

The slide features a dark blue background with a light blue geometric shape on the right side. Inside this shape, there are white waveforms at the top and a colorful grid pattern (yellow, green, blue) at the bottom. The text is white and positioned on the left side of the slide.

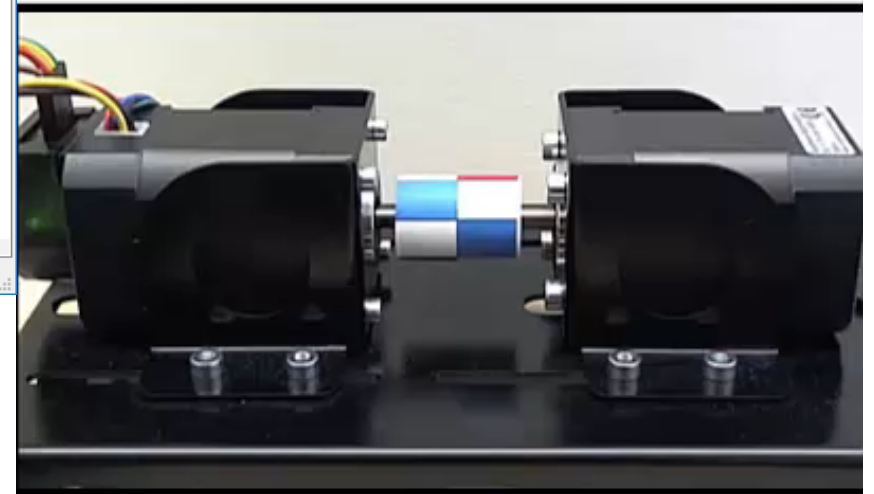
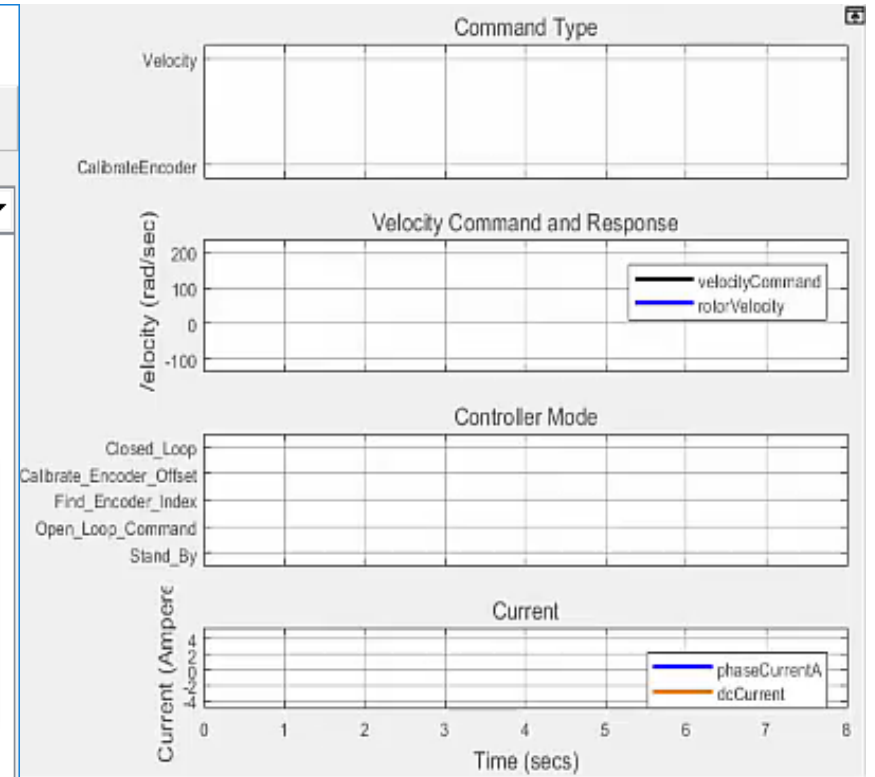
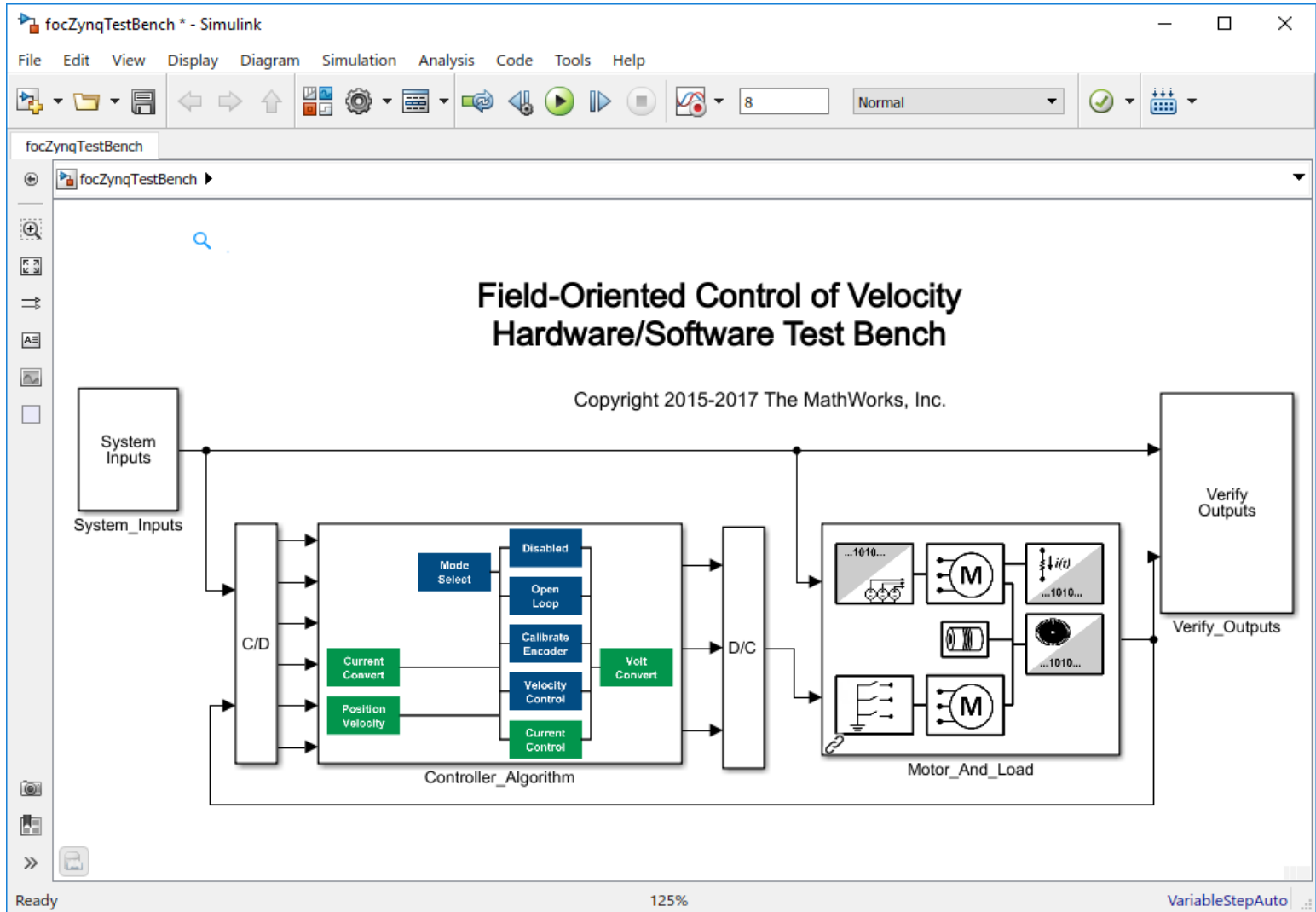
MATLAB EXPO 2017

