

# 将 MOST 控制消息 用到极致

本案例研究将介绍 Xilinx LogiCORE MOST NIC 控制消息处理的设计

作者: **Stacey Secatch**

Xilinx 公司

资深设计工程师

stacey.secatch@xilinx.com

仅使用 2,600 个 slice 的 Xilinx® MOST (Media Oriented Systems Transport, 面向媒体系统传输) 网络接口控制器 (NIC) 大约占用了 Spartan™-3E 500 器件一半的空间, 为微处理器和其他外设留下了很大空间。此功能全面的嵌入式核将数据和网络控制作为主或从节点, 其访问十分便捷。

本文中, 我将提供一个控制消息处理设计案例研究, 说明在没有任何主机软件介入的情况下, MOST NIC 如何独立地处理消息。经过精心设计, 占用资源最少的 PicoBlaze™ 可配置状态机可正确有效地处理控制消息。

## MOST 网络中的控制消息

MOST 是一种面向信息娱乐的汽车网络标准, 采用环型拓扑, 其强势之一在于支持用于环中若干个节点的多个同步媒体流。然而, 像其他网络一样, 节点间不单发送数据, 还需要控制通信。图 1 所示为简单的 MOST 环网络, 其中有一个网络主节点和若干个从节点。它支持 6 种不同的控制消息类型。我们着重探讨一下 MOST 网络操作过程中最常用的三种类型。

### 正常消息

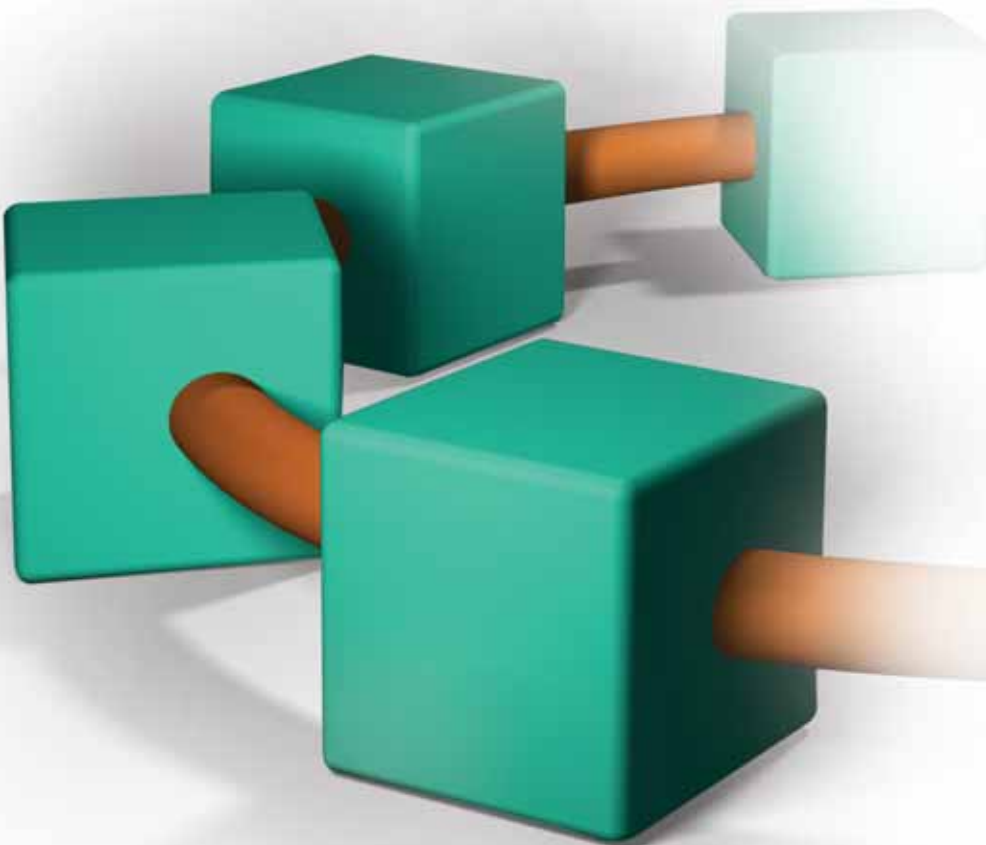
正常消息与其他类型的消息不同, 专用于主机处理器的高级应用。所有其他消

息都用于环操作和调试。例如, 作为环启动序列的一部分, 主节点会发送一个正常消息来识别从节点 B。如果信息多于一条消息的容量, 在应用层可进行正常消息组合。

### 通道分配和释放

分配消息被发送到主节点以获取对帧中留给媒体数据的部分之访问权。请求者

指明需要多少个字节宽度通道, 响应者 (主节点) 确认请求的状态 (准予或拒绝) 及哪些通道准予分配。通道释放消息与分配消息相似, 请求者指明要重新分配哪些通道, 主节点以准予的重新分配做出响应。



