

西南科技大学

硕士学位论文

基于FPGA的经济型MPEG-2运动图像编码器IP核设计

姓名：钟声

申请学位级别：硕士

专业：通信与信息系统

指导教师：李众立

20070408

## 摘 要

现场可编程门阵列（FPGA）具有使用灵活、体系结构可变、价格低廉、执行速度快等优点，很好地适应了现代远程教育中的多元化需求。因此，本文从经济实用的角度出发，为山区、偏远农村现代远程教育，设计了一种基于 FPGA 的经济型 MPEG-2 运动图像编码器的 IP 核。

本文在现代远程教育需求与运动图像编码标准 MPEG-2 的交叉点选题，针对山区、偏远农村现代远程教育这一特定的应用领域，开发基于 FPGA 的经济型 MPEG-2 运动图像编码器 IP 核上进行了一些探索和研究。论文针对 FPGA 设计特点以及本系统的应用环境，采用 VHDL 语言自顶向下设计了 MPEG-2 编码器中的核心部分的 IP 核，重点解决了其中的二维 DCT 模块、可变字长编码模块当中的关键路径时延过大的问题，提高了系统的最大工作频率，并且对运动估计模块采用了一种高度并行及紧凑流水线技术的运动估计电路结构，通过改进搜索窗缓存、PE 运算阵列等单元，获得了较高的运算速度增益并有效地降低了电路功耗。

论文在设计思路上力图体现个人的创新，然后通过 ALTERA 公司的 Cyclone EP1C12Q240C8 器件对设计进行了验证，并将验证结果与 MPEG-2 TM5 中的校验模型比较，证实了该设计的正确性和优质性。

**关键词：**现代远程教育      FPGA      MPEG-2 编码器      IP 核

## Abstract

FPGA with the advantages of flexibility, variable structure, low price ,high implementation speed, is good to meet the diversified needs of the modern distance education. Therefore, from the economic point of view, a FPGA-based economic MPEG-2 encoder IP core is designed for the application to modern distance education in mountain and remote rural.

Based on the needs of modern distance education and MPEG-2 standard, the research is chosen to design an IP core for FPGA-based economic MPEG-2 encoder, for the specific application to modern distance education in mountain and remote rural. The dissertation design the circuit through focusing on the FPGA design characteristics and the application circumstances, using VHDL design of a top-down approach to complete the central part of the MPEG-2 encoder IP core, which focuses on solving the 2-D DCT module, the variable length code module very much delay in the critical path, and enhances the maximum operating frequency of the system. Moreover, in the motion estimation module, a kind of highly parallel and compact pipeline technology is applied, mainly by improving the search window cache, PE computational array module, which are in favour of gaining access to very high computational speed and effectively reducing the power consumption.

Besides, the design thoughts of this dissertation are to embody the innovation, and confirmed by the accuracy and quality of the design result , with comparison to MPEG-2 TM5 calibration models, which is adopted by Cyclone EP1C12Q240C8.

**Key words:** modern distance education; FPGA; MPEG-2; IP core

---

## 独 创 性 声 明

本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得西南科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

签 名：钟 青

日 期：2007.5.25

## 关于论文使用和授权的说明

本人完全了解西南科技大学有关保留、使用学位论文的规定，即：学校有权保留学位论文的复印件，允许该论文被查阅和借阅；学校可以公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存论文。  
(保密的学位论文在解密后应遵守此规定)

签 名：钟 青

导师签名：李 友 多 日 期：2007.5.25

# 1 绪论

## 1.1 选题目的及意义

二十一世纪的悄然到来,人类社会即将进入信息化社会。信息社会对人材知识结构和能力素质提出了新的要求,教育作为培养人的手段,也呈现出了新的教授形式和特点,远程教育(Distance Education)作为一种新兴的教育方式,突破了时间地域的限制,成为教育系统中一支重要的力量。相对于传统教育而言,远程教育具有以下优势:第一,充分的交互性。虽然教师和学生处在不同的地方,但通过远程教育平台,利用现代的通信网络,教师和学生之间能够得到充分的交流,仿佛就在同一个教室里似的。第二,学习的自主性。在远程教育系统中,学生的学习自主性得到了充分的发挥。学生按照自己的时间和计划,选择学习的课程内容。并且,对于学习中的问题,能够及时得到相应的解答。第三,教育资源共享性。在我国现有条件下,教学资源还比较匮乏,并且分布很不平衡。通过远程教育的方式,可以使到最偏远的山区的学生也能享受到大城市里一流的教师的讲课,从而实现资源共享<sup>[1][2]</sup>。

自 2001 年以来,在国家教育部实施农村中小学现代远程教育的实践中,我国逐渐形成了教学光盘播放点、卫星教学收视点和计算机教室三种模式。经过国家、社会的大力推广,这三种教学模式都取得了相应的成绩,为我国人才的能力培养和创造力培养做出了巨大的贡献。但是,另外一方面我们也不得不承认,由于这三种教学模式自身因素的限制——基于 Windows 操作系统和基于 PC 计算机,高昂的系统软硬件价格和操作复杂度使得这三种现代远程教学模式在山区、偏远农村中的推广受到了较大的障碍。因此,为了解决这一难题,满足山区、偏远农村小孩也享有平等受教育的权利,西南科技大学 2003AA11606 课题组在吸收和消化了这三种教育模式的思想精华的前提下,引入国产软硬件技术(Linux-NC 技术平台)和双向互动教育技术,建成了“山村多通道网络教学环境”,即:基于 Linux 平台和 WEB 技术的交互式辅助教学平台、Linux/NC 计算机教室、多通道网络同步课堂、基于 Linux/NC 架构下的教育资源库、Linux 环境下的小学教材,从而大幅度的降低了成本,为远程教育的教学资源建设、网络课程的推广奠定了坚实的技术基础和物质基础。

作为“山村多通道网络教学环境”的配套设施,论文的主要目标是研究

和开发适用于山区、偏远农村现代远程教育的一种低成本、高性能的基于 FPGA 的 MPEG-2 运动图像编码器的 IP (Intellectual Property) 核。该 IP 核的设计应用了当前最新的微电子技术、电路设计自动化技术和运动图像编码技术的发展成果, 具有低价格、功耗低、简单、可靠、开放的特点。并且, 其中的二维 DCT、量化/逆量化、可变字长编码和运动估计模块, 还可以作为独立的 IP 核单元, 根据需要略加修改就可以应用到其他含相关算法的图像/视频压缩芯片的设计中去。

## 1.2 数字视频编码技术的发展与应用现状

### 1.2.1 数字视频编码标准的发展历史及其现状

随着现代通信技术的飞速发展和通信业务的不断拓展, 多媒体通信已经成为通信业务发展的必然趋势, 并已经日渐成为人们进行信息交流的基本方式之一。传统的语音服务已经不能满足需求, 图像信息传输已经成为现代通信系统所应提供的基本服务。人们一直期盼实现远距离的图像通信, 通过可视电话、手机和会议电视等形式实现话音和图像的同步传输。但是, 图像所含的数据量大大超过了语音的数据量, 传输带宽也大大超过语音的传输带宽。随着数据压缩技术的发展, 使经过压缩处理后的图像信号带宽大大减少, 从而可以在一般的通信系统中传输图像。数字视频和图像压缩编码技术的研究, 已历经半个世纪, 在理论和工程上都取得了很多成果。进入 20 世纪 90 年代, 在国际标准化组织 ISO 和国际电信联盟 ITU 等国际组织的协调下, 对视频和图像编码的成果进行了收集、整理、综合和加工, 制定了几个通用的压缩编码标准, 包括适用于二值图像的 JBIG、用于连续灰度和彩色静止图像的 JPEG (Joint Picture Experts Group)、用于 64K 视频传输的 H.261、面向 1.5M 数字视频和音频传输及存储的 MPEG-1, 面向高品质数字视频和音频传输及存储的 MPEG-2, 以及适于低码率视频编码的 H.263。这些标准的算法主要由四类技术混合构成, 即正交变换、量化、熵编码和运动估计<sup>[2-12]</sup>。

#### (1) JPEG 标准

JPEG 是联合图像专家小组的英文缩写。这个小组开发研制除连续色调、多级灰度、静止的数字图像压缩编码方法, 这个压缩编码方法称为 JPEG 算法, 被确定为 JPEG 国际标准。它是以 DCT 变换和变长编码为基础的。JPEG 标准不仅适用于静止图像的编码, 也同时适用于运动图像的帧内图像的编码, 是数字视频和图像压缩编码技术发展的基础。

## (2) H.261 建议和 H.263 建议

H.261 标准是由 ITU-T 第十五研究组为在窄带综合业务数字网(N-ISDN)上开展双向声像业务(会议电视、可视电话)而制定的,它是视频压缩方面提出的第一个标准,目标是把数字视频速率压缩到 64Kb/s~2.048Mb/s。H.261 只对 CIF 和 QCIF 两种图像格式进行处理,编码算法由 DCT、运动补偿和霍夫曼编码组成,由于该标准用于实时业务,为减小编解码延时,它的帧间预测只采用了前向预测(即 P 帧预测)。

为了满足近年来在普通公用电话网和移动电话网上进行可视电话通信的需要,即视频压缩信号速率小于 64Kb/s,ITU-T 在 H.261 建议基础上进行改进,于 95 年提出了 H.263 建议—“甚低码率通信的视频编码”。它与 H.261 的主要区别就是引进了半像素精度的运动估计,添加了一些可选项,如双向预测的 B 帧、用基于句法的算术编码代替霍夫曼编码等等。

## (3) MPEG 标准

MPEG 委员会的活动始于 1988 年,其目标是对视频及其伴音、对数字存储媒质制定一个标准。MPEG-1 于 1992 年正式发布,它的主要指标是:在数字存储介质中实现对活动图像和声音的压缩编码,编码码率最高为 1.5Mb/s,标准的正式规范在 ISO/IEC11172 中。MPEG-1 是一个开放的,统一的标准,在商业上获得了巨大的成功。尽管其图像质量仅相当于 VHS 视频的质量,还不能满足广播级的要求,但已广泛应用于 VCD 等家庭视像产品中。

MPEG-2 可理解为 MPEG-1 的进一步发展,它选择的速率为 1~10Mb/s, MPEG-2 的成功之处在于开发了通用的压缩编码方法,是一种规范化的全系统,可满足不同的图像分辨率和存储成本和处理速度的需要。MPEG-2 主要应用于广播电视、DVD 以及高清晰度数字电视(HDTV),还可应用于交互式的点播视频(VOD),此外,还能适配于 ATM 这种新兴的宽带通信网。MPEG-2 是视频和音频信号数据压缩国际标准 MPEG-1 的扩展,直接面向高数据率的广播格式,提供了有效的隔行视频信号编码算法,支持宽范围的比特率。

MPEG-4 为多媒体应用标准(1999 年发布),它在 MPEG-2 基础上作了很大的扩充,主要目标是多媒体应用(主要侧重于对多媒体信息内容的访问)。MPEG-4 标准是一种适用在低带宽下进行信息交换的音视频处理技术,可以解决 MPEG-1 和 MPEG-2 在码率很低时的图像失真问题,它能够动态侦测图像各个区域变化,利用很窄的带宽通过帧重建技术压缩和传输数据,从

而能以最少的数据获得最佳的图像质量,因此非常适用于动态图像、互联网、实时多媒体监控、移动多媒体通信、可视游戏、DVD 上的交互多媒体应用等应用。它的主要特点是对图像中的内容进行编码,其具体的编码对象就是图像中的音频(Audio)和视频(Video),称为 AV 对象。高效率地编码、组织、存储、传输 AV 对象,是 MPEG-4 标准的基本内容。MPEG-4 具有开放的编码系统,可以随时加入新的有效算法,为多媒体数据提供了一个更为广阔的平台,它定义的是一种格式、一种构架,而不是具体的算法。

### 1.2.2 大规模集成电路发展历史及其现状

自 1958 年美国德克萨斯仪器公司(TI)、仙童公司独立发明了第一块半导体集成电路以来,经过四十多年的发展,微电子产业已成为当代高新技术产业群的核心和维护国家主权、保障国家安全的战略性产业。在市场需求与技术进步的不断推动下,微电子技术已由初始的小规模集成(SSI)发展到当前的巨大规模集成,其影响已渗透到国民经济和社会的各个领域<sup>[13]</sup>。由于微电子产业发展的拉动,以计算机、通信、网络为代表的信息产业得以迅猛发展,使得美国经济保持持续增长,日本发展成为世界第二大经济大国,韩国、新加坡与我国的台湾地区也取得经济飞速发展的奇迹<sup>[14]</sup>。

#### 1.2.2.1 IC 技术的发展及重要地位

自 20 世纪 60 年代中期集成电路产业在工业发达国家形成以来,集成电路技术的发展一直遵循著名的摩尔定律,即:每 18 个月芯片集成度大体增长一倍<sup>[15]</sup>。随着集成电路技术的不断发展,集成电路产品经历了从传统的板上系统(System-on-board)到片上系统 SOC(System-on-a-chip)的过程。集成电路产业为适应技术的发展和市场需求,产业结构经历了三次大的变革,直接导致了集成电路设计业的形成。

##### (1) 第一次变革

以加工制造为主导的 IC 产业发展的初级阶段。上个世纪 70 年代,集成电路的主流产品是微处理器、存储器以及标准通用逻辑电路。这一时期 IC 制造商在 IC 市场中充当主要角色,IC 设计只作为附属部门而存在,这时的 IC 设计和半导体工艺密切相关。70 年代后期,以美国为代表的半导体厂商为了降低成本,将所需劳力较多而技术含量又较低的封装与测试工序分离出来,转移到远离本土的欠发达国家或地区生产加工(如韩国、台湾地区),集成电路产业开始逐步形成封装业单独分列的局面。



## (2) 第二次变革

Foundry 公司与 IC 设计公司的崛起。上个世纪 80 年代, 集成电路的主流产品为微处理器 (MPU)、微控制器 (MCU) 及专用 IC (ASIC)。这时, 无生产线的 IC 设计公司 (Fabless) 与标准工艺加工线 (Foundry) 相结合的方式开始成为集成电路产业发展的新模式。在这一阶段, 随着微处理器和 PC 机的广泛应用和普及 (特别是在通信、工业控制、消费电子等领域), IC 产业已开始进入以客户为导向的阶段。为了改善系统的速度和简化程序, 各种硬件结构的专用集成电路如可编程逻辑器件 (FPGA)、标准单元、全定制电路等应运而生。此外, 随着 EDA (Electronic Design Automation) 工具的发展, PCB (Printed Circuit Board) 设计方法引入到 IC 设计之中, 设计开始进入抽象化阶段, 使设计过程可以独立于生产工艺而存在。有远见的整机厂商和创业者包括风险投资基金纷纷开始成立专业设计公司和 IC 设计部门, 从而一种无生产线的集成电路设计公司或设计部门得到了迅速的发展, 同时也带动了 Foundry 的崛起。

## (3) 第三次变革

四业分离的 IC 产业。90 年代, 随着 Internet 的兴起, IC 产业跨入以竞争为导向的高级阶段, 国际竞争由原来的资源竞争、价格竞争转向人才知识竞争、密集资本竞争。人们逐渐认识到, 越来越庞大的集成电路产业体系并不有利于整个 IC 产业发展, 分才能精, 整合才成优势。于是 IC 产业结构向高度专业化转化成为一种趋势, 开始形成了设计业、制造业、封装业、测试业独立成行的局面。

当今世界经济已从工业化进入信息化的发展阶段, 微电子技术是高科技和信息产业的核心技术, 成为当前新经济时代的基础产业。它在国民经济、国防建设以及现代信息化社会中起着极其重要的战略意义。国际微电子发展的趋势是集成电路的特征尺寸将继续缩小, 集成电路 (IC) 将发展为系统芯片 (SOC)。微电子技术和其他学科相结合将产生很多新的学科生长点, 与其他产业结合成为经济增长点。芯片是信息时代最重要的基础产品之一, 如果把石油比作传统工业“血液”的话, 那么芯片则是信息时代 IT 产业“心脏”和“大脑”。无论是小到日常生活的电视机、VCD 机、洗衣机、移动电话、计算机等家用消费品, 还是大到传统工业的各类数控机床和国防工业的导弹、卫星、火箭、军舰等都离不开这小小的芯片<sup>[4]</sup>。

### 1.2.2.2 我国集成电路产业的发展现状

我国的微电子产业至今已整整走过了 40 多个春秋,近 10 多年来,重点项目建设和广泛的国际合作,使产业的技术水平、生产规模和应用领域迅速提高和扩大,实力大为增强,然而与同期内突飞猛进、日新月异的世界微电子产业相比,我国微电子产业的技术水平还比较落后,产业内部结构不合理,设计企业规模太小,制造企业难以开展规模化经营。此外,前些年我国的电子产品虽然发展很快,但几乎所有国产大型家用电器的关键芯片、国产手机的核心芯片、国产计算机的主要芯片,大都不是国产的。其中的原因是,我国在集成电路领域里最能体现核心竞争力的 IC 设计相对十分薄弱。尽管上个世纪九十年代以来,我国集成电路设计业得到了长足发展,产业化规模已初步形成,但是设计人才特别是高级人才的极度匮乏的问题日益突出。抓住当前世界微电子产业发展的机遇,在世界范围内积极争取技术、资金和管理要素投入中国,是我国信息产业发展的当务之急<sup>[17-19]</sup>。我国政府非常重视 IC 设计产业的潜力与未来发展,将 IC 设计产业列为国家高科技重点发展产业,给予各种优惠等政策,在 2001 年发布的“十八号文件”中,也特别强调加速包括 IC 设计与软件产业发展,并号召在美国硅谷发展的人才回国创业,以丰富 IC 产业的实力。

众所周知,集成电路芯片设计是整个集成电路产业链中最关键的一环,是体现芯片创意、知识产权与专利技术的最重要的载体。在我国加入 WTO 后,以集成电路为核心的电子产品行业必然面临新一轮更加激烈的竞争,目前我国许多整机企业主要还是以组装形式组织生产,关键技术主要依赖国外公司提供的芯片和产品设计。面对这种尴尬的局面,着手进行核心芯片及其应用产品的自主开发,做到从芯片开发着手,拥有自主的知识产权,从而真正掌握市场竞争的主动权,已是刻不容缓。中国集成电路设计企业目前设计研发产品主要分布在以下领域:EDA 软件工具、IC 卡专用集成电路(电话 IC 卡、身份证卡芯片等)、家用电器类专用集成电路(MCU 芯片、MPEG-2 芯片、DVD 芯片等)、计算机通讯领域芯片(手机 SIM 卡,3G 芯片、嵌入式 CPU、数字视频处理芯片、MPEG-4 芯片、无线局域网芯片等)。其中重点开发产品应该有数字视音频产品,包括数字电视(DTV),DVD/SVCD/VCD,机顶盒(STB)、高清晰度电视(HDTV);网络产品:主要是指 Internet 接入和接出设备;通讯产品,如 GSM,CDMA 手机、Bluetooth 产品等。

### 1.2.2.3 我国数字视频编码器芯片的发展现状

目前,由于西方国家的出口限制,我国在芯片技术上受制于人的局面尚未扭转,全国年产 7000 多万台电视中所使用的核心视频处理芯片全部来自进口。在上海,美国 Trident 公司成立了独资的泰鼎(上海)有限公司,该公司设计研发的 DPTV 视频芯片在中国市场得到了广泛应用,康佳、长虹、TCL、海信、厦华、创维等国内主要彩电生产厂商均采用 DPTV 芯片。更有甚者的是,来自美国、日本等国家的视频压缩编码芯片在中国市场上的占有率竟高达 80% 左右<sup>[9]</sup>。面对这种尴尬的局面,着手进行核心芯片及其应用产品的自主开发,做到从芯片开发着手,拥有自主知识产权,从而真正掌握市场竞争的主动权,已是刻不容缓。青岛海尔集团,2000 年在北京设立了北京海尔集成电路有限公司,揭开了国内企业冲破外国芯片垄断的序幕,该公司在 2001 年成立仅一年的时间内就推出了国内第一个可商品化生产的“爱国者 1 号” MPEG-2 视频芯片,接着又推出“爱国者 2 号”,但对于巨大的国内市场,只能是杯水车薪。2002 年 7 月,在成都诞生了国内第一个具有自主知识产权的高清晰度模数视频显示芯片,但毕竟仅是在样品生产阶段,距离大规模的商品化生产还有很长的路要走。2005 年 10 月中科集成电路公司推出了具有中国自主知识产权 AVS 标准的视频编码芯片——凤芯 2 号,但是在早已被 MPEG、JPEG 和 H.26x 等标准所占据的数字视频编码芯片市场上,凤芯 2 号的命运仍然是个未知数。

综上所述,数字视频相关多媒体芯片的设计开发工作,是自主发展我国电子行业的必由之路,也是振兴我国消费电子产业的关键一步,是真正实现数字视频产业化的实质和根本。它对于打破国外的技术垄断和振兴我国的民族产业都具有极为重要的现实意义。因此从某种意义上来说,数字视频产业发展的意义在于它将引发一场信息技术的革命,从而推动我国电子信息产业的更新换代。

### 1.3 数字视频编码技术在现代远程教育方面的应用

随着教育的不断深入,教育现代化的不断推进,传统的教学媒体:如黑板、教科书承载信息的种类和更新速度都十分有限,远远不能满足现代教学的需要。随着电子技术的发展,出现了大量的媒体展现形式<sup>[10]</sup>。如幻灯、投影、录音、录像等。这些媒体虽大大提高了信息的承载能力、并在提高教学质量和推动现代化教育技术方面发挥了较大作用。但这些媒体在展现形式上仍存在一定的弱点。幻灯和投影不易表现运动的事物;录像缺乏灵活的交

互功能; CAI 有取众家之长的优势, 可将多种媒体信息集成于一体, 并有灵活的交互功能, 代表了教学媒体的发展方向。然而, 这种教学形式更适应于多媒体教室内的教学组织形式; 或者在局域网、校园网内的网络教育形式。现代远程网上教育, 由于信息传输量大和 Internet 资源有限, 现阶段实现广域实时在线的交互式教学还存在一定的困难。因而, 在保证教学效果和质量的前提下, 选用合适的视频压缩技术方案, 有助于发展网络教育, 保证网络教育质量。

多媒体信息主要包括图像、声音和文本三大类, 其中视频、音频等信号的信息量是非常大的。而且这些信息的表达方式、输入、输出的要求也各不相同, 因此在多媒体通信中, 对这些数据进行有效的表达和适当处理是非常重要的<sup>[21]</sup>。

欧洲和我国使用的彩色视频图像标准为 PAL 制式, 其视频带宽为 5MHz, 帧速率为 25 帧/s, 样本宽是 24 bit, 采样频率至少为 10 MHz, 因而存储一帧数字化的 PAL 制式的视频图像需要的空间为 1.2 MB, 存储 1 秒 PAL 制式的视频图像需要的空间为 30MB。

由此可见, 存储数字化的多媒体信息要求的存储空间是很大的。再从数据传输的要求看, 多媒体通信系统对海量信息的处理和传送也受到计算机自身和传输线路等因素的限制。因而, 多媒体信息传送而临的最大难题之一是海量数据的问题, 数据压缩技术也就成了多媒体通信领域的关键技术。

MPEG 应运而生, 它是 ISO 与 IEC 于 1988 年联合成立的, 致力于运动图像及其伴音编码标准化工作。它包括 MPEG 系统, 即 MPEG 视频及 MPEG 音频, 它一共包括 MPEG-1、MPEG-2、MPEG-4、MPEG-7、MPEG-21 五个版本, 前三个版本是规定了具体的信息压缩编码算法的标准。后两个则仅仅是规定了多媒体内容的描述, 它们的主要目的是对多媒体信息内容进行有效的检索。

MPEG 系列中的 MPEG-2 标准与其他四个标准相比, 在现代远程教学这一特定应用中具有非常显著的技术特点: MPEG-2 制定于 1994 年, 设计目标是高级工业标准的图像质量以及更高的传输率。MPEG-2 所能提供的每秒 30 帧播放 1MBit~10MBit 间的数据, 在 NTSC 制式下的分辨率可达 720x486, MPEG-2 能够提供广播级的视像和 CD 级的音质。MPEG-2 的音频编码可提供左右中及两个环绕声道, 以及一个加重低音声道和多达七个伴音声道。MPEG-2 的另一特点是, 可提供一个较广范围的可变压缩比, 以适应不同的画面质量、存储容量以及带宽的要求。使用 MPEG-2 的压缩算法压缩一部

120 分钟长的电影可以到压缩到 4 到 8 GB 的大小图象质量等性能方面的指标 MPEG-1 是没得比的。此外, MPEG-2 特别适用于广播级的数字电视的编码和传送, 被认定为 SDTV 和 HDTV 的编码标准。MPEG-2 还专门规定了多路节目的复用分接方式。与 MPEG-4 相比, MPEG-2 虽然是 MPEG-4 的子集, MPEG-4 向下兼容 MPEG-2。

但是, 两者最显著的不同就是, MPEG-4 引入了 VOP (Video Object Plane) 视频对象平面的概念来实现基于内容的表示, 它通过对原始图像帧的内容分割, 对每个 VO 和背景 Sprite 分别编码和传输, 从而有效地降低了码率, 但是值得注意的是, 由于 VOP 引入, 大大的增加编/解码器端的运算复杂度和运算时间, 造成了在硬件实现上的困难和实现平台结构成本的大幅度增加, 而作为“山村多通道网络教学环境”的配套设施, 应用于山区、偏远农村现代远程教育的运动图像编码技术, 使用 MPEG-4 作为视频编码标准却是得不偿失。

