



南京大學

本科畢業論文
(申請學士學位)

院 系 南京大學計算機系

專 業 計算機科學與技術

題 目 基於 NiosII 的邊緣檢測嵌入式系統設計

年 級 2005 級 學 號 051221037

學生姓名 姜 孟 馮

指導老師 俞建新 職 稱 副教授

2009 年 6 月

摘要

近几年来，随着微电子技术的迅猛发展，新式的嵌入式系统设计已经与十年前大不相同。随着消费电子产品的生命周期一再缩短，产品面市时间的压力的增大以及市场对产品设计的快捷性和灵活性要求的提升，FPGA 技术的发展势头强劲。

Altera 公司于 2000 年提出了 SOPC 概念，并同时推出了相应的开发软件 Quartus II，在可编程器件上实现 SOC 技术。

Nios II 软核处理器是 Altera 公司发布的一款 RISC 处理器，它以 IP 核的形式包含在 SOPC Builder 中，是业界第一款专为可编程逻辑优化的可配置处理器，配套的 IDE 提供完整的 C 语言软件开发支持。

本文采用 Altera 公司 Cyclone 系列的 EP2C70F896C6 为实验器件，利用 Quartus II 7.2 与 SOPC Builder 平台，设计了一个边缘检测嵌入式系统。其中边缘检测模块完全由 Verilog HDL 语言实现，开关实时控制。整个系统包括：Nios II CPU 核、边缘检测模块、UART、FLASH、SSRAM、SDRAM、VGA 控制器等组件。

本文详细介绍了整个系统平台的设计与实现。对边缘检测算法的过程，尤其是 Sobel 算子的计算过程以及硬件实现做了重点介绍。本文新颖之处在于利用 SOPC Builder 平台的高效开发与纯 Verilog HDL 语言的计算模块实现。这使整个系统的开发周期缩短、灵活度高、成本低，而且不损失执行效率。

关键词：边缘检测，Nios II 软核处理器，SOPC，FPGA，Verilog

Abstract

In recent years, with the rapid development of microelectronic technology, modern embedded system design has been very different from how it was 10 years ago. With shortened life cycle of consumer electronics products, the increasing time-to-market pressure, and the requirements of the market for the speed and flexibility that the product design contains, FPGA technology has been a strong momentum.

In 2000, Altera Corporation proposed the concept of SOPC, and released the corresponding Quartus II development software at the same time, realizing SOC technology in programmable devices.

Nios II soft-core processor is a RISC processor released by Altera Corporation, in the form of IP core included in the SOPC Builder. It is the first programmable logic optimization for configurable processors in industry. The assorted IDE provides a complete support of the C language to develop software.

In this paper, I chose Altera's EP2C70F896C6 of the Cyclone II series as the experimental device, using Quartus II 7.2 and SOPC Builder platform to design an embedded system of edge detection. In the designed system the edge detection module is entirely realized by the Verilog HDL language and real-time controlled by the switches. The entire system includes Nios II CPU core, edge detection module, UART, FLASH, SSRAM, SDRAM, VGA Controller, etc.

This paper gives a full description of the design and implementation of the entire system platform. The process of edge detection algorithms, in particular the calculation of Sobel operator and the hardware implementation has done a highlight. Novelty of this paper is to use SOPC Builder development platform for high-performance and to achieve the calculation module purely by Verilog HDL language. This makes the whole design's cycle shortened, in addition with higher flexibility and lower cost, with no loss of operating efficiency.

Key Words: edge detection, Nios II processor core, SOPC, FPGA, Verilog

目录

摘要.....	I
Abstract.....	II
目录.....	III
第一章 绪论	1
1.1 引言	1
1.2 论文选题背景.....	1
1.3 论文框架.....	2
第二章 边缘检测介绍以及算法.....	3
2.1 边缘检测介绍.....	3
2.2 边缘检测算法过程	3
2.3 滤波算法.....	4
2.4 Sobel 算法	5
2.5 Sobel 算法的 Verilog HDL 语言实现	5
第三章 SOPC 技术与 Nios II 架构	8
3.1 SOC 技术介绍	8
3.2 SOPC 技术介绍.....	8
3.3 Nios II 处理器介绍.....	9
3.4 SOPC Builder 介绍	10
3.5 DE2-70 硬件环境介绍	11
第四章 SOPC 技术下的嵌入式系统设计	12
4.1 为什么需要 SOPC 系统？	12
4.2 VGA 控制器.....	12
4.3 建立 SOPC 系统.....	13
4.4 使用 Nios II IDE 完成软件	15
4.5 硬件运行结果.....	16
参考文献	18
致谢.....	18

第一章 绪论

1.1 引言

近十年以来，嵌入式系统技术和嵌入式产品发展越来越热，其应用之广目前已包含：军事、航空航天、网络通信、消费电子、汽车工业、工业控制、信息家电等众多领域。嵌入式产品在各大 IT 产业以及电子工业的经济总额中所占的比重越来越大。

嵌入式系统具有高芯片集成度、硬件软件最小化、高度自动化、快速响应以及高稳定高可靠等基本特点，特别适合于要求实现实时控制和多任务控制的场合。

目前，嵌入式技术已经进入了 SOC(System On a Chip)时代，基于 SOC 技术的高集成度、高复杂度的嵌入式系统开发已经越来越被市场所需要。与此同时，现代电子产品的生命周期越来越短，传统的利用固定逻辑器件的开发方法已经不适应现代市场经济的需求，使用可编程逻辑器件已经越来越成为工业领域的优先选择。

随着 FPGA 技术几十年的发展，目前高端的 FPGA 产品已经达到了数百万个逻辑门，完全有能力实现一个完整的嵌入式系统。而 SOPC(System On a Programmable Chip)技术，更是针对 FPGA 设计所提出，设计快速、灵活，附有完整的软件开发平台(如 Altera 公司的 SOPC Builder)，以及大量丰富的 IP 核，让开发者更容易的集成更多的高质量模块，加速嵌入式产品的开发进程。

1.2 论文选题背景

边缘检测是图像处理和计算机视觉中的基本问题，边缘检测的目的是标识数字图像中亮度变化明显的点。图像属性中的显著变化通常反映了属性的重要事件和变化。边缘检测是图像处理和计算机视觉中，尤其是特征提取中的一个重要研究领域。

