

本文档介绍在 Altera® FPGA 系统设计上怎样发挥芯片网络 (NoC) 体系结构的优势。NoC 体系结构在芯片系统设计通信子系统上应用了网络方法和技术。与传统的非 NoC 互联结构相比, NoC 互联体系结构有很明显的优势, 例如, 支持独立层设计和优化等。Altera 的 Qsys 系统集成工具含在 Quartus® II 软件中, 可以根据应用需求, 自动产生灵活的 FPGA 优化 NoC。与传统的互联结构相比, Qsys 互联结构在相对应的延时和资源特性条件下, 进一步提高了工作频率, f_{MAX} 提高了 2 倍。

引言

FPGA 器件密度提高到百万逻辑单元 (LE), 因此, 设计团队需要实现更大更复杂的系统, 在更短的时间内满足越来越高的性能需求。设计人员可以使用系统级设计工具, 以最少的投入迅速设计高性能系统。

Qsys 使用 NoC 体系结构来实现系统工作。Qsys 互联结构的特性包括支持 FPGA 高性能工作, 例如, 针对某些应用只占用很少资源就可以实现灵活的网络互联, 根据所支持系统而随时变化的数据包格式, 以及命令和响应网络分开的网络拓扑, 从而提高了并行处理能力, 降低了对资源的占用。

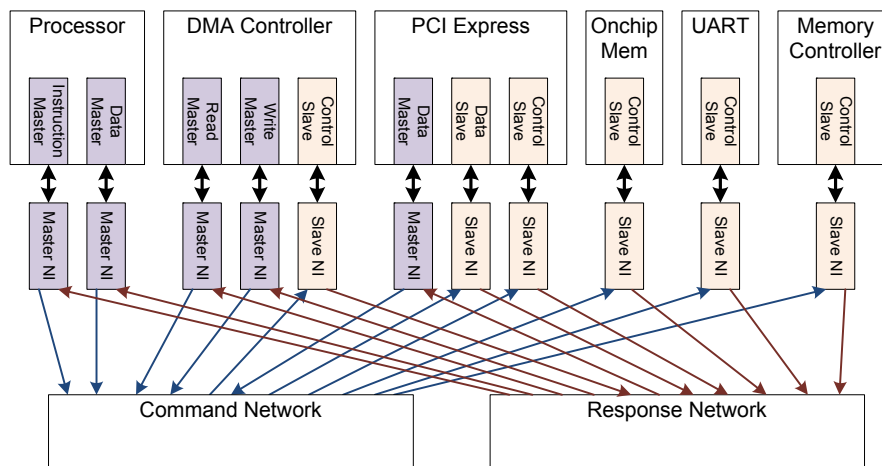
本白皮书解释 Qsys 网络是怎样实现的, 讨论其优点, 对比传统的互联和 Qsys 互联系统的性能结果。这些结果表明, 在相同的延时特性下, NoC 实现的频率更高, 使能流水线选项后, 频率能够提高两倍。

理解 NoC 互联

NoC 互联将实体之间的通信问题分解成小问题, 例如, 怎样在系统节点间传送操作指令, 怎样将操作指令包封成数据包来进行传送等。NoC 互联不同于传统的互联结构, 其实现方式简单却非常高效。NoC 方法并不是将互联结构当做系统中的单个组件来处理, 而是把互联当做协议堆栈, 不同层实现互联结构的不同功能。TCP-over-IP-over-Ethernet 等传统协议堆栈的优势在于每一层的信息由下一层进行包封。Qsys NoC 实现的优势来自于相同源, 在协议堆栈每一层对信息进行包封。

图 1 所示为 NoC 系统的基本拓扑。网络中的每个端点接口、主机或者从机链接至网络接口 (NI) 组件。网络接口使用会话层协议采集操作指令或者响应, 以相应格式的数据包将其传送至网络。数据包网络将数据包传送至相应的数据包端点, 然后传送至其他的网络接口。然后, 网络接口终结该数据包, 使用会话层协议向主机或者从机发送命令或者响应。

图 1. NoC 系统基本拓扑



在这一系统中，处理器等组件与存储器控制器等组件进行通信。每一组件使用网络接口服务，通过会话接口实现彼此之间的通信，例如，Altera 的 Avalon®存储器映射 (Avalon-MM) 接口或者 Advanced eXtensible 接口 (AXI) 等。网络接口相互之间进行通信，依靠命令和响应网络服务来提供会话层服务，传送层提供传送服务。传送层的每一组件（在命令和响应网络中）识别出传送层协议，但是不需要详细确定每一数据包中的操作指令。