由此可见, 在要求低成本、高质量、多信道传输的山区、偏远农村现代远程教学中, 使用 MPEG-2 标准作为视频压缩算法的蓝本, 是一种相对经济、明智的选择。

## 1.4 题目来源、主要研究内容以及章节安排

### 1.4.1 题目来源及主要研究内容

本文来源于国家 863 课题——“缩小数字鸿沟—西部行动”《基于国产软硬件的多通道网络教育关键技术与应用研究》(课题编号: 2003AA116060)。目的是研究和开发研制一种适用于山区、偏远农村现代远程教育的经济型 MPEG-2 运动图像编码的 IP 核, 该软核的设计应用了当前最新的微电子技术、电路设计自动化技术发展成果, 具有低价格、功耗低、简单、可靠、开放的特点。并且, 芯片设计中的二维 DCT、量化/逆量化、可变字长编码和运动估计模块, 还可以作为独立的软 IP 核, 根据需要略加修改就可以应用到其他含相关算法的图像/视频压缩芯片的设计中去。MPEG-2 运动图像编码器包括离散余弦变换、量化/逆量化、Zig-Zag 扫描、可变字长编码、运动估计等几个部分, 每一部分实现的优劣都直接关系到图像的实时解码性能。

本文主要研究内容包括:

针对山区、偏远农村现代远程教育这一特定应用领域, 对 MPEG-2 运动

图像编码系统进行研究,确定用 FPGA 来开发低成本、高质量的 MPEG-2 运动图像编码器;在编码器的块处理模块设计中,对于二维 DCT 模块和可变字长编码模块通过改进算法、优化结构和充分利用 FPGA 自身的资源特性等方法,有效地降低了模块关键路径上的时延和逻辑资源消耗量,使得设计在速度和资源两方面均达到了较优的状态;在运动估计模块的设计中,提出了一种高度并行和多流水线处理的全搜索块匹配运动估计硬件结构,通过改进搜索窗缓存、PE 运算阵列使得在增加少量的芯片面积条件下达到了每秒能处理 56000 个宏块的速度,大大高于 MPEG-2 MP@ML 所规定的 40500 宏块/秒处理速度,满足了实时视频的运动估计编码的要求。

#### 1.4.2 本文章节安排

论文全文共分五部分:

第一部分为绪论,简单介绍了运动图像压缩技术及相关标准,国内外集成电路技术发展、现状及其运动图像编码在现代远程教育方面的应用。

第二部分介绍了 MPEG-2 图像压缩的基本原理和方法,详细论述了运动图像编码中所使用的一些关键技术,并确定了 MPEG-2 运动图像编码器总体设计方案,以及期望达到的设计性能指标。

第三部分重点论述了 MPEG-2 运动图像编码器中块处理单元设计,分别阐述了二维离散余弦变换、量化及逆量化、Zig-Zag 扫描、可变字长编码编码等模块单元的硬件实现及其仿真。

第四部分重点论述了 MPEG-2 运动图像编码器中全搜索运动估计模块单元计的硬件实现及其仿真。

第五部分对本文工作进行了总结并对未来的工作做出了展望。

---

## 2 MPEG-2 运动图像编码的基本原理与方法

### 2.1 MPEG-2 标准简介

MPEG-2 标准 (ISO/IEC11172) 于 1994 年 11 月正式公布。在数字电视广播 (DVB), 数字视频光盘 (DVD) 以及高清晰度电视 (HDTV) 等方面成为视频音频压缩编码的标准。一方面, 与 MPEG-1 相同, MPEG-2 仍然以运动估计补偿和 DCT 变换混合编码为核心, 其原理、码流结构与 MPEG-1 相似, 原始视频数据分为图像层、块组层、宏块层、块层四层; 另一方面 MPEG-2 是 MPEG-1 的进一步发展, 其应用目标更为广泛, 适用于通过卫星、电缆及地面网、各类计算机网络的传输以及磁带、磁盘、光盘记录等数字视频及音频的要求, 为了力求满足这些广泛的需要。MPEG-2 从分辨率和压缩编码方法复杂度两方面对运动图像的压缩进行不同的分类和不同分档次。与此同时, MPEG-2 也吸收了 H.26x 等其它一些国际标准中有效合理的部分, 如: 半像素精度的运动估计, PB 帧预测等。MPEG-2 共有 I 帧, P 帧, B 帧三种帧模式和帧内编码 (INTRA)、帧间编码 (INTER) 两种编码模式<sup>[7]</sup>。

MPEG-2 全部标准共有九部分, 其中前三部分是基本部分:

第一部分是系统部分, 规定了形成数字流的语法, 描述了数据流包层结构及包头文件中为解码器处理提供的各种信息。数据流分成节目码流和传送码流两种不同的数据流类型, 节目码流类似于 MPEG-1 的数据流结构, 数据包长不固定, 用于误码较小的环境; 传送码流包长为 188 字节适用于误码较大的环境。

第二部分是视频压缩编码部分, 规定了以 DCT (离散余弦变换) 及运动估值及补偿组成的混合编码以及分层分级编码的各种扩充编码方法。

第三部分是音频编码部分。

其他各部分为:

四、测试部分; 五、软件模拟; 六、数字存储媒体命令和控制 (DSM-CC) 协议; 七、多通道音频编码; 八、10 比特视频取样编码 (已停止制订); 九、传送码流的实时接口; 十、DSM-CC 测试标准。





## 2.2 MPEG-2 系统模型

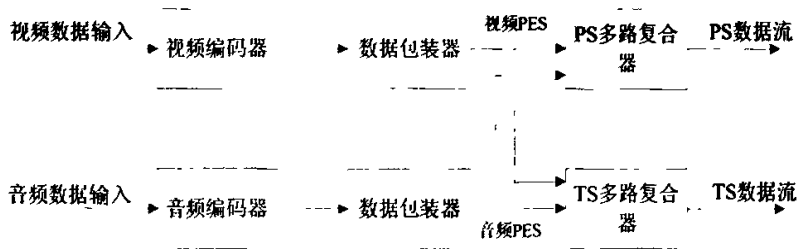


图 2-2 MPEG-2 系统模型

Fig. 2-2 MPEG-2 System Model

MPEG-2 的系统模型如图 2-2 所示。MPEG-2 标准的系统部分主要规定了电视图像数据、声音数据和其他数据的组合，在这些数据组合成一个或者多个适合于存储或者传输的基本数据流。数据流有两种形式，一种成为程序数据流 PS (Program Stream)，另外成为传输数据流 TS (Transport Stream)。程序数据是组合一个或者多个规范化的即包化基本数据流 PES (Pack- etised Elementary Streams) 而生成的一种数据流，可以是固定码率也可以是可变的码率，其数值在系统时钟参考 (SCR) 中定义，用在出现错误相对比较少的环境下，适合使用软件处理的应用；传输数据率也是组合一个或者多个 PES 而生成的一种数据流，其码率可以是固定的或是不固定的，其值在节目时钟参数 (PCR) 中定义，它用在出现错误相对较多的环境下，例如在有损失或者有噪声的系统中<sup>[9]</sup>。

MPEG-2 视频编码标准的视频压缩编码部分是应数字媒体存储媒体、电视广播以及通信等应用领域对于运动图像以及伴音的通用编码方法的要求日益增大而产生的，使运动视频数据作为一种计算机可处理的数据形式，并且可以存储在各种存储媒体中，可以在现在或未来的网络上发送、接受，并且可以在现存或未来的广播信道上传播。

## 2.3 MPEG-2 视频编码基本原理与方法

MPEG-2 视频编码标准是一等级的系列，按编码图像的分辨率分成四个“级 (Levels)”；按所用的编码工具的集合分成五个“类 (Profiles)”。“级”

与“类”的若干组合构成 MPEG-2 视频编码标准在某种特定应用下的子集；对某一输入格式的图像，采用特定集合的压缩编码工具，产生规定速率范围的编码码流，在 20 种可能的组合种，目前有 11 种是已获通过的，成为 MPEG-2 适用点。

众所周知，当前模拟电视存在着 PAL、NTSC 和 SECAM 三大制式并存的问题，因此，数字电视的输入格式标准试图将这三种制式统一起来，形成一种统一的数字演播室标准，这个标准就是 CCTR601，现 ITU-RRec.BT601 标准。MPEG-2 中的四个输入图像格式“级”都是基于这个标准的。低级（Low Level）的输入格式的像素是 ITU-RRec.BT601 格式的  $1/4$ ，即  $352 \times 240 \times 30$ （代表图像帧频为每秒 30 帧，每帧图像的有效扫描行数为 240 行，每行的有效像素为 352 个），或  $352 \times 288 \times 25$ 。低级之上的主级（Main Level）的输入图像格式完全符合 ITU-RRec.BT601 格式，即  $720 \times 480 \times 30$  或  $720 \times 576 \times 25$ 。主级之上为 HDTV 范围，基本上为 ITU-RRec.BT601 格式的 4 倍，其中 1440 高级（High-1440 Level）的图像宽高比 4: 3，格式为  $1440 \times 1080 \times 30$ ，高级（High-1440 Level）的图像宽高比为 16: 9，格式  $1920 \times 1080 \times 30$ 。

在 MPEG-2 的五个“类”中，较高的“类”意味着采用较多的编码工具集，对编码图像进行更精细的处理，在相同位率下将得到较好的图像质量，当然实现的代价也较大。较高类编码除使用较低类的编码工具外，还使用了有些较低级类中没有使用的附加工具，因此，较高类的解码器除了能解码用本类方法编码的图像外，也能解码较低类方法编码的图像，即 MPEG-2 的“类”之间具有向后兼容性。简单类（Simple Profile）使用最少的编码工具。主类（Main Profile）除使用所有简单类的编码工具外，还加入了一种双向预测的方法。信噪比可分级类（SNR Scalable Profile）和空间可分级类（Spatially Scalable Profile）提供了一种多级广播的方式，将图像的编码信息分为基本信息层和一个或多个次要信息层。基本信息层包括对图像解码至关重要的信息，解码器根据基本信息即可以进行解码，但图像的质量较差。次要信息层中包括图像的细节。广播时对基本信息层加以较强的保护，使其具有较强的抗干扰能力。这样，在距离较近、接受条件好的情况下，可以同时受到基本信息和次要信息，恢复出高质量的图像；而在距离较远、接受条件差的条件下，仍能接受到基本信息，恢复出图像，不致造解码中断。高级类（High Profile）实际上应用于位率更高，要求更高的图像，此外，前四个类在处理 Y、U、V 时逐行处理色差信号的，高级类还提供同时处理色差信号的可能性。

**MPEG-2 图像压缩的原理**是利用图像中的两种特性：**空间相关性**和**时间相关性**。一帧图像内的任何一个场景都是由若干像素点构成的，因此一个像素通常与它周围的某些像素在亮度和色度上存在一定的关系，这种关系叫作**空间相关性**；一个节目的一个情节常常由若干帧连续图像组成的图像序列构成，一个图像序列中前后帧图像间也存在一定的关系，这种关系叫**时间相关性**。两种相关性使得图像中存在着大量的冗余信息<sup>[10]</sup>。去掉这些冗余的相关信息，可以大大节省传输频带。一个好的压缩编码方案就是能够最大限度地去除图像中冗余的信息。

为了使得视频信号能够更有效的压缩，往往综合使用各种不同的压缩方法，图 2-3 就是一个通用的活动图像编码器的典型框架。

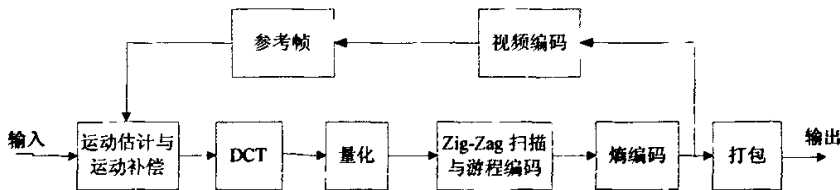


图 2-3 MPEG-2 视频编码器框图

Fig. 2-3 MPEG-2 video encoder diagram

### (1) MPEG 亮度和色度信号的形成

输入的复合视频信号可以是三种电视制式的任何一种。视频压缩编码器的内部有制式检测器，从而控制电子开关对任何制式进行解码，即分离出三基色信号  $R$ 、 $G$ 、 $B$ ，之后形成色差信号  $R-Y$  和  $B-Y$ ，接着应该转换成 MPEG 亮度和色度信号  $Y$ 、 $Cb$ 、 $Cr$ 。其亮度和色度信号的转化公式见式(2-1)：

$$\begin{aligned}
 Y &= 0.30R + 0.60G + 0.10B \\
 Cr &= 0.625(R - Y) + 0.50 \\
 Cb &= 0.5(R - Y) + 0.50
 \end{aligned}
 \tag{2-1}$$

由于色差信号  $R-Y$  和  $B-Y$  经常有正有负，在处理前先向上平移 50%，即各加上 0.5，称为黑电平移动<sup>[11]</sup>。另外，在  $Cb$ 、 $Cr$  的表达式中加的系数是为了色差信号的  $Cb$ 、 $Cr$  的系数值在 0~1 之间，使得 A/D 输出的 8bit 量化的数字信号落在 0~255 之间（在 DCT 前还要下移 0.5，即减去 128，使色差信号恢复到有正有负的状态，以降低传输比特率）。最后就可以按 ITUR.601 标准进行取样和量化，实现模拟电视信号的数字化处理。

(2) MPEG-2 的几种宏块结构

对于分量编码而言，模拟视频信号经数字化后，亮度和色度信号采样点先分别被分割成  $8 \times 8$  的像素块，即为像块。同一个区域的若干像块（如 Y 有 4 个，Cb、Cr 各一个）构成一个宏块。MPEG-2 定义了三种宏块结构格式，如图 2-4 所示，即 4: 2: 0 宏块、4: 2: 2 宏块和 4: 4: 4 宏块，分别包含 6、8、12 个像块。

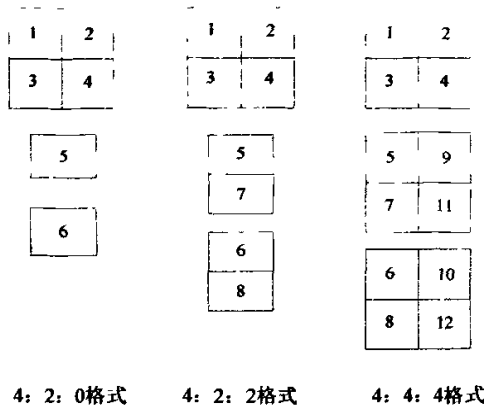


图 2-4      MPEG-2 三种不同的宏块格式

Fig. 2-4      Three different Macroblocks format in MPEG-2

进行像块分割时，MPEG-2 允许逐行扫描和隔行扫描两种扫描格式。如果只采用逐行分割方式，对于 DCT 和运动补偿可能存在某些不良的影响。这是因为在隔行扫描情况下，若有运动发生，由于时间相邻两场之间有一场的延时，像素之间的位移可能很大，于是帧内相邻行的空间相关性将会大大下降，而场内相邻行的相关性可能大于帧内相邻行间的相关性，这时做基于场的处理效果会比基于帧的更好。但若对所有隔行扫描图像都做基于场的处理，则在运动非常小时，由于帧内的两相邻行基本上是没有位移的，而场内相邻行空间上的相似性显然比帧内相邻行间隔远了一倍，会出现帧内相邻行相关性远大于场内相邻行相关性，这是做基于帧的处理效果会比基于场的处理效果更好。

可见，MPEG-2 有两种图像格式—帧图像格式和场图像格式，前者是以整个帧作为考虑对象，后者是以场作为独立的对象来考虑，MPEG-2 有基于场和基于帧的 DCT 编码，以及基于帧的或者基于场的运动补偿，或帧预测

和场预测。帧预测是利用前面解码出来的帧数据对当前的帧做独立的预测，场预测是利用前面解码出来的场数据对当前场做独立的预测。

此外，还有其他两种图像预测和运动补偿方式，即  $16 \times 8$  格式运动补偿和双场预测。前者是 把一个宏块分成两个半块（顶部一半和底部一半），称之为子宏块，它仅用于场图像格式，所以两个子宏块以场方式组织，每块只包括同一场的 8 行；后者是一种基于场预测，但又与之不同的特殊预测方式，它只能用于隔行扫描，而且用于不采用 B 帧的编码器结构。由于它所需要传输的运动矢量比一般的基于场预测的方式要少些，因此这种方式在改进低延时应用时对提高编码器效率特别有用。

### （3）帧/场编码判决和帧内/帧间编码判决

取样、量化后的亮度和色差信号分别形成  $8 \times 8$  的像块，并构成某种结构的宏块。如果采用 4: 2: 2 的结构格式，则安排顺序如图 2-5 所示。其中， $C_j$  均为  $8 \times 8$  数据单元。像块是 DCT 的处理单元，在做 DCT 之前，要做帧的行间和场的行间的相关系数的计算，如果帧行的相关系数大于或等于场行的相关系数，就选择帧 DCT 编码，否则就选场 DCT 编码。这样就可以使 DCT 相对系数较大的信号做处理，得到较高得压缩比。

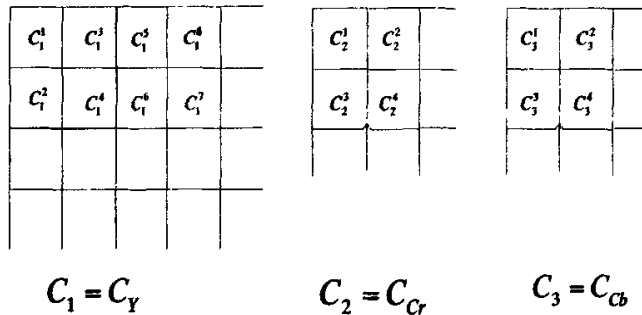


图 2-5 采用 4: 2: 2 格式的宏块排序

Fig. 2-5 The Macroblock scheduling for 4: 2: 2 format

由于相邻帧得的相关性很强，因此允许传送图像时候，有几帧可以不传送整帧的信息。在 MPEG-2 中，图像被分成了 3 种编码类型：帧内编码的图像，称为 I 帧，一定要传；前向预测编码的图像，称为 P 帧；双向预测编码的图像，称为 B 帧；其中后两种帧属于帧间编码，只需要传送两帧之间的差

值, P 帧以前一个 I 帧为预测帧进行编码, P 帧有时也可以从前一个 P 帧预测得到, 但数目不能太多; B 帧是对相邻最近的 I 帧或 P 帧做双向预测进行编码, 在 I 帧和 P 帧之间可以插入若干 B 帧, 一般为 2 个, 因此对于前帧相关性不大的当前帧, 可对整个帧进行 DCT, 并进行帧内编码; 但若当前帧与前一帧的相关性大, 则可以对这两帧的差值进行 DCT, 并进行帧间编码。

为了自动决定是对输入像块的样值进行 DCT, 还是对前后两个像素的样值差值进行 DCT, 以及宏块 (MB) 应采用帧内还是帧间编码, 应先进行帧内/帧间编码的判决。判决的方法是将前帧图像存于帧存储器, 当后帧来临时, 比较它们的相关性, 相关性弱的则采用帧内编码, 相关性强的采用帧间编码。在以下帧图像的帧预测 (它的宏块数据由两场数据共同组成) 为例, 说明帧内/帧间编码判决方法。设前帧、后帧 MB 的亮度像素值分别用  $P(x, y)$  和  $G(x, y)$  表示, 在 MB 内亮度信号有 4 个  $8 \times 8$  的方块, 即它们构成  $16 \times 16$  方块, 共 256 个像素。前帧 MB 的亮度信号方差用 VAROR 表示, 它表示图像变化的缓急程度, 其计算公式见式 (2-2) 所示:

$$VAROR = \frac{1}{256} \sum_{y=1}^{16} \sum_{x=1}^{16} [G(x, y)]^2 - \left[ \frac{1}{256} \sum_{y=1}^{16} \sum_{x=1}^{16} P(x, y) \right]^2 \quad (2-2)$$

前后帧因时间差而引起像素差, 即时间预测变动用 VAR 表示, 它表示前后帧对应像素值之差的均方值, 说明前后帧像素值变动所导致的平均能量变动, 其计算公式见式 (2-3) 所示:

$$VAR = \frac{1}{256} \sum_{y=1}^{16} \sum_{x=1}^{16} [G(x, y) - P(x, y)]^2 \quad (2-3)$$

像素值在 0~255 范围内, VAR 值越小, 相关性越大, 但是还是要考虑图像方差, 反差大的图像, VAR 的值也相应增大, 根据 VAROR、VAR 的值, 可以确定式采用帧内编码还帧间编码:

当  $VAR \leq 64$  时, 相关性大, 采用帧间模式;

当  $VAR \geq 64$  时,  $VAROR \geq VAR$  时, 采用帧间模式;

当  $VAR > 64$  时,  $VAROR < VAR$  时, 采用帧内模式。

#### (4) 运动估计和运动补偿

在帧间编码时, 需要传递前后帧宏块的差值, 此差值不是前后帧对应像素  $P(x, y)$  和  $G(x, y)$  的差值, 而是在前帧内, 对应于后帧的宏块位置附近区域中, 搜索最匹配的宏块, 也就是要寻找最相似的宏块 (当然也有可能找到完全相同的宏块), 并记下这两个区域在水平和垂直方向上的位移 (即运动矢量), 然后传送这两个宏块的差值 (完全相同的宏块, 则差值为零) 以及

运动矢量。

运动估计有各种不同的方法，其中最重要的是块匹配法和像素递归法。前者是把一幅图像分割成固定的  $M \times M$  矩阵块，并认为块内每个像素具有同一位移矢量，并用某种块匹配准则来计算这两个子块之间的运动矢量（该过程称为运动估计）。这种算法工作量十分巨大，容易发生图像编码的方块效应；后者可以提高位移矢量估计的精确度，它可对每一个像素分别估计其位移，因而更接近实际情况，而且可以增加位移矢量的测量范围，因而可以减少计算量。但理论和实践结果表明，对于不同的视频图像信号，块匹配算法具有较好的适应性，所以多数的情况下采用块匹配算法。

如图 2-6 所示，可以用  $f_k(m,n)$ 、 $f_{k-1}(m,n)$  分别表示当前帧和前一帧的信号值，搜索的区域  $R$  的面积大小为  $R = (M + 2d_m)(N + 2d_m)$ 。块匹配准则有很多种，如互相关函数法、均方误差法、帧差平均值绝对值法和最大误差最小函数法等等。以帧差平均值绝对值法为例，帧差平均值绝对值函数为：

$$MAE(i, j) = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N |f_k(m, n) - f_{k-1}(m+i, n+j)| \quad (2-4)$$

显然，绝对值最小的那个子块是与当前子块最匹配的子块，所对应的  $i$ 、 $j$  也就是运动矢量。

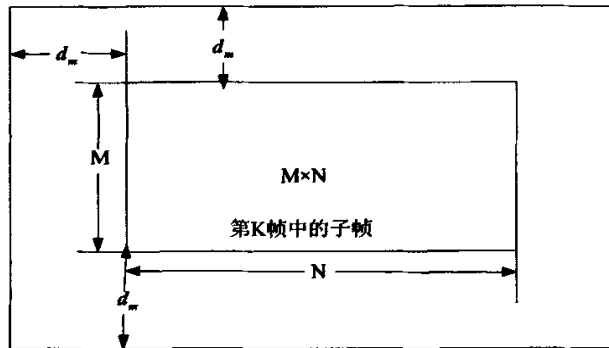


图 2-6 块匹配算法和像素递归算法

Fig. 2-6 Block-matching and pixel recursive algorithm

块匹配准则是计算运动矢量的依据，不同的准则的运动估计复杂度不同，运动补偿的效果也不同。运动补偿的复杂性也取决于运动估计过程的复杂性，比较而言，上述的帧差平均值绝对值准则由于计算量相对较小，因此

目前最常用<sup>[14]</sup>。

#### (5) DCT、量化、Zig-Zag 扫描和可变字长编码

在帧内模式下，对输入帧的数据进行 DCT；在帧间模式下，输入的帧数据与经过运动补偿的前帧信号相减，对差值数据进行 DCT。DCT 之后要进行量化处理。为了使量化后的二维数据序列转化成一维的数据序列，还要进行 Zig-Zag 扫描。Zig-Zag 扫描输出的一维的数据序列再进行熵编码，然后输出。

- ①DCT DCT 是一种准最佳的变换。先将帧数据分成  $8 \times 8$  的矩阵子块，然后以 MB 为单位，对  $8 \times 8$  的子块逐一进行如下的 DCT：

$$\begin{cases} F(u,v) = \frac{2}{N} C(u)C(v) \sum_{x=0}^7 \sum_{y=0}^7 f(x,y) \cos \left[ \frac{(2x+1)u\pi}{2N} \right] \\ f(x,y) = \frac{2}{N} \sum_{u=0}^7 \sum_{v=0}^7 C(u)C(v) F(u,v) \cos \left[ \frac{(2x+1)v\pi}{2N} \right] \end{cases} \quad (2-5)$$

式中，N 是子块的水平和垂直位置方向的像素数，一般  $N=8 \times 8$  的二维数据块经过 DCT 后变成  $8 \times 8$  个变换系数，这些系数都具有明确的物理意义。如当  $U=0, V=0$  时， $F(0, 0)$  是原来 64 个样值的平均值，相当于直流分量；随着 U、V 的增加，相应系数分别代表逐步增加的水平空间频率和垂直空间频率分量的大小。

- ②量化 严格地说，DCT 本身并不能进行码率压缩，因为 64 个样值变换后仍然是 64 个系数甚至所需要的比特率增加了。因此，需要进一步的处理，即量化处理，其目的是对 DCT 处理结果（即 DCT 系数）进行压缩。这一过程实际上是用降低 DCT 系数精度的方法去除不必要的 DCT 系数，从而降低传输比特率。由于在解码端的反量化之后，不能恢复原来的 DCT 系数，因此这种处理对图像有损伤，属于有损编码范畴。