Targeting Motor Control Algorithms to System-on-Chip Devices

Dr.-Ing. Werner Bachhuber

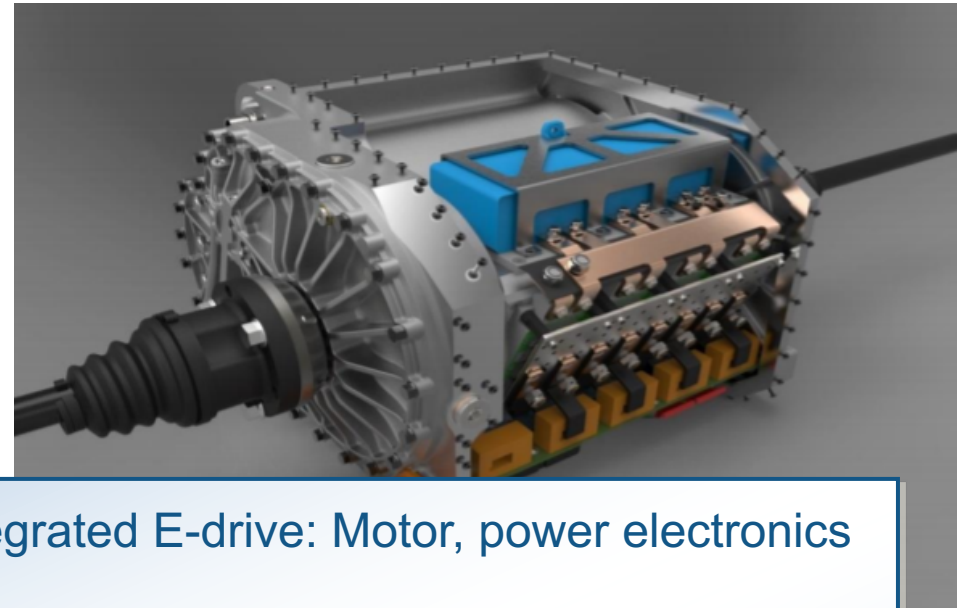
Why use Model-Based Design to develop motor control applications on SoCs?

