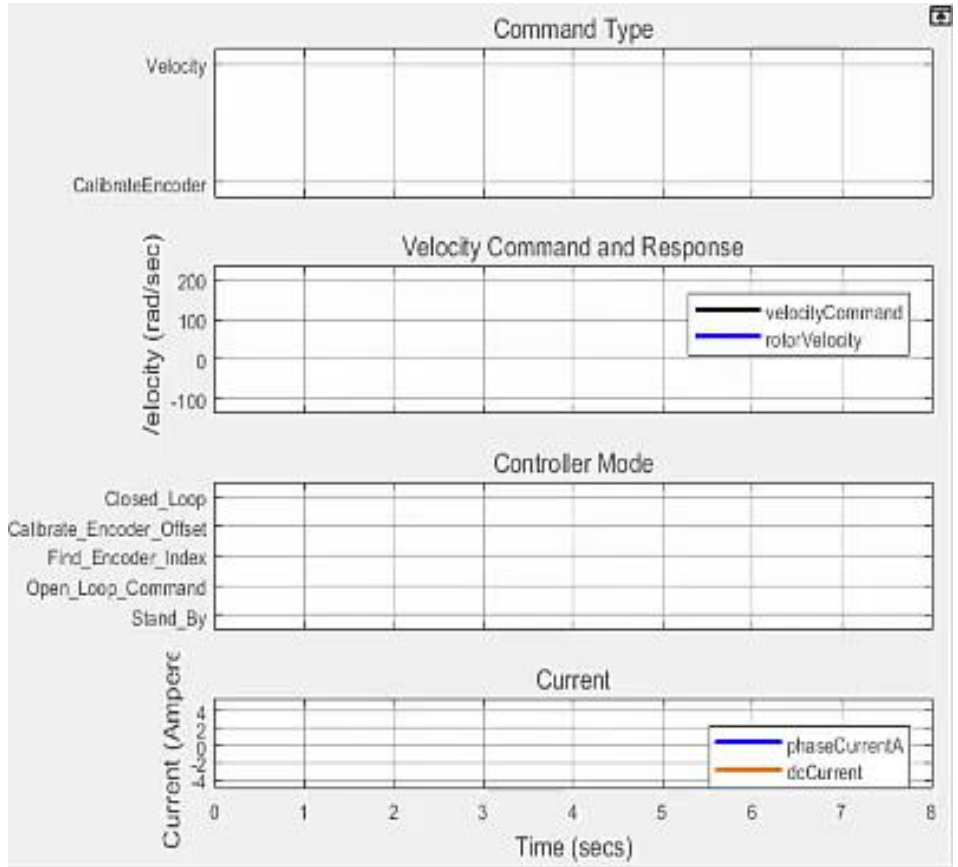
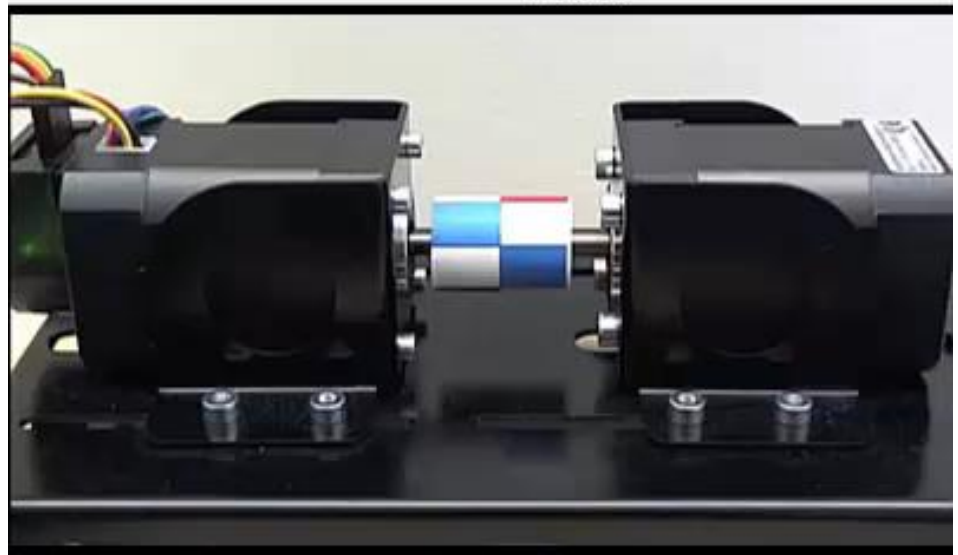
Masterclass

