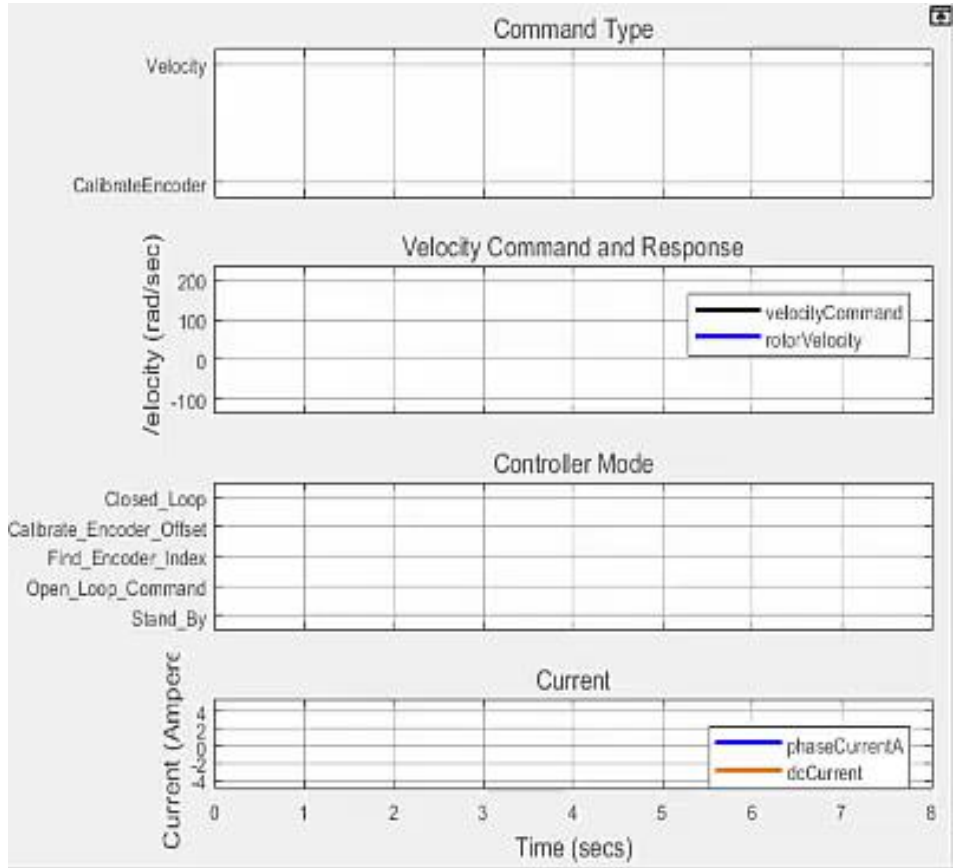
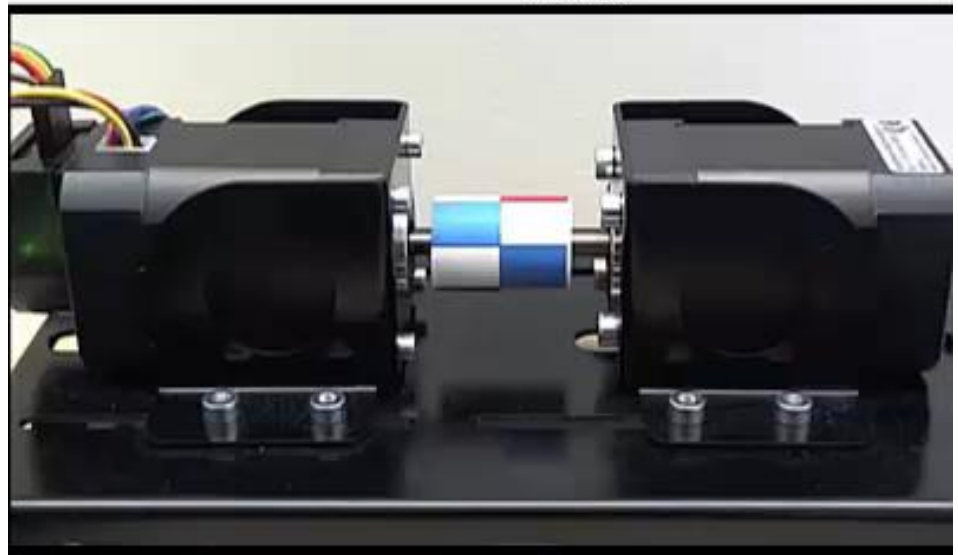
MATLAB EXPO 2018

Hardware and Software Co-Design for Motor Control Applications

Gaurav Dubey
Durvesh Kulkarni

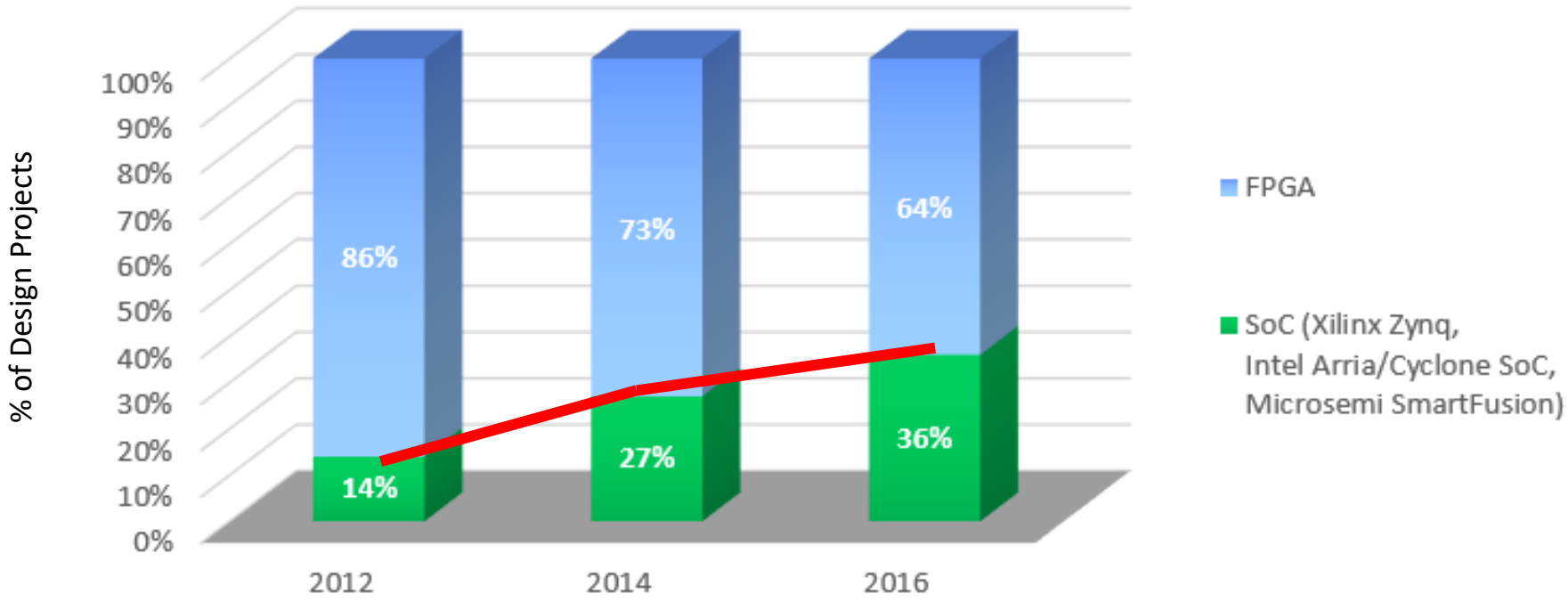




Key trend: Increasing demands from motor drives

- Advanced algorithms require faster computing performance.
  - Field-Oriented Control
  - Sensorless motor control
  - Vibration detection and suppression
  - Multi-axis control

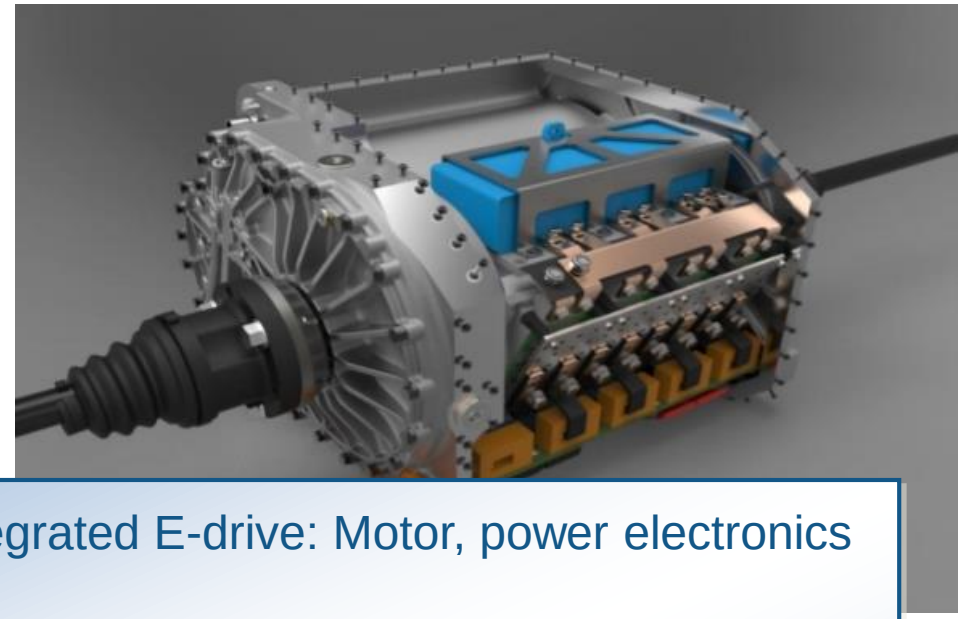
Key Trend: SoCs are now used in 36% of new FPGA projects



Source: Wilson Research Group and Mentor Graphics, 2016 Functional Verification Study

Punch Powertrain develops complex SoC-based motor control

- Powertrains for hybrid and electric vehicles
- Need to increase power density and efficiency at a reduced cost
  - Integrate motor and power electronics in the transmission
- New switched reluctance motor
  - Fast: 2x the speed of their previous motor
    - Target to a Xilinx® Zynq® SoC 7045 device
  - Complex: 4 different control strategies
- No experience designing FPGAs!



