

运用 FPGA 进行控制平面 / 数据平面的视频处理

借以平衡高数据处理带宽和嵌入处理
控制的高清视频识别系统



作者: Glenn Steiner, 赛灵思
嵌入式处理解决方案高级经理

Dan Isaacs,
赛灵思技术营销总监

David Pellerin, Impulse
Accelerated Technologies
创始人兼首席技术顾问

界定系统的性能要求是嵌入式系统设计人员面临的巨大挑战之一。用以确定真实性能要求的所需的信息要么没有, 要么就难以获得。最精确的估算有时也会因意料之外的计算任务而失效。其他时候, 分析的结果可能会表明嵌入式处理系统对于数据处理需求而言成本效益太低。最终系统设计人员发现, 既具有高度的可扩展性, 即能够适应不断变动的性能需求, 同时又能够进行高性能数据处理的架构才是他们最需要的。在 FPGA 内部实施的控制平面/数据平面处理架构能够有效满足上述要求。

什么是控制平面/数据平面处理, 为什么您的下一代嵌入式系统可能会需要它?

在无法或者实际上不能用软件完成所有处理工作的系统中, 设计人员可以通过多种途径来获得其他性能。他们可以采用对称或者非对称处理配置的多处理器; 实施硬件协处理器; 或者将数据处理任务拆分给一个或多个专用处理元件——就像在控制平面/数据平面内进行处理一样。

在这种编程方式下, 数据处理被分成两个截然不同的面。控制平面代表着对性能影响不大的算法元素, 比如管理性任务、用户界面和操作系统功能。同时, 数据平面代表着数据在系统中的移动, 例如视频流或音频流及其处理。在数据平面上, 设计人员采用诸如流水线

这样的技术来增强数据吞吐能力。控制平面/数据平面处理的典型应用包括流视频、网络包处理以及高速信号处理。

让我们来近距离观察一个涉及流数据实时处理的控制平面/数据平面应用。我们准备挑战高清(HD) 视频流特有的模式识别问题。本例是涉及嵌入式微处理器, 要求混用多种高性能数据处理和控制功能的多种应用的典型代表。

HD 视频处理是一种可以通过把问题划分为控制平面和数据平面分别予以处理的常见现实应用要求。

720p、60Hz 的 HD 视频流的像素率为 74.25MHz。这就要求 222.75Mbps 的处理速率。假定采用运行在 2.5GHz 的双核双指令执行处理器来处理该数据, 最佳的指令率为每秒 100 亿条指令。这样的处理器可针对所处理数据的每一字节执行 22.4 条指令。对某些应用而言这已经足够了, 但 22.4 条指令表明所能处理的数据非常有限。复杂的视频处理功能, 比如内核卷积(kernel convolution)、噪音消减和其他过滤功能需要每秒执行的指令更多。该解决方案准备在数据平面上创建并行或者流水线处理等单元。

HD 视频处理是一种可以通过把问题划分为控制平面和数据平面分别予以处理的常见现实应用要求。作为一种高度并行处理的元件, FPGA 在本例中负责视频处理, 同时由 FPGA 内部的中等性能的处理单元负责视频处理流水线。该处理器可专用于单个应用, 也可以运行像 Linux 这样的操作系统。最终形成的硬软件混合实施方案可以把处理交付给能够进行最佳处理的部分, 实现低成本、高性能数据处理解决方案。图 1 显

示的是典型的控制平面/数据平面系统。

采用 FPGA 实现计算负载均衡

除了不菲的 ASIC, FPGA 是性能最高、最具经济效益的流数据处理单元实施方法。FPGA 因其灵活的架构而能让设计人员实施包含并行和流水线单元的处理系统。这样设计人员即可优化系统的性能和时延。