NoC 体系结构的优点

与高级高性能总线 (AHB) 或者 CoreConnect 等传统方法相比，分开协议堆栈层具有以下优点：

- 独立实现和优化每一层
- 简化每一应用的定制实现
- 在网络的不同部分支持多种拓扑和选择
- 简化功能开发，提高接口互操作性和灵活性。

实现并优化每一层

解决复杂工程难题的一种常用方法是将设计问题分解成小问题，这些小问题定义好了相互间的操作。采用 NoC 互联，设计问题不再是“我怎样才能针对一个复杂系统来很好的设计灵活的互联结构呢？”，而是由更简单的问题组成：“我怎样很好的将操作指令映射到数据包中？”，以及“怎样很好的传送数据包？”。将每一层分开，还有助于您单独优化实现每一层，进一步提高每一层的性能，而不用重新设计其他层。例如，设计人员可以考虑并实现多个不同的传送层拓扑，而不用对会话层进行任何改动。

简化每一应用的定制实现

在传送层，命令和响应是由网络承载的简单数据包，网络层完成的任何操作仅支持这些数据包的传送。与传统互联结构相比，这简化了某些应用中互联的定制实现。例如，如果设计人员确定系统需要在一组主机或者从机之间采用流水线或者时钟交叉结构，那么，只要能够安全的传送数据包，设计人员就可以增加所需的组件。时钟交叉和流水线并不需要完成会话层的职能，例如，不同的会话类型，响应类型和突发类型等。

使用多种拓扑和选择

NoC 互联结构支持在网络的不同部分使用不同的优化方法和拓扑结构。例如，设计中可能会有一组高频、大吞吐量组件，这包括处理器、PCI Express®接口、DMA 控制器和存储器；以及另一组低吞吐量外设，包括定时器、UART、闪存控制器和 I2C 接口等。这类系统可以在传送层进行划分。设计人员将高性能组件放在带宽较宽的高频交换网络上；而把外设放在成本较低的网格网上，在两个网络之间建立数据包桥接。

简化功能的开发

互联结构应具有很好的通用性，从而支持新出现的网络功能，例如，新的会话类型或者突发模式。如果互联结构被分成了不同的层，那么，增加新功能只需要改动支持该功能的层。例如，为能够支持新的突发模式，只需要改动网络接口组件。同样的，如果新的网络拓扑或者传送技术进一步提高了性能，那么，这些技术可以替代原有网络技术，而不用重新设计整个网络。

接口互操作性

不同的知识产权 (IP) 内核支持不同类型的接口，例如，AMBA® AXI、AHB 和 APB 接口；以及 OCP 接口、Wishbone 接口和 Avalon-MM 接口等。支持新接口只需要采用所选择的数据包格式，网络接口封装来自该类型接口的操作指令。采用这一体系结构，不需要桥接组件，从而节省了逻辑资源，降低了延时。

灵活性

具有数百个主机和从机的系统已经比较常见了，传统的互联结构很难满足所需的性能要求。针对数十个主机和从机而设计的互联结构不容易进行更新来支持目前系统所需要的数百个组件。采用 NoC 互联结构，比较容易将网络分成子网络，根据需要，在网络中采用桥接、流水线级和时钟交叉逻辑。因此，多跳网络很容易支持数千个节点，甚至能够在多个 FPGA 之间提供传送网络。

Qsys 中的 NoC 系统设计

Qsys 是功能强大的系统开发工具，它包含在 Altera 的 Quartus II 开发软件中。Qsys 简化了 FPGA 系统设计，帮助设计人员轻松建立高性能系统，而不需要详细了解片内互联或者网络。Qsys 包括大量的 IP 库，与使用传统的手动集成方法相比，设计人员可以更迅速的从中构建并实现芯片系统 (SoC)。使用传统的设计方法，设计人员必须编写 HDL 模块来链接系统组件。而采用 Qsys，设计人员使用 GUI 或者脚本系统描述对系统组件进行例化和参数赋值。然后，简单的按下按键，Qsys 就能够产生组件和互联结构。图 2 显示了在 Qsys 中建立的简单系统。