- ✓ Designed integrated E-drive: Motor, power electronics and software
- ✓ 4 different control strategies implemented
- ✓ Completed in 1.5 years with 2FTE's
- ✓ Models reusable for production
- ✓ Smooth integration and validation due to development process – thorough validation before electronics are produced and put in the testbench

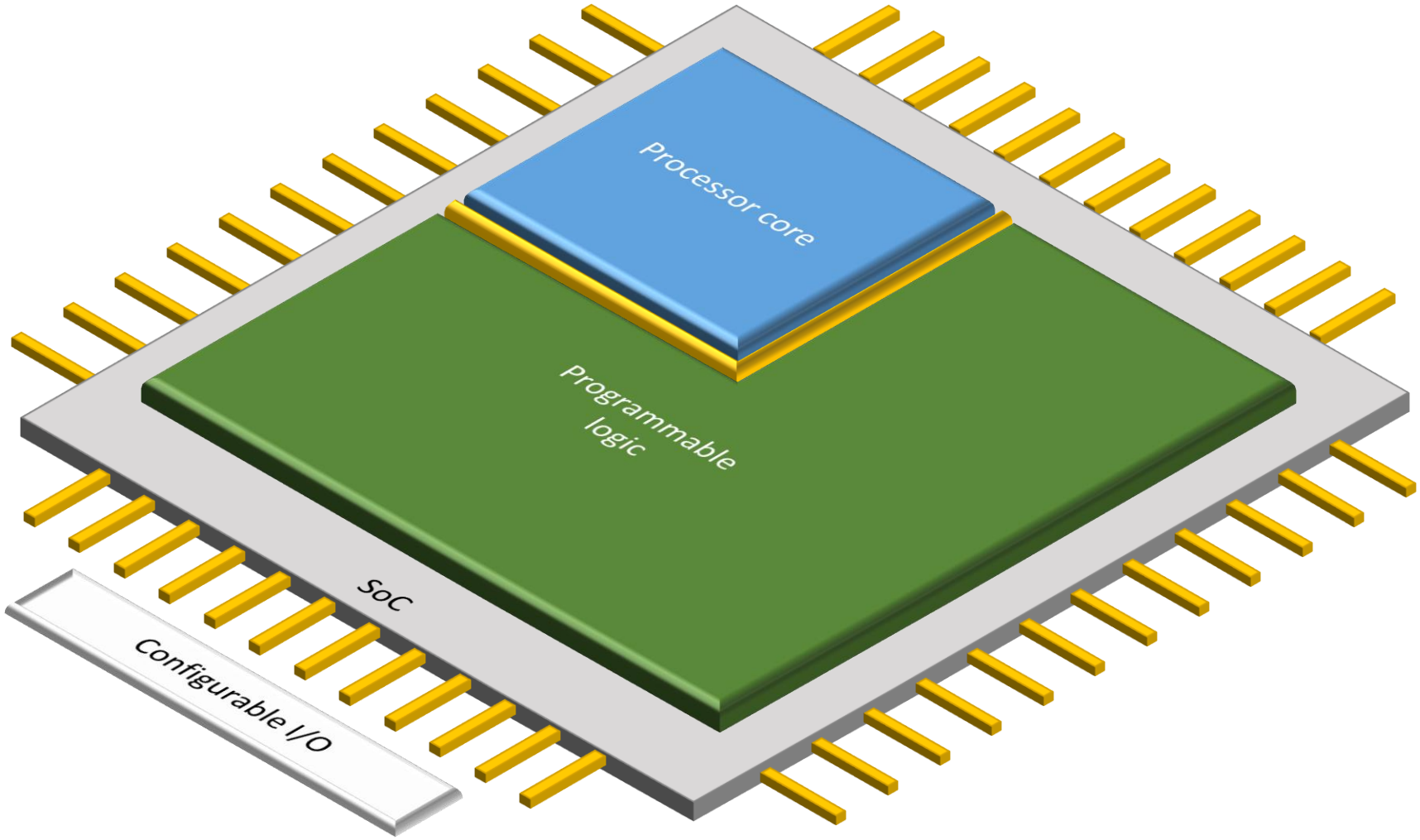
Link to video of presentation

Takeaways

Model-Based Design for SoC FPGAs

- Enables early validation of specifications using simulation
- Improves design team collaboration and designer productivity.
- Reduces hardware testing time by 5x

What's an SoC?



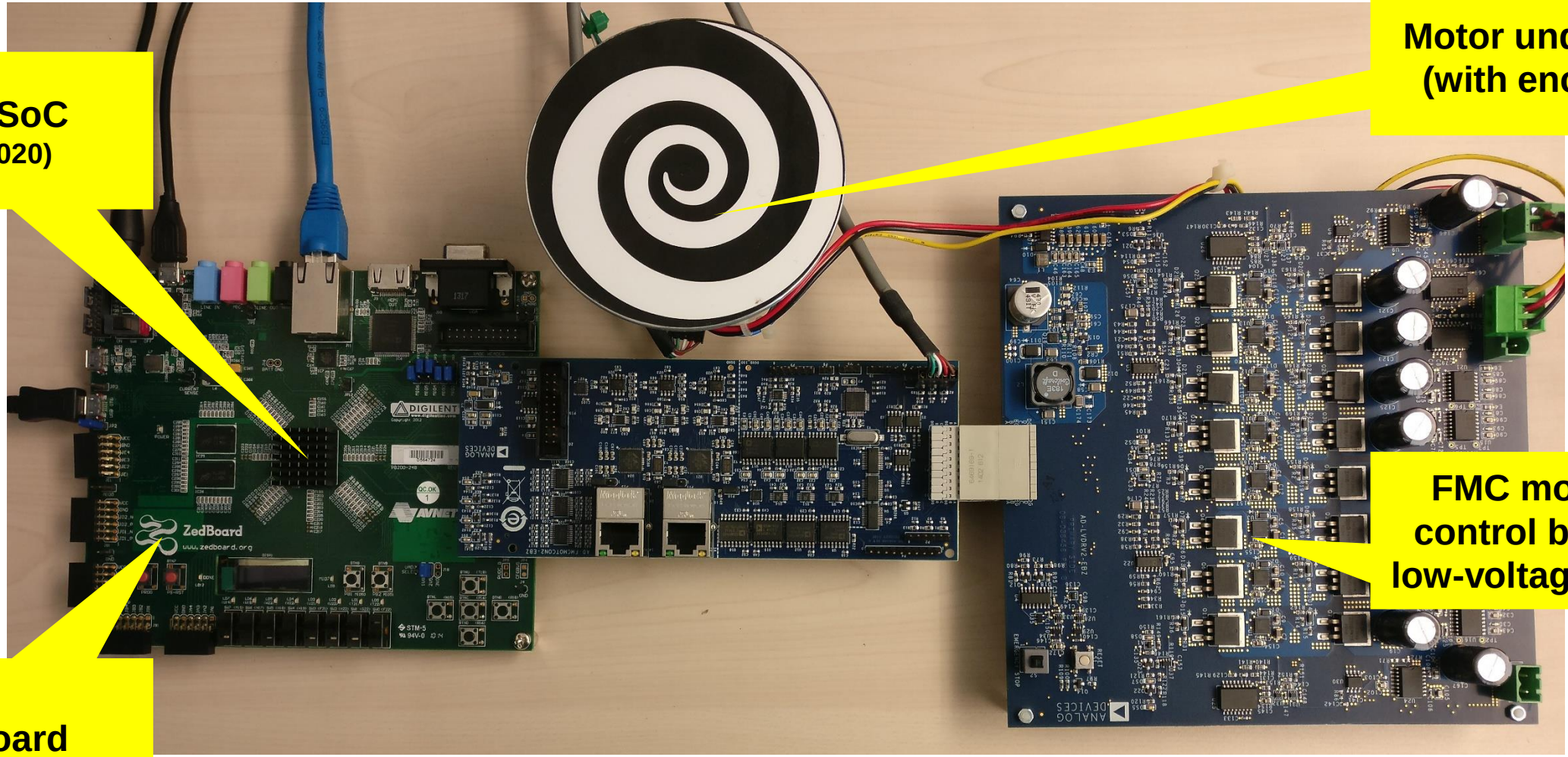
Challenges in using SoCs for Motor and Power Control

- Integration of software and hardware partitions of algorithm on SoC drives need for collaboration
- Validation of design specifications with limits on access to motors in labs.
- How to make design decisions that cut across system components?

Why use Model-Based Design to develop motor control applications on SoCs?