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



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
- Needed to get to market quickly
- No experience designing FPGAs!



- ✓ Designed integrated E-drive: Motor, power electronics and software
- ✓ 4 different control strategies implemented
- ✓ Done 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](#)

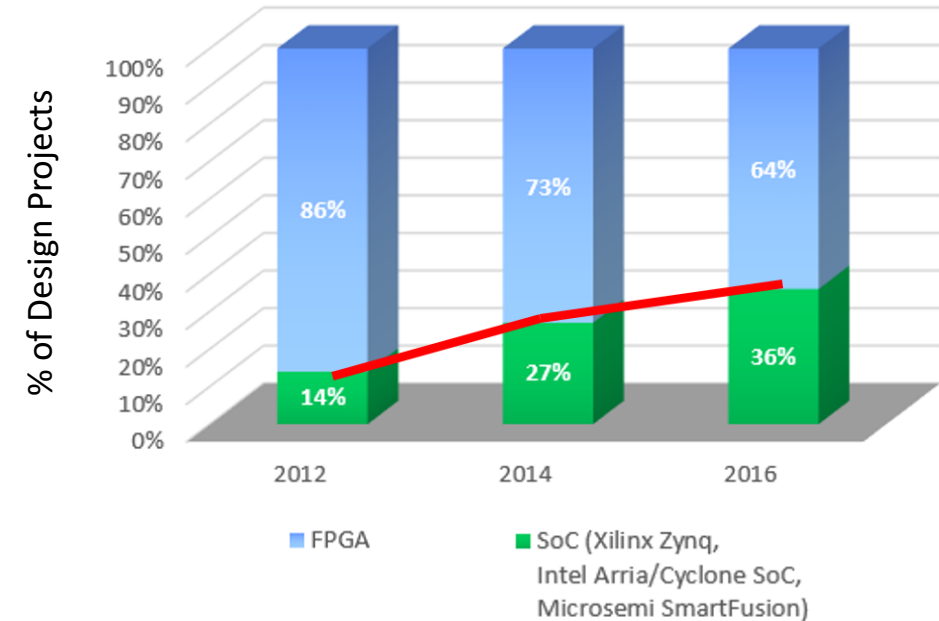
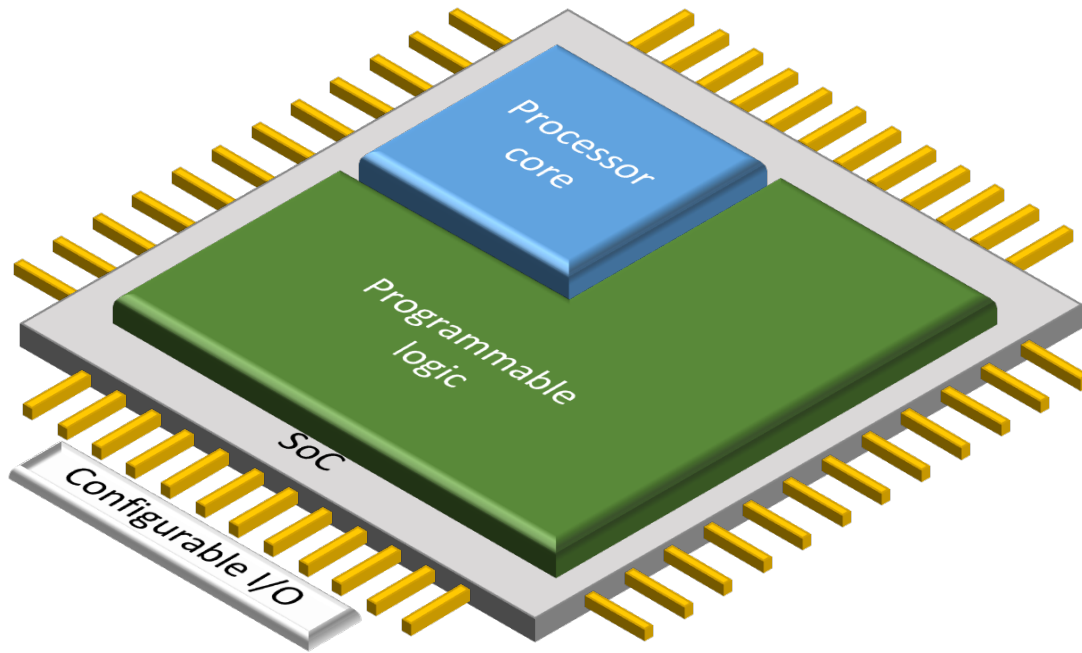
Key trend: Increasing demands from motor drives