设计人员随后可以将该数据平面解

决方案应用于外部的分立微处理器以进行控制。在 FPGA 内部加入该处理器能够带来多项优势。内部处理器能够大大减少处理器和数据平面单元之间的控制时延。时延的减少可以释放出许多处理器周期。外部处理器必须与数据平面保持通信。通信通道可以是 32 位或者更多位数, 并同时需要更多导线用于寻址和控制。增加的导线可能会要求更强大的处理器和 FPGA 封装, 从而导致系统成本增加。而采用 PCI Express® (PCIe) 能大幅度减少引脚数量。遗憾的是, 不是所有的处理器和 FPGA 都支持这种相对新型的接口, 而且即便支持, PCIe 器件的成本也大大高于不采用 PCIe 的同类器件。

在 FPGA 内部实施控制平面处理器和数据平面可以减少器件数量、板级空间和功耗, 最终形成一个低成本的解决方案。在 FPGA 内既可以硬实施 PowerPC® 等处理器, 也可以软实施赛灵思 MicroBlaze™ 等处理器。此外还可以根据应用要求配置基于 FPGA 的处理器。基于 FPGA 的系统能够在处理器和 FPGA 逻辑之间移动决策和计算功能, 从而实现系统级的优化。

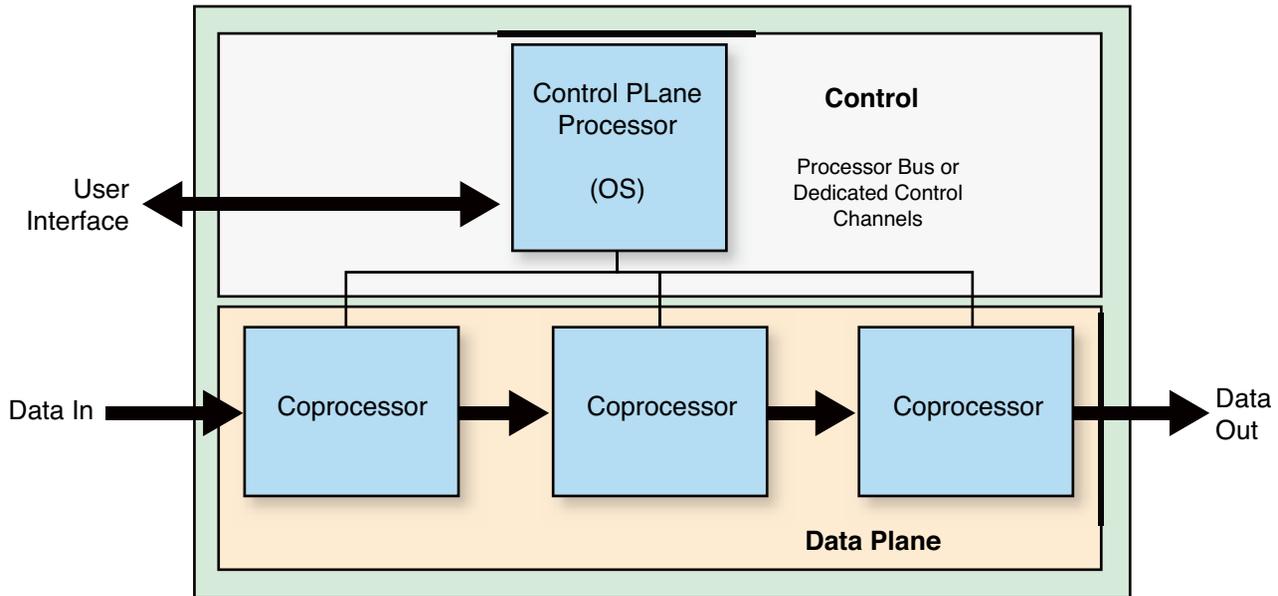


图 1——典型的控制平面 / 数据平面处理系统

控制平面 / 数据平面系统的实施

某些工具可以简化基于FPGA的控制平面 / 数据平面系统的实施。使用向导或者通过调整现有参考设计来编译系统是其中两种常用的方法。

通过向导可以采用FPGA工具迅速编译微处理器系统。使用下拉列表或者复选框，您可以轻松地指定目标部件以及需要的处理器和外设。与此类似，您也可以使用诸如MATLAB®软件这样的工具来迅速编译具有处理器总线接口的信号处理流水线以用于控制。另外，您还可以使用C-to-HDL工具构建数字信号流水线。控制平面 / 数据平面的连接可以简单地通过匹配总线接口来完成。图2显示的是启动向导的说明窗口和用向导构建的最终系统。