图 2. Qsys 中显示的系统组件实例

Connections	Module	Description	Base	End
	[-] nios2_qsys_0	Nios II Processor		
	data_master	Avalon Memory Mapped Master		
	instruction_master	Avalon Memory Mapped Master		
	d_irq	Interrupt Receiver	IRQ 0	IRQ 31
	jtag_debug_module	Avalon Memory Mapped Slave	0x00000000	0x000007ff
	[-] sgdma_0	Scatter-Gather DMA Controller		
	csr	Avalon Memory Mapped Slave	0x00000000	0x0000003f
	descriptor_read	Avalon Memory Mapped Master		
	descriptor_write	Avalon Memory Mapped Master		
	csr_irq	Interrupt Sender		
	m_read	Avalon Memory Mapped Master		
	m_write	Avalon Memory Mapped Master		
	[-] uart_0	UART (RS-232 Serial Port)		
	s1	Avalon Memory Mapped Slave	0x00000000	0x0000003f
	irq	Interrupt Sender		
	[-] pcie_hard_ip_0	Altera PCIe Hard IP		
	Txs	Avalon Memory Mapped Slave	0x00000000	0x000fffff
	Bar1_0	Avalon Memory Mapped Master		
	Cra	Avalon Memory Mapped Slave	0x00000000	0x00003fff
	Cralrq	Interrupt Sender		
Rxmlrq	Interrupt Receiver	IRQ 0	IRQ 63	
[-] onchip_memory2_0	On-Chip Memory (RAM or ROM)			
s1	Avalon Memory Mapped Slave	0x00000000	0x00000fff	
[-] uniphy_ddr3_0	DDR3 SDRAM Controller with UniP...			
avalon_slave_0	Avalon Memory Mapped Slave	0x00000000	0x007fffff	

在 Qsys 中，系统设计人员使用 GUI 向系统中加入所需的 IP 组件，对每一个参数进行赋值，设定系统组件之间的接口级链接。Qsys 自动连好接口之间的每一信号。Qsys 以 RTL 的方式来产生系统，管理系统互联问题，例如，时钟域交叉、接口宽度自适应和突发自适应等。

Qsys 支持多种不同类型的接口，例如，会话（读操作和写操作）接口、流（数据包或者非数据包）接口、中断和复位等。Qsys 会话互联基于 NoC 实现，它专门针对 FPGA 进行了设计。Qsys 互联减少了对 FPGA 资源的占用，而同时支持高性能系统，满足了高频和大吞吐量要求。

针对 FPGA 优化了 Qsys NoC 互联结构

Qsys NoC 互联结构特性使其非常适合 FPGA 应用以及采用了 FPGA 的系统，实现起来非常灵活，其特性还包括为提高适应能力而设计的可参数赋值数据包格式，低延时互联，以及分开的命令和响应网络等。

简单灵活的实现

Qsys 互联不仅仅是为了提高多千兆位数据通路和复杂突发系统的性能，还在于实现由较少组件构成的简单系统。为支持各种各样的系统，Qsys 只实现所需的最少互联结构，以满足某些应用的性能要求。

Qsys 首先将系统分成多个互联域。如果两个接口需要采用系统算法将其链接起来，而系统中没有链接，那么，两个接口将位于不同的互联域中。例如，如果一个主机链接两个从机，这些从机位于相同的互联域中。对于每一个域，Qsys 考虑所有主机和从机宽度，根据域中接口的时钟速率，将网络数据宽度设置为最小，支持系统中最大吞吐量链接的全部吞吐量。

而且，Qsys 只加入应用所需要的互联组件。例如，如果系统中的一个主机只链接至一个从机，那么，可以不采用地址解码组件。如果一个从机只链接至一个主机，那么，可以不采用仲裁组件。如果这一应用并不需要某一类型的突发自适应功能，那么，可以忽略对这一突发自适应功能的支持。

可参数赋值数据包格式提高了适应能力

Qsys 不但减少了对互联资源的占用，而且还判断数据包格式，使用更少的逻辑资源，提高适应能力。例如，尽量减小数据包中地址和突发计数域宽度来支持系统。将数据包中的地址和其他域设置为数据包所有周期中 useful 而且正确的数值，这样，自适应组件就不需要维持数据包的所有状态，在某些情况下甚至可以忽略适配器。

低延时互联结构

设计人员考虑到基于数据包的方法，一般对数据包进行串行处理，每一周期中只进行部分会话操作。很多 NoC 实现使用这一方法。这种 NoC 实现方法的网络延时在 12 到 15 个时钟周期之间，不适合实现微控制器和本地存储器之间的互联。为克服延时问题，Qsys 互联中的组件都采用了组合数据通路。数据包格式足够宽，在一个时钟周期中可以包含完整的会话操作，因此，整个互联能够支持延时 0 周期写操作和 1 周期往返延时读操作。目前的 FPGA 能够很好的支持这些宽带链接。系统设计人员可以改变流水线选项来提高频率，代价是增加了延时。