MATLAB EXPO 2018

Hardware and Software Co-Design for Motor Control Applications

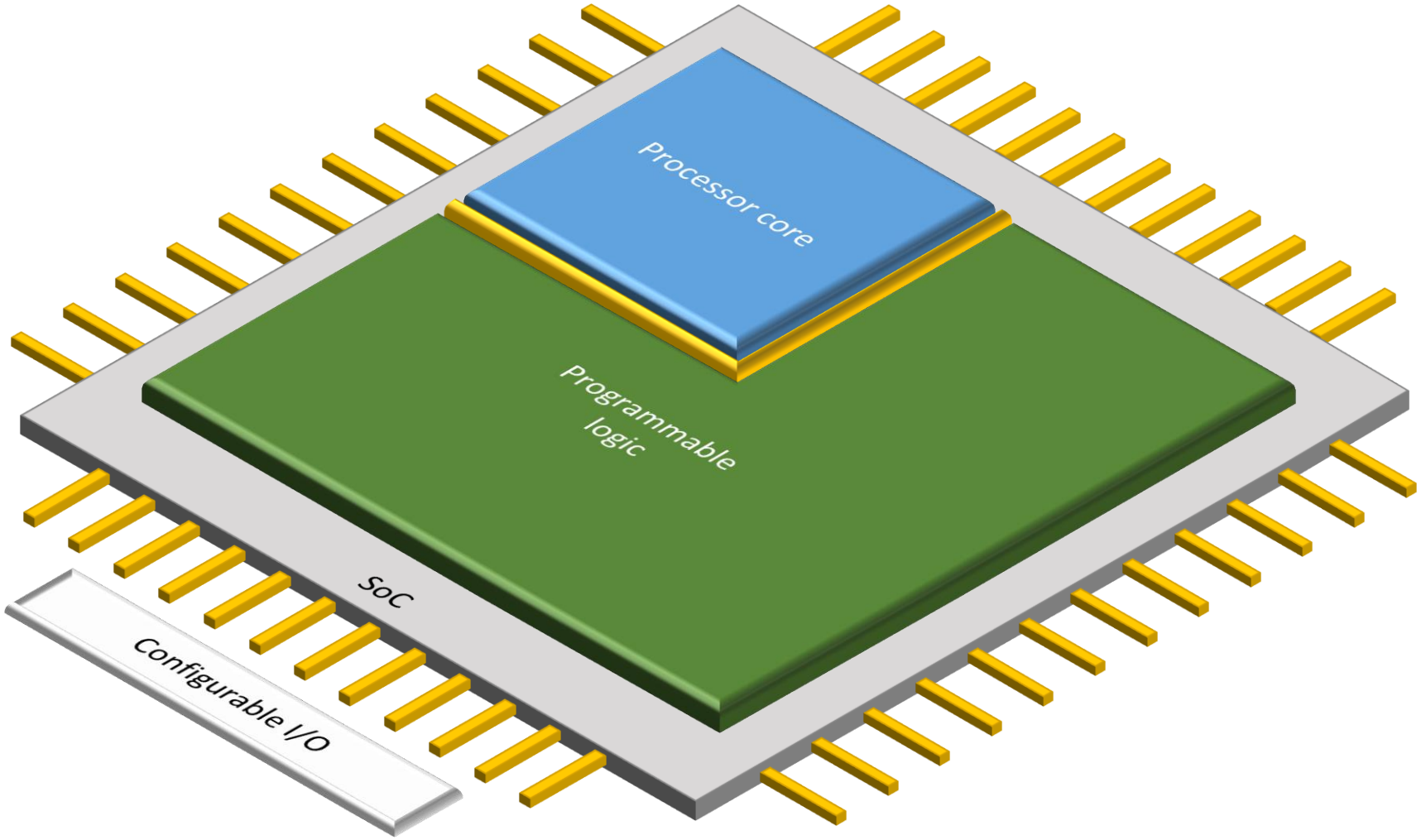
Jonas Rutström
Application Engineering





Masterclass vs. Presentation?

What's a SoC?



What's a SoC?

*"When we refer to SoCs, we're referring to an **FPGA** device equipped with a **hard processor**."*

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

Agenda

- Trends, challenges and MBD in motor control applications
- CASE Study: *Field-Oriented Control of Velocity on Xlilinx Zynq SoC*
- Debugging your design using *Data Capture* R2018a
- Next step?

