

西安电子科技大学
硕士学位论文
MPEG4视频解码系统的设计与实现
姓名：程培星
申请学位级别：硕士
专业：通信与信息系统
指导教师：吴成柯
20050101

摘要

如今信息技术飞速发展,人们对多媒体,特别是包含有大量信息的数字视频的需求也越来越大。同时,超大规模集成电路技术(VLSI)、可编程逻辑器件(FPGA等)的发展使实时的视频编解码成为可能。

本文深入研究了 MPEG-4 标准及其建议的视频解码模型。在此基础上针对 MPEG-4 标准提出了一种基于硬件实现的视频解码系统结构,该结构以现场可编程门阵列(FPGA)为核心,完全用 FPGA 实现了视频解压算法,同时配合以 ARM CPU 构建整个系统。该视频编码系统实现了对于分辨率为 352×288 , 25 帧/秒符合 MPEG4 SP 级的 ES/PS 流的实时解码。用了各种标准测试序列,都得到了良好的主观图像质量。

关键词:MPEG-4,解码器,同步动态存储器,AMBA

Abstract

With the rapid development in the areas of communication and computer technologies, digital video plays a more and more important role in information society. At the same time the development of Very Large Scale Integrate circuit (VLSI) and Programmable Logic Device (such as FPGA etc.) makes real-time video compression and transportation possible.

This dissertation presents the research result of several issues in video decoding and its hardware implementation. Based on studies an MPEG-4 SP video decoding system is proposed. The system is based on FPGA and ARM. Most procedures in MPEG-4 video decoding system are fulfilled by FPGA, and ARM CPU is used to set up the demonstration system. It realizes real-time video decoding when the input picture resolution is 352×288 and its frame rate is 25 frames per second and the input bit rate can be up to 4Mbps. The performance of this system is good.

Keywords: MPEG-4, Decoder, SDRAM, AMBA

创新性声明

本人声明所呈交的论文是我个人在导师的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中所罗列的内容以外，论文中不包含其它人已经发表或撰写过的研究成果；也不包含为获得西安电子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所做的任何贡献均已在论文中做了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

本人签名：程培星

日期：2005.2.22

关于论文使用授权的说明

本人完全了解西安电子科技大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属西安电子科技大学。本人保证毕业后离校后，发表论文或使用论文工作成果时署各单位仍然为西安电子科技大学。学校有权保留送交论文的复印件，允许查阅和借阅论文；学校可以公布论文的全部或部分内容，可以允许采用影印、缩印、或其它复制手段保存论文。（保密的论文在解密后遵守此规定）

本学位论文属于保密，在__年解密后适用本授权书。

本人签名：程培星

日期：2005.2.22

导师签名：[Signature]

日期：2005.2.22

第一章 绪论

§1.1 图像压缩编码技术的发展

研究视频图像数据就可以发现其中存在大量的冗余,减少图像的冗余度就可以极大地减少图像的数据量,这些冗余主要表现在空间冗余度、时间冗余度两个方面,其他还包括信息熵冗余、视觉冗余、结构冗余、知识冗余、纹理统计冗余等等。图像压缩所解决的问题是尽量减少表示数字图像所需要的数据量。减少数据量的基本原理就是除去其中多余的数据。以数学的观点看,这一过程实际上就是将二维像素列变换为一个在统计上无关联的数据集合。这种变换在图像存储或传输之前进行。在以后的某个时候,再对压缩图像进行解压缩来重构原图像或原图像的近似图像。

图像/视频压缩编码已经有五十多年的发展历史了。1948年,Shannon和他的两个学生 Oliver 与 Pierce 联合发表了对电视信号进行脉冲编码调制(PCM)的论文,标志着数字视频压缩编码技术的开端。1966年,Neal 对比分析了 DPCM 和 PCM 并提出了用于电视的实验数据。1969年在美国举行的首届“图像编码会议(Picture Coding Symposium)”表明图像/视频编码以独立的学科跻身于学术界。半个世纪以来,视频压缩编码技术早已走出实验室,广泛应用于现代社会的各个领域。

视频编码方法种类繁多,从信息保持的角度可以分为无损压缩和有损压缩两大类。无损压缩利用数据的统计冗余(数据之间的相关性、可预测性)进行压缩,以预测编码和熵编码为基础,可完全恢复原始数据而不引入任何失真,但其压缩率受到数据统计冗余度的理论限制,一般为 2:1 到 5:1。此类方法广泛用于文本数据、程序和特殊应用场合的图像数据(如指纹图像、医学图像等)压缩。鉴于压缩比的限制,仅使用无损压缩方法不可能解决图像和数字视频在存储和传输应用中的问题。因此,视频有损压缩方法也得到了长足的进步和发展,它除了利用统计冗余进行压缩编码外,还利用了视频数据的视觉冗余特性,即利用人类视觉系统(HVS)对视频信息中某些频率成分不敏感的特性,允许压缩过程中损失一定的信息,这虽然不能在解码端完全恢复原始数据,但所损失的部分信息对理解原始图像的影响相对较小(即视觉无明显失真),同时却换来了相当大的压缩率。有损压缩以变换编码、矢量量化、模型基编码、分形编码和熵编码为基础。

1985年,瑞士学者 Kunt 提出了第一代、第二代编码的概念,他把早期的以去除冗余为基础的编码方法称为第一代编码,如:空间域的 PCM、DPCM、 ΔM 、亚抽样编码方法,变换域的 DFT、DCT、Walsh-Hadamard 变换编码等方法以及

以此为基础的混和编码方法；第二代编码方法则是指 80 年代以后提出的新方法，如金字塔编码、矢量量化、分形编码、基于神经网络的编码、小波变换和子带编码和模型基编码等。

进入 90 年代以后，ITU-T 和 ISO 制定了一系列图像编码国际标准。如：

1990 年为会议电视和可视电话制定的 H.261 标准。

1991 年为静止图像编码制定的 JPEG 标准。

1991 年为电视数字图像存储而制定的 MPEG-1 标准。

1993 年为活动图像及其伴音压缩而制定的通用编码国际标准 MPEG-2。

1994 美国“大联盟”公布数字 HDTV 系统的说明书草案。美国“先进电视系统委员会”拟定“数字电视标准”。

1996 年 ITU-T 为甚低码率视频编码而制定的 H.263 标准。

1998 年 ITU-T 拟定 H.263 Version2 草案，即 H.263+。

1998 年拟定 MPEG-4 草案，首次在编码中引入了视频对象(Visual Object)和基于内容编码的概念。

2003 年 5 月，ITU-T 和 ISO 联合发布了 H.264 (MPEG4 的 Part10)。H.26X 和 MPEG 两大系列在 H.264 上完成了统一。

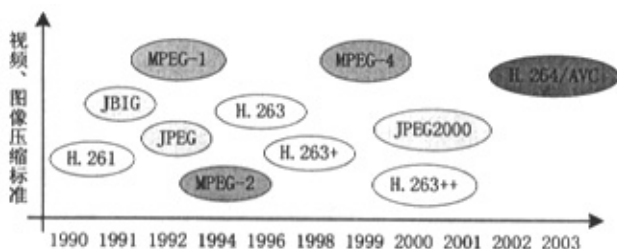


图 1.1 视频/图像压缩编码国际标准的发展历史

这些标准的制定极大地推动了图像编码技术的实用化和产业化。会议电视等各类使用图像编码技术的产品纷纷推出，数字激光唱盘(VCD、DVD)等产品以百万台的数量级走向市场，进入家庭，从而迎来了数字图像通信的黄金时代。

另一方面，图像编码技术产业化进程的加快也推动了图像编码技术以更快的速度发展。目前的研究工作主要分为两个方向：

更好地实现现有的图像编码国际标准，研制出集成度更高、性能更好的图像编码专用芯片(ASIC)，使编码系统成本更低，可靠性更高。几个国际标准的单片或两片的系统级解码 ASIC 均已推出，其中包括对运算速度要求很高的 MPEG-2 解码芯片。不久还将推出要求更高的 HDTV 的单片视频解码芯片。符合 MPEG-2 标准的单片或多片视频编码 ASIC 也已推出。

解决好现有的图像编码系统开发中的技术问题。例如：提高图像质量，提高抗误码能力，以及在 ATM 网等变速信道上的应用等。近几年来对 MPEG 系列编码器具体实现算法有了很多突出的研究成果。国际标准的开放性结构为这种深入的改进提供了条件，它允许人们在不影响兼容性的前提下发挥自己的创造性，对标准中的开放部分进行改进。这些开放性部分包括运动估值和运动补偿方法，自适应量化系数和缓存器控制策略等。在国际标准规定的约束下，对这部分算法作更合理的细化有助于提高编解码器的性能。如：复原图像质量，提高抗误码能力等。

§ 1.2 图像压缩编码技术的应用

图像编码系列国际标准的提出标志着图像编码技术已经成熟，开始由学术研究转为产业化，前景十分诱人。早在 1991 年就有人预言，图像编码技术的突破具有十分巨大的意义。其意义之大已到可以促使现有信息产业的结构发生巨变的程度，它使通信，广播，计算机产业的界限变得模糊。几年来的实践也证实了这个预言。

图像编码技术对数字电视的发展起着至关重要的作用。数字电视具有图像质量高、频谱利用率高、可以实现多种业务的动态组合和统计复用、易于加密、具有可扩展性、可分组性和互操作性、可以灵活组成交互式电视系统等优点。这些优点是模拟电视所无法比拟的。但是数字电视受到数据量庞大的约束，所以近几年来图像编码研究主要集中在视频压缩上，以改进数字电视的性能，从而促使各等级数字电视的成功实现，扩大其应用领域。这从今年来制定的图像编码国际标准中视频压缩编码和静止图像编码国际标准数量比例中即可窥见一斑。以分辨率为基础，数字电视可分为四个等级：可视电话、会议电视、数字标准清晰度电视和高清晰度电视(HDTV)。

图像编码技术在其它方面同样有很多应用，如：

电视计算机(telecomputer)。它是介于电视与计算机之间的应用。它将个人计算机和电视融为一体，构成一个多媒体工作站。

多媒体出版物，包括电子图书，电子报刊等。

各种图像信息系统，如指纹库，遥感图像数据库等。

图像压缩技术已经为开创新的应用领域提供了良好的技术基础。如 MPEG-4 专家组为 MPEG-4 制定的功能目标指出，MPEG-4 的目标不仅是改进的可视电话，而且应该满足广播，通信，计算机领域相互渗透的要求，以交互性，高压缩比以及多种存储与通信信道相连为特色，从而构成现有的国际标准全新的视听应用标准。总之，图像压缩技术为开拓全新的应用领域打下了坚实的基础。

§ 1.3 现有的视频编码标准的简要介绍

自 80 年代以来, 由于数字存储媒体、电视传播及通信等应用中对运动图像编码方法需求的日益增长, ITU, ISO 等国际组织都成立了专门的机构, 致力于制订运动图像压缩编码的国际标准。到目前为止, 已经开发和正在开发的运动图像压缩标准有: H.261、H.263、MPEG-1、MPEG-2、MPEG-4、MPEG-7、H.264、和 MPEG-21。下面对这些标准分别做一简单介绍。

CCITT(即后来的 ITU)第 15 研究组于 1984 年成立了“可视电话专家组”, 经过 1985~1988 三年的研究, 提出了视频编解码器的 H.261 标准草案, 以覆盖 ISDN 基群信道, 满足会议电视和可视电话业务日益发展的需要。1990 年 7 月通过了该标准, 定名为“ $p \times 64\text{kbps}$ 视听业务的视频编码器”, 其中 $p=1\sim 30$ 。为了满足近年来在普通公用电话网或移动电话网上进行可视电话通信的需要, 即视频压缩率低于 64kbps, 在诸如 28.8kbps 等速率的信道上进行可视电话通信, ITU-T 在 H.261 标准基础上进行了改进, 于 1995 年提出了 H.263 标准“甚低码率通信的视频编码”。

MPEG-1 制定于 1992 年, 可适用于不同带宽的设备, 如 CD-ROM、Video CD。他的目的是把 221Mbps 的 NTSC 图像压缩到 1.2Mbps, 压缩率为 200:1。传输速率为 1.5Mbps, 编码速率最高可达 4~5Mbps, 但随着速率的提高, 其解码后的图象质量有所降低。MPEG-1 主要是针对数字存储媒体, 但它也被用于数字电话网络上的视频传输, 如非对称数字用户线路 (ADSL), 视频点播和教育网路等。

MPEG-2 制定于 1994 年, 设计目标是高级工业标准的图象质量以及更高的传输率。它所提供的传输速率在 3M-10Mbps。MPEG-2 技术就是实现 DVD 的标准技术, 现在 DVD 播放器已经在家庭中普及起来了, 除了用于 DVD 外, 还可以为广播、有线电视网、电缆网络以及卫星直播提供广播级的数字视频。

MPEG-4 旨在将众多的多媒体应用集于一个完整的框架内, 为不同性质的视频、音频数据制定通用的编码方案, 提出基于内容 (Content-Based) 的视频对象 (Video Object) 的编码标准。它不仅针对一定比特率下 (4800~6400bps) 的视频、音频编码, 更加注重于多媒体系统的交互性和灵活性。为了达到这个目标, MPEG-4 引入了对象基表达 (object-based representation) 的概念, 用来表达视听对象 (audio/visual objects, AVO); MPEG-4 扩充了编码的数据类型, 由自然数据对象扩展到计算机生成的合成数据对象, 采用合成对象/自然对象混合编码 (Synthetic/Natural Hybrid Coding, SNHC) 算法; 基于内容的压缩编码是 MPEG 研究的热点。MPEG-4 的编码系统是开放的, 为各种多媒体应用提供一个灵活的框架和一套开放的编码工具, 不同的应用可选取不同的算法。

