MathWorks®

# MATLAB EXPO 2018

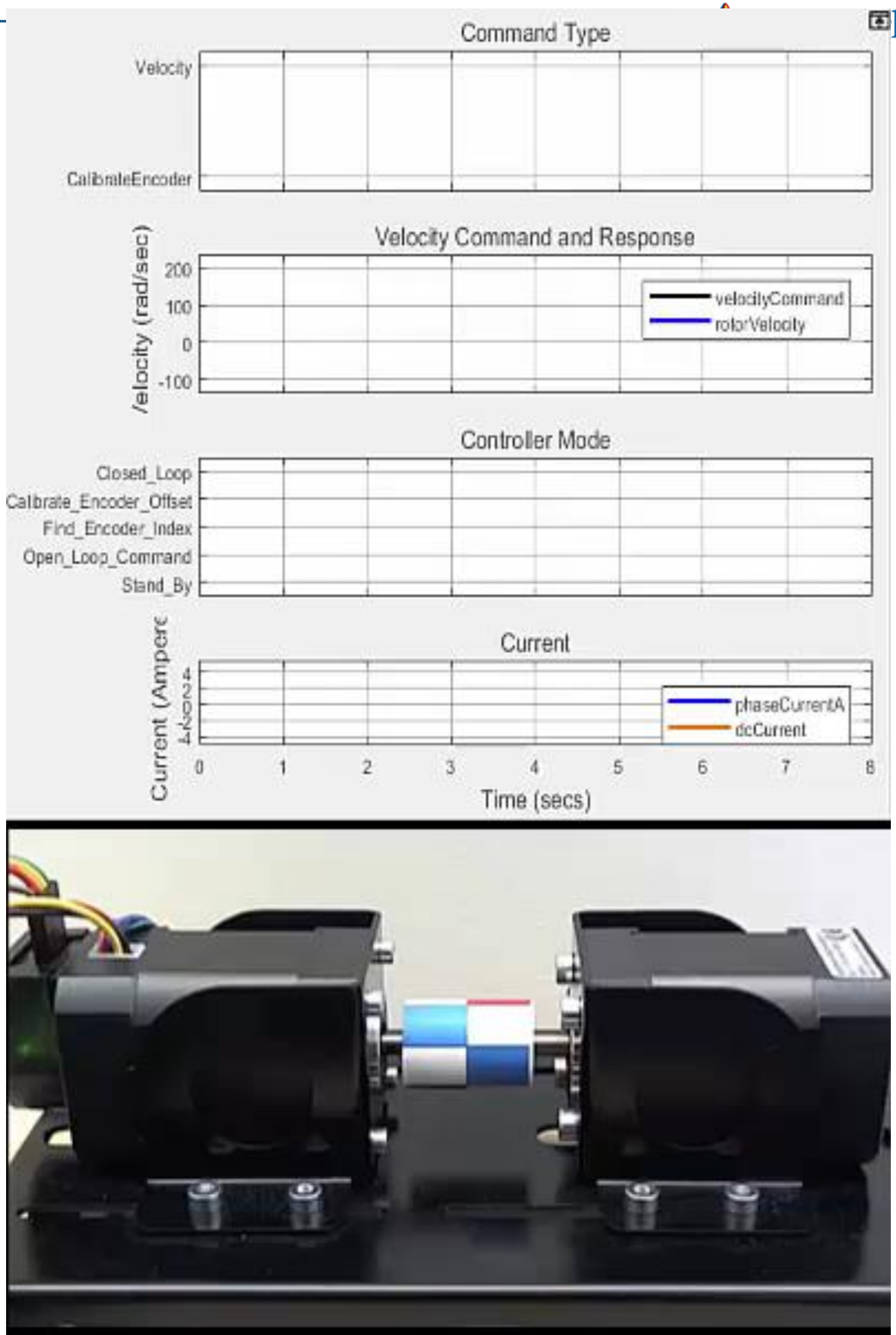
# 软硬件协同开发在电机控制的应用

龚小平

# 内容

基于模型的设计如何帮助 SoC开发：

- 进行设计的早期验证
- 协同开发以提升团队效率
- 降低硬件试验的测试时间



# 邦奇动力(Punch Powertrain)基于SoC开发复杂电机驱动控制系统

- 任务：开发新一代电动和混动车的动力总成系统
- 目标：降低成本的同时提升能量密度和效率
  - 集成电机和电力电子部件到传动系统
- 方案：采用新的开关磁阻电机设计
  - 电机最高转速是以前的两倍
  - 新的硬件平台Xilinx® Zynq® SoC 7045
- 挑战：没有FPGA设计经验!

Link to video of presentation

- ✓ 实现了电驱系统的机电软集成设计
- ✓ 开发了四种不同的控制策略
  - 2名工程师，18个月
- ✓ 得到了可重用于验证的模型
- ✓ 建立了无缝的集成验证流程
  - 在硬件生产前进行充分验证
  - 大大减少了试验台架验证时间

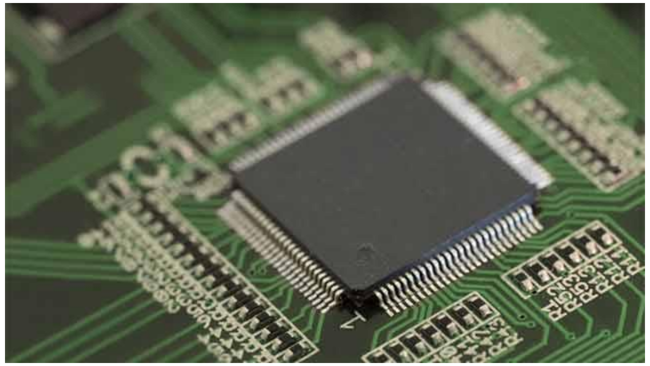
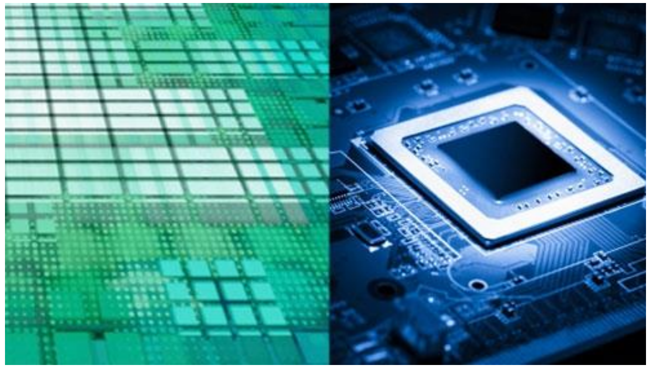
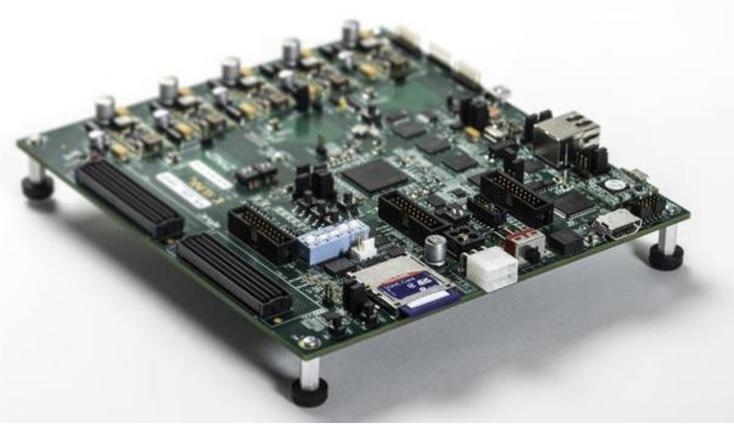
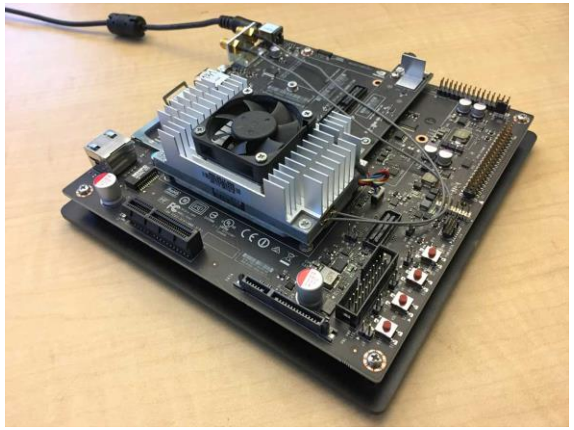
# 技术趋势 – 对电机驱动的性能要求不断提高

