

基于 FPGA 的系统提高电机控制性能

作者：Andrei Cozma 和 Eric Cigan

简介

电机在各种工业、汽车和商业领域应用广泛。电机由驱动器控制，驱动器通过改变输入功率来控制其转矩、速度和位置。高性能电机驱动器可以提高效率，实现更快速、更精确的控制。高级电机控制系统集控制算法、工业网络 and 用户接口于一体，因此需要更多处理能力来实时执行所有任务。现代电机控制系统通常利用多芯片架构来实现：数字信号处理器(DSP)执行电机控制算法，FPGA 实现高速 I/O 和网络协议，微处理器处理执行控制¹。

随着片上系统(SoC)的出现，例如集 CPU 的灵活性与 FPGA 的处理能力于一体的 Xilinx® Zynq All Programmable SoC，设计人员终于能够将电机控制功能和其他处理任务纳入单个器件中。控制算法、网络和其他处理密集型任务被分流到可编程逻辑，而管理控制、系统监控与诊断、用户接口以及调试则由处理单元处理。可编程逻辑可以包含多个并行工作的控制内核，以实现多轴机器或多重控制系统。由于在单芯片上实现了完整的控制器，因此硬件设计可以更简单、更可靠、更便宜。

近年来，在 MathWorks® Simulink®等软件建模与仿真工具的推动下，基于模型的设计已发展成为完整的设计流程——从模型创建到实现²。基于模型的设计改变了工程师和科学家的工作方式，把设计任务从实验室和现场转移到桌面。现在，包括工厂和控制器在内的整个系统都可以建模，工程师可以先调整好控制器的行

为，再将其部署到现场。这样就能降低受损风险，加速系统集成，减少对设备供货的依赖。一旦完成控制模型，Simulink 环境便可将其自动转换为由控制系统运行的 C 和 HDL 代码，节省时间并避免人工编程错误。将系统模型链接到快速原型开发环境可进一步降低风险，因为后者允许观察控制器在实际条件下会如何运作。

一个可实现更高电机控制性能的完整开发环境利用 Xilinx Zynq SoC 实现控制器，MathWorks Simulink 用于进行基于模型的设计和自动生成代码，ADI 公司的智能驱动器套件用于快速开发驱动系统原型。

Xilinx FPGA 与 SoC 电机控制解决方案

高级电机控制系统必须全面执行控制、通信和用户接口任务，每种任务都有不同的处理带宽要求和实时约束。为了实现这样的控制系统，所选的硬件平台必须鲁棒且可扩展，以便为将来的系统改进和扩张创造条件。Zynq All Programmable SoC 集高性能处理系统与可编程逻辑于一体，满足上述要求（如图 1 所示）。这种组合可提供出色的并行处理能力、实时性能、快速计算和灵活的连接。该 SoC 集成了两个 Xilinx 模数转换器(XADC)，用于监控系统或外部模拟传感器。

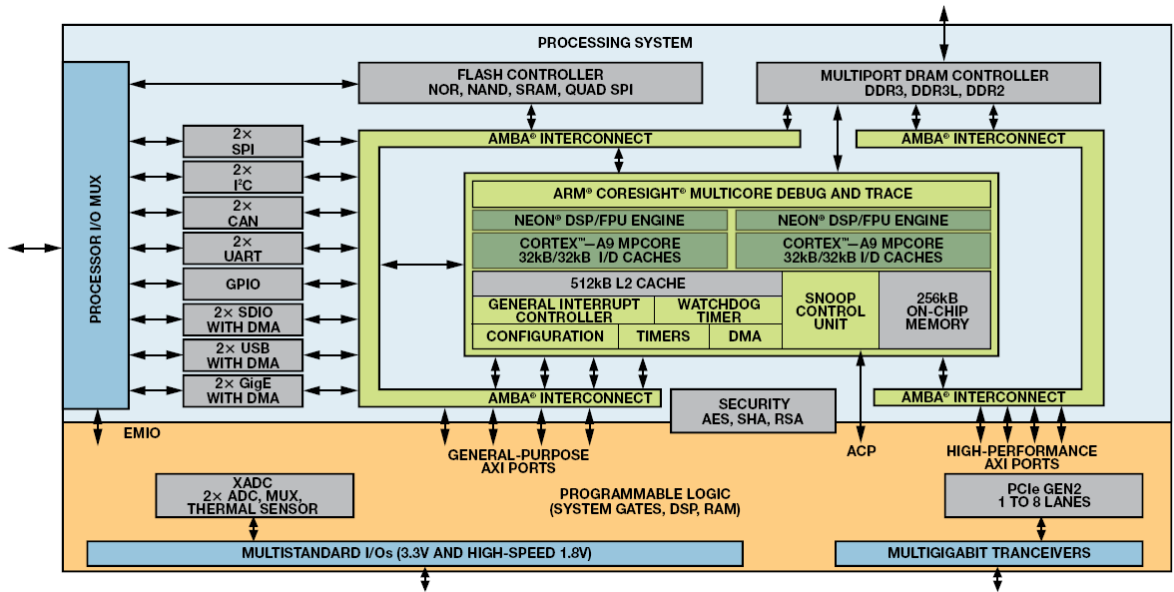


图 1. Xilinx Zynq SoC 功能框图

Zynq 包括一个双核 ARM Cortex-A9 处理器、一个 NEON 协处理器和多个用于加速软件执行的浮点扩展单元。处理系统处理管理控制、运动控制、系统管理、用户接口和远程维护等任务，这些功能非常适合通过软件实现。为了发挥系统的能力，可以采用嵌入式 Linux 或实时操作系统。可以使用自给自足的处理器，而无需配置可编程逻辑。这样，软件开发人员编写代码与硬件工程师设计 FPGA 结构可以同时进行。