- Enables early validation of specifications using simulation months before hardware is available.
- Improves design team collaboration and designer productivity by using a shared design environment.
- Reduces hardware testing time by 5x by shifting design from lab to the desktop

SoC Hardware Setup

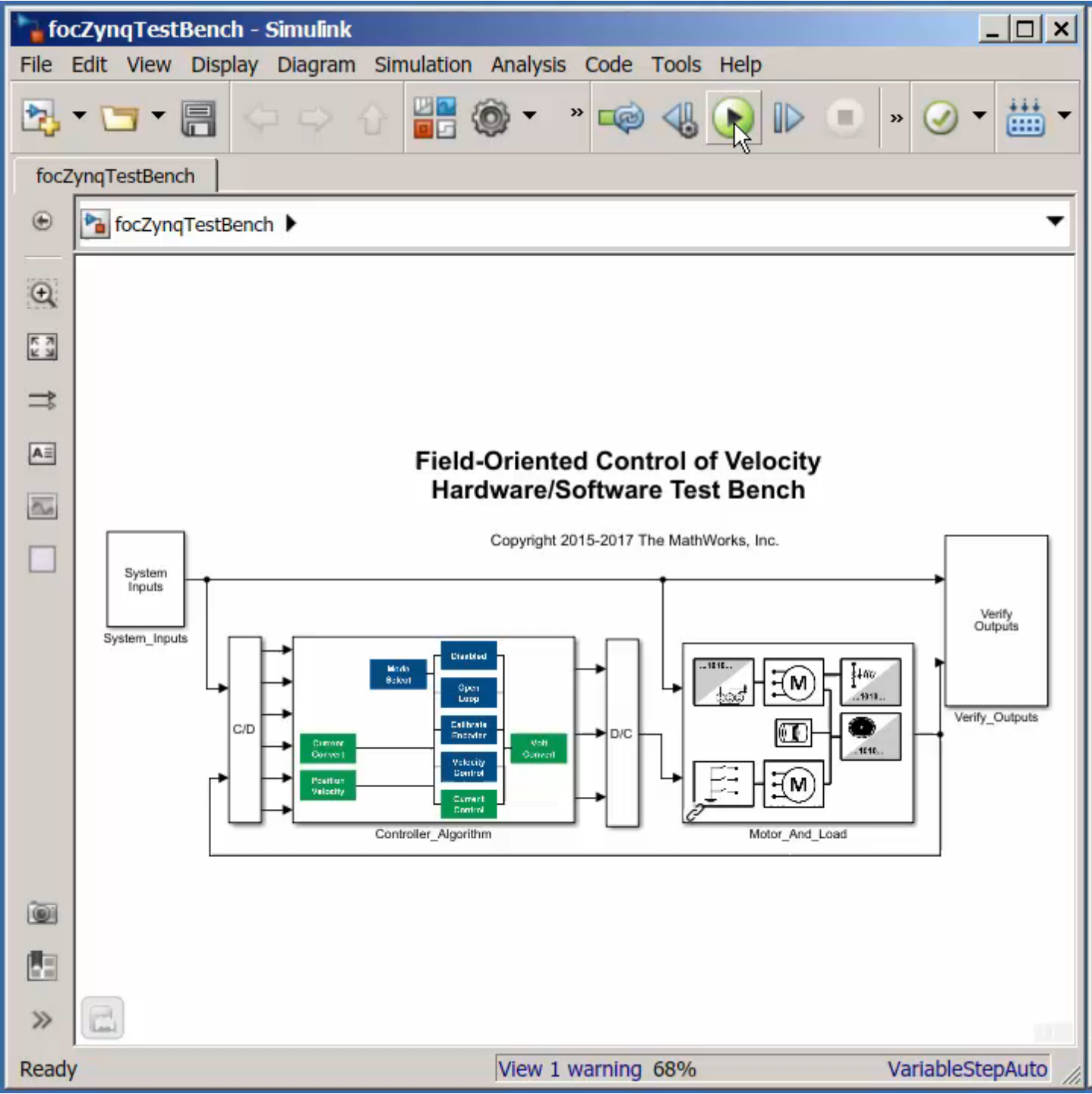


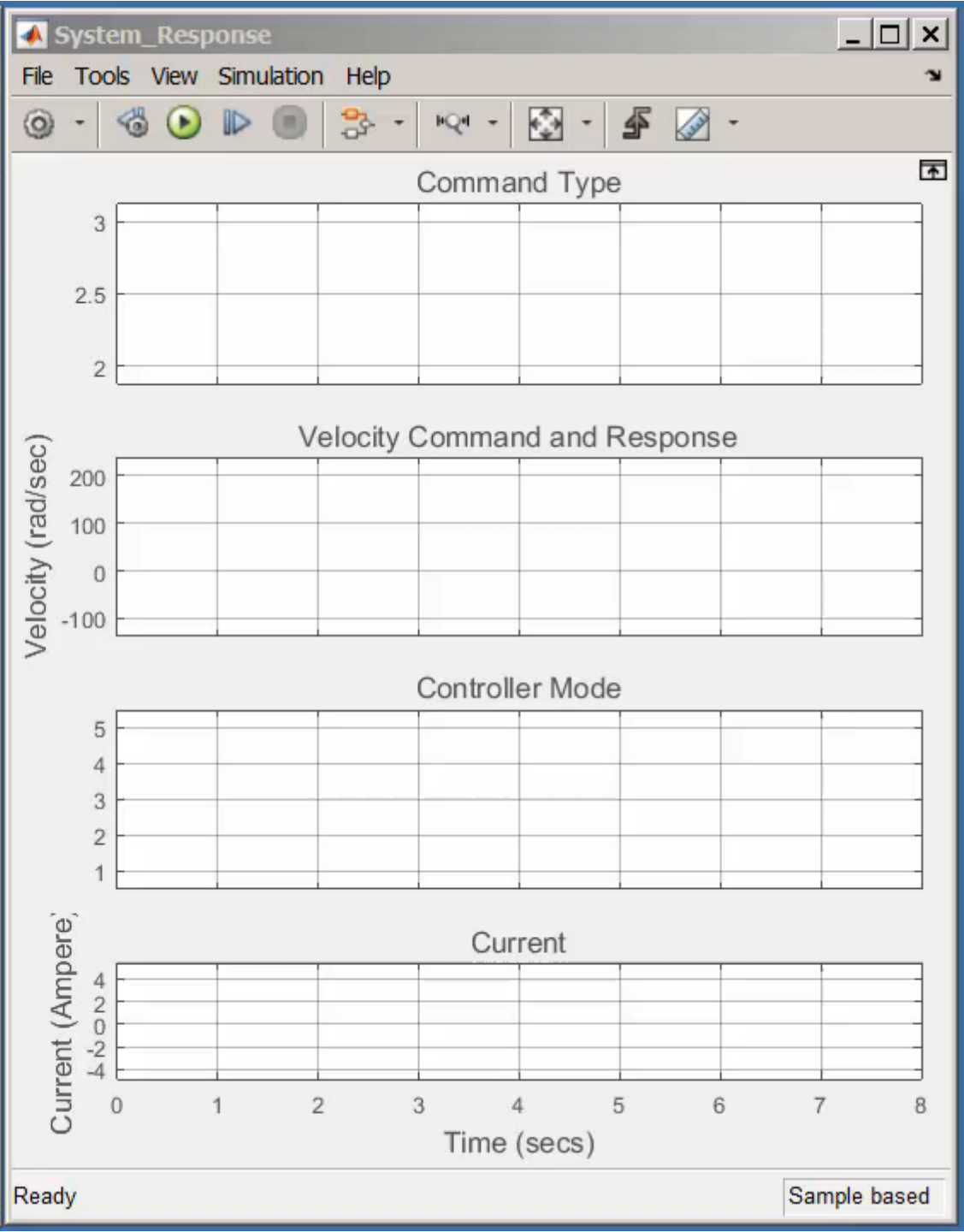
Zynq SoC
(XC7Z020)

Motor under test
(with encoder)

FMC module:
control board +
low-voltage board

ZedBoard


focZynqTestBench - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools Help
focZynqTestBench
focZynqTestBench
Field-Oriented Control of Velocity
Hardware/Software Test Bench
Copyright 2015-2017 The MathWorks, Inc.
System Inputs
System_Inputs
C/D
Controller_Algorithm
D/C
Motor_And_Load
Verify Outputs
Verify_Outputs
Ready
View 1 warning 68%
VariableStepAuto

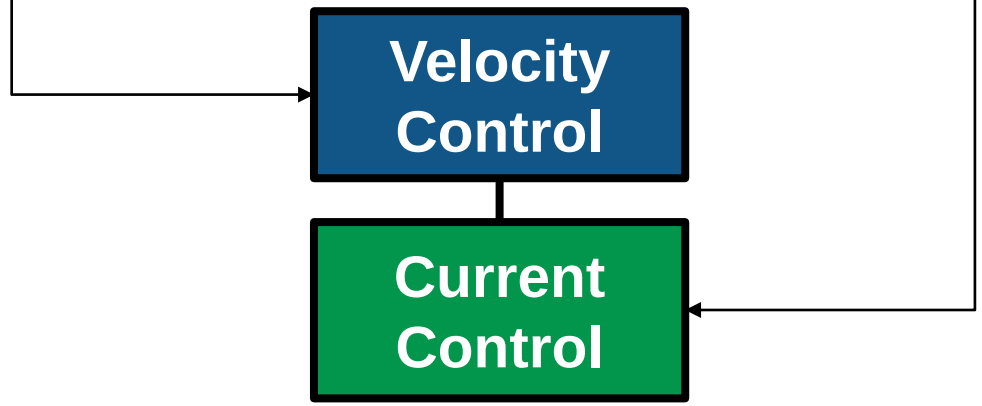

System_Response
File Tools View Simulation Help
Command Type
Velocity Command and Response
Velocity (rad/sec)
Controller Mode
Current
Current (Ampere)
Time (secs)
Ready
Sample based

