

# 基于多核的车牌识别的架构实现

潘佳宾<sup>2</sup>, 胡越黎<sup>1,2</sup>, 陈晓君<sup>2</sup>, 毕卓<sup>1</sup>

(1. 上海大学 机电工程与自动化学院, 上海 200072; 2. 上海大学 微电子研究与开发中心, 上海 200072)

**摘要:** 多核技术是现在提高芯片性能的主要方法; 区别于传统以 PC 和 DSP 为核心的车牌识别系统, 以 FPGA 为核心, 利用 SOPC 技术构建了车牌识别多核处理器; 给出了一种基于多核的车牌识别架构, 在该多核处理器中, 以 3 个 Nios II 软核为主要处理器核处理车牌定位、字符特征识别提取及识别等处理, 同时构建硬件加速器作为协处理器处理图像增强、边缘检测和膨胀、腐蚀等数学形态学处理; 在 CQ 片上路由器基础上, 构建了 NOC 用以实现片上多核通信; 另外, 为了保证路由器与多处理器核之间的快速、并行通信, 加入了数据驱动模块; 整个系统在 Altera Cyclone IV FPGA 上实现了车牌的识别; 这种片上系统设计方法具有硬件设计灵活, 可扩展性强等优点, 能有效地降低系统软硬件设计的难度, 缩短开发周期, 并提高设计的可靠性。

**关键词:** 车牌识别; 数据驱动; 多核; NOC; FPGA

## An Implementation of Multi-core Architecture for License Plate Recognition

Pan Jiabin<sup>2</sup>, Hu Yueli<sup>1,2</sup>, Chen Xiaojun<sup>2</sup>, Bi Zhuo<sup>1</sup>

(1. School of Mechanical and Electronic Engineering and Automation, Shanghai University, Shanghai 200072, China

2. Research and Development Center of Microelectronics, Shanghai University, Shanghai 200072, China)

**Abstract:** Multi-core technology is the main method now to improve chip performance. Difference from the traditional LPR (license plate recognition) system based on PC and DSP, a multi-core processor is built using SOPC Builder to realize license plate recognition based on FPGA. In the multi-core processor, with three main Nios II soft cores to process license plate location, character feature extraction and recognition, we add hardware accelerators to process image strengthening, edge detection and mathematical morphology processing such as corrosion and expansion as coprocessors. On the basis of CQ router, NOC is structured to implement on-chip communication between multi cores. In addition, data-driven module is designed to ensure rapid and parallel communication between the router and multi-core processors. The entire system achieves license plate recognition on the Altera Cyclone IV FPGA. The on-chip system design features hardware design flexibility, scalability, etc., can effectively reduce the difficulty of system hardware and software design, shorten development cycles and improve design reliability.

**Keywords:** LPR; data-driven; multi-core; NOC; FPGA

## 0 引言

自动车牌识别系统是现代智能交通系统的重要组成部分之一<sup>[1]</sup>, 能够检测路面车辆, 并自动提取车辆牌照信息。车牌识别系统主要可以分为车牌定位和字符识别两个部分组成。其硬件基础, 可以大致分为图像采集设备(摄像头)、图像数据处理(处理器)、数据存储设备(各部分 RAM)和显示设备(显示器)等 4 个部分构成。传统的车辆牌照识别大多以 PC 平台或 DSP 处理器为核心来实现。由 PC 机扩展的图像采集系统体积大, 不能满足便携要求, 也不适合在露天环境下使用, 因此该方法主要用于前期算法的研究。而以 DSP 为核心的车牌识别系统外围电路设计复杂, 开发调试困难, 系统的可扩展性和可升级性效果较差。

可编程片上系统(system on programmable chip, SOPC)设计技术是现代计算机辅助技术、EDA 技术和大规模集成电

路技术高速发展的产物。这种技术增强了设计的灵活性和完整性, 缩短了前期开发周期, 提高了设计效率, 具有良好的可扩展性和可升级性。

自单核处理器的发展遇到主频瓶颈, 多核处理器已经成为如今提高处理器性能的主流, 多核处理器的架构引起了广泛关注<sup>[2-3]</sup>。同时, 随着片上系统(SOC)的发展, IP 核复用技术成为 SOC 系统开发的重要方式。而 IP 核复用技术在多核 SOC 系统中的应用成为多核处理器开发的一种主要方式<sup>[4-5]</sup>。