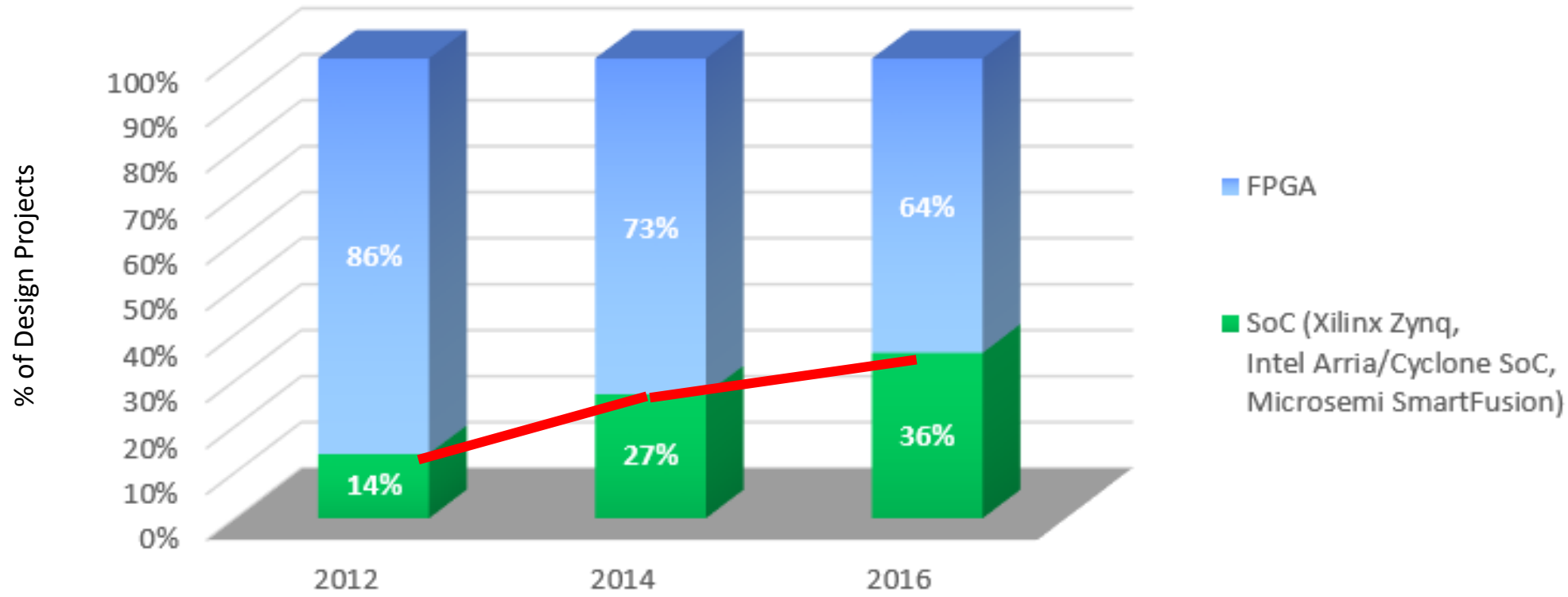
Agenda

- Trends, challenges and MBD in motor control applications
- CASE Study: *Field-Oriented Control of Velocity on Xlilinx Zynq SoC*
- Debugging your design using *Data Capture* R2018a
- Next step?

Key Trend 1: Increasing demands from motor drives

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

Key Trend 2: SoCs are now used in 36% of new FPGA projects

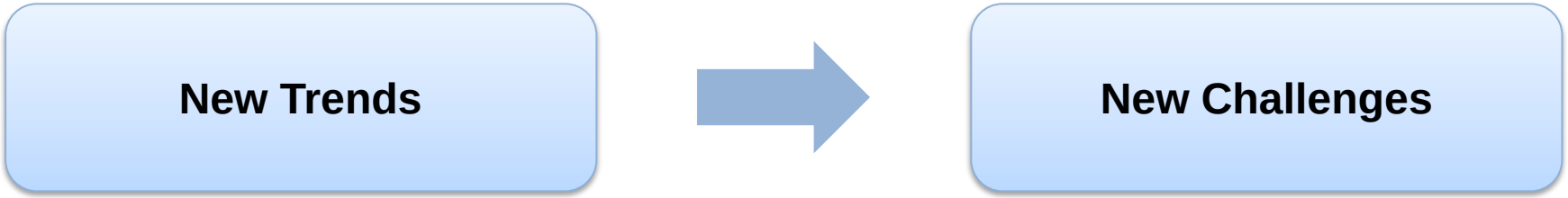


Source: Wilson Research Group and Mentor Graphics, 2016 Functional Verification Study

Why use a SoC?

- Soft processors like Nios or Microblaze are more limited in terms of their maximum **clock speed**
- Configurable input/output pins and ADCs
- Ability to perform **floating-point** math
- They are more **power efficient**
- **Programmable logic** to create custom I/Os or offload compute-intensive tasks
- Memory-mapped streaming between FPGA and processor
- Dedicated **DSP blocks**
- SoC are fabricated onto the **same die**

Challenges in using SoCs for Motor and Power Control



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

Any idea of how to approach these challenges?

Model-Based Design with MATLAB and Simulink will help you develop motor control applications on SoCs by…

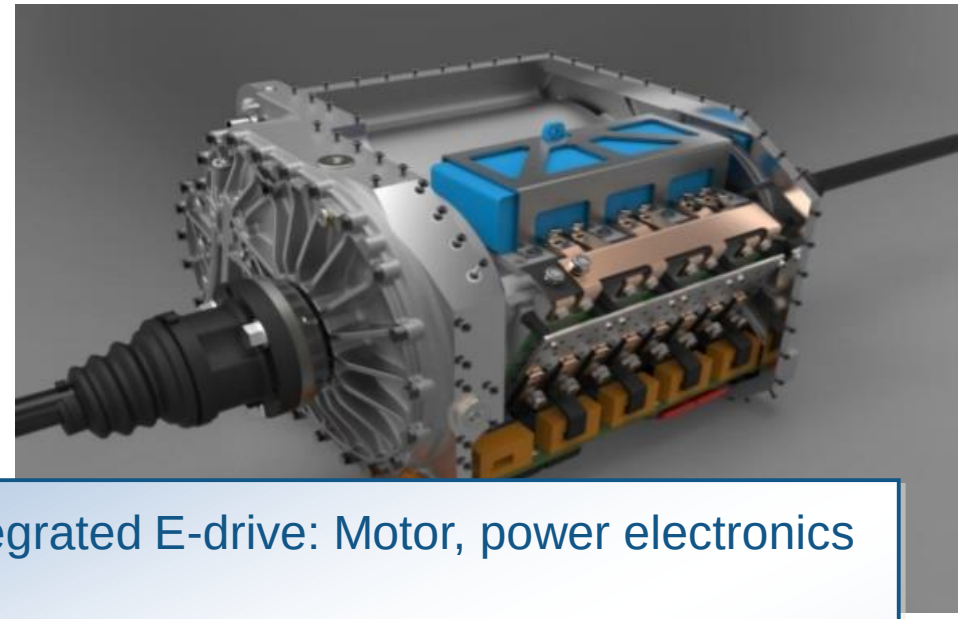
- Enable early validation of specifications using simulation months before hardware is available.
- Improving design team collaboration and designer productivity by using a shared design environment.
- Reducing hardware testing time by shifting design from lab to the desktop

User Stories?