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



What's an SoC?

SoC Key Trend:



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

Challenges in using SoCs for Motor and Power Control

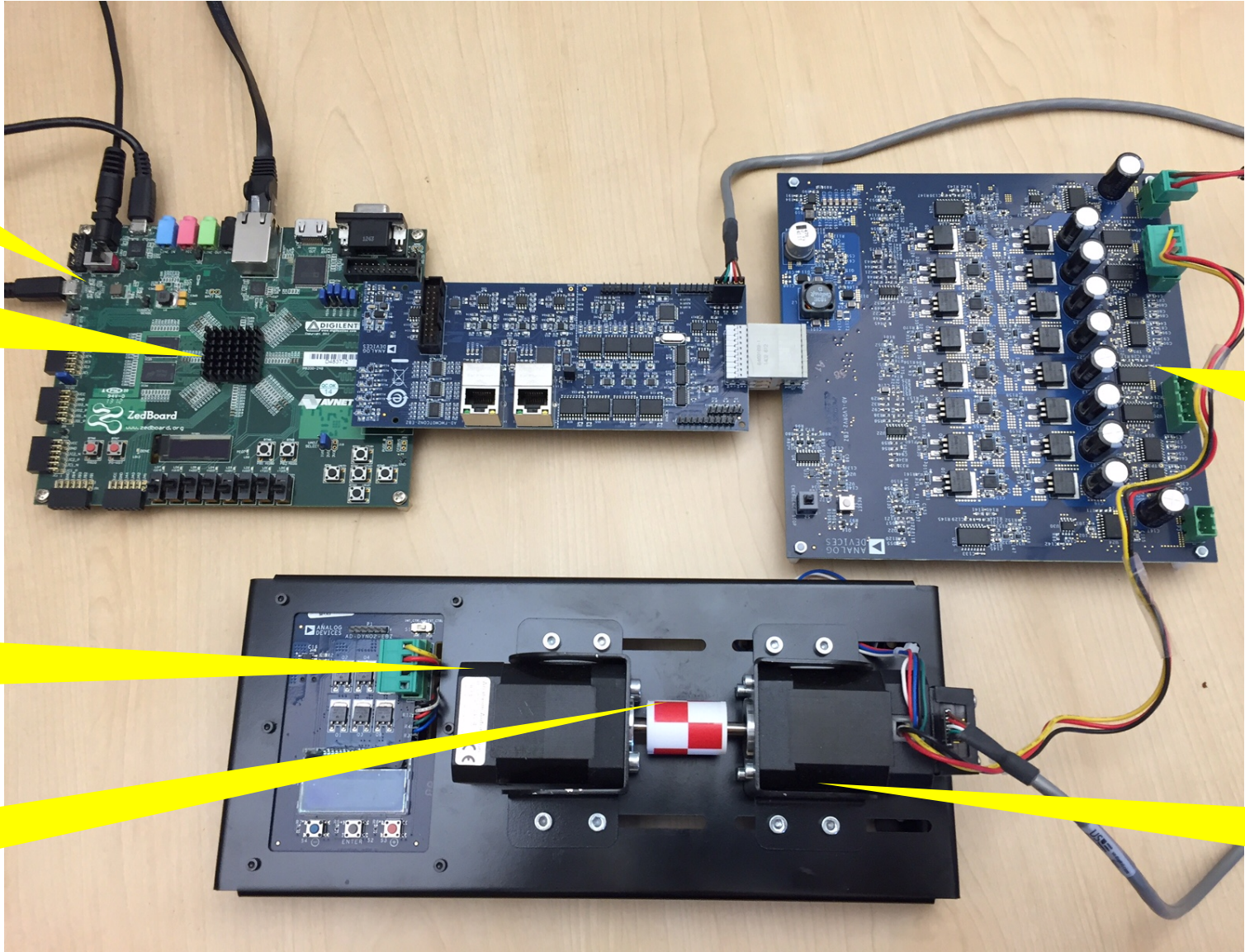
- Integration requires collaboration
- Validation of design specifications with limits on access to test hardware
- How to make design decisions?