MPEG-7 的工作于 1996 年启动, 名称叫做多媒体内容描述接口 (Multimedia Content Description Interface), 目的是制定一套描述符标准, 用来描述各种类型的

多媒体信息及它们之间的关系,以便更快更有效地检索信息。这些媒体材料可包括静态图像、图形、3D 模型、声音、话音、电视以及在多媒体演示中它们之间的组合关系。MPEG-7 的应用领域包括:数字图书馆(Digital library),例如图像目录、音乐词典等;广播媒体的选择,例如无线电频道,TV 频道等;多媒体编辑,例如个人电子新闻服务,多媒体创作等等。

H.264 (MPEG4 的 Part10) 发布于 2003 年,它是由 ITU-T 和 ISO 共同制定的新一代图像压缩标准,能提供比 MPEG4 和 H.263 更高的压缩性能,使图像的数据量减少 50%,有利于有限的空间存储更多的图像数据;对网络传输具有更好的支持,引入面向数据包编码,有利于将数据打包在网络中传输,支持流媒体服务应用;具有较强的抗误码特性,以适应在噪声干扰大、丢包率高的无线信道中传输;对不同应用的时延要求具有灵活的适应性;编码和解码复杂度具有可扩展性,支持编码和解码复杂度的不等分配和扩展。H.264 中引入了一些新的压缩方法,以提高压缩效率,它具有如下特点:增强的运动补偿性能,采用更小块进行变换编码,采用块间滤波器提高性能,高性能的熵编码,采用 SP 和 SI 帧支持视频流间切换等。

§ 1.4 FPGA 相比通用处理器的优点

MPEG-4 是一个十分复杂的标准。MPEG-4 技术的实现,离不开满足系统级参数(如每秒需要处理的宏块数)所需要的并行机制以及一种软件协调下的视频流水线结构。FPGA 可提供必需的并行机制,从而可为视频编码解码器提供经济的实时解决方案。对比目前的硬件器件 MOPS 性能,我们发现通用处理器的性能约为 2000 MOPS。DSP 处理器将这一数字提高到 8000 MOPS 左右,但仍然受限于只能通过有限数量的计算单元完成数据流的处理。采用专用处理引擎(如位操作单元)的媒体处理器可将这一数字提高到 20000 MOPS,而 FPGA 和 ASIC 则有更大的自由度,性能可轻松超过 10 万 MOPS。

因为 FPGA 器件基于 SRAM 技术,FPGA 还具有一个明显的优势,就是可重新编程,能够比较方便的实现系统的集成,维护和升级。不同的 MPEG-4 技术可根据需要重新编程到 FPGA 器件中。这也提供了支持未来 MPEG-4 类和级的潜力,采用 FPGA 开发出有限的几种设备,并利用 FPGA 的可重编程能力来满足不同的市场需求,这种做法有相当的优点。

§ 1.5 论文内容及作者所完成的工作

本文简要介绍了视频压缩编解码的原理和基本的解压缩方法,并在此基础上重点研究了 MPEG-4 视频压缩编码的标准,针对 MPEG-4 提出了一种解码器的 SoC 实现,采用现场可编程门阵列(FPGA)和 ARM 构建演示系统,并对解码器中的数据流调度和总线实现进行了深入研究。本文作者参与了系统方案的设计,在

MPEG-4解码系统中主要负责数据流调度模块的硬件实现。

论文章节安排如下：

第一章绪论对视频编码技术发展的历史和应用作了简要的回顾，对图像压缩编码方法在实际中的应用和现行的图像压缩编码标准作了简要的介绍，并介绍了作者的主要工作。

第二章简要的介绍了MPEG-4视频编码标准的基本原理和框架，并在此基础上对比其他视频标准讨论了MPEG-4的新功能和特点。

第三章详细介绍了基于FPGA的MPEG-4解码系统的一种硬件方案。在给出系统设计的基础上，讨论了各个模块的原理和硬件实现。

第四章提出了一种在MPEG-4 SP级解码器中的数据存储方案。在简单介绍了SDRAM的特点以后，提出了解码器IP核与SDRAM接口模块的设计方案，并巧妙地利用了一种新颖的填充方法，使得程序执行的效率大幅度提高。

第五章介绍了AMBA AHB总线传输的过程，提出了一种硬件实现的方法。

第二章 MPEG-4 视频编码标准

§ 2.1 MPEG-4标准简介

MPEG-4与以往传统的视频压缩标准的一个重要区别在于, MPEG-4定义了一套视频压缩的实现框架, 它从原理一级规定了采用该标准进行压缩时所应遵循的方式, 它为符合该国际标准的压缩视频定义了统一的理论基础, 但它并没有详细规定标准的具体实施细则, 并没有从工具角度来定义标准的实现。这种区别就使得对于MPEG-4的研究有了更多实践方面的意义。研究人员可以从MPEG-4的理论基础入手, 采用不同的压缩技术以不同的考虑角度对一幅视频图像进行符合MPEG-4标准的实现, 在实现细节方面可以采取各种各样的方法。通过这种实现方式可以使得研究人员在深入理解MPEG-4标准的同时对于它在具体实现方面有更多可能去发挥各自的解决问题的能力, 从而使得MPEG-4标准可能达到制定时的两个目标: 一是低比特率下的多媒体通信;二是多种产业多媒体通信的融合。

2.1.1 MPEG-4 标准的构成

MPEG-4标准主要由6个部分构成

(1)多媒体传送整体框架(DMIF, Delivery Multimedia Integration Framework)

它主要解决交互网络中、广播环境下以及磁盘应用中多媒体应用的操作问题。通过传输多路合成比特信息来建立客户端和服务端端的握手和传输。MPEG-4的应用可以从DMIF请求而建立具有特定服务质量(Qos)的频道, 以及用于每个基本数据流的带宽。

(2)数据平面

MPEG-4中的数据平面可以分为两部分: 传输关系部分和媒体关系部分。为了使基本流和视音频对象(AVO)在同一场景中出现, MPEG-4引用了对象描述符(OD)和流图桌面(SMT)的概念。对象描述符传递与各种特定AVO伴随的基本数据流的数目和特性的信息。数据流映射表把每个数据流链接到一个信道联系标记上, 用于管理携带此数据流的信道, 以实现该数据流的顺利传输,

(3)缓冲区管理和实时识别

MPEG-4定义了一个系统解码模式(SDM)。该解码模式描述了一种理想的处理比特流语法和语义的解码装置。它没有对设施的各种细节作不必要的各种假定, 目的是使实施者可以自由地以各种方式设计各种MPEG-4终端和解码设备。通过有效地管理, 可以更好地利用有限的缓冲区空间。

(4)音频编码

MPEG-4 不仅对自然声音的编码标准化, 而且支持语音合成和音乐合成。其音

频对象可以分为两类：自然音频对象和合成音频对象。自然音频对象包括了从 2Kbit/s 到超过 64Kbit/s 的各种传输质量的编码，在技术上借鉴了已有的音频编码标准，如 G.723, G.728, MPEG-1 及 MPEG-2 等。为了在整个码率范围内得到高质量的音频，MPEG-4 定义了 3 类编码器：最低码率采用参数编码技术；中等速率码率采用各种码激励线性预测(CELP)技术；较高码率采用自适应音频编码(AAC)的时频编码技术。对于更高的码率范围，工具中直接引入 MPEG-2 AAC 标准，以提供通用音频压缩方法。

自然音频对象的编码支持各种分级编码功能和错误恢复功能。合成的音频对象主要包括结构音频和文语转换。结构音频是一种类似 MIDI 的音乐语言，采用描述语音的方法来代替压缩语音，允许一个结构化的音频解码器根据输入数据产生各种输出声音，由一种结构化的音频交响乐语言(SAOL)来驱动。文语转换是把文字信息合成为语音信息，允许以文本或含有各种韵律学参数(如音高线和因素持续时间等)的文本作为输入，合成输出相应的易懂语音，这部分在应用时通常与脸部动画、唇动合成等技术结合起来使用。

(5) 视频编码

MPEG-4 中提出了基于 VOP 的视频编码。这一思想着重将图像表示成消息序列，图像中被编码的实体不再是基于像素块的划分，而是基于内容划分，称之为视频对象。这些对象既可以是自然的，如视频序列中的人物或具体的事物；也可以是合成的，包括 2D、3D 动画和人面部表情动画等。由于 MPEG-4 标准结构的可扩充性，根据实际应用的需要可以很容易添加新的对象类型。以下是几种 MPEG-4 中支持的视频对象：

自然的视频对象：它是从自然图像和视频中分割出来的某一视频区域，对象不一定是规则的矩形，并且也可能不在一个连通的区域内，它可能有比较复杂的形状，矩形图像只是视频对象的一个特例。为了描述视频对象的形状，MPEG-4 专门定义与图像大小相同的二值 α 平面，二值 α 平面分别用 0 和 255 表示在一个图像区域内所有像素点的状态。 α 平面的编码采用基于象素块的运动补偿和基于上下文的二进制算术编码相结合的方法，视频对象的亮度信息和色度信息采用运动补偿和形状自适应的 SA-DCT 编码。

Sprite 对象：Sprite 对象是针对背景对象编码而提出的。某些情况下背景对象自身没有任何运动，而图像序列中的背景变化是由于摄像机的运动和前景物体的运动所造成的，因此可以将整个序列的背景图像拼接成一个完整的背景图像，此后出现的每一帧图象的背景均可由此背景图像重构，此完整的背景图像就称为 Sprite 图像。Sprite 图像只需要编码传输一次并存储在解码端，在随后的图像只需要传输摄像机相对于背景的运动参数，就可以从 Sprite 上恢复所有的图像的背景。

2D 网格对象：2D 网格将一个 2D 图像平面分成许多多边形的小块，块与块之

间既不重叠也没有缝隙,多边形块的顶点叫网格的节点。MPEG-4 目前只考虑由三角形组成的动态网格模型,所谓的动态网格模型不仅指网格的形状和拓扑结构,还包括所有网格的节点在一段时间内的运动信息。

3D人脸和身体对象:3D人脸对象是用3D线框模型来描述人脸的形状、表情和口形变化等各种面部特征,MPEG-4定义了两套参数来描述人脸的形状和运动,面部定义参数 FDP和面部动画参数FAP。FAP参数有两种编码方法:基于帧的编码和基于DCT的编码。3D身体对象和人脸对象一样也是由身体定义参数BDP和身体动画参数 B A P来描述人体的形状、姿势、纹理和运动。

静态的纹理对象:在前面讨论的2D网格对象、3D人脸、身体对象以及3D网格对象的编码中,为了在解码端得到具有真实感的合成对象,常常需要用基于图像的着色技术,把对象的实际的纹理映射到模型上。这些纹理信息只需要在第一帧或者一些关键帧编码传输,因此把这些要编码传输的纹理信息叫做静态的纹理对象。静态纹理的编码采用零树小波变换和算术编码相结合的方法,静态纹理对象也可以是任意形状。

3D网格对象:3D网格模型在计算机图形、CAD、有限元分析和可视化技术等方面有着广泛应用,MPEG-4 标准中提供一个有效的编码3D网格模型的工具,编码 3D 网格模型的几何形状、拓扑结构和纹理等,包括规则的三维网格的编码和通用的 3D 网格的编码。

(6)场景描述

MPEG-4提供了一系列工具,用于组成场景中的一组对象。一些必要的合成信息就组成了场景描述,这些场景描述以二进制格式BIFS(Binary Format for Scene description)表示,BIFS与AV对象一同传输、编码。场景描述主要用于描述各AV对象在一具体AV场景坐标下,如何组织与同步等问题。同时还有AV对象与AV场景的知识产权保护等问题。MPEG-4为我们提供了丰富的AV场景。

2.1.2 MPEG-4 的数据结构

上文已经提到MPEG-4视频流中包括以下几种数据:视频数据、静态纹理数据、2-D网格数据和人脸活动参数数据等。这些结构化数据称为对象,对象和它们的属性按分层形式组织,以便支持码流可扩展性和对象可扩展性。编码的视频数据是由一种被称为层的视频码流组成的有序集。如果只有一层,那么称这个视频流是不可扩展的;反之,如果有两层以上,那么就称这个视频流是可扩展的。这些层中有一层称为基本层,它总是可以单独解码的。其他层称为增强层,它们只能同基本层和较低的增强层一起解码。