量化以人眼的生理特性为依据。人眼对低频分量和亮度信号比较敏感，而对高频分量和色度信号不太敏感。因此，亮度和色度信号、低频和高频分量可采用不同的量化。实际实现时，采用量化表除以 DCT 系数，所得的值用四舍五入取整。亮度、色度各有一个量化表，每个量化表有  $8 \times 8$  个数值，也称为量化步长，这些数值是通过实验确定的。实验方法是，有较低的数值开始，比较输入图像与经量化、去量化后的输出图像的区别，逐步提高量化步长，直到主观感觉发现差别为止，由此获得感觉门限。达到该门限所得的量化步长就是



实际使用的量化表中的数值，它是使压缩效果达到最好的量化系数。由于 DCT 系数左上方对应于图像的低频分量，右下方对应于图像的高频分量，故量化步长左上方小、右下方大。这样，经量化之后所得的数据一般都集中在左上方，右下方高频系数多数为零，从而达到压缩 DCT 系数的目的。

③Zig-Zag 扫描 从量化后的 DCT 系数表中读出数据和表示数据的方式也是减少码率的一个重要过程。读出的方式可以由多种方式选择，如水平逐行读出、垂直逐列读出、交替读出和之字形扫描读出，其中之字形读出是最常见的一种，它实际上是按二维频率的高低顺序读出系数。

④可变字长编码 对 Zig-Zag 扫描后的数据进行编码，可分为两个步骤：游程长度编码和 Huffman 编码。可变字长编码是无损编码，即对 DC 和 AC 系数进行编码之后，在解码器端进行解码时，DC 和 AC 系数可恢复原值。其中游程长度编码是指一个码可以同时表示码字和前面有几个零。这种编码正好可以把之字形读出的优点显示出来了，因为之字形读出的系数在大多数情况下出现连续零的机会比较多，尤其在最后面部分。如果后面全为零，在读到最后一个数后，只要给出“结束块”(EOB)码，就可以结束读出，因此降低了码率。具体的做法如下：把一串零值系数和与其相邻的非零的 AC 系数组成一个数组，用一对符号表示（即用符号“1”和符号“2”表示）。符号“1”中包括两个数据：跨越长度和比特长，跨越长度就是非零 AC 系数前连续零的个数，比特长则是非零系数的编码位数，可通过查“DC 差分值、AC 系数比特长表”得到。符号“2”只包含非零系数的值，即振幅。若最后一个的非零 AC 系数后还有零系数，则用专门符号“EOB”来表示子图像的结束。DC 系数反映该子图像中包含的直流分量的大小，通常它和邻近的子图像的直流系数有较强的相关性，常用差分编码，即只对本子图像与前一个子图像的 DC 系数进行编码。由于它前面无零值，故符号“1”中没有跨越长度，只有比特长，符号“2”仍是差值的振幅。把 DC 和 AC 系数编成符号“1”和符号“2”的目的是为了便于进行可变长度编码。实际上，符号“1”用霍夫曼编码，符号“2”仍用二进制编码。

(6) 编码比特率的控制

不同的应用目的，可能需要不同的编码质量或编码比特率。另外，

MPEG-2 编码的输出比特率将随图像局部的特性而变化,而大多数传输信道是固定比特率的,而 MPEG-2 的输出比特流是可变比特率的,为此,需要在编码器的输出端和信道之间的接入一个数据缓冲器,并实时的按照该缓冲器的情况调整控制因子  $Q$ ,以保证缓冲器不发生溢出或空出。要求能够控制 MPEG-2 的编码质量或编码比特率,以适应用户或信道的需要。MPEG-2 是通过改变量化步长实现这一要求的。设定一个质量控制因子  $Q$ ,在量化时,用该因子和量化表中的量化步长相乘以后,作为实际的量化步长。这样一来,当要求较高的比特率时, $Q$  取较小的值,使量化步长按同一比例减小;反之, $Q$  取较大的值。这样, $Q$  应该作为一个编码参数和编码比特流一起传输给解码器。

以上就是 MPEG-2 标准的视频信号的压缩过程。实践证明,对于主级主类 (MP@ML, 720 × 576, 25 帧),在压缩率为 30:1 或更小时,可以提供广播质量的编码图像,完全了满足现代远程教育的视频编码的需求。

## 2.4 MPEG-2 视频码流的帧结构

经过压缩编码后的视频信号形成视频基本码流 (ES),MPEG-2 标准的视频基本码流可分成 6 个层次,从高往低依次是视频序列层 (Sequence)、图像组层 (GOP, Group of Picture)、图像层 (Picture)、像条层 (Slice)、宏块层 (Macro Block) 和像块层 (Block),如图 2-7 所示。除了宏块和像块层,上四层都从相应的起始码(SC)开始。起始码作为同步识别用<sup>[7]</sup>。在起始码后是序列头,包括图像尺寸、宽高比、图像速率等信息。在序列头后面是包含附加数据的序列扩展数据。为了确保能在不同时间随时进入视频序列,MPEG-2 允许重复发送序列头,序列层结束在序列结束码 (SEQEC)。在序列层下面是图像组层 (GOP)。所谓图像组是指相互间有预测和生存关系的一组图像,分成图像组的一个目的是在同一序列内可随时进入不同的图像,在图像组下是图像层,它主要包括了三种类型编码的图像:帧内编码的图像 (I 帧),双向预测编码的图像 (B 帧),前向预测编码的图像 (P 帧)。还有只使用 DCT 数据中的 DC 分量编码的图像(d 帧)。在实际传输的帧顺序是 IPBBPBBIBBP...,而显示的帧顺序是 IBBPBBPBBIBBP...。在图像层中,在起始码的后面是图像头,包括图像编码类型和时间参考信息。图像头后面是扩展数据,再后面是实际的图像数据。在图像层下面是像条层。每一个像条包括一定数量的宏块,其顺序是和行扫描顺序一致。像条是最低的比特流级

别，即一旦因误码失步可根据起始码重新同步。在像条层下面是宏块层，宏块层是整个结构中最下层。该层最前面是宏块层说明，表示用什么编码模式，后面是运动矢量以及  $16 \times 16$  位数据。MPEG-2 定义了三种宏块结构为：4: 2: 0 宏块，4: 2: 2 宏块和 4: 4: 4 宏块，分别包括 6, 8, 16 个像块。在 MPEG-2 中，“Block”可以是  $8 \times 8$  样值，可称为像块，也可以是  $8 \times 8$  DCT 系数或

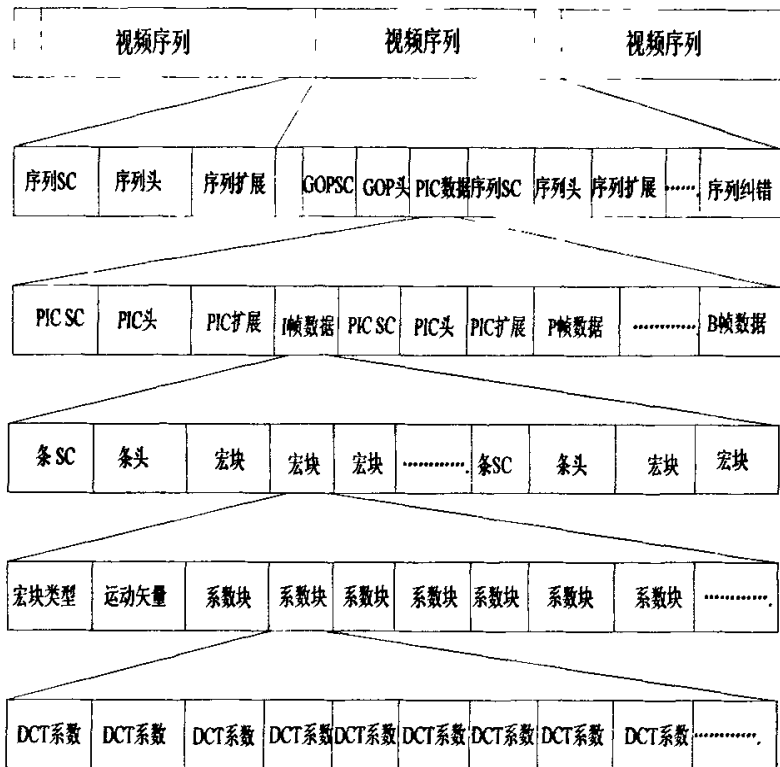


图 2-7 视频基本码流层次结构

Fig. 2-7 Video Bitstream basic hierarchical structure

重组数据，这时应称为系数块或数据块。所以上述各层次都与一定的信息处理有关。如视频序列实际上是节目的随机进入点，而 GOP 则是视频编辑的随机进入点，图像或帧是编码处理的单位，像素是用于同步的单位，宏块是运动补偿处理的单位，像块则是 DCT 处理单位。

## 2.5 基于硬件实现的 MPEG-2 视频编码系统

### (1) 系统的性能预期指标

针对山区、偏远农村现代远程教育这一特定应用领域，以 MPEG-2 作为运动图像编码算法的蓝本，提出了一个视频图像编码系统的硬件结构。

该系统主要性能预期指标如下：

- ① 视频输入为标准 PAL 制复合全电视信号；
- ② 数字化后的图像为 2CIF 格式，图像分辨率为  $720 \times 576$ ；
- ③ 宽范围的比特率输出  $1\text{Mbps} \sim 10\text{Mbps}$ ；
- ④ 输出帧率为 25 帧/秒；
- ⑤ 运动补偿精度为整像素，运动矢量范围为  $[-8, 7]$ ；

### (2) 系统结构和功能描述

系统框图如图 2-8 所示。图中被阴影框住的部分的运算量占整个系统运算量的 80% 以上，它是整个 MPEG-2 编码器的核心部分，也是本文所将要讨论主要内容。它包括块处理单元、运动估计单元、控制单元和内存接口。其中块处理单元主要完成  $8 \times 8$  大小块的二维离散余弦变换、量化/逆量化、Zig-Zag 编码、可变字长编码等一系列的块处理操作，最后输出 16bits 的可变字长码流；运动估计单元完成基于  $8 \times 8$  大小块的全搜索运动估计操作，最后输出 5bits 的运动向量 (MV)；最后，由控制单元将运动向量和可变字长码流送入到上层码流打包单元，进行码流打包，并输出完整的 MPEG-2 码流。

通过对 MPEG-2 运动图像编码器功能结构的划分、化整为零，有效地简化了设计的复杂度，这对整个编码器的设计是十分有利的。因此，在接下来的几章里将对图 2-8 中阴影框中的主要模块单元——块处理单元和运动估计单元的设计分别作以详细地介绍。

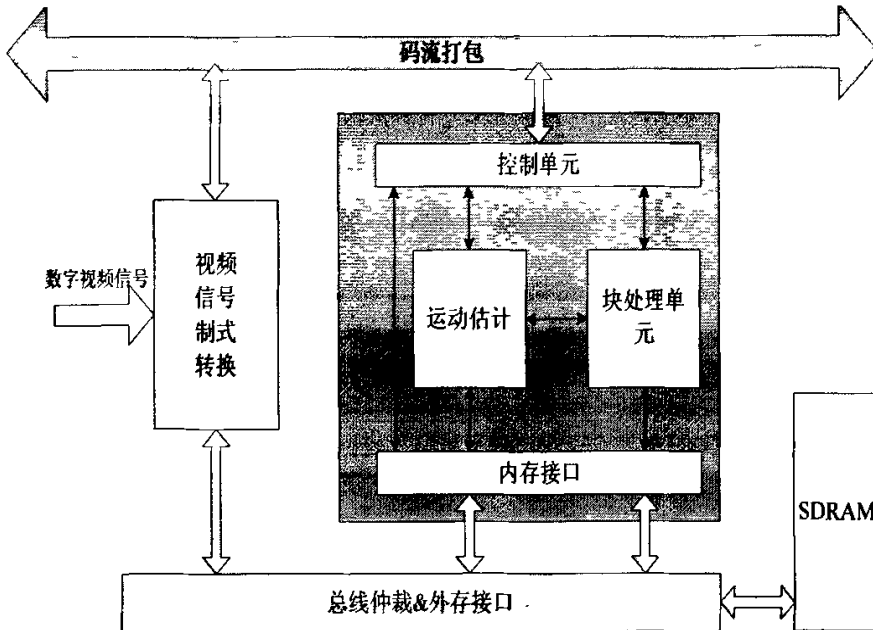


图 2-8 MPEG-2 编码器硬件电路结构图

Fig. 2-8 MPEG-2 encoder hardware circuit structure

## 2.6 本章小结

MPEG-2 的应用面很宽，因此对所需编码图像的性质，如大小、彩色分量的数量、编码方式等，均不能预先规定，在具体应用时须靠通信临时传递。为了适应各种不同的场合，MPEG-2 提供了几种编码方式以供选用，量化表、码表等也未加规定，仅提供推荐表。若效果不理想，用户可自己定义量化表和码表，并将这些表同数据一起传到解码器，供解码器端正确使用。因此，基于以上 MPEG-2 理论知识，并针对山区、偏远农村远程教育这一特定应用领域，将它灵活应用到本次设计中，提出了一个大体的 MPEG-2 运动图像编码器的硬件电路结构；而且，在本文后续内容中的 MPEG-2 编码器电路设计都是以 8 位表示每像素点的基于帧扫描工作模式的 MPEG-2 MP@ML 运动图像编码算法，其压缩与解压缩过程都是以  $8 \times 8$  大小的图块为数据处理单位。

### 3 块处理单元 IP 核设计

#### 3.1 二维离散余弦变换模块设计

数字图像处理 DCT 是数字图像处理中的一种重要处理手段, 被广泛地用于图像的压缩编码算法中, 已有的各种成熟的压缩标准如 JPEG、MPEG、H.26x 以及 HDTV 等都无一例外地采用基于 DCT/IDCT 的压缩编码<sup>[22]</sup>。离散余弦变换 (Discrete Cosine Transform) 是一种最主要的正交变换。它将图像信号从空间域变换到 DCT 域, 在保持原始信号的熵和能量不变的同时, 却使得 DCT 域系数之间的相关性减弱, 然后再对 DCT 域系数进行量化和编码, 以达到压缩的目的。

##### 3.1.1 二维 DCT 快速算法

离散余弦变换是一种有损压缩方案, 它是将一个  $N \times N$  大小图像块从空间域转换到 DCT 域。我们通常把信号分解到空间频率成份称为 DCT 系数。DCT 系数低频率位于 DCT 的系数矩阵的左上角, 较高的频率位于 DCT 系数矩阵的靠右区域, 由于人类视觉系统对低频率系数部分更加敏感, 所以在后续的量化操作中去处高频系数, 以到达压缩的目的<sup>[23]</sup>。

在进行 DCT 之前, 二维空间图像亮度数据通常较高, 为了减低传输的位率, 应先进行向下的电平移位, 设原样值的采样精度为 8 位, 是无符号整数, 范围  $0 \sim 255$ 。输入之前先减去 128, 范围变为  $-128 \sim +127$  的有符号整数。在设计硬件时, 由于是有符号数的运算, 无符号整数变为有符号整数, 需增加一位符号位, 使得下面的每一步运算都要加一位, 降低了速度, 增大了面积。而电平移位, 消除了这一缺陷。在解码时, 再使 DCT 反变换后的数值加上 128, 获得重建图像。而且由于余弦值为小于 1 的小数, 而在硬件中实现浮点小数运算会大大增加芯片面积, 降低系统速度<sup>[24]</sup>。因此要将余弦值改变形式, 以适应定点整数运算。可先将余弦值变为二进制小数, 再通过移位和四舍五入转换成二进制整数, 参与运算。最后, 将运算结果反向移位, 即得到真实结果, 且误差在可接受范围内。

二维信号的 DCT 变换定义为:

$$y(u, v) = \frac{2}{N} C(u) C(v) \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} x(i, j) \cos \frac{(2i+1)u\pi}{2N} \cos \frac{(2j+1)v\pi}{2N} \quad (3-1)$$

$$u, v = 0, 1, \dots, N-1$$

式中, 当  $u=0, v=0$  时,  $C(u), C(v) = 1/\sqrt{2}$ ; 当  $u=1, 2, \dots, N-1, v=1, 2, \dots, N-1$  时,  $C(u), C(v) = 1$ 。

二维 DCT 是一种基于实数的正交变换, 其变换核具有可分离性, 因此, 可以把二维 DCT 分解成两个一维 DCT:

$$\begin{aligned} y(u, v) &= \sqrt{\frac{2}{N}} \sum_{i=0}^{N-1} C(u) z(v, i) \cos \frac{(2i+1)u\pi}{2N} \\ z(v, i) &= \sqrt{\frac{2}{N}} \sum_{j=0}^{N-1} C(v) x(i, j) \cos \frac{(2j+1)v\pi}{2N} \end{aligned} \quad (3-2)$$

式中可以用矩阵的方式来表示:

$$Z = CXC^T \quad (3-3)$$

其中  $C$  和  $C^T$  代表  $8 \times 8$  DCT 系数矩阵, 令  $C(i, j)$  表示矩阵  $C$  的第  $i$  行,  $j$  列的元素则有:

$$\begin{aligned} C(i, j) &= K \cdot \cos[(2j+1) \cdot i \cdot \frac{\pi}{2N}] \\ C^T(i, j) &= K' \cdot \cos[(2i+1) \cdot j \cdot \frac{\pi}{2N}] \end{aligned} \quad (3-4)$$

其中  $K = \sqrt{1/N} (i=0)$ ,  $K' = \sqrt{1/N} (j=0)$ ,  $K = \sqrt{2/N} (i \neq 0)$ ,  $K' = \sqrt{2/N} (j \neq 0)$  对(3-3)式可以改写成:

$$\begin{aligned} Y &= CZ \\ Z &= XC^T \end{aligned} \quad (3-5)$$

式中  $C$  和  $C^T$  为矩阵系数, 因此, 我们便可以算出  $C$  和  $C^T$ :

$$C = \begin{bmatrix} a & a & a & a & a & a & a & a \\ b & d & e & g & -g & -e & -d & -b \\ c & f & -f & -c & -c & -f & f & c \\ d & -g & -b & -e & e & b & g & -d \\ a & -a & -a & a & a & -a & -a & a \\ e & -b & g & d & -d & -g & b & -e \\ f & -c & c & -f & -f & c & -c & f \\ g & -e & d & -b & b & -d & e & -g \end{bmatrix}, \quad C^T = \begin{bmatrix} a & b & c & d & a & e & f & g \\ a & d & f & -g & -a & -b & -c & -e \\ a & e & -f & -b & -a & -g & c & d \\ a & g & -c & -e & a & d & -f & -b \\ a & -g & -c & e & a & -d & -f & b \\ a & -e & -f & b & -a & -g & c & -d \\ a & -d & f & g & -a & b & -c & e \\ a & -b & c & -d & a & -e & f & -g \end{bmatrix}$$

$$\text{其中 } [a \ b \ c \ d \ e \ f \ g]^T = \frac{1}{2} \left[ \cos \frac{4\pi}{16} \ \cos \frac{\pi}{16} \ \cos \frac{2\pi}{16} \ \cos \frac{3\pi}{16} \ \cos \frac{5\pi}{16} \ \cos \frac{6\pi}{16} \ \cos \frac{7\pi}{16} \right]^T$$

因此, 通过式(3-5)的变换, 一个二维 DCT 变换就被分解成了两个一维 DCT 变换的运算, 通过这样的分解, 我们就可以用相对简单的多项式四则运

算来代替复杂的二维矩阵运算，从而达到简化电路设计复杂度的目的。

3.1.2 二维 DCT 电路设计

3.1.2.1 电路总体结构

由于二维 DCT 变换的核的可分离性<sup>[26]</sup>，因此在硬件设计上本文采用了两个一维 DCT 变换和两个乒乓结构的双口 buffer 来实现二维 DCT，前级 1D-DCT 模块完成行的 DCT 变换，后级 1D-DCT 完成列的 DCT 变换，具有乒乓结构的深度为 128 双口 Buffer 实现中间结果的缓存和转置的功能。



图 3-1 2D-DCT 电路总体结构

Fig. 3-1 2D-DCT circuit architecture

如图 3-1 所示，2D-DCT 电路首先对逐行对 8bits 的原始图像灰度值进行前级 1D-DCT 变换，变换的中间结果存储在两个双口 buffer（buffer1 和 buffer2）中：两个 buffer 交替的工作在读/写状态；例如，前级 1D-DCT 把原始图像灰度值的一维行变换后的结果逐行地写入 buffer1 中，当前级 1D-DCT 完成一次  $8 \times 8$  大小的变换后，buffer1 将被写满；后级 1D-DCT 逐列地从 buffer1 中读出数据进行列变换，同时前级 1D-DCT 开始进行下一个  $8 \times 8$  变换，这次的变换结果将写入到 buffer2 中。后级 1D-DCT 完成一个  $8 \times 8$  变换之后，它在下一次变换中将从 buffer2 中读取数据。按照这样的规律交替读写 buffer1 和 buffer2，便可以用两个一维 DCT 来实现二维 DCT。

3.1.2.2 前级 1D-DCT 电路结构

由式 (3-5) 可以看出，计算每个 DCT 系数需要 8 次乘法运算和 7 次加法运算，所以完成一个  $8 \times 8$  块的一维 DCT 变换共需 64 次乘法运算和 56 次加法运算，所需要的计算量是相当大的。但是根据式 (3-5) 还可以发现，DCT 的系数矩阵具有良好的对称性，因此可以将把中间结果 Z 分成偶数列和奇数列来计算。由于 X 是逐行进行变换的，设 X 的第 K 行为  $[X_{K0}, X_{K1}, X_{K2}, X_{K3}, X_{K4},$



$X_{K5}, X_{K6}, X_{K7}]$ , Z 的第 K 行为  $[Z_{K0}, Z_{K1}, Z_{K2}, Z_{K3}, Z_{K4}, Z_{K5}, Z_{K6}, Z_{K7}]$ 。

则 Z 的第 K 行偶数列为:

$$\begin{bmatrix} Z_{K0} \\ Z_{K2} \\ Z_{K4} \\ Z_{K6} \end{bmatrix} = \begin{bmatrix} a & c & a & f \\ a & f & -a & -c \\ a & -f & -a & c \\ a & -c & a & -f \end{bmatrix} \cdot \begin{bmatrix} X_{K0} + X_{K7} \\ X_{K1} + X_{K6} \\ X_{K2} + X_{K5} \\ X_{K3} + X_{K4} \end{bmatrix} \quad (3-6)$$

且 Z 的第 K 行奇数列为:

$$\begin{bmatrix} Z_{K1} \\ Z_{K3} \\ Z_{K5} \\ Z_{K7} \end{bmatrix} = \begin{bmatrix} b & d & e & g \\ e & -b & g & d \\ g & -e & d & -b \\ a & -c & a & -f \end{bmatrix} \cdot \begin{bmatrix} X_{K0} - X_{K7} \\ X_{K1} - X_{K6} \\ X_{K2} - X_{K5} \\ X_{K3} - X_{K4} \end{bmatrix} \quad (3-7)$$

由此看出, 一维 DCT 变换可以通过奇偶项分离, 使得每个系数所需要的乘法运算次数从 8 次减少为 4 次, 加法运算减少为 4 次, (其中一次是对输入数据的预处理, 其它三次是对乘积项累加) 这样一个完整的  $8 \times 8$  一维 DCT 运算共需 32 次乘法运算和 32 次加法运算, 计算量已经大大减少。通过奇偶项分离后的前级 1D-DCT 的变换就可以按照如图 3-2 所示的电路结构工作, 首先 8bits 的原始图像灰度值信号 X 按照  $X_{00} - X_{07}, X_{10} - X_{17}, \dots, X_{70} - X_{77}$  的顺序以每个时钟周期一个像素的方式输入到一个 8 位串型移位寄存器中。同时另外一个数据暂存器每 8 的时钟从 8 位串型移位寄存器中读取数据  $X_{K0} - X_{K7}$ , 这样就可以一次读出 8 个像素的值 (矩阵的一行), 从而可以根据式 (3-8) 计算出一维 DCT。

$$\begin{aligned} Z_{K0} &= a \times (X_{K0} + X_{K7}) + a \times (X_{K1} + X_{K6}) + a \times (X_{K2} + X_{K5}) + a \times (X_{K3} + X_{K4}) \\ Z_{K1} &= b \times (X_{K0} - X_{K7}) + d \times (X_{K1} - X_{K6}) + e \times (X_{K2} - X_{K5}) + g \times (X_{K3} - X_{K4}) \\ Z_{K2} &= c \times (X_{K0} + X_{K7}) + f \times (X_{K1} + X_{K6}) + (-f) \times (X_{K2} - X_{K5}) + (-c) \times (X_{K3} + X_{K4}) \\ Z_{K3} &= d \times (X_{K0} - X_{K7}) + (-g) \times (X_{K1} - X_{K6}) + (-b) \times (X_{K2} + X_{K5}) + (-e) \times (X_{K3} - X_{K4}) \\ Z_{K4} &= a \times (X_{K0} + X_{K7}) + (-a) \times (X_{K1} + X_{K6}) + (-a) \times (X_{K2} + X_{K5}) + a \times (X_{K3} + X_{K4}) \\ Z_{K5} &= e \times (X_{K0} - X_{K7}) + (-b) \times (X_{K1} - X_{K6}) + g \times (X_{K2} + X_{K5}) + d \times (X_{K3} - X_{K4}) \\ Z_{K6} &= f \times (X_{K0} + X_{K7}) + (-c) \times (X_{K1} + X_{K6}) + c \times (X_{K2} + X_{K5}) + (-f) \times (X_{K3} + X_{K4}) \\ Z_{K7} &= g \times (X_{K0} - X_{K7}) + (-e) \times (X_{K1} - X_{K6}) + d \times (X_{K2} + X_{K5}) + (-b) \times (X_{K3} - X_{K4}) \end{aligned} \quad (3-8)$$