Key controller and peripheral components

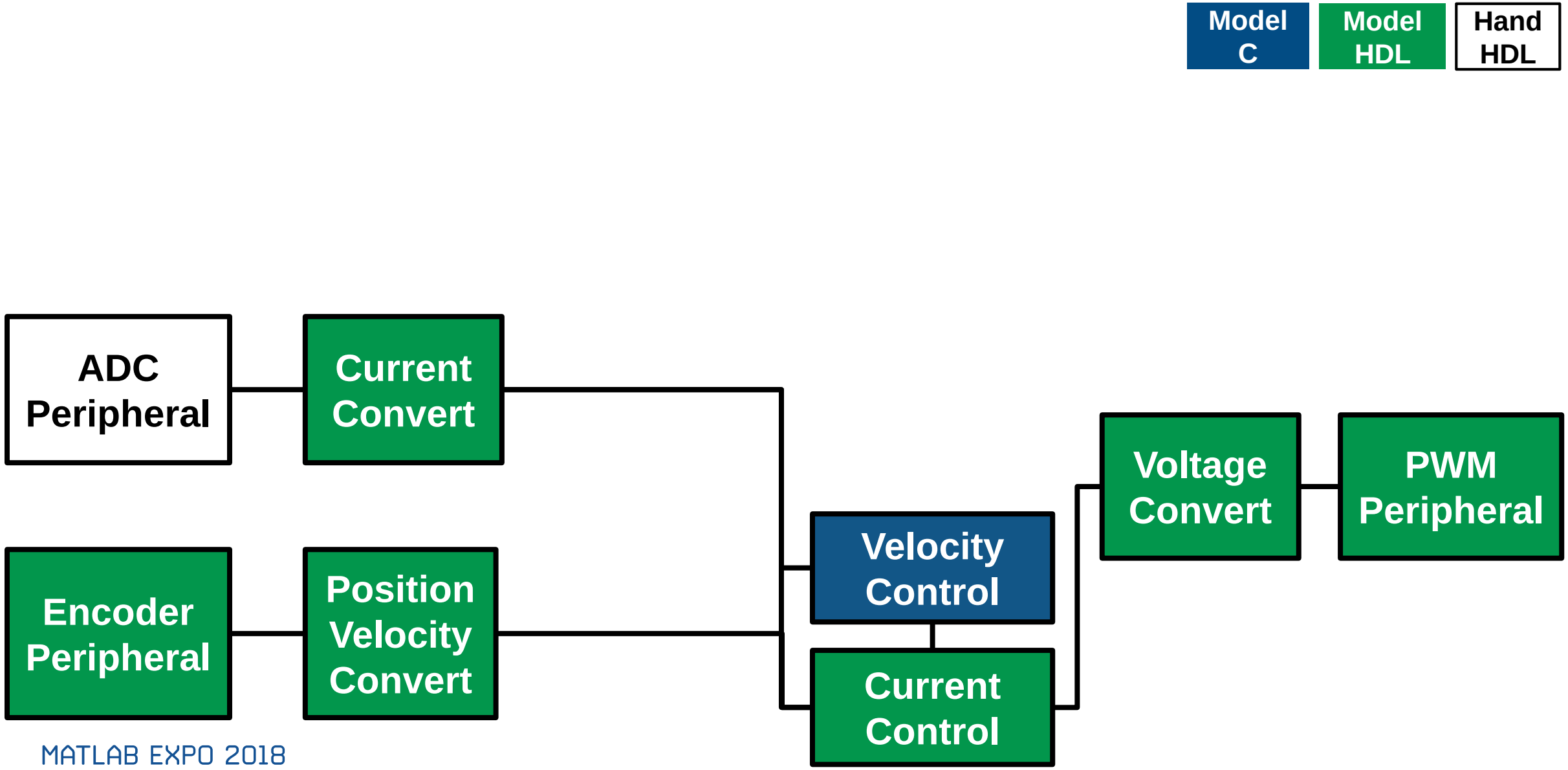
Model C | Model HDL | Hand HDL

Velocity Control
- 1 kHz
- ARM Core

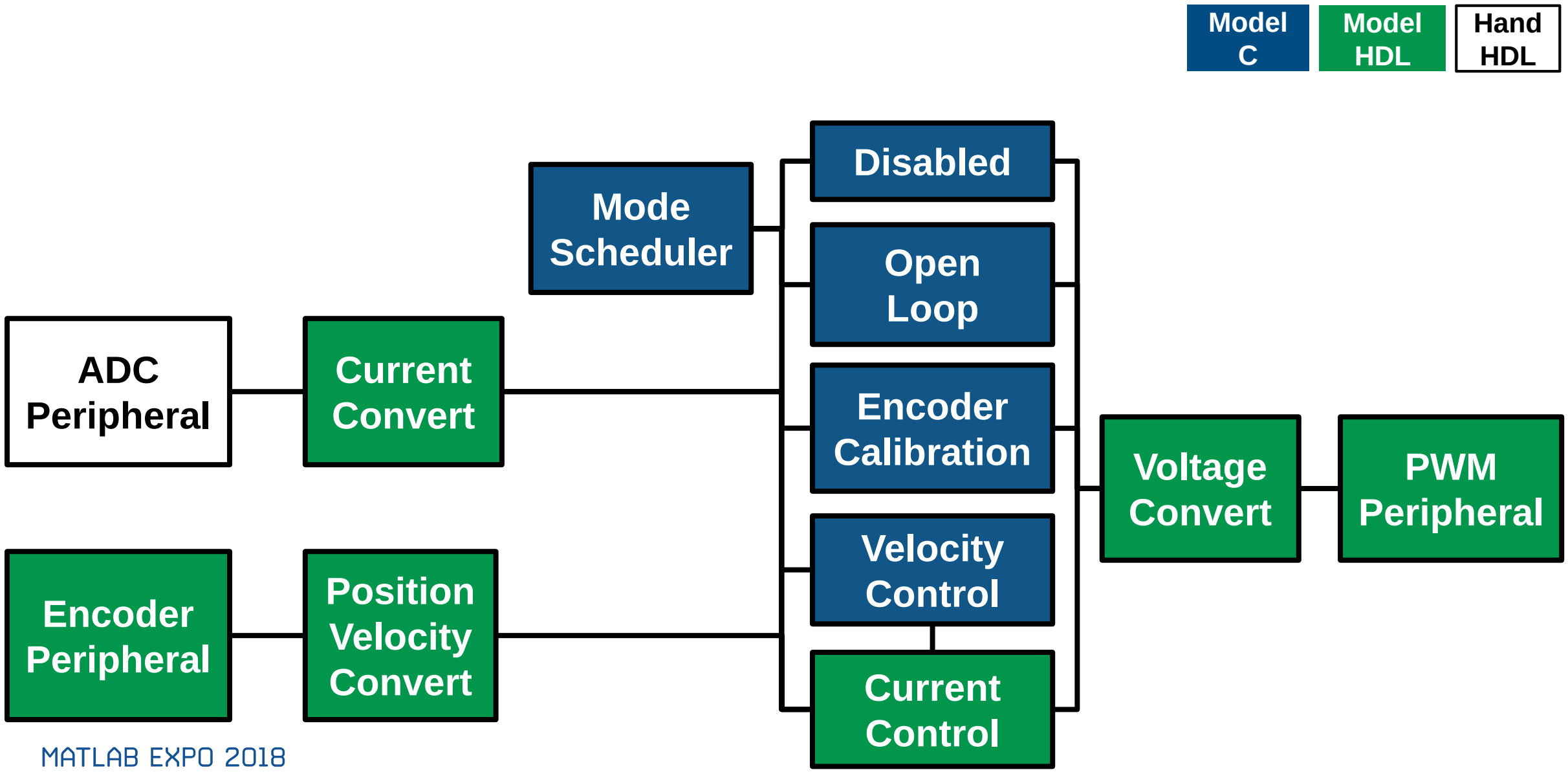
Current Control
- 25 kHz
- Sine & Cosine
- Clarke & Park Transformations
- FPGA Core



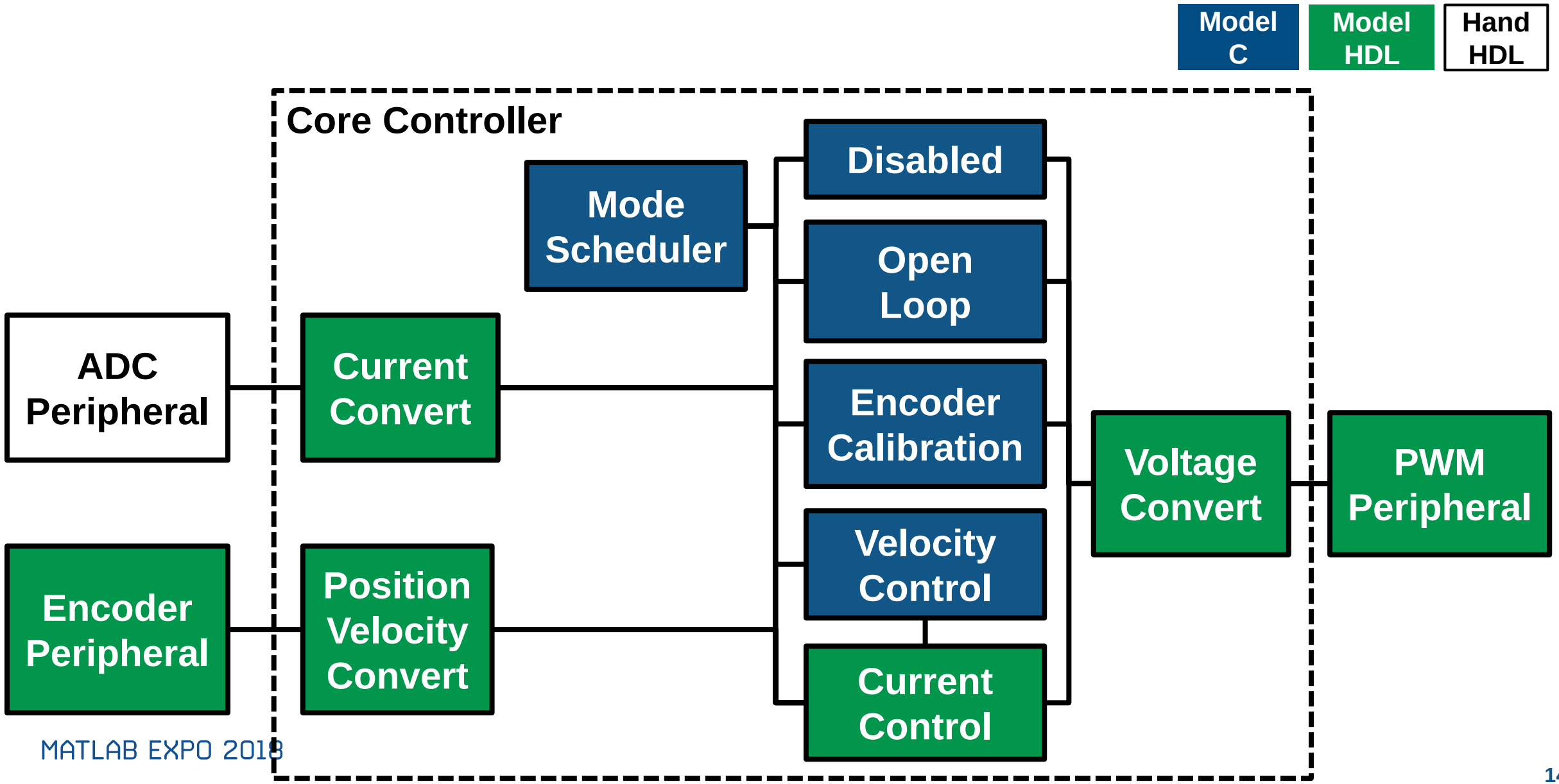
Key controller and peripheral components