当前大多数边缘检测程序都是基于 matlab，以软件的方法实现，缺少硬件实现。效率低，实时性差，不适于在嵌入式产品中实现。

使用 FPGA 以硬件的方式进行边缘检测，可以大大提高算法运行速度，使实时性大大加强。同时，深入学习 SOPC 技术本身，对将来的嵌入式系统设计与开发也有着十分重要的意义，对培养软硬件协同设计意识，从底到高整体系统把握都能提供良好的经验基础。

1.3 论文框架

本文利用 SOPC 这种新的 SOC 设计方法进行嵌入式系统开发,以 Sobel 算法为算法核心,台湾友晶公司 DE2-70 开发板为实验器件,建立了一种基于 SOPC 与 Nios II 的边缘检测嵌入式系统,探讨了新方法较传统方法的优势。

本文的其余章节安排如下:

第二章介绍边缘检测的基本理论,重点分析 Sobel 算法及其硬件实现。

第三章介绍 SOPC 技术、Nios II 架构与 DE2-70 开发板,这些是本文实现 Sobel 边缘检测应用采用的主要技术与硬件平台。

第四章具体介绍边缘检测嵌入式系统的设计与实现,并给出硬件实现结果图。

第二章 边缘检测介绍以及算法

2.1 边缘检测介绍

边缘检测是一种图像处理上的基础工具，通常作为一种特征提取与目标分割的预处理步骤。这个过程会在物体与背景图之间检测到物体的轮廓与边界。一种边缘检测滤镜还可以用来改善模糊或反锯齿视频流的外观。

所谓边缘，是指图像中灰度急剧变化的区域。

在数字图像处理中，边缘检测属于图像分割的范畴，而其计算则属于邻域运算的领域。如果某一个像素落在图像中一个物体的边界上，那么它的邻域将成为一个灰度级变化的带状区。对这种变化最有用的两个特征是灰度的变化率和方向，它们分别以梯度向量的幅度与方向来表示。

基础的边缘检测算子是基于图像梯度的，它把算子的中心对准欲检查的像素，利用卷积运算对图像中算子覆盖的部分进行卷积，然后根据结果计算边缘的强度，最后根据阈值确定边缘。由于这种算法基于图像的一阶导数，通常把对应算子称为一阶算子，常见的一阶算子有 Roberts 边缘算子、Sobel 边缘算子、Prewitt 边缘算子、Kirsch 边缘算子。

一阶导数的局部最大值对应于二阶导数的零交叉点（见图 1），常见的二阶算子是 Laplacian 算子，在这基础之上，有 Marr 边缘检测算法。

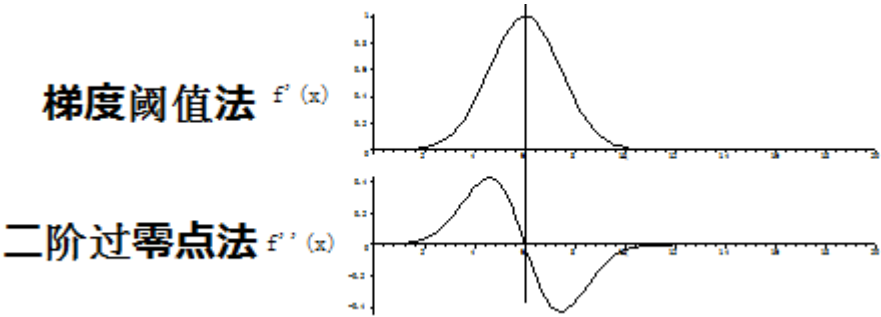


图 1 梯度阈值法与二阶过零点法

目前最优的阶梯型边缘检测算法是 Canny 边缘检测算法，与上不同的是，它使用了双阈值检测。

2.2 边缘检测算法过程

一般边缘检测算法的过程分为四步：

(1) 滤波。边缘检测主要基于导数计算，但受噪声影响，故先行滤波。但滤波器在降低噪声的同时也导致边缘强度的损失。

(2) 增强。增强算法将邻域中灰度有显著变化的点突出显示。一般通过计算梯度幅值完成。

(3) 检测。但在有些图像中梯度幅值较大的并不是边缘点。最简单的边缘检测是梯度幅值阈值判定。

(4) 定位。精确确定边缘的位置。

2.3 滤波算法

通用的滤波方法是利用滤波器函数对原函数求卷积： $g(x) = f(x) * h(x)$

滤波器 $h(x)$ 应满足以下条件：

(1) $|x| \rightarrow \infty$ 时， $h(x) \rightarrow 0$ ， $h(x)$ 为偶函数

$$(2) \int_{-\infty}^{\infty} h(x) dx = 1$$

(3) $h(x)$ 一阶二阶可微

常用的是高斯滤波器：

$$h(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

二维形式：

$$h(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

离散形式：

5 阶， $\sigma^2 = 2$ ，整数化归一化：

$$\begin{bmatrix} 1 & 2 & 3 & 2 & 1 \\ 2 & 4 & 6 & 4 & 2 \\ 3 & 6 & 7 & 6 & 3 \\ 2 & 4 & 6 & 4 & 2 \\ 1 & 2 & 3 & 2 & 1 \end{bmatrix}$$

2.4 Sobel 算法

Sobel 算子

$$H_1(x,y)=\begin{bmatrix}-1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1\end{bmatrix}, H_2(x,y)=\begin{bmatrix}+1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1\end{bmatrix}$$

利用 Sobel 算子对图像求卷积。

$$\phi_1(x,y)=g(x,y)*H_1(x,y)$$

$$\phi_2(x,y)=g(x,y)*H_2(x,y)$$

卷积结果是纵横两个正交方向上的梯度值，边缘强度与方向由下式给出

$$\phi(x,y)=\sqrt{\phi_1(x,y)^2+\phi_2(x,y)^2}$$

$$\theta_\varphi(x,y)=\tan^{-1}\frac{\phi_1(x,y)}{\phi_2(x,y)}$$

然后根据给定阈值对边缘进行测定，没有连接边缘这一步。

Sobel 算子的优点是只使用 2 个 3 阶算子，计算方便，对灰度渐变和噪声较多的图像处理效果较好。相比之下，Roberts 算子使用 2 个 2 阶卷积核，去噪效果不好，而 Kirsch 算子使用 8 个 3 阶卷积核，计算较为繁琐。Sobel 算子在实际中最常用。