本系统采用 Altera 公司的 32 位嵌入式 Nios II 软核处理器, 基于 Altera Cyclone IV FPGA 结合图像处理和模式识别算法, 通过 IP 核复用技术, 采用 SOPC Builder 配置生成片上系统, 并将系统内嵌在 FPGA 内部, 灵活地执行数据的并行处理。

## 1 车牌识别主要模块硬件结构

车牌识别的主要流程如图 1 所示, 整个车牌识别流程主要可以分为 3 部分: 车牌图像预处理、车牌定位和字符特征提取及识别。

车牌图像预处理包括单通道提取和图像增强 2 部分, 通过对读取车牌图像的整体和局部的处理, 使图像中车牌数据更突出, 防止图像中其他物体对车牌数据的干扰, 有助于后期对车

收稿日期: 2014-06-10; 修回日期: 2014-07-14;

基金项目: 国家自然科学基金(61376028); 上海市科委专项基金(1311110460)。

作者简介: 潘佳宾(1989-), 男, 浙江湖州人, 硕士研究生, 主要从事汽车电子及多核微处理器方向的研究。

牌数据的处理。

车牌定位包含 3 个部分：边缘检测<sup>[6]</sup>、数学形态学处理（膨胀、腐蚀）<sup>[7-8]</sup>和车牌切割。其主要目的是在经图像预处理后的原始图像中确定牌照的具体位置，并将包含牌照字符的一块子图像从整个图像中分割出来，供字符识别子系统识别之用，分割的准确与否直接关系到整个牌照字符识别系统的识别率。

字符特征提取及识别则由字符切割和模板匹配两部分构成。在车牌区域切割出来之后，通过字符切割和模板匹配，完成最后的车牌识别。

在图 1 中，虚线部分为硬件加速器。硬件加速器是为处理大量数据的实时性应用或计算密集型的应用专门设计的硬件电路，以提高芯片的性能和处理能力：1) 处理器工作在较低的频率上，大幅度降低了系统的总功耗；2) 通过释放处理器，提升系统的整体性能；3) 降低对处理器的要求，降低系统的成本。

在车牌识别过程中，主要设计了 3 个硬件加速器：图像增强、边缘检测和数学形态学处理（膨胀、腐蚀）。

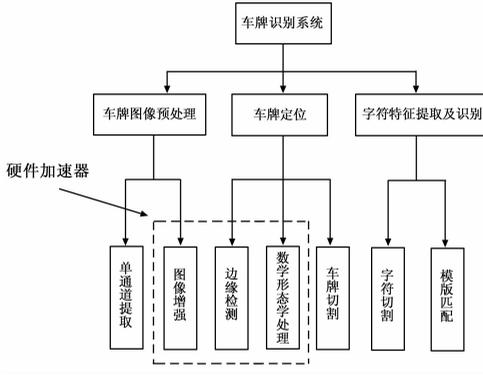


图 1 车牌识别流程

### 1.1 车牌图像预处理模块

通过摄像头传感器读取分辨率为 640 × 480 的某一静态汽车图像进行分析。图 1 中的单通道提取主要是指对车牌识别的 RGB 图片采用红色分量进行处理，主要基于两个方面的考虑：1) 车牌的图像为 RGB565 格式的图片，共 16 位，采用 5 位宽度的 R 分量数据，既可以减少图片数据量，又加快了图像的处理速度；2) 蓝色车牌由于是蓝底白字，蓝色在 R 分量中观察显示为黑色图像，与白色有更强的对比度。基于这个特点，采用前列减后列差分算法，可以去掉大部分亮度相当区域，保留出车牌这一区域。