- 性能指标的提高要求更先进的控制算法

- 先进的控制算法要求更快的计算能力
  - 磁场定向控制
  - 无传感器电机控制
  - 振动检测和抑制控制
  - 多轴系统控制

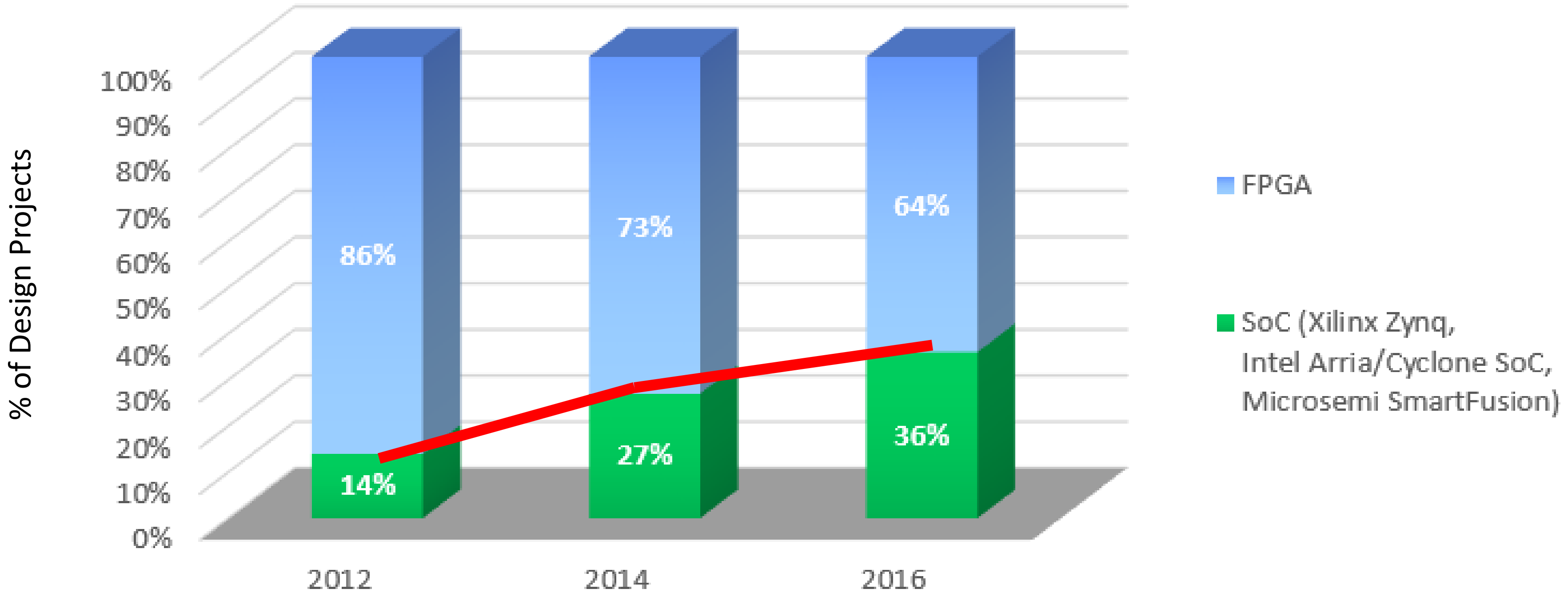
# 如何获取更快的计算能力 – 硬件平台的应对方案

- 多核处理器
- 多处理器系统
- FPGA
- ASIC
- GPU*
- SoC

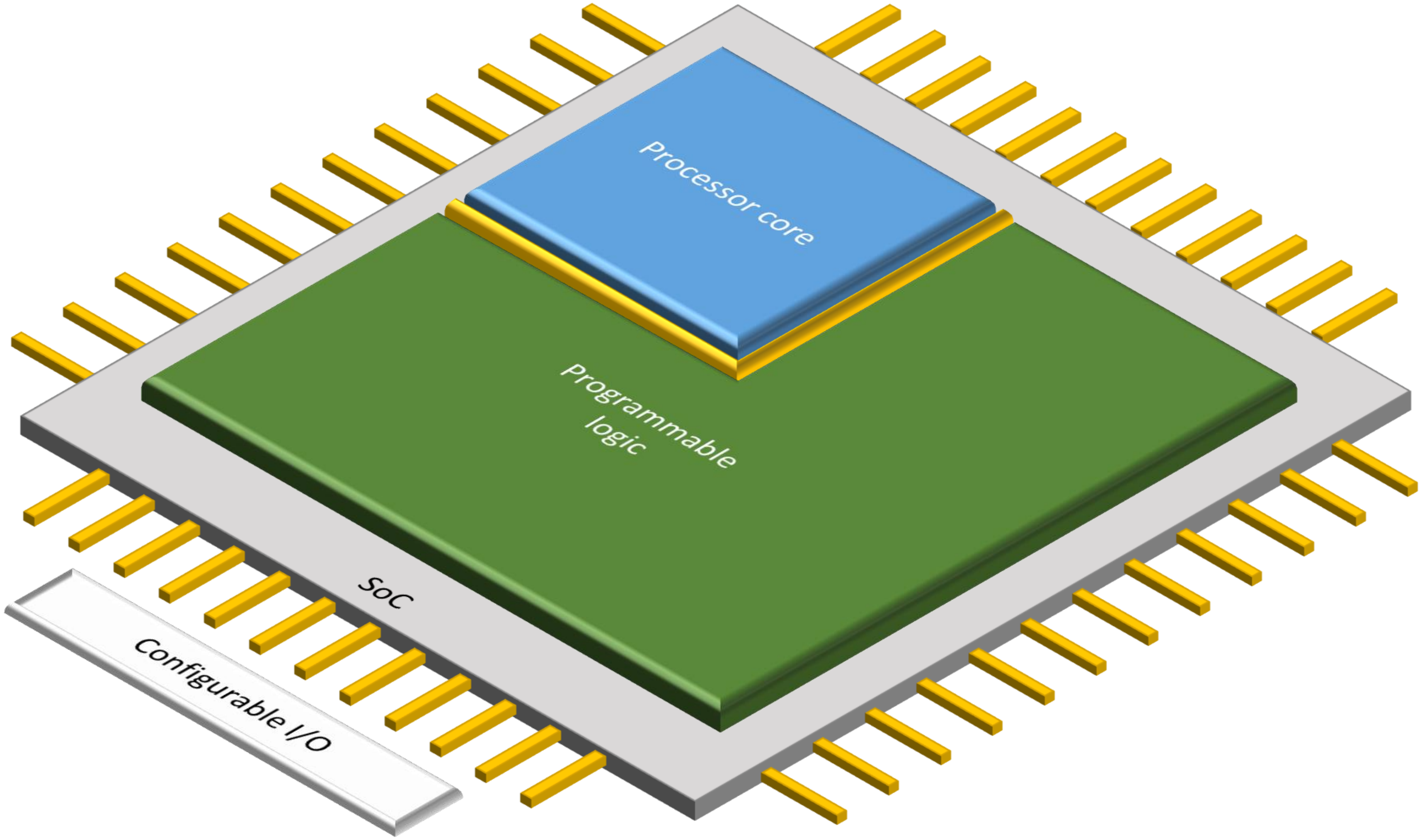
*particularly for vision applications

# 应用趋势 – SoC在新项目的占比快速提升




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

# SoC的内部结构



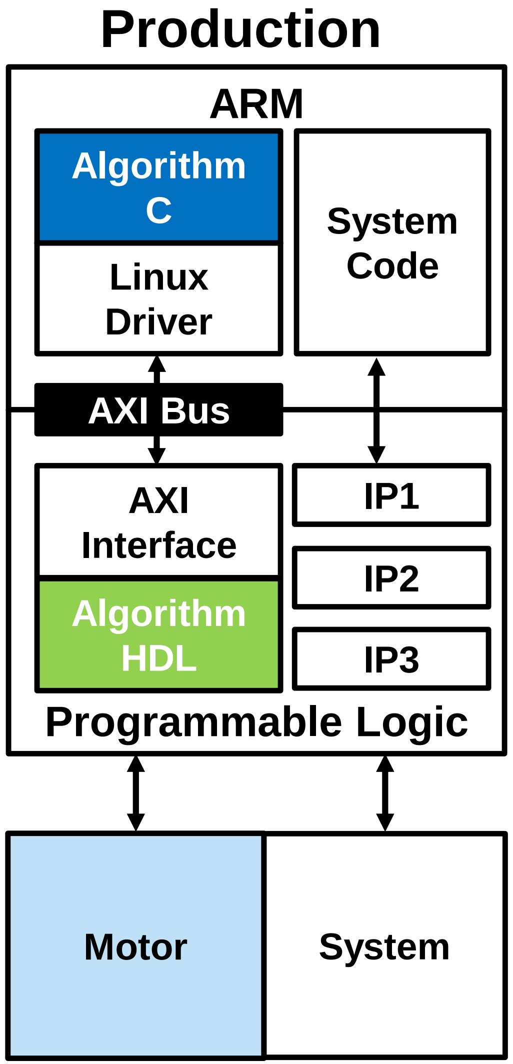
# SoC应用到电机控制



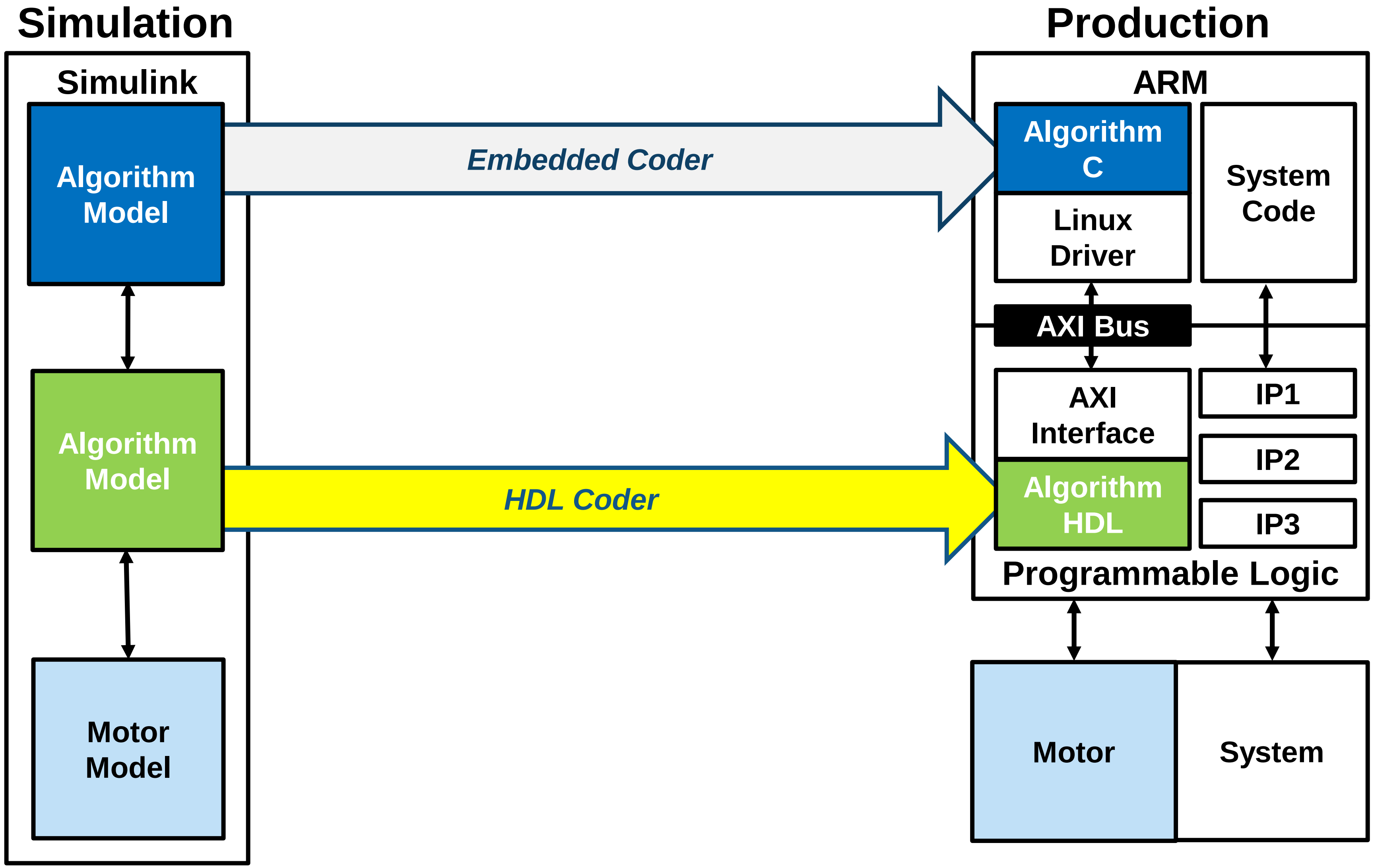
- 在有限的试验条件下验证设计规范
- 软硬件算法的开发和集成需要协同
- 快速有效地进行设计权衡和决策

- 采用仿真进行设计的早期验证
- 统一平台实现团队的协同设计
- 目标硬件快速原型加速设计迭代

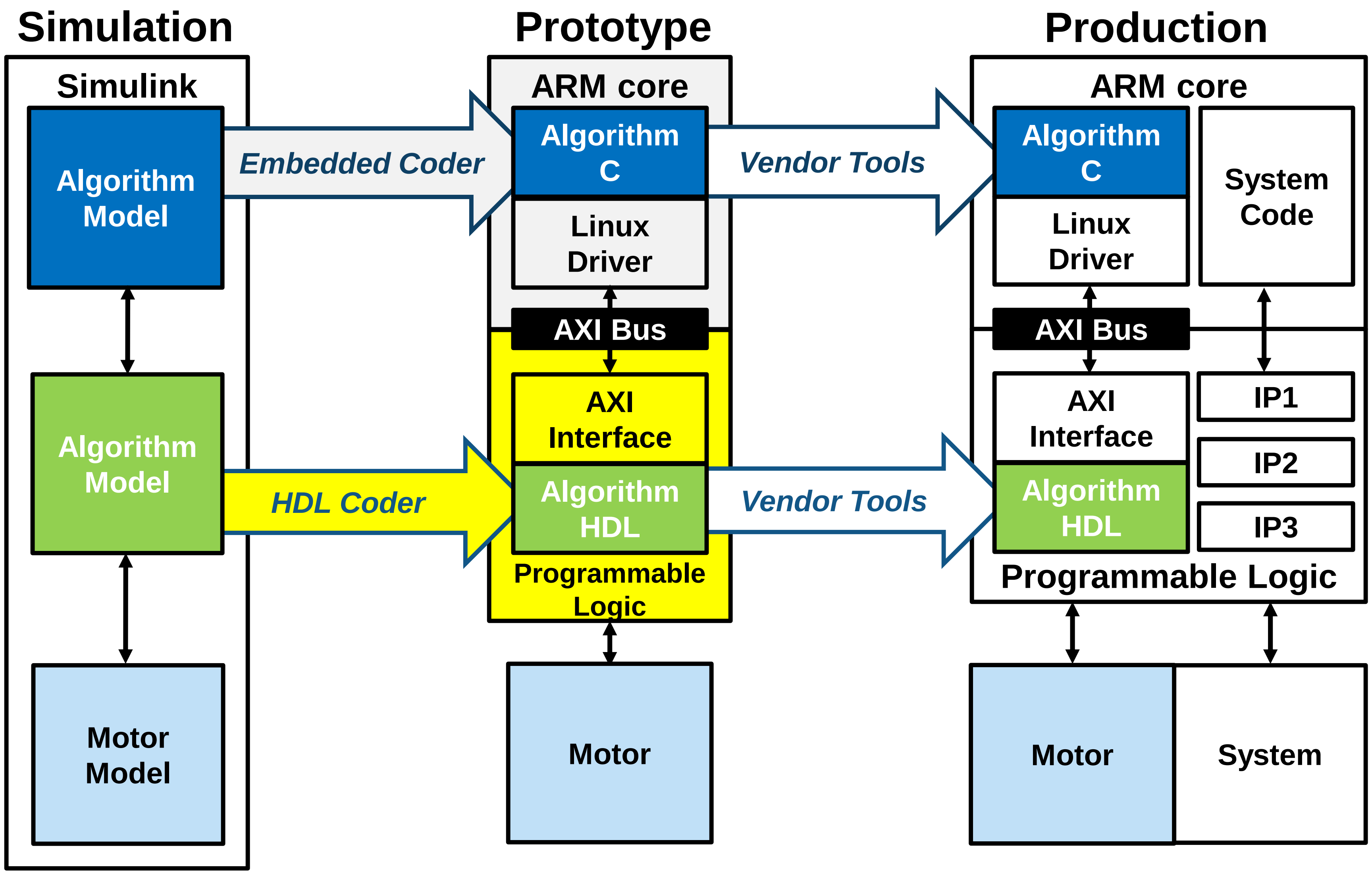
# 电机控制的产品实现


Production
ARM
Algorithm C
Linux Driver
System Code
AXI Bus
AXI Interface
Algorithm HDL
IP1
IP2
IP3
Programmable Logic
Motor
System