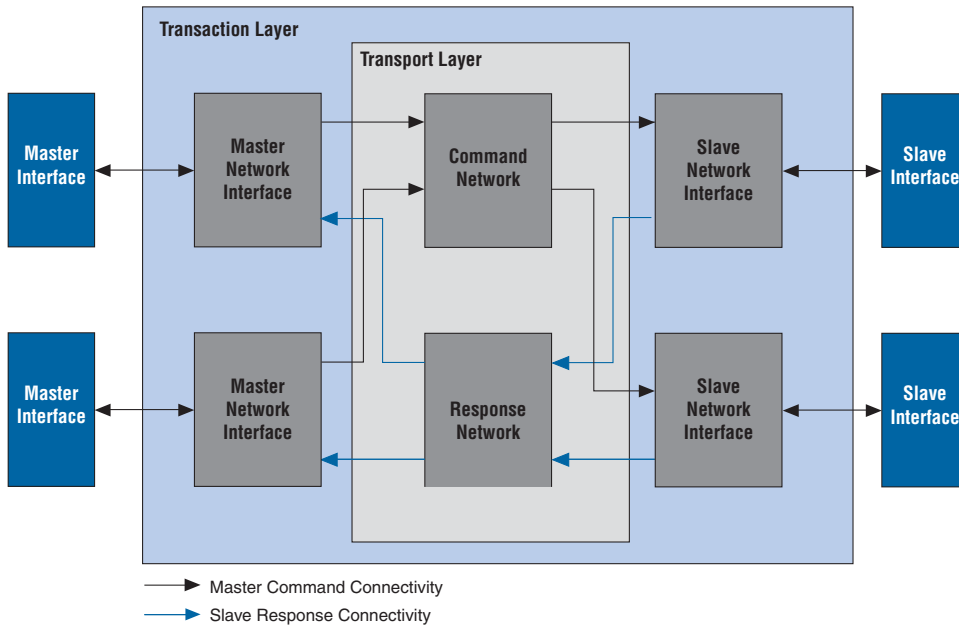
分开的命令和响应网络

对于每一会话操作域，Qsys 例化两个独立的数据包网络，一个用于命令数据流，一个用于响应数据流，而不是例化支持两种数据流的一个网络。命令数据流和响应数据流不会竞争使用网络节点间的链路等资源，因此，这提高了并行处理能力。Qsys 还支持单独对两个网络进行优化，因此，两个网络可以采用不同的网络拓扑和数据包格式。

优化命令和响应网络

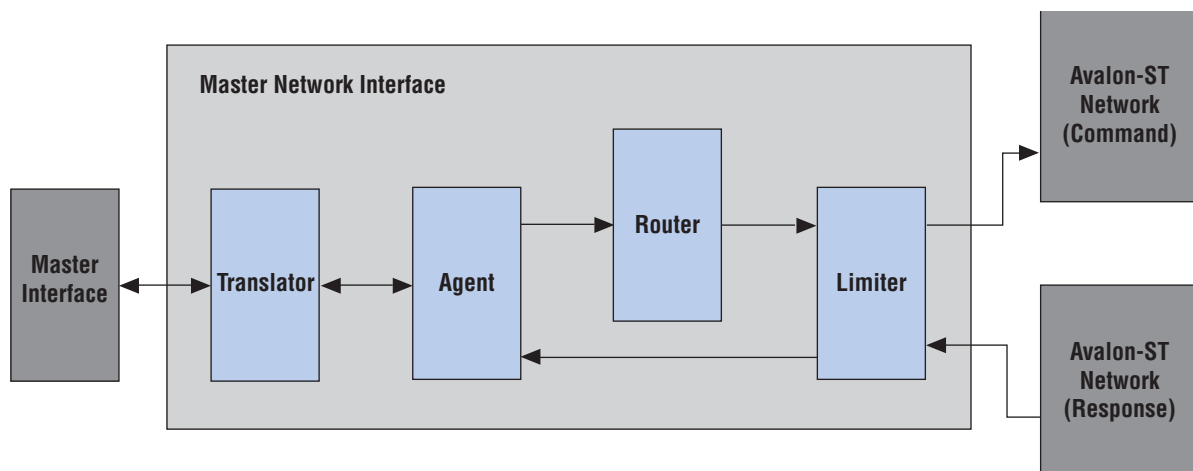
下面的步骤介绍了主机怎样向目的从机发送读命令，以及从机怎样向主机返回响应，简要介绍了 NoC 互联中的命令和响应网络，如图 3 所示。