图像增强是要改善图像的视觉效果，针对特定应用场合，有目的突出图像的整体或局部特性，扩大图像中不同物体特征之间的差别，使之改善图像质量，加强图像判读和识别效果。在车牌识别系统中，主要是为了提取像素变化，去除大部分亮度相当区域，保留车牌的轮廓。根据计算前列像素与后列像素差值的方法，并以硬件实现。

### 1.2 车牌定位模块

由于车牌图象在原始图象中是很有特征的一个子区域，确切说是水平度较高的横向近似的长方形，它在原始图象中的相对位置比较集中，而且其亮度值与周边区域有明显的不同，因而在其边缘形成了灰度突变的边界，这样就便于通过边缘检测

来对图象进行分割。

采用了 sobel 算子边缘检测的方法来提取车牌的位置<sup>[9]</sup>，在图像的任何一点使用此算子，将会产生对应的梯度矢量。图像的每个像素横向梯度  $G_x$  以及纵向梯度  $G_y$  计算公式为：

$$G_x = \begin{vmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{vmatrix} * f(x,y),$$

$$G_y = \begin{vmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{vmatrix} * f(x,y)$$

其中， $f(x,y)$  表示原图像的灰度分布图像，“\*”表示卷积。图像的每一个像素的横向及纵向梯度近似值来计算梯度  $G$  的大小，其计算公式：

$$G = \sqrt{G_x^2 + G_y^2}$$

在本文中，采用 CORDIC 计算 Sobel 算子的实时边沿检测电路。电路主要结构包括两个深度为 640 级的 FIFO 和 3 × 3 的移位寄存器，实现 Sobel 算子 3 × 3 窗口的卷积计算，6 级流水线 CORDIC 开方计算单元提高计算精度<sup>[10]</sup>。

图像二值化的硬件处理是将输入数据和阈值进行比较，比较结果作为选通器的使能信号。当计算完后输出图像只用两种颜色，白色的为数值 255，黑色的则为 0。整个边缘检测、二值化硬件结构如图 2 所示。

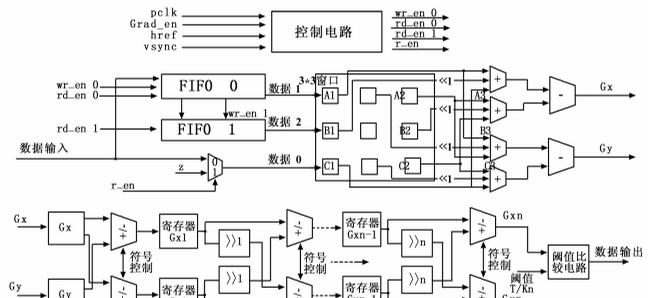


图 2 边缘检测和二值化硬件结构

在图中，pclk 为像素时钟，href 和 vsync 为 VGA 的水平同步信号和垂直同步信号，Grad\_en 为计算梯度信号，wr\_en、rd\_en 分别为写使能和读使能信号。每个时钟周期，数据会先缓存在深度为 640 的 FIFO 中。当两个 FIFO 都存满，下一个时钟周期到来，新输入的数据和两个 FIFO 内读出的数据会同时输入 3 × 3 的移位寄存器内，实现 Sobel 算子 3 × 3 窗口的卷积计算。在经过 3 个时钟周期 3 × 3 寄存器数据写入后，用寄存器内的数据按照 Sobel 算子的 3 × 3 窗口的分别相加，对于需要乘以 2 的数值按左移一位处理，相加得到两个值后，分别相减得到两个方向的梯度值  $G_x, G_y$ 。本文 CORDIC 算法结构采用 6 级流水线的结构，使用与向量旋转所需的总共迭代次数相同个数的迭代单元来并行处理，将前一级迭代单元的输出作为后一级迭代单元的输入。数据  $G_x, G_y$  经过 6 个时钟周期的流水线计算后，最终输出数据  $G_{16}$ 。将输出结果输入到阈值比较电路，与设定的阈值比较得出结果，完成二值化。