ZedBoard

**Zynq SoC
(XC7Z020)**

Load motor

**Mechanical
coupler**

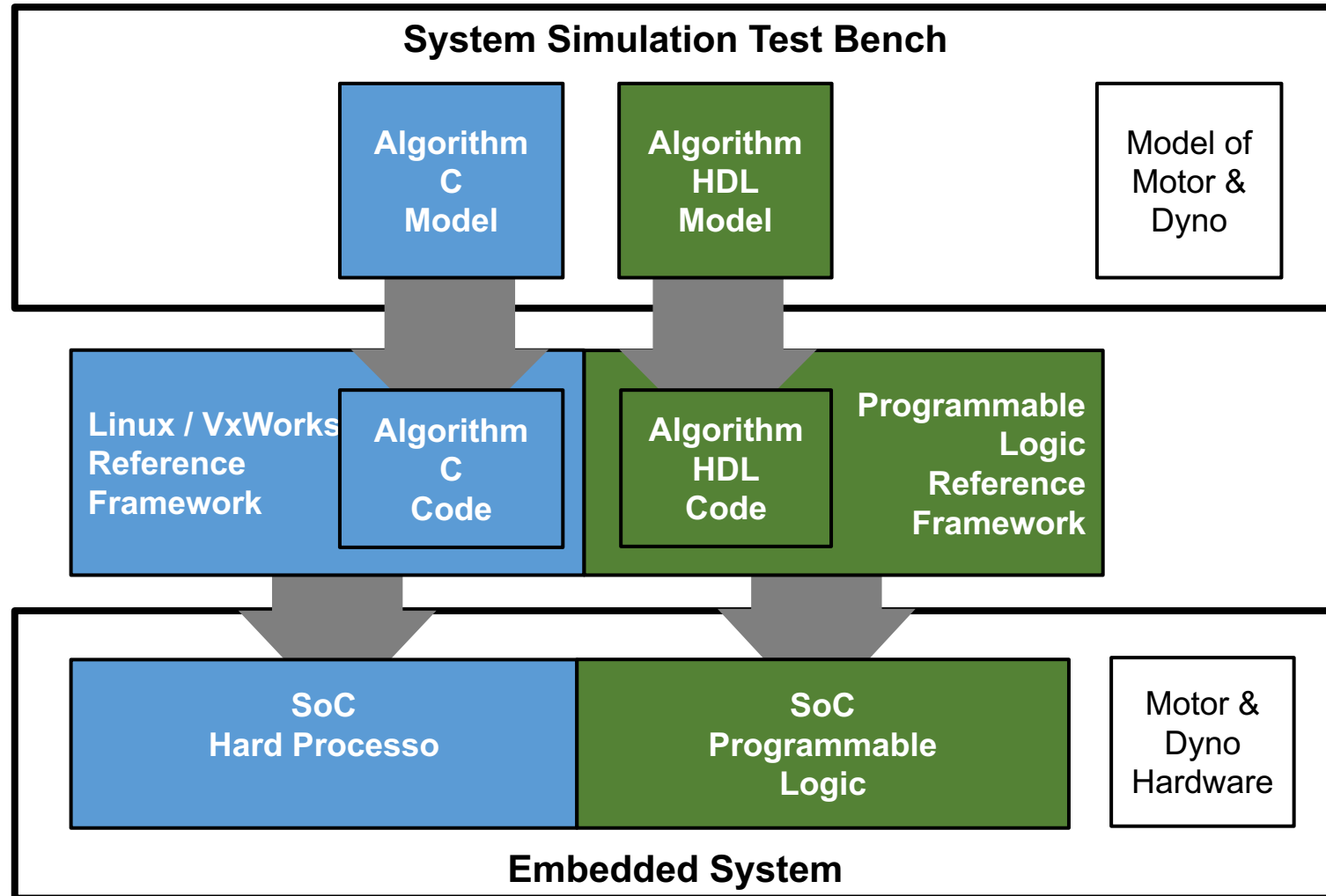
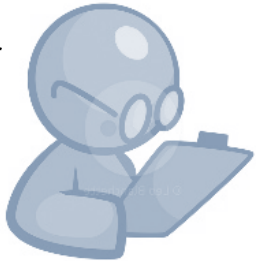