在可编程逻辑方面，该器件拥有多达 444,000 个逻辑单元和 2200 个 DSP Slice，可提供巨大的处理带宽。FPGA 结构可扩展，因而用户的选择范围很广——从包含 28,000 个逻辑单元的小型器件到支持最具挑战性的信号处理应用的高端器件。5 个 AMBA-4 AXI 高速互连将可编程逻辑紧密地耦合到处理系统，提供相当于 3000 以上引脚的有效带宽。可编程逻辑适合执行时间关键的处理密集型任务，如实时工业以太网协议等，而且它支持多个控制内核并行工作，以实现多轴机器或多重控制系统。

基于 Xilinx All Programmable SoC 的解决方案和平台满足当今复杂控制算法（如磁场定向控制 FOC）和复杂调制方案（如 Xilinx 和 Qdesys 设计的再生脉冲频率调制器³）所提出的关键时序和性能要求。

利用 MathWorks Simulink 实现基于模型的设计

Simulink 是一种支持多域仿真和基于模型设计的框图环境，非常适合对包括控制算法和工厂模型的系统进行仿真。为了实现精确定位等目的，电机控制算法会调节速度、转矩和其他参数。利用仿真评估控制算法可以有效地确定电机控制设计是否合适，判断其合适后再进行昂贵的硬件测试，从而减少算法开发的时间和成本。图 2 给出了设计电机控制算法的有效工作流程：

- 构建精确的控制器和工厂模型，常常是根据电机、驱动电子、传感器和负载的资源库
- 对系统行为进行仿真以验证控制器的表现是否符合预期
- 产生 C 代码和 HDL 进行实时测试和实施

- 利用原型硬件测试控制算法
- 在原型硬件上进行仿真和测试后，如果控制系统证明令人满意，则将控制器部署到最终生产系统上

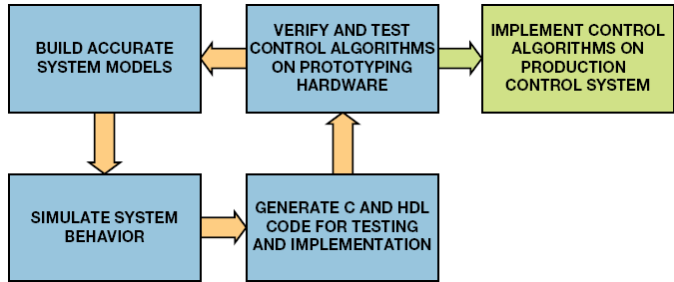


图 2. 电机控制算法设计的工作流程

MathWorks 产品包括 [Control System Toolbox™](#)、[SimPowerSystems™](#) 和 [Simscape™](#)，提供工业标准算法和应用程序以对线性控制系统进行系统化分析、设计和调谐，此外还有元件库和分析工具，用于对机械、电气、液压和其他物理领域的各种系统进行建模和仿真。利用这些工具可以创建高保真的工厂和控制器模型，进而验证控制系统的行为 and 性能，成功之后方移交实际实施。该仿真环境是验证极端功能情形和工作状况的理想场所，确保控制器已为这些状况做好准备，并且其实际操作对设备和工作人员都将是安全的。

一旦控制系统在仿真环境中完成了全面验证，就可以利用嵌入式转码器和 HDL 转码工具将其转换为 C 代码和 HDL，并部署到原型硬件上进行测试，然后部署到最终生产系统上。此时要指定定点和时序行为等软硬件实施要求。自动生成代码有助于缩短从概念到实际系统实现所需的时间，消除编程错误，确保实际结果与模型相符。图 3 给出了在 Simulink 中进行电机控制器建模并将其转移到最终生产系统所需的实际步骤。