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!

Link to video of presentation



- ✓ 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

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

Link to user story

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

Agenda

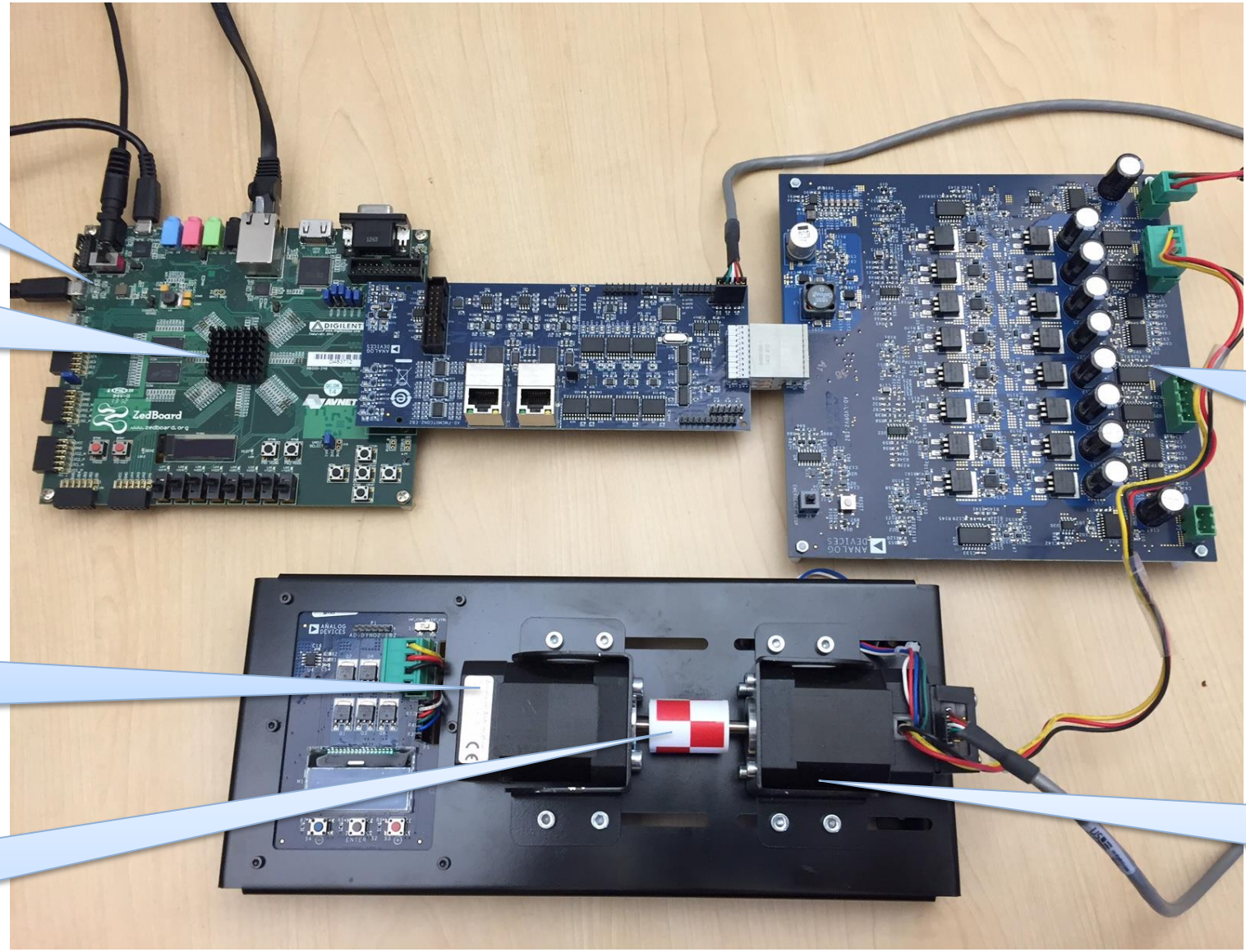
- Trends, challenges and MBD in motor control applications
- CASE Study: *Field-Oriented Control of Velocity on Xlilinx Zynq SoC*
- Debugging your design using *Data Capture* R2018a
- Next step?

