

利用赛灵思 FPGA 平台构建汽车驾驶辅助系统算法

System Generator for DSP 是一款具有高抽象层的设计工具，为算法开发人员和系统架构师从 Simulink 算法参考模型过渡到 FPGA 硬件实施技术提供了一种高效的途径，且无需任何 HDL 编码工作。

作者：Daniele Bagni，赛灵思公司 DSP 专家

联系方式：daniele.bagni@xilinx.com

Roberto Marzotto，Embedded Vision Systems (eVS) S.r.l.

设计工程师

联系方式：

roberto.marzotto@evsys.net

Paul Zoratti，赛灵思公司汽车高级系统架构师

联系方式：paul.zoratti@xilinx.com

汽车驾驶辅助系统这一新兴市场需要高性能数字信号处理技术，同时还要降低器件成本，这样才能满足大规模应用的需求。赛灵思 FPGA 器件提供了适当的平台，能够满足上述这对互相矛盾的要求。即便是不熟悉 FPGA 实施方法的算法设计人员也不必担心从 PC 算法模型转向 FPGA 硬件原型设计时的复杂性。

赛灵思推出了一款工具 System Generator for DSP，为算法开发人员和系统架构师从 Simulink® PC 模型转向实时 FPGA 硬件实施技术提供了一种高效、直观的方法。这种具有高抽象层的设计工具在赛灵思和 eVS 工程师联合开展的设计项目中发挥了关键作用。该项目的目标就是利用 System Generator for DSP 推出一款适用于采用 Xilinx FPGA 的汽车车道偏离警告系统的图形处理算法，旨在提高整体性能，降低成本，并缩短开发时间。

驾驶辅助系统开发面临的挑战

汽车驾驶辅助 (DA) 系统工程师通常使用 PC 模型来创建复杂的处理算法，以便实现高度可靠的自适应巡航控制、车道偏离警告及行人检测等功能。开发人员高度重视 PC 算法模型，因为这种模型使他们能够尝试使用并快速评估不同的处理算法。不过，说到底，我

们还是需要一款合理设计的电子硬件解决方案，来实现经济有效的大规模生产与部署。

验证可部署目标硬件与软件算法模型之间的算法性能一致性，这对许多开发人员来说都是个问题。从浮点转到定点计算（比如，三角函数采用的不同方法）有时会导致参考软件算法和硬件实施模型之间出现明显差异。另外，输入图案资料集群 (input stimulus) 有着很大的不确定性，这对 DA 系统开发人员来说，使验证算法性能一致性工作变得更加复杂。

对通常依赖远程感应装置（摄像头、雷达等）的输入的 DA 系统来说，输入信息来就是驾驶员在实际行车中可能遇到的各种路况和环境条件。工程师会发现，设计一款充分满足所有情况需求的处理算法极具挑战性，而且验证软件模型与电子硬件实施之间的算法性能一致性至关重要。

对涉及图形处理的许多应用来说，赛灵思 Spartan®-3 FPGA 器件的并行资源相对于 VLIW DSP 平台而言性价比更高（参见我们发表于 Xcell Journal 第 63 期上的文章：http://www.xilinx.com/publications/xcellonline/xcell_63/xcell_63_block.pdf）。不过，一些系统设计人员仍然误以为 FPGA 编程的唯一方法就是采用 VHDL 这种硬件描述语言，而许多工程师并不了解这种语言。但事实上，FPGA 编程不一定非得采用硬件描述语言不可。我们的设计方法则采用了 Simulink 建模工具和赛灵思 System Generator for DSP FPGA 综合工具，不仅能轻松、有效地实施 FPGA 设计，而且还能采用软硬件协同仿真加速算法一致性测试。此外，设计人员使用 System Generator 也不一定非得熟悉 HDL 不可。