图 3. Qsys NoC 互联拓扑结构



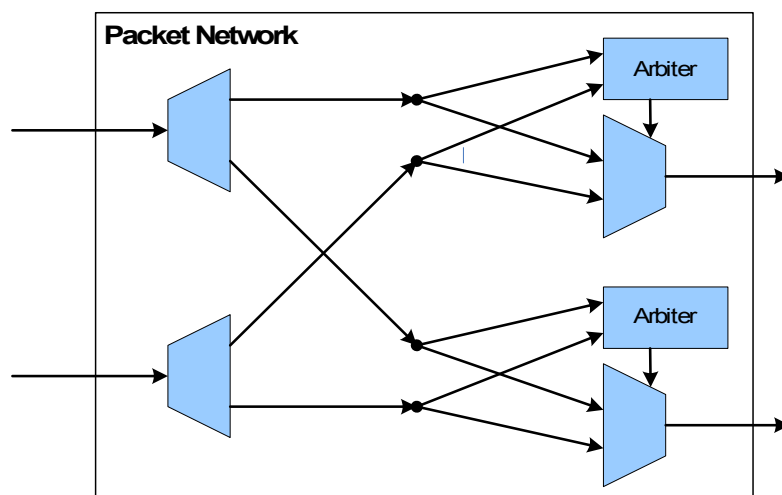
1. 当主机发出命令时，接收会话操作的第一个互联组件是转换器，如图 4 所示。转换器处理会话协议规范的大部分变量，例如，高电平有效和低电平有效信号选项以及可选的读流水线。


图 4. 主机网络接口



2. 代理组件是接收命令的下一模块。代理组件将会话操作封装为命令数据包，通过传送层将数据包发送至命令网络。代理组件还接收并向主机转发来自响应网络的响应数据包。
3. 然后，路由器确定数据包格式中的地址域，以及数据包的目的从机 ID，下一跳的路由信息等。
4. 限制器跟踪不同主机的主要会话操作，防止某一命令导致出现无序或者同时到达读响应。
5. 然后，将组件放到数据包网络中。Qsys NoC 网络最大限度的支持并行处理，只要两个主机不会同时访问相同的从机，所有主机和从机都可以在某一时钟周期中进行通信，如图 5 所示。

图 5. 提高了并行处理能力的数据包网络

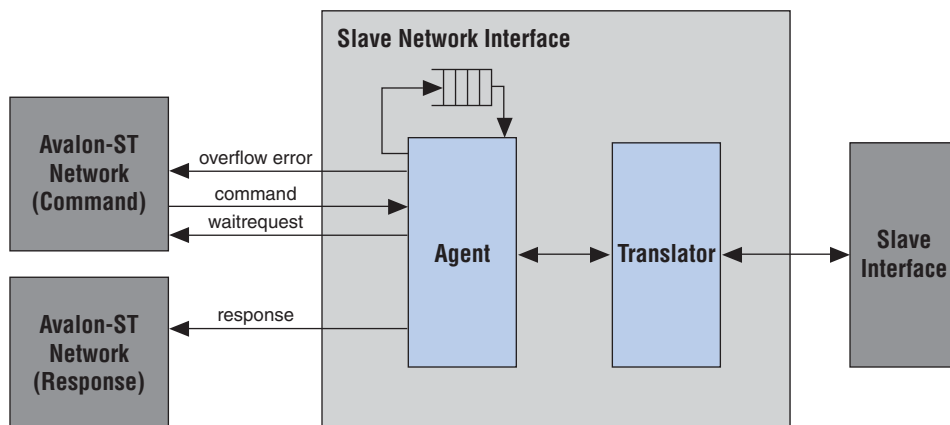


 注意，NoC 体系结构支持采用任何其他兼容网络来替代数据包网络。

6. 解复用器是数据包在传送层网络遇到的第一个组件。解复用器将数据包发送至下一从机。
7. 数据包到达分离器组件（以黑点表示），将数据包复制到仲裁器的输入，以及复用器的输入。
8. 系统设计人员如果需要专用仲裁，而不是 Qsys 默认提供的加权罗宾带仲裁，可以采用自己的仲裁器来替代 Qsys 仲裁器。为支持这一点，Qsys 仲裁器接收整个数据包，这样，替代仲裁器可以使用详细的会话操作信息，完成其仲裁判决，包括与数据相关的仲裁等。
9. 仲裁器的判决被发送至复用器，将所选择的数据包转发到从机网络接口，如图 6 所示。