# 从系统仿真到产品实现

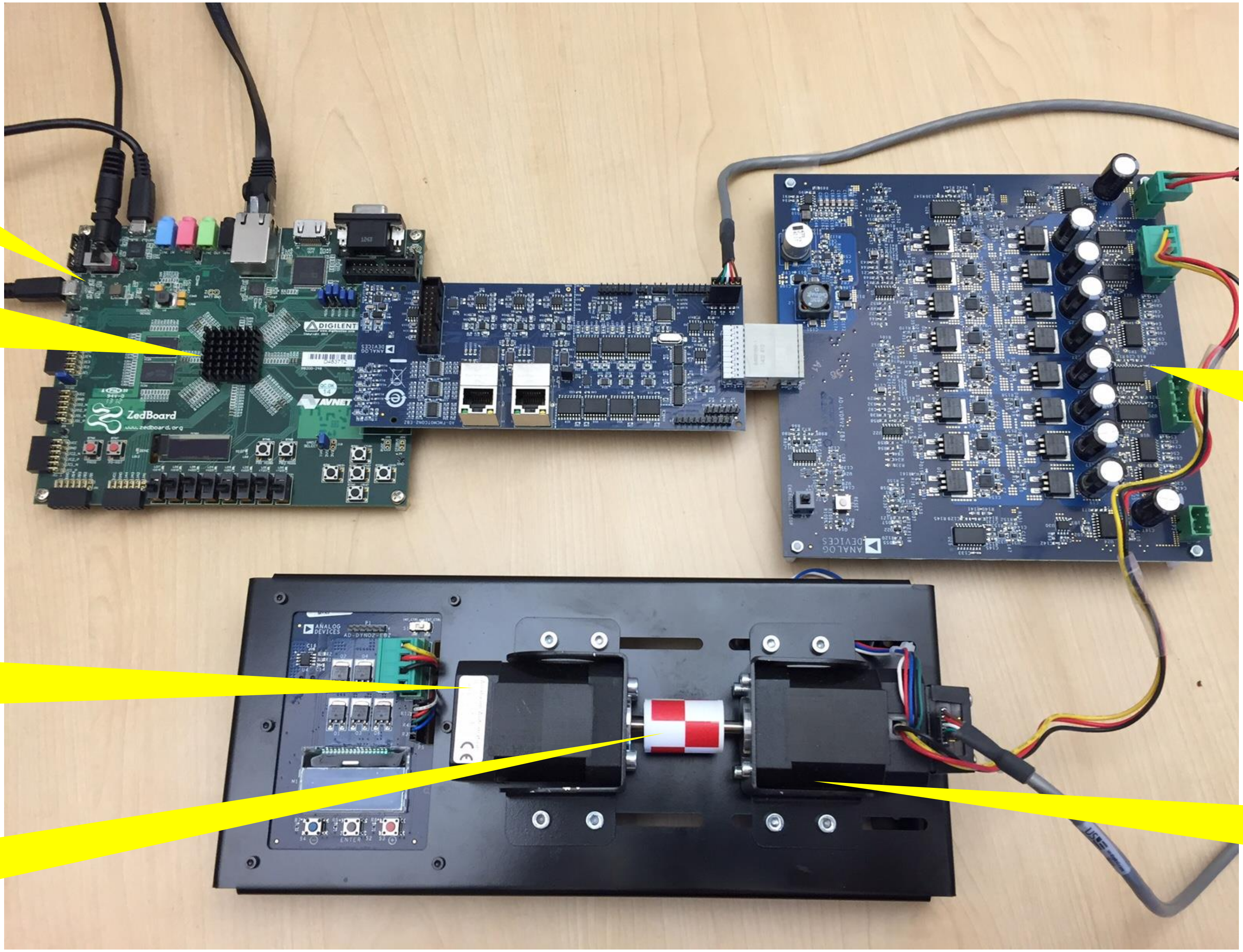

Simulation
Simulink
Algorithm Model
Algorithm Model
Motor Model
Embedded Coder
HDL Coder
Production
ARM
Algorithm C
Linux Driver
System Code
AXI Bus
AXI Interface
Algorithm HDL
IP1
IP2
IP3
Programmable Logic
Motor
System

# 软硬件协同的快速原型设计加速迭代过程



ZedBoard

Zynq SoC
(XC7Z020)

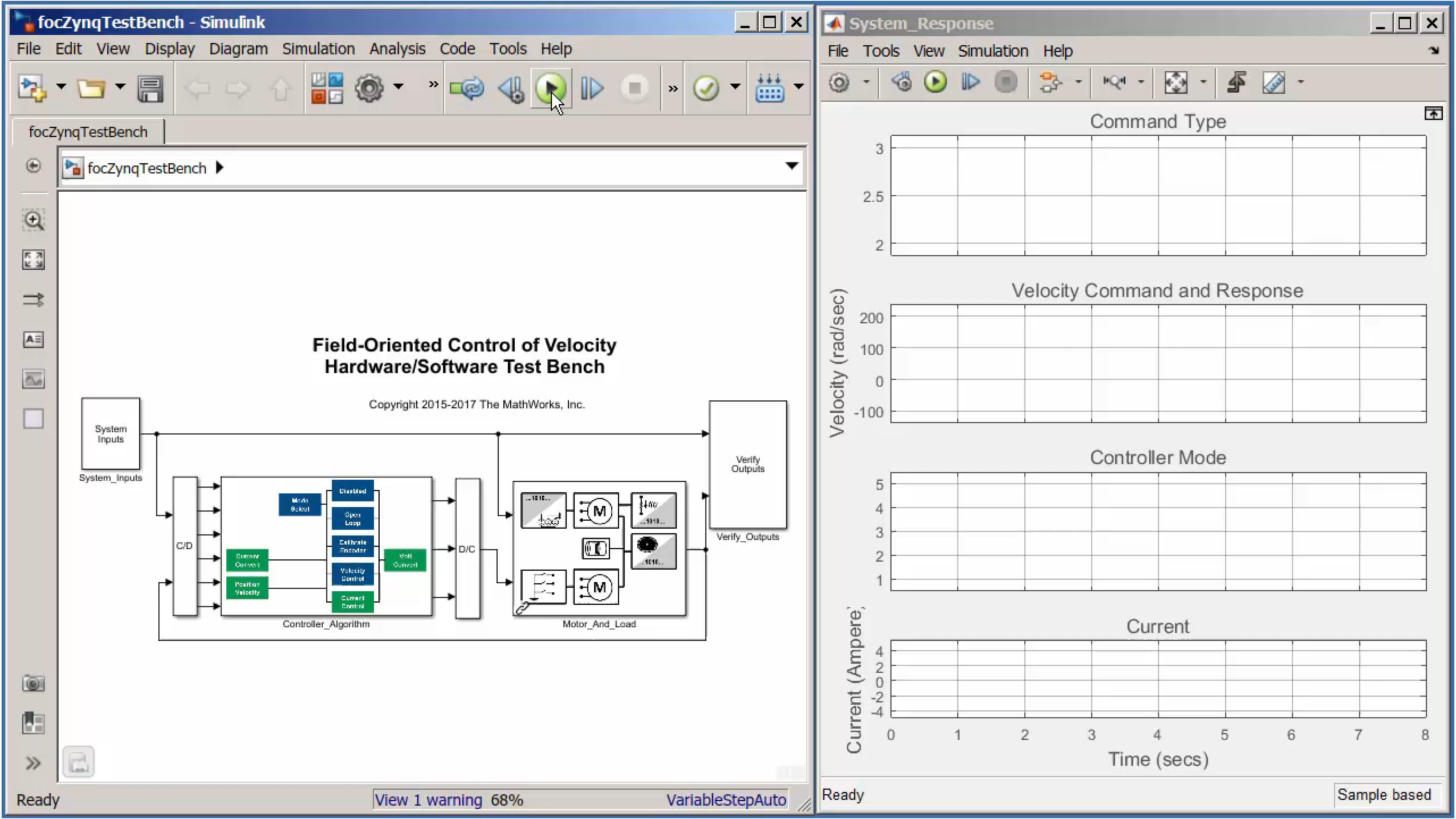


FMC module:
control board +
low-voltage board

Load motor

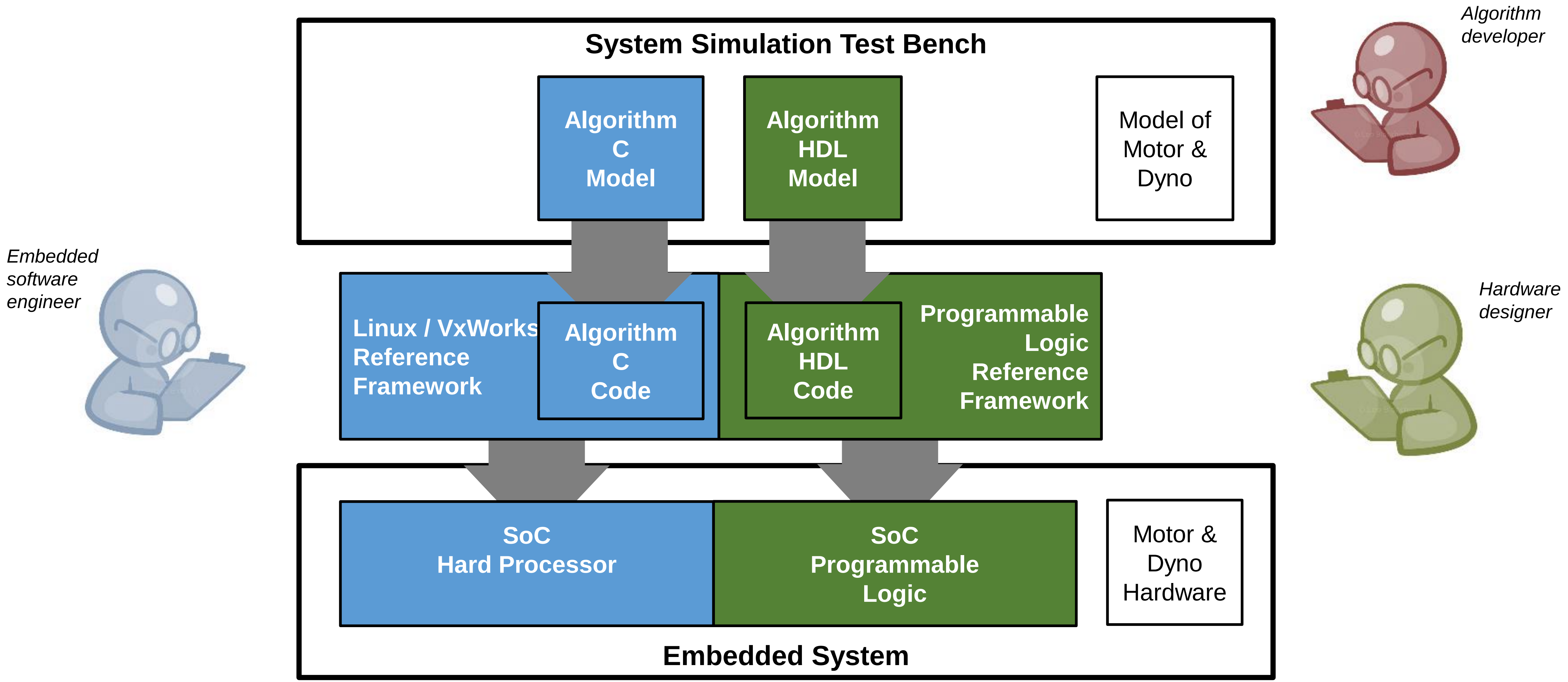
Mechanical
coupler

Motor under test
(with encoder)