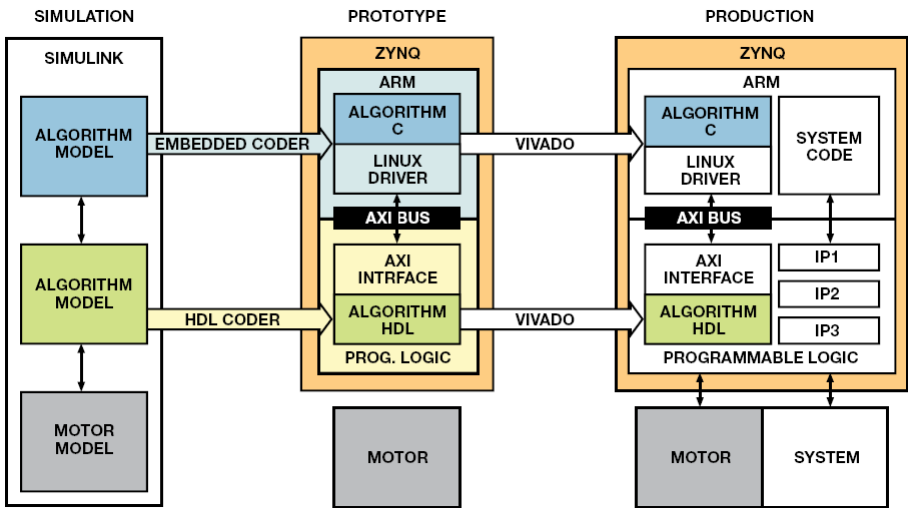


图 3. 从仿真到生产的过程

第一步是在 Simulink 中对控制器和工厂进行建模和仿真。在这一阶段，控制器算法被划分为在软件中实现的模块和在可编程逻辑中实现的模块。划分和仿真完成后，利用嵌入式转码器和 HDL 转码器将控制器模型转换为 C 代码和 HDL。基于 Zynq 的原型系统验证控制算法的性能，并且帮助进一步调谐控制器模型，然后转移到生产阶段。在生产阶段，将自动生成的 C 代码和 HDL 集成到复杂的生产系统框架中。此工作流程确保控制算法在达到生产阶段之前经过全面验证和测试，使得系统鲁棒性具有高可信度。

利用 ADI 公司智能驱动器套件快速完成原型开发

选择合适的原型开发硬件是设计过程的一个主要步骤。ADI 公司的智能驱动器套件支持快速、高效的原型开发。Avnet Zynq-7000 All Programmable SoC/Analog Devices 智能驱动器套件将 Zynq-7000 All Programmable SoC ARM 双核 Cortex-A9 + 28 nm 可编程逻辑与 ADI 公司最新一代高精度数据转换器和数字隔离相结合，可实现高性能电机控制和双通道千兆以太网工业网络连接。该套件带有 Avnet ZedBoard 7020 基板和 ADI 公司的 AD-FMCMOTCON1-EBZ 模块，构成一个完整的驱动系统，能够高效控制多种类型的电机。另外，该套件可以利用 ADI 公司的 AD-DYNO1-EBZ 测功器驱动系统进行扩展，后者是一个动态可调的负载，可用来测试实时电机控制性能。AD-FMCMOTCON1-EBZ 模块由控制器和驱动板组成，如图 4 所示。

控制板是一个混合信号 FPGA 夹层卡 (FMC)，可利用低引脚数 (LPC) 或高引脚数 (HPC) FMC 连接器连接到任何 Xilinx FPGA 或 SoC 平台。它具有以下特性：

- 利用隔离式 ADC 测量电流和电压
- 隔离式 Xilinx XADC 接口
- 完全隔离的数字控制和反馈信号
- 霍尔、差分霍尔、编码器和旋变器接口
- 2 个千兆以太网物理层，支持 EtherCAT、ProfiNET、Ethernet/IP 或 Powerlink 等高速工业通信协议
- FMC 信号电压自适应接口，支持在所有 FMC 电压电平上无缝工作

隔离是任何电机控制系统的一个关键方面，其目的是保护控制器和用户。控制板上模拟和数字信号的完全隔离，确保 FPGA 平台永远不受电机驱动侧可能产生的危险电压影响。

驱动板包含驱动电机以及电流电压检测与保护电路所需的全部电源电子。该板具有以下特性：

- 以最大 18 A 电流驱动 12 V 至 48 V 范围的 BLDC（无刷直流）/PMSM（永磁同步电机）/有刷直流/步进电机
- 动态制动功能和集成式过流与反向电压保护
- 利用隔离式 ADC 测量相电流；可编程增益放大器使电流测量输入范围达到最大
- 向控制板提供直流总线电压、相电流和总电流反馈信号
- 集成 BEMF 过零检测，用于实现 PMSM 或 BLDC 电机的无传感器控制