10. 在从机网络接口中，数据包进入从机代理组件，在这里终结数据包，将所包含的会话操作转发至从机转换器。同时，从机代理组件将会话操作信息放入从机代理 FIFO 缓冲，完成需要响应的会话操作，例如，读操作和无标记写操作等。从机转换器的功能和主机转换器相同，处理接口规范中的所有变量。如果从机忙，不能接收更多的会话操作，那么，命令会反馈到代理组件入口。

图 6. 从机网络接口



11. 当从机响应读操作时，转换器将响应转发至从机代理组件。从机代理组件取出从机代理 FIFO 缓冲中的会话操作信息，例如，发起主机 ID，将该信息与会话操作响应合并，建立响应数据包。响应网络暂时不能接收响应数据包时，读数据 FIFO 可以存储响应数据包。
12. 然后，从机路由器检查数据包，以确定主机 ID，分配本地路由信息。
13. 响应与命令相同，只是反向而已。响应数据包通过解复用器，到达仲裁器，被选中后，通过复用器转发回限制器。然后，限制器记录接收到了这一响应，将其送回主机代理，最后以会话操作响应的方式到达主机。

除了所介绍的组件，Qsys 还根据需要增加了突发适配器和宽度适配器。这些都是数据包组件，检查数据包中数据的某些域，从而做出正确的自适应判断。当主机和从机位于不同的时钟域中时，Qsys 能够增加流水线级以帮助满足时序要求，自动增加握手或者双时钟 FIFO 组件。

性能实例

下面的例子对比了两个不同系统的性能：一个 16 主机 /16 从机系统，一个 4 主机 /16 从机突发和宽度自适应系统。这一对比描述了 Qsys NoC 互联与传统互联实现的频率、延时和资源应用之间的不同。在这些例子中，所有系统均在 Altera Stratix® IV 器件上实现，使用了 C2 速率等级器件。对于相同的系统，Qsys NoC 互联系统性能与传统的 Avalon-MM 互联相当，后者的开发采用了 Altera 前一代 SOPC Builder 工具。

16 主机 /16 从机系统

16 主机 /16 从机系统总共采用 256 个链接实现全部链接。简单主机和从机 IP 组件只用于测试互联的特性，意味着系统代表了完全同类的系统，而不是典型的嵌入式系统。

表 1、图 7 和图 8 显示了传统互联和 NoC 不同延时选择的频率和资源利用结果。

表 1. 16 主机 /16 从机系统：性能结果（相对于传统互联的百分比）

互联实现	f_{MAX} (MHz)	资源占用情况 (ALM)
传统互联	131	12766
Qsys NoC, 完全组合	161 (+23%)	13999 (+10%)
Qsys NoC, 1 周期网络延时	225 (+71%)	11260 (-12%)
Qsys NoC, 2 周期网络延时	243 (+85%)	12761 (+0%)
Qsys NoC, 3 周期网络延时	254 (+93%)	14206 (+11%)
Qsys NoC, 4 周期网络延时	314 (+138%)	26782 (+110%)

图 7. 16 主机 /16 从机系统：NoC 与传统互联的频率对比 (MHz)