2.5 Sobel 算法的 Verilog HDL 语言实现

假设用 P 来表示图像，X 与 Y 来表示算子：

P ₁	P ₂	P ₃
P ₄	P ₅	P ₆
P ₇	P ₈	P ₉

X ₁	X ₂	X ₃
X ₄	X ₅	X ₆
X ₇	X ₈	X ₉

Y ₁	Y ₂	Y ₃
Y ₄	Y ₅	Y ₆
Y ₇	Y ₈	Y ₉

以 P * X 为例，卷积操作所做的是先沿纵轴翻转，再沿横轴翻转后再加权平均，即：

$$P * X=P_9*X_1+P_8*X_2+P_7*X_3+P_6*X_4+P_5*X_5+P_4*X_6+P_3*X_7+P_2*X_8+P_1*X_9$$

Sobel 算子的运算基于矩阵运算，这在 C 语言等高级语言中十分容易实现，不管你用一维数组还是二维数组，总是可以通过数组下标直接定位数组元素，得到图像的 RGB 像素信息。

但在 Verilog HDL 中，不存在数组的概念，没有办法随意的定位。

为此，需要一个 LineBuffer 来实现邻域像素的预存储，并且按照时钟逐步给出需要的像素值，而现成可用的设备是移位寄存器（见图 2）。

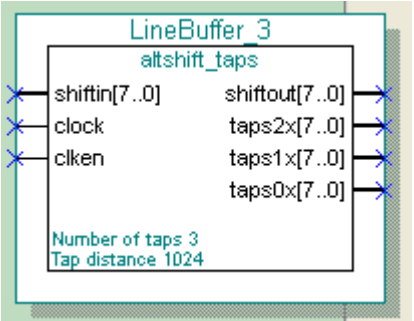


图 2 移位寄存器实现的 LineBuffer

这里 LineBuffer 的长度必须与输出信号的横向宽度一致，由于本文使用的输出设备是 1024x768 分辨率的 VGA 显示器，这里 LineBuffer 长为 1024，shiftout[7..0]信号没有用到。之后即使用常用的 DSP 计算器件，乘加器（见图 3）与并行加法器（见图 4）。

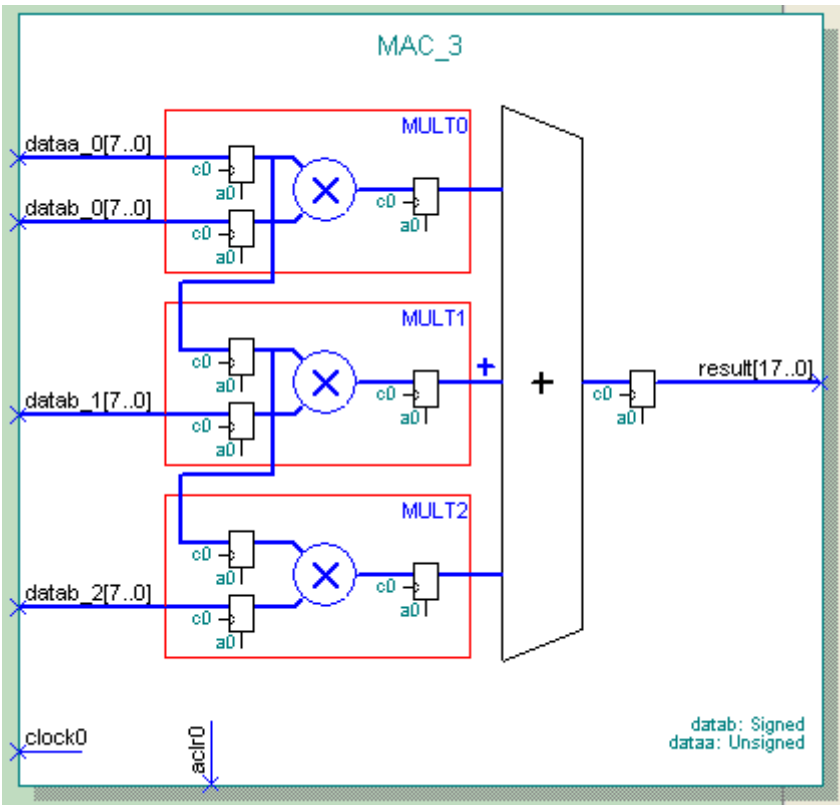


图 3 乘加器

注意乘加器的两个输入口，其中 A 口是从 LineBuffer 中按时钟给出的图像像素值，无符号变量，B 口是算子确定，有符号常量。

如此的乘加器在计算 $P * X$ 时一共需要三个，分别接受 LineBuffer 输出的 3 组 taps，之后将这 3 个乘加器的输出同时接到一个并行加法器的输入上，即完成了 $P * X$ 的计算

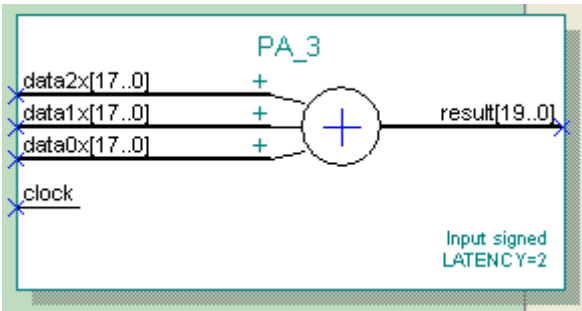


图 4 并行加法器

并行加法器的输入与输出全是有符号数。
整个 $P * X$ 的计算流程如图 5 所示(常量引脚未标注)