由式 (3-8) 可以看出, 常数  $\pm a, \pm g$  存放于循环寄存器中, 每个时钟周期该循环寄存器循环右移一位输出常数 U, 前级加/减法器轮流切换完成

$X_{kj} \pm X_{k(n-1-j)}$ 。流水乘法器完成  $V = U \times [X_{kj} \pm X_{k(n-1-j)}]$ , 后级流水加法器完成  $(v1+v2)+(v3+v4)$  的计算。计算结果  $Z_{k0} - Z_{k7}$  以串行方式输出到两个乒乓结构的 buffer 中的一个。为了提高处理器的时钟频率, 电路广泛地采用了流水线结构, 在经过复位之后的 77 个时钟周期流水延迟后每个时钟周期将输出一个一维 DCT 的变换结果到乒乓 buffer 中。

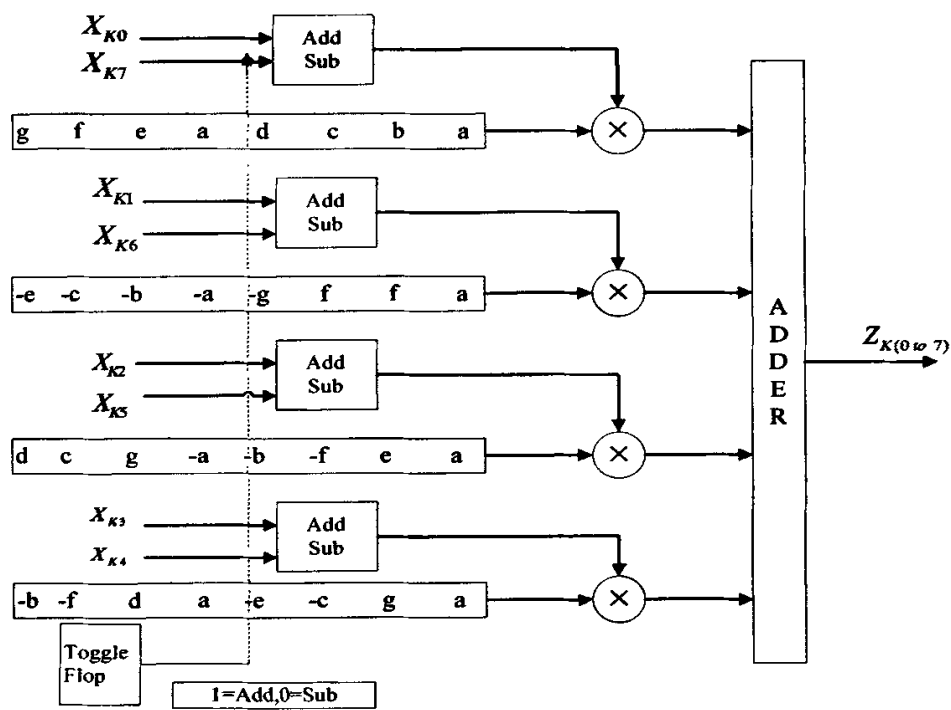


图 3-2 前级 1D-DCT 结构图  
Fig. 3-2 The 1st 1D-DCT structure

3.1.2.3 后级 1D-DCT 电路结构

后级 1D-DCT 变换采用式(3-5)来计算  $Y=CZ$ 。同理, 注意到  $C$  的偶数行为偶对称, 奇数行为奇对称。因此, 我们把最终所需的 2D-DCT 变换结果  $Y$  分成偶数行和奇数行来计算。计算思路和前级 1D-DCT 一样, 区别在于中间结果  $Z$  需要逐列输入。在前级 DCT 完成  $8 \times 8$  像素的 2D-DCT 的行余弦变换后, 后级 DCT 就开始从乒乓 buffer 中逐列地读取前级 1D-DCT 的变换结

果进行 2D-DCT 变换。后级 1D-DCT 的结构如图 3-3 所示。在经过前级 DCT 变换 77 个初始流水延时后, 后级 DCT 开始计算, 再经过 69 个时钟周期后, 输出第一个二维 DCT 的变换结果, 即在经过初始的 146 个流水延时二维 DCT 变换结果  $Y(u, v)$  将按照  $Y_{00} - Y_{70}, Y_{01} - Y_{71}, \dots, Y_{07} - Y_{77}$  的顺序连续输出。

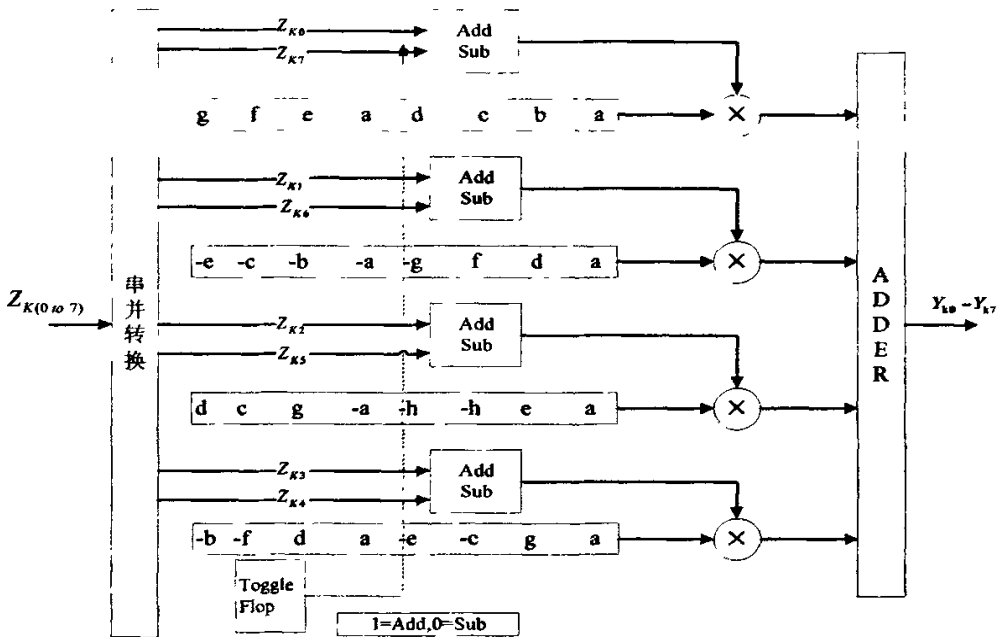


图 3-3 后级 1D-DCT 结构图

Fig. 3-3 The 2nd 1D-DCT structure

#### 3.1.2.4 关键路径上专用加法器的优化设计

在上述的基于行列分离的二维 DCT 变换中, 涉及加法地方虽然不多, 但是它们都是位于整个二维 DCT 运算的关键路径上, 而对于关键路径上的设计<sup>[23]</sup>, 我们有必要千方百计提高其运算速度。因此, 在本文中对于加法器的设计, 使用了超前进位加法器, 并根据实际情况对加法器结构进行改进, 使其它更加符合本设计的需要。

设一个  $n$  位的加法器的第  $i$  位的输入为  $a_i$ 、 $b_i$ 、 $c_i$ , 输出  $s_i$ 、 $c_{i+1}$ , 这里  $c_i$  表示低位的进位,  $c_{i+1}$  ( $i=n-1, n-2, \dots, 1, 0$ ) 是向高位的进位,  $c_0$  是整个加法器的进位输入, 而  $c_n$  是整个加法器的进位输出。

$$\text{则 Sum: } s_i = a_i \oplus b_i \oplus c_i \quad (3-9)$$

$$\text{进位: } c_{i+1} = a_i b_i + a_i c_i + b_i c_i \quad (3-10)$$

$$\text{令: } g_i = a_i b_i \quad (3-11)$$

$$p_i = a_i + b_i \quad (3-12)$$

$$c_{i+1} = g_i + p_i c_i \quad (3-13)$$

称为  $g$  为进位产生函数,  $p$  为进位传递函数。把式(3-10)展开, 得到:

$$c_{i+1} = g_i + p_i g_{i-1} + p_i p_{i-1} g_{i-2} + \dots + p_i p_{i-1} \dots p_1 g_0 + p_i p_{i-1} \dots p_0 c_0 \quad (3-14)$$

随着位数的增加式(3-14)会不断的加长, 但是总保持 3 个逻辑级的深度, 因此形成进位的延迟是与位数无关的常数。一旦进位 ( $c_1 \sim c_{n-1}$ ) 算出来以后, Sum 就可以根据式(3-9)得出, 这就是超前进位加法器的基本思想。

但是, 对于本设计, 因为是 9 位 (扩展一位符号位, 由于 Xin 都是正整数, 所以加法器的设计不考虑符号位) 的加法器, 如果直接按照式(3-14)进位逻辑, 形成的电路是很不规则的, 并且需要长线驱动, 产生较高位的进位延时, 需要大驱动信号和大扇入门。因此, 这种实现方式并不太现实。

这里, 为了克服上述的缺点, 对于一个 8 位的加法器。可以按 4 位一组的形式对其分组, 组内实现超前进位, 组间也实现超前进位。相应地超前进位逻辑分成 2 级。第  $m$  ( $0 \sim 15$ ) 位的  $g, p$  可以表示为:  $g_{4k+j} = a_{4k+j} b_{4k+j}$ ,

$p_{4k+j} = a_{4k+j} + b_{4k+j}$ ;  $k=m/4$ , 代表组的位置;  $j$  为余数, 代表该位在该组中的位置。各个 4 位超前进位加法器的组间进位产生函数:

$$G_k = g_{4k+3} + p_{4k+3} g_{4k+2} + p_{4k+3} p_{4k+2} g_{4k+1} + p_{4k+3} p_{4k+2} p_{4k+1} g_{4k} \quad (3-15)$$

组间进位传递函数:

$$P_k = p_{4k+3} p_{4k+2} p_{4k+1} g_{4k} \quad (3-16)$$

组间进位:

$$C_k = G_k + P_k c_0 \quad (3-17)$$

每个 4 位的超前进位加法器模块计算各组内每一位  $g, p$  和组间的  $P, G$ ,

第二级根据各组的  $P$ 、 $G$  和  $c_0$  计算出各组间的进位  $c_{4i+4}$ 。计算出所有组进位都要送回各个 4 位的超前进位加法器模块，并计算出每一位的和。如图 3-4 所示。这种结构适用于所有的  $n$  ( $n > 4$ ) 位加法器，只是根据  $n$  的不同，级数也不同而已。

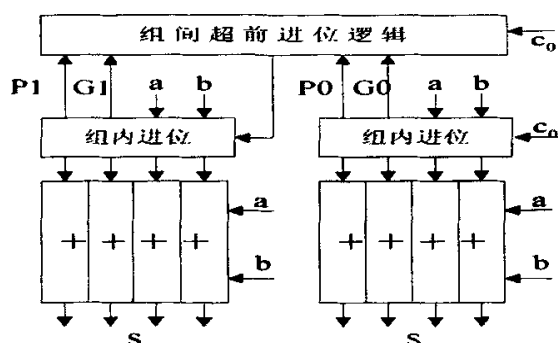


图 3-4 8 位超前进位加法器结构图

Fig. 3-4 8-bit CLA chart

经过这种改造，超前进位加法器延时包括式 (3-12) 和式 (3-13) 产生  $p_i$  和  $g_i$  的 1 级门延时；超前进位电路产生的 2 级门延时；用式 (3-9) 计算  $s_i$  的 3 级门延时。总的延时为 9 级门延时，与串联加法器的 32 级门延时相比，超前进位加法器的延时大大缩短了。

### 3.1.3 二维 DCT 电路的综合与仿真

使用 Synplify Pro 7.6 对代码进行逻辑综合后，得到两个 1D-DCT 单元及整个 DCT 模块，该模块共消耗了 1763 个 LE 和 1204bits 内嵌 EAB 资源(内存)，与文献[26]的数据进行比较，由于本设计充分利用了 FPGA 的 EAB 资源，消耗的逻辑资源比文献[26]少将近 5096。并且通过引入了专用的加法器设计，使得模块的关键路径上的时延减少到 5.65ns，系统时钟比文献[27]提高了 4 倍之多。在节约硬件资源的同时大大的提高了处理速度，虽然在选用的器件上有所出入，但是从前文论述的设计结构就可以看出，本文的设计具有相当的优势，最终的综合数据进一步证明并体现了这一优势所在。

二维 DCT 处理器模块的接口设计如图 3-5 所示，采用异步置位方式，当复位信号 RST 从高电平跳变到低电平，并持续保持低电平 10 个时钟周期后，二维 DCT 处理器模块开始工作，在工作时，每个时钟周期输入一个 8bits

灰度数据 *xin*,输出一组 12bits 的有符号数据 *DCT\_2d*, 以及一个输出标志信号 *rdy\_out*, 其接口时序图如 3-6 所示。图 3-7 给出了经过 Synplify Pro 7.6 综合器优化了的二维 DCT 电路部分 RTL 图。

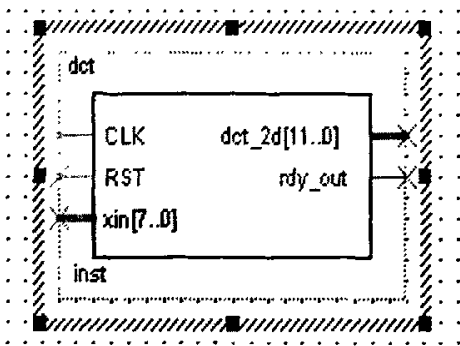


图 3-5 二维 DCT 处理器模块接口图

Fig.3-5 Interface of 2D-DCT processor module

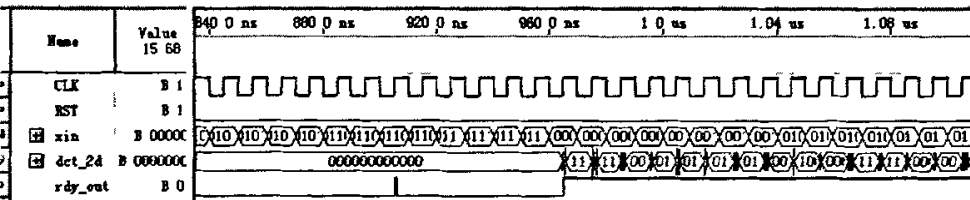


图 3-6 二维 DCT 处理器模块的接口时序图

Fig.3-6 Timing diagram for 2D-DCT processor

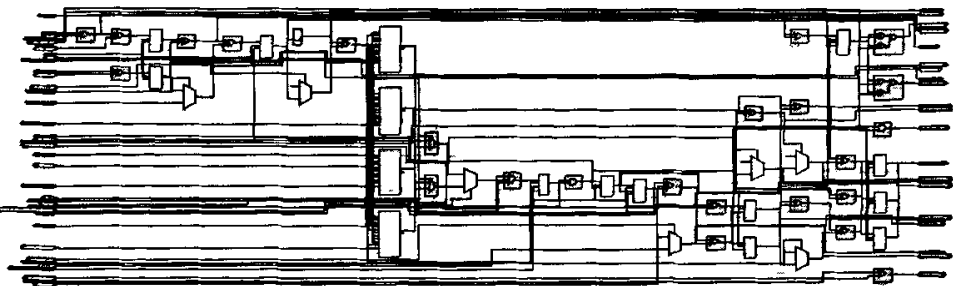


图 3-7 二维 DCT 处理器模块部分 RTL 图

Fig.3-7 The RTL map for one part of 2D-DCT

使用《Image and Video Compression Standard》第二版中给出的测试矩阵 RAW 作为 DCT 模块的输入数据, 得到的结果设为 A; 由 matlab 语言测试程序对相同的输入计算结果设为 B, 通过比较发现 A 与 B 误差在可接受的  $\pm 2$  范围之内, 进一步说明了二维 DCT 模块电路设计的正确性和高效性。

$$RAW = \begin{bmatrix} 40 & 33 & 33 & 22 & 26 & 40 & 36 & 26 \\ 43 & 26 & 33 & 22 & 29 & 43 & 22 & 26 \\ 43 & 40 & 19 & 36 & 36 & 33 & 15 & 26 \\ 36 & 43 & 26 & 33 & 29 & 29 & 19 & 4 \\ 33 & 33 & 29 & 26 & 15 & 33 & 26 & 4 \\ 36 & 33 & 33 & 26 & 22 & 29 & 26 & 12 \\ 33 & 40 & 29 & 26 & 33 & 12 & 4 & 26 \\ 33 & 29 & 22 & 12 & 12 & 4 & 8 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 241.5 & 45.8 & 3.2 & 19.1 & -5.5 & -4.0 & 4.9 & -6.2 \\ 35.5 & -22.0 & 1.9 & 12.2 & -1.3 & 1.6 & 10.7 & -5.1 \\ -7.0 & -4.5 & 13.4 & -6.5 & 5.5 & -4.5 & 7.4 & 10.9 \\ 10.7 & -11.0 & 1.7 & -1.5 & -15.6 & 9.2 & 4.9 & 4.7 \\ -16.0 & 7.2 & -1.4 & 5.3 & -17.5 & 12.5 & -7.5 & 1.4 \\ 8.5 & 6.0 & -5.7 & -4.9 & 8.6 & -11.4 & -10.9 & -1.1 \\ -2.1 & 1.2 & 0.6 & 5.5 & -3.4 & 7.3 & -6.4 & -1.5 \\ 1.1 & -1.7 & 1.4 & -2.2 & 5.2 & -4.6 & -2.0 & -4.7 \end{bmatrix}$$

$$B = \begin{bmatrix} 241.5000 & 46.1788 & 3.2992 & 19.5139 & -5.5000 & -4.7890 & 4.0454 & -6.2886 \\ 34.8102 & -21.2867 & 2.8521 & 12.2000 & -1.0228 & 1.6529 & 10.7072 & -5.9146 \\ -7.4007 & -5.5439 & 13.3995 & -6.6779 & 5.9492 & -4.9896 & 7.3891 & 10.9100 \\ 10.8920 & -10.6442 & 1.7496 & -1.2231 & -15.2818 & 9.5442 & 5.3650 & 4.7000 \\ -16.0000 & 7.2202 & -1.4843 & 5.3073 & -17.5000 & 12.5211 & -7.5031 & 1.4746 \\ 8.5232 & 6.0553 & -5.7610 & -4.9056 & 8.6709 & -11.4110 & -10.9253 & -1.1943 \\ -1.8147 & 1.2603 & 0.3891 & 5.5368 & -3.2700 & 7.3679 & -6.3995 & -1.4548 \\ 0.7701 & -1.3647 & 1.2240 & -2.1026 & 5.7104 & -4.2350 & -2.0589 & -4.0727 \end{bmatrix}$$

### 3.2 量化及逆量化模块设计

DCT 编码本身不直接进行数据压缩, 而是把原始图像采样数据变换成一种更容易进行压缩的表达形式。量化相当于采用一个低通滤波, 将高频成分

滤去，即将系数减小或变为 0，而不至造成太大的图像质量下降<sup>[20][21]</sup>。DCT 系数量化是 DCT 变换编码数据压缩技术的关键所在。量化的作用是在满足图象质量的前提下进一步提高压缩率，也就是要去掉那些对图象质量影响不明显的信息，通过量化可以调节数据压缩比，使 DCT 系数值的精度减少，量化是压缩算法中产生失真的根源<sup>[20]</sup>。

### 3.2.1 量化及逆量化算法描述

#### 3.2.1.1 量化算法描述

每个  $8 \times 8$  块输入图像经过 DCT 变换后，产生 64 个系数要进行量化。视频压缩系统中利用了视觉加权，即根据视觉重要性对每一系数进行归一化，然后所有归一化后的系数以同样的方式进行量化。视觉加权由量化器矩阵确定。系数量化后，精度损失对图像改变视觉效果的影响比其他部分大。因而可以使用量化器比例因子，又叫做量化步长，来改变每一个宏块的量化级。对视觉上被认为是不重要的整个宏块可以量化得更粗糙些，减少图像编码的比特数，而视觉上重要的系数可以用更精细的步长来量化。量化步长也可以控制编码数据速率。量化后的 DCT 系数通常会出 0 值，可采用熵编码技术对系数进行进一步压缩。

MPEG-2 标准中的图像帧由 I 帧、P 帧、B 帧这三种不同的图像帧组成。因此 MPEG-2 量化器支持两种量化器矩阵，一种用于帧内编码的宏块；另外一种用于非帧内编码的宏块。视频编码系统为这两种矩阵定义了默认值。量化器矩阵可以在图像层上修改。帧内编码块中的 DC 系数（即直流系数矩阵中的左上角的系数）的量化不受量化器比例因子的影响，只取决于帧内量化器矩阵中的 (0, 0) 项<sup>[21]</sup>。

对于 4: 2: 0 格式的宏块，在量化的时候将会采用上述的 MPEG-2 标准支持的默认量化矩阵。而对 4: 2: 2 和 4: 4: 4 格式宏块量化时候，MPEG-2 标准将会使用到四种量化矩阵，它们分别针是针对帧内宏块和帧间宏块的色差信号量化矩阵和帧内宏块和帧间宏块的亮度信号量化矩阵<sup>[22]</sup>。同时，MPEG-2 标准还允许用户自定义量化矩阵，比如在 MPEG-2 MP@ML 编码时，编码器首先根据 `q_value_new_intra` 和 `q_value_new_non_intra` 两个控制信号判断是否采用用户自定义的量化矩阵，若两个控制信号都为假，MPEG-2 编码器将会自动选取默认的量化矩阵。



4: 2: 0格式帧内宏块默认量化矩阵

$$\begin{bmatrix} 8 & 16 & 29 & 22 & 26 & 27 & 29 & 34 \\ 16 & 16 & 22 & 24 & 27 & 29 & 34 & 37 \\ 19 & 22 & 26 & 27 & 29 & 34 & 34 & 38 \\ 22 & 22 & 26 & 27 & 29 & 34 & 37 & 40 \\ 22 & 26 & 27 & 29 & 32 & 35 & 40 & 48 \\ 26 & 27 & 29 & 32 & 35 & 40 & 48 & 58 \\ 26 & 27 & 29 & 34 & 38 & 46 & 56 & 69 \\ 27 & 29 & 35 & 38 & 46 & 56 & 69 & 83 \end{bmatrix}$$

4: 2: 0格式帧间宏块默认量化矩阵

$$\begin{bmatrix} 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \end{bmatrix}$$

AC 系数量化公式:

$$QDCT(i,j) = \left\lfloor \frac{32 \times DCT(i,j)}{Qmatrix(i,j) \times Quantizer Scale} + k \right\rfloor$$

当对帧内宏块编码时,  $k=0$ ;

当对帧间宏块编码时,  $k=1$ , 当  $DCT(i, j) > 0$ ;

$k=-1$ , 当  $DCT(i, j) < 0$ ;

$k=0$ , 当  $DCT(i, j)=0$ ;

(3-18)

式中, Quanter scale 编码是一个 5bits 表示的, 范围在 1 到 31 的无符号整数。MPEG-2 编码器/解码器可以在条层或宏块层对其进行修改, 用来进一步的控制视频流比特率<sup>[32]</sup>。其中 0 值是被禁止的, 其他的 31 个无符号整数被映射成两组范围在 1 到 112 的整数, 称为量化步长。

对于帧内宏块的 DC 系数量化, 量化器的量化步长由所选择的 DCT 系数的精度所决定, 其 DC 系数量化公式如下:

$$QDCT(0,0) = \frac{DCT(0,0)}{k} \quad (3-19)$$

式中  $k$  是由 DCT 系数精度所决定的常数：当 DCT 系数为 8bits 精度时， $k=8$ ；当 DCT 系数为 9bits 精度时， $k=4$ ；当 DCT 系数为 10bits 精度时， $k=2$ ；当 DCT 系数为 11bits 精度时， $k=1$ 。同样，MPEG-2 标准也允许用户自定义的 DC 量化步长来达到用户所要求的压缩比例。

### 3.2.1.2 逆量化算法描述

对逆量化而言，在数学上来说即为量化的逆过程。二维系数  $QF[v][u]$  逆量化后产生重构的 DCT 系数。逆量化过程是用变字长解码后的数据乘以量化步长<sup>[3]</sup>。量化步长由量化矩阵和量化标尺决定。其逆量化公式如下：

$$DCT = \frac{(2 \times QDCT(i,j) + k) \times Qmatrix(i,j) \times quantiser\_scale}{32}$$

当对帧内宏块编码时， $k=0$ ；

当对帧间宏块编码时， $k=1$ ，当  $DCT(i,j) > 0$ ； (3-20)

$k=-1$ ，当  $DCT(i,j) < 0$ ；

$k=0$ ，当  $DCT(i,j)=0$ ；

式中的逆量化后的数值结果被限制在  $-2048$  到  $+2047$  之间，如果超出此范围便会出现饱和，因此必须通过饱和控制对超过此范围的数据进行取舍：对超过  $+2047$  的数值取  $+2047$ ；对超过  $-2048$  的数值取  $-2048$ ；对  $[-2048, +2047]$  之间的数值取逆量化的结果数值。

由于编码器端的 DCT 变换和解码器端的 IDCT 变换的差异，在逆量化的时候，还会产生失配错误。在理想情况下，一个  $8 \times 8$  块先后通过 DCT 变换和 IDCT 变换后会还原出  $8 \times 8$  块的原始值。但事实上，一个  $8 \times 8$  块的 DCT 变换值经过量化后会产生量化误差，并且会在接下来的编码过程中传播，因此该量化值在逆量化的时候，不会还原出  $8 \times 8$  块的原始值。因此有必要对其进行失配控制，即对所有饱和后的 DCT 系数求和，若 Sum 为奇数，则所有的数值保持不变，若 Sum 为偶数，则式(3-20)中的 DCT(7,7) 项应改为：

$$\begin{cases} DCT'(7,7) = DCT(7,7) - 1 & \text{如果 sum 为偶数, } DCT(7,7) \text{ 为奇数} \\ DCT'(7,7) = DCT(7,7) + 1 & \text{如果 sum 为偶数, } DCT(7,7) \text{ 为偶数} \end{cases} \quad (3-21)$$