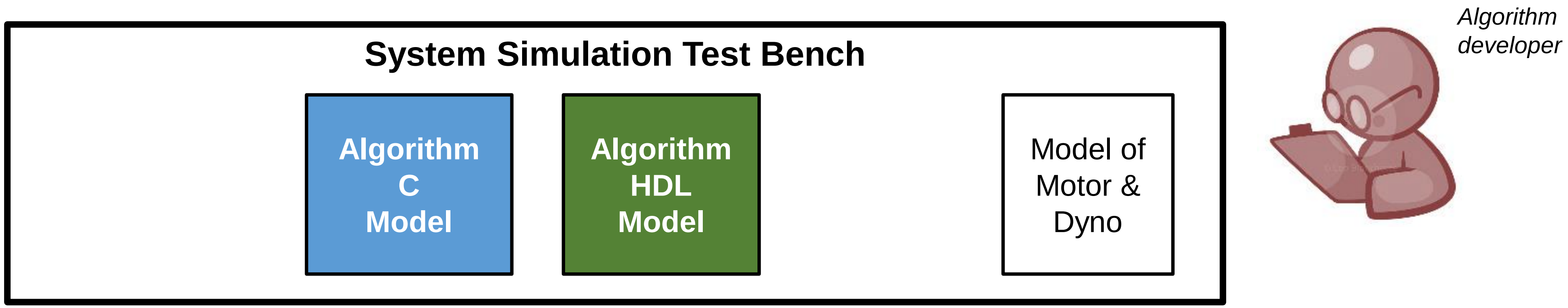


focZynqTestBench - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools Help
focZynqTestBench
Field-Oriented Control of Velocity
Hardware/Software Test Bench
Copyright 2015-2017 The MathWorks, Inc.
System Inputs
System_Inputs
C/D
Controller_Algorithm
D/C
Motor_And_Load
Verify Outputs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

# 基于模型设计的软硬件协同开发工作流程

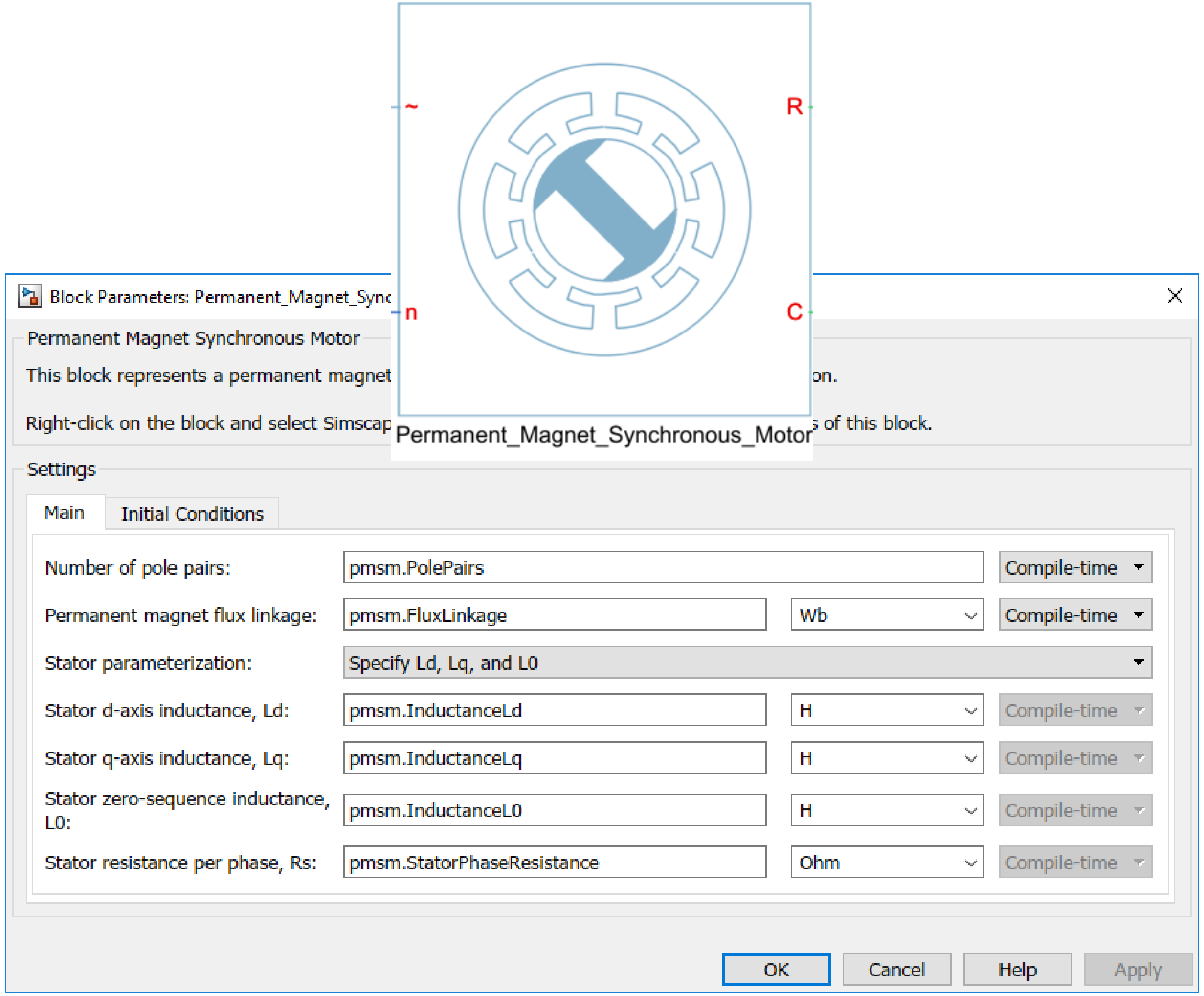


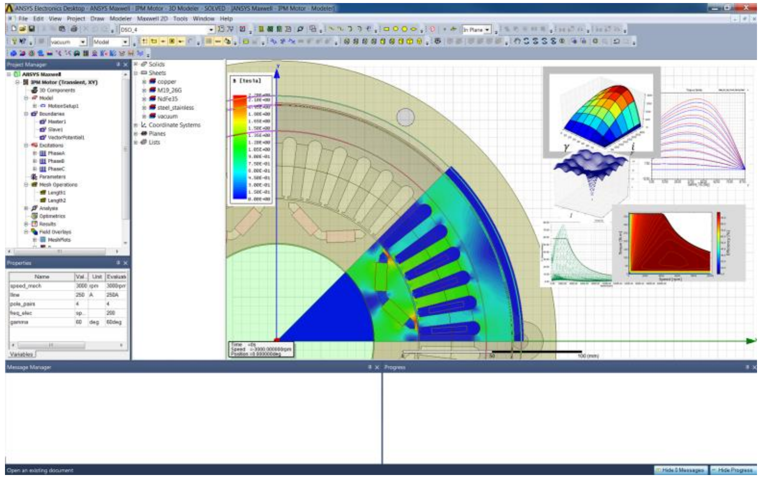
# 建立系统仿真测试平台

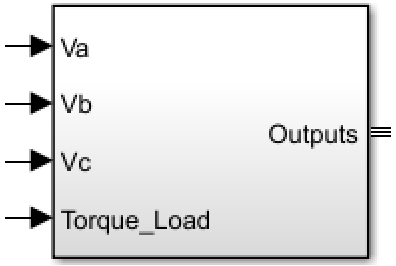



- 如何建立适当的电机模型？
- 如何获得正确的电机参数?
- 如何快速开发控制算法？

# 如何建立适当的电机模型



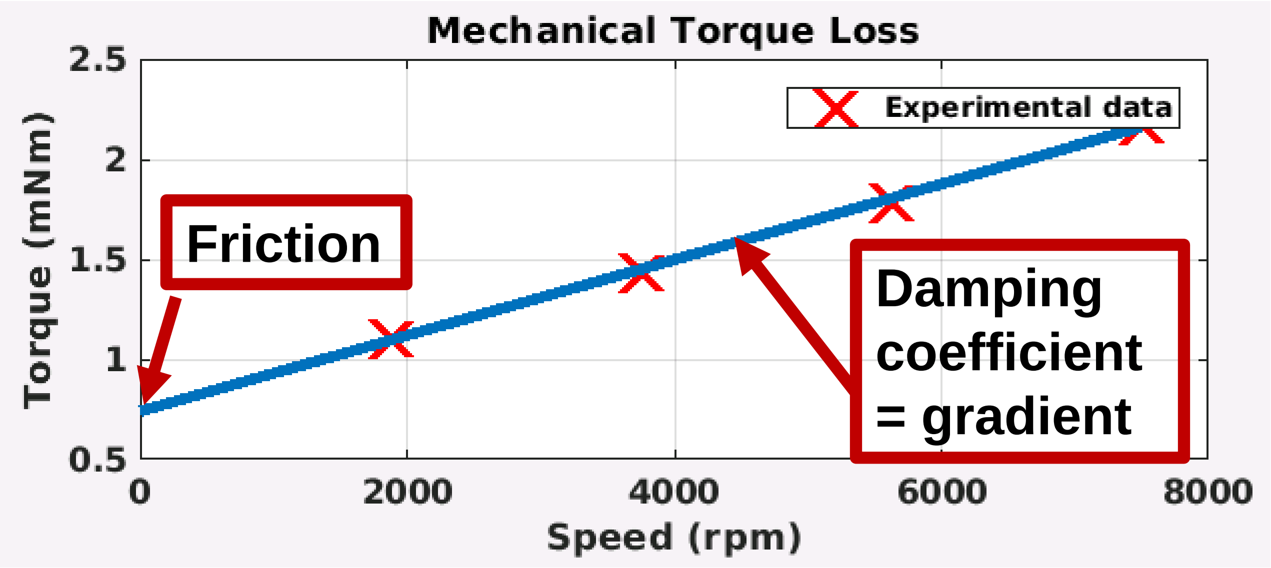



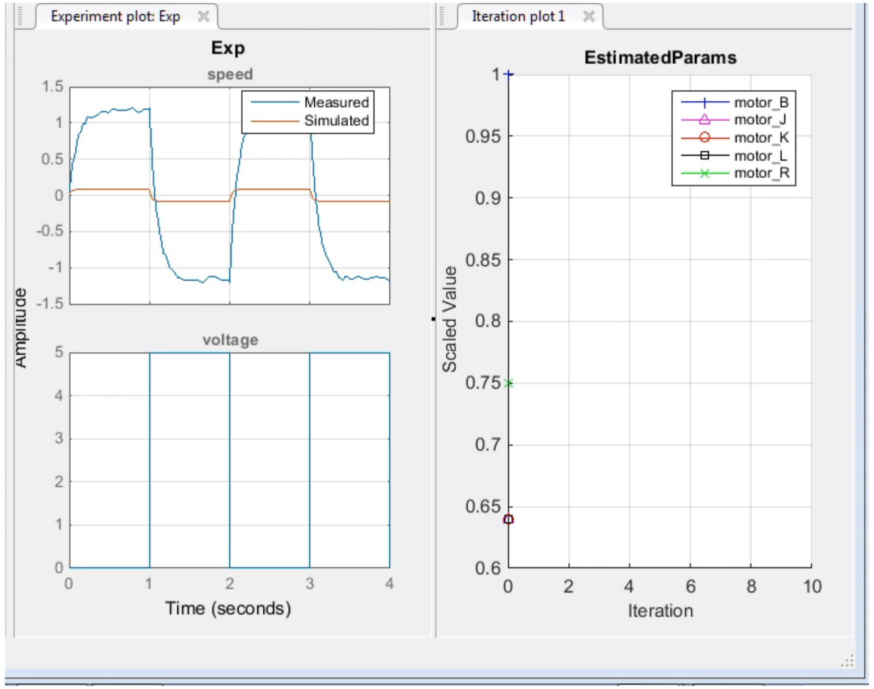



[Example - Import IPMSM Flux Linkage Data from ANSYS Maxwell]