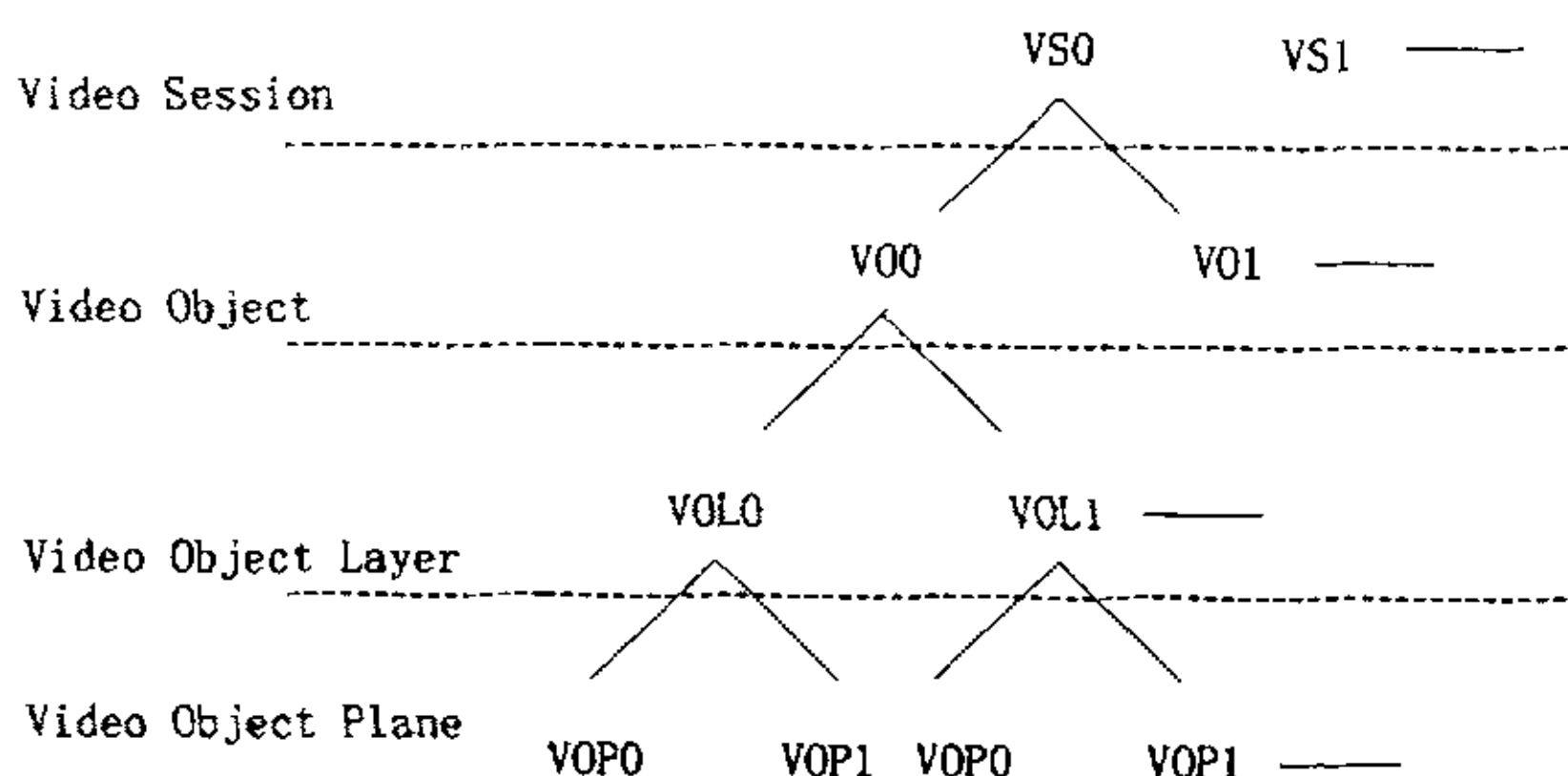


图2.1 MPEG-4的数据层次

图2.1给出了MPEG-4的层次化的数据结构，它们都以类的形式定义。它主要包含以下几个部分：

视频序列VS(Video Sequence): 一个完整的MPEG-4景物包括2-D, 3-D自然景物或合成景物和它们的增强层。一个VS可以由一个或多个视频对象组成。

视频对象VO(Video Object): VO是场景中具体的人或物体，可以是矩形，也可以是任意形状。视频对象是MPEG-4编码的独立单元。MPEG-4的场景可以由一个或多个视频对象组成，每个视频对象的形状、运动和纹理用其相关的时间和空间信息给出。因而，它有生命周期，由时间上连续的许许多多的帧构成。

视频对象层VOL(Video Object Layer): VO的三种属性信息包括纹理、形状和运动编码于这个类中，这个类的引入主要用来扩展VO的时域或空域分辨率。MPEG-4通过VOL实现分级编码，一个VO可以由一个或多个视频对象层组成，每层代表VO的一级分辨率。

视频对象平面VOP(Video Object Plane): 是VO某一时刻的取样，即某一帧中的VO。与MPEG-2类似，VOP可分为I-VOP、P-VOP和B-VOP三种。对I-VOP独立地进行编码，对P-VOP和B-VOP采用运动补偿编码。VOP可以是任意形状。

在图2.1中，视频对象序列VS0, VS1, ... 是整个视频场景序列在各个时间段上的图像序列；V00, V01, ... 是从VS中提取的不同的视频对象；VOL0, VOL1, ... 是对应于VO的不同分辨率层，其中包括一个基本层和多个增强层；VOP0, VOP1, ... 是VO在不同分辨率层的时间采样。

概括来说，MPEG-4视频流由多个VS组成，每个VS又是由一个或多个VO构成，而每一个VO可能有一个或多个VOL层次，如基本层，增强层，每一层次就是VO的某个分辨率的表示。每个VOL在时间上体现为连续的一系列VOP。MPEG-4整个编码过程最关键的就是VO的形成和表示。

§ 2.2 MPEG-4视频编码和解码框架

MPEG-4 编码和解码是针对 VOP 进行的, 其逻辑结构如图 2.2 所示。编码时首先由输入的视频序列定义出 VOP, 针对每一个 VOP 分别进行编码, 将所有 VOP 编码的结果合成在一起, 形成压缩视频数据流。解码时首先将压缩视频数据流分解, 得到每一个 VOP 的编码数据流, 针对它们分别进行 VOP 解码, 解码结果组合在一起形成输出视频。

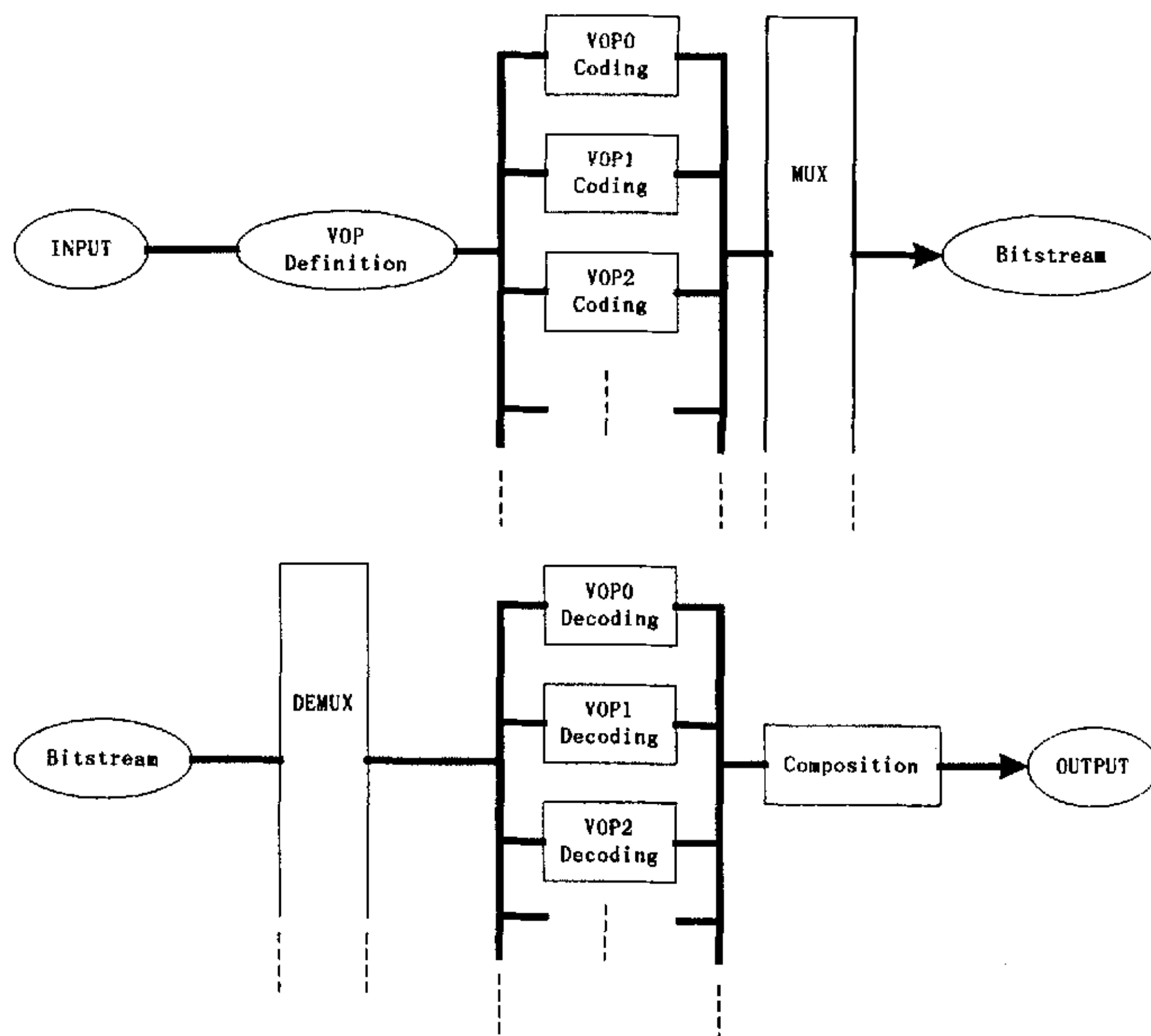


图 2.2 MPEG-4 编码和解码原理图

§ 2.3 MPEG-4与以前的视频标准比较的优点

与前面的 MPEG 标准相比较, MPEG-4 标准支持一些新的功能, 这些功能可以粗略的被划分为三类: 基于内容的交互性、高压缩率和灵活多样的存取模式。现分别介绍如下:

(1) 基于内容的交互性

基于内容的操作与比特流编辑: 支持无须编码就可进行基于内容的操作与比特流编辑。例如: 使用者可在图像或比特流中选择一具体的对象(例如图像中的某个人、某个建筑等), 然后后改变它的某些特性。

自然与合成数据混合编码: 提供将自然视频图像同合成数据(如文本、图形等)

有效结合的方式，同时支持交互性操作。

增强的时间域随机存取：MPEG-4 将提供有效的随机存取方式。在有限的时间间隔内，可按帧或任意形状的对象，对音、视频序列进行随机存取。例如以一序列中的某个音、视频对象为目标进行“快进”搜索。

(2) 高压缩率

提高编码效率：与现有的或正在制定的标准相比，在可比拟速率上，MPEG-4 标准将提供更好的主观视觉质量的图像。这一功能可望在迅速发展的移动通信网中获得应用，但值得注意的是提高编码效率不是 MPEG-4 的唯一的的主要目标。

对多个并发数据流的编码：MPEG-4 提供对一景物的有效多视角编码，加上多伴音声道编码及有效的视听同步。在立体视频应用方面，MPEG-4 将利用对同一景物的多视点观察所造成的信息冗余，这一功能在足够的观察视点条件下，将有效地描述三维自然景物。

(3) 灵活多样的存取

错误易发环境中的抗错性：“灵活多样”是指允许采用各种有线、无线网和各种存储媒体，在这些媒体中 MPEG-4 具有更高的抗错误能力，尤其是在易发生严重错误的环境下的低比特应用中。应当注意到，MPEG-4 是第一个在其音、视频表示规范中考虑信道特性的标准，其目的不是取代已有通信网提供的错误控制技术，而是提供一种对抗残留错误的坚韧性。例如：选择前向纠错、错误遏制，或错误掩盖。

基于内容的尺度可变性：内容尺度可变性意味着给图像中的各个对象分配优先级。其中，比较重要的对象用较高的空间或时间分辨率表示。基于内容的尺度可变性是 MPEG-4 的核心，因为一旦图像中所含对象的目录及相应的优先级确定后，其他的基于内容的功能就比较容易实现了。对于极低比特率应用来说，尺度可变性是一个关键的因素，因为它提供了自适应可用资源的能力。例如，这个功能允许使用者规定：对具有最高优先级的对象以可接受的质量显示，第二优先级的对象则以较低的质量显示，而其余内容(对象)则不予显示，可见，这种方式可最有效地利用有限的资源。

§ 2.4 MPEG-4的框架和级别

MPEG-4 提供了大量的、丰富的音频视频对象的编码工具，能够满足各种各样的应用需要。对于某一特定的应用，只有一部分系统、视频和音频的编码工具被采用，框架(profile)就是针对特定的应用确定要采用的编码工具，它是 MPEG-4 提供的工具集的一个子集。每一个框架又有一个或多个级别(level)来限制计算的复杂度。MPEG-4 共有 4 类框架：视频框架、音频框架、图形框架和场景描述框架。在选用不同的框架时各部分是相互独立的。其中视频框架，共有九种

Profile, 每种 Profile 下面对应几种 Level。一个特定的 profile 支持一个到几个视频对象类型,例如本论文主要讨论的 Simple Profile 支持 simple object,而 advanced simple profile 支持 simple object 和 advanced simple object。表 2.1 就是 simple object 和 advanced simple object 支持的工具集

Visual Tools	Visual Object Types	
	Advanced Simple	simple
I-VOP	X	X
P-VOP	X	X
B-VOP	X	
DC Prediction	X	X
AC Prediction	X	X
4-MV, Unrestricted MV	X	X
Slice Resynchronization	X	X
Data Partitioning	X	X
Reversible VLC	X	X
Short Header	X	X
Method 1/Method 2 Quantization	X	
Interlace	X	
Global Motion Compensation	X	
Quarter-pel Motion Compensation	X	

表 2.1 视频对象类型所支持的工具集合

每个 profile 还分不同的级别,表示不同的复杂度,表 2.2 所示:

Visual Profile	Simple	Simple	Simple	Simple
Level	L3	L2	L1	L0
Typical Visual Session Size	CIF	CIF	QCIF	QCIF
Max objects	4	4	4	1
Maximum number per type	4 x Simple	4 x Simple	4 x Simple	1 x Simple
Max unique Quant Tables	1	1	1	1
Max. VMV buffer size (MB units)	792	792	198	198
Max VCV buffer size (MB)	396	396	99	99
VCV decoder rate (MB/s)	11880	5940	1485	1485
VCV Bound-ary MB decoder rate (MB/s)	N. A.	N. A.	N. A.	N. A.
Max total VBV buffer size (units of 16384 bits)	40	40	10	10
Max vbv buffer size (units of 16384 bits)	40	40	10	10
Max. video packet length (bits)	8192	4096	2048	2048
Max sprite size (MB units)	N. A.	N. A.	N. A.	N. A.
Wavelet restrictions	N. A.	N. A.	N. A.	N. A.
Max bitrate (kbit/s)	384	128	64	64
Max. enhance-ment layers	N. A.	N. A.	N. A.	N. A.