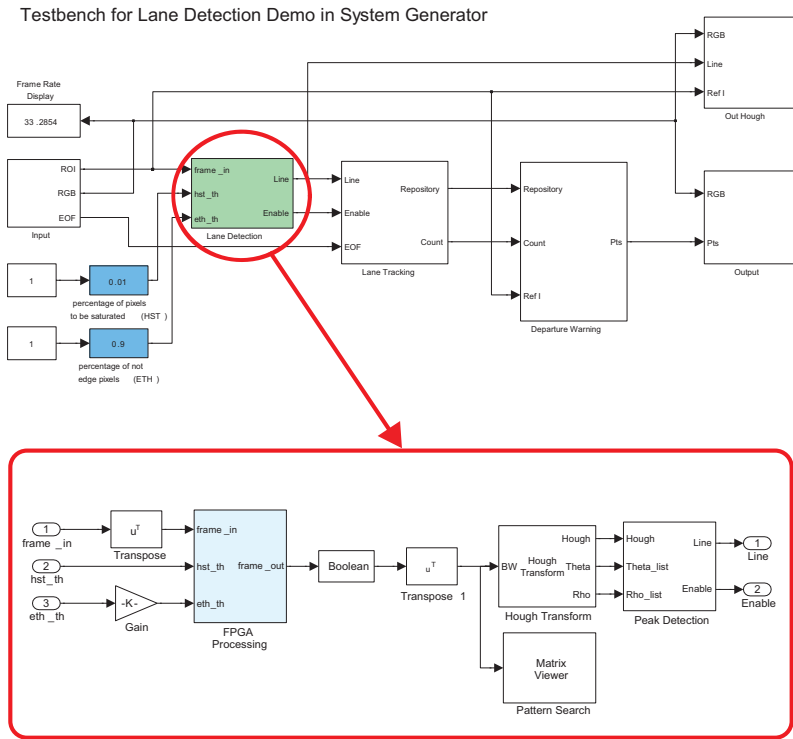


图 1——LDW Simulink 顶层方框图

车道偏离警告模型简介

车道偏离警告 (LDW) 系统的总体

功能就是在车辆无意中偏离正在行驶的车道时提醒驾驶员。安装该系统的车辆前安放摄像

头，可捕获到路况图形，以识别出车道边界标志。

车道偏离警告系统持续跟踪车道标线以及车辆相对于车道标线的位置。如果车辆越过了车道标线，那么系统就会发出警告。

汽车产业和学术界普遍采用 MATLAB® 和 Simulink 作为算法和系统级设计工具。尤其是 Simulink 具有高抽象层，而且提供图形化输入，因此能帮

助汽车算法工程师轻松快速地开发出复杂的 DSP 算法。

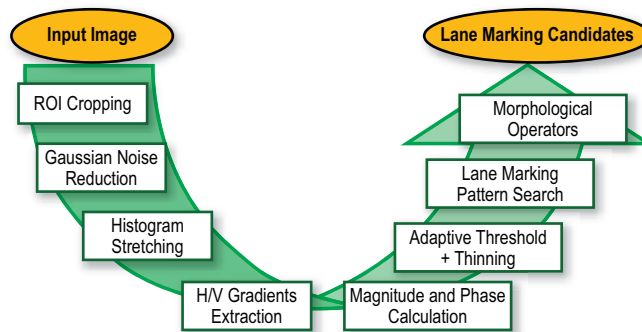


图 2——LDW 预处理功能链

图

1 显示了 LDW 系统模型的顶层方框图，该模型就是用 Simulink 设计而成。标记为车道检测的绿色块

包括图形预处理子系统，图 2 中将给出该子系统的各处理步骤。车道检测功能旨在提取出最有可能代表车道标线的路况图形。

System Generator 中车道检测演示测试基准

为了提高边缘设备噪声检测的性能，处理流程 (pipeline) 第一步为 2-D 5x5 高斯噪声抑制 (GNR)；第二步为

直方图拓宽 (HST), 开发人员可用该技术增强图形对比度, 尽可能利用整个灰度范围; 第三步则为水平/垂直梯度 (HVG), 可在本地强度大幅变化情况下增强像素。开发人员可通过计算图形的 2-D 5x5 梯度 (通过 2D 欧几里德距离), 来执行 HVG。

边缘细化 (ETH) 块可通过设定梯度幅值的阈值并应用非最大抑制生成细化轮廓线 (宽度为一个像素) 来确定哪些点是边缘。车道线模式搜索 (LMPS) 相当于一个滤波器, 选择显示与车道线一致的特定配置的边缘点子集, 并去除由于阴影、其它车辆、树木、路标等造成的虚假边缘点。处理流程的最后一步就是 3x3 形态滤波 (MRP), 系统采用这种过滤技术可获得车道线备用图的最终清样。