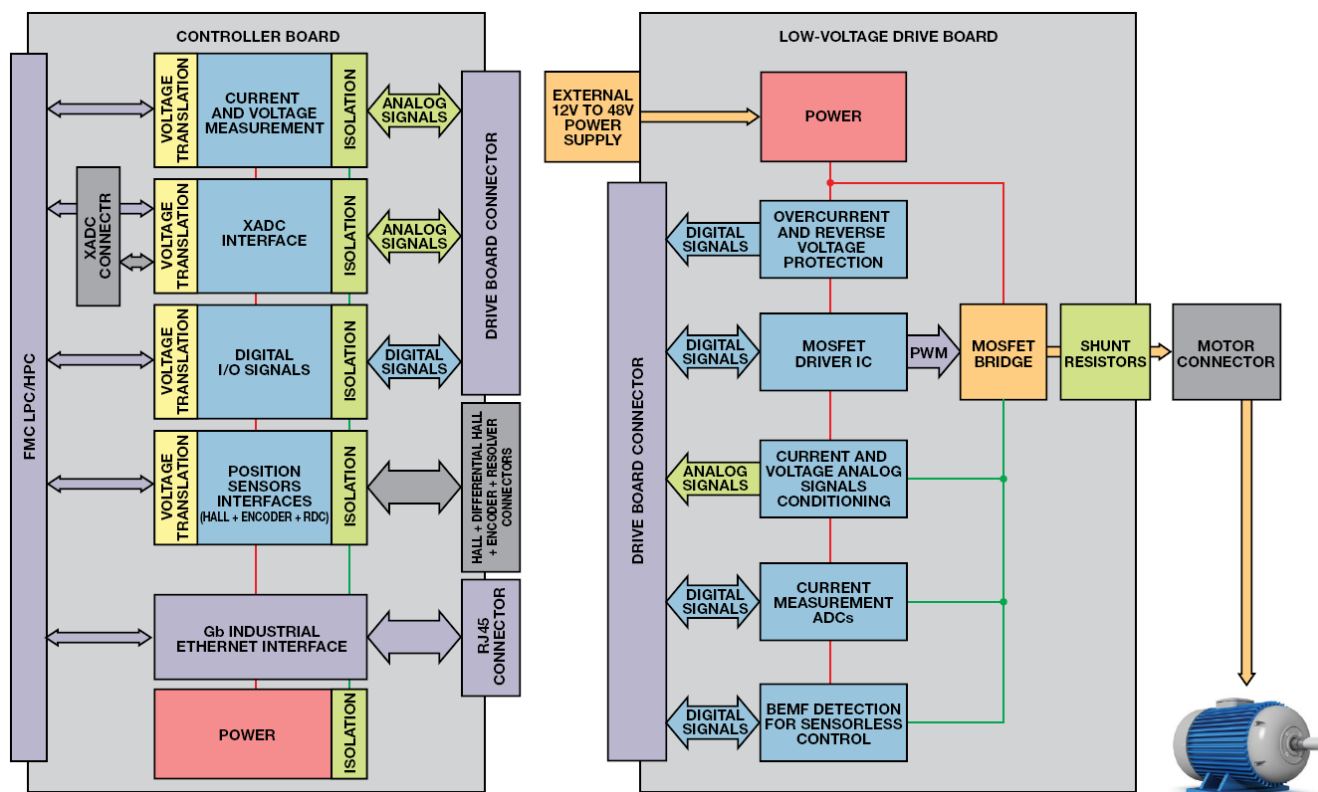


图 4. AD-FMCMOTCON1-EBZ 功能框图

测功器是一个动态可调的负载，可用于测试实时电机控制性能，由两台 BLDC 电机通过刚性连接直接耦合而成。其中一台 BLDC 电机用作负载，由测功器的嵌入式控制系统控制，而另一台由 ADI 公司的智能驱动器套件驱动，如图 5 所示。该系统配有一个用户界面，用于显示有关负载电流和速度的信息，并支持设置不同的负载曲线。要实现外部控制，可利用 Analog Discovery USB 示波器捕捉负载信号；要从 MATLAB® 直接控制，可使用 MathWorks Instrument Control Toolbox™。

电机电流和电压测量的质量对电机控制系统的性能有很大影响。通过利用高性能模拟信号调理器件和 ADC，ADI 公司智能驱动器套件提供精密电流和电压测量。测量路径分为控制器和驱动板两部分，如图 6 所示。

相电流通过测量分流电阻上的电压来检测。取决于 ADC 是否靠近分流电阻，有两条可能的测量路径可以获得最高测量精度。如

果 ADC 靠近分流电阻，则信号路径非常短，不易受到噪声耦合影响。分流电阻上的小差分电压由隔离式 $\Sigma-\Delta$ 调制器 AD7401 直接测量，而无需其他接口和信号调理电路。如果 ADC 远离分流电阻，则信号路径很长，容易受到噪声耦合影响，尤其是电源开关噪声和电机的噪声耦合。必须采取特别措施确保 ADC 与分流电阻之间的 PCB 走线和信号调理电路受到适当的屏蔽。分流电阻上的小差分电压被驱动板上的差动放大器 AD8207 放大，后者置于分流电阻附近以避免噪声耦合。信号从 $\pm 125\text{ mV}$ 满量程输入范围放大至 $\pm 2.5\text{ V}$ 范围，以最大程度地降低耦合噪声的影响。放大后的信号又经过一个采用可编程增益仪表放大器(PGIA) AD8251 的放大级，以确保 ADC 始终接收到经过适当缩放以适应输入范围的输入信号。放大后的模拟信号经过连接器进入控制板。连接器会屏蔽每个模拟信号，以降低噪声耦合影响。来自驱动板的模拟信号利用运算放大器 ADA4084-2 重新转换到 AD7401 输入范围。