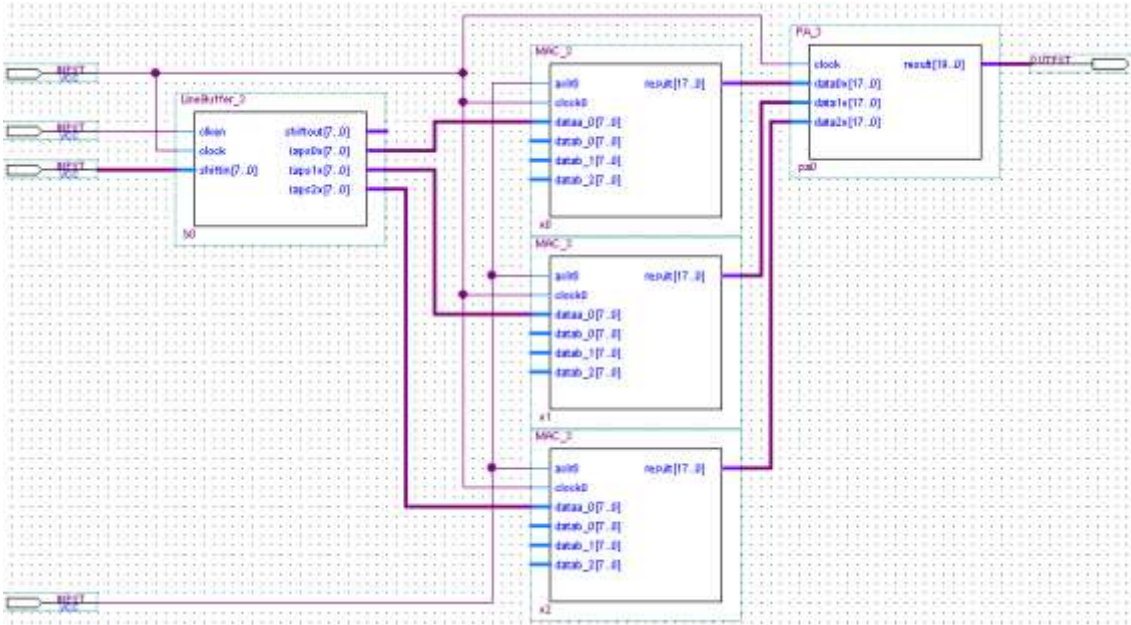


图 5 $P * X$ 计算流程

利用同样的方法可以计算 $P * Y$, P 值同样从同一个 LineBuffer 的 taps 输出引脚上引出。
之后对两个并行加法器的结果 $P * X$ 与 $P * Y$ 求平方和再求平方根（见图 6）即可得到边缘强度。

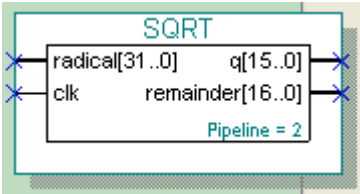


图 6 平方根器

最后的阈值可以用 8 位开关控制，范围为 0~255，足以满足与边缘强度相比较的需要。

第三章 SOPC 技术与 Nios II 架构

3.1 SOC 技术介绍

20 世纪末, 微电子技术在摩尔定律作用下迅猛发展, 集成电路设计和工艺水平逐年精进, 使得原本的大规模集成电路组成的电子系统集成在单个硅片上成为可能, 即所谓的片上系统 SOC(System On a Chip)。

SOC 可以大幅度减小系统所占的片上面积, 大量投产时, 生产成本远低于原本需要多片芯片制成的电路系统。此外, SOC 还具有如下优势:

- 使用改变内部工作电压的方案, 降低芯片功耗。
- 减少芯片对外的管脚数, 简化制造过程。
- 减少外围驱动接口单元和电路板间的信号传递, 加快微处理器处理数据的速度。
- 内嵌的线路可以避免外部电路板上信号传递时所造成的系统干扰。

随着电子设计自动化 (Electronic Design Automation, EDA) 技术和 VLSI 设计的推广, 硬件描述性语言标准的确立, 越来越多的用户开始使用可编程逻辑器件来替代固定逻辑器件。他们使用硬件描述性语言(Hardware Description Language, HDL)定义出一个系统, 用仿真工具进行仿真, 然后将源代码或版图设计递交给半导体芯片代工公司制作样品, 经过严密的测试后, 样品批量生产, 这样就得到了一块集成大部分系统功能的 SOC 芯片。

然而, 现代社会的消费电子生命周期越来越短, 经典的开发流程已经逐渐不能满足需要。开发者需要更短的开发周期, 更高的开发效率。

3.2 SOPC 技术介绍

SOPC, 即 System On a Programmable Chip, 片上可编程系统, 是 Altera 公司在 2000 年专门针对 SOC 设计难题提出的一种基于当前大规模 FPGA 的设计方案。使用百万门级的 FPGA 芯片、功能复杂的 IP 核以及可重构的嵌入式处理器软核来完成一个软硬件协同工作的复杂系统, 设计周期上由于 IP 核的高复用性而得到了大大的缩短, 开发成本也因为 FPGA 的现场可编程能力得到了大大的降低。当前, SOPC 技术已经成为国际上电子系统设计领域的新热点, 具有广阔的应用前景。

Altera 公司、Xilinx 公司、Lattice 公司、QuickLogic 公司等全球最重要的 FPGA 及 EDA 公司, 都分别推出了自己的 SOPC 解决方案。Altera 公司在其最新的 EDA 开发工具 Quartus II 中集成了 SOPC Builder, 并为一些特定的 FPGA 开发板提供了对应的 University

Program IP Core 为教学资源使用，培养新一代的人才。

SOPC 结合了 SOC 与可编程逻辑器件各自的优点，一般具有以下基本特征：

- 至少包含一个嵌入式处理器内核。
- 具有小容量高速片内 RAM 资源。
- 丰富的 IP Core 资源可供使用。
- 足够的片上可编程逻辑资源。
- 处理器调试接口与 FPGA 编程接口。
- 包含部分可编程模拟电路。
- 单芯片、低功耗、微封装。

与同样用来实现 SOC 的 ASIC(Application Specific Integrated Circuit)技术相比，SOPC 更加灵活，设计更加方便，更适合教学、研究使用。

3.3 Nios II 处理器介绍

Nios II 处理器是 Altera 提出的一种软核处理器，也是目前使用得最广泛的软核处理器。它重点强调的是灵活的指令集，而不是某种特定的硬件实现。Nios II 处理器核是用来实现 Nios II 指令架构的硬件逻辑电路，它不包括外设和一些与外设连接的逻辑。