在 Simulink 模型中, 我们执行图形预处理子系统的各阶段任务, 该系统混合采用了 Simulink 模块集功能和 MATLAB 模块。由于 FPGA 能通过并行硬件路径处理大量数据, 因此非常适合实施本模型的车道检测功能。正由于此, 我们将这一功能作为 LDW Simulink 设计向 FPGA 设计转型的起点。

根据分区功能, FPGA 对每一帧执行高处理强度的像素级分析, 将数据从 10 位灰度图形简化为简单的二进制图形, 便于下游处理。对整个系统设计来说, 我们面向的是 XA Spartan-3A DSP 3400, 不过也能让系统采用较小的 3A DSP 1800 或 3E 1600。

System Generator 概述

System Generator for DSP 设计工具运行于 Simulink 中。它采用赛灵思面向 Simulink 的 DSP 模块集, 并将自动调用赛灵思 CORE Generator™ 工具生成高度优化的 DSP 构建块网表。我

们能通过 Simulink 库浏览器访问赛灵思 DSP 模块集。库浏览器可从标准的 MATLAB 工具栏中启动。构建 DSP 系统可用的 DSP 构建块有 90 多个, 此外还包括 FIR 滤波器、FET、FEC 内核、嵌入式处理内核、存储器、算术块、逻辑块以及按位块 (bit-wise block) 等。每个块都实现了周期精确和位精确, 我们可就延迟、面积与速度性能优化、I/O 端口数、量子化以及取整等对其逐一进行配置。

Gateway-In 与 Gateway-Out 这两个模块定义了 FPGA 系统与 Simulink 仿真模型的界线。Gateway-In 模块将浮点输入转化为定点数字, 此后该工具能正确控制定点分辨率的所有位增长, 具体取决于随后实用阶段中所实施的数学运算。

由于 Simulink 建立在 MATLAB 基础之上, 因此 System Generator 能使用完整的 MATLAB 语言用于输入信号生成及输出分析。我们可用 Simulink Source and Sink 库中的 From-Workspace 和 To-Workspace 模块来读取来自 MATLAB 变量 (From-Workspace) 的输入信号, 或者将信号的部分结果存储于 MATLAB 变量 (To-Workspace) 中。此外, 我们还能通过 MATLAB 变量设置 System Generator 的众多参数, 这样只需将包含所有这些变量的 MATLAB 脚本加以更新 (我们可将 MATLAB 函数分配给模型, 并在打开前, 甚至还能在仿真开始前或停止后调用函数), 就能实现设计方案的复杂定制。

System Generator for DSP 的另一重要特性就是软硬件协同仿真。我们能将设计的一部分综合到目标 FPGA 板 (硬件模型) 上, 让其余部分作为主机 PC 中的软件模型。这样, 我们就能逐渐从软件模型向硬件实施过渡。该工具

能通过以太网透明地创建和管理通信基础设施以及主机 PC 与目标 FPGA 器件之间的共用存储器。这样, 在运行仿真时, 硬件上实施的部分设计就能真正运行在目标硅器件上, 而软件模型则在主机 PC 上实现其余部分的仿真。我们可用共用存储器来存储输入图形和生成的输出图形等内容。以太网通信为伪实时处理提供了足够的带宽。您可在用户手册中了解到更详细的信息。

软件模型和硬件处理之间实现了灵活分区, 再加上软硬件协同仿真功能, 这就为您提供了一款功能强大的验证工具, 能验证原纯软件算法与用于生产的硬件实施模型之间是否一致。我们可用 Simulink 来比较软硬件处理的数据结果。该功能在驾驶辅助应用中特别有用, 因为一般的系统输入图形都是不确定的。

下面, 我们不妨来仔细探讨如何在 System Generator for DSP 中构建图形处理算法模型, 为了简单起见, 这里我们选择 GNR 为例, 这也是图形预处理流程的第一个模块。

System Generator 实施 GNR 功能

强度值 (即噪声) 的随机变化通常会损坏图像质量。这种变化表现为高斯或正态分布, 在不同传感器 (即 CMOS 摄像头) 中较为常见。线性平滑滤波器是消除高斯噪声的最佳方法, 在许多情况下, 它还可消除其它类型的噪声。为实现该功能, 我们可通过使用连续窗口中的像素加权和实施线性有限脉冲响应 (FIR) 滤波器。

