

NI CompactRIO高性能控制器：性能和吞吐量基准测试

发布日期：2014年7月10日

概览

CompactRIO高性能控制器基于LabVIEW RIO架构，采用了功能强大的Intel Atom片上系统(SoC)和Xilinx Kintex7 FPGA等最新技术。Intel Atom SoC提供了丰富的应用功能和极高的处理性能，包括一个内嵌的GPU和一颗双核1.33 GHz处理器。Kintex-7325T FPGA，与其他CompactRIO控制器的FPGA相比，包含大约3倍的复杂逻辑块和超过13倍的DSP处理单元。其与处理器的通信采用PCI Express总线，大大提高了数据吞吐量，并可接入16个DMA通道。

全新CompactRIO高性能控制器采用的LabVIEW RIO架构提供了极大的软件灵活性，使得人机界面（HMI）和控制任务可同时集成到同一个硬件和同一个操作系统中。这主要得益于Intel Atom SoC和NI Linux Real-Time操作系统提供的一些特性。NI Linux Real-Time操作系统将任务优先级开放给开发人员，使其可以在同一应用程序中为控制任务分配比HMI更高的优先级。

这些技术的结合大大提高了闭环控制应用的系统吞吐量并降低了延迟。CompactRIO高性能控制器为嵌入式设计人员提供了灵活强大的硬件，同时还可帮助他们降低系统复杂性和成本。

为了验证CompactRIO高性能控制器的性能，NI研发团队设计了一系列基于cRIO-9031高性能控制器的控制和监测应用基准测试。这些测试同样适用于其他使用相同CPU的CompactRIO高性能控制器产品。

目录

- 1. 基准测试设置和测量
- 2. 测试1——LabVIEW FPGA控制采样范例的性能基准测试
- 3. 测试2——监测应用的吞吐量基准测试
- 4. 测试3——复杂真实应用的性能基准测试
- 5. 嵌入式用户界面的影响
- 6. 结论
- 7. 下一步

1.基准测试设置和测量

我们选择了具有代表性的常见控制和监测应用以及这些应用中常见的主要任务进行基准测试。下表列出了本次实验要进行的测试。

TEST TYPE	APPLICATION	METRIC
Test1 Performance	LabVIEW FPGA Control on CompactRIO Sample Project	CPU Usage vs. Control Loop Rate
Test 2 Throughput	Monitoring Application	CPU Usage vs. I/O Data Streaming
Test 3 Performance	Complex Real-World Application	CPU Usage vs. Control Loop Rate

表 1 - 用于测量 NI CompactRIO 控制器性能和吞吐量的测试项

本实验的测试对象是以下硬件终端：

- NI cRIO-9025 | 800 MHz, PowerPC
- NI cRIO-9068 | 667 MHz, ARM Cortex-A9双核
- NI cRIO-9031 | 1.33 GHz, Intel Atom双核

• NI cRIO-9082 | 1.33 GHz, Intel Core i7双核

虽然本实验重点关注的是CompactRIO控制器的性能及其与同一产品家族的现有控制器（如cRIO-9025）的性能对比，但是其他CompactRIO产品系列的代表性硬件终端，如cRIO-9068和cRIO-9082，也包含在测试中，以便读者更完整地了解CompactRIO系统所提供的各种可能性。

我们选择CPU使用率作为通用指标来比较这些CompactRIO控制器的相对性能。对于每项测试，CPU使用率的测量是在改变数据传输的通道数或循环控制的周期后，CPU达到稳态时进行的。这些测量可以帮助我们了解可用于进一步扩展代码、提高循环速率或增加数据传输通道的资源。

其他考量因素：

对于搭载多核处理器的CompactRIO系统，取每个内核测试结果的平均值。

对于NI cRIO-9031控制器，分别在启用和禁用嵌入式用户界面的条件下进行测试，以测量该功能对系统整体性能的影响。

2. 测试1——LabVIEW FPGA控制采样范例的性能基准测试

全新高性能CompactRIO系统采用的技术组合为复杂的实际应用带来了无可比拟的性能提升。针对控制应用的测试，我们选择基于CompactRIO的LabVIEW FPGA控制采样范例。并在该采样范例的基础上增加了在Real-Time控制器上执行八通道三次样条轨迹生成算法的任务，以进一步增大CPU的负荷。