在车牌识别系统中，应用数学形态学膨胀和腐蚀两种运算。膨胀的作用是与物体接触的所有背景点合并到物体中，使目标增大，可填补目标中的空洞；腐蚀的作用是消除物体边界点，使目标缩小，可以消除小于结构元素的噪声点。

膨胀的过程: 用结构元素 B 扫描图像 A 的每一个像素; 对结构元素与其覆盖的二值图像做“与”操作; 如果都为 0, 结果图像的该像素为 0, 否则为 1。

腐蚀的过程: 用结构元素 B 扫描图像 A 的每一个像素; 对结构元素与其覆盖的二值图像做“与”操作; 如果都为 1, 结果图像的该像素为 1, 否则为 0。

膨胀和腐蚀的结构元素的大小为  $3 * n$ , 结构元素的大小直接影响膨胀腐蚀的效果。为了保留车牌区域, 除去其他不相关的区域, 对车牌的二值图进行多次膨胀与腐蚀。膨胀与腐蚀效果是相反的, 所以需要相同次数的膨胀与腐蚀。在本文中, 共采用 6 次膨胀腐蚀, 3 次膨胀与 3 次腐蚀相对应, 其中, 膨胀采用结构元素大小为  $3 * 16$ , 腐蚀采用结构元素大小为  $3 * 8$ , 其硬件结构<sup>[11]</sup>如图 3 所示。

在时钟控制下, 数据源首先将数据存入 FIFO 0, 然后 FIFO 0 存入 FIFO 1。数据源、FIFO 0 和 FIFO 1 同时给  $3 * n$  窗提供数据。在膨胀腐蚀运算中, 需要计算的像素点位于结构元素中心, 当数据开始存入 FIFO 0 时, 计算器开始计数, 每存入一个数据计算器加 1。当计数为  $(640 + n/2)$  时, 开始膨胀腐蚀运算。在  $3 * n$  窗中,  $B_0 \sim B_n$  为 FIFO 0 中的数据, 即为需要计算的像素点, 每过 1 个时钟, 数据右移一位, 实现中心点移动。通过控制器, 控制  $3 * n$  窗的大小, 膨胀运算为  $3 * 16$  窗, 腐蚀运算为  $3 * 8$  窗。将  $3 * n$  窗中数据取出, 作相与运算, 并将运算结果转换成 8 位数据存入存储器, 地址初始值为 19 000 每运算一次自减 1。当地址为 0 时, 运算结束。

车牌的定位主要是基于腐蚀膨胀后的二值图片计算水平投影的直方图, 然后根据车牌的固有特征如长宽比, 车牌的水平投影的亮度在一定的阈值内等找到车牌的区域, 然后对这一区域做垂直投影最后定位出车牌的位置。

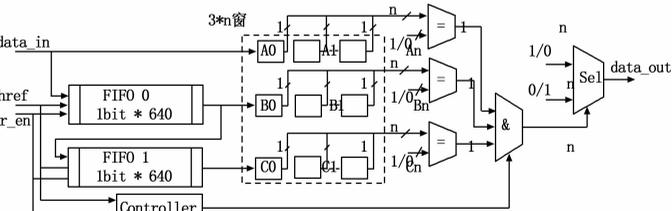


图 3 膨胀、腐蚀硬件结构

### 1.3 字符特征提取及识别模块

找到车牌位置坐标后, 在原车牌图像中取出车牌, 使用大津法 (OTSU) 对车牌进行二值化处理, 根据水平投影和垂直投影去除车牌的可能存在的边框和多余部分。由于车牌字符之间存在着空隙, 使用垂直投影的方法切割出字符。

目前用于车牌字符识别 (OCR) 中的算法有基于模板匹配的 OCR 算法、特征分析匹配法和神经网络识别法。在本文的车牌识别过程中, 采用模板匹配算法, 其基本过程是: 首先对待识别字符将其尺寸大小缩放为字符数据库中模板的大小, 然后与所有的模板进行匹配, 最后选最佳匹配作为结果。所有汉字和字符的模版大小均为  $40 * 20$  的二值矩阵数据, 在膨胀腐蚀后, 将截取的车牌区域缩放到大小为  $40 * [20 * 7]$  的二值矩阵数据, 将其切割后与模板相匹配。