在开始实施 GNR System Generator 模块之前, 我们已在 MATLAB 中实现了其行为模型。而这只需两行代码即可实现。首先, 我们需要计算内核, 具体描述掩模尺寸 (本例设为 5x5) 和高斯的 Σ 值。然后, 我

们可以通过卷积过滤输入图像：

```
n_mask = fspecial('gaussian', 5, 0.8);
out_img = conv2(in_img, n_mask, 'same');
```

此外，我们还可使用这种行为模型，并通过实际视频数据测试滤波器，进而调节掩模系数。我们还可以通过验证 System Generator for DSP 子系统的输出是否与 MATLAB 函数的输出相等（在规定的误差范围内，这是因为 MATLAB 是以浮点方式工作，而 System Generator 则以固点算法方式工作）来验证硬件。

2-D GNR 模块以流线方式（即逐行方式）对输入图像进行处理。图 3 显示了整个预处理链的 System Generator

顶级方框图以及高斯噪声抑制功能的顶级方框图。

图 3 中的 data_in 和 data_out 端口分别用于接收输入像素流和返回滤波后的像素流，而其余的端口则用于邻接模块间的定时同步与处理控制。

如图中黄色和蓝色模块所示，我们将 GNR 块的内部架构建立在两个主要子系统上。我们将对上述各模块进行深入探讨，详细了解 System Generator 设计。

我们需要第一个主要子系统，即线路缓冲器（如黄色所示），来缓冲四条输入图像流线路，以输出 5 x 5 排列的像素。对每个输入像素 $I(u,v)$ ，线路缓

冲器将返回由当前像素及同一行的前四个像素组成的 5x1 矢量，即 $[I(u,v-4); I(u,v-3); I(u,v-2); I(u,v-1); I(u,v)]$ 。如图 4 所示，我们可通过串联两个双线缓冲器来实施线路缓冲(line buffer)块，每个缓冲器使用一个双端口块 RAM（寄存于 FPGA 器件上的内存资源）、一个内存寻址计数器、一些简单二进制逻辑以及寄存器来实现适当的延迟。

蓝色阴影和赛灵思徽标表示 System Generator 原始块，分别对应精心优化的可同步 HDL 代码。通过这种图形界面，System Generator 使算法开发人员能够轻松地在硬件中实施 DSP 算法，且无需掌握任何 HDL 编码技术。