ZedBoard

Zynq SoC
(XC7Z020)

Load motor

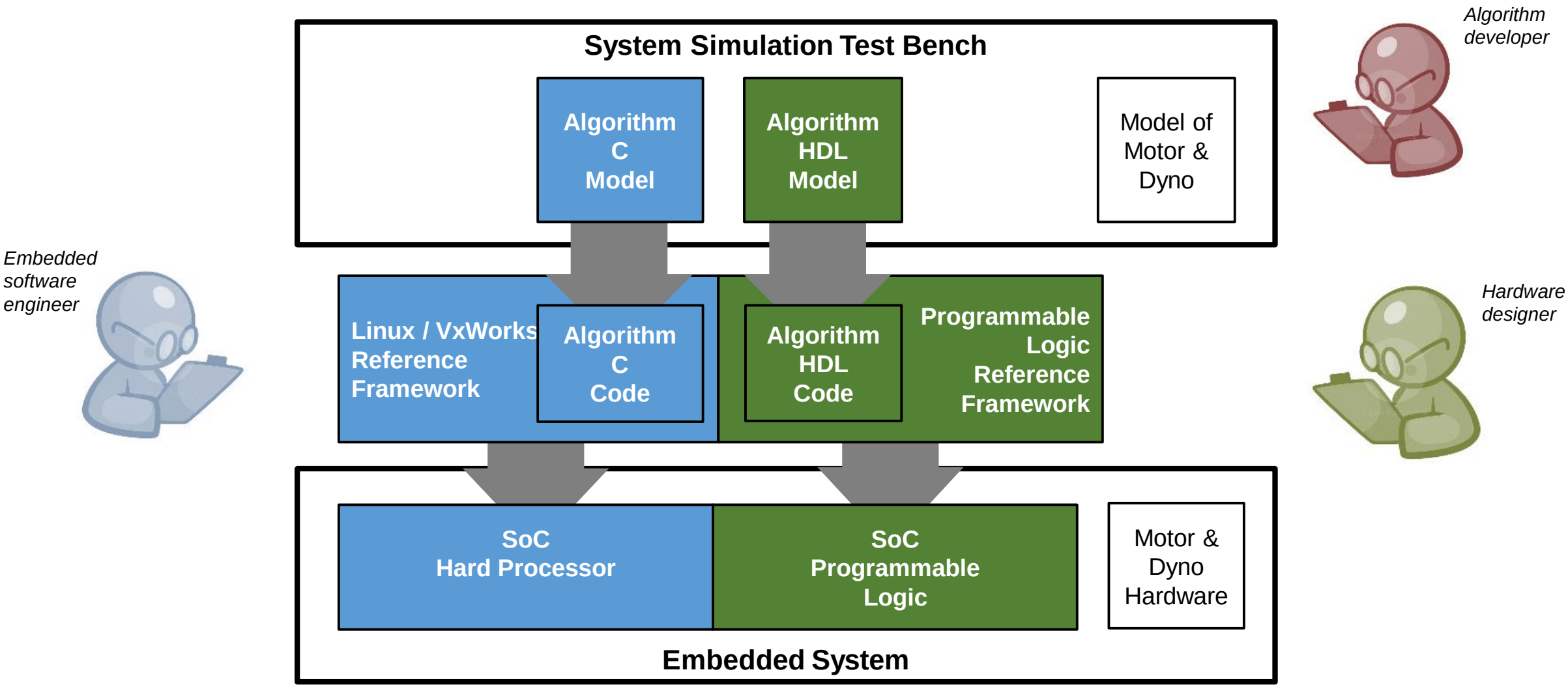
Mechanical coupler



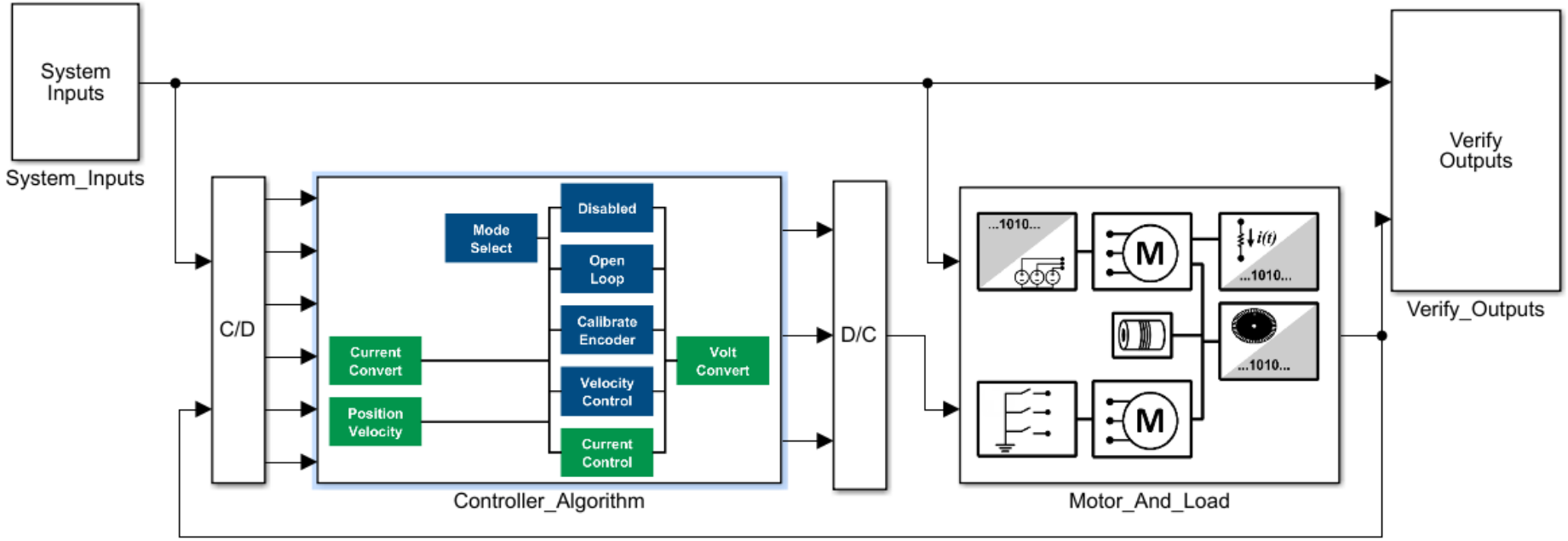
FMC module: control board + low-voltage board