## 2 数据驱动硬件多核结构

### 2.1 数据驱动模块

数据驱动计算机<sup>[12-13]</sup>相比于控制驱动计算机, 有着更好

的异步性、并行性、确定性、函数性等特点。它以“令牌”方式进行数据传输, 一旦到达的数据符合匹配机制的要求, 立即驱动 CPU 执行。数据驱动模块结构如图 4 所示, 由输入缓冲队列、参数存储模块、函数队列以及数据封包产生器组成。它的作用如下:

- 1) 接受来自路由器的数据, 并把它存储到相应的地址, 同时将对应标签置为有效;
- 2) 把标签同预先存储的函数所需数据表做比对, 对于所需数据全部有效的函数, 生成正确的函数标号, 供系统服务程序查询, 并调用相应函数;
- 3) 一旦该数据被使用过, 就将其标志置为无效。数据驱动器接收到一个数据封包后将其存储在输入队列中, 然后通过直接匹配机制驱动处理器执行存储在程序存储器的函数。

通信队列用于缓存来自于片内路由器的数据包; 数据包处理模块用于控制通信队列接收数据包, 并提取数据包中的函数号、数据号与数据; 参数存储模块可根据数据包的函数号和数据号生成相应的地址并在对应单元中存储数据; 标签更新模块若检测到某个函数号的 2 个数据号都“到达”, 即将该函数的“数据完备”, 则将函数号送至函数队列。函数队列用于缓存函数号, 等待处理器提取。执行时, 处理器根据函数号从参数存储模块提取与该函数号关联的有效数据。

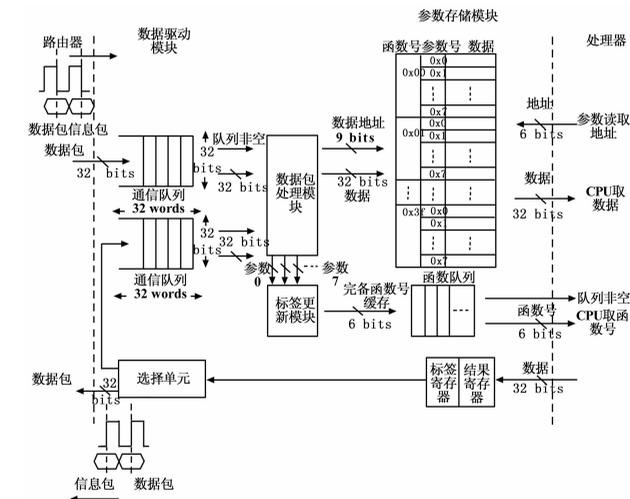


图 4 数据驱动硬件结构

### 2.2 多核硬件结构

多核处理器设计的核心主要集中在处理器核间通信问题上。比较主流的片上高效通信机制分为二种: 基于共享 Cache 的总线或交叉开关结构和基于片上网络 (NOC, Network on-chip) 的互连结构。而相比于总线和交叉开关来说, NOC 结构更适用于并行计算, 具有更好的可扩展性和更强的并行通信能力<sup>[14]</sup>。因此, 在本文的多核结构的设计中, 采用 NOC 作为多核处理器的通信互连模式。其中, 片上路由器是 NOC 的最关键部分。Kanizo 等人在总结前人经验的基础上提出了 CQ (Crosspoint Queued) 路由器<sup>[15-16]</sup>, 无论是时延、功耗、面积都取得了更加优越的性能。在本文的多核结构中, 通过 CQ 路由器构成 NOC, 从而实现片上多核通信, 其硬件结构如图 5 所示。

表 1 硬件资源消耗

模块	资源	计算单元		模块存储器	BUFFER
		组合逻辑	寄存器		
CQ 路由器		5735 (5%)	12720 (11%)	32768 (0.8%)	9152 (0.2%)
数据驱动		2274 (2%)	3390 (3%)	/	1824 (0.05%)
处理器	主核部分	7354 (6%)	4562 (5%)	1741696 (44%)	1039 (0.03%)
	从核部分	16230 (14%)	12057 (10%)	1360896 (34%)	4143 (0.1%)
SOPC		7535 (7%)	4692 (4%)	1408192 (35%)	/
系统总计		28844 (25%)	22005 (19%)	3133376 (79%)	8215 (0.2%)