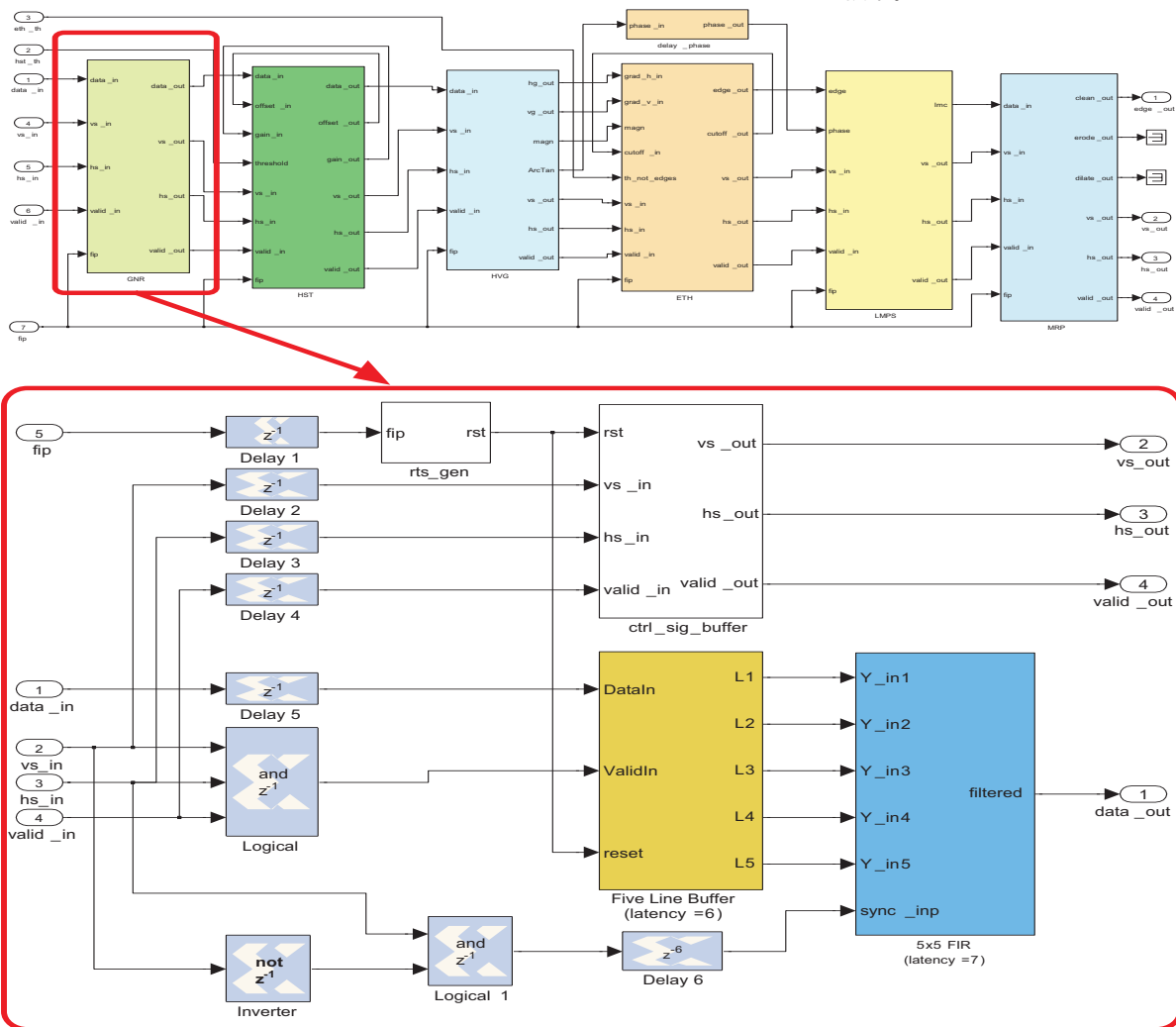


图 3——顶级预处理与高斯噪声抑制方框图

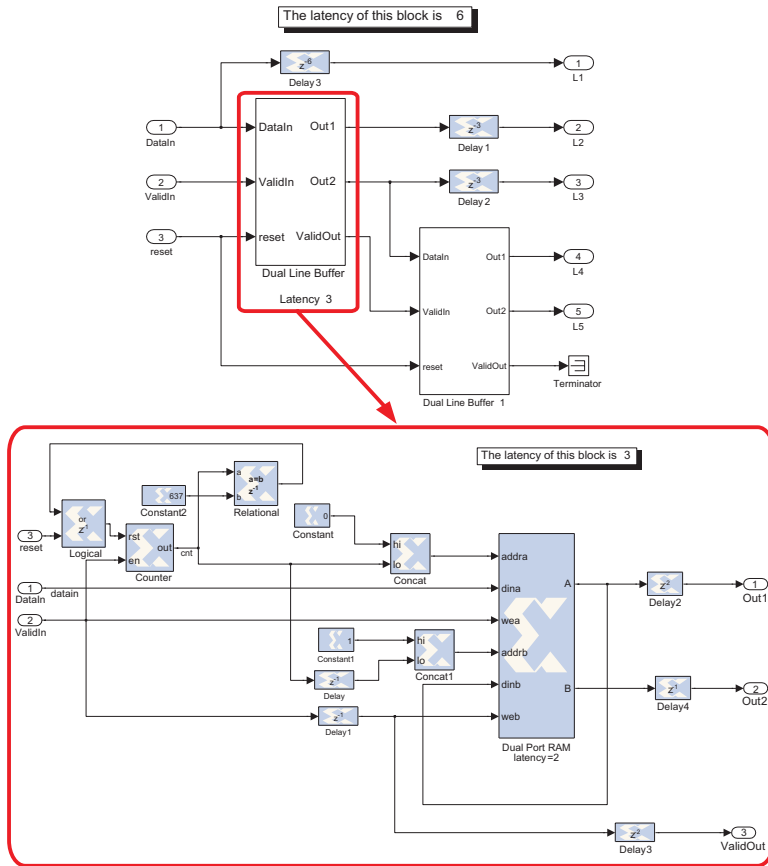


图4——System Generator 四线缓冲器的实施

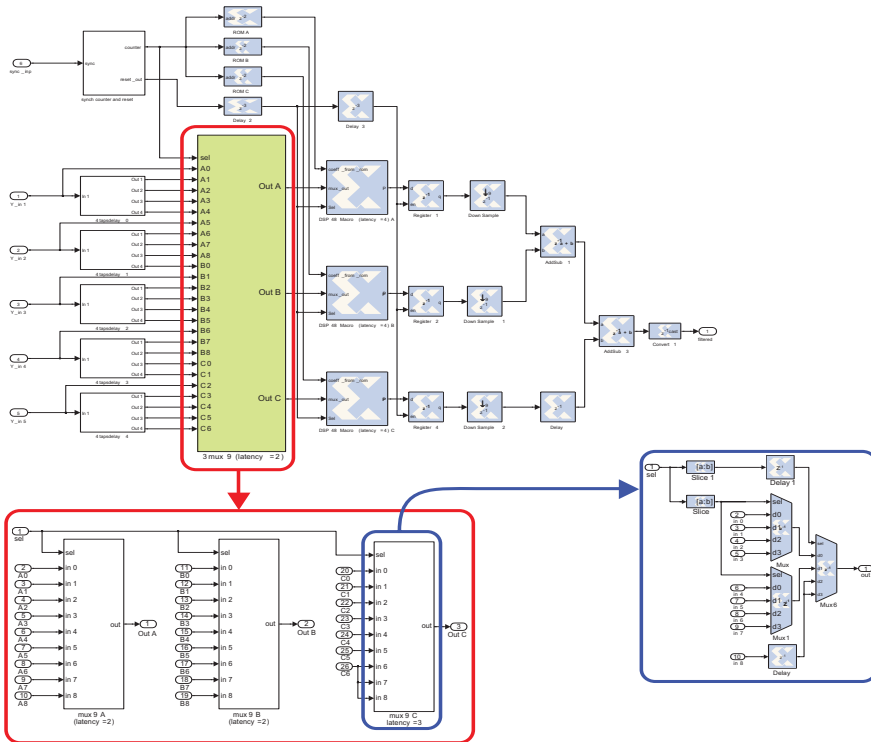


图5——System Generator FIR (卷积) 滤波器的实施

我们用第二个主要子系统（即 5x5 FIR 内核，如图 3 蓝色部分所示）来实施卷积运算。该子系统可从线路缓冲块接收一个像素为 25 的块输入，然后将它们乘以高斯掩模系数 25，最后将结果累加至寄存器。实施细节如图 5 所示。

为了符合目标器件的性能要求并实现每秒 900 万 (9 MSPS) 的采样率，我们选择同时使用三个乘法累加器（通过使用 FPGA 的 DSP48 可编程乘法累加器功能组件实施）。通过采用时分多路复用技术，每个累加器最多可实现 9 次乘法运算（请参阅图 5 顶级方框图中较大的 System Generator 块）。我们将一组系数为 25 的掩模分组并储存于三个 ROM 中，以便将这三个 ROM 作为分布式 RAM (FPGA 提供的另一内存资源) 实施。

在实施过程中，我们选择在编辑时期输入掩模系数。但是，我们本可以轻松地将设计进行升级，以将上述值接受为实时输入。采用这种方式，我们可以根据环境条件灵活地改变过滤的运行时间。

（实施说明：由于高斯分布具有各向同性的特点，内核可在两个分离的掩模中分解，过滤可作为 1x5 和 5x1 两个连续的沿水平和垂直方向的卷积来实现。虽然这种方法可减少所需的 FPGE 资源，但我们一般不用它，以避免失去设计的一般性与可读性。）

System Generator 的另一强大功能是能够在编程前运用预加载的 MATLAB 函数来定制设计。通过设置 System Generator 模块参数（比如，图像分辨率、FIR 内核值以及计算精度位数等）以及初始化我们在运行时间仿真过程中使用的工作空间信号，我们可以快捷地以不同处理方法和输入数据组进行实验。