# 如何获取正确的电机参数

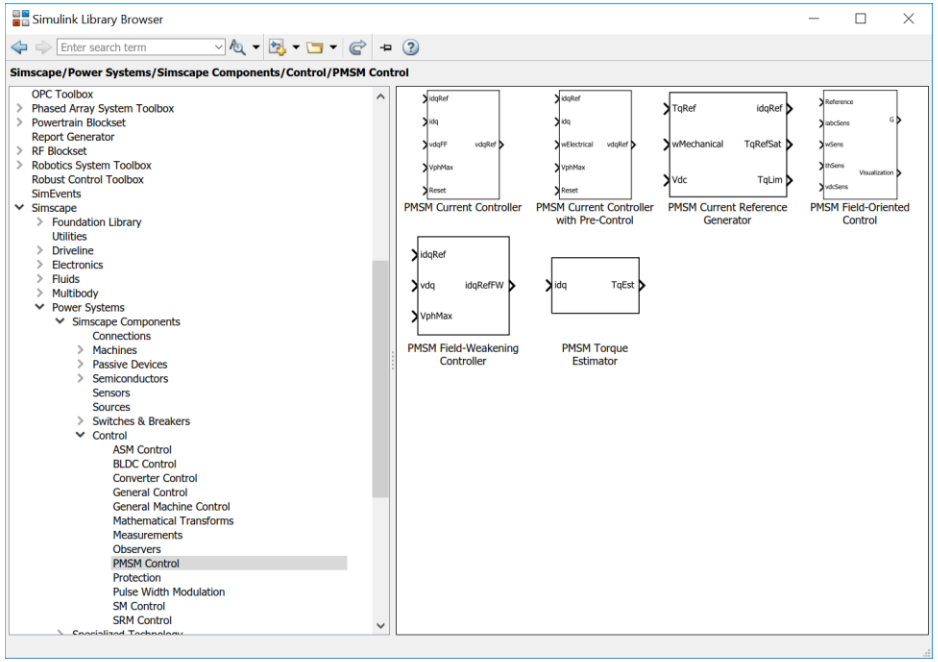
- 从手册或供应商处获取
- 从台架试验数据直接或间接获得



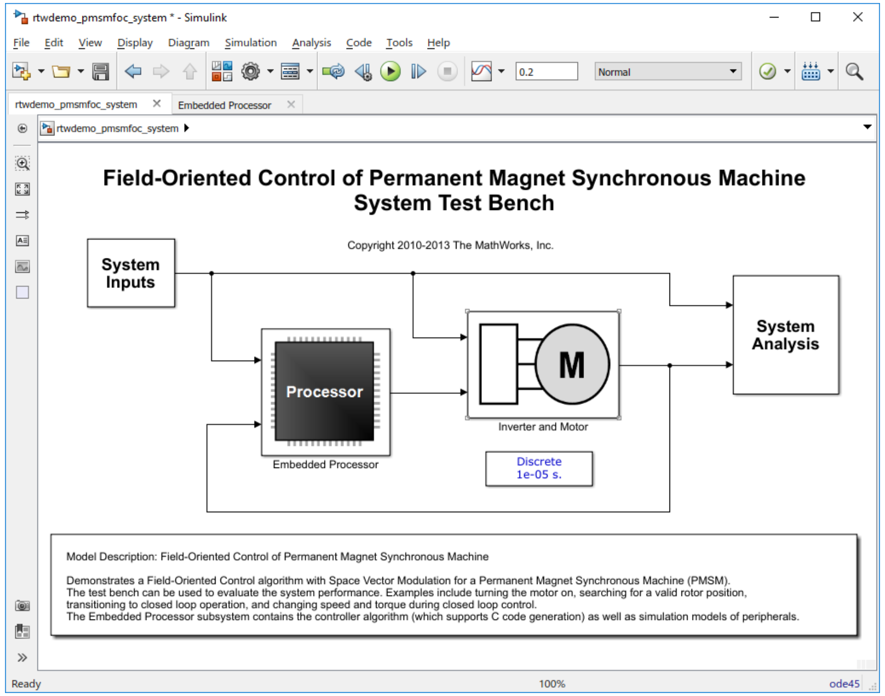



# 如何快速开发控制算法

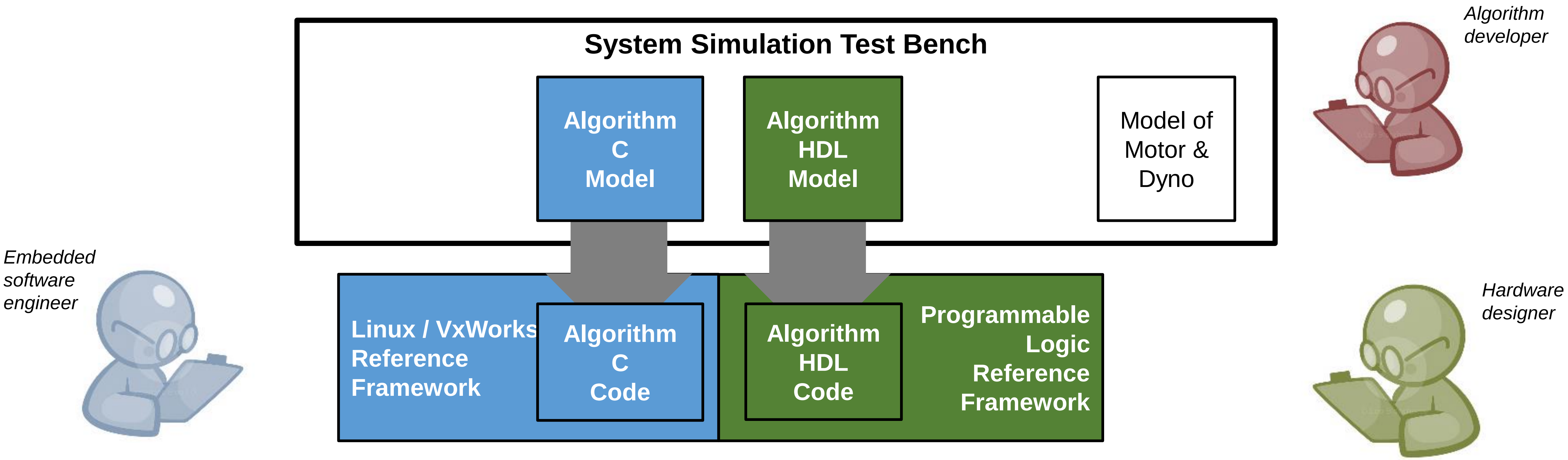
- 电机控制算法库




- 参考设计模型

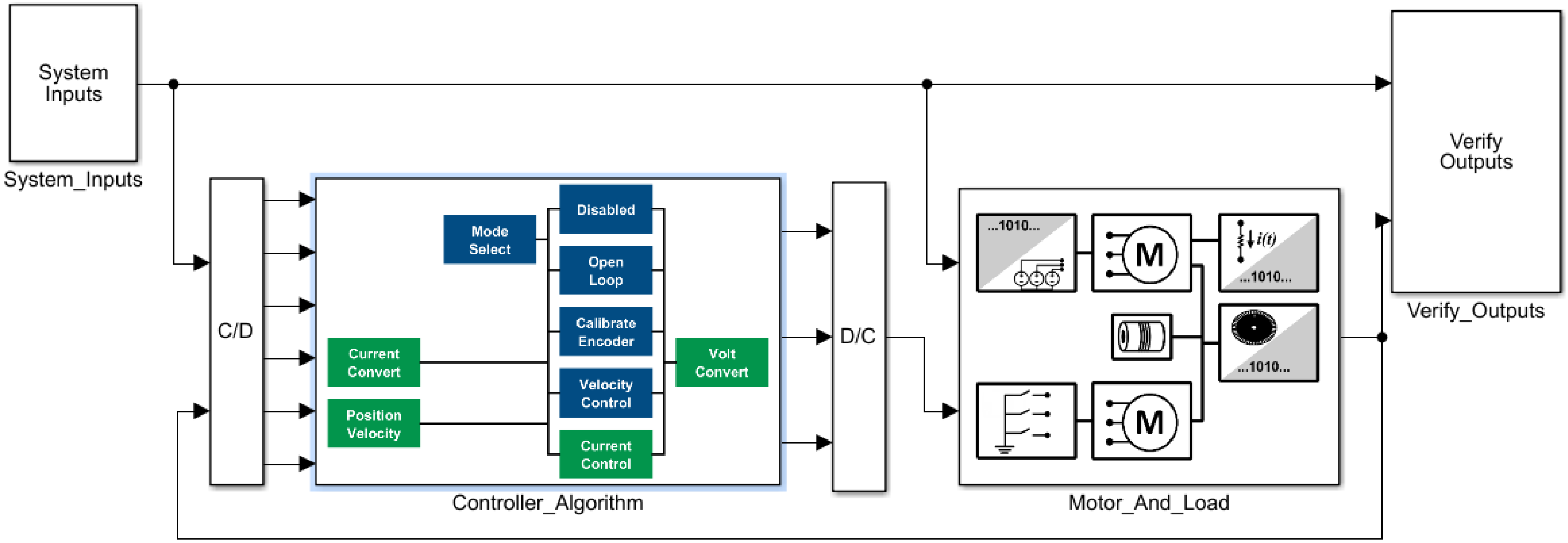


# 设计权衡和算法实现

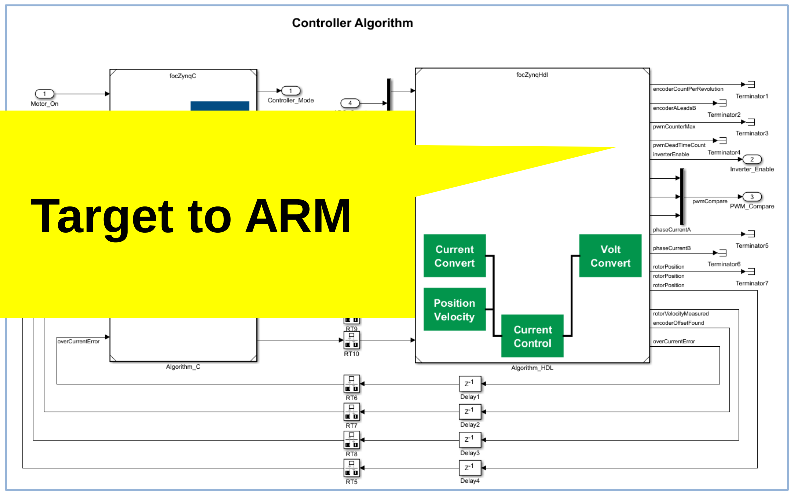



- 如何进行软件和硬件的分割？
- 采用定点算法还是浮点算法?
- 如何实现从模型到代码的转换？

# 软硬件算法的分割



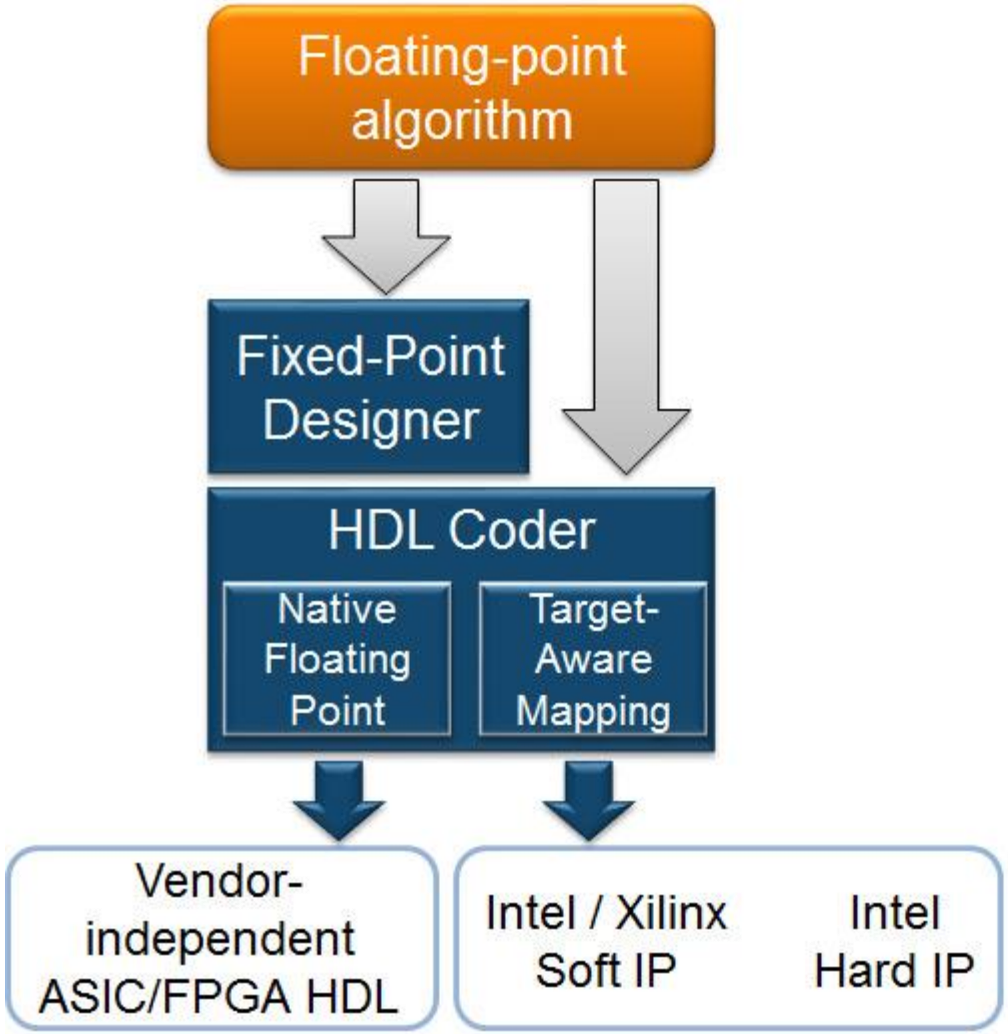

**Target to Programmable Logic**



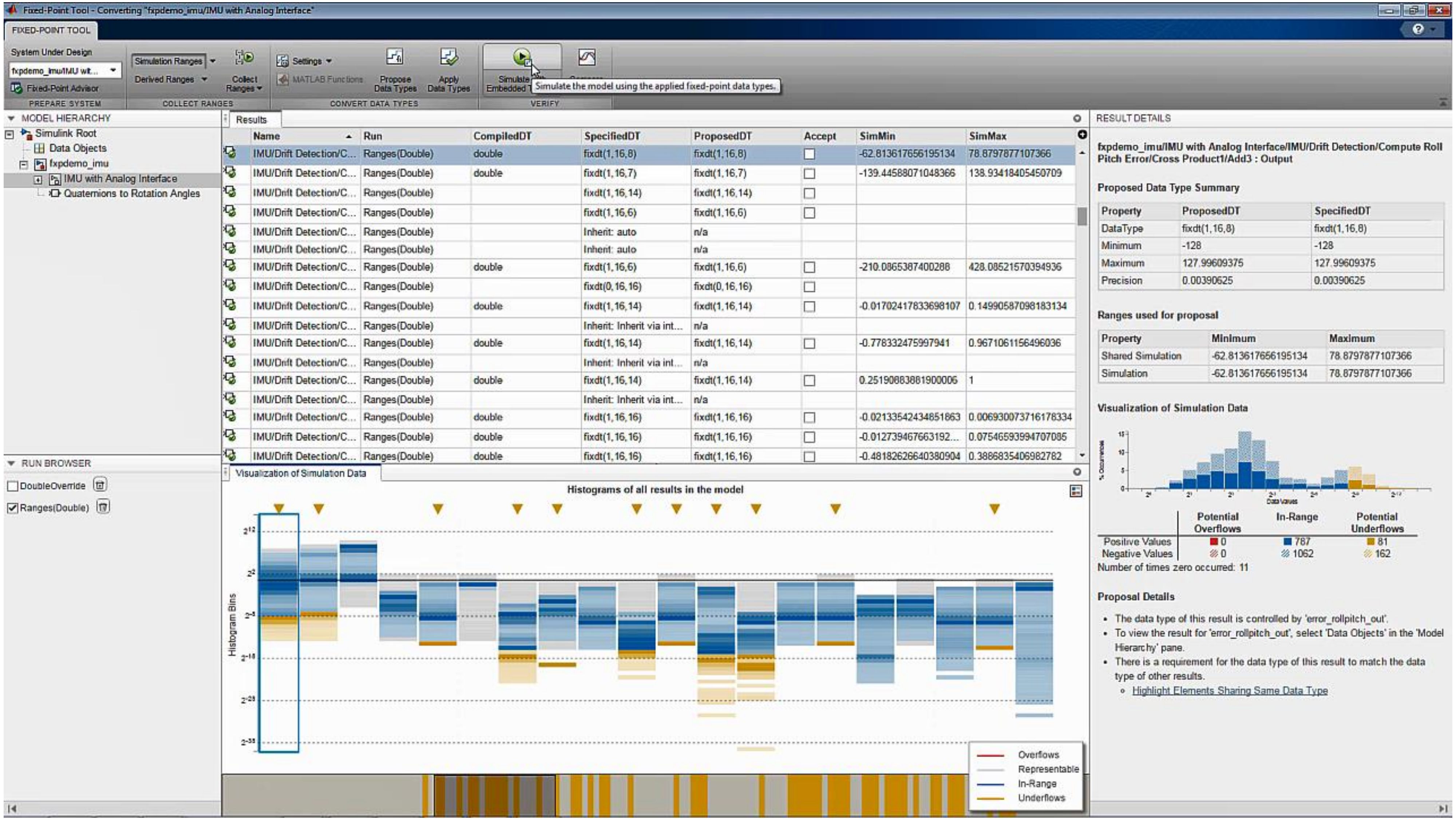

**Target to ARM**

# 定点和浮点算法的选择

- 硬件资源和开发周期
- 数据精度和动态范围
- Fixed-Point Designer™ 支持定点的自动转换
- HDL Coder™ 支持浮点算法的标准代码生成

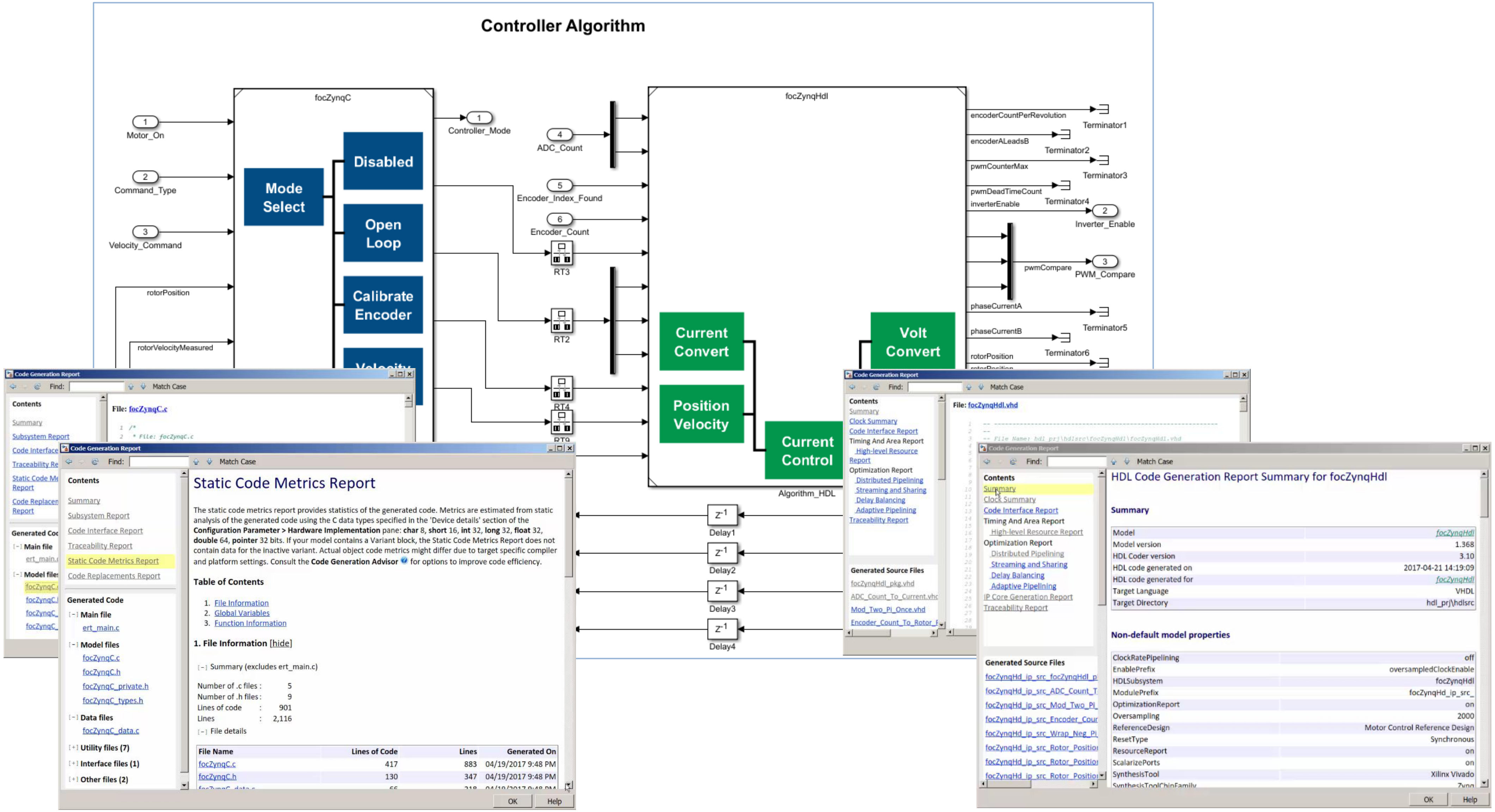


# 浮点到定点的自动转换

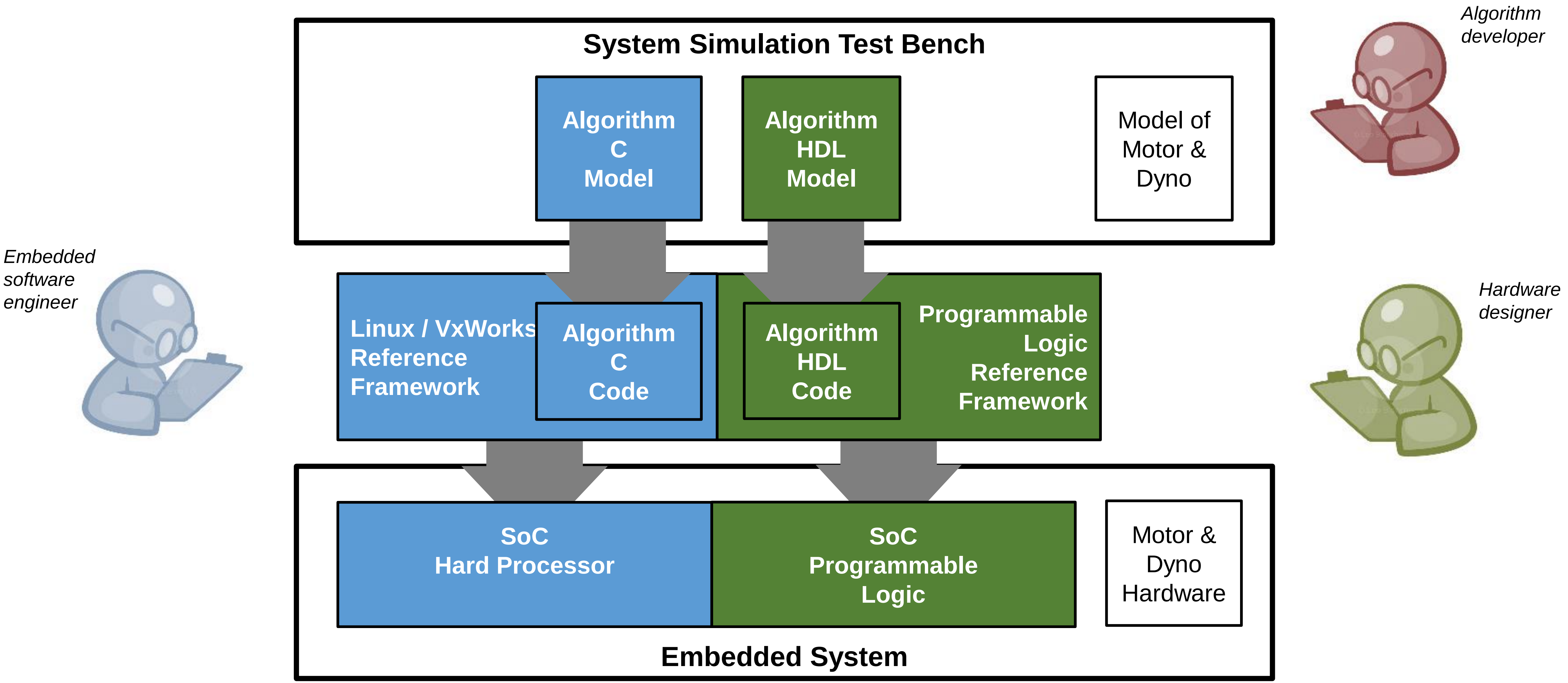


Fixed-Point Tool - Converting "fxpdemo_imu/IMU with Analog Interface"
FIXED-POINT TOOL
System Under Design
Fixed-Point Advisor
PREPARE SYSTEM
Simulation Ranges
Derived Ranges
Collect Ranges
COLLECT RANGES
Settings
MATLAB Functions
Propose Data Types
Apply Data Types
CONVERT DATA TYPES
Simulate Embedded
Simulate the model using the applied fixed-point data types.
VERIFY
MODEL HIERARCHY
Simulink Root
Data Objects
fxpdemo_imu
IMU with Analog Interface
Quaternions to Rotation Angles
RUN BROWSER
DoubleOverride
Ranges(Double)
Results
Name
Run
CompiledDT
SpecifiedDT
ProposedDT
Accept
SimMin
SimMax
Visualization of Simulation Data
Histograms of all results in the model
Histogram Bins
Overflows
Representable
In-Range
Underflows
RESULT DETAILS
fxpdemo_imu/IMU with Analog Interface/IMU/Drift Detection/Compute Roll Pitch Error/Cross Product1/Add3 : Output
Proposed Data Type Summary
Ranges used for proposal
Visualization of Simulation Data
Potential Overflows
In-Range
Potential Underflows
Positive Values
Negative Values
Number of times zero occurred: 11
Proposal Details
The data type of this result is controlled by 'error_rollpitch_out'.
To view the result for 'error_rollpitch_out', select 'Data Objects' in the 'Model Hierarchy' pane.
There is a requirement for the data type of this result to match the data type of other results.
Highlight Elements Sharing Same Data Type