## 3.2.2 量化及逆量化电路设计

### 3.2.2.1 量化电路设计

量化实际是执行除法操作。硬件除法是通过迭代减法来完成的，速度很慢，为此采用乘法代替除法。因为量化表的值是固定的，其倒数的值也是不

变的, 计算量化值的倒数, 则量化的除法变为与量化值倒数的乘法。由于量化矩阵的倒数是小数, 故意先将其乘上 2 的 N 次幂再取整, 量化时直接相乘, 最后结果再右移 N 位即可 (截去后 N 位)。所以需要把量化公式 (3-18) 式改写成如式 (3-22) 的形式, 以便将顺序的矩阵行列式乘除法混合运算变成并行的单一乘法运算, 从而达到降低了运算复杂度和提高系统时钟频率的目的。

$$QDCT(i, j) = [DCT(i, j) \times (16 / \text{quantiser\_scale}) \times (1 / Qmatrix(i, j))] + \frac{k}{2} \quad (3-22)$$

式中  $16 / \text{quantiser\_scale}$  的所有值都存放在名为  $q\_scale\_mem$  且地址范围为  $[0, 63]$  的片内 ROM 中。对  $1 / Qmatrix(i, j)$  部分, 同样也采用了地址范围  $[0, 255]$  大小的名为  $q\_value\_mem$  的 ROM 来存储其值。

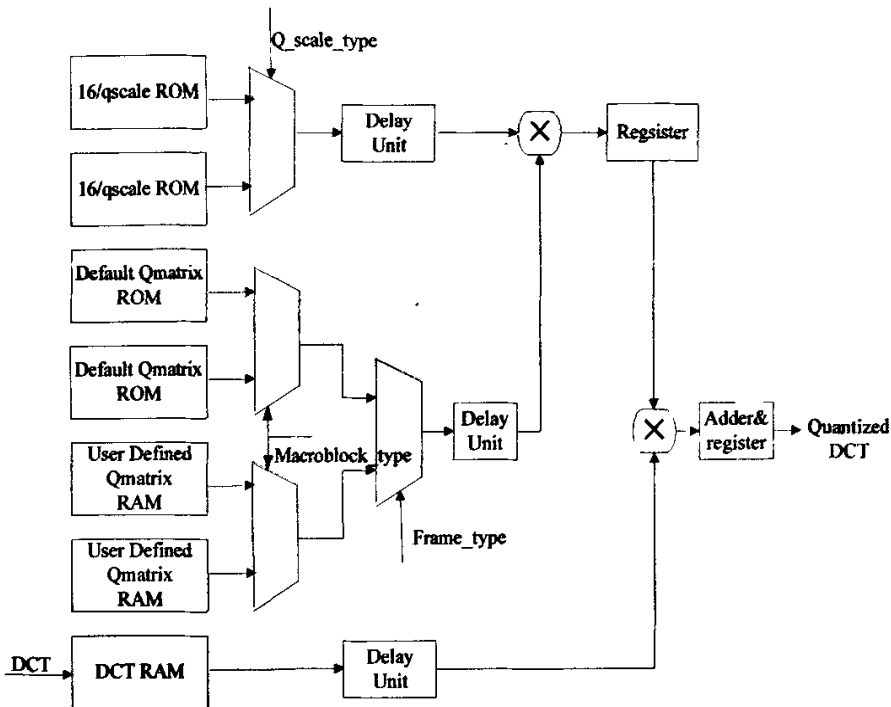


图 3-8 量化器结构图

Fig. 3-8 Quantization Structure

同理, 默认量化矩阵、用户自定义量化矩阵分别储存于名为  $def\_q\_mem$ 、 $q\_value\_new\_intra$  和  $q\_value\_new\_non\_intra$  的 RAM 当中, 其具体的量化器结构图如 3-8 所示。

### 3.2.2.2 逆量化电路设计

逆量化算法决定于 DCT 系数的重构公式<sup>[30]</sup>。同理,为了使设计符合 FPGA 器件特性,逆量化公式式(3-20)可以改写成如下形式:

$$DCT = (2 \times QuantizedDCT + k) \times [Qmatrix \times (Qscale / 32)] \quad (3-23)$$

可以看出,由式(3-23)所决定的逆量化器的电路结构同量化器电路结构具有相似之处。因此,同理中  $Qscale/32$  的所有值都存放在名为  $q\_scale\_mem$  且地址范围为  $[0, 63]$  的片内 ROM 中。常数量化矩阵  $Qmatrix(i, j)$ , 同样也采用了地址范围  $[0, 255]$  大小的名为  $q\_value\_mem$  的 ROM 来存储其值。同理,对于 MPEG-2 标准的默认量化矩阵、用户自定义量化矩阵分别储存于名为  $def\_q\_mem$ 、 $q\_value\_new\_intra$  和  $q\_value\_new\_non\_intra$  的 RAM 当中。其具体的量化器结构图如图 3-9 所示。

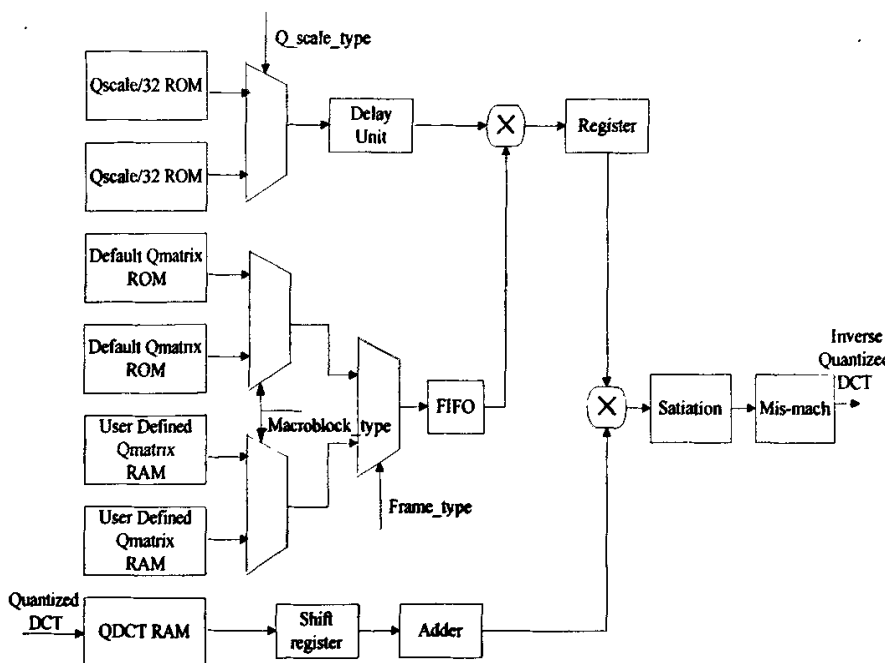


图 3-9 逆量化器结构图

Fig. 3-9 Inverse quantization Structure

### 3.2.3 量化及逆量化电路的综合与仿真

整个量化器和逆量化器模块采用 VHDL 设计。源程序输入、模块综合与时序验证均在 Altera QuartusII 中完成。采用 Cyclone EP1C12Q240C8 器件实现

后，其中量化器模块共占用 1612 个 LE，4 个 DSP block，最高运行速度为 55.51MHz，逆量化器模块共占用 1678 个 LE，8 个 DSP block，最高运行速度为 55.56MHz，并将综合结果与 MPEG-2 TM5 比较发现，所设计的量化和逆量化电路均达到了 MPEG-2 TM5 的规范要求。量化器模块在器件复位的 10 个时钟之后，开始输出量化数据，模块接口图、时序图如图 3-10 和 3-11 所示；逆量化器模块在器件复位的 13 个时钟之后，开始输出逆量化数据，其模块接口图、时序图如图 3-13 和 3-14 所示。图 3-12 和图 3-15 分别给出了经过 Synplify Pro 7.6 综合器优化了的量化/逆量化电路部分 RTL 图。

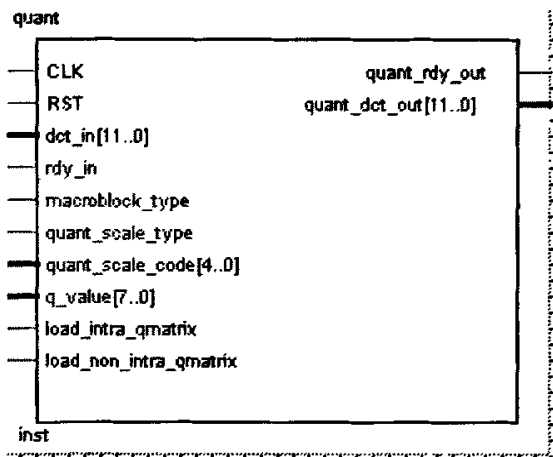


图 3-10 量化器模块接口图

Fig.3-10 Interface of quantization module

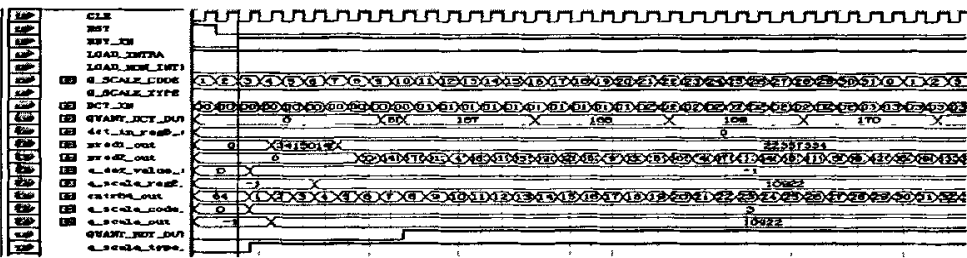


图 3-11 量化器模块接口波形图

Fig.3-11 Timing diagram for quantitation

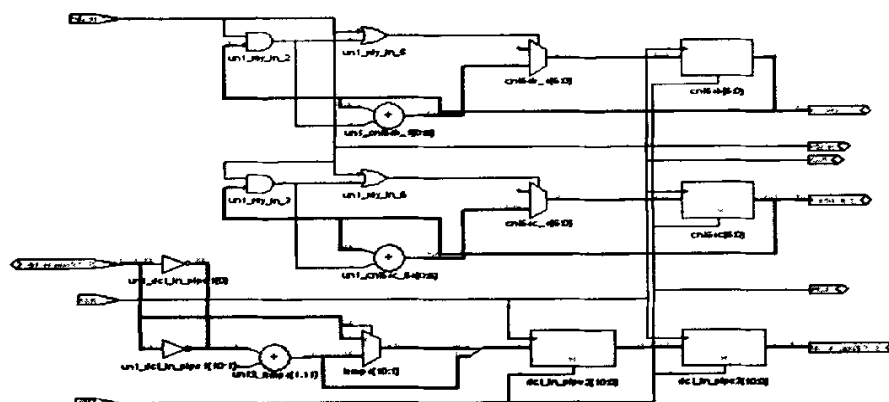
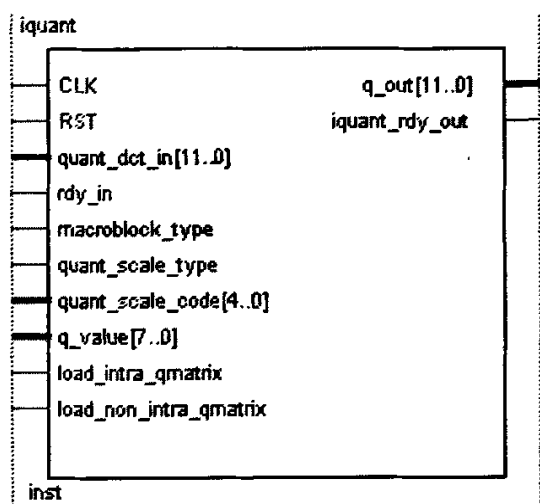


图 3-12 量化器模块部分 RTL 图

**Fig. 3-12** The RTL map for one part of quantitation



**图 3-13 逆量化器模块接口图**

**Fig. 3-13**    **Interface of inverse quantization module**

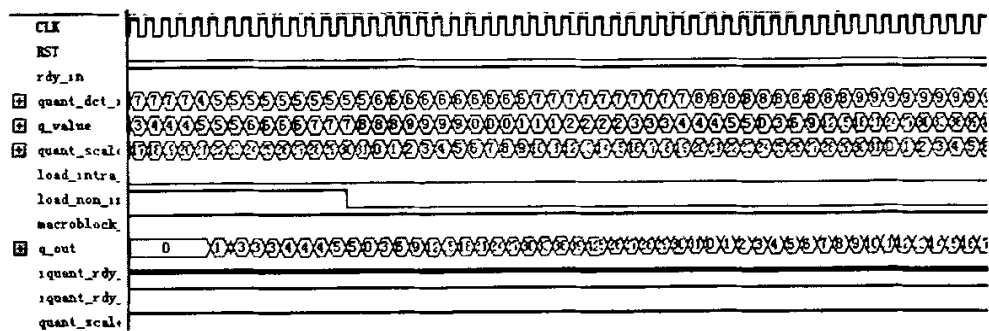


图 3-14 逆量化器模块接口波形图

Fig. 3-14 Timing diagram for inverse quantitation

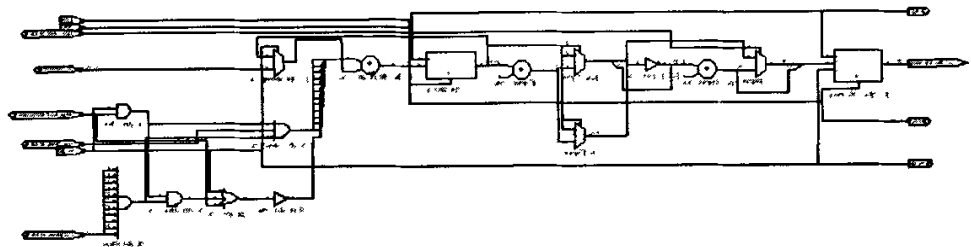


图 3-15 逆量化器模块部分 RTL 图

Fig. 3-15 The RTL map for one part of inverse quantitation

3.3 Zig-Zag 编码模块设计

在 MPEG-2 运动图像编码模块中变长编码（VLC）是最后一项无失真编码，它能进一步地压缩已被量化的数据<sup>[28][29]</sup>。从硬件结构上来说，它主要是由 Zig-Zag 扫描器、游程长度编码器，哈夫曼编码器三个模块组成，如图 3-16 所示。

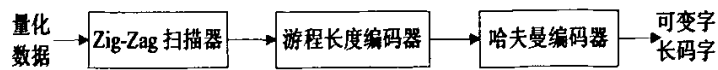


图 3-16 可变长编码器结构图

Fig. 3-16 Variable length coder structure

3.3.1 Zig-Zag 扫描算法描述

由于 DCT 变换后的变换系数矩阵中除主对角线外的系数很多是 0, 或接近于 0, 再加上量化处理, 有更多的 0 产生, 而这些 0 往往是连在一起成串出现<sup>[16]</sup>。因而不对单个的 0 编码, 而对 0 的游程(连续 0 的个数)编码, 无疑会明显提高编码效率。为了增加连续 0 的个数, 对已量化的 DCT 系数通过 Zig-Zag 扫描器进行交替扫描或之字型扫描两种不同的扫描格式排序, 将变换系数按顺序变成一维数组, 如图 3-17 所示。由于随空间频率增高, 0 出现的概率越来越大。因此, 这种数据排列顺序的转变对编码压缩十分有利。特别是很多经变换和量化后的像块通过这样的排序后, 排在队尾的很长一串系数全是 0, 这时可以根本不需要对这些 0 进行游程编码, 而只在位于此 0 游程之前的那一个非 0 系数之后加一个专用的块结束码(EOB), 就可以结束这个像块的编码, 解码器端当收到 EOB 信号后将自动补 0, 直到补足 64 个系数为止, 因此又可以节省不少传输码位。

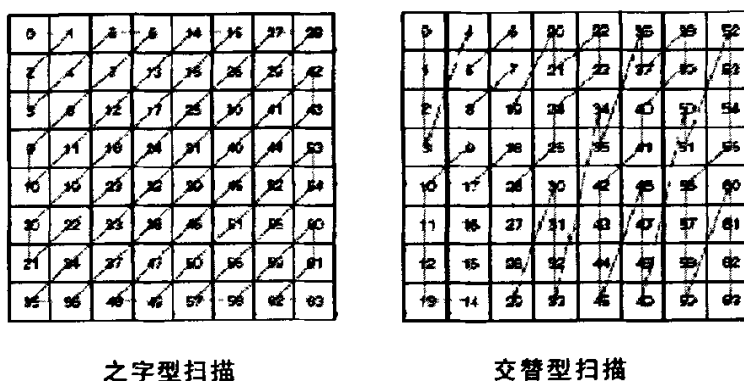


图 3-17 Zig-Zag 扫描

Fig. 3-17 Zig-Zag scanning

### 3.3.2 Zig-Zag 扫描电路设计

作为量化的后续操作, Zig-Zag 编码器接收量化模块产生的 12bits 有符号码位的输入, 同时输出 MPEG-2 MP@ML 语法允许的之字型或交替形扫描方式这两种不同的扫描方式产生的 12bits 有符号排序码位。考虑到这两种扫描方式都具有扫描顺序固定和 Zig-Zag 编码要具备连续性的特点。本文采用了如下的电路结构, 如图 3-18 所示: 首先每个时钟周期 12bits 的量化数据, 在 64 计数器的控制下以 64 次为一轮交替输入到两个具有乒乓结构的地址范围为[0, 63]的双口 RAM 中, 同时, Zig-Zag 地址发生器分别按照之字



型扫描的  $\{0, 1, 8, 16, 9, 2, 3, 10, 17, \dots\}$  或交替型扫描的  $\{0, 8, 16, 24, 1, 9, 2, 10, 17, 25, \dots\}$  的顺序，产成用来控制这两个乒乓结构的双口 RAM 的地址信号，双口 RAM 在地址信号的控制下连续的输出所需的之字型或交替型排序码流。

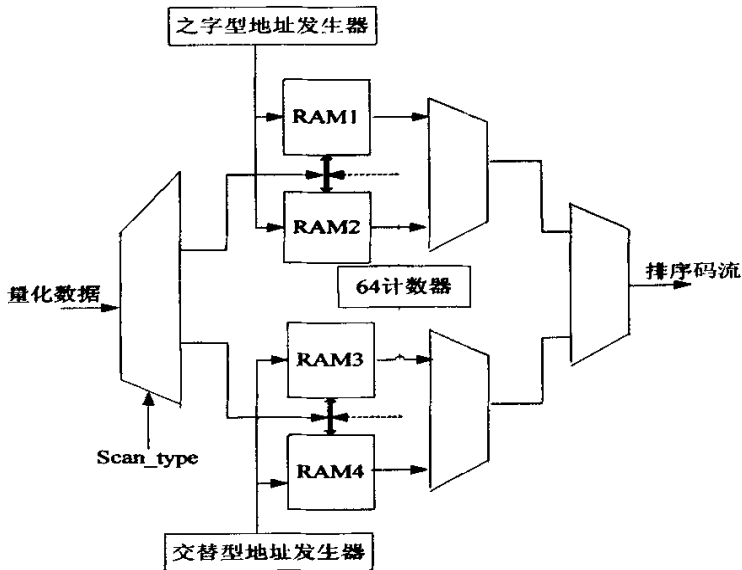


图 3-18 Zig-Zag 编码器结构图

Fig. 3-18 Zig-Zag coder structure

### 3.3.3 Zig-Zag 扫描电路的综合与仿真

我们用 Quartus II 对 Zig-Zag 扫描电路进行了综合和仿真，综合结果显示该模块共消耗资源 2064 个 LE，占整个器件资源的 19%。模块最高运行频率 102.29MHz，关键路径时延为 5.31ns。该模块的接口设计如图 3-19 所示，输入为经过量化处理的 12bits 数据，在 rdy\_in 有效的第一次 64 个周期延迟之后，rdy\_out 开始变成高电平，同时 zigzag\_out 开始连续输出 12bits 排序数据，其接口仿真结果如图 3-20 所示。图 3-21 给出了经过 Synplify Pro 7.6 综合器优化了的扫描电路部分 RTL 图。

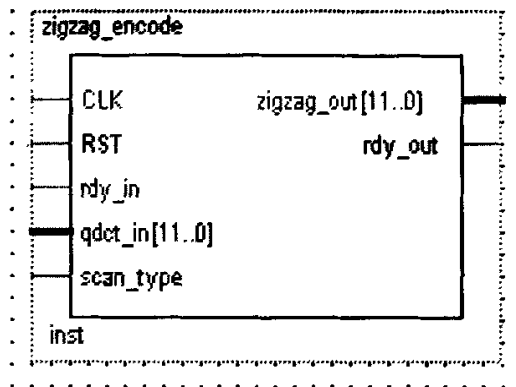


图 3-19 Zig-Zag 扫描电路接口图

Fig. 3-19 Interface of Zig-Zag circuit module

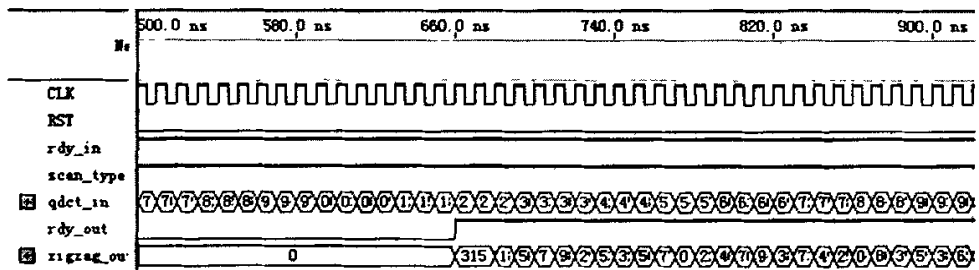


图 3-20 Zig-Zag 扫描电路接口波形图

Fig. 3-20 Timing diagram for Zig-Zag circuit

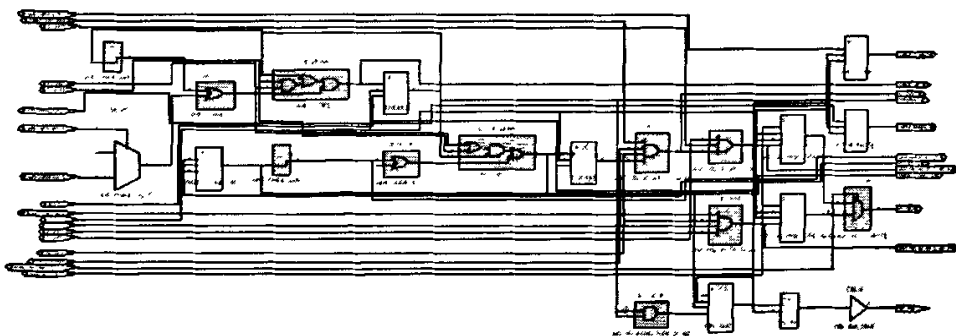


图 3-21 Zig-Zag 扫描电路部分 RTL 图

Fig. 3-21 The RTL map for one part of Zig-Zag circuit

### 3.4 可变字长编码电路设计

#### 3.4.1 可变字长编码算法描述

在 DCT 变换以及量化后,为进步一步减小平均传输比特率又采用了可变字长编码,又称熵编码(Entropy Coding),在 MPEG-2 标准中采用了游程长度编码(RLE)及哈夫曼(Huffman)编码,它们都是基于 DCT 量化系数统计特性所进行的无失真编码<sup>[36][37]</sup>。在本文当中这两种编码统称为可变字长编码。下面就这两种编码作以简要得介绍:游程长度编码是为了消除编码存在的统计冗余度,它的主要思想就是把相同值的连续串用一个值和串长来表示,即只传送非 0 系数和 0 系数的个数。游程长度编码最早用于二值图像的压缩编码。在游程编码中,1 个非 0 系数前连续系数的数量(游程)被编码,后随的是非 0 系数。这样,多个 0 系数和非 0 系数编成组,从而提高游程编码的效率<sup>[38]</sup>。而 Huffman 编码的基本思想是对于一组给定概率的符号,给出现概率较大的符号以长度较短的码字,概率小的符号以较长的码字,从而相对于等长编码,获得最短的平均码字长度<sup>[39]</sup>。

#### 3.4.2 可变字长编码电路设计

可变字长编码器接收 Zig-Zag 编码器模块产生的 12bits 的排序码流,同时异步输出未经整理的 MPEG-2 视频编码层码字。编码器输入同步输出异步的原因是:第一,游程编码 RLE 模块的输出为异步信号,而这种异步性又会延续到其它的结构;第二,Huffman 编码生成的是变长的码字,其 16bits 的输出速率完全依靠于输入的情况。由于这种异步性,因此,本文把可变字长编码器分割成游程编码器,Huffman 编码器这两个相对独立电路单元分别设计。

##### 3.4.2.1 游程长度编码电路设计

为了高度利用 DCT 系数的空间相关性,因此对于系数矩阵中的 DC、AC 系数都应有不同的编码方法。当每次对帧内宏块的 DC 系数进行编码时,宏块中第一个 8×8 子块中第一个 DC 系数完整的发送出去,接下来的 DC 系数发送本次帧内 DC 系数与上一个的帧内 DC 系数的差值。对于 AC 系数的编码,都应该编码为游程、幅度、符号的码流序列。另外当序列中的剩余系数都为 0 时,要用一个特殊的块结束标志 EOB。因此,我们就可以把游程长度编码器可以分成差分编码电路(DC 系数编码模块)、游程长度编码电路