表 2.2 simple profile 中的不同级别

对 simple profile 中的 L0 级别, 有下列限制:

- 最大帧率只能为每秒 15 帧
- 最大的 f_code 值为 1
- intra_dc_vlc_threshold 只能为 0
- 亮度的水平宽度最大为 176 像素
- 亮度的垂直高度最大为 144 个像素
- 如果使用了 AC 预测, 则有下列限制: QP 值在一个 VOP 中不能发生变化 (或者在一个视频包中不能发生变化, 如果视频包使用了的话). 如果 AC 预测没有使用, 对 QP 值的变化没有限制。

第三章 MPEG-4 SP 级解码器的硬件实现

§ 3.1 解码器的设计目标

现在多媒体的应用越来越广泛,而且随着 VLSI 技术的成熟,FPAG 的价格也越来越低。根据实际的市场需要,我们设计的 MPEG-4 解码器系统应该达到以下一些指标:

- ◆图像分辨率: CIF,后处理支持到 VGA
- ◆帧率:25f/s
- ◆码速率: 最大 4Mbps
- ◆总线接口: 支持 AMBA AHB
- ◆外部 CPU: ARM7TDMI

§ 3.2 系统方案介绍

根据视频压缩的特点,我们采用软硬件来联合实现基于 MPEG-4 视频压缩解码器,系统解码及显示部分主要在一块 FPGA 中完成,同时提供一片 ARM 的 CPU 为系统以后可扩展性打下基础。FPGA 我们采用 XILINX 公司最新 Virtex-II 系列的 XC2V1500-4BG575, 其为 150 万门的资源规模, 具体如表 3.1 所示:

参数	XC2V1500
系统门	1,500,000
逻辑单元	7,680
最大内部 RAM	240Kbits
18Kbits 块 RAM	48
总的块 RAM	864Kbits
18 * 18 硬件乘法器	48
时钟 DCM 单元	8
配置器件规模	5,659,296
核电压	1.5V
I/O 支持电压	2.5V/1.3V/1.8V/1.5V/1.2V
最大可用 IO 数	528

表3.1 FPGA的具体参数

我们采用的是基于ARM7TDMI核的Samsung公司的S3C440B0X CPU, 其内部包括8K字节Cache、一个多主机I2C接口、2通道UART、8 个外部中断源、71 个通

用IO口、在芯片PLL及32位的硬件乘法器,核电压2.5V, IO 电压3.0V~3.6V, 最大系统时钟66MHz, 芯片为LQFP160 封装。ARM7 处理器的仿真器选用与ARM 公司的MultiICE全兼容的FFT-ICE, 编译软件选用ARM 公司的ADS1.2。

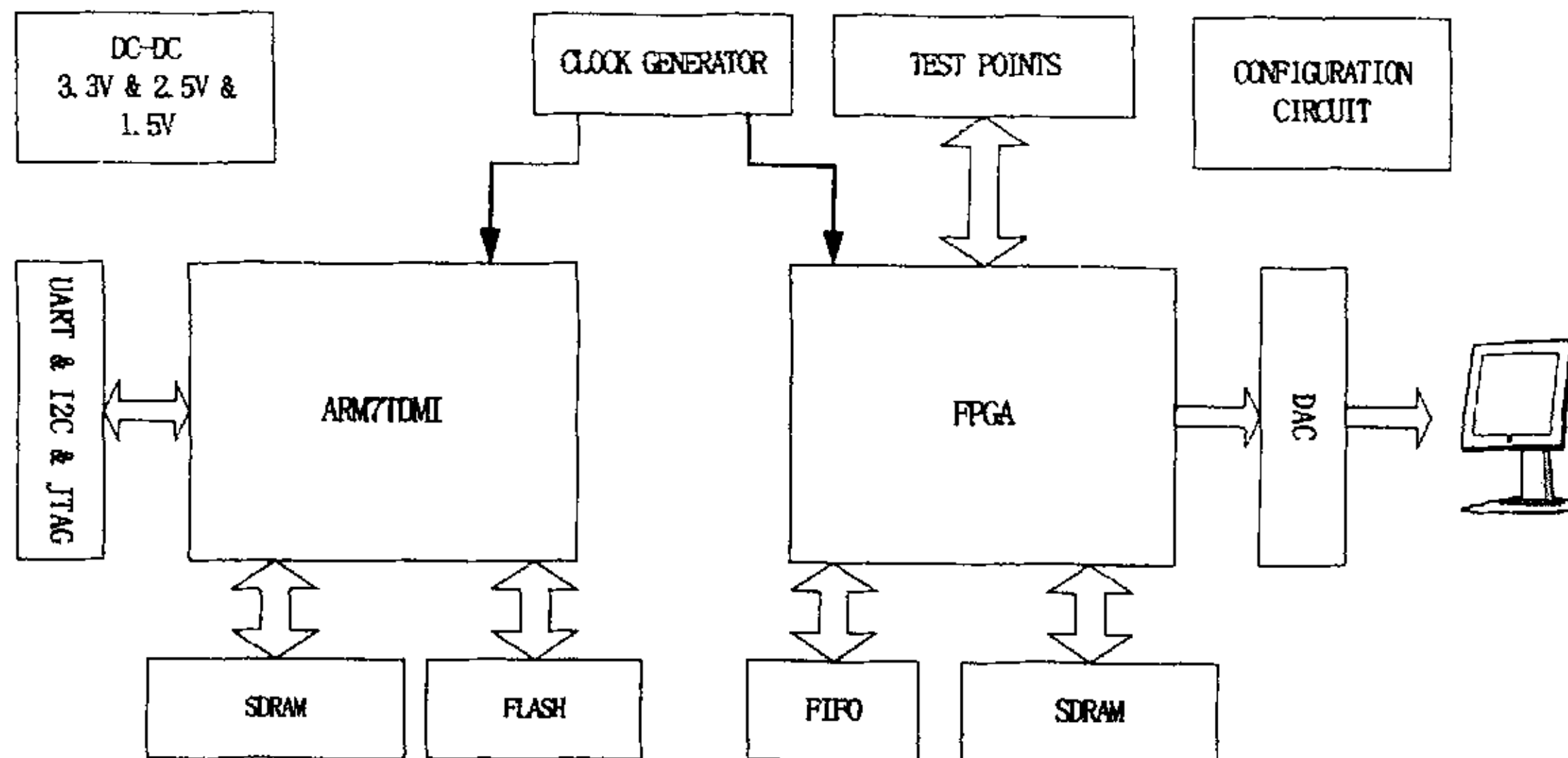


图3.1 系统总体的结构图

系统总体结构图3.1中FPGA外扩一片16Mbits的SDRAM(只用了数据线的低8位), 完成解码部分参考帧的存储及必要的数据中转; 同时外扩一片3Mbits的FIFO, 完成接收码流的数据缓冲; DAC完成数字R、G、B信号的数模转换。CPU外扩一片4Mbits的FLASH, 存放ARM7TDMI的程序代码, 同时存储演示系统的ES/PS流数据; 外扩2片4M x 16bit 的SDRAM组成4M x 32bit的memory, 若采用16位数据总线, 则只使用其中一片。CLOCK GENERATOR 部分提供FPGA 的工作时钟、显示部分的像素时钟、ARM7TDMI的系统时钟及RTC时钟; UART&I2C&JTAG 部分为ARM7TDMI的通用外部通信接口及JTAG接口; CONFIGURATION CIRCUIT 部分选用Xilinx 公司的在系统可编程(ISP)系列PROM XC18V04, 为3.3V 供电, 4Mbits 容量。DC-DC 模块输出1.5V、2.5V、3.3V三种电源, 输出电流均大于1.5A, 芯片考虑选用LT1585 或LT1086, 此模块还应该具有上电自复位及看门狗功能, 计划选用MAX706/MAX705。同时为了便于系统调试及外部扩展, FPGA预留一并行数据采集口, 以74LVT16245驱动, FPGA其他暂时未用I/O以测试点引出。

其中FPGA和ARM CPU 分别完成的具体功能如下所示:

◆FPGA主要完成解码模块的实现, 具体包括:

- 1) 码流分析;
- 2) 纹理解码;
- 3) 运动补偿;
- 4) 图像重建;

- 5) 色度空间转换;
- 6) 行场同步产生以及图像分辨率转换;
- 7) AMBA AHB总线规范;

◆ARM7TDMI用来构建解码的测试系统:

- 1) 通过JATG或UART口向FLASH写入编好的图像数据流;
- 2) 定时向FPGA发送存储在FLASH中的ES/PS流。

§ 3.3 MPEG-4 解码模块的实现

该模块实现了从 ES 流到可供显示器显示的信号的解码过程。首先, 进入 FPGA 的 ES/PS 流被缓存到 FIFO 中, 其中 FIFO 的大小是 3M bits, 实践表明它可以应对各种码速率的码流输入; 然后 FPGA 读取 FIFO 中的数据进行解码, 外扩的 16Mbits 的 SDRAM 就作为必要的中转; 解码出的数据是 4:2:0 的 YCrCb 信号, 通过色度空间转换变为 RGB 信号, 再加上用于控制显示的行、场同步以及消隐信号, 传送给 ADV7120 转化成模拟信号, 在显示器上看到实际解出的画面。

3.3.1 解码流程框图

下面对 MPEG4 SP 的解码流程作一个完整的介绍。首先给出解码框图, 然后按照解码的先后顺序对每一个功能模块进行详细的说明。

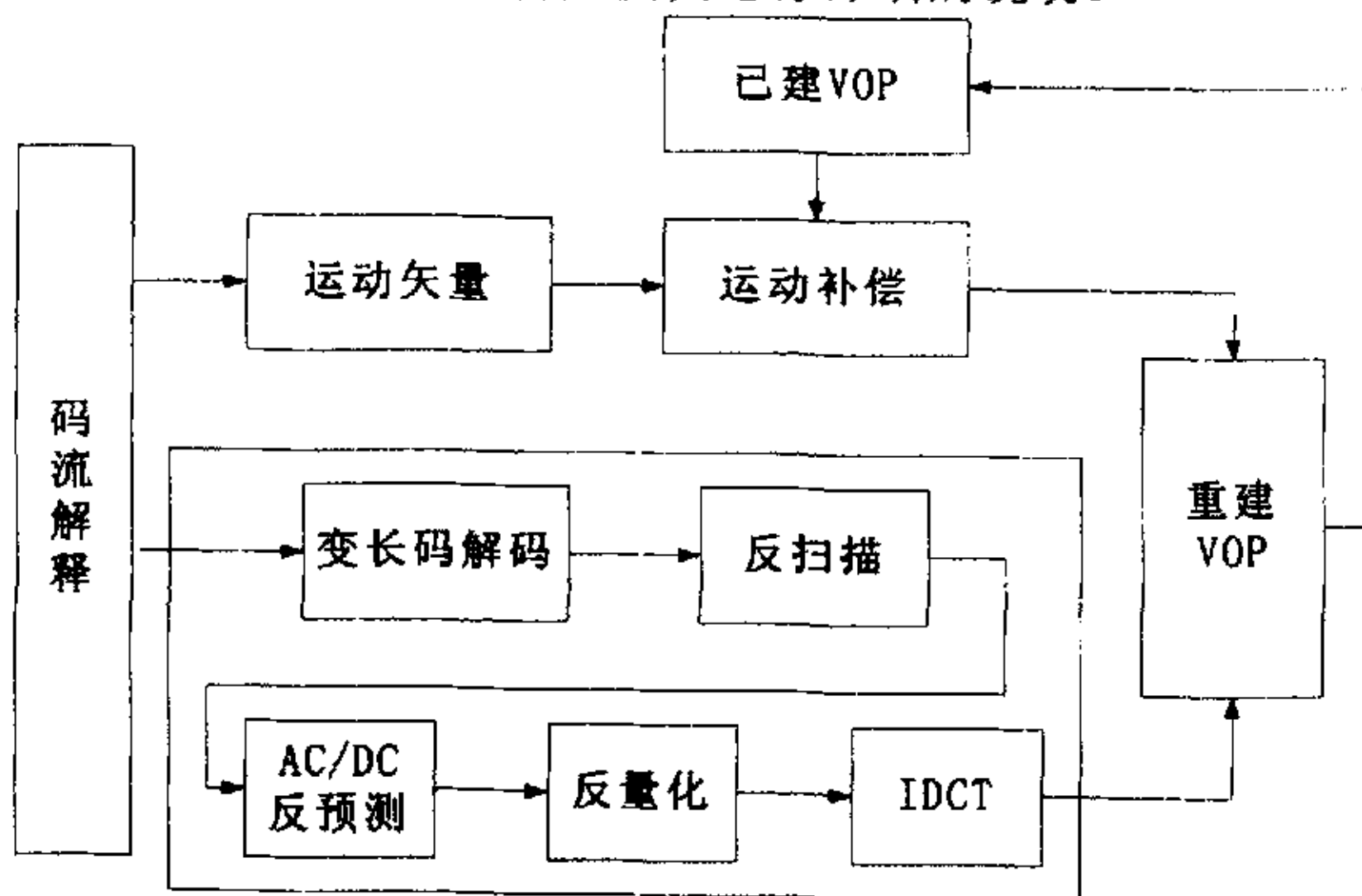


图 3.2 MPEG-4 SP 级解码器框图

如图 3.2 所示, 它给出了基本的解码流程。其中, 码流解析就是从码流中得到头信息, 并把纹理数据送交纹理解码过程, 运动数据送交运动解码过程。下面按照解码的顺序, 逐个进行介绍。在解码流程介绍中, 将先介绍纹理解码, 再介绍运动解码。