第二种方法是调整现有的参考设计。FPGA参考设计不断发展并已成为市场的焦点。在我们的案例研究中使用的参考设计具有完整的微处理器系统、存储器、外设和720p HD数字信号处理

流水线。因此，该系统相当于一款完整的控制平面 / 数据平面解决方案。在该参考设计中，由处理器控制增益和流水线中的FIR滤波器。使用C-to-FPGA工具创建目标探测和高亮显示模块，让整个系统具备功能所化的时间不到20个小时。

该处理器可以使用板支持包提供的补充驱动程序来控制数据流水线。目前已有面向Linux的驱动程序，可以让处理器直接控制数据处理流水线。Linux调用包括从Linux应用中打开I/O器件，然后针对该器件进行读出或者写入。

HD 视频识别系统案例研究

目标检测和识别广泛应用于监控、医学成像和工厂自动化等行业。图像的分辨率越高，目标识别的准确度越高。因此，对HD摄影机和相关的HD视频流处理功能的需求较为强劲。我们的案例研究从这个问题（受一部著名的动画电影启发）出发：我们能否检测并高亮显示720p HD视频流中的小丑鱼？

该设计需要16位的色谱，以便识别小丑鱼的条纹图案。一旦识别，小丑鱼将被打上移动的聚光灯并在屏幕上高亮显示。此外，聚光灯的大小经设计后可以随匹配的似然性而调整（实际上，系统降低了除小丑鱼周围聚光灯外的其他区域图像亮度）。聚光灯的尺寸变化和形状计算以及为在每个像素位置搜索克隆鱼所进行的比较，将需要大量的计算，对其执行需要74.25MHz的时钟周期。显然，这样的处理要求大大超出了一般嵌入式处理器的能力。

在这种情况下，最好的解决办法是把流数据处理转移到协处理器上。在FPGA内部实施协处理器能够以最低成本灵活构建能够满足性能要求的解决方案。因此，基于FPGA的控制平面 / 数据平面架构就是最佳的选择。FPGA嵌入式处理可以通过总线接口控制负责接收视频数据、探测小丑鱼、高亮显示小丑鱼并输出供显示之用的视频数据的数字信号处理流水线。

因此，在本目标检测和高亮显示案例中，我们选择了 50MHz 的 MicroBlaze 嵌入式处理器来管理和控制 74.25MHz 的数据处理流水线，同时管理用户界面。在免除了实际执行视频处理的责任后，处理器可以处理许多其他的功能，比如托管以太网数据通信、管理图形用户界面、对数据处理流水线进行精细控制（例如，逐帧增益控制）等。

诸如 Linux 这样的操作系统是提供多任务功能、网络协议栈和用户界面语言支持的理想选择。图 3 显示了所实施系统的方框图。该解决方案可以在高数据处理带宽需求和用软件控制数据处理方式之间实现理想的平衡。

采用 C-to-FPGA 工具实现软硬件协同设计

C-to-FPGA 编译器可以让开发人员使用新的开发工具集和新技术解决软件/硬件开发问题。开发人员可以先在软件中进行算法编码。经验告诉我们，在软件中开发算法较在硬件中开发算法

的效率更高。具体原因如下：首先，诸如 C 语言这样的软件语言能够让编程人员在更高的层面上开发算法，而这种层次是使用 Verilog 或者 VHDL 这样的硬件定义语言所谓福达到的；其次，与同类的硬件开发工具相比，针对 C 语言的调试和测试工具运行速度更快、效率更高，通常也更易于使用。相当于硬件算法，C 语言算法可以在目标处理器上全速运行，而硬件算法需要先在模拟程序上完成测试和调试；最后，C 语言开发工具较同类硬件开发工具的成本要低得多。因此，工程人员一般倾向于在 C 语言或者类似的高级语言中开发算法。