Nios II 架构是一种可配置的架构，其指令集固定，但不同的硬件实现可以针对特定的目标进行优化，比如说选择较小的核或者选择更高的性能，这使得 Nios II 体系结构可以适应不同的应用需要。

Nios II 系列嵌入式处理器目前有三种处理器核，提供常用指令集架构，每一种内核都针对特定的消耗/性能点进行优化，由相同的软件工具链提供支持。设计者可以根据应用需要进行选择。Nios II 处理器内核的三种类型分别是：快速型、标准型和经济型。快速型 Nios II 内核具有最高的性能，经济型 Nios II 内核具有最低的资源占用，而标准型 Nios II 在性能和面积之间做了一个平衡。它们之间的性能对比如下表所示。

三种 Nios II 处理器核性能对比

特性	Nios II/e	Nios II/s	Nios II/f
目标	最小核	较小核	最快的执行速度
DMIPS/MHz	0.15	0.74	1.16
Max.DMIPS	31	127	218
Max.f/MHz	200	165	185
面积/LE	700	1400	1800
流水线	1 级	5 级	6 级
外部地址空间 /GB	2	2	2

指令 Cache	无	512B~64KB 静态分支 预测	512B~64KB 动态分支 预测
数据 Cache	无	无	512B~64KB
ALU	只能移位操作、无硬件乘法、除法	硬件乘法、除法和移位 操作	硬件乘法、除法和移位 操作

嵌入式产品的可用资源是受限的，资源占用与执行效率的矛盾尤其突出。可配置的处理器的特定处理器优势巨大。你可以选择使用多大的片上缓存，你可以选择是否加入调试模块，你可以选择使用硬件还是软件实现你想要的特性。

3.4 SOPC Builder 介绍

在 Quartus II 中，一个 SOPC 系统对应一个 Quartus 工程，这个系统由多个组件(component)构成，常见的组件有：

- Nios II 等处理器
- 微控制器外设
- 定时器
- UART、SPI 等串行通信接口
- 通用输入输出(GPIO)
- 存储器接口
- 总线和总线桥
- 数字信号处理(DSP)内核

SOPC Builder 可以使用图形界面配置的方式添加各个组件，为此设计者需要了解自己所使用的 FPGA 各个部件的配置参数。SOPC Builder 会根据设计者的需求将各个组件与处理器相连接，并自动完成外设和存储器的地址映射、中断控制和总线控制等工作

完成配置之后，SOPC Builder 会根据配置生成 VHDL 或 Verilog HDL 语言描述的系统级设计代码，并自动生成部分外设的硬件抽象层(HAL)代码和底层硬件驱动代码。同时会有一个.ptf 文件生成，记录了这个系统中各个组件的端口、基址等信息，被用于 Nios II IDE 生成 C 语言头文件。SOPC Builder 本身的配置信息被记录在一个.sopc 文件中。

SOPC Builder 也允许用户自定义逻辑组件，像默认的组件一样添加到系统中。这些自定义逻辑组件可以由硬件描述语言写成，但必须满足一定的规范。这些硬件描述语言文件的位置以 tcl 语法被记录在.qip 文件中。

3.5 DE2-70 硬件环境介绍

本文采用的是台湾友晶公司出产的 DE2-70 实验平台，使用 Altera 公司的 FPGA 芯片 EP2C70F896C6，适用 Altera 公司软件工具 Quartus II 7.2/8.0。

EP2C70F896C6 芯片中含有数量达 68,416 个逻辑单元，622 个引脚，1,152,000bits 的片上内存，300 个 9-bit 乘法单元，并且还拥有 4 个不同的 PLL。

开发板本身集成了一块 2-Mbyte 的 SSRAM，两块 32-Mbyte 的 SDRAM，一块 8-MByte 的 Flash 闪存，资源十分丰富。红色 LED 计 18 个，绿色 LED 计 9 个，HEX 7 位发光管 8 组。下方 18 个开关，4 个按钮一般用于输入简单控制信号。左方中央小型的 LCD 可以滚动输出两行字符。除此之外，上方一排标准接口可以方便的为系统添加多样外设，如鼠标、键盘、耳机、VGA 显示器。详细参见图 7