而且，当我们完成仿真后，可调用 Stop MATLAB 函数，计算某些有用信息并将固点结果与浮点参照值进行比较，便可显示结果。这种方法使算法开发人员能够详细分析硬件实施的任何部分并将其与原软件模型相比较，以验证统一性。

System Generator FPGA 综合结果

开发人员在开发驾驶辅助系统时必须以适合大规模生产的成本水平进行设计。

达到一定处理性能所需的裸片资源将决定他们所需的 FPGA 器件的尺寸，进而决定其成本。

在实施车道偏离警告预处理器过程中，我们以 XA Spartan-3A DSP 3400 为目标，这是目前赛灵思汽车产品线中最大的器件。采用这种方法，并利用该模型来支持未来规划中的开发活动。但是，对预处理功能所占用资源的分析表明，该设计适合小得多的器件。

下表给出了 XA Spartan-3A DSP 3400 器件上 GNR 块的资源占用情况。计算时，假定在 VGA 分辨率下灰度输入图像的帧速率为 30-Hz（即输入数据

| | | | | |
|--------|-----|--------|-------|----|
| DSP4Bs | 3 | out of | 126 | 2% |
| BRAMs | 3 | out of | 126 | 2% |
| Slices | 525 | out of | 23872 | 1% |

速率为 9.2 MSPS)。

从定时性能角度看，GNR 设计以 168.32 MHz 的时钟频率运行，可接受数据速率高达 18.72 MSPS 的输入数据。

整个车道检测预处理子系统所需资

| | | | | |
|--------|------|--------|-------|-----|
| DSP4Bs | 12 | out of | 126 | 9% |
| BRAMs | 16 | out of | 126 | 12% |
| Slices | 2594 | out of | 23872 | 10% |

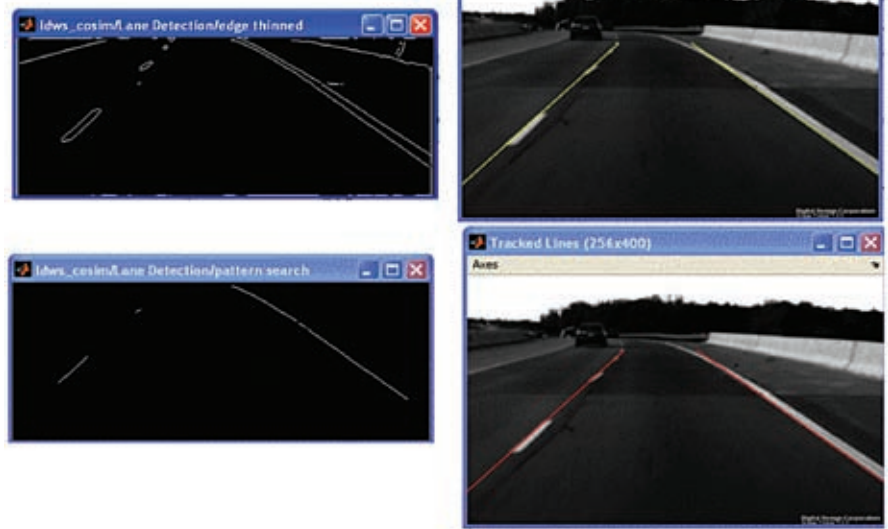


图6——LDW 处理模型输出

源总情况如下：

相应的定时性能分析表明，时钟频率为 128.24 MHz，最高输入数据速率为 14.2 MSPS。

根据上述所需资源情况分析，预处理功能甚至可用于 XA Spartan-3E 500，其密度大概为 XA Spartan-3A 3400A 器件的七分之一。

结果及未来工作

图 6 给出了一个 LDW 系统的性能图样，包括用于车道线备用物提取的基于 FPGA 的图像预处理功能。你可以看到，右边两个图片中的输入帧。左边的一对图像显示的是我们在 FPGA 中实施的预处理功能的性能。左上角的图片显示的则是阈值化后边缘检测函数的幅值。左下角图片是在边缘细化和车道线模式搜索处理后拍摄的。很明显，我们的 LDW 预处理器能非常有效地摄取道路图景，并能将数据减少至基本车道线选取物。右上角和右下角图片中的黄线和红线分别表示对基于简单直线道路模型的车道标线的瞬间跟踪计算结果。