(AC 系数编码模块)和符号编码电路(把差分值、游程值、幅度值转换成供后续 Huffman 编码器编码的中间符号模块)这三个独立的电路单元模块分别设计。

### (1) 差分编码电路

差分编码,即用每个 DC 值与前一个 DC 值(PRED)的差值来编码,是可变字长编码的第一步。它是一个简单的减法运算  $DIFF=DC-PRED$ ,由一个 12 比特的加法器和一个 12 比特的寄存器(用来存储前一个差分结果 DIFF)组成。

### (2) 游程长度编码电路

游程编码,用来记录 AC 系数中连续零的个数。MPEG-2 定义了连续零

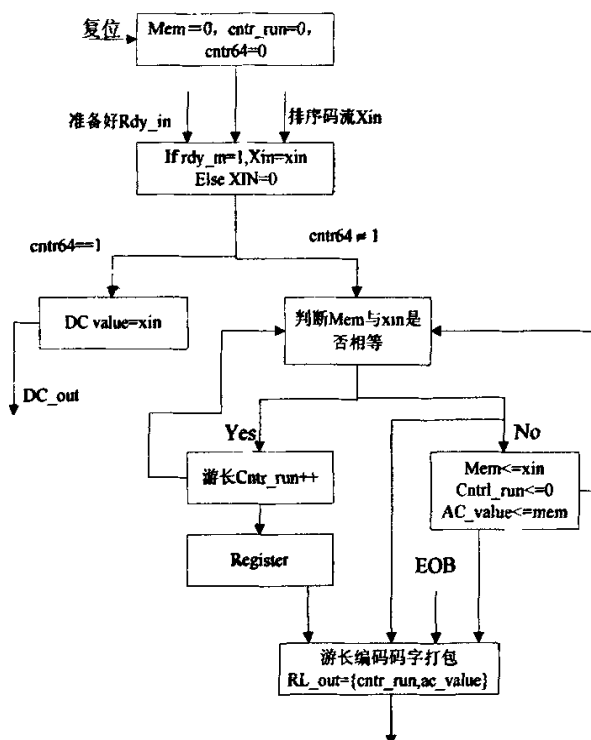


图 3-22 游程编码结构框图

Fig. 3-22 Run-length coder structure

个数的最大值为 15,即零计数器 4bits。当序列中的零个数超过 16,那么计数器会复位,并输出码对 Run/AC=15/0,这意味着在一个非零值之后有连

续 15 个零。此外，当矩阵的最后一个数是为零时，输出为一个特殊码对 0/0，表示最后一个矩阵元素是零，其结构框图如 3-22 所示。

(3) 符号编码电路

在进行 Huffman 编码之前，要将 DC 和 AC 系数转换成中间符号。DC 系数转换成 VLC(SSSS)和 VLI(VVVV)，VVVV 就代表差分结果 DIFF,SSSS 表示 DIFF 的值所需的二进制位数。AC 系数转换而成的中间符号为 VLC(RRRR/SSSS)和 VLI(VVVV)，其中 VVVV 代表各个 AC 值，SSSS 表示 AC 值所需的二进制位数<sup>〔4〕</sup>。其中 SSSS 的计算就是字长计算，在电路上本文采用选择器编码方法来实现。如图 3-23 所示，设字长值的二进制表示为 a3a2a1a0，将需要计算字长的数据分成高位和低位两部分。若高位含有 1，则 a3=1，且高位数据传入第二级；否则为 a3=0，低位数据传入第二级。若

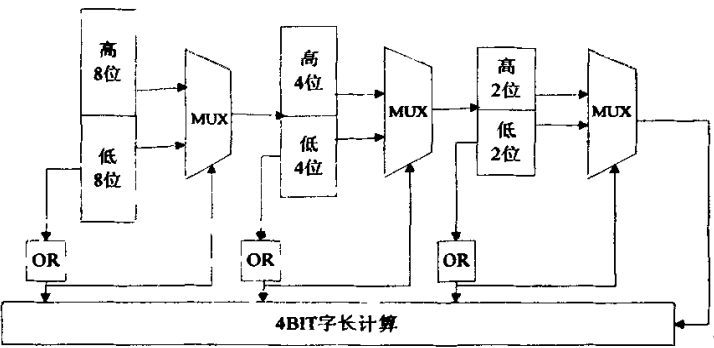


图 3-23 字长计算  
Fig.3-23 Length calculation

第二级高位含有 1，则 a2=1，且高位数据传入第三级；否则为 a2=0，低位数据传入第三级。以此类推，最后得到字长值为 a3a2a1a0。这种方法扩展性强，在当字长大于 16 时，速度优势相对于优先编码器法和直接编码法非常明显。所以本文采用选择器编码的方法来计算字长。VLC 编码的目的是保存 DC, AC 系数的有效位，丢弃无效位（高位的 0），生成 VVVV。当系数为正数时，VVVV 与系数值相等，当系数为负数时，VVVV 与系数的反码相等。所以有符号数要自减其符号位，即正数不变，负数减 1，将补码转换为反码，得到的数值的有效位个数为字长值，在根据字长将高位屏蔽（1 变 0，0 不变）。

由于符号编码电路与 Huffman 编码器的密切关系，因此在实际的电路设计中，符号编码电路的设计被整合到了 Huffman 编码器电路中。

3.4.2.2 哈夫曼编码电路设计

由于在 MPEG-2 TM5 中一共推荐了 16 个的哈夫曼编码表<sup>[4]</sup>，因此在设计哈夫曼编码电路的时候，采用了 MPEG-2 标准推荐的哈夫曼码表，通过查表的方式将中间格式的符号序列转化为哈夫曼码流输出，这样就简化了器件结构，降低了器件资源的开销。

在查表操作时，首先分别将 DC 系数和 AC 系数中间格式的序列符号的第一个字节分类码和游程/幅度作为查表的地址指针。然后分别设计两个 ROM 单元块用于存储 DC 系数和 AC 系数。DC 系数码表寻址空间为 0~11，详细结构见表 3-1。AC 系数码表寻址空间为 0~255，详细结构见表 3-2。

表 3-1 DC 系数寻址码表

Table3-1 Table for DC codes searching address

ROM 地址	分类码 (Size)	码长	差分幅值 (Diff Amplitude)
00	0	2	00
01	1	2	01
10	—	10	1111111110
11	—	11	1111111110

表 3-2 AC 系数寻址码表

Table3-2 Table for AC codes searching address

ROM 地址	游程/幅度	码长	码字 (ROM 内数值)
00	0/0 (EOB)	4	00100, 1010
01	0/1	2	00010, 00
0A	0/A	16	10000, 11111111110000011
0B0F	无意义	—	任意值
11	1/1	4	00100, 1100
12	1/2	5	00101, 11011
1A	1/A	16	10000, 11111111110001000
1B20	无意义	—	任意值

DC 系数和 AC 系数的 Huffman 编码的码长是随编码状态而变化的，其中 DC 系数的码长最多为 11，AC 系数的码长最多为 16。为了使系统资源可

以共用，将所有码字的码长统一为 16，不足 16 的码字，高位对齐，作末尾补零处理。且后续码流输出过程中需要码长信息，所以，又将 Huffman 码字的码长信息用 5 位表示，并且加在码字的前面。这样 ROM 的宽度统一设计为 21Bits。最高 5 位表示码长，后面 16 位表示 Huffman 编码的码字。

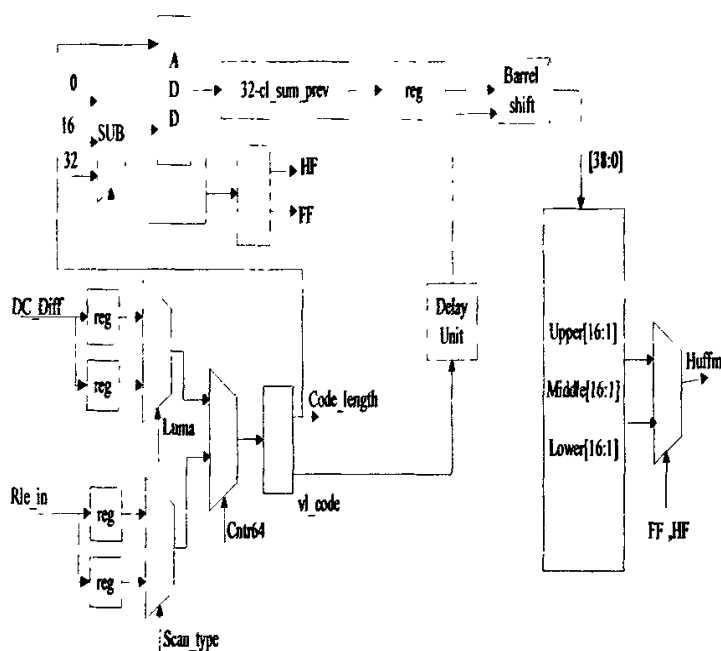


图 3-24 Huffman 编码结构图

**Fig. 3-24** Huffman coder structure

值得注意的是, DC 系数和 AC 系数的幅值码均由第二个字节表示, 且都采用补码形式输入, 经过 Huffman 编码器之后将转换为二进制的自然码形式输出, 转换后的幅值码的位数由第一个字节的尺寸部分控制。其哈夫曼编码器(含符号编码器)具体结构图如图 3-24 所示。

### 3.4.3 可变字长编码电路的综合与仿真

利用 Quartus II 对可变字长编码模块的结构进行编码并逻辑综合后,共消耗资源 3730 个逻辑单元。占总逻辑单元 21%, 占用嵌入式内存 (EAB) 4032 比特, 占内存总数的 16%, 关键路径时延为 6.74ns, 远远小于文献[38-41]中的关键路径时延, 最高工作频率可达到 116MHz。其中游程长度编码电路

接口图、时序图如图 3-25 和 3-26 所示。Huffman 编码电路接口图、时序图如图 3-28 和 3-29 所示。图 3-27 和图 3-30 分别给出了经过 Synplify Pro 7.6 综合器优化了的游程长度编码和 Huffman 编码电路部分 RTL 图。

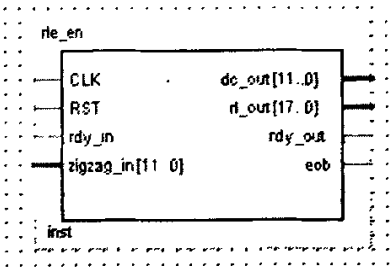


图 3-25 游程长度编码电路接口图

Fig. 3-25 Interface of run-length coder circuit

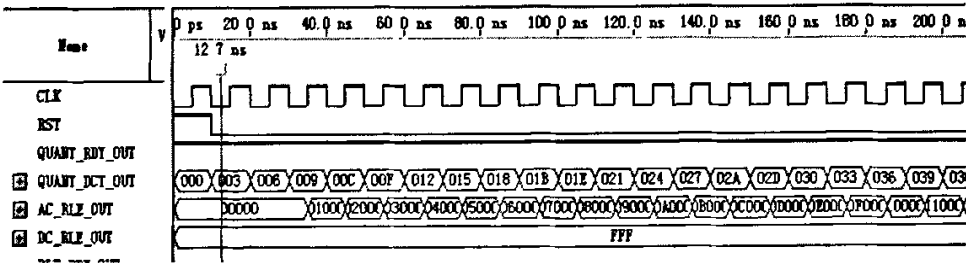


图 3-26 游程长度编码波形图

Fig. 3-26 Timing diagram for Run-length coder

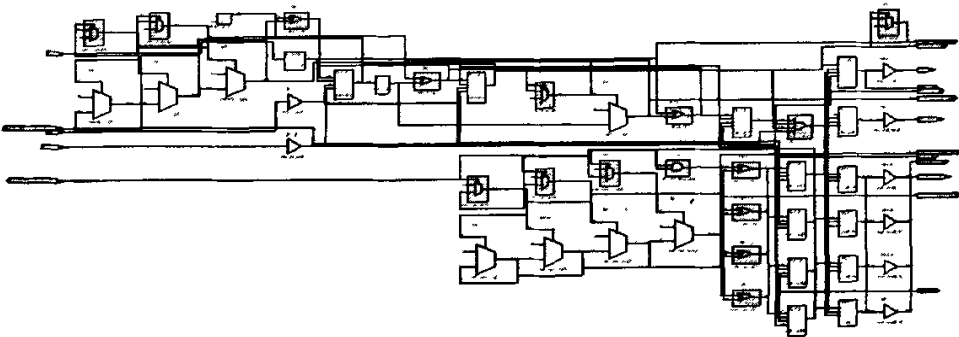


图 3-27 游程长度编码电路部分 RTL 图

Fig. 3-27 The RTL map for one part of Run-length coder circuit



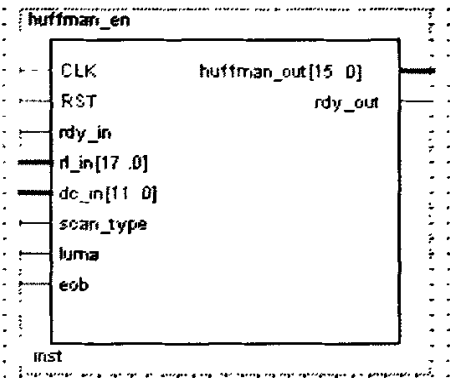


图 3-28 哈夫曼编码电路接口图

Fig.3-28 Interface of huffman coder circuit

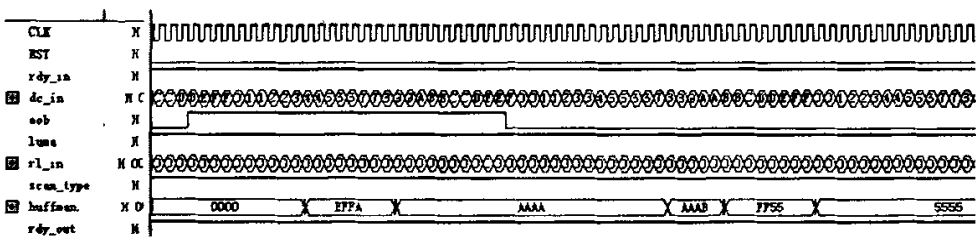


图 3-29 哈夫曼编码电路波形图

Fig.3-29 Timing diagram for huffman coder circuit

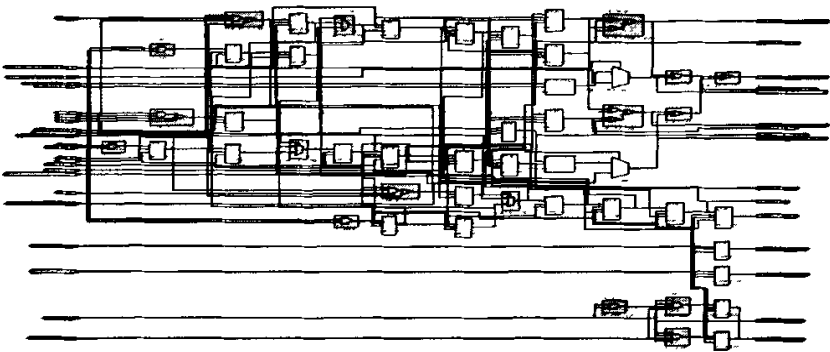


图 3-30 哈夫曼编码电路部分 RTL 图

Fig.3-30 The RTL map for one part of huffman coder circuit

### 3.5 本章小结

本章主要研究了适用于山区、偏远农村现代远程教育的经济型 MPEG-2 运动图像编码器 IP 核中的块处理单元，对于其中的二维 DCT 模块和可变字长编码模块通过改进算法、优化结构和充分利用 FPGA 自身的资源特性等方法，有效地降低了模块关键路径上的时延和逻辑资源消耗量，使得设计在速度和资源两方面均达到了较优的状态，可满足实时视频编码的要求。

## 4 运动估计单元 IP 核设计

运动估计是压缩编码过程中的关键步骤，目的就是要在可接受的运动估计运算复杂度上将编码信息量减少到最低，据统计，运动估计的计算量占到了整个编码计算量的 60%以上，要实现实时的运动图像编码，运动估计性能的好坏直接影响到整个编码器的编码质量与压缩比率<sup>[42][43]</sup>。

目前，基于块的运动估计方法主要有全搜索法(Full Search)，二维对数搜索法，三步搜索法(3SS)，四步搜索法(4SS)，交叉搜索法(CS)，菱形搜索法(DS)等。在众多的运动估计算法中，全搜索块匹配算法数据流规则且没有复杂的动态反馈和决策逻辑，非常有利于硬件结构实现，而且其匹配准确率比其它的快速块匹配算法高<sup>[44][45]</sup>。而由全搜索算法计算量大导致的计算速度问题，可通过并行处理单元、流水线结构，这种面积换速度思想就能很好地解决计算质量和计算速度问题。为了便于实现，本文中采用了固定的块大小，以  $8 \times 8$  子块为单位采用改进的全搜索法进行运动预测方法。

### 4.1 全搜索块匹配运动估计算法

在运动图像编码技术中，运动估计被用来消除图像序列间的时间冗余。如图 4-1 所示，它将当前帧划分成  $M \times N$  个宏块，对于每一宏块，在先前帧找到对应的宏块位置，将左上角点定义为零点坐标，任意宏块其左上角点的坐标在水平为  $-p_x \sim p_x$ ，垂直为  $-p_y \sim p_y$  的范围内称之为候选块，由  $-p_x \sim p_x$  和  $-p_y \sim p_y$  构成的区域称为搜索区域。计算搜索区域中与参考块匹配最好的候选宏块，其行列坐标即定义为运动向量 (Motion Vector, MV)<sup>[46][47]</sup>。令  $x(i, k)$  表示当前帧在相对坐标  $i, k$  位置的像素的亮度值， $y(i, k)$  表示先前帧在相对坐标  $i, k$  处的像素的亮度值， $p_x = p_y = p$ ，全搜索的匹配标准可表示为如式 (4-1) 的形式：

$$SAD(m, n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |x(i, j) - y(i + m, j + n)| \quad (4-1)$$

$$MV = \arg\{\min SAD(m, n)\}, -p \leq m, n \leq p$$

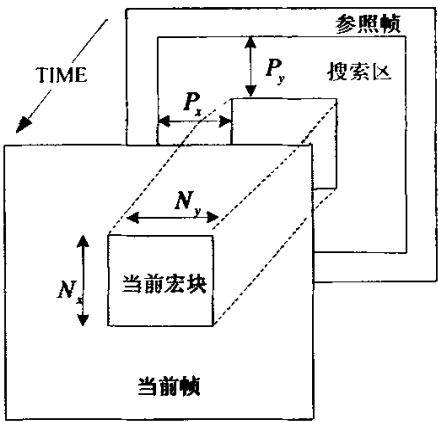


图 4-1 全搜索块匹配运动估计算法

Fig. 4-1 Full-search block matching motion estimation algorithm

由式(4-1)可以看出，在全搜索算法中当前块是与搜索区域内所有的宏块作比较，因此共需要比对  $(2p+1) \times (2p+1)$  个像块，总共有  $N \times N \times (2p+1) \times (2p+1)$  个像素的比较，以找到最佳匹配块。

4.2 全搜索块匹配运动估计电路设计

图 4-2 描述了全搜索运动估计算法的硬件结构，该结构主要包括 PE 运算阵列 (PE Array)、运动向量判决单元、搜索窗缓存、当前块缓存、地址产生单元 (AGU) 以及控制单元, 其中两个缓存是用来分别存放当前块和搜索窗的数据。

运动估计是由控制单元、PE 运算阵列和运动向量判决单元协同完成，控制单元将当前块和搜索窗数据从外部存储器中取出放在内部的缓冲区中，并设置搜索的内部缓冲区起始地址。并且为了降低 I/O 端口数据吞吐的压力，在运动估计之前首先将缓存数据的进行初始化。在运动估计开始后，充分利用连续两个块及其搜索窗数据的交叠，只更新处于交叠区域外的数据，进一步缓解 I/O 数据带宽的压力此外，为了提高运动估计的效率，将缓冲区在原来基础上多增加了一列宽度的存储空间，这多出来的列在本次运动估计进行的同时载入下一次估计要用到的位于交叠区外的像素值，使前后两次运动估计之间不需要经过数据预载的间隔。

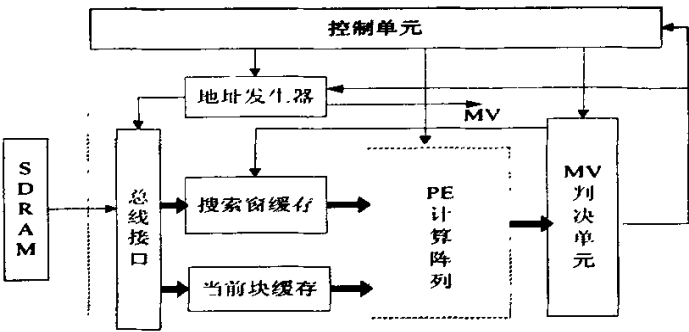


图 4-2 运动估计硬件结构系统框图

Fig. 4-2 Motion Estimation circuit structure

PE 运算阵列完成搜索运算，输出相对于搜索窗的偏移向量及 SAD 值。控制单元通过运动向量判决单元的反馈信号取得待搜索的点的坐标，再经过地址映射从缓冲区取得计算的数据输入 PE 运算阵列，PE 阵列运算计算 SAD 值，并将计算结果送入运动向量判决单元，当搜索完毕后，运动向量判决单元比较 255 个 SAD 值，并记录其中最小的一个 SAD 值所对应的偏移地址，便获得该块的最佳匹配点。下面就运动估计电路的各个模块设计作以说明。

4.2.1 搜索窗缓存电路设计

MPEG-2 标准的 TM5 (Test Model 5) 中提到，在运用全搜索运动估计时，搜索窗范围为  $[-2^{f\_code+3}, -2^{f\_code+3}-1] \times [-2^{f\_code+3}, -2^{f\_code+3}-1]$ ，其中  $0 \leq f\_code \leq 7^{[1]}$ 。为了简化硬件结构，在本文中  $f\_code$  取为 0，即得到搜索窗的范围为  $[-8, 7] \times [-8, 7]$ ，那么就有  $15 \times 15$  个候选块，而块的大小为  $8 \times 8$ ，即每个块有 64 个像素点。则搜索窗相应像素点数为  $22 \times 22$ ，即 484 个像素点，每个像素点都是用一个字节表示其灰度值。如此多的像素数据，如果用一个存储单元和一组地址指定一个像素值数据的话<sup>[14]</sup>，那么搜索窗缓存为了表示所有的像素值，则需要 484 个存储单元和 484 组相应的地址线，这无疑大大增加了像素数据流的控制复杂度和器件资源。

因此，本文在借鉴文献[45]中的蛇形数据缓存结构的基础上，提出了一种新颖的搜索窗缓存结构，该结构能够将参考块中的任意一行像素值（八个像素）的数据仅仅由一个存储器单元储存，并由一组地址指定，故有效减少了存储器地址数目和储存器输入引脚数目，该结构与蛇形数据缓存架构比较，本文提出的结构功耗更低，其具体内部结构图如图 4-3 所示。

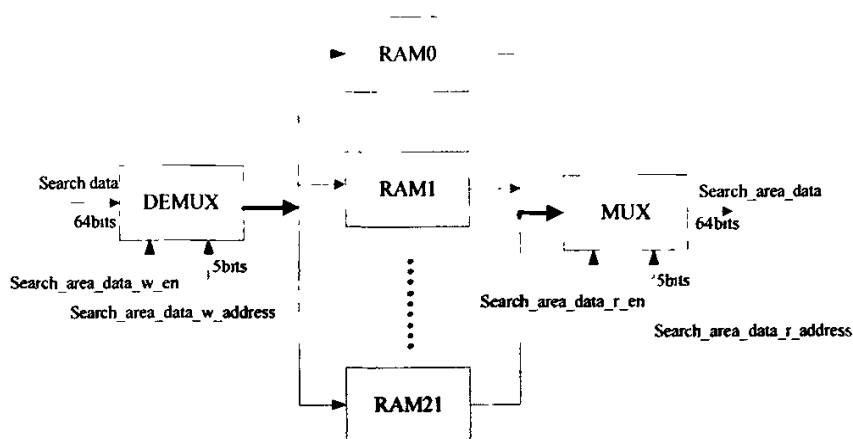


图 4-3 搜索窗缓存结构图

Fig. 4-3 Searching area buffer structure

因为当搜索块大小  $N=8$ , 位移量  $p=7$  的时候, 每一次获得最佳匹配块的操作, 共需要搜索窗中  $22 \times 22$  个像素值数据, 为了提高运算速度、减少外存与搜索窗缓存的数据交换的频率, 因此设计搜索窗缓存大小为 22 个储存单元, 由 5bits 的 `search_area_data_w_address`、`search_area_data_r_address` 地址信号分别指定。每一个存储单元的大小为 8 字节, 即每个储存单元存储八个像素值数据, 每组存储器地址代表八个像素的数据; 故存储器的大小为  $8 \times 22$  字节, 可以存储  $8 \times 22$  个像素值。

第一次存储时, 图 4-1 中搜索窗中的第 0 行  $(0, 0)$ 、 $(0, 1) \dots (0, 7)$  位置的像素值由 RAM0 存储; 第一行的  $(1, 0)$   $(1, 1) \dots (1, 7)$  位置的像素值由存储 RAM1 存储; 依次类推, 第二十一行的  $(21, 0)$   $(21, 1) \dots (21, 7)$  位置的像素值由存储 RAM21 存储。当完成一次 SAD 计算时, 搜索窗的位置开始向左移动, RAM0 中不再存储  $(0, 0)$  坐标的像素值, 取而代之的是新增加的搜索点  $(0, 8)$ , 同时 RAM0 中  $(0, 1) \dots (0, 7)$  的数据在 RAM0 中逻辑左移。同理 RAM1~RAM21 也应该完成同 RAM0 相似的工作, 不同的是, 它们所新存储的搜索点不同罢了。