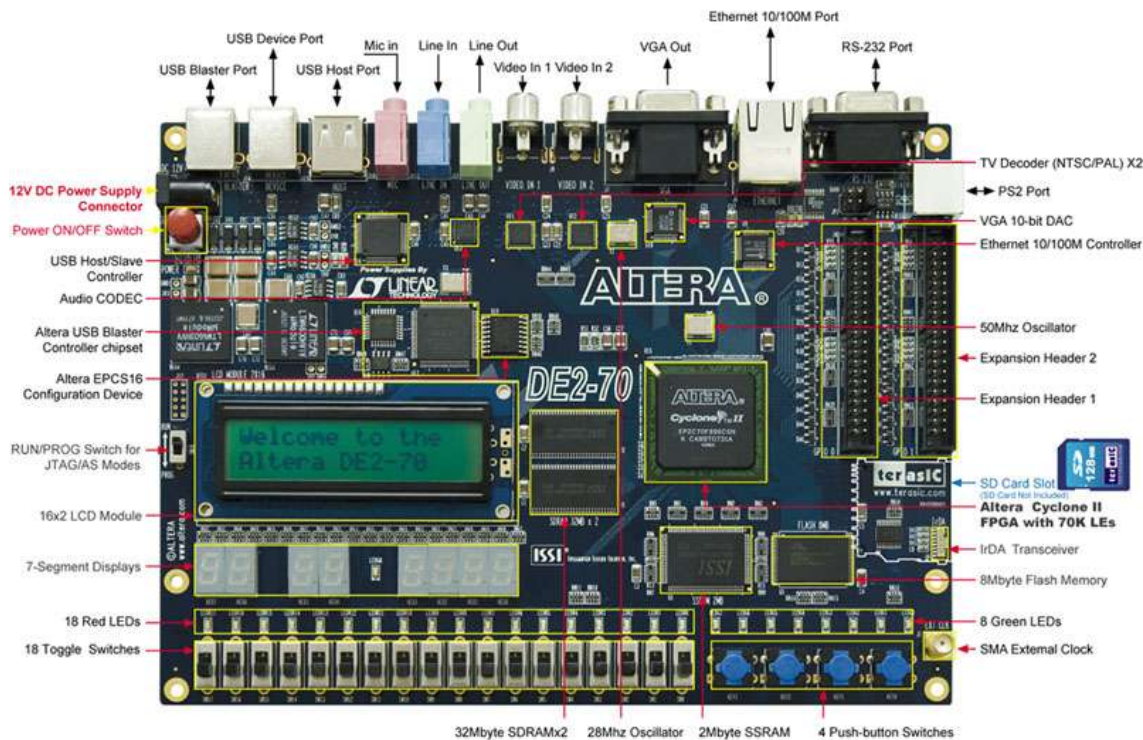


图 7 DE2-70 开发板布局

第四章 SOPC 技术下的嵌入式系统设计

4.1 为什么需要 SOPC 系统？

前文已经介绍了 SOPC 系统是什么，现在要利用 SOPC Builder 建立一个真正的 SOPC 系统来完成边缘检测嵌入式系统的设计。

正如第 2.5 节 Sobel 算法的 Verilog HDL 实现所述的一样，边缘处理模块本身可以完全用硬件实现，而且可以达到实时检测，开关控制。但是为了解决输入源问题，必须要有软件支持不可，说得简单一些，也就是“如何读取一张图片”。

平时我们所用的图片，无论是 jpg 格式还是 bmp 格式，它都必须是一个文件，而想要读取文件则必须要通过文件系统。而且文件系统的存储媒介还必须是非易失的，否则建立的系统根本没有意义。

如此一来，如果要完全通过硬件读取文件，则必须要对存储媒介上的文件系统(如 FAT、FAT32)的格式有非常清楚的了解，有可能还牵扯到分区问题，这不在本文讨论范围之内。一般来说，“如何读取一张图片”的问题由 C 语言中的 fopen() 函数负责，也就是由软件控制，通过 Nios II 指令的方式完成。

4.2 VGA 控制器

Altera 公司的 SOPC Builder 中，已经默认提供了大多数所需的组件，唯一缺少的是 VGA 控制器。必须由用户自己完成一个 VGA 控制器的组件，然后添加到 SOPC 系统中。

由于工业标准的 VGA 显示器分辨率实在太低，本文采用的显示器分辨率是 1024*768，必须首先根据这个信息确定 VGA 信号的 Timing 参数。TinyVGA.com 官方网站上的 VGA Signal Timing 参数如下：

General timing	
Screen refresh rate	60 Hz
Vertical refresh	48.363095238095 kHz
Pixel freq.	65.0 MHz

Horizontal timing (line)			Vertical timing (frame)		
Polarity of horizontal sync pulse is negative.			Polarity of vertical sync pulse is negative.		
Scanline part	Pixels	Time [μs]	Frame part	Lines	Time [ms]
Visible area	1024	15.753846153846	Visible area	768	15.879876923077
Front porch	24	0.36923076923077	Front porch	3	0.062030769230769
Sync pulse	136	2.0923076923077	Sync pulse	6	0.12406153846154
Back porch	160	2.4615384615385	Back porch	29	0.59963076923077
Whole line	1344	20.676923076923	Whole frame	806	16.6656

首先确定时钟为 65MHz，由 50MHz 经 4 倍频再 3 分频可得，见图 8 中的 c2。

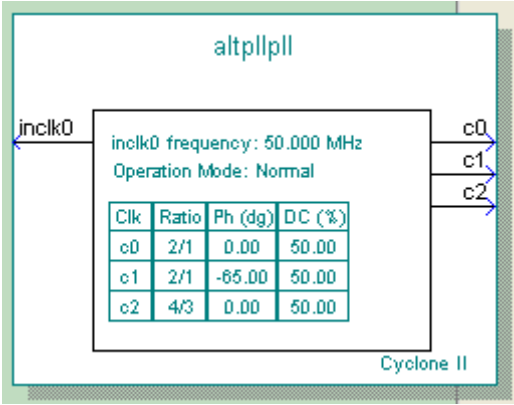


图 8 SOPC 系统中的时钟信号一览

VGA 显示器采用逐行扫描，从左上到右下，每扫完一行，电子束回到屏幕下一行的起始位置，在回扫期间，显示器对电子束进行消隐，每行结束是用行同步信号 HS 进行行同步；扫描完所有行，再由场同步信号 VS 进行场同步，并使扫描回到屏幕的左上方，同时进行场消隐，预备下一场的扫描。

驱动需要输出 5 个主要信号：

R、G、B 三个像素值的信号以及 H_SYN 与 V_SYN 两个同步信号。

除此以外仍需 3 个次要信号：

时钟信号 CLK，同步使能信号 SYN_N 和计算逻辑信号 Blank_N（用于下文模块间的同步）。

4.3 建立 SOPC 系统

使用 SOPC Builder 建立系统即可，系统与 SDRAM 主频 100MHz，VGA 主频 66.66MHz。

Target

Device Family: Cyclone II

Clock Settings

Name	Source	MHz
clk	External	50.0
pll_system	pll.c0	100.0
pll_memory	pll.c1	100.0
pll_vga	pll.c2	66.666666

Use	...	Module Name	Description	Clock	Base	End	IRQ
<input checked="" type="checkbox"/>		cpu	Nios II Processor	pll_system	0x05400800	0x05400fff	
<input checked="" type="checkbox"/>		pll	PLL	clk	0x05401000	0x0540101f	
<input checked="" type="checkbox"/>		jtag_uart	JTAG UART	pll_system	0x05401040	0x05401047	
<input checked="" type="checkbox"/>		tristate_bridge_flash	Avalon-MM Tristate Bridge	pll_system			
<input checked="" type="checkbox"/>		cfi_flash	Flash Memory (CFI)	pll_system	0x04800000	0x04ffffff	
<input checked="" type="checkbox"/>		tristate_bridge_ssram	Avalon-MM Tristate Bridge	pll_system			
<input checked="" type="checkbox"/>		ssram	Cypress CY7C1380C SSRAM	pll_system	0x05200000	0x053fffff	
<input checked="" type="checkbox"/>		sdram	SDRAM Controller	pll_system	0x02000000	0x03fffff	
<input checked="" type="checkbox"/>		vga_controller_gray_inst	vga_controller_gray	multiple	0x05401020	0x0540103f	
<input checked="" type="checkbox"/>		sysid	System ID Peripheral	pll_system	0x05401048	0x0540104f	