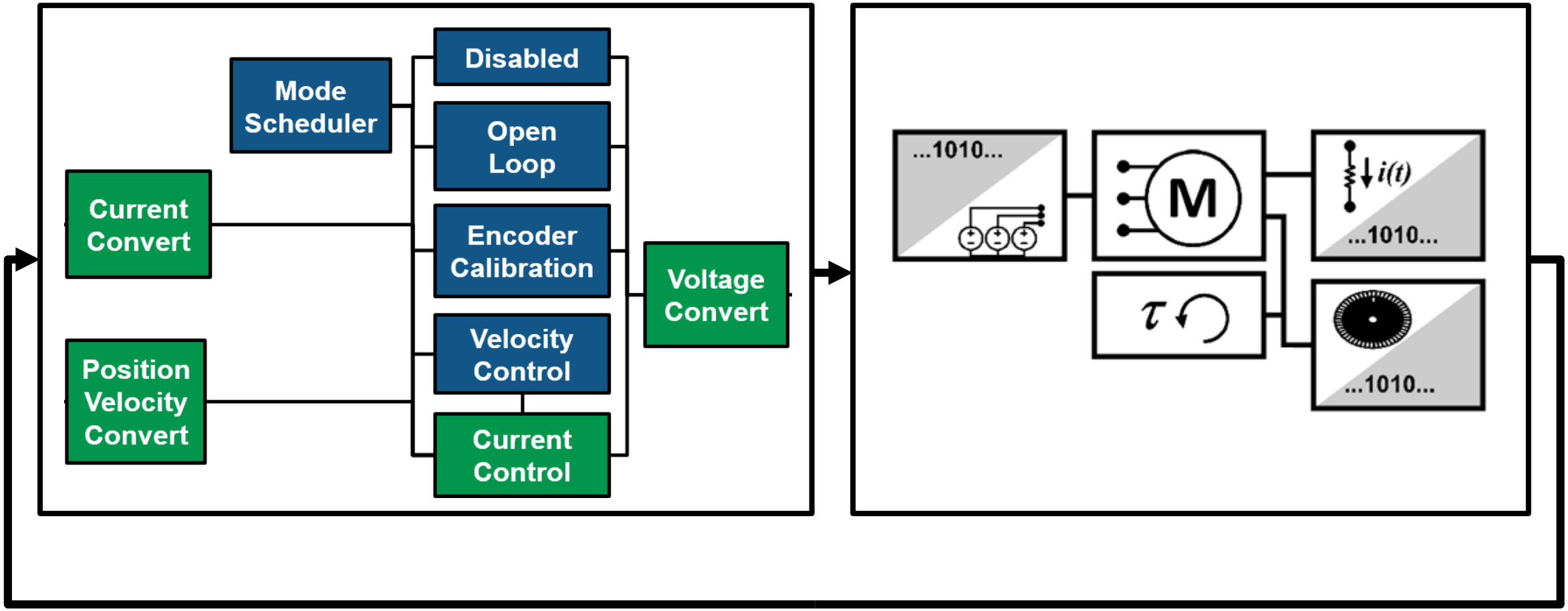
Key controller and peripheral components



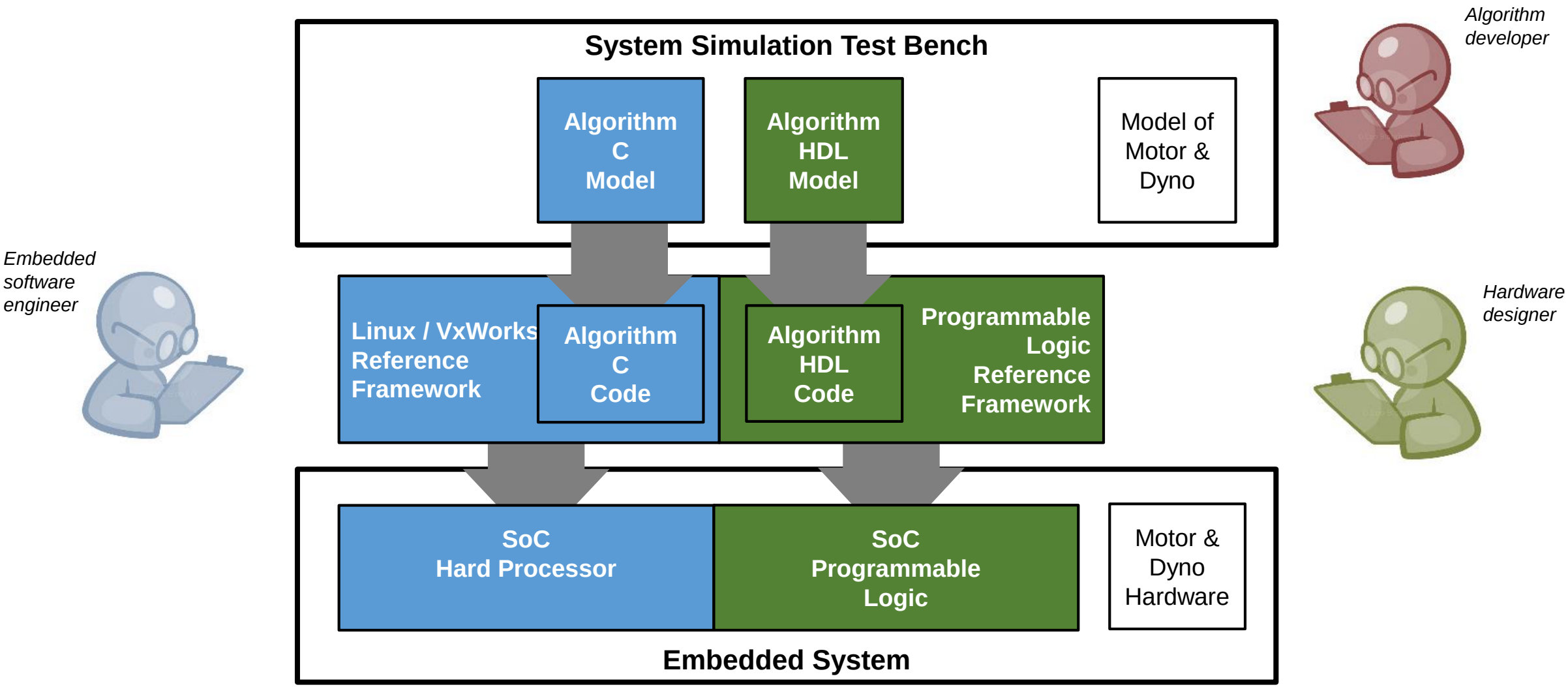
Key controller and peripheral components



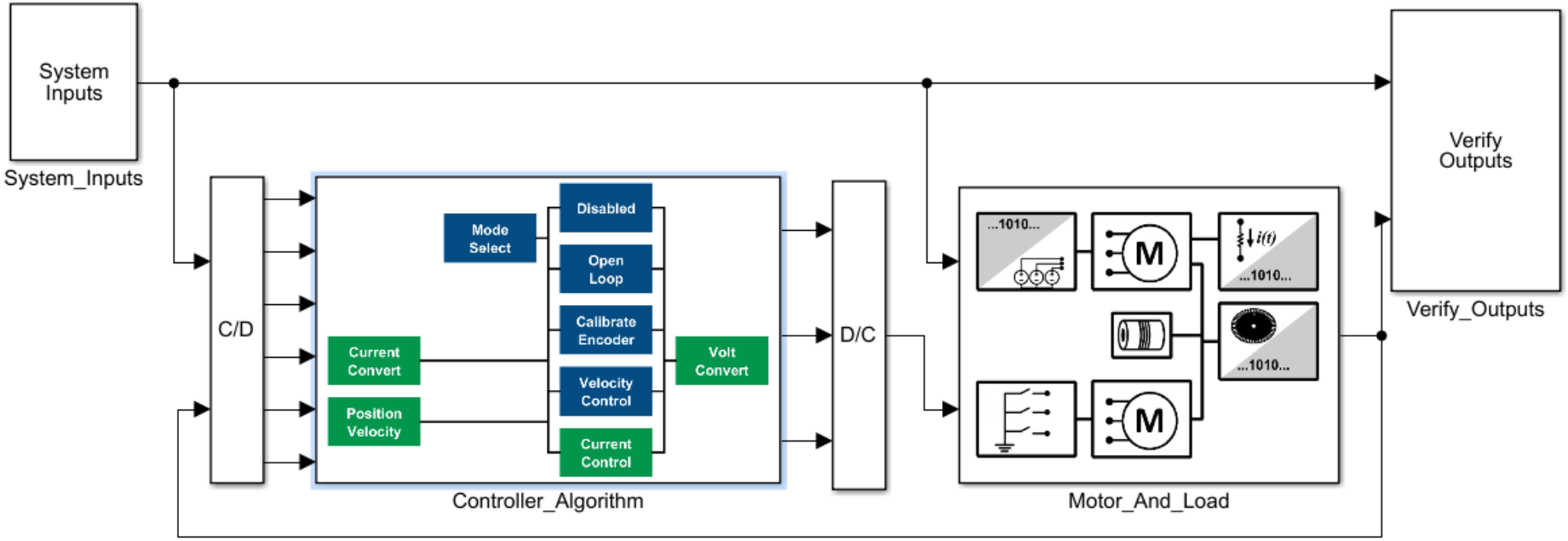
Test controller algorithm with simulation