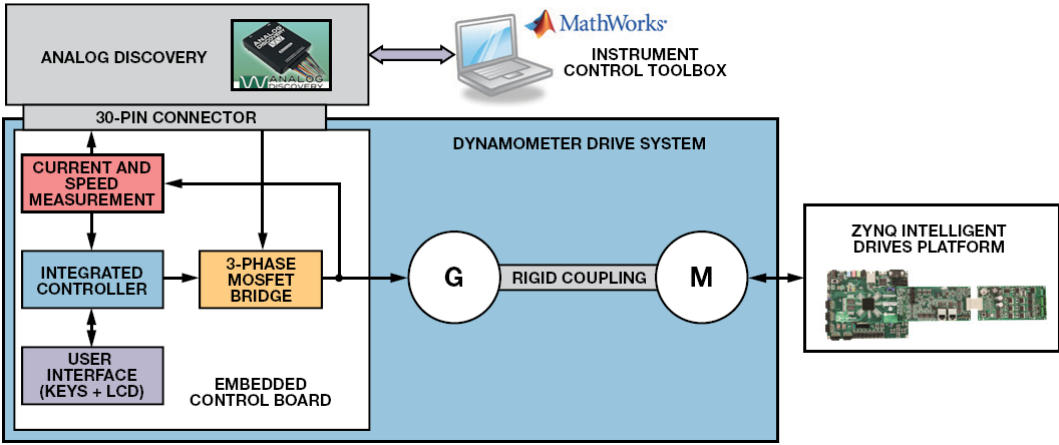


图 5. 测功器驱动系统

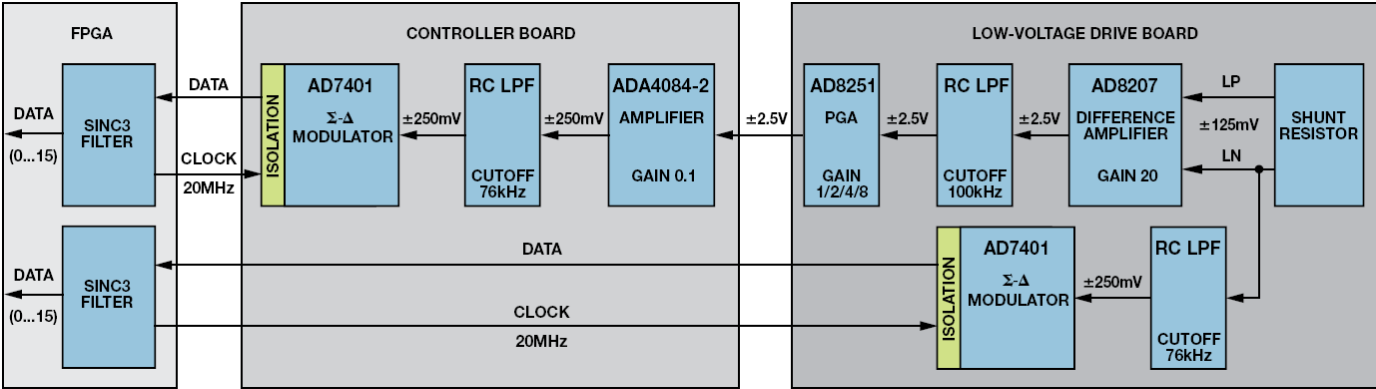


图 6. 相电流信号链

电流和电压反馈信号链中的最重要器件是二阶隔离式 Σ - Δ 调制器 AD7401A。这款高性能 ADC 具有 16 位分辨率（无失码）、13.3 有效位数(ENOB)和 83 dB SNR。2 线数字接口包括一个 20 MHz 时钟输入和一个 1 位数字位流输出。ADC 输出利用 sinc^3 数字滤波器重构。数据手册中提供了一个针对 16 位输出和 78 kHz 采样速率的滤波器模型和 HDL 实现方案。输出分辨率和采样速率可通过改变滤波器模型和抽取来控制。78 kHz 采样速率对许多应用可能是足够的，但某些情况需要更高的速率。这些情况下，可利用图 7 所示的滤波器库来将系统采样速率提至最高 10 MSPS（真 16 位数据）。滤波器库包含 n 个 sinc^3 滤波器，其采样时钟延迟 T （即 sinc^3 滤波器传播时间除以 n ）的倍数。数据选择器以周期 T 输出 ADC 码。