图 9 SOPC 系统组件一览

之后用 Verilog HDL 语言完成顶层实体的设计，将 VGA 控制器的输出接到第 2.5 节中实现的 Sobel 边缘检测模块上，再将 Sobel 边缘检测模块的输出连到顶层 VGA 输出引脚。SOPC 系统模块与 Sobel 边缘检测模块间时钟的同步信号利用 VGA 控制器中 H_SYN 与 V_SYN 两个信号的“位与”所确定，即表示图像在画面中时才执行 Sobel 边缘检测模块中的计算。

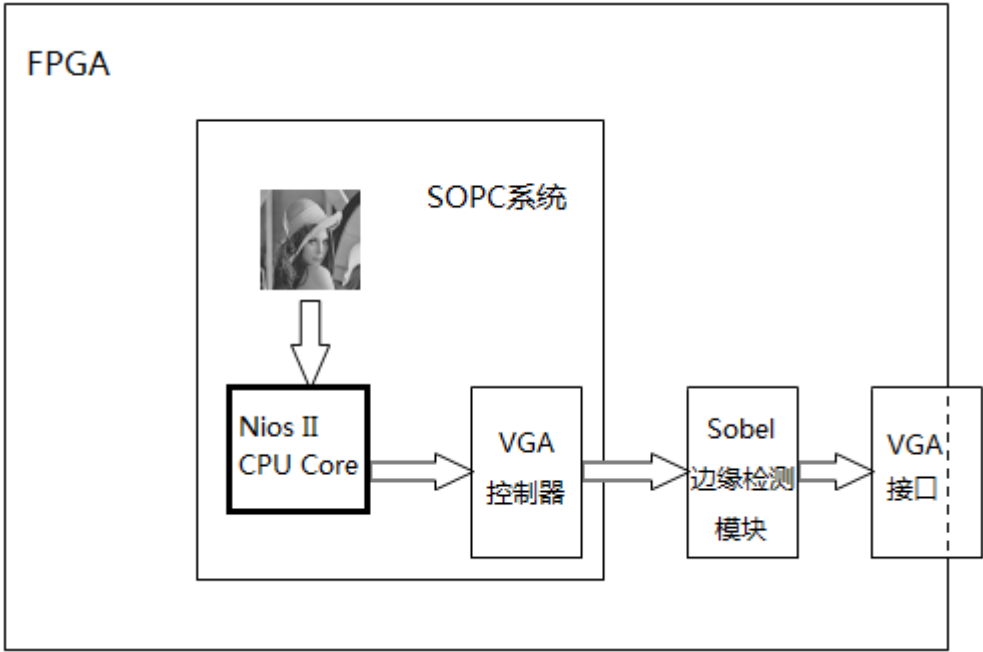


图 10 整个边缘检测嵌入式系统结构示意图

Flow Status	Successful - Tue May 26 16:01:03 2009
Quartus II Version	7.2 Build 151 09/26/2007 SJ Full Version
Revision Name	EdgeDetection
Top-level Entity Name	EdgeDetection
Family	Cyclone II
Device	EP2C70F896C6
Timing Models	Final
Met timing requirements	No
Total logic elements	3,251 / 68,416 (5 %)
Total combinational functions	2,908 / 68,416 (4 %)
Dedicated logic registers	1,791 / 68,416 (3 %)
Total registers	2076
Total pins	253 / 622 (41 %)
Total virtual pins	0
Total memory bits	51,152 / 1,152,000 (4 %)
Embedded Multiplier 9-bit elements	30 / 300 (10 %)
Total PLLs	1 / 4 (25 %)

图 11 编译结果图

由编译结果可见，系统使用的芯片是 Cyclone II EP2C70 芯片。整个系统用了大约 3200 个逻辑单元，使用了 6KB 多的 FPGA 内存资源和 30 个 9bit 乘法器资源。本文设计的边缘检测嵌入式系统并没有占用较多的 FPGA 资源，也因此未能充分的发挥 FPGA 芯片的作用，本系统只是起了一个演示作用。

系统硬件部分的设计到此结束，接下来是软件部分设计。

4.4 使用 Nios II IDE 完成软件

这里采用 Nios II 7.2 IDE 自带模版的 zip_filesystem 来实现文件系统，图片以无压缩 zip 格式利用 JTAG 下载到 flash 中，通过 C 函数 fopen() 打开来使用。

软件部分并不处理数据，仅仅把得到的图片数据送往上文 SOPC 系统中的 VGA 控制器模块，由其生成所需的 VGA 信号。

需要注意的是，BMP 文件的存放方式是由左下到右上，而 VGA 的扫描方式是从左上到右下。如果按序输出图片，则图片必定上下颠倒。故在数据传送过程中，由软件控制将图片沿横轴纵向翻转，借助 C 语言中的数组下标，这是很容易实现的。