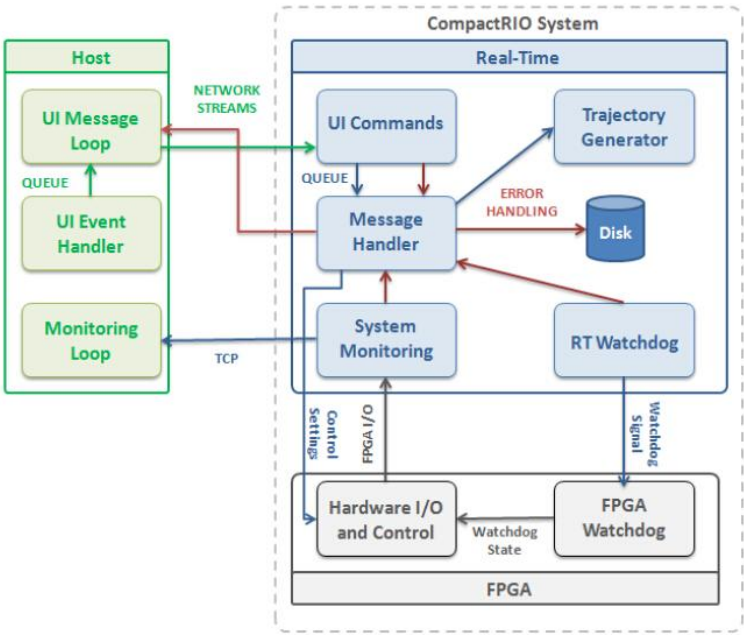


图 1 LabVIEW FPGA 控制采样项目中基于 CompactRIO 的 8 并行通道三次样条轨迹生成算法架构框图

下图显示了循环控制速率为 1.5 kHz 时的 CPU 使用率。在该速率下，cRIO-9031 高性能控制器使用约 11% 的处理器资源，而 cRIO-9068 和 cRIO-9025 控制器分别需要 21% 和 64% 的处理器资源，相比之下，cRIO-9031 高性能控制器具有明显的优势。

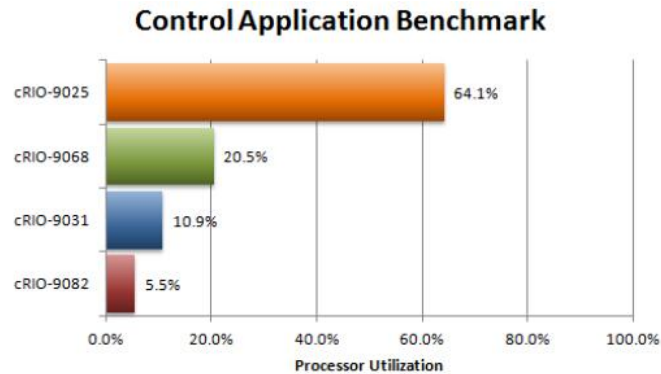


图2 - 循环控制速率为 1.5 kHz 时运行 8 通道三次样条轨迹生成算法的控制应用程序所需的处理器资源比较

对于cRIO-9031高性能控制器，我们在启用嵌入式用户界面的设置下又重复进行了该测试，由于本次测试对人机界面部分要求并不高，因此CPU占用率并没有明显的变化。

3.测试2——监测应用的吞吐量基准测试

监测应用中一个常见的任务是将数据从 I/O 通道搬移到实时处理器进行进一步处理、记录或可视化。我们设计了一个简单的测试来重现这一场景，并测量数据传输通道数量发生变化时的 CPU 使用率。每个通道以 100kHz 的速率传输 16 位采样数据。测试不包含任何数据流盘或在线处理。

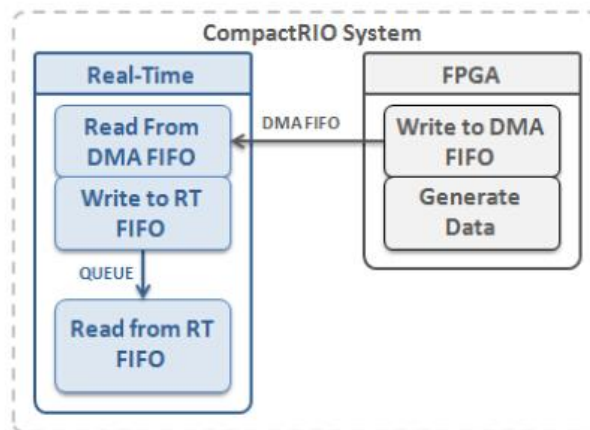


图3 - 监测应用中一个简化的数据传输任务的架构框图。

下图显示了以100kHz的速率读写60个通道数据的测试结果。在这个测试中，我们必须意识到由于硬件架构之间的细微差别可能会给测试带来潜在的瓶颈。例如，cRIO-9068和cRIO-9031均具有16个DMA通道和较高的理论最大数据传输速率：cRIO-9068控制器为320MB/s和cRIO-9031控制器为250MB/s。然而，cRIO-9031高性能控制器由于采用性能明显较高的处理器而能够以较低的CPU利用率维持更多的数据流通道。