### 选择硬件/软件分区

嵌入式设计涉及到根据功能和资源选择硬件/软件接口之间的最佳边界。选择过多的硬件实现意味着高昂的造价，因为需要额外的资源来实现设计。选择过多的软件实现会导致微处理器超负荷和逻辑梗阻，如果处理器丢失事件的话，甚至可能导致设计完全失去功能。

对于 MOST LogiCORE™ 解决方案，我们找到了分割控制消息处理的清晰边界。

### 延迟时间需求

控制消息被分解为多个帧。当消息沿着环传输时，通信节点之间需要特定的信号交换，周转时间大约为一帧的时间（23 μs 左右）。消息-响应生成通常需要计算。例如，分配响应取决于确定是否有足够的空闲通道可用。

因为该服务要求很紧迫，最好在硬件上进行核内控制消息处理，而将外部处理器留作网络堆栈和面向媒体的用户应用之用。

### 留给软件的事项

虽然几乎所有类型的控制消息都完全可以由 LogiCORE 系统处理，但是，正常控制消息必须交给应用。因此，MOST NIC LogiCORE 解决方案提供中断，以显示一个正常消息已被接收，并可供处理。而且因为所有的消息都由应用发起，系统中置有若干个配置寄存器，利用它们，应用可显示有消息供发送，中断可提醒应用消息序列已完成。

### 选择实现

确定在硬件中处理控制消息后的下一步是尽可能选择最有效的设计。我们本来可以决定采用传统的以自定义状态机的方式实现最有效的设计。根据初步估计，状态机大小为 1,000 多个 slice，再加上一个 Block RAM 来包含所有消息缓冲器。请注意，控制消息仅使用 64 字节帧中的 2 个字节，而要处理这 2 个字节却会占用核的三分之一还要多。

最后我们弃此方案不用，而选择将 PicoBlaze 可配置状态机作为控制通道处理的主干，将核资源最小化。在我们的实

现中，核资源包括用于维持指令存储器的 2 个 Block RAM、用于 PicoBlaze 组件的约 100 个 slice、用于外设的 150 个 slice 和用于消息缓冲的原有 Block RAM。

如果选择 PicoBlaze 控制器，使用的资源已接近最少。我们所获得的好处您多半也会得到，即借助程式化编码消息处理的简便，使设计周期更短，编码更容易维护。

### 组织数据存储

在设计 RTL 状态机的时候决定采用哪种变量存储器相当简单。一般情况下，应将 RAM 用于较大的缓冲器，其他一切都留在寄存器中。使用 PicoBlaze 控制器时，需要做更多前期计划，以在将资源最小化的同时提供对您所有数据的访问。

### 状态机变量

要存储永远不会在 PicoBlaze 控制器以外使用的变量，64 个“空闲”的便笺式寄存器再理想不过。例如，存储 MOST 控制消息的即时校验和结果的累加器就只有状态机需要。

### 更大的缓冲器

Block RAM 是用于缓冲的理想位置。虽然 PicoBlaze 控制器只提供对 256 个位置的直接访问，但为所有的缓冲器使用一个 Block RAM 比例化分布式存储器更有效。因为 PicoBlaze 可配置状态机按顺序执行任务，所以每次需要访问的缓冲器不会超过一个，这就使组合所有存储器变得很容易。另一个额外好处是，Xilinx Block RAM 会自动转换数据宽度，使您可以从主机端以 32 位访问，而从 PicoBlaze 控制器端以 8 位访问，二者之间天衣无缝。

### 共享变量

有些配置和状态变量需要持续可用。您可以将这些变量映射到 PicoBlaze 控制器外部存储器空间的寄存器，供状态机访问，跟 MOST LogiCORE 控制处理模块中的存储器并无二致（图 2）。在 LogiCORE 系统中，PicoBlaze 控制器外部存储器端口被映射到一个共享 Block RAM，唯一不同是存储器范围末端为由存储器映射的寄