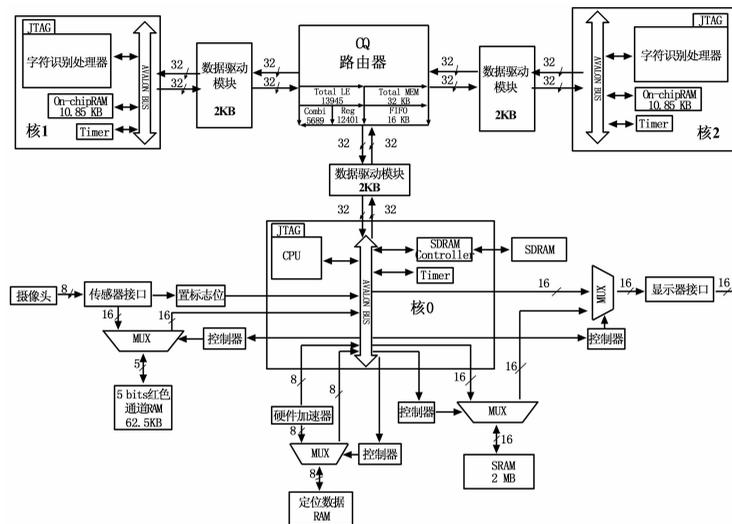


图 5 多核车牌识别总体硬件结构

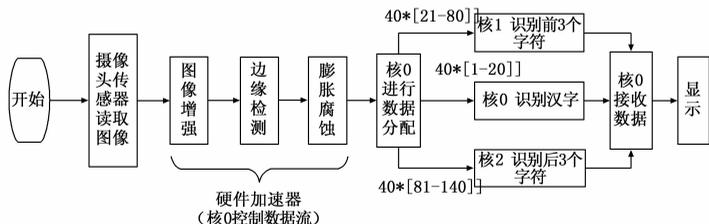


图 6 多核处理器车牌识别流程



图 7 车牌识别实测效果图

该结构包括 3 个软核。其中主核（图 5 中核 0）部分包括频率为 100 MHz 的 Altera NIOS II 软核，通过 JTAG UART 接口实现 PC 与 Nios II 系统间的通信。定时器核 Timer 用于记录程序运行时间。SDRAM 和 SRAM 用于数据的存储。同时，在主核部分，加入了硬件加速器，通过 AVALON 总线与 NIOS 软核相连，作为主核的协处理器，完成车牌识别的图像增强、边缘检测和数学形态学 3 个环节的处理。用硬件来实现

复杂耗时的运算，可以取得比软件计算更好的性能。

从核部分包括 2 个 100 MHz NIOS II 软核，每个核都有各自的 on-chip RAM、Timer 和 JATG UART。数据驱动模块实现数据驱动功能，根据片上路路由器的通信协议，接收和发送 CPU 数据。由于在车牌数据中，第一位为汉字，后六位为英文字母和数字的字符组合。因此，在本设计中，主核用以对汉字进行识别，而将 2 个从核（图 5 中核 1，核 2）设为字符识别，用以处理英文字母和数字，其中核 1 识别前 3 个字符，核 2 识别后 3 个字符。在 2 个从核的片上 RAM 中存储用于识别的字符模版。车牌数据经过主核部分处理后为二值矩阵数据，通过总线传递到数据驱动模块，按照路由器传输协议封装打包数据，经由 CQ 路由器，传送到从核。同样，从核在识别完成后，将识别结果通过总线，数据驱动模块和路由器将数据传回主核。