3.3.2 码流解释原理及其实现

码流解析就是从编码码流中解析到头信息,并把对我们有用的信息保存下来。接着将纹理数据交给纹理解码模块,运动数据交给运动解码模块,实际上也就是把对码流的控制权交给它们。

将输入基本视频流解析,是一个拆包的过程。由于其码流结构不同,对于 shorthead 和 mpeg4 有不同的方法。这部分功能主要可分为三部分:

(1)固定长度码解码: slice 层以上码流每部分为固定长度编码,这部分主要完成找同步头(包括 vol 头,vop 头,slice 头),提取各种头信息。

(2)变长码解码: 宏块层采用了变长码编码,这部分主要解析运动矢量,dct 系数。其结构框图 3.3,这部分是码流解释的核心。

这里我们采用了状态机和 Lei-Sun 基于 PLA 的 VLC 解码算法。首先必须根据已经解码的信息来判断该查那一个码表。码表之间的转换是通过状态机的转换来实现的。然后用 Lei-Sun 算法进行 VLC 解码。

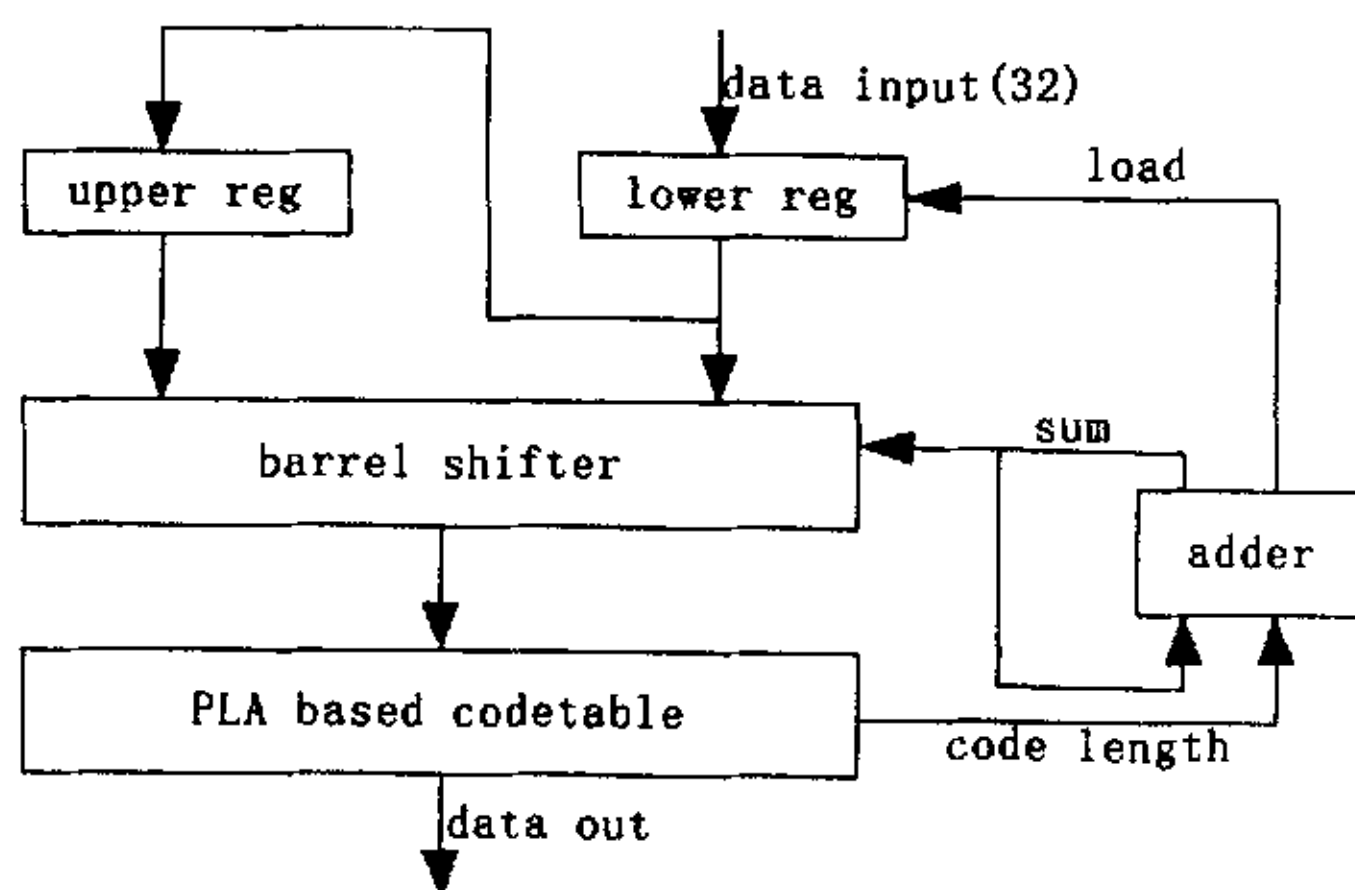


图 3.3 变长码解码的结构框图

图 3.3 中,输入码流每次读入 4 个字节(从 FIFO 中)放入 lower reg 中,而 lower reg 中的内容存入 upper reg 中,barrel shifter 把 64 位码流左移 sum 位。PLA based codetable 根据当前的状态查相应的码表,输出对应信息和 code length。Adder 负责累加码字长度,指示 barrel shifter 要左移的位数,当累加至 32 后,发出读取码流指示 load。

(3)运动矢量差分解码;运动矢量采用的是差分编码,变长码解码解出差分矢量,这部分解出绝对矢量,提供给补偿模块。

3.3.3 纹理解码原理及其实现

这一部分讲述了 VOP 的纹理解码过程,此过程的框图如图 3.4 所示:

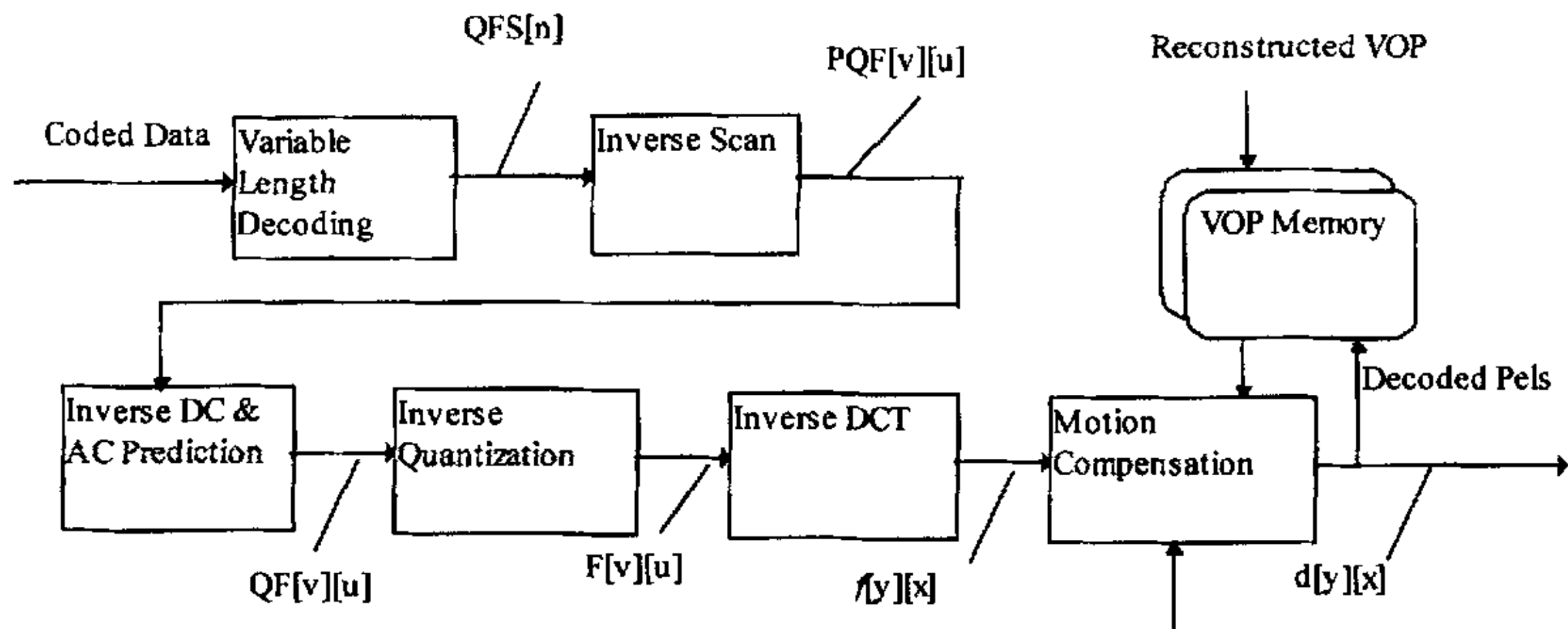


图 3.4 VOP 的纹理解码流程图

3.3.3.1 反扫描

这一部分讲述了一维数据 $QFS[n]$ 转换为二维数组 $PQF[v][u]$ 的方法。其中， $QFS[n]$ 表示变长解码输出端的数据， $QF[v][u]$ 表示反扫描的结果。(其中 n 的范围从 0 到 63， u 和 v 的范围从 0 到 7)。

图 3.5 定义了三种扫描方式，究竟使用何种方式由下面决定。对于帧内宏块，如果 $acpred_flag=0$ ，对宏块中的所有块使用锯齿形扫描。否则 DC 预测方向决定了块的扫描方式，举例说明，如果 DC 预测参考水平相邻块，那么使用垂直交替扫描；反之如果 DC 预测参考垂直相邻块，那么使用水平交替扫描。对所有其他形式的块，变换系数的 8×8 块用锯齿形扫描。

0	1	2	3	10	11	12	13
4	5	8	9	17	16	15	14
6	7	19	18	26	27	28	29
20	21	24	25	30	31	32	33
22	23	34	35	42	43	44	45
36	37	40	41	46	47	48	49
38	39	50	51	56	57	58	59
52	53	54	55	60	61	62	63

0	4	6	20	22	36	38	52
1	5	7	21	23	37	39	53
2	8	19	24	34	40	50	54
3	9	18	25	35	41	51	55
10	17	26	30	42	46	56	60
11	16	27	31	43	47	57	61
12	15	28	32	44	48	58	62
13	14	29	33	45	49	59	63

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

图 3.5 (a) 水平交替扫描

(b) 垂直交替扫描

(c) 锯齿形扫描

3.3.3.2 帧内宏块系数的预测

这一部分讲述了解码系数的预测过程。这个预测过程只在非短头格式下的帧内宏块中进行。当短头格式情况下，或者此宏块不是内部宏块时，这个过程不使用。如果需要作预测，则 DC 预测肯定要做，AC 预测是可选择的。

DC 与 AC 的预测方向的自适应选择是基于被解码块周围的水平和垂直梯度的比较上的。图 3.6 显示了被解码块周围的三个块。块“X”、“A”、“B”和“C”分别代表了当前块、左边的块、左上角的块和上边的块。

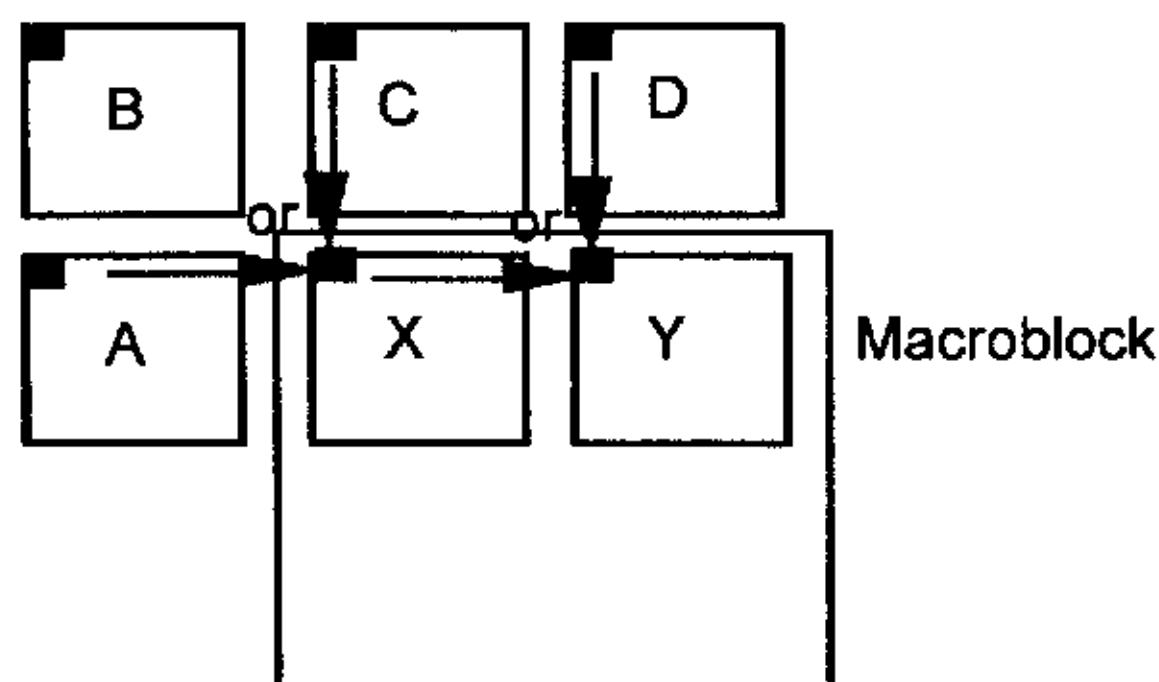


图 3.6 DC 预测中使用的邻居块

先前解码块的 DC 系数的逆量化值 $F[0][0]$ ，用如下方法来确定 DC 和 AC 的预测方向：

if ($|F_A[0][0] - F_B[0][0]| < |F_B[0][0] - F_C[0][0]|$)

predict from block C

Else

predict from block A

如果 A、B 或 C 中任意块超出了 VOP 的边界或视频包的边界，或者它们不属于帧内编码块，那么此块的 $F[0][0]$ 用 1024 作预测值。

对于 AC 预测，先前编码块的第一行系数或第一列系数都用来预测当前块的系数。在一个块基准下，DC 系数预测的最佳方向也作为 AC 系数的最佳预测方向。因此在一个宏块内，预测每一个块可以使用的方式是独立的。