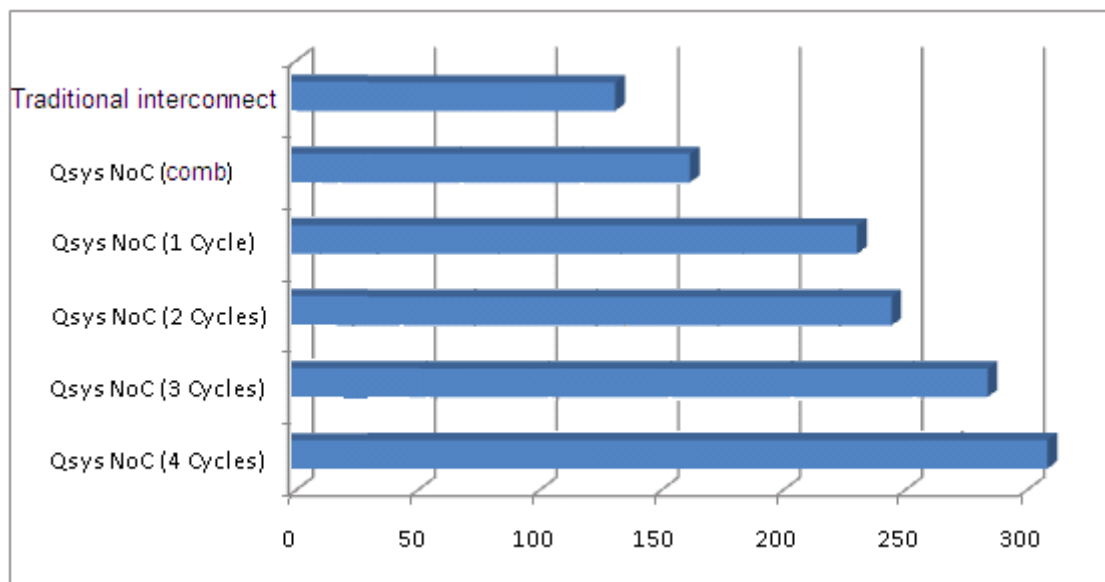
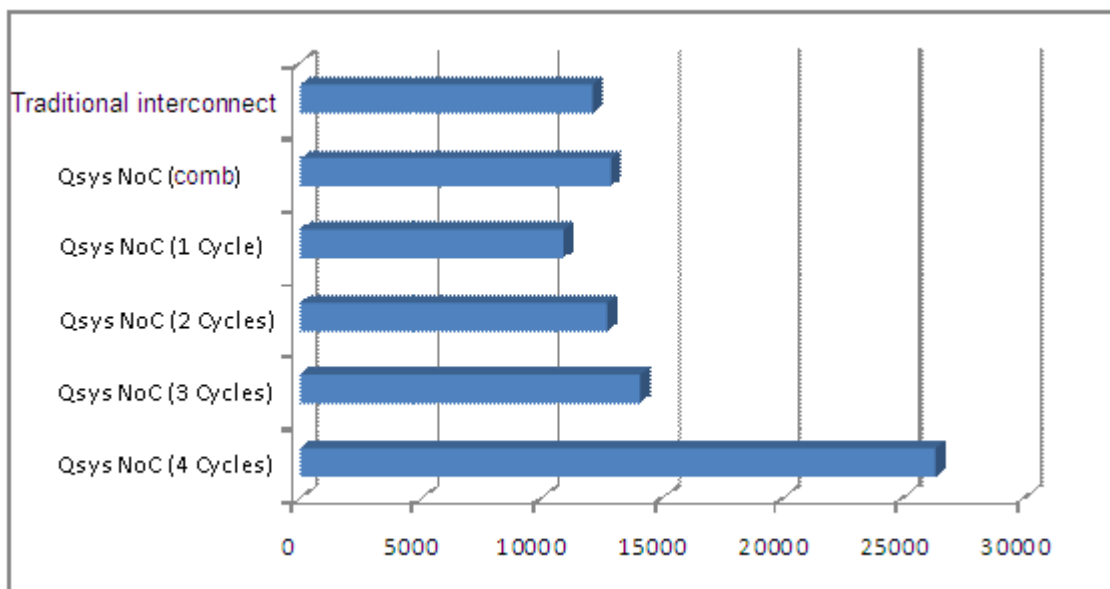


图 8.16 主机 /16 从机系统：NoC 与传统互联的资源占用情况对比 (ALUT)



4 主机 /16 从机突发以及宽度自适应系统

4 主机 /16 从机突发以及宽度自适应系统含有典型同类系统的功能，支持不同宽度和不同突发的主机和从机，需要在互联中采用突发自适应。表 2.4、图 9 和图 10 显示了传统互联和 NoC 不同延时选择的频率和资源利用结果。

表 2.4 主机 /16 从机系统：性能结果（相对于传统互联的百分比）

互联实现	f_{MAX} (MHz)	资源占用情况 (ALM)
传统互联	123	11658
Qsys NoC, 完全组合	125 (+2%)	9655 (-17%)
Qsys NoC, 1 周期网络延时	150 (+22%)	9423 (-19%)
Qsys NoC, 2 周期网络延时	164 (+33%)	9847 (-16%)
Qsys NoC, 3 周期网络延时	154 (+25%)	13156 (+13%)
Qsys NoC, 4 周期网络延时	171 (+39%)	16925 (+45%)

图 9.4 主机 /16 从机系统：与传统互联的频率对比 (MHz)