### 3 实验结果

本文设计的多核结构是在 FPGA 上实现的，实验平台是 Altera Cyclone IV DE2-115 的 FPGA 板。图像采集是用 CMOS 图像传感器 OV9650，通过 IIC 控制寄存器参数配置输出分辨率为 640 × 480 的图像。通过 Quartus II 11.0 和 NIOS II 11.0 软件建立 SOPC 系统，对 CMOS 采集的数据进行边缘检测处理的结果存储在双口 RAM 内然后通过 SDRAM 将 VGA 控制器显示在屏幕上，主要硬件资源消耗如表 1 所示。

图 6 所示整个车牌识别流程是摄像头 CMOS 图像传感器读取图像数据，通过单通道提取模块，取出红色通道数据。主核处理器控制数据流，将图片流数据分别通过图像增强、边缘检测和膨胀、腐蚀 3 个硬件加速器，实现二值化、车牌截取，获得车牌区域。由主核对车牌区域进行分割，同时，按照主核处理汉字，从核 1、2 处理字符的原则，根据路由器通信协议和对应核所需识别的数据大小，主核无需进行数据传输，从核 1 和从核 2 按照路由器协议，将车牌数据分割后传输。从核完成对接收字符数据的匹配，并将匹配结果返回主核。主核获得匹配结果通过显示器显示。实验结果如图 7 所示。

### 4 总结

本文给出了一种基于 SOPC 技术车牌识别系统设计方案，采用 ALTERA 公司 32 位 Nios II 软核处理器，通过 IP 核

复用技术构建车牌识别多核处理器。这种片上系统设计方法具有硬件设计灵活, 可扩展性强等优点, 能有效地降低系统软硬件设计的难度, 缩短开发周期, 并提高设计的可靠性。另外, 在本文设计的多核处理器中, 利用 CQ 路由器构建了 NOC 作为片上通信机制, 具有更好的可扩展性和更强的并行计算能力。数据驱动作为区别于传统冯诺依曼控制驱动的另一计算机体系能更加快速的进行数据传输, 提高处理效率。通过本文构建的数据驱动多核处理器, 可以有效地实现车牌的文字识别。

#### 参考文献:

- [1] 刘勇, 吴勇, 周芳. 对我国智能交通系统 (ITS) 发展的探讨 [J]. 交通与安全, 2006, 49 (1): 48-50.
- [2] 胡越黎, 朱卫, 丁倩: 基于多处理器系统的超级指令任务调用实现 [J]. 计算机测量与控制, 2009, 17 (12): 2541-2543.
- [3] 向慧芳, 胡越黎. 基于 SDZX-MV-02 MCU 核的多处理器架构设计 [J]. 计算机测量与控制, 2006, 14 (7): 942-945.
- [4] 胡越黎, 周谔: 基于 IP 复用的片上系统的构建与验证 [J]. 计算机测量与控制, 2010, 18 (3): 629-631.
- [5] 黄俊凉, 胡越黎, 王昆: MV10 微处理器与 APB 总线的接口设计 [J]. 计算机测量与控制, 2011, 19 (12): 3089-3091.
- [6] Sanka M, Hlavac V, Boyle R. Image processing, Analysis, and Machine Vision [M]. PWS Pub. 1999: 52-78.
- [7] Zhao Y Q, Gui W H, Chen Z C. Edge detection based on multi-structure elements morphology [A]. IEEE Proceedings of 6th World Congress of Intelligent Control and Automation [C]. Dalian; 2006: 9795-9798.

(上接第 212 页)

### 3 试验系统及试验结果

试验系统由测量通信系统模拟器、星载测量通信系统以及地面测试系统构成。如图 4 所示, 模拟器与测量通信系统之间构成互联互通的双方, 而地面测试系统用于实现时间同步与试验任务规划。试验过程中, 模拟器时频单元以及星载测量通信系统通过地面测试系统实现时间同步, 而管理控制计算机根据地面测试系统统一下发的任务规划信息, 控制模拟器在规定时间内开展特定场景的互联互通试验。