Motor under test (with encoder)

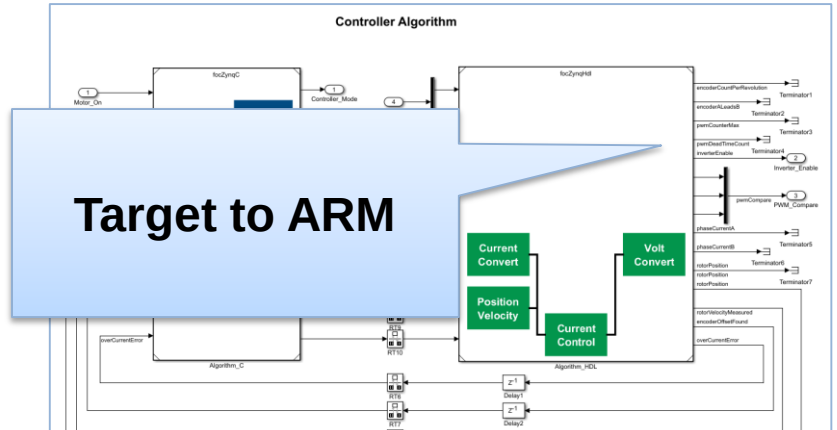
Conceptual workflow targeting SoCs



Hardware/software partitioning

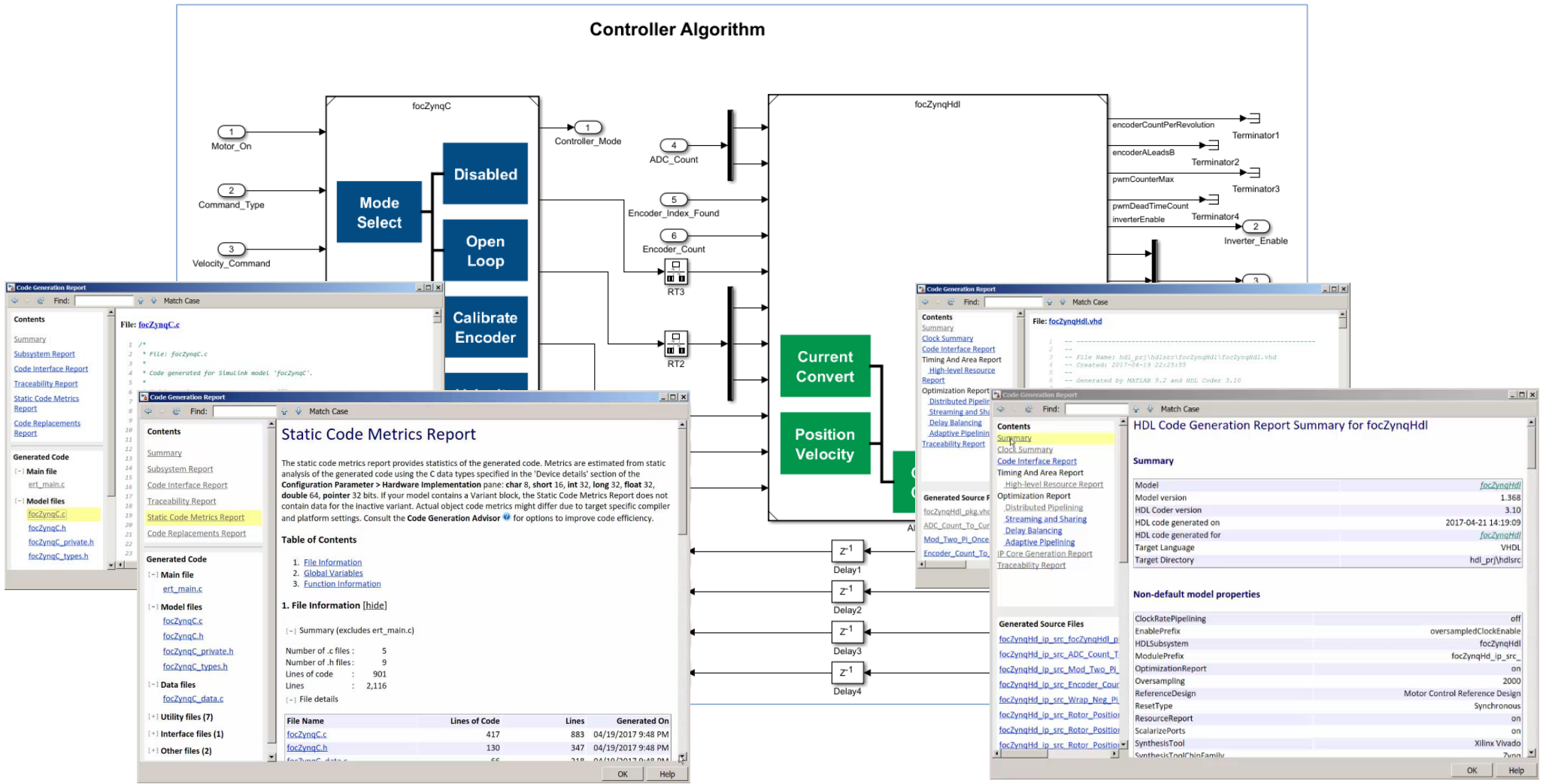


Target to Programmable Logic



Target to ARM

Code Generation



DEMO

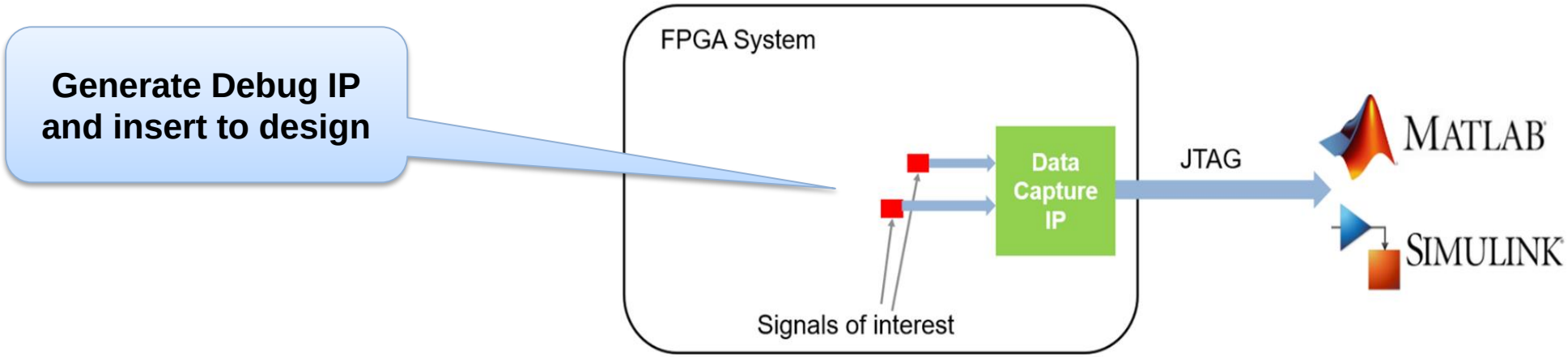
Field-Oriented Control of Velocity

Agenda

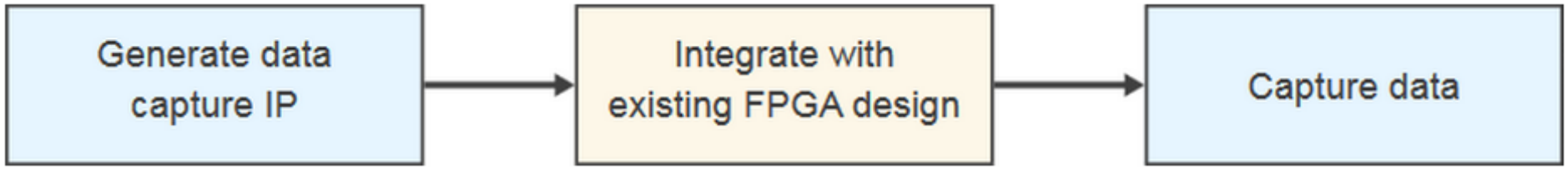
- Trends, challenges and MBD in motor control applications