# 算法的快速实现 – 自动代码生成

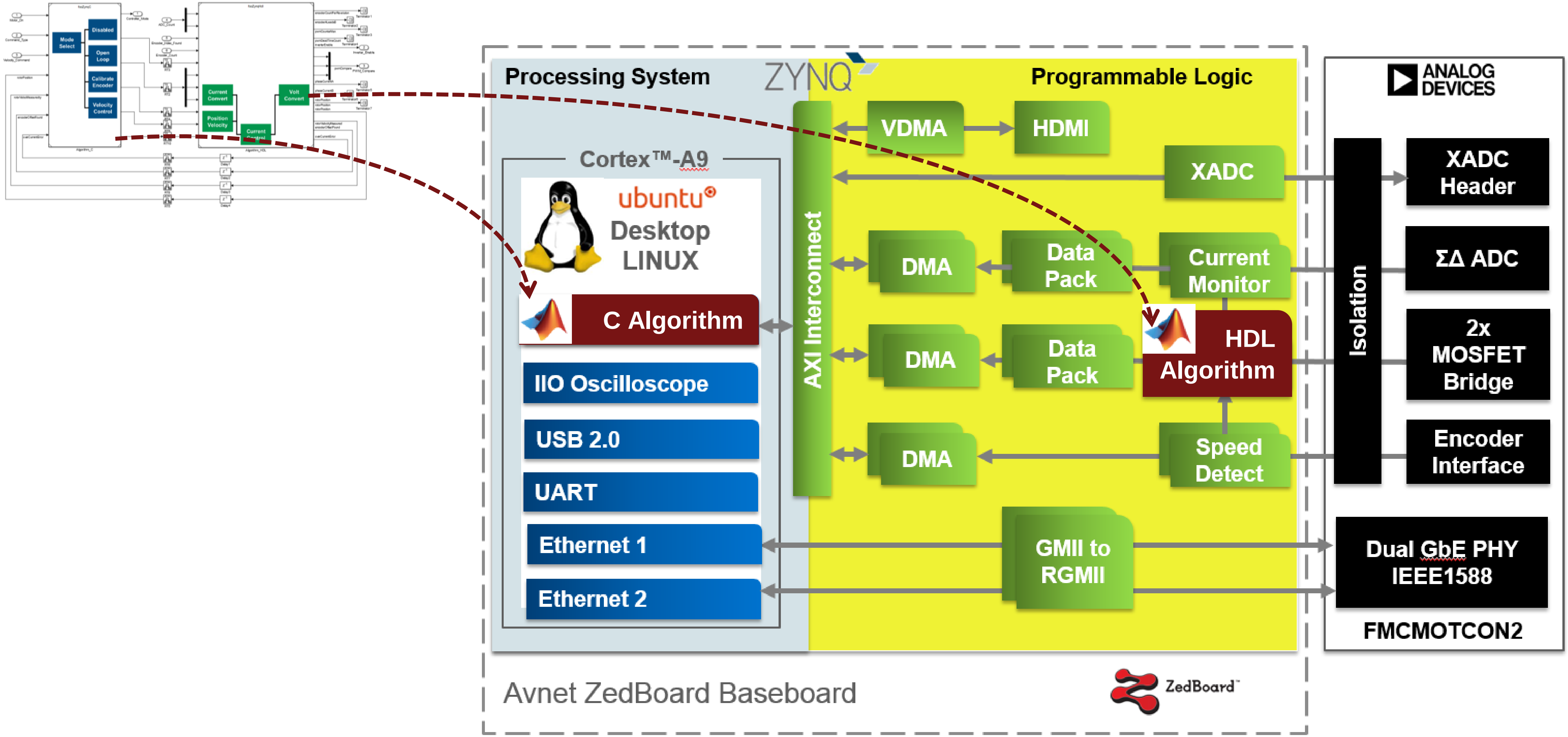


# 系统集成和硬件部署

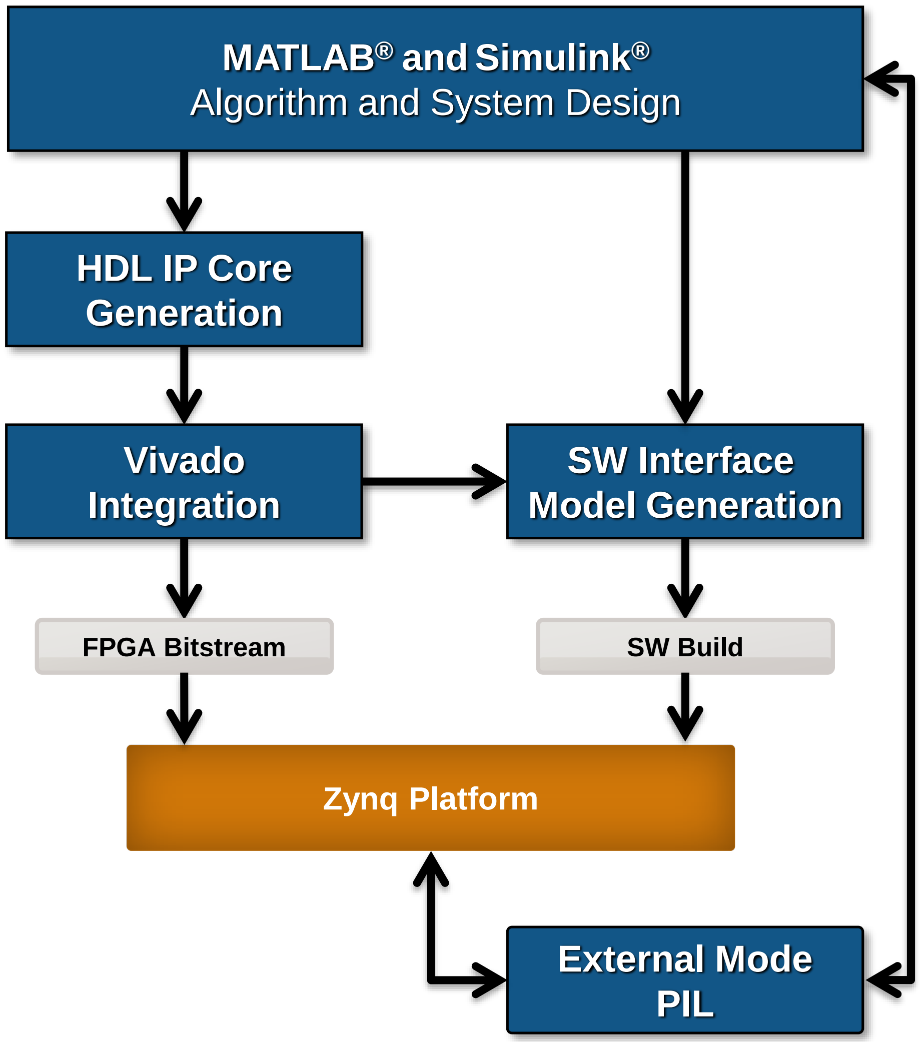


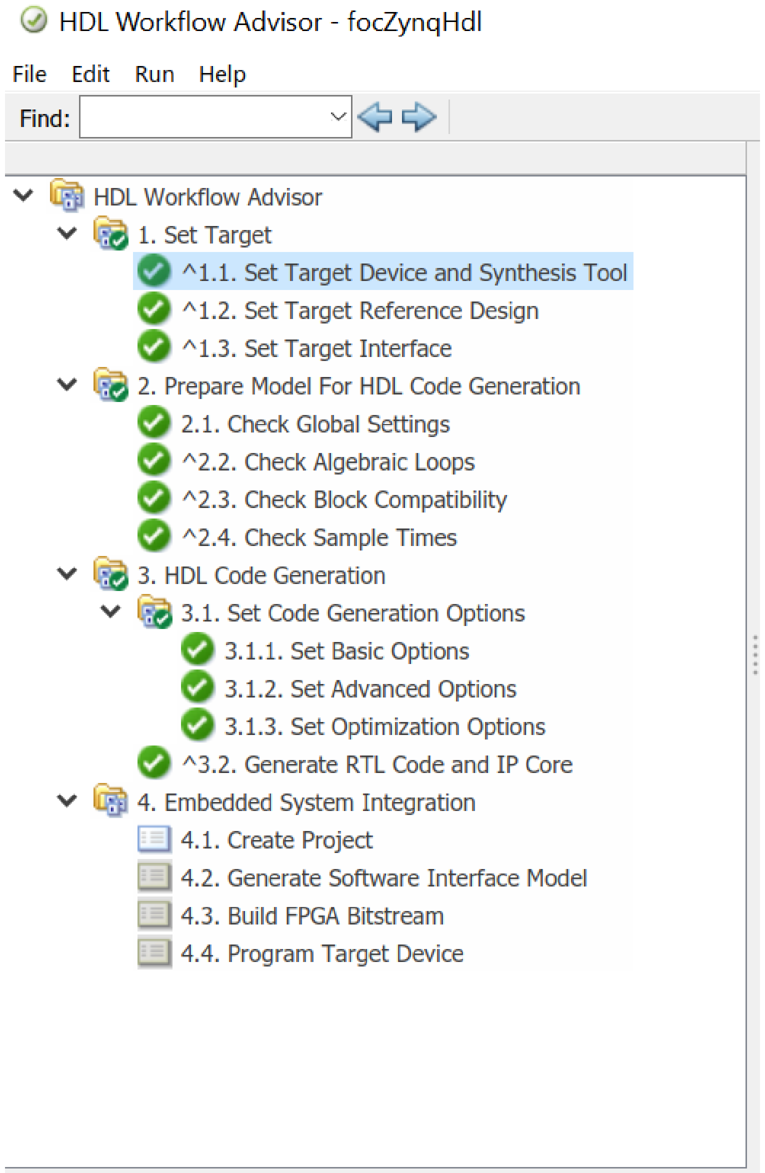

- 如何实现快速系统集成和硬件部署？

# Zynq SoC的系统集成



# 快速原型的实现工作流程



# FPGA端的IP核生成和部署

**1.1. Set Target Device and Synthesis Tool**

Analysis (^Triggers Update Diagram)

Set Target Device and Synthesis Tool for HDL code generation

Input Parameters

Target workflow: IP Core Generation

Target platform: ZedBoard and FMCMOTCON2

Synthesis tool: Xilinx Vivado

Family: Zynq

Package: clg484

Project folder: hdl_prj

Run This Task

**1.3. Set Target Interface**

Analysis (^Triggers Update Diagram)

Set target interface for HDL code generation

Input Parameters

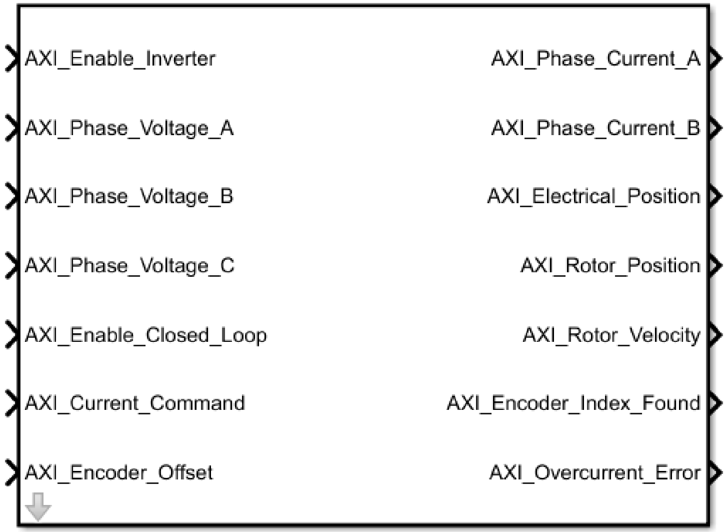
Processor/FPGA synchronization: Free running