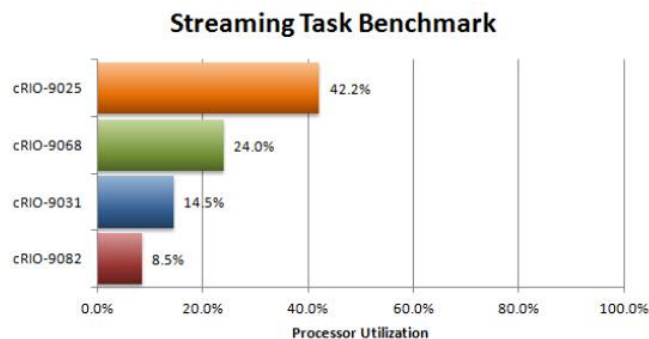


图4 – 每通道速率为100kHz时维持60个通道的16位采样数据流所需的处理器资源

与控制应用性能测试类似，由于本测试对人机界面要求不高，因而cRIO-9031的嵌入式用户界面选项对测试结果并没有太大的影响。如需了解嵌入式用户界面功能的影响，请查看测试3。

4.测试3——复杂真实应用的性能基准测试

充分发挥cRIO高性能控制器功能的最佳方式也许是基于复杂真实应用的测试。大型应用通常结合了各种常见的监测和控制任务，包括具有多速率控制的多个处理循环、数据处理、从I/O通道读写数据、流盘、通过网络将数据传送到远程人机界面以及执行非时间关键型健康和状态监测任务。

我们开发了一个包含上述任务的复杂应用程序，使用诸如RT FIFO、定时循环和网络流等常见的LabVIEW架构组件来协调各种应用程序组件并实现它们之间的通信。此外，还设计了一个专门的任务，以较高的速率将数据发布到前面板显示控件，以充分发挥cRIO高性能控制器的嵌入式用户界面功能。

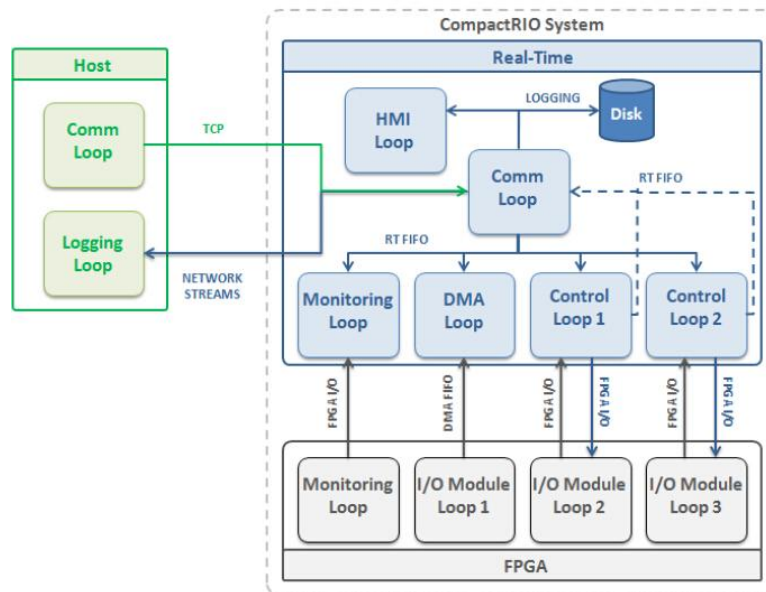


图5 – 一个复杂真实控制和监测应用的架构框图

使用这一应用程序，NI cRIO-9025可实现大约900Hz的最大控制循环速率。在该速率下，NI cRIO-9025的CPU使用率是86%，而cRIO-9031高性能控制器仅使用可用处理器资源的18%就能达到相同的速率。其富余的大量CPU资源可用于额外的应用程序任务或者将本应用的循环速率提高到2kHz以上。

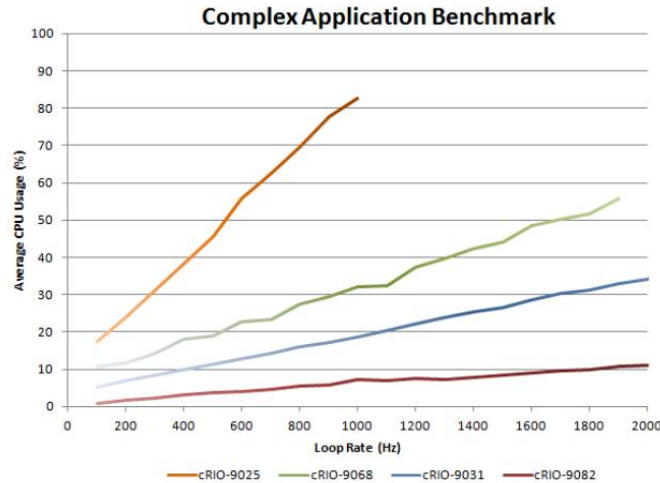


图6 – 在不同控制循环速率下运行包含多个常见控制和监测任务的复杂应用所要的CPU使用率。

5. 嵌入式用户界面的影响

CompactRIO高性能控制器通过利用**嵌入式用户界面**支持来实现本地HMI，降低了系统成本和复杂性。这主要归功于这款全新的控制器采用了最新的Intel Atom处理器，该处理器及其附属的图形化支持为使用NI LabVIEW构建控制系统逻辑和触摸用户界面提供了高性能基础。为了实现该集成，LabVIEW 开发环境和NI Linux Real-Time操作系统将任务优先级开放给开发人员，为控制任务分配比在同一个应用程序中运行的HMI任务更高的优先级。