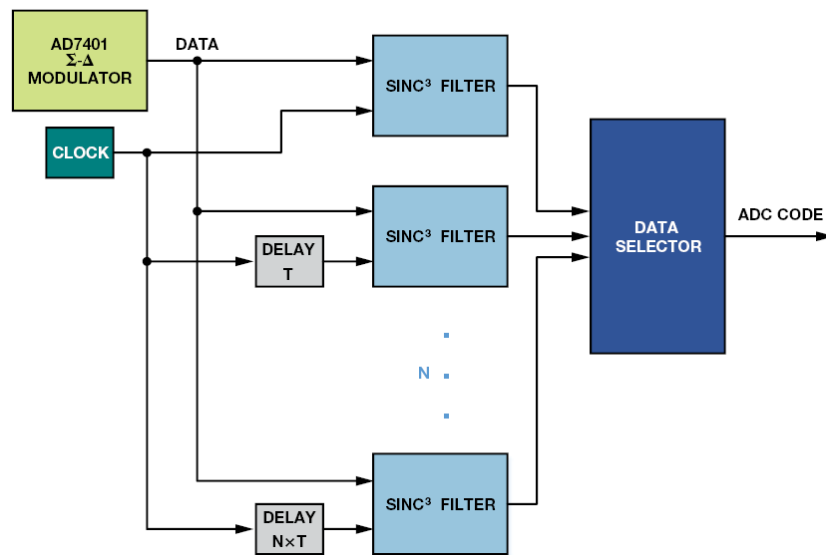


图 7. 滤波器库

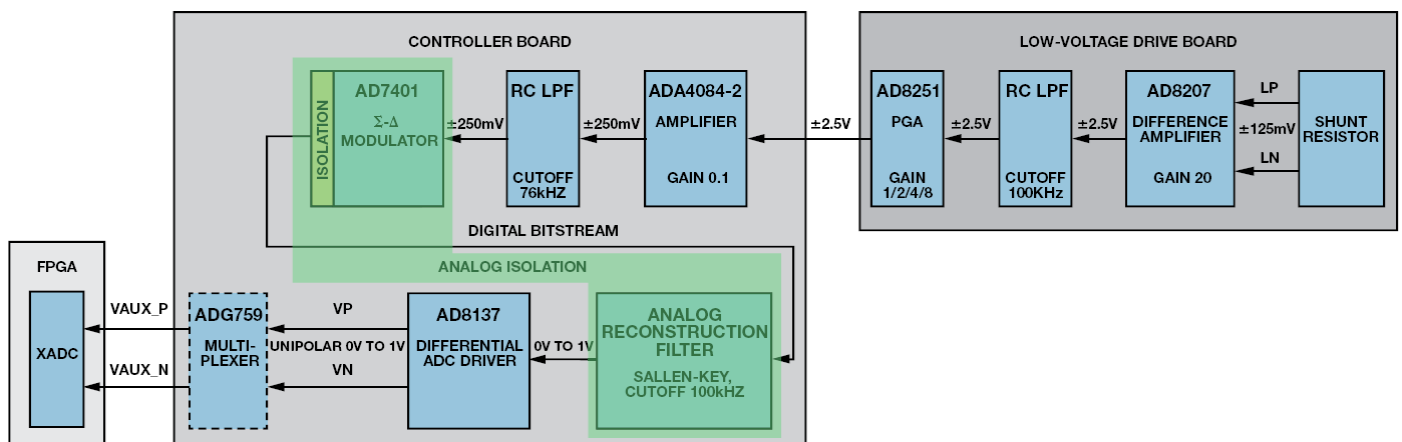


图 8. XADC 信号测量链

可以利用两个控制器模型（一个六步控制器和一个 PMSM 磁场定向控制器）来启动设计过程。图 9 显示了这两个控制器的高级视图。六步控制器实现一个用于 BLDC 电机的梯形控制器；FOC 控制器提供一个 FOC 内核以便集成到控制系统中。

工厂和控制器模型在仿真阶段创建，通过完整系统的行为仿真来验证控制器符合预期。控制器模型划分为由 C 代码和 HDL 实现的多个部分，并指定时序、定点实现、采样速率和环路时间等约束条件以确保控制器模型的行为与在硬件实现中一样。图 10 显示了六步控制器的软件和 HDL 划分。