**FMC module:
control board +
low-voltage board**

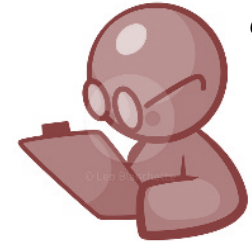
**Motor under test
(with encoder)**

Conceptual workflow targeting SoCs

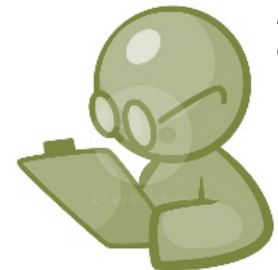
Embedded software engineer



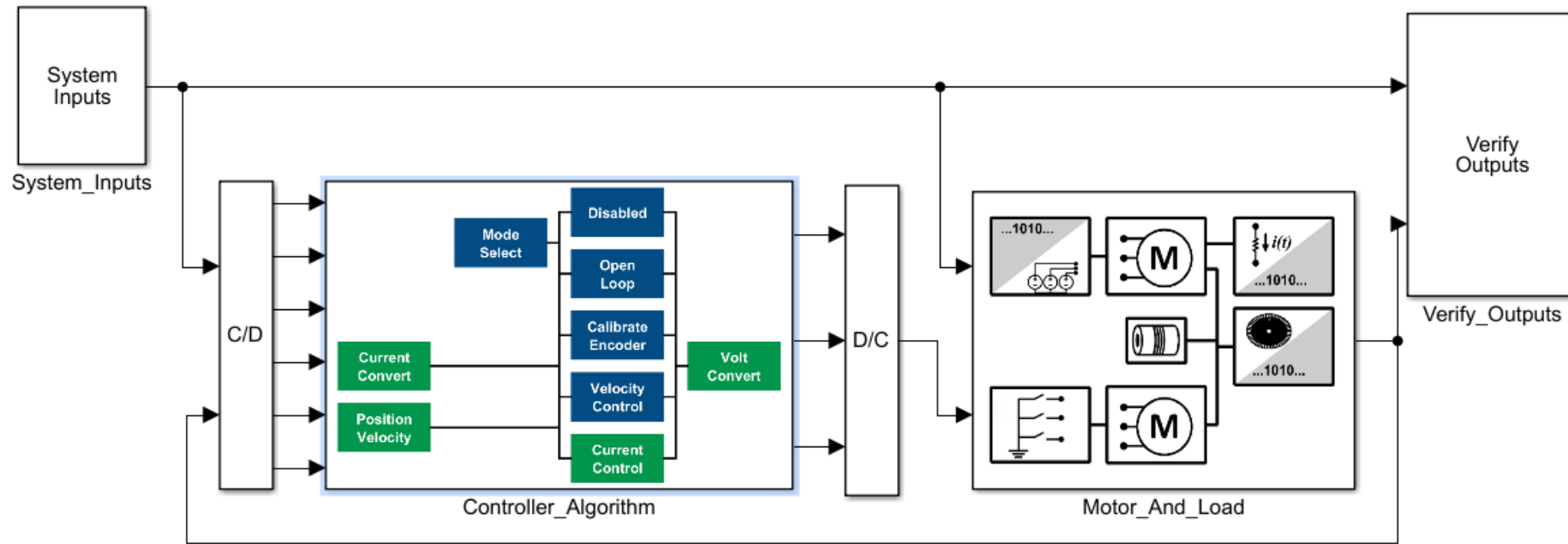
Algorithm developer



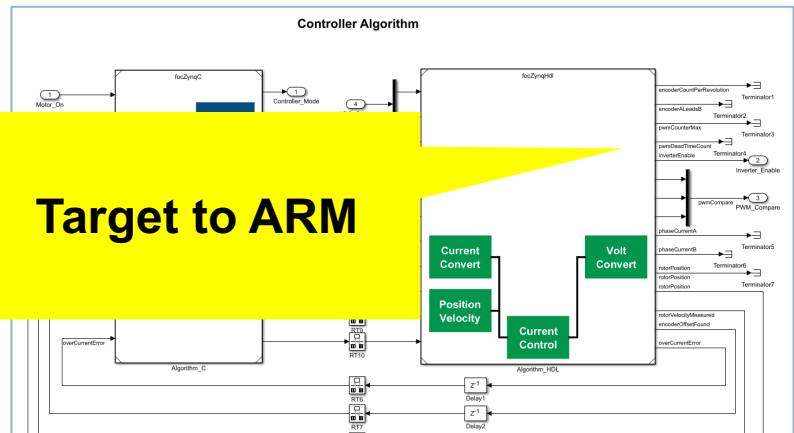
Hardware designer



Hardware/software partitioning



Target to Programmable Logic



Code Generation

Controller Algorithm

Code Generation Report

File: focZynqC.c

Contents:

- Summary
- Subsystem Report
- Code Interface Report
- Traceability Report
- Static Code Metrics Report
- Code Replacements Report

Generated Code:

- ert_main.c
- focZynqC.c
- focZynqC.h
- focZynqC_private.h
- focZynqC_types.h

Static Code Metrics Report

The static code metrics report provides statistics of the generated code. Metrics are estimated from static analysis of the generated code using the C data types specified in the 'Device details' section of the Configuration Parameter > Hardware Implementation pane: char 8, short 16, int 32, long 32, float 32, double 64, pointer 32 bits. If your model contains a Variant block, the Static Code Metrics Report does not contain data for the inactive variant. Actual object code metrics might differ due to target specific compiler and platform settings. Consult the Code Generation Advisor for options to improve code efficiency.

Table of Contents

- File Information
- Global Variables
- Function Information

1. File Information [hide]

[...] Summary (excludes ert_main.c)

File Name	Lines of Code	Lines	Generated On
focZynqC.c	417	883	04/19/2017 9:48 PM
focZynqC.h	130	347	04/19/2017 9:48 PM
focZynqC_private.h	66	218	04/19/2017 9:48 PM

Code Generation Report

File: focZynqHdl.vhd

Contents:

- Summary
- Clock Summary
- Code Interface Report
- Timing And Area Report
- High-level Resource Report
- Optimization Report
- Distributed Pipelining
- Streaming and Sharing
- Delay Balancing
- Adaptive Pipelining
- Traceability Report

Generated Source Files:

- focZynqHdl_pkg.vhd
- ADC_Count_To_Current.vhc
- Mod_Two_Pl_Once.vhd
- Encoder_Count_To_Rotor_J.vhd

HDL Code Generation Report Summary for focZynqHdl

Summary

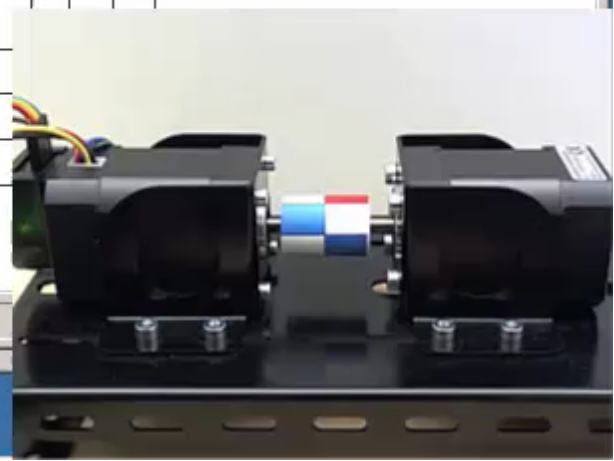
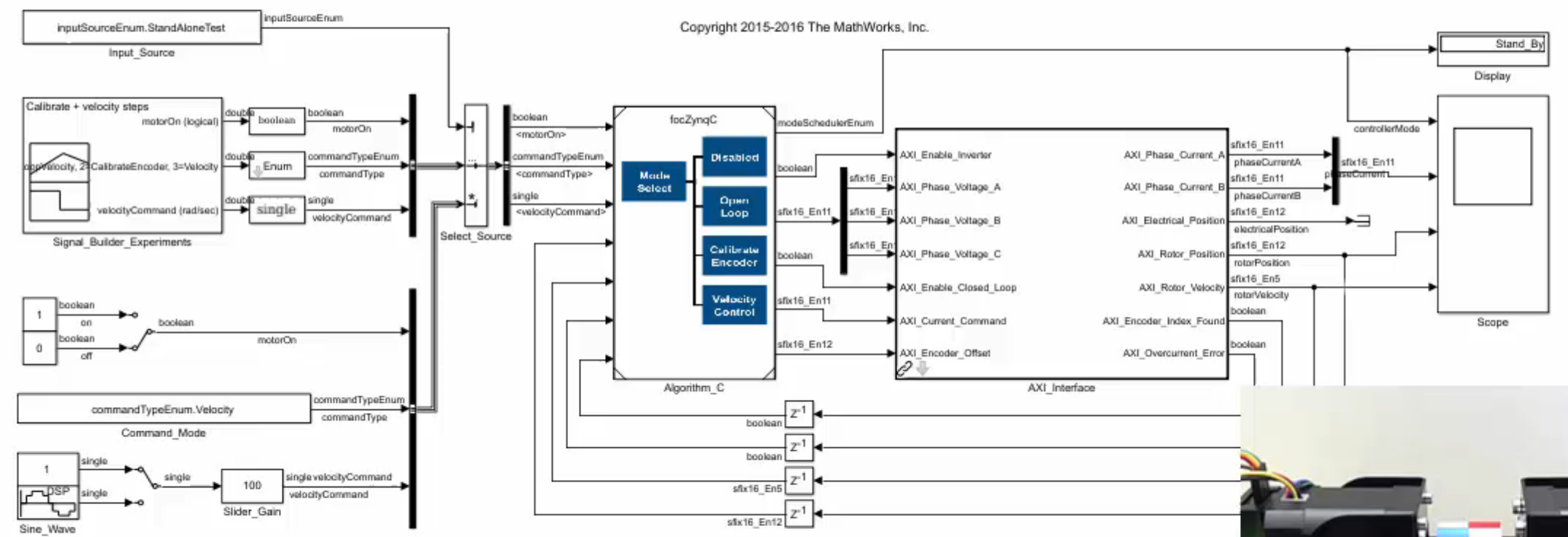
Model	focZynqHdl
Model version	1.368
HDL Code version	3.10
HDL code generated on	2017-04-21 14:19:09
HDL code generated for	focZynqHdl
Target Language	VHDL
Target Directory	hdl_prj\hdlsrc