为了能准确预测车辆相对于车道标线的轨道，我们未来的 LDW 系统将采用曲率模型。目前，我们在物体空间采用抛物线车道模型，假定车道宽度在近处保持不变并位于平坦地面上。我们可以通过创建包括位置、角度和曲率在内的四个参数，来描述抛物线车道模型。如果方法使用适当，可以计算视频序列每一帧的四个参数。

噪声、灯光的变化、摄像头抖动、车道线缺失以及沥青条纹都可能削弱模型提取。为了补偿这种信息缺口并使这一阶段更加稳定可靠，系统需要一个跟踪环节。可以通过使用车道模型参数空间中的 Kalman 滤波器进行跟踪。

车道模型的提取和跟踪是我们不久将在 FPGA 中实施的两个环节。为完成这一工作，我们计划使用 AcceIDSP™ 综合工具，这是赛灵思的另一个高级 DSP 设计工具。由于该工具支持线性代数信息库，我们可使用 AcceIDSP 实施四或六状态 Kalman 滤波器。

AcceIDSP 还可直接从 MATLAB 代码创建栅极，我们可以将其与 System

Generator for DSP 协同使用。另外，AccelDSP 综合工具非常适合可行性分析和快速原型设计。该工具可自动将原始浮点 MATLAB 转换为固点，并将各种 MATLAB 指令映射为 FPGA 资源。这很可能是赛灵思为驾驶辅助算法设计人员和系统架构师提供的最易于使用的 DSP 工具。

简而言之，致力于研究驾驶辅助技术的算法设计师和系统架构师如今可以借助一种功能强大的 DSP 设计工具来构建参考算法模型，然后轻松地在低成本的赛灵思 FPGA 器件中实施这些模型。最终将实现高质量、高性能、低成本的系统。

System Generator for DSP 工具的一个关键特性是能够在目标（通过以太网连接的专用线路板）硅器件中实施部分设计，同时将其余设计部分在主机 PC 上运行。通过这种软硬件共同仿真功能，在加速仿真过程的同时还能轻松验证硬件行为。

你可以看到，我们采用赛灵思的 System Generator for DSP 工具为 LDW 创建了一个图像预处理流程。

这里，我们仅探讨了其中一种模块（即 GNR 2-D FIR 滤波器）的一些细节。其实，整个车道检测预处理功能（见图 2）仅使用了 XA Spartan-3A DSP 3400 器件的 12 DSP48、16

BRAM 和 2,594 个切片 (slice)。而该器件运行速率为 128.24 MHz，输入数据速率为 14.2 MSPS，比 VGA 图像分辨率所需的高出 50%。整个算法设计和 FPGA 实施只需几个星期的时间即可完成，且无需编写任何 VHDL 代码。

我们期待继续开发这一项目：在 AccelDSP 设计工具中实施车道模型的提取和跟踪功能，然后将上述环节集成到 System Generator for DSP 模型。欲了解更多详情，敬请给我们任何一个人发电子邮件（本文作者非常感谢维罗纳大学 (Verona University) 计算机科学系的 Vittorio Murino 教授，感谢他的大力支持与卓越贡献）。

The widest selection
of COTS board-level solutions using Virtex-5 FPGAs

FPE650 - 6U VPX Quad
FPGA Processor Board

AD3000/AD1500 -
3.0GSPS or dual 1.5GSPS
A/D PMC/XMC

PMC-FPGA05 -
Virtex-5 LX110 PMC Board

HPE640 & HPE720
-6U VPX PowerPC & FPGA
Processor Boards

VPF2 - 6U VXS PowerPC &
FPGA Processor Board

MM-6171 - XMC Buffer
Memory Node

User programmable Xilinx®
Virtex®-5 FPGA signal
processors and analog,
digital and fiber-optic I/O
A new generation of performance

Single or multiple FPGA
solutions
Simple solutions for complex tasks

PMC, XMC, VXS and VPX
form factors
Flexibility based on open standards

Commercial and Rugged
variants
*Easily migrate from development to
deployed systems*

Libraries and Example Code
*Easy to use with head-start time-to-
market*

VIRTEX-5 **VPX** **VXS**

Embedded Computing - Data Recorders & Storage - Bus/Protocol Analyzers

For more information, please visit
<http://www.vmetro.com/virtex-5> or call (281) 584-0728

VIMETRO
Innovation deployed