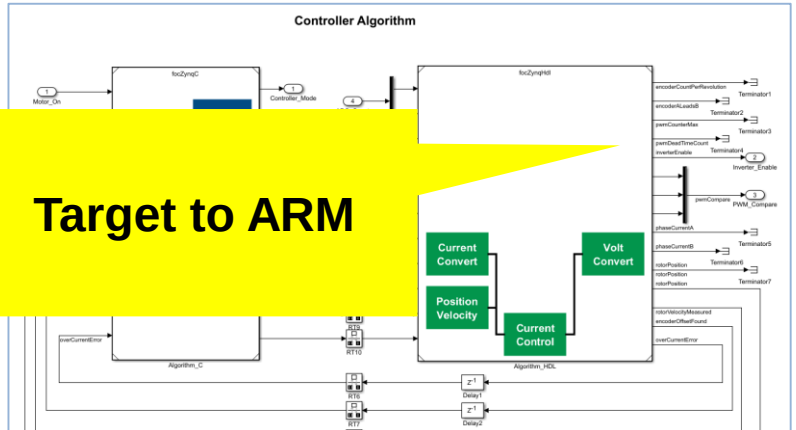
Conceptual workflow targeting SoCs



Hardware/software partitioning

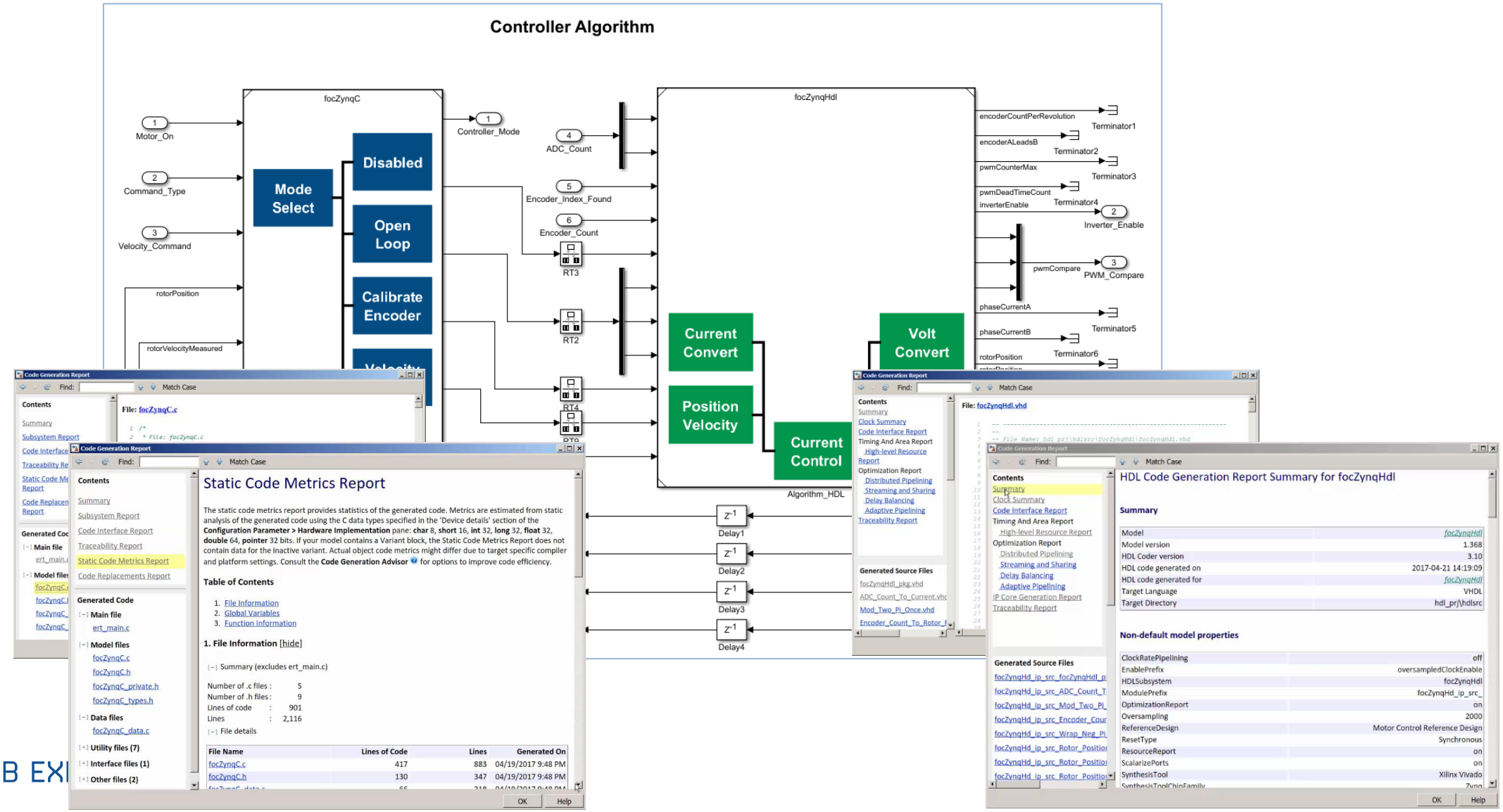


Target to Programmable Logic



Target to ARM

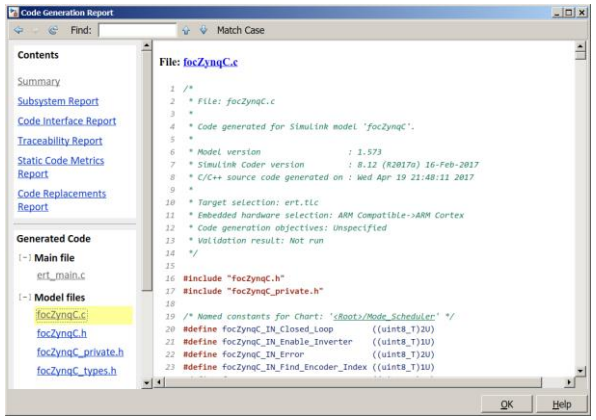
Code Generation



Code Generation

