

工业领域的出色表现

MATLAB 助力电机驱动向 Zynq SoC 发展

作者: **Tom Hill**

赛灵思公司 DSP 解决方案高级经理

tom.hill@xilinx.com

工业设计人员可以利用快速原型与基于模型的设计将电机控制算法移至 Zynq SoC 环境。

工业领域的出色表现

自 20 世纪 90 年代以来，电机驱动开发人员一直采用多芯片架构实施电机控制与处理需求。在这种架构中，离散数字信号处理(DSP)芯片负责执行电机控制算法，FPGA 实施高速 I/O 与网络协议，离散型处理器负责处理执行控制。然而，随着赛灵思® Zynq®-7000 All Programmable SoC 的问世，设计人员已经找到了能将这些功能组合到单个器件的方法，而且还可以集成其它处理任务。减少部件数量与复杂性不仅可以降低系统成本，而且还能提升系统性能与可靠性。

但是，驱动开发人员如何才能改变自己长期形成的设计实践，转而采用 Zynq SoC 呢？

长期以来，工业设计人员一直采用基于模型的设计，通过仿真和 C 代码生成，在 DSP 芯片上研发定制电机算法。

如今，赛灵思与 MathWorks 合作开发了一种全新工作流程——将基于模型的设计扩展至可用 Zynq-7000 All Programmable SoC 的处理系统和可编程逻辑。

针对电机控制的 ZYNQ SOCS

如今，先进的电机控制系统都属于控制算法与工业网络的组合，包括 EtherCAT、Profinet、Powerlink 以及 Sercos III，其能够从计算资源中提取处理带宽。此外，控制系统现在还聚合了其它需求，其中包括运动控制层、PLC 层、诊断层以及用户界面，以便进行调试和保养，或远程监控。采用与处理系统自然贴合的元件，这些需求能转化为逻辑分组和实体分组，而其它元件则最好适合于硬件辅助减负与加速。

您所选择的硬件平台应该提供稳健的可扩展型系统。赛灵思 Zynq SoCs 就能满足上述需求，其配有高性能处理系统，以支持联网、运动、soft-PLC、诊断以及远程维护功能，并与可编程逻辑相结合，从而加速硬件中性能临界的功能。对于处理方面，Zynq SoC 与带有 NEON 控制处理器、浮点扩展的 ARM® Cortex™-A9 双核处理系统相结合，可加速软件执行速度。对于可编程逻辑，该器件配有 444000 个逻辑单元和 2200 个 DSP48Slice，能提供大量的处理带宽。利用五个高吞吐量 AMBA®-4 AXI 高速互连，可编程逻辑与处理系统紧密耦合，相当于 3000 多个 PIN 的有效带宽。

表 1 列出了 Zynq SoC 器件所能实现的处理性能。

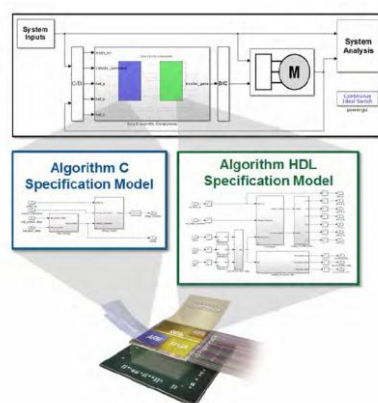


图 1--针对 Zynq SoC 的 MathWorks 工作流程，采用 C 代码与 HDL 代码生成

元件	性能（最高）
处理器（每个）	1 GHz
处理器（合计）	5,000 个DMIP
DSP（每个）	741 MHz
DSP（合计）	2,662 个GMAC
收发器（每个）	12.5 Gbps
收发器（合计）	200 Gbps
软件加速	10x

表 1--Zynq SoC 的处理性能

采用 Simulink 与控制系统工具箱 (CONTROL SYSTEMS TOOLBOX) 的设备与电机模型

现代控制算法的系统时间与系统变量可跨越多个数量级，从而导致硬件/软件分组成为一项艰巨、耗时、重复的任务。图 2 给出了一个典型的电动驱动。其电源频率通常为 50Hz-60Hz，经过整流后，形成连续电压（直流）。然后，该连续电压转换成变频电压，可控制馈送电机终端的电功率级。此外，控制器还必须读取电机基本变量，其中包括电流和电压。控制器同样也必须读取或确定轴的位置，包括轴的转速和通信网络或监视控制器的处理命令。

Simulink®可提供用于多域系统仿真和基于模型设计的方框图环境，其完全适用于含有控制算法和设备模型的仿真系统。基于广泛用于 Simulink 中对控制系统建模进行系统分析、设计与调节的方法，MathWorks 的控制系统工具箱等产品可提供各种各样的“应用”。利用 Simulink 执行系统建模，可以加快电机控制系统的开发速度，而且还可以从以下几个方面降低风险：