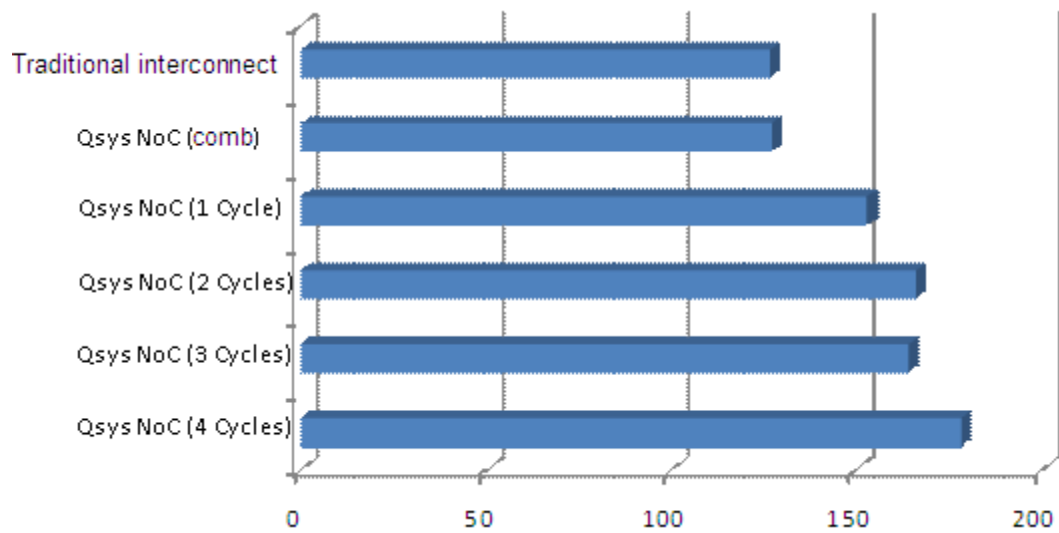
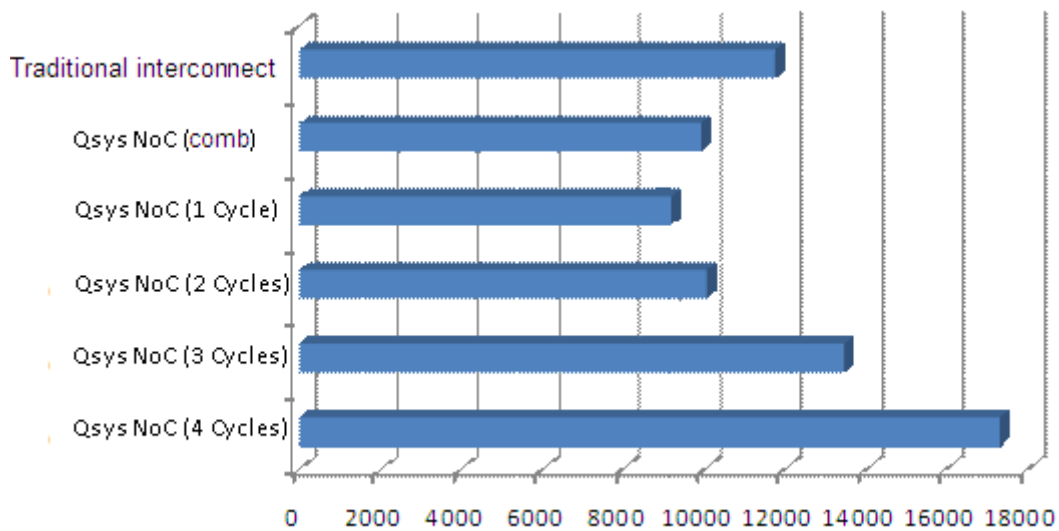


图 10.4 主机 /16 从机系统：与传统互联的资源占用情况对比 (ALUT)



结论

与传统的非 NoC 互联结构相比，NoC 互联体系结构有很多明显的优势，支持会话层和传送协议层的独立设计和优化。Qsys 系统集成工具根据应用需求，产生非常灵活的 FPGA 优化 NoC。与传统的互联结构相比，Qsys NoC 互联结构在相同的延时和资源特性条件下，进一步提高了工作频率， f_{MAX} 提高了 2 倍。

详细信息

- Altera 网站的 Qsys 软件支持页面：
<http://www.altera.com/support/software/system/qsys/sof-qsys-index.html>
- *Quartus II 手册* 第一卷的 *采用 Qsys 进行系统设计* 一章
http://www.altera.com/literature/hb/qts/qsys_section.pdf
- *AN632: SOPC Builder 到 Qsys 的移植指南*
<http://www.altera.com/literature/an/an632.pdf>
- *Qsys 系统设计教程*
http://www.altera.com/literature/tt/tt_qsys_intro.pdf

致谢

- Kent Orthner，软件和 IP 高级经理，Altera 公司。

文档修订历史

表 3 列出了本文档的修订历史。

表 3. 文档修订历史

日期	版本	进行的修改
2011 年 4 月	1.1	更新了性能和资源使用信息。
2011 年 1 月	1.0	初次发布。