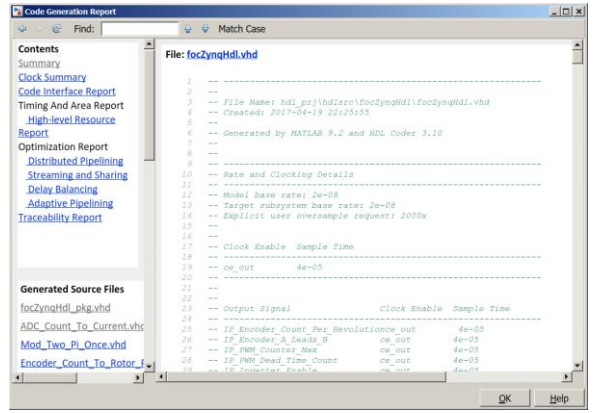
C Code

Generating Industry Standards Production C Code Using Embedded Coder
16:30–17:15

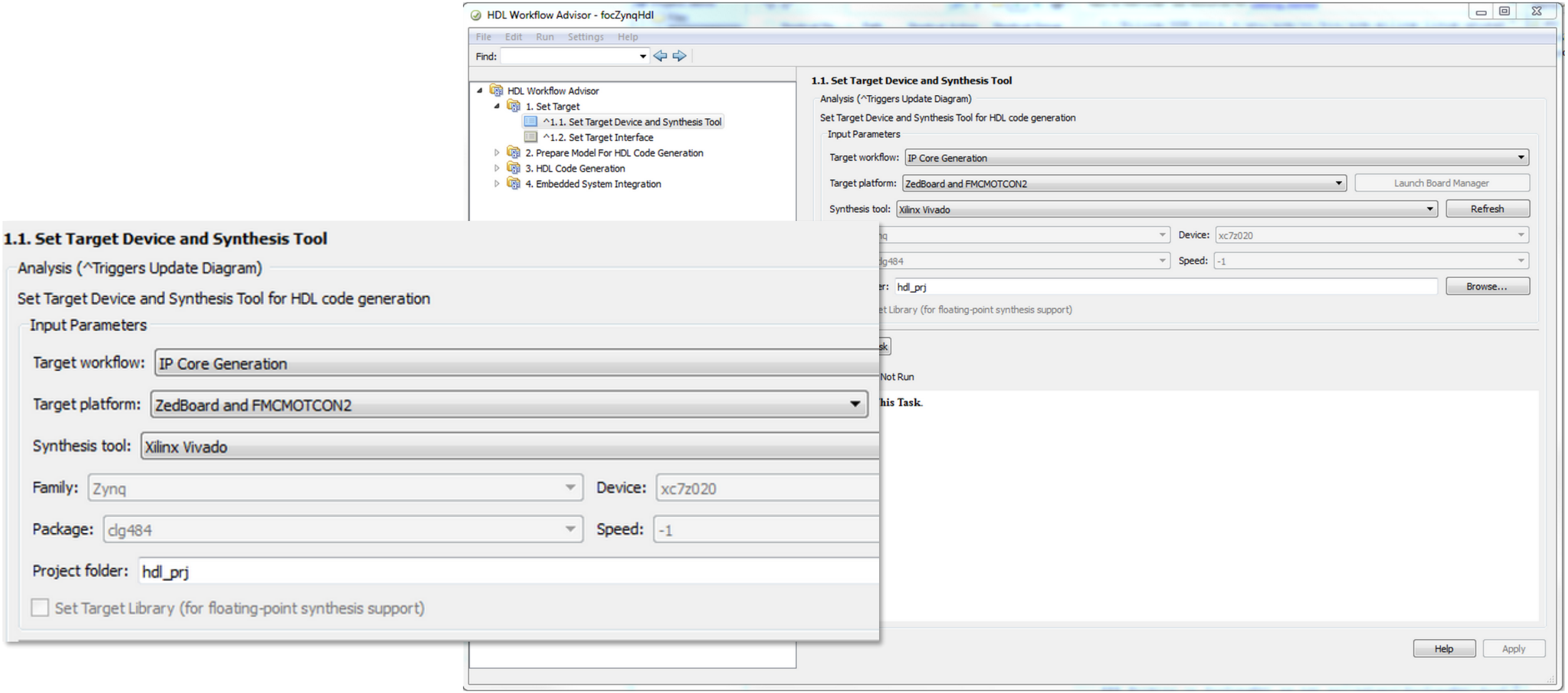


HDL Code

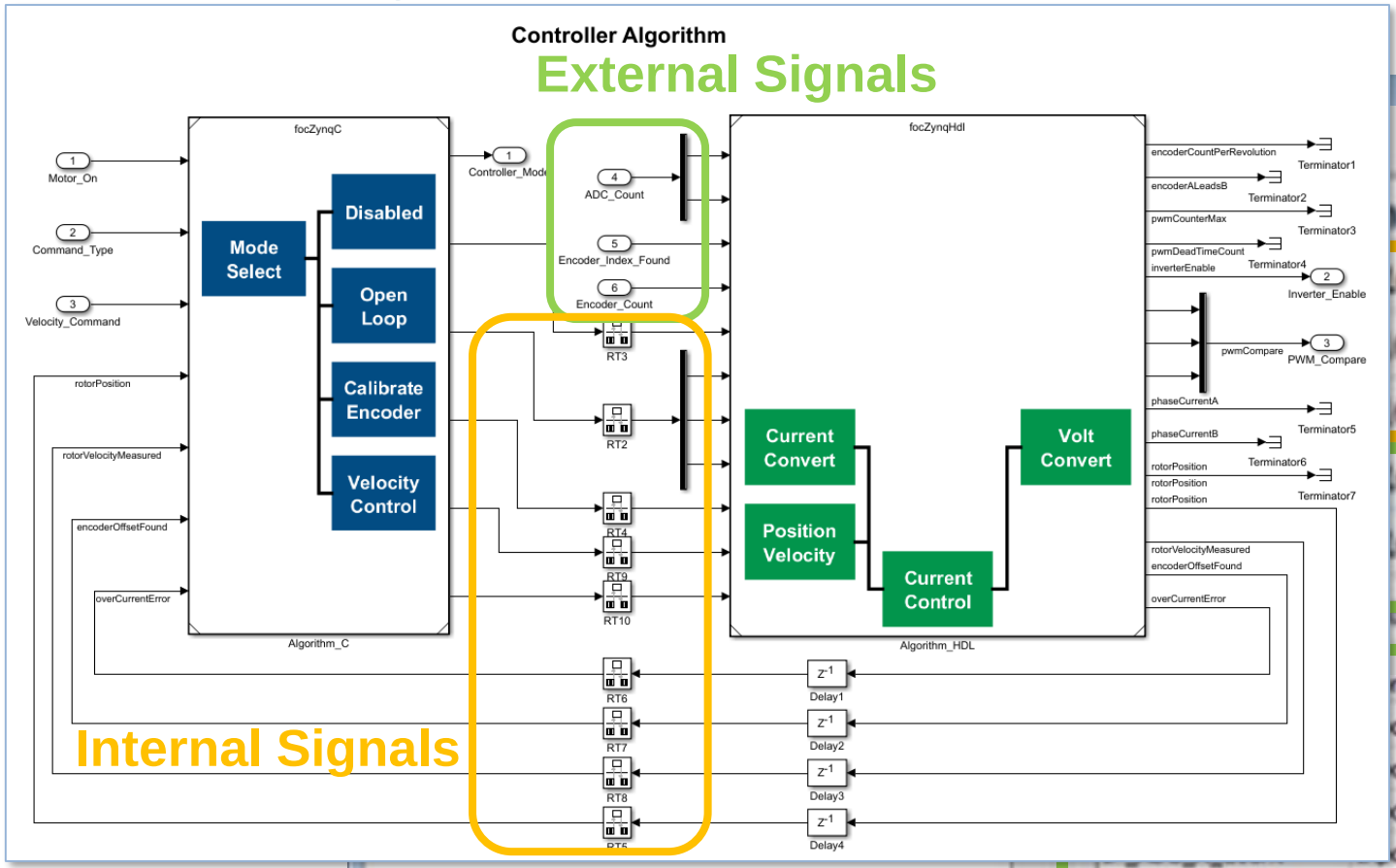
Designing and Prototyping Digital Systems on SoC FPGAs
16:30–17:15