- 降低受损风险——在全新控制系统算法用于量产硬件测试之前，仿真法可对此算法进行全面检验，这样就可以规避硬件测试存在损害驱动电子系统、电机以及系统其它组件的风险。
- 加快系统集成速度——支持员工需要将控制系统新算法集成至产品系统，这意味着部署新的控制器会消耗有限的时间，而且可能令部署过程变得冗长。
- 减少对设备可用性的依赖——在有些情况下，生产环境本身并不可用，例如定制驱动电子系统或电机尚在开发当中或者其所处位置使得控制系统设计人员无法对其进行访问。

鉴于上述因素，仿真即是利用生产硬件进行测试的最佳替代方案。Simulink 等仿真环境可以提供框架，以便利用机电组件既存构建模块库创建装置模型，进而利用该装置模型对新的控制系统架构进行评估。

将系统模型与快速原型环境以及最终生产系统相连接，从而进一步降低计划的风险。快速原型工作流程允许算法开发人员可以在不依赖硬件设计人员的情况下进行原型设计。算法开发人员可在高度自动化过程中采用特定平台支持包，进而将系统硬件和软件组件部署到可以编译为特定硬件开发平台的设计模板。硬件与软件设计团队可以将这些相同的硬件和软件组价重新应用于最终生产系统之中，无需做出任何修改即可加速研发过程、减少失误。

采用 Avnet 智能驱动套件实现快速原型

设计人员可以将 vnet Zynq-7000 AP SoC / 模拟器件智能驱动套件与 Simulink 以及 Zynq SoC 工作流程相结合，实现针对电机控制应用的整套快速原型系统。该套件整合了 Zynq SoC 与最新一代模拟器件高精度数据转换器以及数字隔离技术。该套件可支持高性能电机控制与双端口千兆位以太网工业联网 (<http://www.xilinx.com/products/boards-and-kits/1-490M1P.htm>)。

该套件配有 Avnet ZedBoard 7020 基板、模拟器件 AD-FMCMOTCON1-EBZ 模块、额定转速为 4000RPM 的 24V BLDC 电机，同时还配有霍尔效应传感器和 1250-CPR 索引编码器，其中 AD-FMCMOTCON1-EBZ 模块可以驱动带有 24V 外部电源（该套件本身带有）的无刷直流步进式电机。此外，该套件还含有 Zynq SoC 磁场导向控制参考设计以及包括驱动程序、应用软件和源代码的模拟器件 Ubuntu Linux 框架。

实例：电机梯形控制

现在让我们通过 Simulink 仿真将上述工作流程应用于图 1 中的电机梯形控制系统，通过仿真装置评估控制器，然后采用智能驱动套件对该控制器进行原型设计。最后，我们利用硬件

测试的结果来对 Simulink 模型进行验证。

在该实例中,我们将使用智能驱动套件来驱动以铝盘形式存在的惯性载荷以及基础梯形控制器。该控制器的主要组件如下:

霍尔效应传感器——检测电机位置

速度估计器——根据传感器信号计算转子转速

六步换向器——根据转子位置与转速计算相位电压与变频器使能信号。

脉宽调制 (PWM) ——通过驱动回路驱动控制器输出。

我们首先采用适用于控制环路分析的系统行为、控制环路模型。首先,我们利用仿真对模型进行评估,模型接受脉冲测试,指令模型按照 150 转/秒的转速旋转,持续 2 秒,然后停止。通过调节控制回路的比例积分 (PI) 控制器增益,我们可以获得 1.2 秒稳定时间,并且过冲可以忽略不计(图 3 中紫色阴影标志即为控制环路仿真结果;有关本实例的详情,敬请访问 mathworks.com/zidk)。