3.3.3.2 反量化

系数的二维数组 $QF[v][u]$ ，通过逆量化产生重建 DCT 系数。上图描述了逆量化的全部过程通过适当的逆量化算法得到系数 $F'[v][u]$ 后，经饱和运算得到了 $F''[v][u]$ ，最后进行错误匹配控制操作，得到最终的重建 DCT 系数 $F[v][u]$ 。

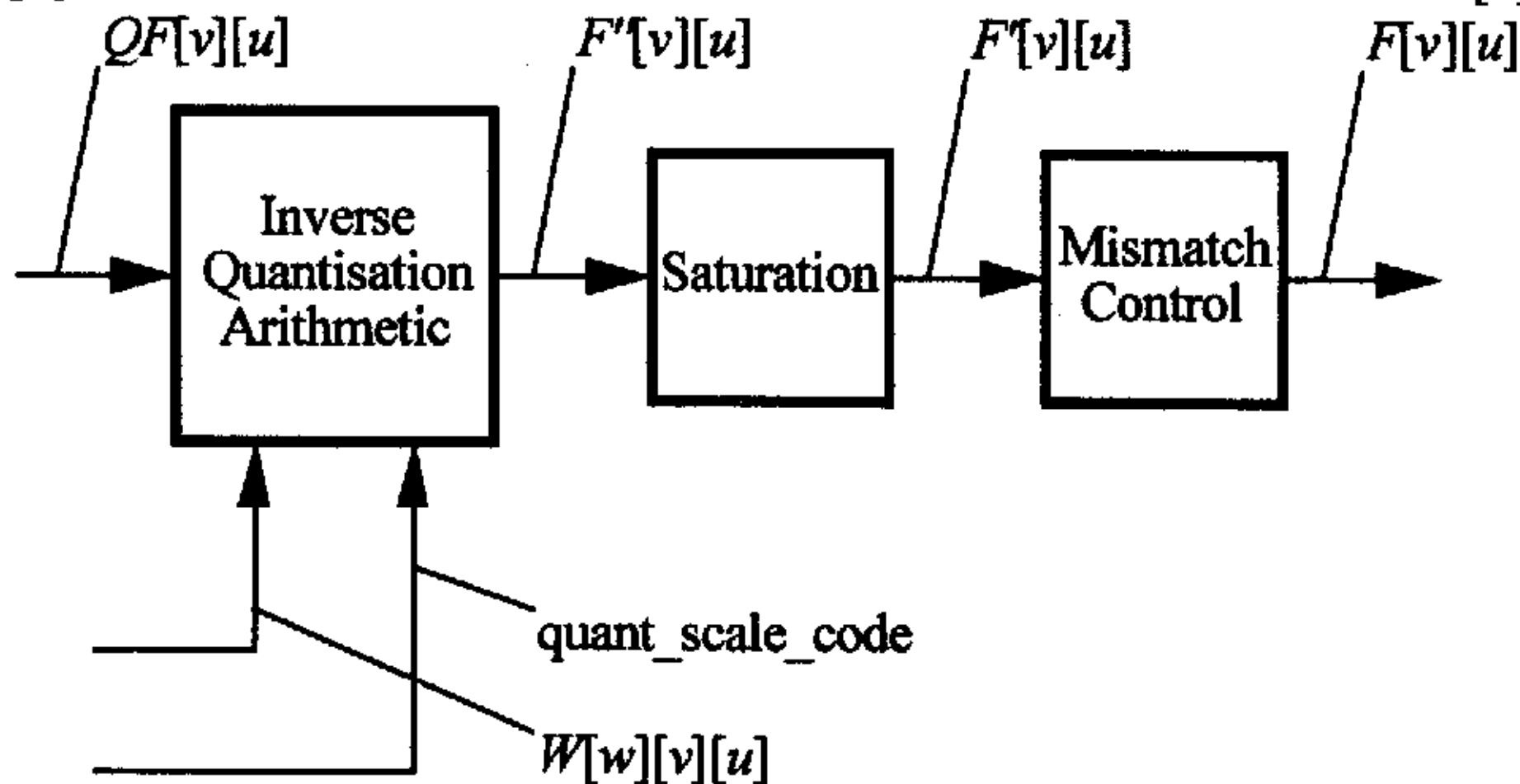


图 3.7 逆量化的过程

量化是针对 DCT 变换系数进行的, 量化过程就是以某个量化步长去除 DCT 系数。对一个 8×8 的 DCT 变换块中的 64 个 DCT 变换系数采用不同的量化精度, 以保证尽可能多地包含特定的 DCT 空间频率信息, 又使量化精度不超过需要。DCT 变换系数中, 直流系数对视觉感应的重要性较高, 因此分配的量化精度较细; 交流系数对视觉感应的重要性较低, 分配的量化精度较粗, 通常情况下, 一个 DCT 变换块中的大多数高频系数量化后都会变为零。

在我们的解码器硬件实现中, 帧内宏块的直流系数用 MPEG4 反量化方法, 交流系数用 H.263 反量化方法; 帧间宏块的系数用 H.263 反量化方法。下面分别介绍这两种反量化方法:

(1) MPEG4 反量化方法

对于帧内编码的 DCT 系数的 DC 分量(INTRADC):

$$F''[0][0] = dc_scaler * QF[0][0]$$

(2) H.263 反量化方法

对于除帧内编码的 DCT 系数的 DC 分量(INTRADC), 其它为

$$\begin{aligned} |REC| &= 0 && \text{若 } LEVEL = 0 \\ |REC| &= QP \times (2 \times |LEVEL| + 1) && \text{若 } LEVEL \neq 0, QP \text{ 为奇数} \\ |REC| &= QP \times (2 \times |LEVEL| + 1) - 1 && \text{若 } LEVEL \neq 0, QP \text{ 为偶数} \\ REC &= sign(LEVEL) \times |REC| \end{aligned}$$

其中 QP 为量化参数, LEVEL 为量化后的变换系数; REC 为反量化后的变换系数。反量化后的数据截短为 12 位, 传给 IDCT 模块。

反量化模块的实现如图 3.8 所示

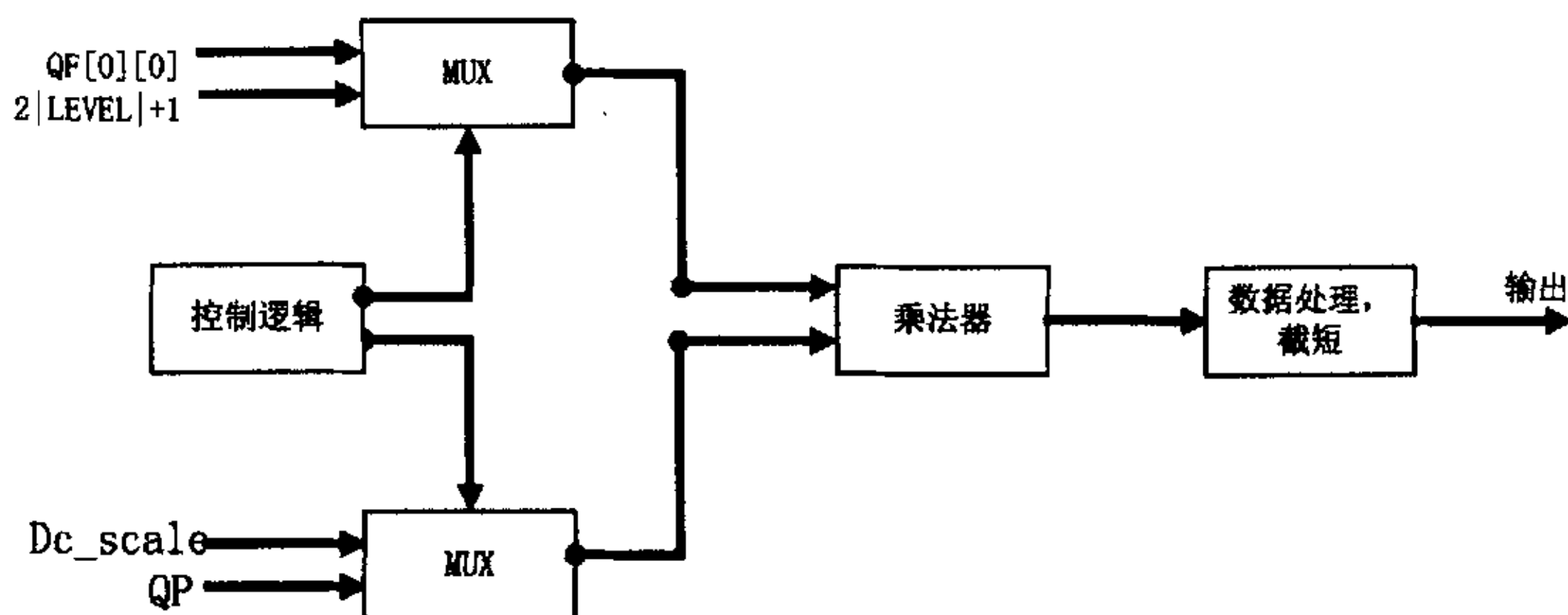


图 3.8 反量化模块的实现如图所示

3.3.3.3 逆 DCT 变换

逆 DCT 变换定义为:

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

$$u, v, x, y = 0, 1, 2, \dots, N-1$$

其中 x, y 是原始域中的空间坐标

$$C(u), C(v) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } u, v = 0 \\ 1 & \text{otherwise} \end{cases}$$

u, v 是变换域中的空间坐标

二维 IDCT 可以分解成二次一维 IDCT 运算, 如以下公式。

$$X(u) = \sum_{v=0}^{2N} F(u, v) \cos \frac{(2y+1)v\pi}{2N}$$

$$f(x, y) = \frac{2}{N} \sum_{u=0}^N X(u) \cos \frac{(2x+1)u\pi}{2N}$$

基于以上的原理在实际应用中通过两次 1-D IDCT 变换来完成 2-D IDCT 变换, 这种方法通常被称为行—列分离法。这种方法在结构上的对称性更好, 并且可以重复使用硬件资源, 所以这种方法被广泛的使用。

基于 RAC 的一维 IDCT 结构如图 3.9 所示:

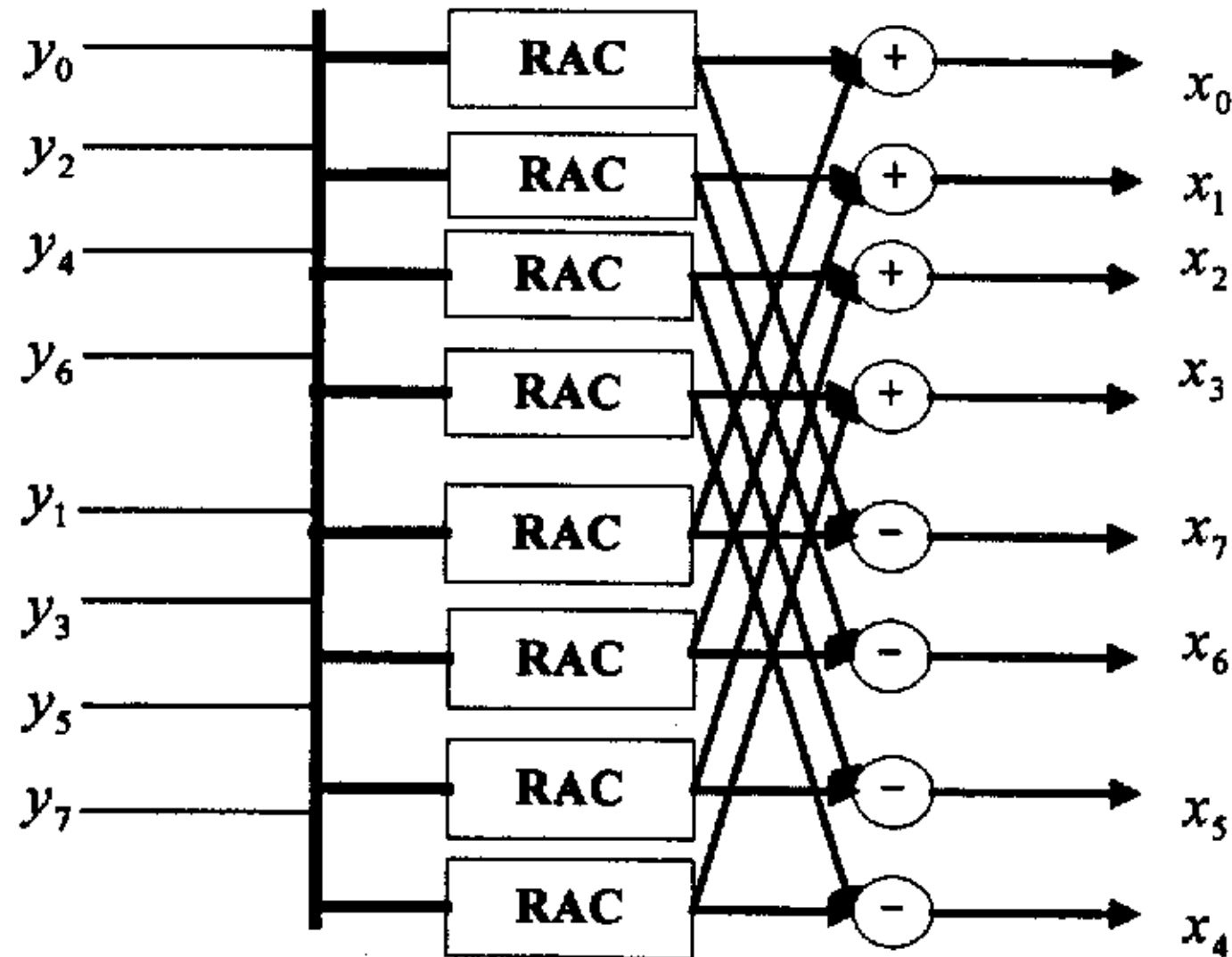


图 3.9 基于 RAC 的一维 IDCT 结构框图

基于 RAC 的 DCT/IDCT 实现, 需要两个一维 DCT/IDCT 变换模块, 分别做二维 DCT/IDCT 的行变换和列变换。在它们两之间还需要一个双端口 RAM 用来