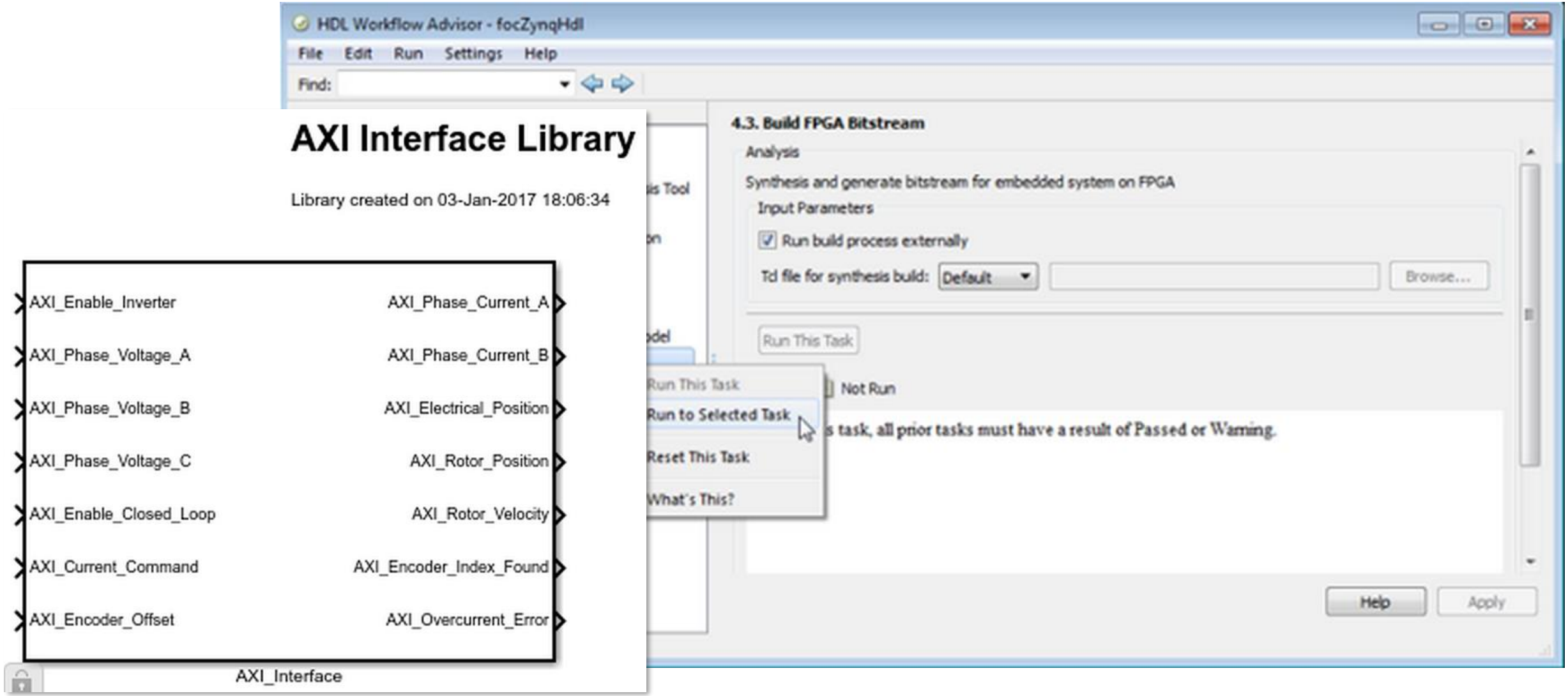
Deploy Bitstream to Programmable Logic



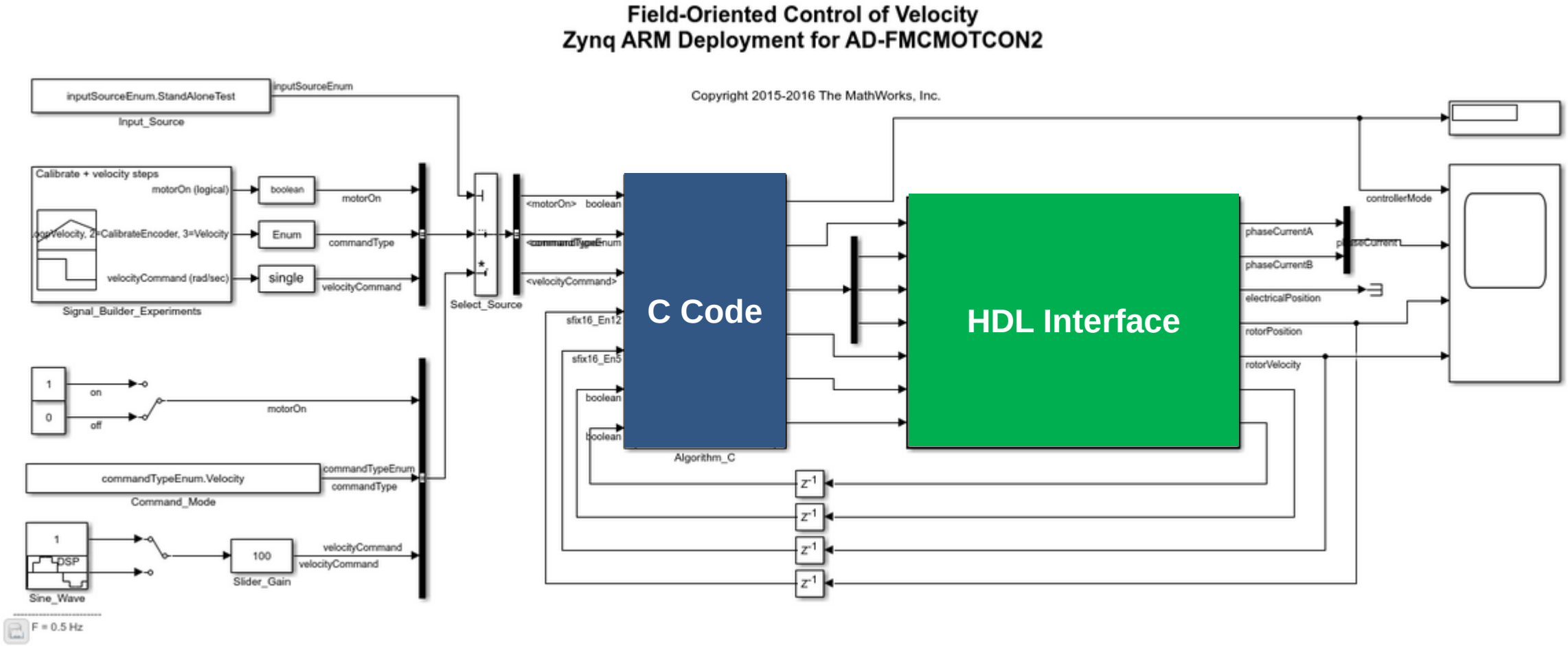
Set Target Interface for Bitstream

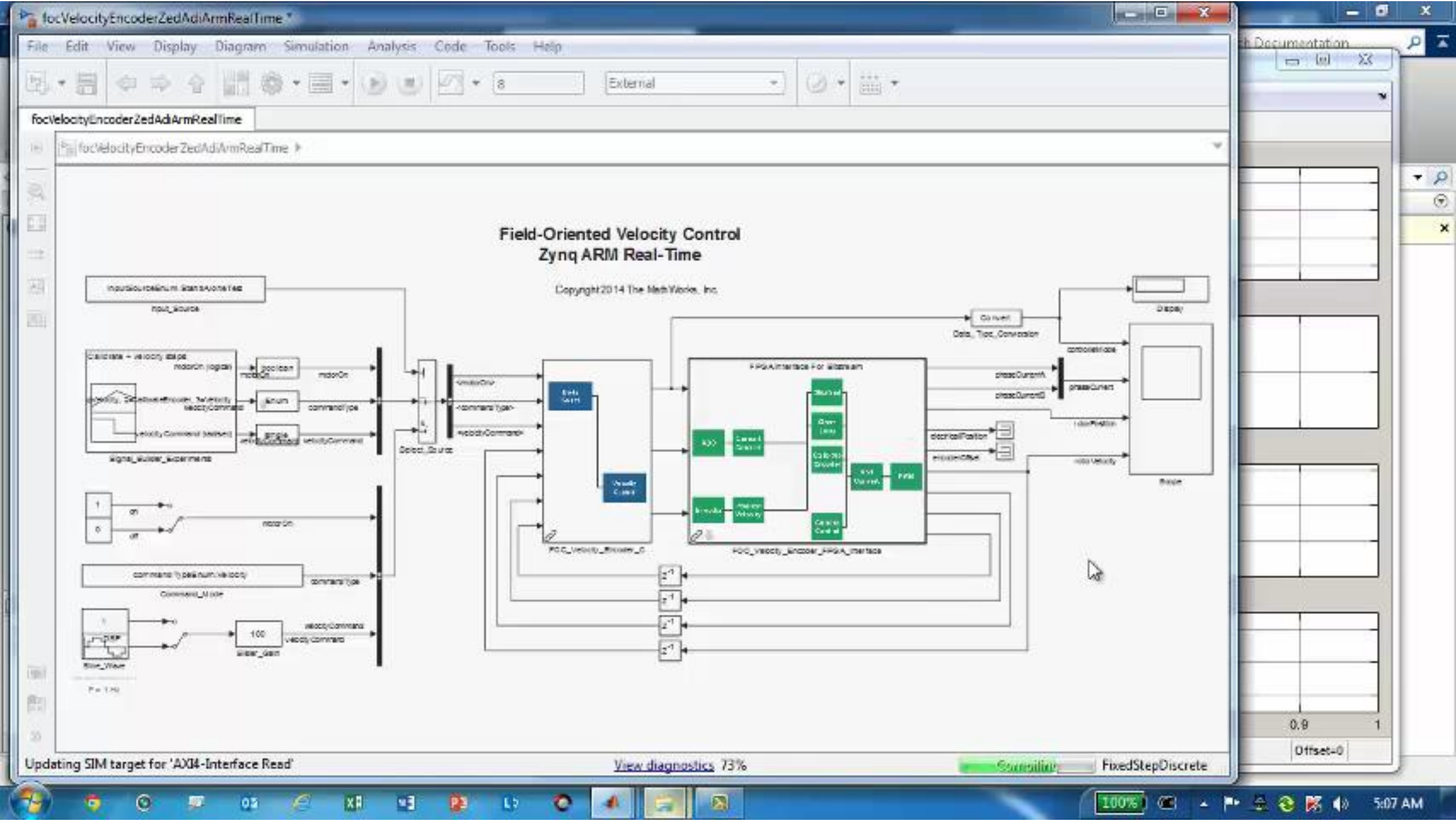


Build FPGA Bitstream



Zynq ARM Deployment




focVelocityEncoderZedAdiArmRealTime *
File Edit View Display Diagram Simulation Analysis Code Tools Help
External
focVelocityEncoderZedAdiArmRealTime
Field-Oriented Velocity Control
Zynq ARM Real-Time
Copyright 2014 The MathWorks, Inc.
Updating SIM target for 'AXI4-Interface Read'
View diagnostics 73%
FixedStepDiscrete
5:07 AM

3T Develops Robot Emergency Braking System with Model-Based Design

Challenge

Design and implement a robot emergency braking system with minimal hardware testing

Solution

Model-Based Design with Simulink and HDL Coder to model, verify, and implement the controller

Results

- Cleanroom time reduced from weeks to days
- Late requirement changes rapidly implemented
- Complex bug resolved in one day

A SCARA robot.

"With Simulink and HDL Coder we eliminated programming errors and automated delay balancing, pipelining, and other tedious and error-prone tasks. As a result, we were able to easily and quickly implement change requests from our customer and reduce time-to-market."

Ronald van der Meer

3T

Link to user story

Why use Model-Based Design to develop motor control applications on SoCs?

Challenges:

- Integration of software and hardware partitions of algorithm on SoC drives need for collaboration
- Validation of design specifications with limits on access to motors in labs.
- How to make design decisions that cut across system?

Model-Based Design

- ✓ Enables early validation of specifications using simulation months before hardware is available.
- ✓ Improves design team collaboration and designer productivity by using a shared design environment.
- ✓ Reduces hardware testing time by 5x by shifting design from lab to the desktop

Learn More

- Visit us in the Technology Showcase
  - Field-Oriented-Control based Motor Control Application using System-on-Chip (SoC) Architecture
  - Achieve Industry & Safety Standards Compliance using Efficient Model Verification & Validation and Production Code Generation
  - Model-Based Design for Software-Defined Radio
- Videos
  - HDL Coder: Native Floating Point
- Webinars
  - Prototyping SoC-based Motor Controllers on Intel SoCs with MATLAB and Simulink
  - How to Build Custom Motor Controllers for Zynq SoCs with MATLAB and Simulink
- Articles
  - How Modeling Helps Embedded Engineers Develop Applications for SoCs (MATLAB Digest)
  - MATLAB and Simulink Aid HW-SW Codesign of Zynq SoCs (Xcell Software Journal)
- Tutorials:
  - Define and Register Custom Board and Reference Design for SoC Workflow
  - Field-Oriented Control of a Permanent Magnet Synchronous Machine on SoCs

Speaker Details

Email: Durvesh.Kulkarni@mathworks.in

LinkedIn: https://www.linkedin.com/in/durvesh-kulkarni-17402527/

Speaker Details

Email: Gaurav.Dubey@mathworks.in

LinkedIn:https://www.linkedin.com/in/gauravdubey4/

- **Share your experience with MATLAB & Simulink on Social Media**
  - Use #MATLABEXPO
  - I use #MATLAB because…… Attending #MATLABEXPO
  - Examples
    - I use #MATLAB because it helps me be a data scientist! Attending #MATLABEXPO
    - Learning new capabilities in #MATLAB and #Simulink at #MATLABEXPO.

- **Share your session feedback:**

  Please fill in your feedback for this session in the feedback form

Contact MathWorks India

Products/Training Enquiry Booth

Call: 080-6632-6000

Email: info@mathworks.in