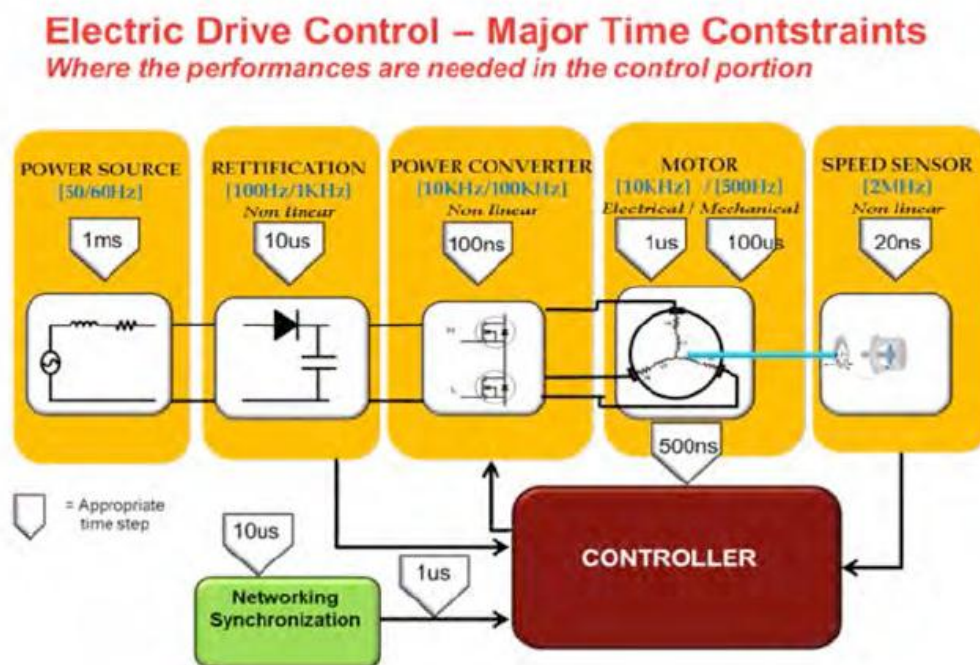


图 2——电机驱动控制器的主要时间约束

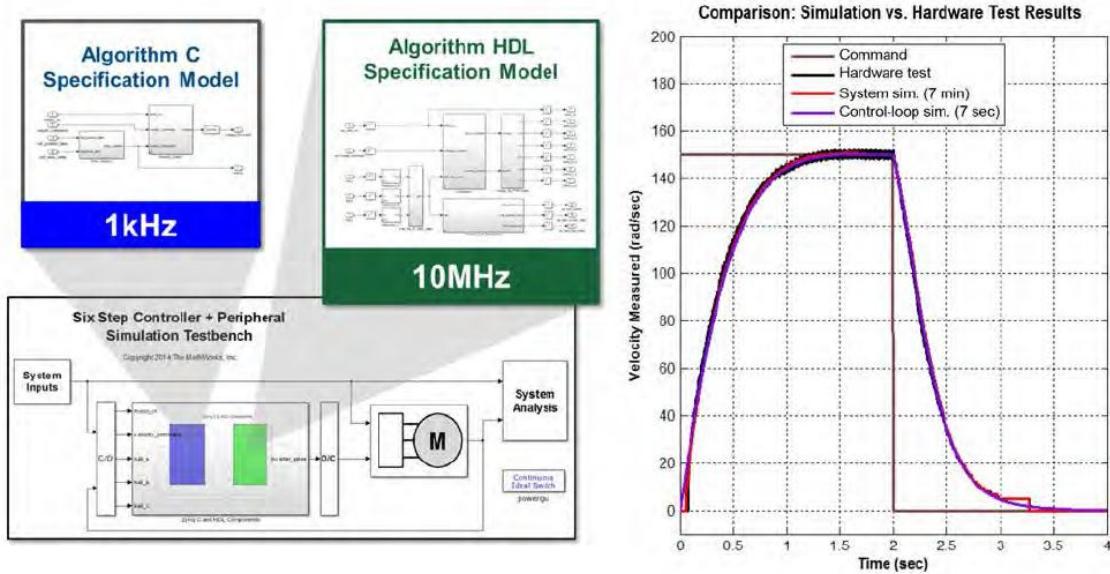


图 3——用于根据硬件测试结果进行验证的硬件与软件仿真模型

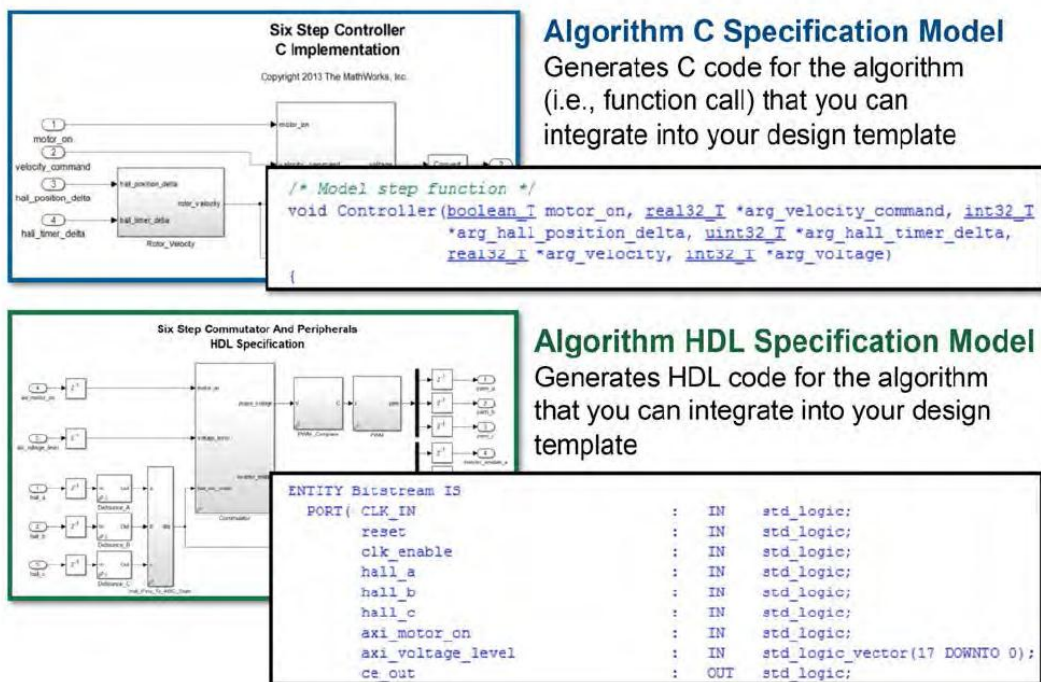


图 4——由分割后的 Simulink 模型生成的 C 代码与 HDL 代码

设置好控制环路增益之后，我们现在可将控制器置于更加精准的系统模型上进行测试。与控制环路模型相比，系统模型内含更加细化的驱动电子系统模型，更为重要的是，系统模型所包含的细化模型可以指定控制器与外设实施，其中包括 PWM 与霍尔效应传感器处理过程中所使用的准确时序模型。

我们已针对 Zynq SoC 规划出了控制器，速度控制器与速度估计器在 ARM 核上运行，频率

为 1kHz，换向器、霍尔传感器以及 PWM 均在 Zynq SoC 可编程逻辑上运行。

我们可以对比控制环路和系统模型的仿真结果（图 3 中红色标志即为系统模型结果）。一般来说，两个波形的相似度非常高，但电机速率除外，它趋向于零。在这些点上，霍尔传感器粗糙度非常明显，而且电机轴每次转动只有六个标志脉冲。这种高保真系统模型 7 分钟内即可运行一次为时 4 秒的仿真，而对于保真较低的控制环路模型而言，只能运行一次为时 7 秒的仿真。对于控制系统设计人员而言，其意义在于上述仿真结果可以让我们更加坚信，控制环路模型对于进一步评估控制器替代方案来说足够准确，无需使用系统模型进行硬件测试，即可验证替代方案。

有了这些发现，我们即可利用智能驱动套件设计出控制器的原型。通过 Zynq SoC 推荐的工作流程，我们可以利用 Simulink 模型生成 C 代码与 HDL 代码，Simulink 模型已被分割成数个特定于 ARM 核以及可编程逻辑的子系统（图 4）。