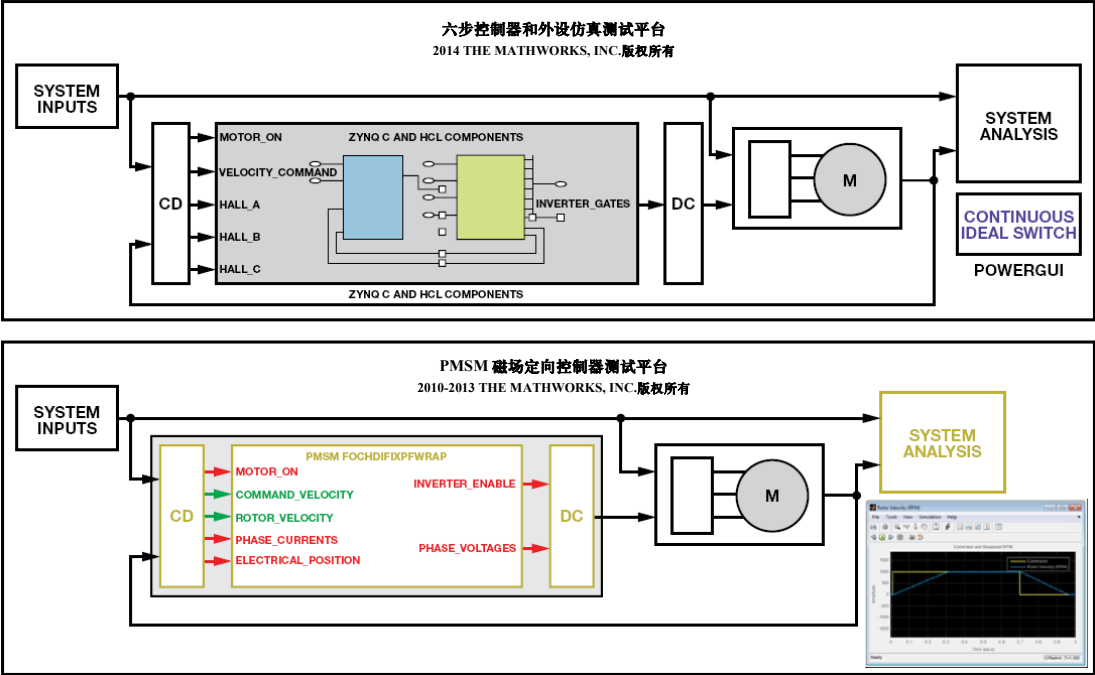


图 9. Simulink 控制器模型

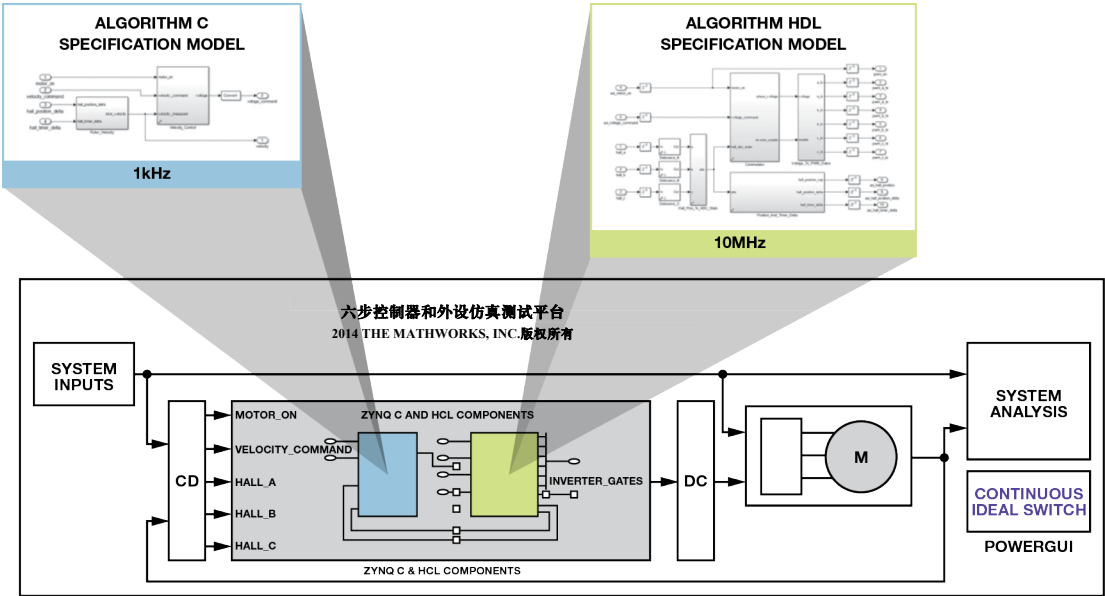


图 10. 控制器的 C 代码和 HDL 划分

一旦控制器在仿真中经过全面验证，下一步便是在硬件平台上制作原型。针对 ARM 内核和可编程逻辑，Zynq SoC 引导工作流程从划分为多个子系统的 Simulink 模型产生 C 代码和 HDL。利用此工作流程，HDL 转码器生成针对可编程逻辑的 HDL，嵌入式转码器则生成针对 ARM 的 C 代码。MathWorks Zynq 支持包支持从模型生成由算法 C 代码组成的 ARM 可执行文件（与 AXI 总线接口），并支持从模型生成由 HDL 代码组成的位流（与可编程逻辑引脚和 AXI 总线接口）。图 11 显示了控制器实现及其与 ADI 智能驱动器硬件的关系。

一旦将位流和可执行文件加载到硬件中，就可以开始控制器的运行测试。利用 Simulink 与运行开源 Linux OS 的嵌入式系统之间的以太网链路执行硬件在环(HIL)测试。轴转速等电机参数可以在 Simulink 中捕捉，并与仿真结果相比较，确保实际系统实现与模型相符。一旦控制算法测试完毕，便可将控制器转移到生产系统上。

除了智能驱动器套件以外，ADI 公司还提供完整的 Vivado 框架和 Linux 基础设施以用于原型开发和最终生产。图 12 显示了支持智能驱动器套件的 Zynq 基础设施。该高级框图说明了 ADI 参考设计在 Xilinx Zynq SoC 上是如何划分的。可编程逻辑实现 IP 内核，用于与 ADC、位置传感器和电机驱动级接口。由 HDL 转码器生成的 HDL 代表电机控制算法，集成到 ADI 公司 IP 中。所有 IP 都有低速 AXI-Lite 接口用于配置和控制，并有高速 AXI-Streaming 接口用于通过 DMA 通道向软件传输实时数据。高速以太网接口可以利用 ARM 处理系统的硬 MAC 外设或可编程逻辑中的 Xilinx 以太网 IP 实现。

ARM Cortex A9 处理系统运行 ADI 公司提供的 Ubuntu Linux，其中包括：与 ADI 公司智能驱动器硬件接口所需的 Linux IIO 驱动，用于监测和控制的 [IIO Oscilloscope](#)（示波器）用户空间应用程序，支持实时数据采集和通过 TCP 控制系统的 [libiio](#) 服务器，在远程计算机上运行的客户端，以及整合嵌入式转码器所生成 C 代码的可选用户应用程序。