存储变换的中间结果。因为一维 DCT/IDCT 变换模块的计算输入和输出是同时有好多个数据并行运算的,所以在模块的输入端加一个串并变换模块,结果的输出端加一个并串变换模块。

3.3.4 运动补偿解码

运动补偿解码过程包括,根据从码流中解析到的运动矢量的数据得到当前宏块的运动矢量(可能不止一个),然后根据当前宏块的运动矢量,从参考帧中找出相应的参考宏块。如果运动矢量不是整象数值,还需要通过内插来获得参考宏块。

一个帧间宏块可以只有一个运动矢量;也可以有四个运动矢量,每个亮度块都对应一个运动矢量,这种模式也被称为四矢量模式。究竟是使用一个运动矢量还是四个运动矢量由码流中的 mcbpc 变长码字决定,准确来说是由 mcbpc 给出的 mb_type 决定。当 mb_type 等于 0 时,只有一个运动矢量,当 mb_type 等于 1 时,有四个运动矢量。

3.3.5 VGA 后处理模块的实现

此模块实现的主要功能有:

(1)将从前端 SDRAM 读取并送来的 4:2:0 的 YCrCb 信号,通过色度空间转换变为 RGB 信号。

(2)产生用于控制显示的行、场同步以及消隐信号。

(3)将分辨率为 352x288 的信号扩展为 640x480 的信号。

模块按功能基本可以分为三部分:读宏块控制逻辑以 40Mhz 的时钟频率从 SDRAM 取出亮度和色度的实时数据,这些数据经过色度空间转换,换算成 RGB 信号作为输入数据送入双口 RAM 中暂存,这部分可称之为色度空间转换部分;第二部分用于产生视频控制信号,包括行、场同步及消隐信号等;双口 RAM 和控制它的读写控制逻辑构成第三部分,通过写控制逻辑把前端得到的 RGB 信号按宏块顺序存储于 RAM 中,当存储的数据达到某一标准后,读控制逻辑将对双口 RAM 进行读出操作,在读出数据时会同步完成分辨率的扩展。最后,由右侧双口 RAM 读出的 RGB 信号在视频控制信号的控制下将通过 ADV7120 芯片送往显示器显示。

§ 3.4 解码数据流模块的实现

这个模块利用 ARM 定时器产生一定频率的时钟信号并产生定时中断,以此定时中断来驱动 8 位数据总线发送一定速率的 ES 流。所需要的图像数据流(已编码好的 ES/PS 数据)可以通过 JATG 或 UART 口写入 FLASH 中,在测试时循环发送这些数据以检测 FPGA 是否可以正确解码,在需要时还可以通过相应的通信端口更新 FLASH 中的图像数据。

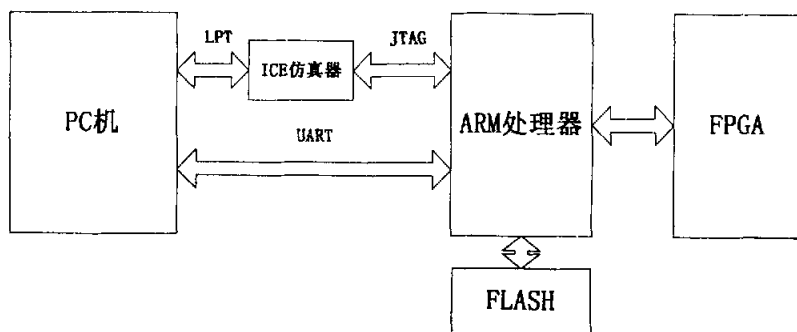


图 3.10 测试系统设计框图

图 3.10 中与 FPGA 接口包括：一个 8 位并行数据总线；一路同步时钟信号；一个发送允许控制端；两个预留端口；负责传输图像数据。图中的 Flash 中烧制好的三段数据分别是：引导程序、发送程序和图像数据。下面系统加电时，ARM 处理器会自动运行引导程序，主要是对 ARM 进行一些初始化工作并把发送程序和图像数据载入到 SDRAM 中，当上述工作完成后，系统就会从 SDRAM 运行发送程序从而实现 ARM 处理器发送测试图像的功能。

下图是解码器解码测试序列得到的实际显示效果图：



第四章 MPEG-4 解码器的数据存储

§ 4.1 引言

目前, 图像处理系统中都需要用到容量大、读写速度高的存储介质。SRAM 操作简单, 但昂贵的价格, 使得产品成本上升。相比较而言, SDRAM 的控制较 RAM 复杂, 但具有价格便宜、体积小、速度快、功耗低等优点, 所以从降低成本的角度出发, 我们采用 SDRAM 作为 MPEG-4 SP 级解码器的数据缓冲。

本章提出了一种在 MPEG-4 SP 级解码器中的数据存储方案, 设计了解码器 IP 核与 SDRAM 的接口, 并巧妙地利用了一种新颖的填充方法, 使得程序执行的效率大幅度提高。

§ 4.2 SDRAM简介

SDRAM 是高速的动态随机存取存储器, 它的同步接口和完全流水线的内部结构使其拥有极大的数据速率, 目前 SDRAM 时钟频率已达 100MHz 以上。另外他们的行、列地址线共用, 由行地址选通(CAS)、列地址选通(RAS)信号分时控制。SDRAM 内部以 bank 为组织, 可由行、列地址寻址。另外为了保持内部数据还必须进行刷新。

SDRAM 具有多种工作模式, 内部操作是一个复杂的状态机。SDRAM 器件的管脚分为以下几类:

(1)控制信号: 包括片选, 时钟, 时钟使能, 行列地址选择, 读写选择, 数据有效。

(2)地址: 时分复用管脚, 根据行列地址选择管脚, 控制输入的地址为行地址或列地址。

(3)数据: 双向管脚。

SDRAM 的所有操作都同步于时钟。根据时钟上升沿控制管脚和地址输入的状态, 可以产生多种输入命令:

(1)模式寄存器设置命令;

(2)激活命令;

(3)预充命令;

(4)读命令;

(5)写命令;

(6)带预充的读命令;

(7)带预充的写命令;

(8)自动刷新命令;

(9)自我刷新命令;