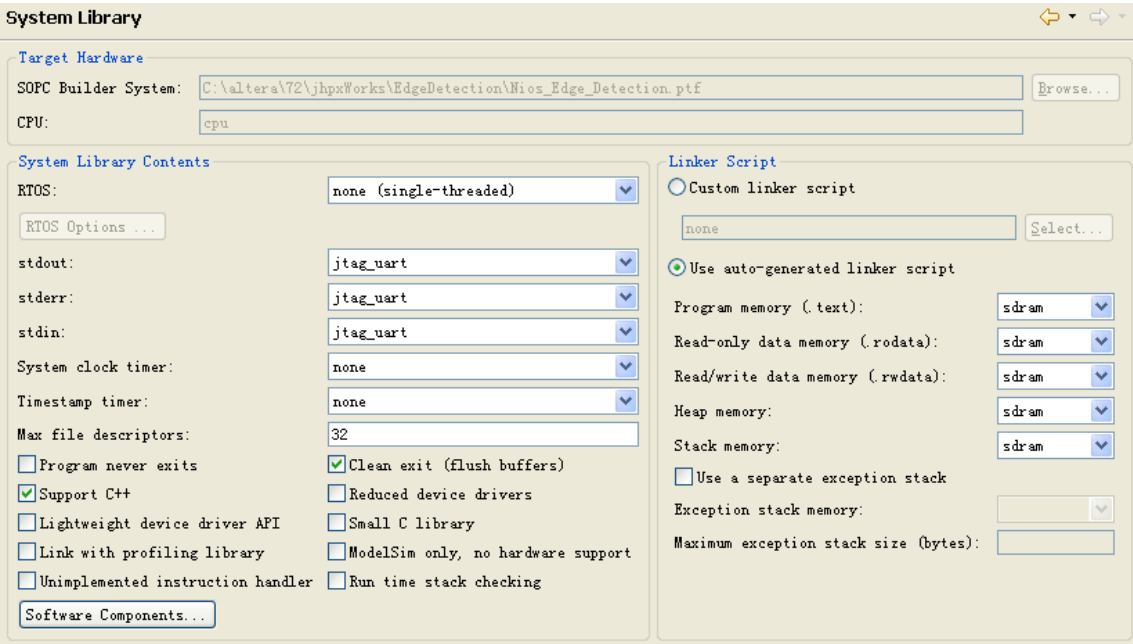


图 12 System Library 配置

4.5 硬件运行结果

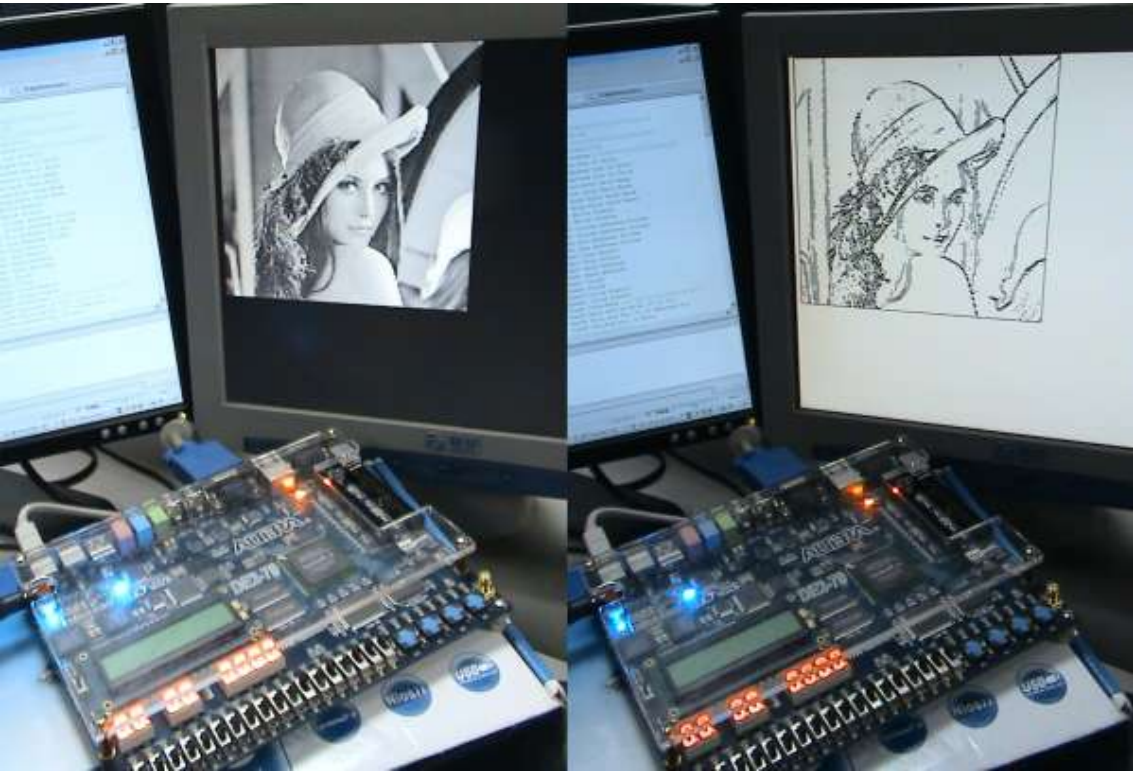


图 13 系统实际效果图



图 14 开关控制下的阈值调整

可以看到，结果图已经可以达到预期的边缘检测效果，即开关控制选择灰度输出或边缘检测结果输出与开关控制选择阈值来调节边缘检测结果。

参考文献

- (1) Altera Corporation, White Paper, *Video and Image Processing Design Using FPGAs*, 2007
- (2) Altera Corporation, *Adaptive Edge Detection for Real-Time Video Processing using FPGAs*
- (3) 俞建新, 王健, 宋健建, *嵌入式系统基础教程*, 机械工业出版社, 2008 年 3 月
- (4) 张志刚, *FPGA 与 SOPC 设计教程——DE2 实践*, 西安电子科技大学出版社, 2007
- (5) Kenneth R. Castleman, *Digital Image Processing*, 2002
- (6) 孙睿, *基于 Nios II 的小波变换嵌入式系统设计*, 2008

致谢

在这里我应该首先感谢培养我的南京大学, 感谢计算机科学与技术系全体领导与老师们, 感谢在 4 年学习生活中给过我帮助、陪我一起走过无数困难与坎坷的同学们。

在整个毕业设计过程中, 我得到了俞建新老师的悉心指导与大力支持。俞老师给我提供了很好的学习环境和实验条件, 大量的资料与充实的教学, 使我的专业知识与实践能力都得到了很大的提高。老师在工作中的认真负责与对学生的关心爱护, 都是我在以后学习工作中的榜样。

本文的完成还得益于同一教研组的学长学姐们, 与他们的交流、合作以及在各种问题上的讨论使我受益匪浅。感谢已经毕业的孙睿学长, 他留下的实验源码与学习资料, 使我在整个毕设过程中少走了不少弯路。

感谢我的舍友们: 姜堃、姜鹏、贾新润、李飞、向圆圆、黄宇飞。他们在生活和学习上都给了我很大的帮助。

最后要衷心地感谢我的父母和家人, 感谢他们生活上和精神上对我的支持和鼓励, 帮助我顺利完成学业。愿他们永远健康、幸福!