上述对搜索窗像素值数据流的操作, 一方面通过简化地址线的方法降低了数据流操作的复杂度, 但另一方面, 像素值数据的预载和对齐却增加了搜索窗缓存处理的时钟周期数, 为了使存储单元能够可同时在不同的地址上进行读和写操作, 因此, 搜索窗缓存在结构上又多增加了一列的 22 字节的存

储单元。这样以来，多出来的存储单元就可以在本次运动估计进行的同时载入下一次估计要用到的位于交叠区外的像素值，使前后两次运动估计之间不需要经过数据预载的间隔，从而使得 SAD 的计算时间比文献[45]节约了 18 个时钟周期。

#### 4.2.2 当前块缓存电路设计

当前块缓存的设计中，本文用 64 字节的 FIFO 队列表示  $8 \times 8$  当前块像素点的数据流结构，其每行 8 个像素点值以串型移位的方式同时输入到 PE 运算阵列这里来进行 SAD 计算，下一个周期又输入下一行的像素点，直到 8 个时钟周期后又回到当前块的第一行像素点，准备开始下一次 SAD 计算。

#### 4.2.3 PE 运算阵列电路设计

由式(4-1)全搜索的匹配标准可以看出，运动向量的计算过程是：将当前块和搜索窗中的  $8 \times 8$  大小的块群做比较，得到最佳匹配块，最后通过计算当前块与最佳匹配块的坐标差值得出运动向量；该算法涉及到了大量的数学运算，但是只有少量的输出结果，因此全搜索块匹配算法则是属于计算负载计算，这种计算方式十分适合运用脉动阵列架构来实现运算<sup>[47][49]</sup>。

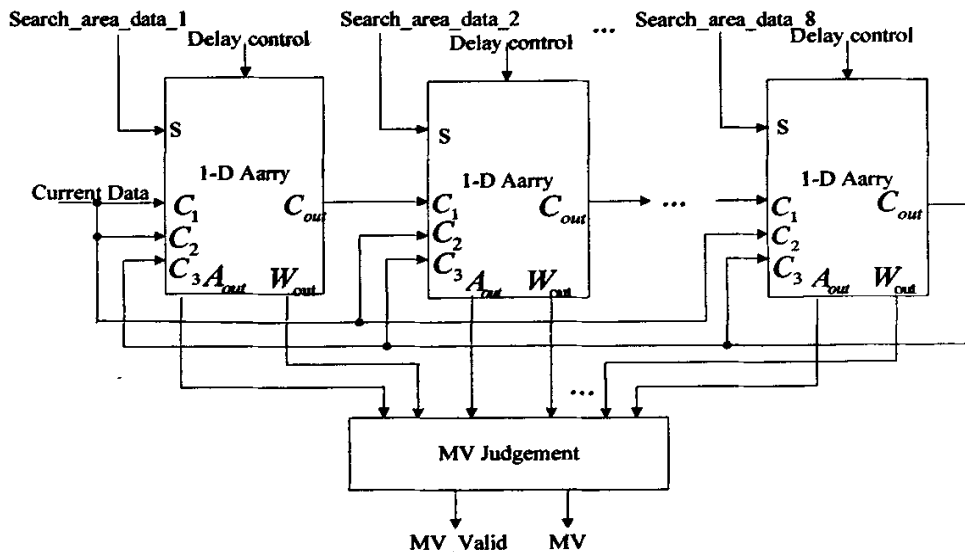


图 4-4 脉动运算阵列

Fig. 4-4 PE Array structure

基于这种理论基础,本文中的 PE 运算阵列也是基于一维脉动阵列(AB1 型一维脉动阵列)来设计的,具体结构图如图 4-4 所示。在运算阵列中,串接了 8 个具有相同结构一维脉动阵列(1-D Array),串接的好处就是该 8 个 1-D Array 可以同时处理搜索窗中八个相邻的参考块数据的 SAD 运算,这样便通过面积换速度的思想获得了较高的运算速率,其内部结构图如图 4-5 所示;S 端口接受来自搜索窗缓存中的参考块像素值数据,同时,因为在一维脉动阵列(AB1 型)进行运算时,有时候需要偏斜地输入数据,所以设计两个端口  $C_1$ ,  $C_2$  分别接受带偏斜的当前块像素值数据和不带偏斜的当前块像素值数据;  $A_{out}$  为参考块和当前块的 SAD 输出端口,  $W_{out}$  则输出参考块和当前块的坐标差值(即运动矢量)。同时,由于在全搜索块匹配算法中,当前块的像素数据值需要反复的使用,故  $C_3$  为重复使用的当前块的输入端,  $C_{out}$  则用来串接 8 个 1-D Array 实现当前块数据流的逻辑左移,并且在逻辑移动过程中,  $C_{out}$  需要的时延由 Delay Control 来控制产生。Search data 以及 Current data 均为 64bits。运算后得到的运动向量经过运动向量判决单元比较判断后由 MV 端口输出,  $MV\_valid$  则表示目前输出的运动向量是否有效,是否找到最佳匹配块。

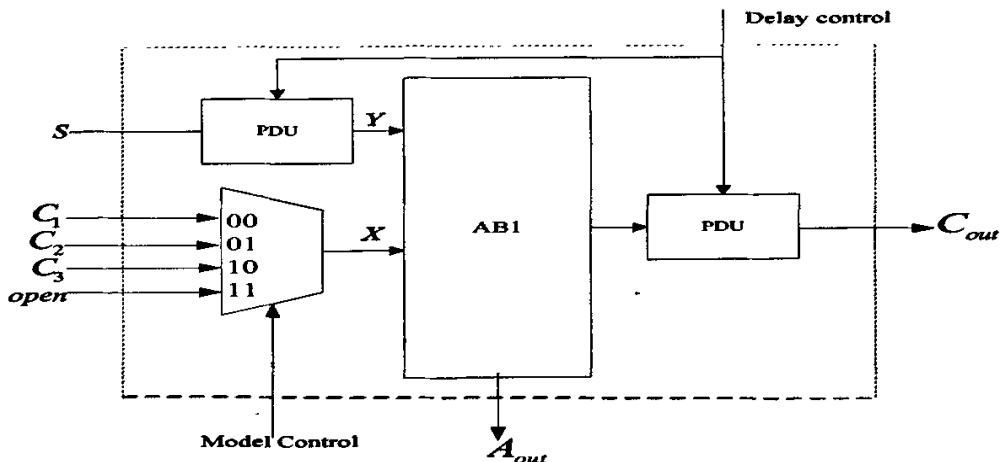


图 4-5 一维阵列结构图

Fig. 4-5 1-D Array structure

本文所提出的运动估计硬件结构中包含 8 个并行的运算处理单元(1-D Array)。每个 1-D Array 主要由一维脉动阵列(AB1)、PDU(Programmable



Delay Unite)、一个多路选择器以及其中的基于流水线技术的延时寄存器组成,如图 4-5 所示。8 个 1-D Array 总共包括个 256 个算术逻辑单元 (ALU) 和 8 个累加器 (ACCM), 这是整个运动估计电路中最主要的并行计算资源, 用增加芯片面积的方法获得了较高的计算速度。

#### 4.2.4 运动向量判决电路设计

运动向量判决单元包括两个 SAD 比较模块以及两个运动向量选择模块, 如图 4-6 和图 4-7 所示, 每 11 个时钟周期 1-D Array 同时输出八个 SAD 值, 故分成两组比较模块以满足同样在 11 个周期内完成 SAD 的比较, 避免流水线额外的延时。第一组模块处理 0~3 号 1-D Array 处理单元输出的 SAD

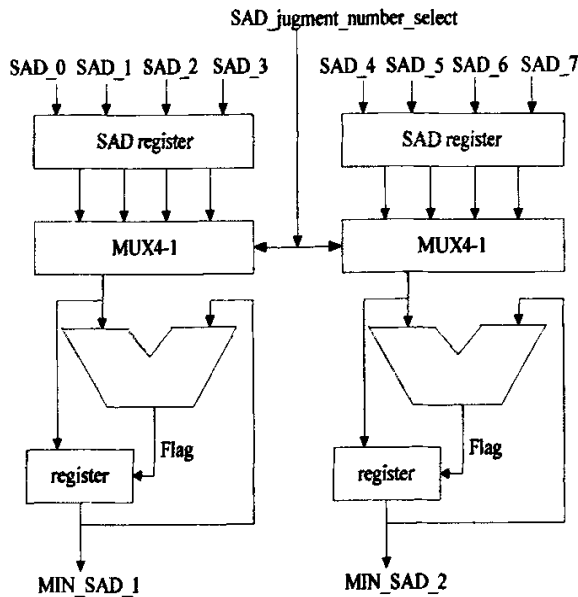


图 4-6 SAD 比较模块

Fig. 4-6 SAD judgment module

比较; 第二组模块处理 4~7 号 1-D Array 处理单元输出的 SAD 比较。现在以第一组为例, 简要介绍其工作方式: 在控制信号 'SAD\_jugmentnet\_number\_selcet' 的控制下, MUX4-1 (四选一选择器) 首先选择 'SAD\_1' 与 'SAD\_0' 比较, 并记录下两者之中的值小者; 然后 MUX4-1 再选择 'SAD\_2' 与 'SAD\_1'、'SAD\_0' 之间值小者比较, 然后得到一个 'SAD\_0'、'SAD\_1'、

SAD\_2 当中的最小值;然后 MUX4-1 再选择 SAD\_3 与 SAD\_2、SAD\_1、SAD\_0 之间值小者比较, 然后得到一个 SAD\_0、SAD\_1、SAD\_2、SAD\_3 当中的最小值 MIN\_SAD\_1。

当所有 225 个候选块的 SAD 完成计算和比较后, 这两组 SAD 比较模块各自产生一个最小 SAD 值, 即 MIN\_SAD\_1 和 MIN\_SAD\_2, 并被送入到运动向量选择单元。在运动向量选择单元中, 则只需在 MIN\_SAD\_1 和 MIN\_SAD\_2 之间, 选择较小 SAD 所对应的偏移值即可, 如图 4-7 中所示。这也就是最终的运动向量值, 因为水平和竖直方向的偏移量本身就是在  $[-8, 7]$  范围内的有符号值, 无须额外偏置。在运动向量选择单元在比较过程中, 如果比较的两个 SAD 值相同则以偏移较小的 SAD 为准。

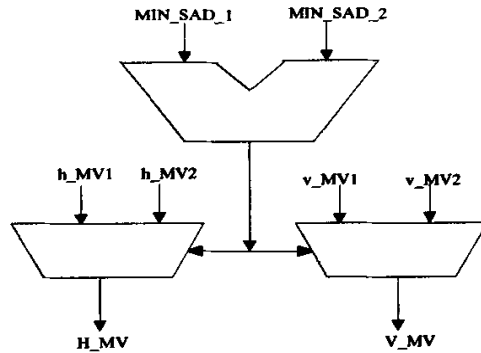


图 4-7 运动向量选择框图

Fig. 4-7 Motion vector choice diagram

#### 4.2.5 控制单元电路设计

控制单元用来控制 PE 运算阵列 (PE Array) 运算, 搜索窗缓存数据的写入, 与地址产生单元正确地输出像素地址, 以协调三个单元之间的动作, 使得运动估计电路能够正常工作。控制单元工作方式为一有限状态机 (Finite State Machine, FSM), 其接口示意图如图 4-8 所示。

图 4-8 中, 信号 `r_en_current_block` 为告知地址产生单元通知外存输入当前块数据; 信号 `r_en_search_area_data_1` ~ `r_en_search_area_data_8` 的作用是告知地址产生单元, 通知搜索窗缓存输出某个特定的参考块像素值。同时由地址产生单元输出地址信号 `search_area_data_1_r_address` ~ `search_area_data_8_r_address`, 以确定参考块数据在搜索区域中的坐标位置。

信号 out\_search\_area\_data\_w\_en 则为通知外存输出搜索窗像素值数据以便写入搜索窗缓存。search\_area\_data\_w\_en 为使能信号，使得搜索窗缓存可以写入数据，数据写入的位置由 search\_area\_data\_w\_address 决定。

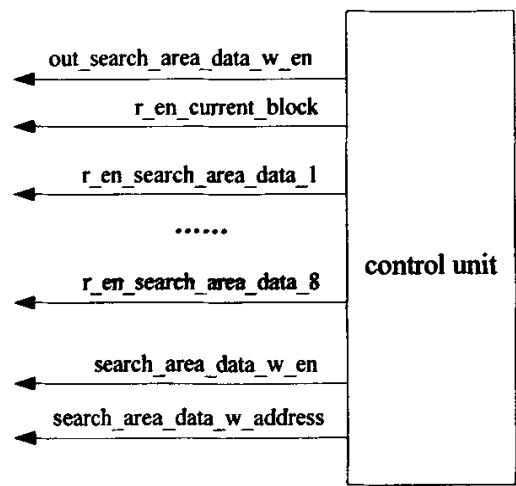


图 4-8 控制单元接口示意图

Fig. 4-8 Sketch map of control unit interface

4.2.6 地址产生单元电路设计

由于本论文提出的搜索窗缓存架构，有效地减少了存储器地址数目，因而地址产生单元输出的地址所需要的引脚数目同样也被减少。故与之对应的地址产生单元则是为一架构简单、有效的的设计。

地址产生单元(Address Generation Unit,AGU)的功能，在于产生信号告知搜索窗缓存应该输出那一部份的数据给 PE 运算阵列，其具体结构图如图 4-9 所示。

由于搜索窗缓存可以同时输出八个参考块像素值数据：search\_area\_data\_1~search\_area\_data\_8，则 AGU 结构中八个单元分别输出地址信号 search\_area\_data\_1\_r\_address~search\_area\_data\_8\_r\_address 告知搜索窗缓存 search\_area\_data\_1~search\_area\_data\_8 的输出数据分别该为何。

当 AGU 工作的时候，search\_area\_data\_1\_r\_address 依次分别输出 (0, 0) ~ (7, 7) 坐标地址、search\_area\_data\_2\_r\_address 输出 (1, 0) ~ (8, 7) 坐标地址，依次类推 search\_area\_data\_8\_r\_address 则应该输出 (7, 0) ~

(14, 7) 坐标地址。

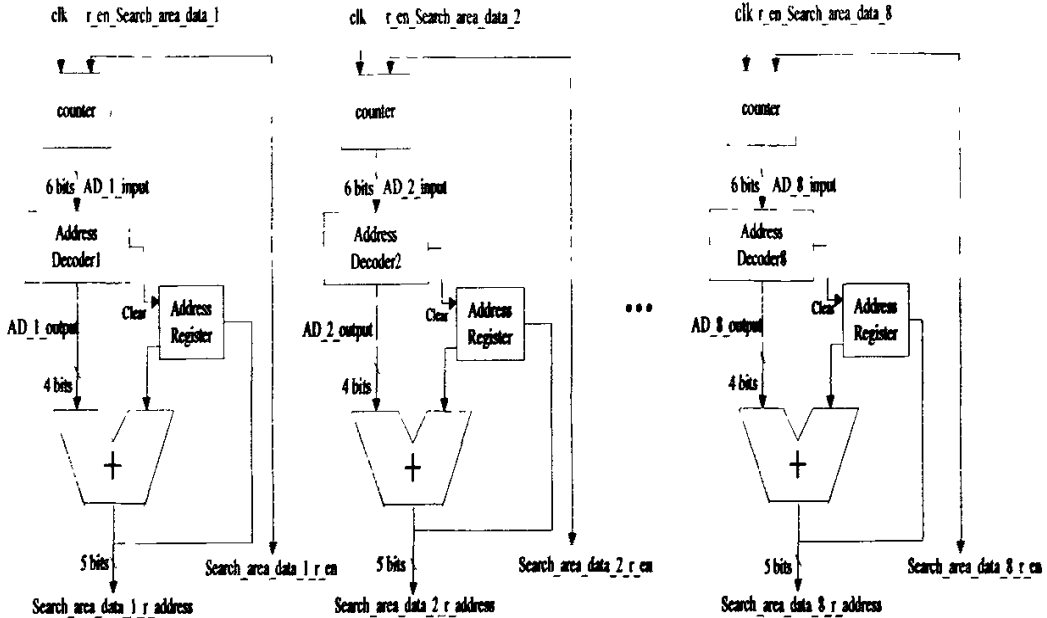


图 4-9 地址产生单元结构图

Fig. 4-9 Address generation unit structure

在地址产生单元内部，有八个计数器与八个地址编码器，计数器为一向上计数器，地址编码器则由计数器的值决定怎么输出，并于上一次地址相加的到此刻的地址，同时存入到暂存器中以便累加下一次地址编码器的输出值，得到下一次的输出地址。

### 4.3 运动估计模块的综合与仿真

本文的运动估计硬件结构的内置存储单元，主要由两个缓冲区和流水线寄存器等组成。其中搜索窗缓存共有 1408bits，当前块缓存共有 512bits。另外还有 PE 运算阵列中 176bits 的 ALU 流水线寄存器，224bits 的 SAD 流水线寄存器和 28bits 比较单元寄存器。所有这些存储或寄存单元的内置存储器总共不到 5Kb 分布基本和 MPEG-2 TM5 建议的硬件结构一致<sup>[7]</sup>，内置存储器要求都很少。该模块在 VHDL 和原理图混合输入方式完成设计，并通过功能仿真，在 Quartus II 综合下，模块占有 17054 个 LE，系统时钟频率达到

35MHz，及每秒能处理超过 56000 个宏模块，而 MPEG-2 MP@ML 所需要的处理能力仅为 40500 宏块/秒，完全能够达到实时编码的要求，该模块的主要处理单元实现仿真图和结构图如图 4-10、图 4-11、图 4-12、图 4-13、图 4-14 所示。

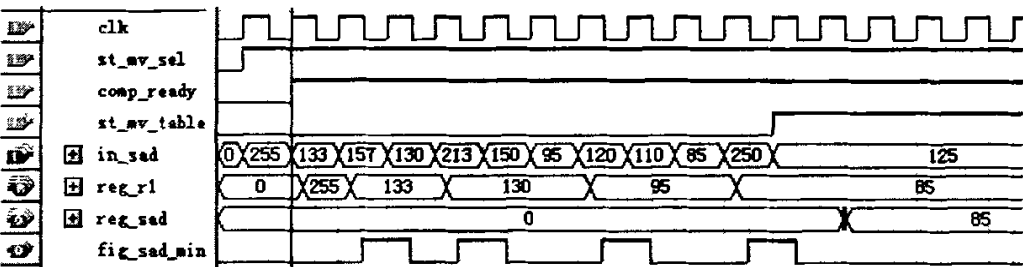


图 4-10 运动向量判决模块时序图

Fig. 4-10 Timing diagram for judgment MV

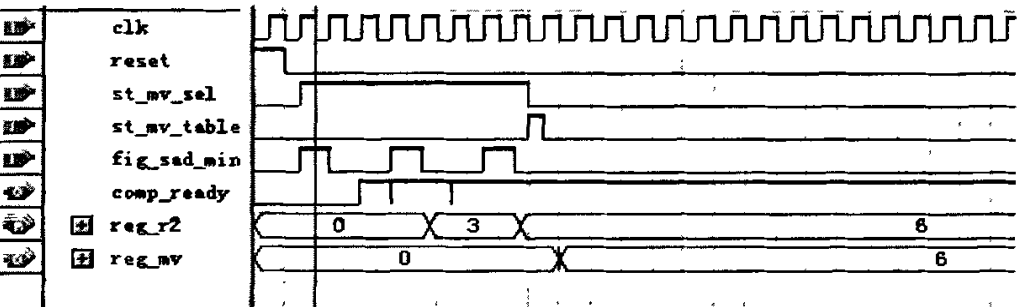


图 4-11 MV 选择时序图

Fig. 4-11 Timing diagram for MV choice

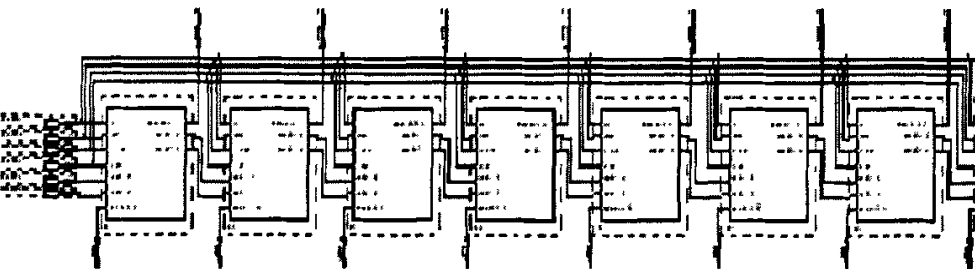


图 4-12 PE 运算阵列

Fig. 4-12 PE computational arrays

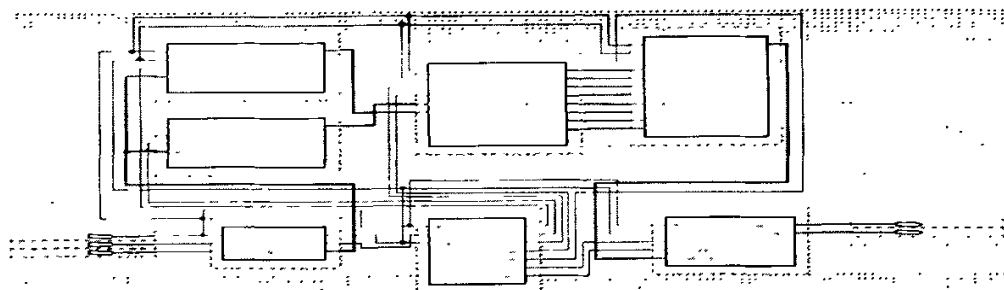


图 4-13 顶层模块结构

Fig. 4-13 Motion estimation top-level module

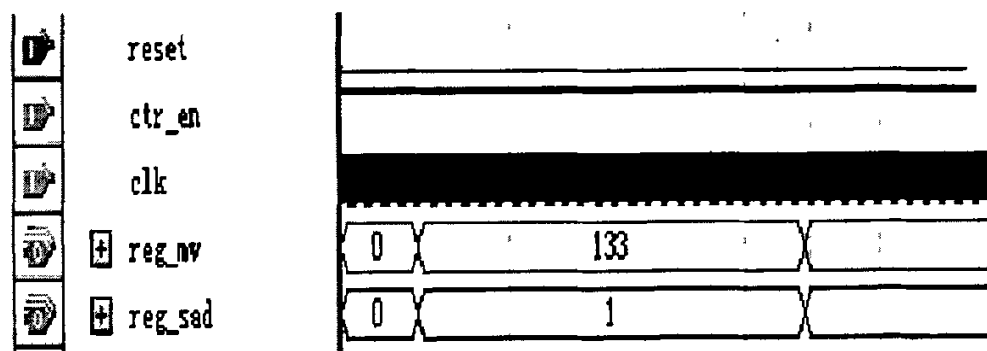


图 4-14 顶层仿真波形

Fig. 4-14 Timing diagram for motion estimation top-level module

#### 4.4 本章小结

综合上述设计和分析,对于适用于山区、偏远农村现代远程教育的经济型 MPEG-2 运动图像编码器 IP 核中的运动估计模块的设计,本文提出了一种高度并行和多流水线处理的硬件结构,实现了 MPEG-2 视频部分的全搜索块匹配运动估计算法,该硬件结构能实时地通过全搜索块匹配运动估计来搜索每个像素块最佳的匹配运动向量,并通过改进搜索窗缓存、PE 运算阵列使得在增加少量的芯片面积条件下达到了每秒能处理 56000 个宏块的速度,大大高于 MPEG-2 MP@ML 所规定的 40500 宏块/秒处理速度<sup>[7]</sup>,可满足实时视频的运动估计编码的要求。

## 结 论

本论文是以国家 863 课题——“缩小数字鸿沟—西部行动”《基于国产软硬件的多通道网络教育关键技术与应用研究》(课题编号: 2003AA116060) 基金项目为依托, 结合现代远程教育需求和运动图像编码技术, 设计了一种基于 FPGA 的适用于山区、偏远农村现代远程教育的经济型 MPEG-2 运动图像编码器的 IP 核, 然后将设计结果在 ALTERA 公司的 Cyclone EP1C12Q240C8 器件进行了验证, 并将验证结果与 MPEG-2 TM5 中的校验模型比较, 证实了该设计的正确性和优质性。

笔者认为该设计最大的特点就是: 在针对山区、偏远农村现代远程教育的特定应用领域上, 摒弃了传统的、昂贵的 MPEG-2 编码器 ASIC 芯片, 将整个编码器完全搭建在价格相对低廉的 FPGA 平台上, 不断的改进算法, 优化结构, 合理地利用硬件资源, 最大限度的简化了编码器电路构造, 并以单片 FPGA 实现了整个 MPEG-2 编码器的设计, 真正实现了现代远程教育中视频编码系统的小型化、经济化。

在本次设计中个人有创新性的工作有:

(1) 在二维 DCT 模块的设计中, 本文在基于行列分离算法的基础上, 充分的利用了 FPGA 的 EAB 资源, 使得消耗的逻辑资源数有了大幅度的降低, 并且通过引入了专用的加法器设计, 使得模块的关键路径上的时延减少到 5.65ns, 有效地提高了系统运行的最大工作频率。

(2) 在可变字长编码模块的设计中, 虽然本文设计的模块在资源利用率上与传统的可变字长编码模块相比虽然没有较大提高, 但是在速度上, 依靠算法和结构的优化, 比如对设计中的关键路径一字长计算单元进行分析, 采用选择器编码单元来实现, 解决了 Huffman 编码模块的速度瓶颈问题。

(3) 在运动估计的设计中, 本文提出了一种高度并行及紧凑流水线的运动估计硬件的大规模集成电路结构, 可完全满足 MPEG-2 MP@ML 的要求, 通过分析可知, 该结构只需相对较少的内嵌存储器和很低的系统运行时钟即可获得实时视频处理能力, 非常适合于低功耗应用领域, 而且通过改进搜索窗缓存、PE 运算阵列等使得在增加少量的芯片面积条件下达到了每秒能处理 56000 个宏块的速度, 相比于 MPEG-2 视频部分标准的运动估计硬件结构, 本文的运动估计硬件结构在硬件复杂度上增加很少的情况下, 获得了极高的速度增益。