在该方法中，终端的资源用于系统控制和驱动用户界面的图形化显示。这降低了系统的硬件成本和维护负担，简化了软件开发的复杂性，但相对于其他的显示方案会消耗更多的控制器资源。

为了测量嵌入式界面对处理器资源的影响，我们对测试3进行修改，在高速率下更新其显示组件，以进一步增大其HMI负载。下图显示了在测试3中用户界面非常活跃的情况下使用嵌入式用户界面功能的影响。

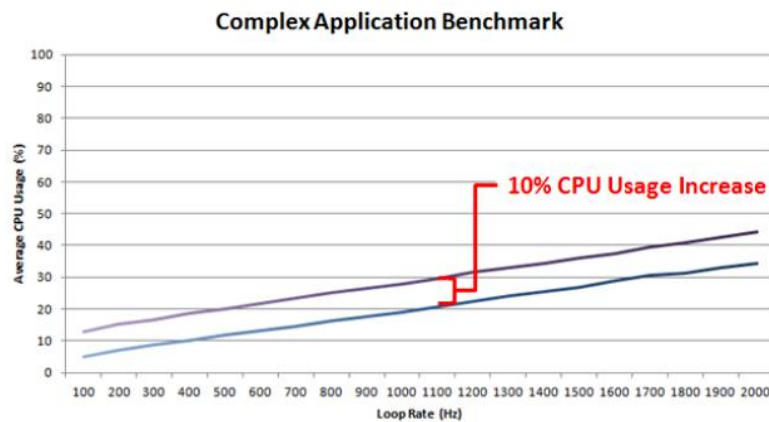


图7 – 在启用嵌入式用户界面功能并高速率地更新用户界面时，增加的CPU使用率大约固定在10%。

在高更新率下，使用嵌入式用户界面功能会导致CPU使用率增加约10%。类似于测试1和2，无论数据流通道的数量如何变化，嵌入式用户界面功能的影响保持恒定。

为了支持嵌入式用户界面的图形处理能力，Intel Atom SoC采用了板载GPU来增强CPU。GPU通过频繁的处理器中断调用来与处理器进行通信，中断调用需要占用一定的LabVIEW Real-Time应用程序处理时间。由于Intel Atom CPU的高性能，在低kHz级的循环控制速率下更新GPU导致变化的处理时间的增加是可以接受的，无需修改架构或配置。为了实现接近5 kHz或以上的控制循环速率，同时仍然利用内嵌式用户界面，可以禁用GPU或将LabVIEW控制代码转移到板载FPGA上，详情可参考

以下链接。

对于嵌入式用户界面可能会影响系统性能的应用，请遵守以下建议：

- 通过LabVIEW中的处理器内核任务分配执行来进一步隔离任务。[了解更多](#)
- 如果需要进一步隔离和确保硬件可靠性，则将LabVIEW控制代码转移到板载FPGA上。[了解更多](#)
- 禁用Intel Atom SoC 的GPU，以牺牲CPU利用率来减小抖动。[了解更多](#)。

6. 结论

嵌入式应用可执行的功能越来越多，如运动控制、数据采集、视觉采集和人机界面。由于NI CompactRIO高性能控制器集成的先进技术和独特功能，可大幅加快开发时间并降低系统复杂性和成本，因而成为这些应用的理想选择。

CompactRIO高性能控制比其他控制器系列产品有了显著的性能提升，而且扩展了可兼容诸如NI Linux Real-Time等关键技术的终端范围。NI CompactRIO高性能控制器基于LabVIEW RIO架构为各种嵌入控制和监测应用提供了灵活、强大的硬件和软件支持。

通过一系列的基准测试，证明了cRIO-9031高性能控制器相比同类cRIO控制器产品家族中的现有产品，如cRIO-9025，性能大约提升了4倍。然而，整个测试也表明，cRIO-908x控制器仍是目前CompactRIO产品线中性能最高的控制器。有关这些控制器的更多信息，请访问www.ni.com/compactrio。

7. 下一步

了解更多关于[NI CompactRIO](#)高性能控制器

了解更多关于[Intel Atom](#)片上系统

了解更多关于[Xilinx Kintex 7 FPGA](#)