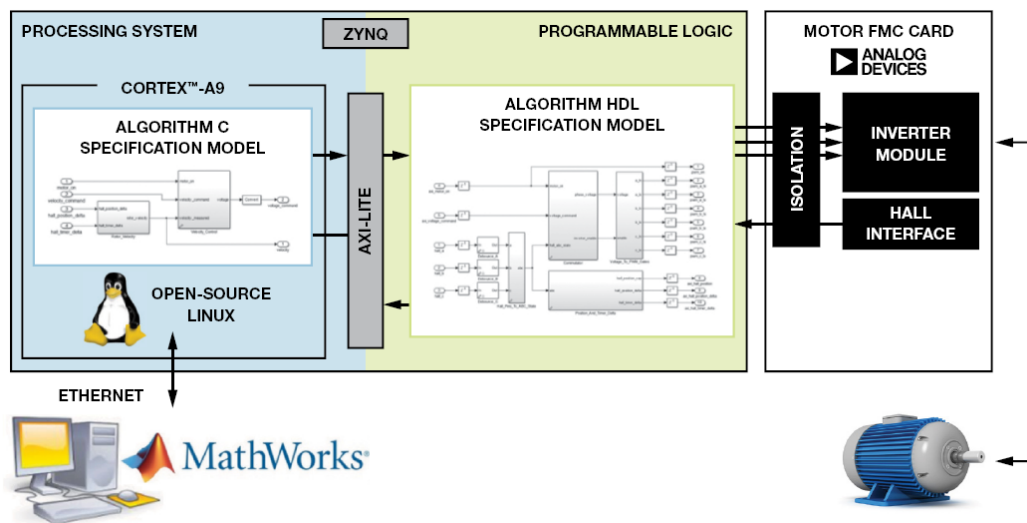


图 11. 原型系统上的控制器实现

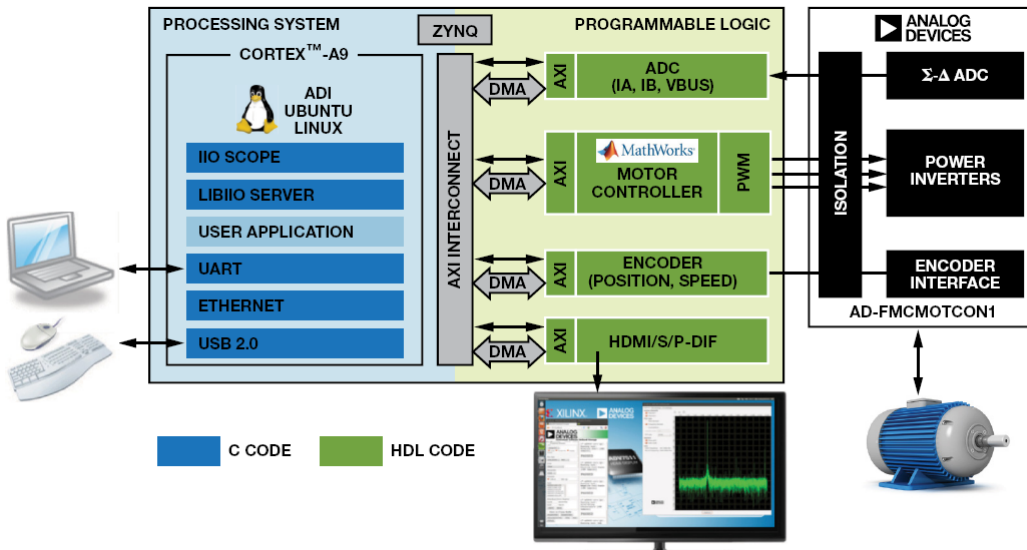


图 12. ADI Linux 基础设施

所有 ADI Linux 驱动均基于 Linux 工业 I/O (IIO) 子系统, 其现已包括在所有主流 Linux 内核中。IIO Scope 是 ADI 公司开发的一款开源 Linux 应用程序, 运行在 Xilinx Zynq 中的双核 ARM Cortex A9 上, 能够显示连接到 Xilinx Zynq 平台的 ADI FMC 卡所获取的实时数据。这些数据可以在时域中、频域中或以星座图的形式显示。支持以不同的常用文件格式 (如逗号分隔值或 .mat Matlab 文件等) 保存所捕获的数据以供进一步分析。IIO Scope 提供一个图形用户界面, 用于更改或读取 ADI FMC 卡的配置。

libiio 服务器支持实时数据采集、通过 TCP 控制系统以及运行于远程计算机上的客户端。服务器运行于 Linux 下的嵌入式目标上, 通过 TCP 管理目标与远程客户端之间的实时数据交换。IIO 客户端可以作为系统对象集成到 MATLAB 和 Simulink 原生应用程序中。一路 HDMI 输出用于在监视器上显示 Linux 界面, 鼠标和键盘可通过 USB 2.0 端口连接到系统。

ADI 公司为智能驱动器套件提供的 Linux 软件和 HDL 基础设施, 连同 MathWorks 和 Xilinx 提供的工具, 非常适合开发电机控制应用原型。它们还包含适用于生产的组件, 可将其集成到最终控制系统中, 从而减少从概念到生产所需的时间和成本。

结论

本文说明了采用 FPGA 的现代电机控制系统的要求和趋势, 以及为满足这些约束条件和帮助实现更高效、更精确的电机控制解决方案, MathWorks、Xilinx 和 ADI 公司带给市场的工具和系统。通过将 MathWorks 基于模型的设计和自动生成代码工具与强大的

Xilinx Zynq SoC、ADI 公司的隔离、功率、信号调理和测量解决方案相结合, 电机驱动系统的设计、验证、测试和实现可以比以前更有效率, 进而提高电机控制性能并缩短上市时间。ADI 公司智能驱动器套件与 Avnet Zynq-7000 All Programmable SoC 配合使用, 为利用 MathWorks Simulink 设计的电机控制算法提供出色的原型开发环境。该智能驱动器套件带有一组参考设计⁴, 旨在为所有希望评估该系统的人士提供一个起点, 并且帮助启动任何新的电机控制项目。

参考文献

1. Hill, Tom. “借助 Matlab 将电机驱动迁移到 Zynq SoC 设计中。” *Xcell 杂志*, 87 期, 2014 年第二季度。
2. Dara O’ Sullivan, Jens Sorensen 和 Aengus Murray “闭环电机控制中基于模型的设计工具。” *PCIM Europe*, 2014.
3. Corradi, Dr. Giulio. “频率空间矢量调制—第一部分。” *EDN 网络*, 2012 年 10 月 4 日。
4. [AD-FMCMOTCON1-EBZ 用户指南](#)。



Andrei Cozma [andrei.cozma@analog.com] 是 ADI 公司工程设计经理, 负责支持系统级参考设计的设计与开发。他拥有工业自动化与信息技术学士学位及电子与电信博士学位。他参与了电机控制、工业自动化、软件定义无线电和电信等不同行业领域的项目设计与开发。



Andrei Cozma

Eric Cigan [Eric.Cigan@mathworks.com] 在 MathWorks 从事技术营销工作, 负责支持 SoC 和 FPGA 设计工作流程。加入 MathWorks 之前, 他先后在 MathStar、AccelChip 和 Mentor Graphics 从事技术营销工作。Eric 拥有麻省理工学院机械工程学士学位和硕士学位。



Eric Cigan