同时,对于本文的 MPEG-2 运动图像编码核心部分 VLSI 的设计,应用了 VHDL 硬件描述语言自顶向下设计,并在 ALTERA 公司的 EDA 工具 Quartus II 6.0 平台上进行了逻辑综合及功能和时序仿真。综合和仿真结果表明,基于 FPGA 的 MPEG-2 运动图像编码芯片消耗较少的器件资源,达到了较高的工作频率,在速度和资源利用率方面均达到了较优的状态,可满足实时 MPEG-2 运动图像编码的要求。对于其中的二维 DCT、量化/逆量化、可变字长编码和运动估计模块的设计,可以作为独立的软 IP 核,根据需要略加修改就可以应用到其他含相关算法的图像芯片设计中。

由于该课题的实践性较强,受限于具体应用开发环境和个人的知识背景,尚存在一些不足之处有待进一步完善。首先,硬件逻辑设计仅仅是编码器 IP 核设计的第一步,虽然借助 EDA 工具完成了综合与仿真,并将该 IP 核下载到了相应的 FPGA 器件中去了。但是由于硬件系统尤其是高频系统的复杂性,则需要在更先进的工具上用大量的数据进行测试和验证,并与其他配套设备进行联机调试,才能最大限度地找出其中的设计缺陷,并不断完善,以至于最终得到真正能运用到山区、偏远农村现代远程教育中去的 MPEG-2 编码器 IP 核。



## 致 谢

本论文是在导师李众立教授的悉心指导下完成的。从选题、研究、实验，一直到论文的最后完成，其中任何一点成绩的取得都离不开李老师的关怀和点拨。李老师亲切和善的教学态度、深厚渊博的知识体系、耿直真诚的个性和严谨认真的科研作风都给我留下了十分深刻的印象。这段学习生活不仅使我明白了如何做一名合格的科研工作者，并将在我今后的人生路途中继续起着指导作用。在将近三年的研究生生活中，李老师不仅在学习上对我们耐心指导、严格要求，而且在生活上也给我们很多关心、理解和帮助。他平易近人的生活作风、渊博的学识，以及严谨的治学态度都是我们学习的榜样。在此致以真诚的感谢。感谢李老师在这三年里对我的教导与培养。

在论文的研究过程中，还得到了副导师胡莉副教授的很多指导和帮助。胡老师对工作、对学生认真负责的态度让我们油然而生敬意。在此向胡老师和表示衷心的感谢。

另外，还要感谢研究室郑伟、吴磊、廖小波等同学，他们给我提供了良好的学习、生活环境。感谢室友鲜华、周鹏、周浩，在三年的同窗生活中，他们给予了我许多无私的帮助和快乐的回忆。

同时，感谢女友张莹对我学业上的支持和生活上的照顾。

特别感谢多年以来一直关爱和支持着我的父母、姨夫一家，他们的无私奉献是我永远难以报答的。

最后，向所有关心和帮助过我的人表示谢意。

---

## 参考文献

- [1] 贺文慧. 农村现代远程教育平台的构建与应用研究[D]. 安徽农业大学硕士论文, 2004
- [2] 马戎, 龙山. 中国农村教育问题研究[M]. 福州: 福建教育出版社, 2000
- [3] 吴立德. 计算机视觉[M]. 复旦大学出版社, 1993
- [4] 马颂德, 张正友. 计算机视觉—计算理论与算法基础[M]. 科学出版社, 1998
- [5] 黎洪松. 数字视频技术及其应用[M]. 清华大学出版社, 1997
- [6] 崔之枯, 江春, 陈丽鑫. 数字视频处理[M]. 电子工业出版社, 1998
- [7] 钟玉琢, 乔秉新, 祈卫. 运动图像及其伴音通用编码国际标准——MPEG-2[M]. 清华大学出版社, 1997
- [8] 卢官明, 宗昉. 数字电视原理[M]. 机械工业出版社, 2004
- [9] 张益贞, 刘滔. Visual C++实现 MPEG/JPEG 编解码技术. 北京: 人民邮电出版社, 2002
- [10] 徐孟侠. 新的数字电视体系中的视频编解码[J]. 电视技术, 第 10 期, pp. 4-10, 1997
- [11] 赵荣椿. 数字图象处理[M]. 西北工业大学出版社, 1996
- [12] 余松耀, 张文军, 孙军. 现代图像信息压缩技术[M]. 科学出版社, 1998
- [13] 高德远, 樊晓娅, 张盛兵. 超大规模集成电路—系统和电路的设计原理[J]. 高等教育出版社, 2002
- [14] 任爱峰, 楚秀琴. 基于 FPGA 的嵌入式系统设计[M]. 西安电子科技大学出版社, 2004
- [15] 徐光辉, 程东旭, 黄如. 基于 FPGA 的嵌入式开发与应用[M]. 电子工业出版社, 2006
- [16] 黄正谦, 徐坚, 章小丽. CPLD 系统设计入门与应用[M]. 电子工业出版社, 2001
- [17] 任艳颖, 王斌. IC 设计基础[M]. 西安电子科技大学出版社, 2003
- [18] 侯伯亨, 顾新. VHDL 硬件描述语言与数字逻辑电路设计[M]. 西安: 西安电子科技大学出版社, 2001
- [19] 陈曦, 徐宁仪. SystemC 片上系统设计[M]. 科学出版社, 2003
- [20] 苏静. 嵌入式网络教育质量保障体系的设计与系统实现[D]. 天津师范大学硕士论文, 2003
- [21] 陈雷. 数字视频在远程教育中的设计与应用[D]. 东北师范大学硕士论文, 2004

- 
- [22] 叶波, 郭辉. 图象数据压缩技术及其 VLSI 实现[J]. 半导体技术, 1997
- [23] H Yamauchi, Tashiro, T Minami and Y Suzuki. Architecture and implementation of a highly parallel single-chip video DSP. IEEE Trans.Circ.and Syst.for video Technol. vol.2, no.2, pp.207-220, Jun. 1992
- [24] Koay Kah Hoe. JPEG Encoder IP Module Designed VHDL Module Generator. Project Paper Microelectronic and Computer Engineering Department(MICE) Faculty of Electrical Engineering University of Technology Malaysia, 2000
- [25] Kovac M, Ranganathan N Jaguar. A Fully Pipeline VLSI Architecture for MPEG-2 Image Compression Standard. Proceedings of the IEEE, Feb 1997, Volume:81 Issue:2, Pages:247-258
- [26] J Guo, Jia-Wei Chen, Cheng-Chung Wu. An Efficient Adder-Based 2-D DCT/IDCT Design Image Compression Applications. Proc. 2001 20th VLSI DESIGN/CAD Symposium, Aug.26-31, 2001
- [27] Zhueng-Kim, Ben J Meu. A Cost-Effective Architecture for 8x8 Two-Dimensional DCT Using Direct Method. IEEE TRAN. vol. 4, NO. 7, Jun. 2003
- [28] L D Los and M Stegherr. Parameterizable VLSI architectures for the 2D-DCT algorithm. IEEE Trans. Circ. And syst., vol. 36, no.10, pp.1309-1316, Oct.1994
- [29] P A Ruetz, P Tong, D Bailey, D A Luthi. A high-performance DCT Buffer Structure video compression chip set. IEEE Trans.Circ.and Syst.for video Technol., vol.2, no.2, pp.111-122, Jun.1997
- [30] Agostini L, V Silva, I S Bampi. Pipelined Entropy Coders for JPEG image compression. Integrated Circuits and Systems Design, 2001
- [31] H Fujiwara, M L Liu, T. Sun, K M Yang, M Maruyama, K Shomura and K Ohyama. An all-ASIC implementation of low bit-rate video code. IEEE Trans.Circ.and Syst.for video Technol., vol.2, no.2, pp.123-134, Jun.1997
- [32] Pirsch, W Gehrke and R Hoffer. A hierarchical multiprocessor architecture for video coding applications. Int.Symp.On Circuits and Systems, Chicago, IL, pp.in Proc.IEEE 1750-1753, May 3-6, 1995
- [33] 曾昭平, 马仲华. H.26L 中变换编码和量化的原理与分析[J]. 通信学报 2002, 23(2):102-108
- [34] Docef A Gee, Ismaeil I R, Khanh Nguuyen-Phi, Kossentini F. The Inverse Quantized DCT and Its Application to DCT-Based Video Coding. Image Processing, IEEE Trans., on, Mar2002, Volume:11 Issue:3, Page (s):177- 187
-

- 
- [35] 徐孟侠. 视频编码中的固定和可变数码率[J]. 电视技术, 1999, 09
  - [36] 倪泽峰, 王振华, 谭毅华, 田金文, 柳健. 并行哈夫曼编码器的硬件设计与实现[J]. 微电子学与计算机, 2002, 10
  - [37] A Fast Asynchronous Huffman Encoder for Compressed-Code. <http://www.cs.Columbia.edu/async/publications/>
  - [38] Aggarwal M, Narayan. An Efficient Huffman Encoding. Image Processing, 2000 International Conference on, 2000. vol.1, Page(s):936-939
  - [39] Rudberg M K, Wanhammar L. High speed pipelined parallel Huffman encoding. and RLE encoding Circuits and Systems, 1997. ISCAS'97, Proceedings of 1997 IEEE International Symposium on, 9-12 Jun 1997, vol. 3, Page (s):2080-2083
  - [40] Aspar Z, Mohd Yusof Z, Suleiman I. Parallel Huffman encoder with an optimized look up table option on FPGA. TENCON 2000. Proceedings, 2000, vol.1, Page(s):73-76
  - [41] Hashemian R. Design and hardware implementation of memory efficient Huffman decoding. Consumer Electronics, IEEE Trans., on, 21-23 Jun 1994, Volume:40 issue:3, Pages:345-352
  - [42] 何卫锋, 毛志刚, 吕志强, 尹海丰. 一种基于 FBMA 算法的整像素运动估计芯片的 VLSI 设计计算机研究与发展[J]. 数字电视与数字视频, 2005
  - [43] 唐泽鹏, 秦雷, 朱秀昌, 刻峰. 运动估计算法分析[J]. 数字电视与数字视频, 2001, 12
  - [44] K M Yang, M T Sun and L Wu. A block-matching algorithm family of VLSI design for the motion compensation and system. IEEE Trans. vol.36, no.10, pp.1317-1325, Oct.1998
  - [45] Chaur-heh Hsieh, Ting-Pang Lin. VLSI Architecture for Block-Matching Motion Estimation Algorithm. IEEE Trans. Circuits and Systems Vol.2, No.2, 2002
  - [46] 付宇卓, 胡铭曾, 方滨兴. 运动估计芯片中降低局存与脉动阵列数据宽度的设计方法[J]. 计算机研究与发展, 1999, 36(8):943-947
  - [47] P R Burfield. A VLSI implementation study of a 10 Mbit/s video decoder. Signal Processing: Image Communication, pp. 59-74, May 1997
  - [48] Yeu-Shen Jehng, Liang-Gee Chen and Tzi-Dar Chiueh. An Efficient and Simple VLSI Full Search Architecture for Motion Estimation. Algorithm. IEEE Trans. signal processing, Vol.41 No.2, 2004
  - [49] G hanbari. The cross-search algorithm for motion estimation. IEEE Trans. Comun.
-

**vol.38, no.7, pp.950-953, Jul.1998**

- [50] L Chen, W T Chen, T D Chiueh. An estimation algorithm for digital image processing. IEEE Trans. Technol. vol.1, no.4, pp.378-385, Dec.1996
  - [51] <http://www.altera.com>
  - [52] <http://www.opencore.org>
  - [53] <http://www.edacn.net>
  - [54] <http://www.modelsim.com>
-

## 攻读硕士学位期间发表的论文及科研成果

- [1] 钟声. 一种基于 Kalman 滤波的  $4 \times 4$  块运动估计方法 [M]. 微计算机信息, 2006 (11): 302—304
- [2] 合著. 一种提高吞吐量的 SCTP 协议改进 [M]. 电子技术应用, 2007 (1): 93—95

作者：[钟声](#)

学位授予单位：[西南科技大学](#)

## 相似文献(9条)

### 1. 学位论文 [侯志刚](#) 基于FPGA的频率域MPEG-2码率转换硬件实现 2007

近年来,随着网络技术的发展和视频编码标准受到广泛接受,视频点播、视频流和远程教育等基于网络的多媒体业务逐渐普及。为了对拥有不同终端资源,不同接入网络以及不同兴趣的用户提供灵活的多媒体数据访问服务,多媒体数据的内容需要根据应用环境动态调整,转码正是实现这一挑战性任务的关键技术之一。

视频转码对时间的要求非常苛刻,以至于用高速的通用微处理器芯片也无法在规定的时间内完成必要的运算。因此,必须为这样的运算设计一个专用的高速硬连线逻辑电路,在高速FPGA器件上实现或制成高速专用集成电路。用高密度的FPGA来构成完成转码算法所需的电路系统,实现专用集成电路的功能,因其成本低、设计周期短、功耗小、可靠性高、使用灵活等优点而成为适合本课题的最佳选择。

本文根据MPEG-2中可变长编码(VLC)理论,采用了两级查找表减少了VLC存储空间的使用,完成VLC编码的实现。根据MPEG-2中关于System Packet的定义,针对FPGA可实现性,以空间换取复杂度的减少,实现了PES包的打包模块。根据MPEG-2相应的转码理论,完成了对系统解码模块相应的连接和调试,对解码模块以真实的bit流进行了贴近板级的情况的仿真。根据MPEG-2中TM5的算法的局限性,分析得出只需要对P帧进行相应处理即可改进场景变换对视频质量的影响,完成对TM5的算法的改进。通过性能估算和电路仿真,各模块的吞吐率能够满足转码系统的要求。

### 2. 学位论文 [李天顺](#) 基于WSRF的FPGA远程实验服务的研究与设计 2010

Internet的普及,多媒体技术,虚拟仪器技术以及计算机仿真技术的出现使得远程教育得到了长足的发展。面对高校学生数量的急剧增加,实验室设备昂贵、实验资源满足不了当前需要的情况,远程实验的建立能够很好地利用有限的资源,满足实验教学的需要。本课题源于学校网络环境下计算机硬件实验新模式的研究。

本文描述并实现的基于WSRF(Web Service Resource Framework)的FPGA(Field Programmable Gate Array,现场可编程门阵列)远程实验服务,是针对目前高等学校所开设的基于FPGA的计算机硬件实验课程的远程化而研制的。本文从分析Web服务资源规范技术的优势入手,系统地概述WSRF用来实现互操作性的一系列技术规范,其中着重研究WSRF服务中的状态保持和服务封装模式。研究了解其中的状态资源属性(Resource)的定义和自动更新引用功能,结合FPGA硬件实验“设计——配置——验证”的过程和WSRF协议通过暗含的资源模式把Web服务和“状态资源”联系起来的特性,依据面向服务的体系架构思想,提出基于WSRF的FPGA远程实验四层架构。

在客户端的实现上,以提高远程实验客户的高效体验为宗旨,设计实现灵巧实用的客户端。通过JS与Flex交互配合调用的技巧,解决了B/S模式下读取文件的问题。同时很好地利用了Flex自身所具有的丰富的界面元素和丰富的数据模型,使得学生在实验过程中能够很好地交互实验,并从人性化和直观、形象上很好地呈现了实验结果,极大地增强了用户体验,很好地提高了学生实验的兴趣。

在服务端,首先介绍采用服务资源模式封装实验服务的过程和优势,然后使用WSRF.NET实现了基于WSRF的实验服务——ALU运算器实验服务。按照WSRF规范封装成Web服务,采用标准的Web Service描述语言(WSDL),对实验服务的方法接口,服务调用的入口参数,及返回值的类型进行描述。利用WSRF服务的资源属性定义和端点引用,很好地解决了实验过程中的状态问题。在服务端的资源调度上,比较现有的先来先服务、优先级先服务和小服务优先服务的调度策略,提出利用服务调度中反馈机制返回的历史服务时间未预测当前服务的执行时间,优先服务能够最快执行完的服务,增强实验设备的利用率。本文在设计好客户端和服务端之后,对客户端和服务端进行了联合调试,运行完全正常,达到预期目标。

关键词: 远程实验, FPGA, WSRF, 状态资源, 智能客户端

### 3. 会议论文 [钱志滨,肖铁军](#) 基于远程测试的虚拟FPGA/CPLD实验系统 2001

随着Internet的普及,利用Internet的远程教学已成为可能,在我国远程教育中将随着试点的成功而推广开来。远程和教学实验是远程教育中必不可少的重要环节。本文提出了一种有关FPGA/CPLD的虚拟实验系统。它不同于纯软件的仿真实验,是一种基于硬件的虚拟实验系统。通过使用该系统,用户可通过Internet操作远程实验室中的硬件,进行远程配置、测试、验证,以真正达到实验的效果和目的。

### 4. 学位论文 [钱志滨](#) 数字IC测试及其在虚拟实验系统中的应用 2002

在该文中首先阐述了一种高校电子专业数字逻辑以及现代可编程器件(FPGA/CPLD)等课程的远程实验系统,在这个系统中使用远程测试(数字IC测试)来实现实实在在的硬件实验,使得这个系统不同于纯软件的仿真。接着该文详细叙述了该实验系统中虚拟实验环境软件和实验服务提供端的数字IC测试系统的设计。虚拟实验环境软件提供一个可灵活配置、形象直观的实验界面,这个界面为使用者提供了实验的感性认识。该文叙述的数字IC测试系统可对多达96通道的可编程器件进行实验,另外它还作为面向维修的测试仪器,具有在线测试、连线测试、V-I测试、施加上拉电阻、调节门槛比较电平等功能。最后,在该文的结尾对该文所作的工作进行了总结。

### 5. 学位论文 [梁君秋](#) 机械参量测试远程虚拟实验室的构建与研究 2006

Internet的发展和普及使方兴未艾的远程教育得到了强大的技术支持。其中以Internet为传播途径的实验室实现了对传统实验室的有力补充,是对传统实验室时间上的扩展和空间上的延伸。但目前大部分虚拟实验室存在着交互性不足、功能不够强大、系统开发和维护代价高及忽视远程硬件的实时性、通用性和可扩展性等弊端,造成了虚拟实验室配置的资源浪费和不灵活性。

本文从虚拟实验室建设的原则出发,分析各种实验室实现方式的利弊,根据应力应变、振动等机械参量测试虚拟实验室对远程交互和系统建模仿真等要求,选用了基于Java、Active技术和虚拟仪器技术相结合来实现的虚拟实验室。在测试现场硬件实验平台搭建中,开发了以PC为上位机,可编程逻辑器件(FPGA)为下位机所构成的高速数据采集系统,实现了数据采集的高速控制、存储和预处理,处理结果通过串行总线传给上位机。上位机中将网络技术与虚拟仪器技术相结合,构建C/S、B/S双模式网络通信,在LabVIEW平台下完成Active组件化、远程仪器控制服务器模块及数据库接口模块的开发,实现了通过Internet为远程用户提供现场的测试数据和图像。

通过对机械参量测试远程虚拟实验室性能分析和运行观察,结果表明:FPGA硬件可重构、现场可编程,满足了虚拟实验室系统用户可定制的灵活性要求,其ms级的执行速度提高了虚拟实验室实时性能的要求;Gweb服务器、DataSocket技术及Active组件技术的应用使虚拟实验室网络体系层次结构明确、接口标准化,降低了系统更新和维护的代价;实验操作简便、人机界面友好;双模式的网络通信可以满足不同知识背景的实验者的实验要求。机械参量测试远程虚拟实验室为以后虚拟实验室向着以个人计算机为核心构成通用测试平台、功能模块及软件包形式的自动测试系统的发展奠定了基础,其

开放性、可扩展性和交互性已成为现有实验教学的有益补充，必将大大提高实验教学的伸缩性和适应性。

## 6. 学位论文 [王楠 基于数字视频技术的远程教育系统的构筑与实现](#) 2004

以数字视频技术为核心的远程教育系统,是教育信息传播的一种重要形式.随着数字视频技术的成熟和相关标准的确定,有望形成第四代远程教育系统一以数字视频技术为核心的远程教育系统.依靠数字视频高效、易于交互等特点,必将对远程教育产生积极影响.该文旨在通过对数字视频技术的应用研究与实践,明晰其对远程教育产生的深远影响.全文共分三大部分:第一部分:通过对以数字视频技术为基础的数字电视系统概述,以及关键技术的解析(视频编码技术、码流复用技术等),阐明其对远程教育系统的必然支持,通过框图形式表述以数字视频技术为核心的远程教育系统构成;第二部分:选取视频讲演模式作为教学模式,就视频交互式网络课程播放平台进行实践探究,设计并完成了一种以学习者为中心、一体化的视频交互式网络课程播放平台;第三部分:在HDTV TV板卡设计中,就作者设计环节(TTL-LVDS电平转换单元、FPGA控制单元)作以解释说明.

## 7. 学位论文 [黄国辉 非标医学视频转换及增强系统的研究与实现](#) 2008

随着计算机技术与信息技术的飞速发展,医学影像领域取得了令人目眩的进步.基于影像的现代医疗设备正以其独特的魅力步入数码时代.DSA(digital subtractionangiography)数字减影设备作为X光线学成像技术的典型应用,也由原来的影像增强器发展为数字平板成像.但是,目前开发DSA数字减影设备的各厂商在信号分辨率、行频场频、消隐脉宽等参数上都是采用各自的标准,而且只有单路模拟信号输出,这就大大降低了设备之间的兼容性,阻碍了医学视频影像的采集、存储和网络传输的发展.为了扩展DSA数字减影设备的用途,本文在对非标标准格式的 video 信号转换算法基础上进一步对图像处理算法和实现方法进行了完善.

本课题前期研究状况:在转换系统的自适应算法研究过程中,用非标模拟 video 信号的二进制数作为研究对象,在Matlab工具的辅助下,完成对非标 video 信号特征参数的分析,分别建立了同步信号检测、转换参数运算、视频编码输出等仿真模型,使得对非标 video 信号的特征参数有直观的了解.

本文应用FPGA嵌入式系统开发技术,以可编程逻辑器件FPGA为软硬件核心实现自适应转换系统中数字化医学 video 的处理,在研究 video 图像处理算法的基础上,基于FPGA用VHDL语言编程并仿真实现各 video 处理算法:如滤波去噪、图像增强、去隔行等;另外,改善了原有硬件设计能够从 video 采集前端得到更为稳定的行、场同步时序信号以得到稳定的 video 信号并增加了USB总线通信端口,将单帧 video 图像以DICOM标准格式存储在符合DICOM协议医用工作站上,实现诸如远程医疗、远程教育等功能的PACS系统。

## 8. 期刊论文 [肖铁军. 韩晓茹. 张焕春. Xiao Tiejun. Han Xiaoru. Zhang Huanchun 基于Internet的虚拟EDA实验系统](#)

-[电子测量与仪器学报](#)2005, 19(4)

针对远程教育中对实验资源共享的需求,本文介绍了基于Internet的虚拟EDA实验系统.系统设计了由客户机-管理服务器-实验服务器构成的三角形系统控制模型,提出了多客户-多实验服务器的工作模式,并通过管理服务器完成资源的分配.实验服务器设计实现了数字电路的远程测试,解决了FPGA/CPLD的功能验证问题.客户端软件设计了虚拟实验板和时序波形图二种人机界面.论文分别论述了系统的控制流和数据流,最后说明了三项系统测试的结果,证明能够通过Internet共享多种类型的FPGA/CPLD实验资源.

## 9. 学位论文 [张潇 正交幅度调制器的设计与实现](#) 2006

目前,我国的数字有线电视的推广和发展正在如火如荼的进行当中,数字有线电视在推广的过程中所持有的优点主要是高清晰的收视效果,更多的频道选择和准VOD点播频道.而数字有线电视的更大的优越性,像远程教育、家庭炒股、电视购物等即将推广的业务均是需要通过双向传输来实现的.所以,目前我国的有线电视网络正在进行双向改造,即在用户方推广含有CABLEMODEM调制解调功能的电视.但是在目前的有线电视网络在从单向网络向双向网络改造的过程中遇到的主要困难就是双向改造的费用相当高昂,阻碍了改造进行的速度.为此本设计定位在以较低的开销在用户方实现CABL EMODEM的部分功能,从而推动有线电视网络的双向改造,提高使用双向业务用户的数量.

在设计该调制器时由于我国的相关标准还未出台,所以更多的参考了欧洲的DVB-C< '[1]>标准和DAVIC< '[2]>规范中的相关技术要求,采用16QAM的调制方式.在具体的接入网指标,如频率稳定度等方面则参照了我国广电总局出台的相关规定.在设计方法上,采用了首先在MATLAB环境中进行系统级仿真,并利用C语言自行编写了仿真文件,比较了不同的实现方案,并从中筛选出了与以往实现方式所不同的,采用FFT进行处理,随后在频域加权进行滤波的方式,和利用DDS方式实现一对严格同频、正交载波的实施方案.

在具体硬件实现时,采用在单片FPGA中主要利用VHDL硬件编码的方法来实现QAM调制器的主要功能的思路,自主设计研发了基带成型滤波器、DDS子系统的模块,并在Q1Jartus II开发环境和板级仿真中验证了设计的结果。

本文链接: [http://d.g.wanfangdata.com.cn/Thesis\\_Y1196713.aspx](http://d.g.wanfangdata.com.cn/Thesis_Y1196713.aspx)

授权使用: 陕西理工学院(sxlgxy), 授权号: 6eea36eb-6007-43cc-8dc6-9df201105f76

下载时间: 2010年9月15日