- CASE Study: *Field-Oriented Control of Velocity on Xlilinx Zynq SoC*
- Debugging your design using *Data Capture* R2018a
- Next step?

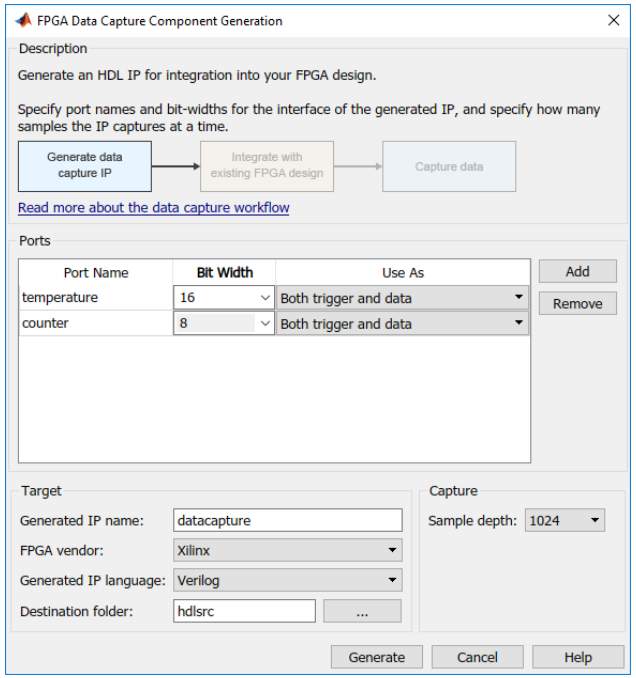
Debugging using Data Capture

- Why would you like to capture signals from the FPGA?
  - It enables you to use MATLAB to **analyze the real-world signals**.
  - It enables you to capture data and drive it to simulations which **makes design debugging easier**.