Target platform interface table

| Port Name | Port Type | Data Type | Target Platform Interfaces | Bit |
|---|---|---|---|---|
| IP_ADC_A_Count | Inport | uint16 | IP_ADC_PhaseCurrentA [0:15] | [0:15] |
| IP_ADC_B_Count | Inport | uint16 | IP_ADC_PhaseCurrentB [0:15] | [0:15] |
| IP_Encoder_Index_F... | Inport | boolean | IP_ENC_IndexFound | [0] |
| IP_Encoder_Count | Inport | uint16 | IP_ENC_Count [0:15] | [0:15] |
| AXI_Enable_Inverter | Inport | boolean | AXI4-Lite | x"100" |
| AXI_Phase_Voltage_A | Inport | sfix16_E... | AXI4-Lite | x"104" |
| AXI_Phase_Voltage_B | Inport | sfix16_E... | AXI4-Lite | x"108" |
| AXI_Phase_Voltage_C | Inport | sfix16_E... | AXI4-Lite | x"10C" |
| AXI_Enable_Closed_... | Inport | boolean | AXI4-Lite | x"110" |
| AXI_Current_Comm... | Inport | sfix16_E... | AXI4-Lite | x"114" |
| AXI_Encoder_Offset | Inport | sfix16_E... | AXI4-Lite | x"118" |
| IP_Encoder_Count_... | Outport | ufix15 | IP_ENC_Count_Per_Revolution [0:1 | [0:14] |

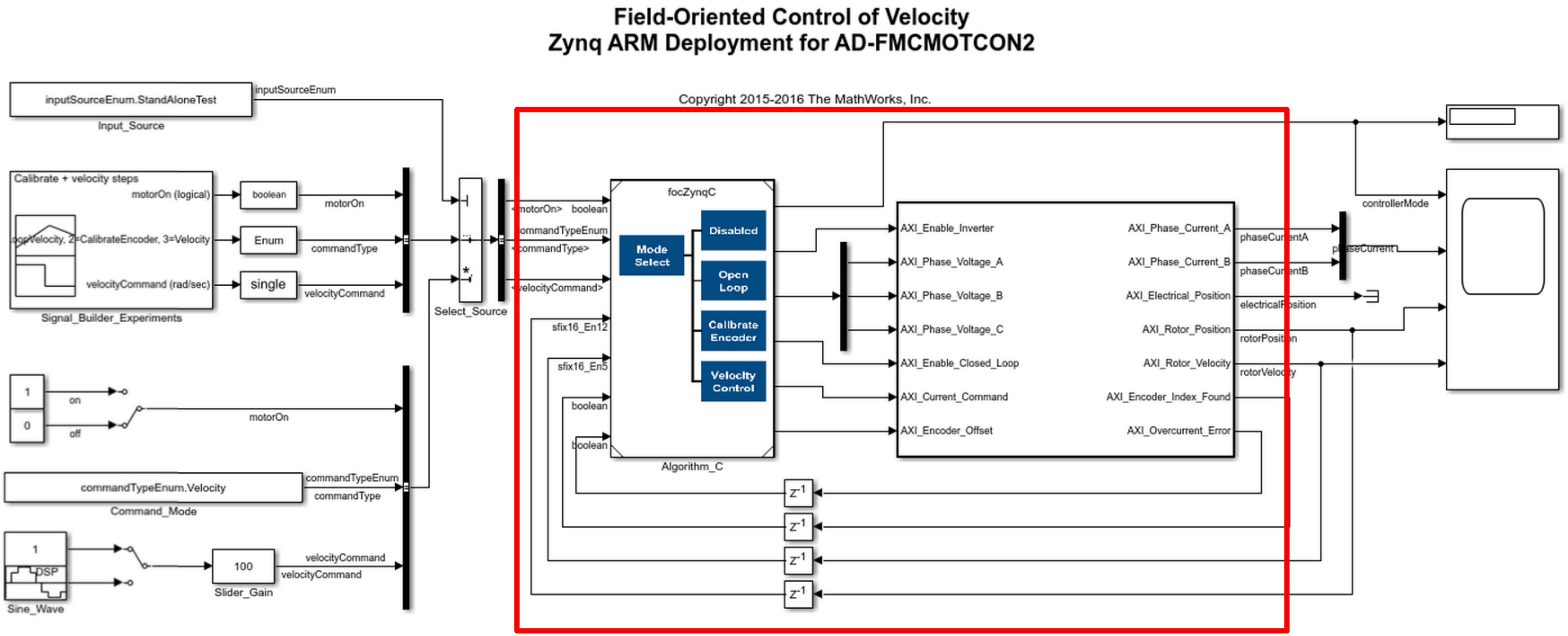
## AXI Interface Library

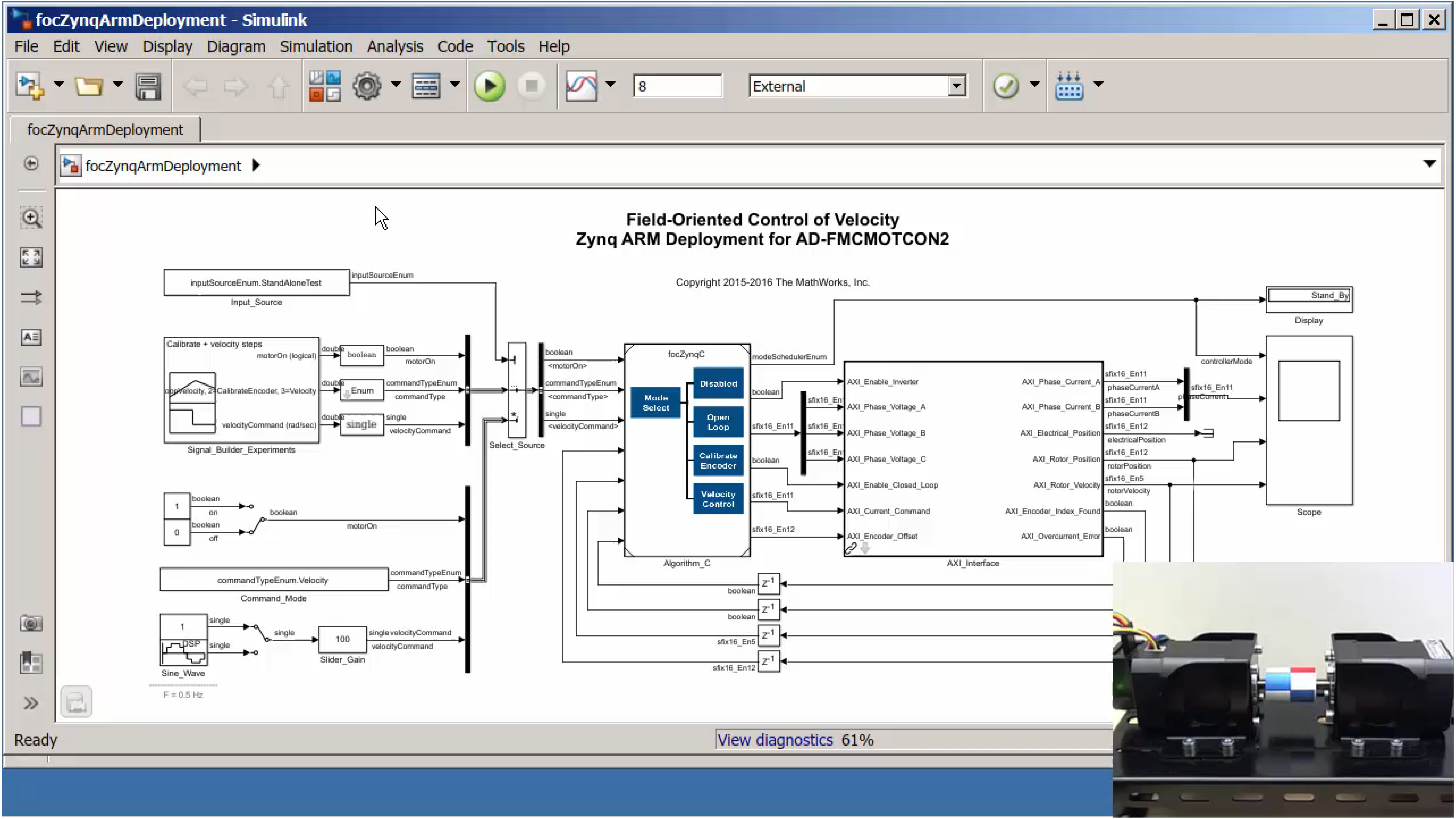
Library created on 26-Apr-2018 11:15:57




AXI_Interface

# ARM端的代码生成和部署

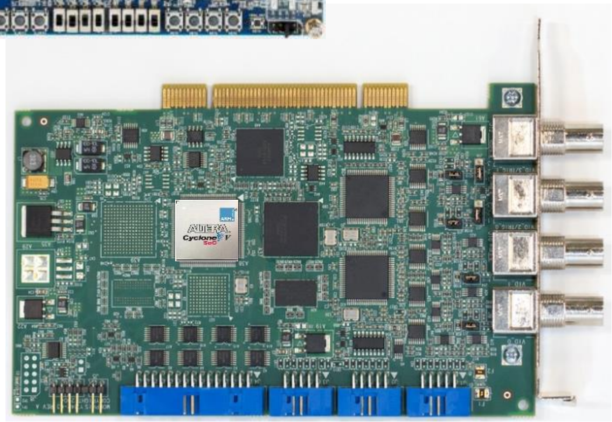



focZynqArmDeployment - Simulink
File Edit View Display Diagram Simulation Analysis Code Tools Help
8
External
focZynqArmDeployment
Field-Oriented Control of Velocity
Zynq ARM Deployment for AD-FMCMOTCON2
Copyright 2015-2016 The MathWorks, Inc.
inputSourceEnum.StandAloneTest
inputSourceEnum
Input_Source
Calibrate + velocity steps
motorOn (logical)
CalibrateEncoder, 3=Velocity
velocityCommand (rad/sec)
Signal_Builder_Experiments
boolean
motorOn
Enum
commandTypeEnum
commandType
single
velocityCommand
Select_Source
<motorOn>
<commandType>
<velocityCommand>
focZynqC
Mode Select
Disabled
Open Loop
Calibrate Encoder
Velocity Control
Algorithm_C
modeSchedulerEnum
sfix16_En11
sfix16_En12
sfix16_En5
AXI_Enable_Inverter
AXI_Phase_Voltage_A
AXI_Phase_Voltage_B
AXI_Phase_Voltage_C
AXI_Enable_Closed_Loop
AXI_Current_Command
AXI_Encoder_Offset
AXI_Phase_Current_A
AXI_Phase_Current_B
AXI_Electrical_Position
AXI_Rotor_Position
AXI_Rotor_Velocity
AXI_Encoder_Index_Found
AXI_Overcurrent_Error
AXI_Interface
phaseCurrentA
phaseCurrentB
electricalPosition
rotorPosition
rotorVelocity
controllerMode
Stand_By
Display
Scope
on
off
commandTypeEnum.Velocity
Command_Mode
DSP
Sine_Wave
F = 0.5 Hz
100
Slider_Gain
Ready
View diagnostics 61%

# Altera SoC的目标硬件平台开发支持

Altera Cyclone V SoC Development Kit

Arrow SoCKit



Custom Cyclone V SoC boards

# 3T采用基于模型的设计开发工业机器人紧急制动系统

## 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

# 总结 – 基于模型的设计应用在SoC开发的优势


- 在有限的试验条件下验证设计规范
- 软硬件算法的开发和集成需要协同
- 快速有效地进行设计权衡和决策


- 采用仿真进行设计的早期验证
- 统一平台实现团队的协同设计
- 目标硬件快速原型加速设计迭代

# 更多资源

- 网络研讨会
  - 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

- 技术文章
  - How Modeling Helps Embedded Engineers Develop Applications for SoCs (MATLAB Digest)
  - MATLAB and Simulink Aid HW-SW Codesign of Zynq SoCs (Xcell Software Journal)

- 帮助手册
  - Define and Register Custom Board and Reference Design for SoC Workflow
  - Field-Oriented Control of a Permanent Magnet Synchronous Machine on SoCs