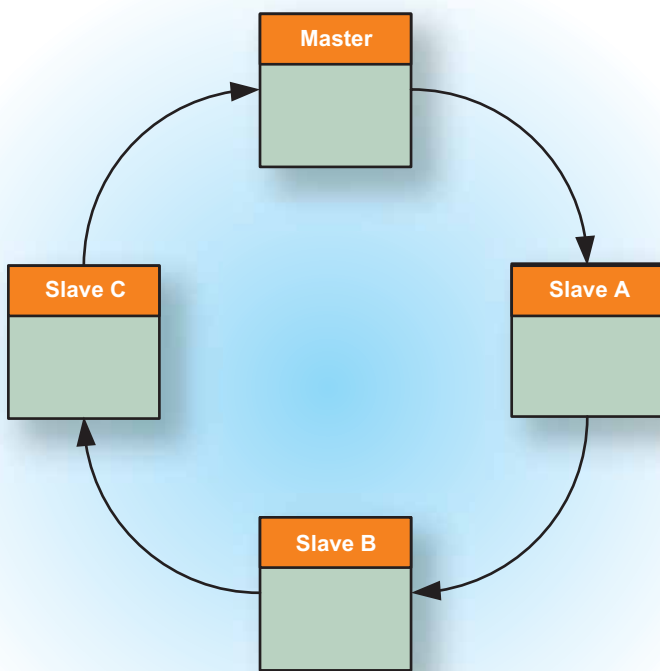


图 1 - 示例 MOST 环

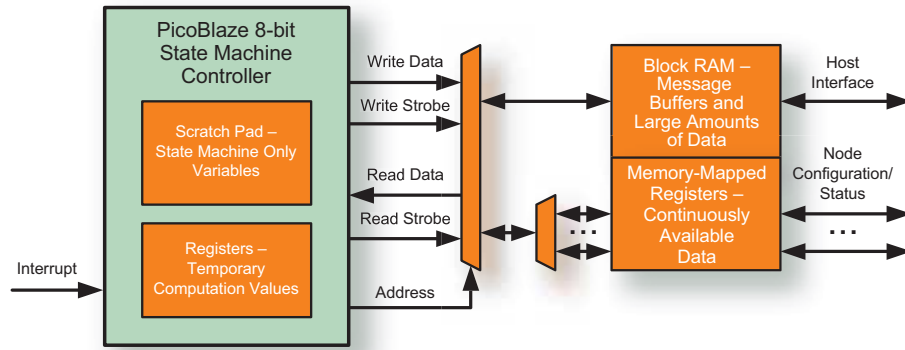


图 2 - 针对 Xilinx MOST NIC 控制消息处理的存储器组织

寄存器。

例如，MOST 节点上当前已编程的地址被映射到一个只读“寄存器”（多路器输入），以确定接收到的数据是否针对此节点。有关缓冲器当前占用情况的状态信息会放入一个只写寄存器，供外部主机访问。

### 排序处理

复杂状态机通常由较小的、并行操作的互锁状态机构建而成。PicoBlaze 可配置状态机按顺序执行任务，因此具有避免冲突和竞争条件方面的优势。但是，您还需要仔细计划，以确保事件调度准确无误，并且不会丢失排队等候的事件。以下指南可能对设计 PicoBlaze 应用有所帮助：

- 对事件处理分别轻重缓急。如果有不重要的事件，可选择放到最后执行。例如，缓冲器占用情况计数的值由 MOST 主机缓冲器读更改了。但此更新可能被延迟几帧。因为对消息的响应必须在一帧内完成，所以核优先对服务消息做出响应。
- 保持事件排序。如果处理要求完成一个事件后再执行另一事件，就需要

先执行首个事件的代码，即便排在后面的事件的时间要求更紧。例如，控制字节必须在当前帧中的字节尚未收到之前就发出。因此，对于给定的帧，发送事件总是要在接收事件之前处理。

- 考虑使用等待事件环。如果有一个很长的事件，无法在其他事件要求的时间帧内处理，可考虑将这一长事件放置于主处理等待事件环中。

例如，对一个供发送用的 MOST 控制消息的准备，可能需要多个帧才能完成。我们本来可以设置多个级别的中断服务，以确保在进行上述准备的同时还可以处理帧数据，但这一类型的任务非常适合等待事件环，该环会自动给予其最低优先权。

如果确实要选择在主环中处理数据，还需要格外小心，确保发出中断时不会损坏主环。在开始中断服务时，应复制所有可能被覆盖的寄存器，并在中断服务完成时恢复这些寄存器。执行代码的关键部分时，还可以使用 PicoBlaze 命令来禁用中断。

### 结论

Xilinx MOST NIC 已被证明可以有效地处理控制消息。由于能提供类似于流数据的效率，它堪称汽车信息娱乐系统的理想选择。对 Spartan-3A DSP 系列的器件支持即将推出，届时您的设计处理多媒体的效率定会更上层楼。

### 下一步（请点击下列资料了解详细内容）

- 在 MOST NIC 下载区了解有关采用 Xilinx MOST NIC LogiCORE 解决方案的嵌入式设计的详情。
- MOST NIC 用户指南提供了有关用于 MOST 链路层的所有 MOST 控制消息格式的详情，生成 LogiCORE 实现后，即可获得此应用指南。
- 下载 PicoBlaze 微控制器参考设计，包括汇编程序和扩展技术文档（如含有许多代码示例和完整指令集的技术文档的用户指南）等支持文件。