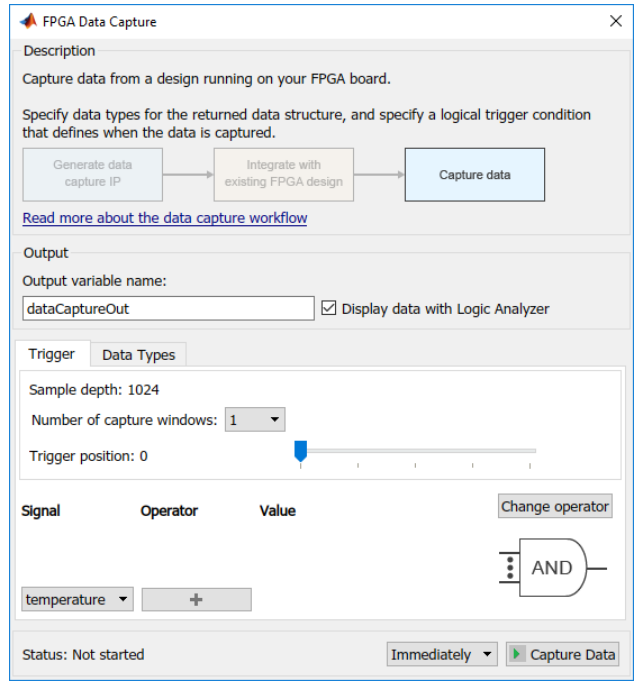
Data Capture IP Workflow






```
datacapture u0 (
  .clk(clk),
  .clk_enable(clk_enable),
  .ready_to_capture(ready_to_capture),
  .data_in(data_in));
```

Added one output port to the existing Data Capture IP



DEMO

Debugging using Data Capture

Agenda

- Trends, challenges and MBD in motor control applications
- CASE Study: *Field-Oriented Control of Velocity on Xlilinx Zynq SoC*
- Debugging your design using *Data Capture* R2018a
- Next step?

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

Challenges:

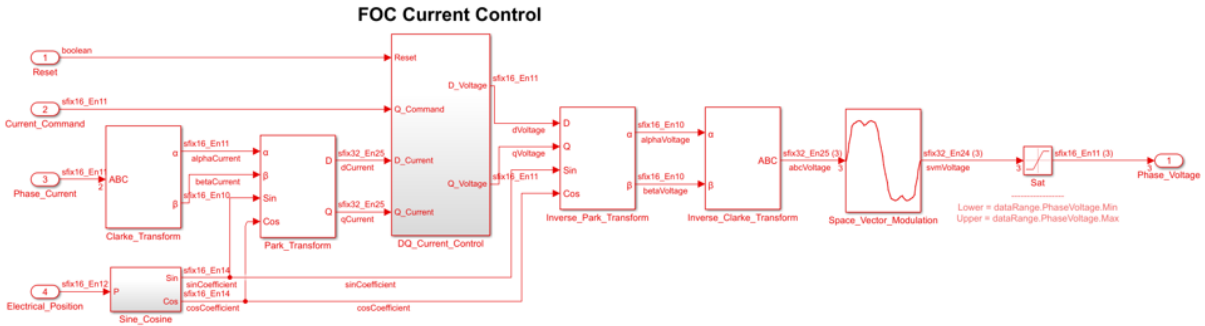
- Integration of software and hardware partitions of algorithm on SoC drives need for collaboration
- Validation of design specifications with limits on access to motors in labs.
- How to make design decisions that cut across system?

Model-Based Design

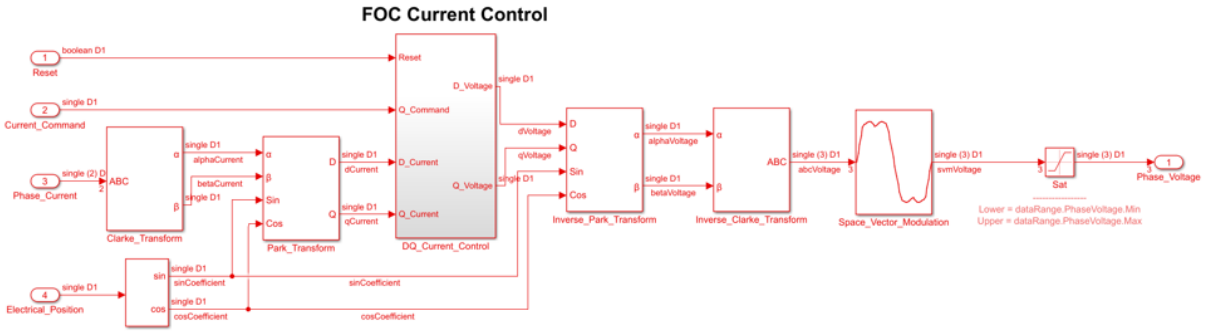
- ✓ Enables early validation of specifications using simulation months before hardware is available.
- ✓ Improves design team collaboration and designer productivity by using a shared design environment.
- ✓ Reduces hardware testing time by shifting design from lab to the desktop

Learn More

- Visit us in the Technology Showcase
  - New: see award-winning Native Floating Point in HDL Coder



Fixed-point design



Floating-point design

MathWorks
58.426 volgers
16 d

MathWorks is honored to receive the Embedded World Award 2017 in the Tools Category for HDL Coder. http://owl.li/nBzd309XYxW

288 interessant • 6 commentaren

Learn More

- Visit us in the Technology Showcase
  - New: see award-winning Native Floating Point in HDL Coder
- 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

Thank You!