

No detail can be too detailed

思则变 变则通 通则达

目录视图

摘要视图

RSS 订阅

个人资料



gffsky1990



访问：42596次
积分：1880
等级： BLOG > 4
排名：第18364名

原创：136篇
转载：8篇
译文：0篇
评论：7条

文章搜索

文章分类

- linux (21)
- c/c++ (99)
- verilog HDL (8)
- FPGA (16)
- 存储器 (6)
- gnu (30)
- linux内核2.6研究 (2)
- 编程规范 (18)
- AT&T汇编 (7)
- 链接 (5)
- 网络编程 (5)
- linux0-12 (9)
- git (1)
- 编程之美 (43)
- 编译器 (8)
- unix (4)
- python (3)
- 数据库 (1)
- php (2)
- 框架 (2)

关于模块复用和Resource Sharing 的思想（在FPGA中）

2014-10-22 17:15 465人阅读 评论(0) 收藏 举报

分类： **FPGA (15)** verilog HDL (7)

版权声明：本文为博主原创文章，未经博主允许不得转载。

在FPGA中，逻辑单元以及各种乘法加法器是相对比较珍贵的，在行业中可能提到的一个原则为“速度与面积”兼顾。要想速度快，并行处理，但是并行处理的代价就是需要消耗许多逻辑资源。在实际工程中一些小思想很重要。

Sharing模块复用和Resource Sharing是一种节约面积的思想，在设计电路中，在不影响性能的情况下，实现最少的逻辑资源开销。

下面是一个实例，求8bit补码平方器；需要注意的是正数补码为自身，负数的补码为取反加一。

方案一

```
[cpp] view plain copy print ?
01. module resource_sharing(
02.     data_in,
03.     square
04. );
05.
06. input [7:0] data_in;
07. output [15:0] square;
08.
09. wire [7:0] data_in_neg;
10. assign data_in_neg = ~data_in + 1'b1;
11.
12. assign square = (data_in[7] == 1'b1) ? (data_in_neg*data_in_neg) : (data_in*data_in);
13.
14. endmodule
```

方案二

```
[cpp] view plain copy print ?
01. module resource_sharing(
02.     data_in,
03.     square
04. );
05.
06. input [7:0] data_in;
07. output [15:0] square;
08.
09. wire [7:0] data_multi;
10. assign square= data_multi*data_multi;
11. assign data_multi = (data_in[7] == 1'b1) ? (~data_in + 1'b1) : (data_in);
12.
13. endmodule
```

文章存档

2016年07月 (2)

2016年06月 (2)

2016年05月 (2)

2016年01月 (5)

2015年09月 (2)

展开

阅读排行

地址解析协议（ARP）的 (1299)

关于校验算法的学习（学 (1001)

解决关于哈夫曼编码计算 (863)

关于同步RAM和异步RAI (769)

FPGA实现网络通信时的I (712)

关于nios下载出现（Con (644)

本以为理解const，其实 (632)

将Bmp图像文件转成mif (565)

每天小练笔6-空瓶换饮料 (504)

研究研究看似简单的swit (500)

评论排行

本以为理解const，其实 (4)

关于 linux 管道中 read 和 (2)

小编程题--解决一维迷宫 (1)

const用法详解 (0)

在c/c++ 中extern的研究 (0)

关于 ubuntu与 linux arm (0)

socket 关于INADDR_AN (0)

linux 中信号量 和 互斥量 (0)

Copy-on-write (0)

Python中的callable是基 (0)

推荐文章

* Android安全防护之旅---带你把Apk混淆成中文语言代码

* TensorFlow文本摘要生成 - 基于注意力的序列到序列模型

* 创建后台任务的两种代码模式

* 一个屌丝程序员的人生（六十）

* WKWebView与js交互之完美解决方案

* 年轻人，“砖砖瓦瓦”不应该成为你的梦想！

最新评论

关于 linux 管道中 read 和 write的
gffsky1990: @qq_32105589:不好意思呀，当时学这个没多久，有些资料不知道怎么查，写的比较随意，下次我会...

关于 linux 管道中 read 和 write的
qq_32105589: 你这说和没说一个样啊

本以为理解const，其实那只是冰
gffsky1990: @astrotycoon:果然如此，非常感谢！！

本以为理解const，其实那只是冰
astrotycoon: @u010442328:我也只是猜测，你看下你在VS里的代码源文件是不是不是以.CPP为后缀的，因为我改...

本以为理解const，其实那只是冰
gffsky1990: @astrotycoon:我上面是不是哪个写的有问题呀，请

比较这两个方案：实现要求的功能，但是很明显第一种使用了两个乘法器，第二个使用了一个乘法器。

拓展一下，这里是如何体现模块共享的，首先不管是正数还是负数都需要乘法器，而且这两者并不是并发的，要进行选择。所以第一种方案是对各种情况运算求结果，然后进行选择结果。第二方案是对源进行选择，然后再运算求结果。

最近在完成图像采集板卡的项目，里面有一点还是蛮相似的，就是对原始24bit图像首先进行二值化，然后再进行边沿检测等操作，而不是直接对原始图像做处理。

顶

踩

0

0

上一篇

FPGA中的I/O电平研究

下一篇

利用C指针编写自己的一些数据结构（链表）

我的同类文章

FPGA（15）		verilog HDL（7）	
• nios软核cpu中架构类shell...	2015-07-31 阅读 292	• FPGA实现网络通信时的网...	2015-07-16 阅读 712
• 关于nios下载出现（Conne...	2015-06-11 阅读 642	• arp协议在fpga上的一个小...	2015-05-04 阅读 405
• nios项目中的小型人机交互...	2015-01-24 阅读 224	• 动态延迟模块的verilog编写	2015-01-16 阅读 397
• FPGA中各种分频的verilog ...	2014-11-14 阅读 272	• 将Bmp图像文件转成mif格...	2014-11-03 阅读 564
• 数据结构之FIFO的一些问题	2014-10-30 阅读 217	• 关于NIO中Avalon总线的...	2014-10-29 阅读 482
更多文章			

猜你在找

C语言及程序设计提高	Cross-Origin Resource Sharing Specification
python函数及常用模块	Cross-Origin Resource Sharing1 跨域资源共享
PHP魔鬼训练之设计模式篇	Rails应用使用跨域资源共享Cross-origin resource
PHP设计模式	Cross-Origin Resource Sharing协议介绍
iOS架构设计	Spring 31 MVC REST 支持之跨域访问Cross-origin

查看评论

暂无评论

您还没有登录,请[登录]或[注册]

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题

Hadoop

AWS

移动游戏

Java

Android

iOS

Swift

智能硬件

Docker

OpenStack

VPN

Spark

ERP

IE10

Eclipse

CRM

JavaScript

数据库

Ubuntu

NFC

WAP

jQuery

BI

HTML5

Spring

Apache

.NET

API

HTML

SDK

IIS

Fedora

XML

LBS

Unity

Splashtop

UML

components

Windows Mobile

Rails

QEMU

KDE

Cassandra

CloudStack

FTC

coremail

OPhone

CouchBase

云计算

iOS6

Rackspace

Web App

SpringSide

Maemo

Compuware

大数据

aptech

Perl

Tornado

Ruby

Hibernate

ThinkPHP

HBase

Pure

Solr

Angular

Cloud Foundry

Redis

Scala

Django

Bootstrap

指正，虚心学习

本以为理解const，其实那只是冰
astrotycoon: VS与GCC的处理是
一样的，只是你源文件是以.cpp
为后缀的。

小编刷题--解决一维迷宫问题
傻丫头与科技: 谢谢小伙伴的分
享，学习了(*^__^*)