一旦使用 C 语言这样的软件语言完成对某个算法的验证，设计人员必须测量其性能，并确定该算法是否能够完全在嵌入式处理器或是完全在硬件上运行、或者硬软件混合协处理实施方案是否为最佳选择。在这种判断过程中可以采用性能分析工具。如果代码必须被转到硬件上，则设计人员必须手工转换算法，或者使用 C-to-FPGA 工具。

C-to-FPGA 工具可以让开发人员迅速把算法转为 HDL，对生成的硬件处理器进行优化，并在一些假设的场景下平衡性能和 FPGA 资源。该工具还能够让软件工程师使用 FPGA 内部的高性能数据处理逻辑，从而变身为硬件工程师。

使用 Linux 把处理器连接到 FPGA

与 FPGA 制造商合作的 Linux 供应商已经开发完成了能让处理器与 FPGA 通信并对其进行控制的驱动程序。首先，您必须针对该 I/O 器件配置 Linux。配置步骤由两个步骤组成。首先，把定制的驱动程序加载到 Linux 内核中：

```
module_init(xll_example_init);
```

随后，将驱动程序注册到特定的器件号码（比如 253）：

```
err = register_chrdev_region(devno, 1, 'custom_io_exmple');
bash# mknod /dev/custom_io_exmple0 c 253 0
```