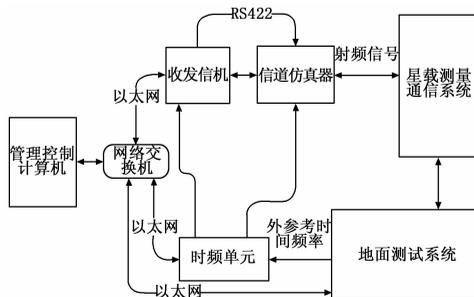


图 4 互联互通试验系统

2014 年 3 月, 利用该试验系统完成了模拟空间环境下共计 36 个场景的互联互通试验, 星间互联互通性能得到了充分测试。试验过程中模拟器状态良好、性能稳定并且易于操作, 试验结果验证了星间互联互通方案的有效性, 为评估星间在轨测量通信性能提供了有效支撑。

- [8] Pedrino E C, Saito J H, Senger H, et al. Color Mathematical Morphology In A FPGA [A]. 17th International Conference on Systems, Signals and Image Processing [C]. 2010.
- [9] Vincent O R, Folorunso O. A Descriptive Algorithm for Sobel Image Edge Detection [A]. Proceedings of Informing Science & IT Education Conference [C]. 2009: 97-107.
- [10] Meher P K, Valls J, Juang T B, et al. 50 Years of CORDIC: Algorithms, Architectures, and Applications [J]. Circuits and Systems I: Regular Papers, IEEE. VOL. 9 pp. 1893-1907. 2009.
- [11] Zhai X, Bensaali F, Ramalingam S. Real-Time License Plate Localisation on FPGA [A]. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops [C]. June 2011: 14-19.
- [12] Hurson A R, Kavi K M. Dataflow Computers; Their History and Future [J]. Wiley Encyclopedia of Computer Science and Engineering. 2008.
- [13] VoI H T, Osmari D K, Summa B, et al. Streaming-Enabled Parallel Dataflow Architecture for Multicore Systems [J]. Computer Graphics Forum. June, 2010, Volume 29, Issue 3, pp: 1073-1082.
- [14] Dally W J, Towles B, Packets R. Not Wires: On-Chip Interconnection Networks [A]. Design Automation Conference [C]. 2001: 684-689.
- [15] Kanizo Y, Hay D, Keslassy I. The Crosspoint-Queued Switch [A]. INFOCOM 2009 [C]. IEEE. 2009: 729-737.
- [16] Wang W F Li, J S, Lu C S. Design of a Crosspoint Queued Router for a Micro-network in SoCs [A]. International Conference on Advanced Computer Control [C]. 2011: 533-537.

### 4 结论

为了在地面阶段开展不同卫星之间星载测量通信系统的互联互通试验, 设计了一种具有空间传输信道仿真能力的星载测量通信系统模拟器。系统设计方面, 根据设备间信息交互数据量大小以及实时性的不同, 采用了以太网及 RS422 两种接口协议, 既提高了设备配置的灵活性又增强了运行管理的协调性。管控与数据处理软件的模块化设计以及基带模块的信号分层处理设计方案为模拟器实现性能优化以及功能扩展创造了有利条件, 使得模拟器便于维护以及升级改造。信道仿真器提高了测试的覆盖性以及试验结果的可信性。

除了支持有线测试外, 在加装天线后, 模拟器可实现同星载测量通信系统的无线对接。

#### 参考文献:

- [1] 喻心, 王跃科, 陈建云, 等. 星载测距收发信机的测试系统设计 [J]. 计算机测量与控制, 2014 (2): 317-318.
- [2] 陈晓峰, 孟景涛. 卫星信道模拟器的设计 [J]. 无线电工程, 2011, 41: 51-54.
- [3] 侯博, 刘光斌. 基于 PXI 射频测试平台的多模卫星信号模拟器 [J]. 西安电子科技大学学报 (自然科学版), 2013, 40: 183-188.
- [4] 黄波, 胡修林. 北斗 2 导航卫星星间测距与时间同步技术 [J]. 宇航学报, 2011, 32: 1271-1275.
- [5] 石磊玉, 周益, 王东会, 等. 一种适用于卫星导航系统星间链路的可抢占时隙 TDMA 体制 [J]. 武汉大学学报 (信息科学版), 2012, 37: 714-718.