Non-default model properties

ClockRatePipelining	off
EnablePrefix	oversampledClockEnable
HDLSubsystem	focZynqHdl
ModulePrefix	focZynqHdl_ip_src_
OptimizationReport	on
Oversampling	2000
ReferenceDesign	Motor Control Reference Design
ResetType	Synchronous
ResourceReport	on
ScalarizePorts	on
SynthesisTool	Xilinx Vivado
SynthesisToolChinFamily	zynq

Field-Oriented Control of Velocity Zynq ARM Deployment for AD-FMCMOTCON2

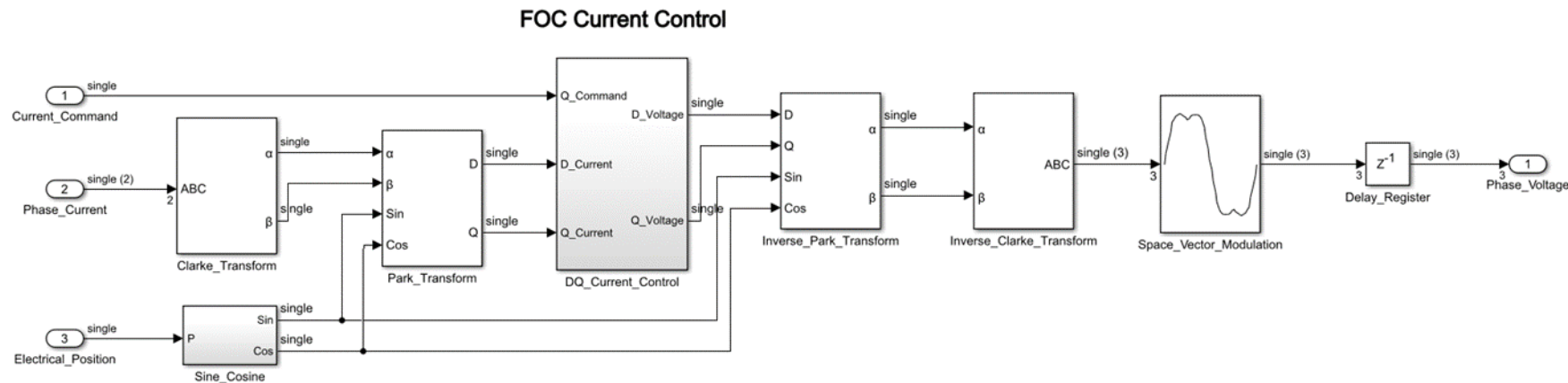
Copyright 2015-2016 The MathWorks, Inc.



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

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

New: Award-Winning Native Floating Point



- Vendor-independent VHDL/Verilog for FPGA and ASIC design
- Full range of IEEE-754 features
 - Optional support for Denormals, INF, NAN, Rounding, ...
- Extensive Math and Trigonometry Block support
- Videos
 - [HDL Coder: Native Floating Point](#)

Learn More

- Get an in-depth demo in the Technology Showcase
- 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](#)
- Training
 - [Entwicklung von HDL Code aus Simulink](#)
 - [Programmierung von Xilinx Zynq SoCs mit MATLAB und Simulink](#)
 - [DSP für FPGAs](#)