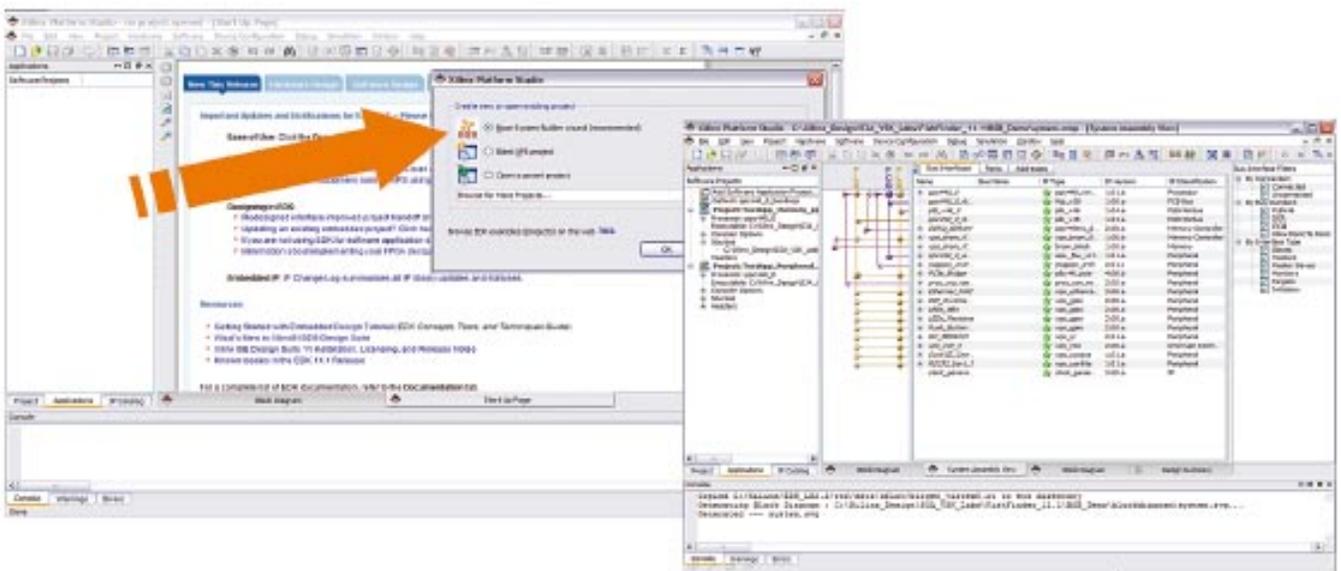


图 2——向导启动屏幕截图和已完成的系统

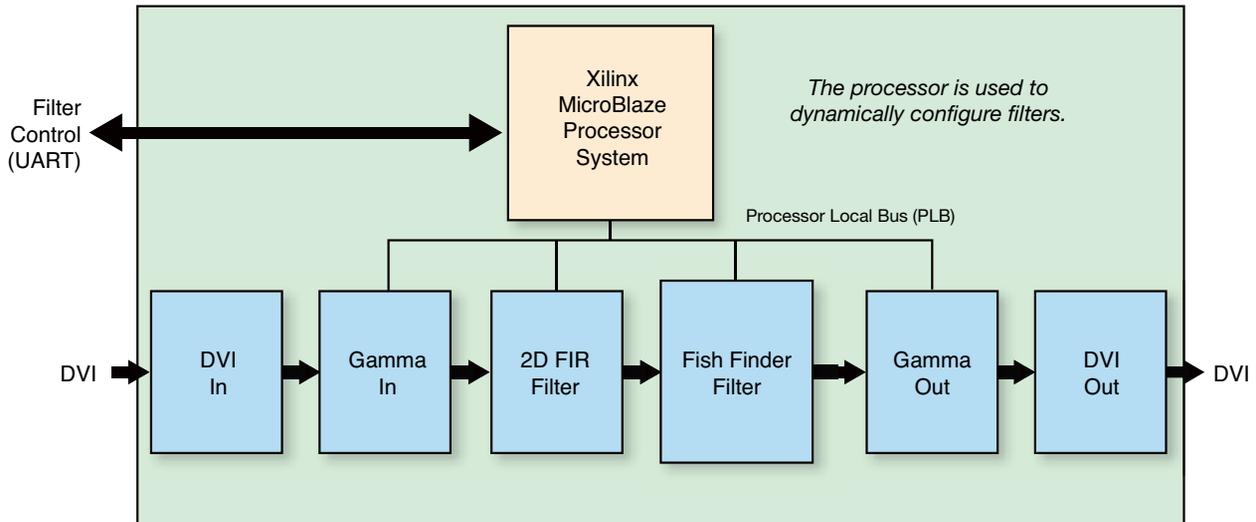


图 3——小丑鱼探测器控制平面/数据平面系统

通信是通过开启该I/O器件、然后对该器件进行读写而完成的，示例代码段如下所示：

```
// Open custom I/O device from
Linux application

int custom_io_ex_open(struct inode
*inode, struct file *filp);

// Read / Write to custom peripheral I/O using standard Linux
read/function function calls

ssize_t custom_io_ex_read(struct
file *filp, char __user *buf,
size_t count, loff_t *f_pos);

ssize_t custom_io_ex_write(struct
file *filp, const char __user *buf,
size_t count, loff_t *f_pos);
```

FPGA 的优势

信号处理器系统的数据带宽要求往往会超过通用处理器在经济适用的条件下能够提供的水平。在这种情况下，设计人员一般会把他们的数据处理系统划分为两个处理功能：使用通用处理器进行控制处理；使用诸如 FPGA 的硬件加速器进行数据处理。这样就构成了一个控制平面/数据平面处理系统。

FPGA 非常适用于同时实施控制平面和数据平面功能。一个 FPGA 可以包含一个或者多个像 MicroBlaze 这样软处理器，和/或像 PowerPC 这样的硬处理器。将它们集成到 FPGA 中可以实现控制平面处理器和数据平面处理系统之间的低时延、高带宽通信。

借助向导和预先构建的参考设计，针对嵌入式和数据处理功能的系统编译简单明了。通过把在 C 语言中构建的算法原型转换到高性能硬件处理单元，C-to-FPGA 工具有助于优化这一进程。最后，可以利用现在可用的 Linux 驱动程序方便地完成处理器和 FPGA 信号处理流水线之间的通信及控制编码。

我们的案例研究是一个典型的应用示例。在这个例子中，通过低成本通用处理器处理 HD 视频流是不合实际的，但是可以通过 FPGA 内部的信号处理流水线轻松加以解决。处理器随即被释放出来用于提供用户界面、网络和系统管理功能，并同时监测和控制信号处理流水线。

今天 超越昨天

创新 停滞

赛灵思全新7系列FPGA隆重发布
[Http://www.xilinx.com/cn/7](http://www.xilinx.com/cn/7)