我们按照上述工作流程，使用 MathWorks 的 HDL 代码生成器生成 IP 核，此 IP 核将在 Zynq SoC 器件的可编程逻辑中运行，以便建立在 ARM 核上运行的可执行程序，并且在 ARM 核与可执行程序之间建立 AXI 总线接口。

将 bit 文件加载载入可编程逻辑以及在 ARM 核上运行可执行程序之后，我们即可进行环路中的硬件测试。对于此测试，我们采用经过改良的 Simulink 测试平台模型，我们已经拆除该模型中的驱动电子系统、电机以及传感器模型，因为我们采用了环路中的硬件取代被仿真的装置模型。为了方便我们检查测试结果，同时方便与仿真结果进行对比，我们可将 Zynq SoC 设置成将电机轴速度测量以及其它数据存储于 ARM 核内存之中（图 3 中黑色阴影标志即为硬件测试结果）。这样，我们即可将结果上传至 MATLAB session，在试验结束时将脉冲输入应用于测试平台，即可处理和查看结果。如此一来，我们即可在硬件中准确重复我们采用仿真进行的测试。原型设计的测试结果与仿真结果非常接近，其中包括因霍尔传感器导致电机速度实测值发生中断。

本简述证明了 MathWorks 的 Zynq SoC 工作流程可应用于基于模型的设计，实现仿真和原型设计。如果想继续深入生产，您可以将所生成的 C 代码和 HDL 代码导入 Vivado® 设计套件，在该套件中您可将这些代码与执行程序、联网 IP 以及完整执行系统所必需的其它设计组件进行整合。

如需下载本文所述模型以及了解更多有关如何利用 Zynq-7000 All Programmable SoC / Avnet 模拟器件智能驱动套件完成基于模型的设计，敬请访问：mathworks.com/zidk。您还可以从本页浏览在 Zynq SoC 器件上执行完整磁场导向控制模型的 Simulink 模型，也可以观看上述实例的视频，了解更多详情。

如需了解 MathWorks 产品如何为赛灵思 Zynq-7000 All Programmable SoC 系列提供支持，敬请访问：mathworks.com/zynq。