(10)突发停命令。

另外还有空操作命令。

根据输入命令,SDRAM 状态在内部状态间转移。图 4.1 是 SDRAM 器件的状态转移图。

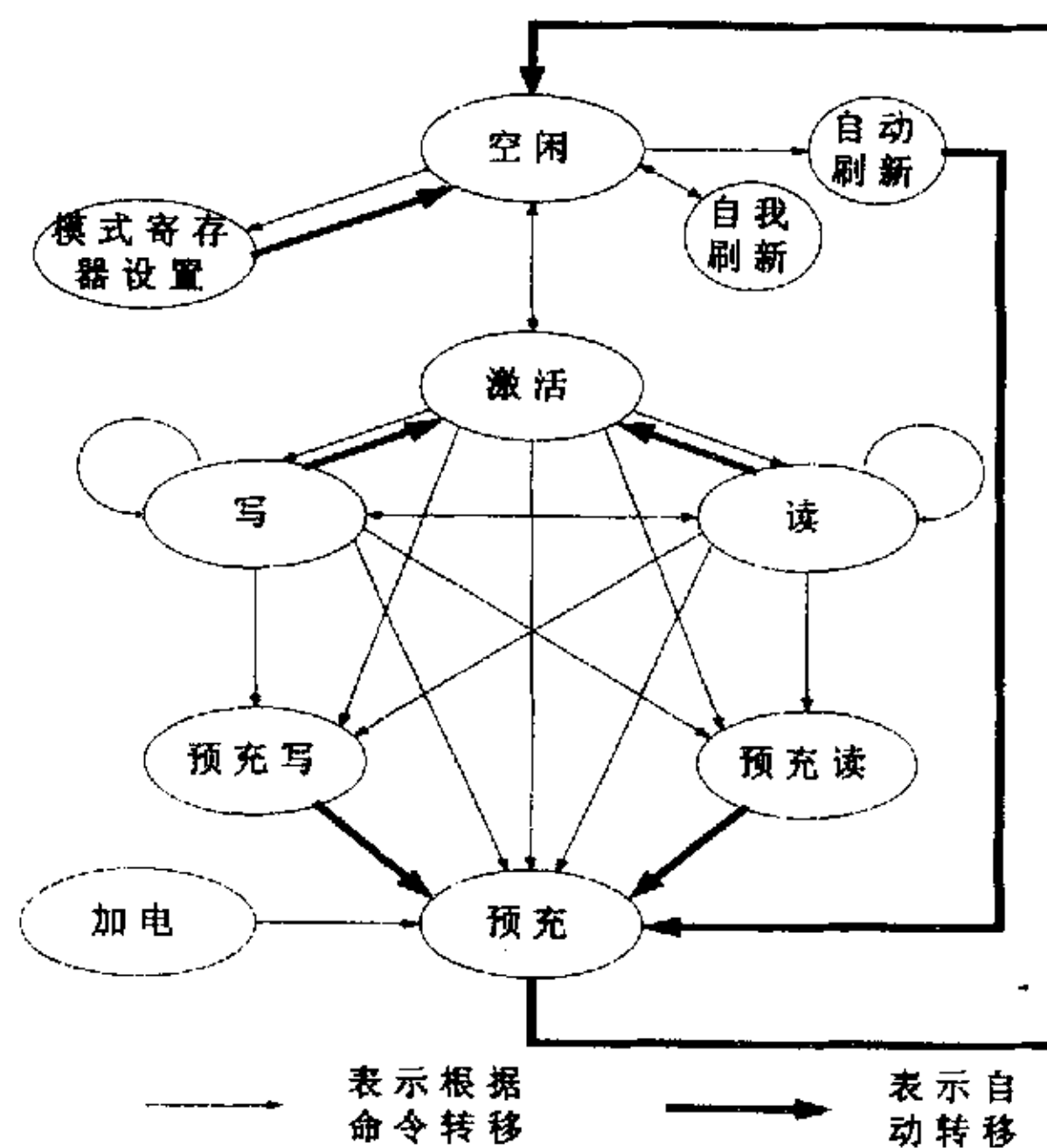


图 4.1 SDRAM 器件的状态转移图

§ 4.3 SDRAM接口电路的整体设计

4.3.1 整体的时序和结构安排

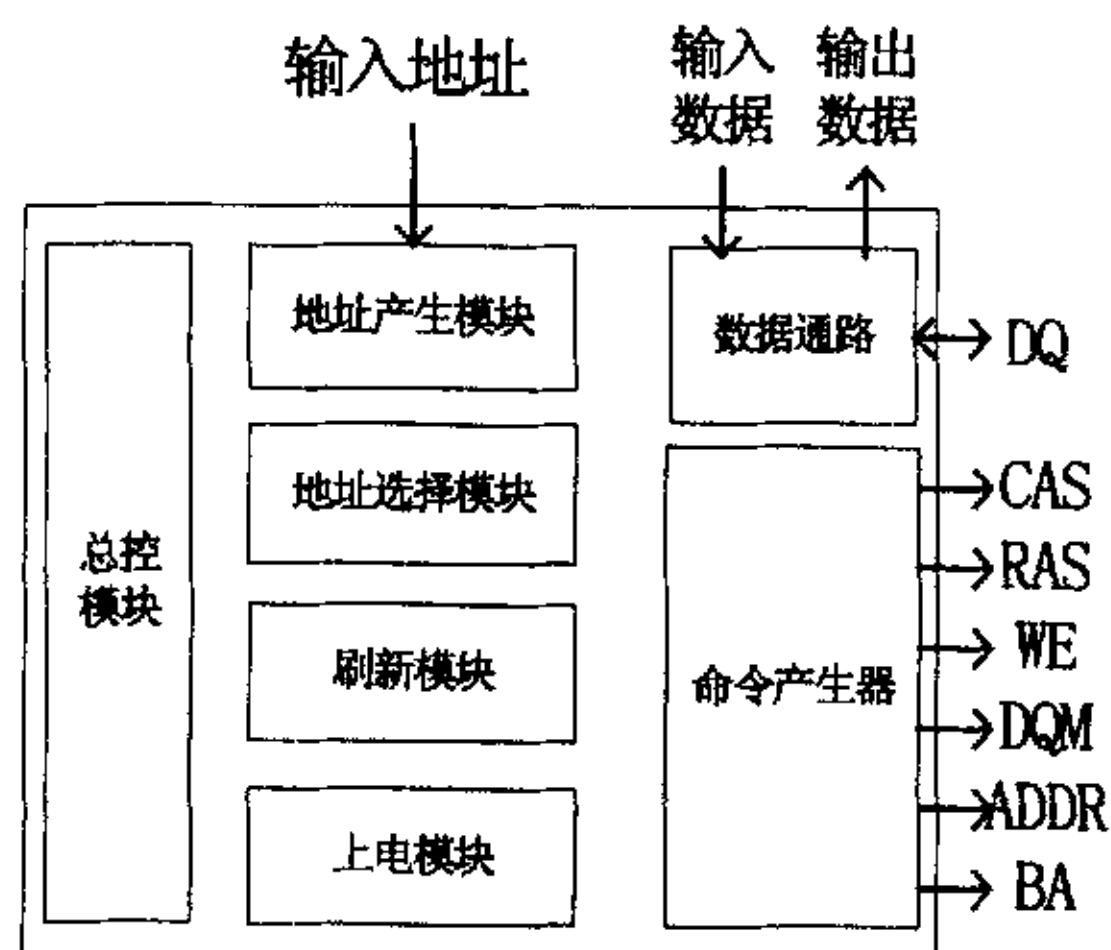


图 4.2 SDRAM 的接口电路结构框图

图 4.2 是 SDRAM 的接口电路结构框图。SDRAM 接口负责 SDRAM 与 FPGA

内部模块的数据交互,其中命令产生器与 SDRAM 直接打交道,产生 SDRAM 能识别的控制信号;地址产生模块负责地址映射,因为总控制器给出的地址是解出的宏块在重建帧中的相对位置和 P 帧中宏块对应的 MV,经过地址映射后,就变成 SDRAM 命令产生器能识别的地址;上电模块完成正常工作前的 SDRAM 初始化。SDRAM 总控模块控制着整个接口,主要负责启动和控制相关模块。总控模块在上电时启动了上电模块;在场消影期启动刷新模块;解码时与 SDRAM 的数据交互主要分三部分,分别为:读显示数据,填充并且读参考帧数据,写入解出的宏块数据,在这个期间总控模块的控制信号启动相应的地址映射模块产生读写地址,并由地址选择模块负责分配。

4.3.2 存储策略和地址映射策略

在 SDRAM 中总共存储了两帧图像的数据,其中一帧作为参考帧,在解 P 帧时要根据运动矢量提供参考宏块的数据形成重建帧,同时它还用来做显示数据缓冲;另外一帧是写入正在解出的重建帧数据。解码器处理的图像格式为 CIF,数据格式为 4:2:0,所以每一帧占用空间为 $352 \times 288 \times 1.5 \times 8$ bits。SDRAM 中的两个 bank,分别存放正在重建的帧和参考帧。每一个 bank 被分成三个连续空间,一块用来顺序存储亮度信息,另外两块分别顺序存储色度信息。在下一个帧起始信号到来时,原来存储参考帧的 bank 变成存储重建帧,而另外一个 bank 中存储的解完的重建帧将作为正在重建的下一帧的参考帧。

§ 4.4 SDRAM 接口的主要任务

解码器工作时是按照宏块作为单位处理的。解一个宏块时,首先从已经存在 SDRAM 中的参考帧中读取参考宏块数据,加上残差数据后得到解出的宏块,存储到 SDRAM 中;而且给显示模块的数据也是以宏块为单位提供的。所以考虑到解码器工作的实际情况,我们在选择 SDRAM 的工作模式是长度为 8 的突发读写,这样可以高效的存储数据。

SDRAM 的具体任务分为以下几个部分:

(1)和解码模块的数据交互

SDRAM 主要和解码模块和显示模块进行大量的数据交互,具体的数据流向(实线)和地址线(虚线)如图 4.3 所示

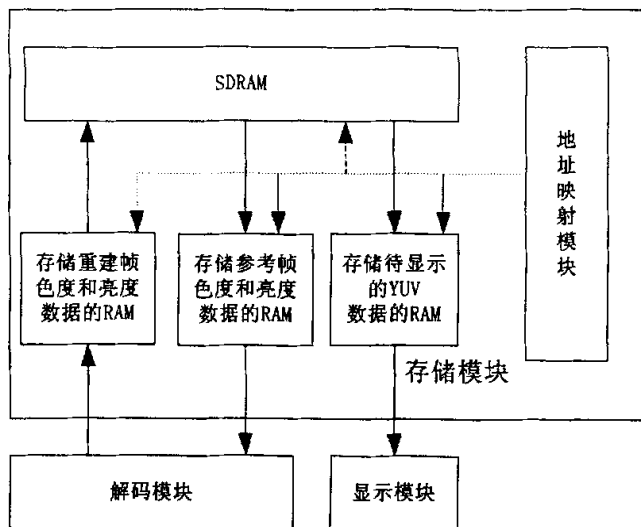


图4.3 接口设计中的RAM分布

图4.3中的RAM在FPGA内部，负责缓存SDRAM读出和将要存储的数据。本方案存储以宏块为基本单位，所以每个RAM存储的都是一个宏块的数据。其中：存储重建帧色度和亮度数据的RAM和存储待显示的亮度和色度数据的RAM大小都为384bytes，包括亮度的 16×16 和色度的 $8 \times 8 \times 2$ 个像素点；存储参考帧色度和亮度参考数据的RAM大小为486bytes，其中包括亮度的4个 9×9 的像素点和色度的 $9 \times 9 \times 2$ 个像素点。MPEG-4解码器支持半像素的精度，所以先根据半像素精度的运动矢量取出一个 9×9 的块，再进行半像素插值成 8×8 的宏块以后，与残差相加得到重建宏块。

(2) 填充

MPEG-4 支持非限制运动矢量，所以填充就成为数据存储操作的一个重要内容。在这里本文提出了一种简便的方法，该方法在把数据存储到SDRAM中的时候，并不进行填充操作，而是在读出数据的时候用了一种等价的方法，得到了相同的结果，大大减少了数据操作的复杂度。使得操作变得简单易行。

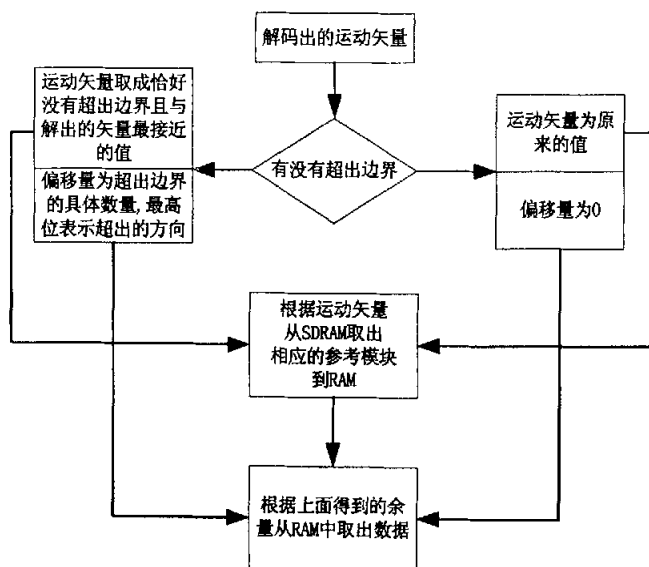


图 4.4 填充具体的实现过程

具体的实现过程如图 4.4 所示, 在程序中增加了一个变量这里称为“偏移量”。解码出来的矢量如果没有超出图像的边界, “偏移量”值设为 0; 如果判断, 超出了边界, 则送给读取 SDRAM 模块的矢量取成刚好没有超出边界但离解出的运动矢量最接近的值, 并把“偏移量”取成超出边界的象素数, 并表明超出的方向。

求解出偏移量, 要根据 16×16 宏块的位置和解码出的四个 8×8 块运动矢量的大小:

```

if 宏块位置的位置位于边界处 and 宏块的四个  $8 \times 8$  块有超出边界
then

```

```

    按照上述的规则分别设置四个  $8 \times 8$  块的运动矢量和偏移量;

```

```

else

```

```

    运动矢量不变;

```

```

    偏移量  $\leq 0$ ;

```

```

end if;

```

如果偏移量不等于 0, 从 SDRAM 读到 RAM 中的数据并不是我们想要的数, 里面有很多多余的数据, 但是我们可以根据“偏移量”的值, 在从 RAM 读取补偿数据时, 配合上适当的逻辑, 准确的取出相应的非限制运动矢量对应的数据块, 然后和残差数据相加得到了最后的重建块。

以位于左上角 8×8 的块为例:

假设此时运动矢量的横纵坐标分别为 $MV_x = -3$, $MV_y = +2$;

那么变换后的运动矢量取成: $MV_x = 0$, $MV_y = 0$;

偏移量取成: $OFFSET_x = -3$, $OFFSET_y = +2$;

0	0	0	0	0	1	2	3	4	5	6	7
0	0	0	0	0	1	2	3	4	5	6	7
0	0	0	0	0	1	2	3	4	5	6	7
0	0	0	0	0	1	2	3	4	5	6	7
0	0	0	0	0	1	2	3	4	5	6	7
8	8	8	8	8	9	10	11	12	13	14	15
16	16	16	16	16	17	18	19	20	21	22	23
24	24	24	24	24	25	26	27	28	29	30	31
24	24	24	24	32	33	34	35	36	37	38	39
40	40	40	40	40	41	42	43	44	45	46	47
48	48	48	48	48	49	50	51	52	53	54	55
56	56	56	56	56	57	58	59	60	61	62	63

图 4.5

0	0	0	0	1	2	3	4
0	0	0	0	1	2	3	4
0	0	0	0	1	2	3	4
8	8	8	8	9	10	11	12
16	16	16	16	17	18	19	20
24	24	24	24	25	26	27	28
32	32	32	32	33	34	35	36
40	40	40	40	41	42	43	44

图 4.6

这样从 SDRAM 中取出存到 RAM 中的数据为图 4.5 中的非阴影区,而实际需要的是图 4.6 中的地址存放的数据。为了达到目的,在读出 RAM 中数据时利用变量 OFFSET,通过一个简单的逻辑,把图 4.5 非阴影区的地址转换成图 4.6 中的地址空间,从而达到一个等效填充的结果。

(3)刷新:在解码器中选用的 SDRAM 必须在 128ms 内至少刷新 4096 次。在时序安排上主要利用了显示时的消影期进行集中刷新。这样就避免了分散刷新有可能带来的一些控制和数据存储延时方面的问题,而且也完全满足 SDRAM 的刷新次数的要求。

§ 4.5 硬件实现

解码器用的 SDRAM 是 ISSI 的公司的 IS42S1600,规格为 512K Words x 16 Bits x 2 Banks (16-MBIT),采用 VHDL 硬件描述语言,SDRAM 接口模块工作在 80Mhz 的频率上,已经成功的应用到了 MPEG-4 的解码器中。通过对各种码流的测试,表明接口模块在各种情况下都可以正确的工作。接口模块占用了 FPGA 的 668 个 slices,518 个 slice flip flops,分别占资源的 8%和 3%。现在我们只是使用了 SDRAM 的 16 位数据总线的低 8 位,如果升级成更高版本的解码器的话,配合上数据拼接模块,还有一半的吞吐潜力可以提升。

第五章 AMBA 总线的硬件实现

§ 5.1 引言

AMBA 总线协议是一个公开的标准。标准中规定了一个系统中的各个功能模块间的连接核管理的方案,可以使得设计者在不需要更改总线接口的情况下,直接整合现有的 IP 将使得系统的整体效益倍增。集成电路设计业界普遍认为,AMBA 是最受欢迎、便于使用和支持广泛的现有互联标准之一。随着第三方知识产权基础领域的扩大,并且更多业界领先的电子设计自动化供应商大力支持电子设计自动化工具,AMBA 技术将为高性能系统界面设计提供全面的解决方案。迄今为止,AMBA 标准被 ARM 90%以上的合作伙伴采用,被 2000 余名工程师通过 ARM 的公司网站下载,并已成功应用于上千个 ASIC 设计中。

AMBA 设计套件提供的类属单机环境有助于迅速创建基于 AMBA 技术的部件和单芯片系统设计。AMBA 设计套件包括齐全的基本部件和数个样品系统设计,为基于 AMBA 界面的设计提供通用基础,从而缩短产品上市周期。另外,AMBA 独立于处理器和技术,增强了各种应用中的外设和系统宏单元的再使用性。正是基于以上的原因,越来越多的 Soc 系统中采用 AMBA 作为系统总线。

§ 5.1 AMBA AHB总线概述

AMBA (Advanced Micro-controllerBus Architecture)中包含了四大部分: AHB, ASB, APB, Test Methodology。其中我们比较常用到的是 AHB 和 ASB。AMBA 系统总线中的 AHB 主要针对高效率,高频率以及高带宽系统模块组所设计的总线,它可以连接微处理器,片上或片外的存储器, DMA 控制器以及存储器接口。而 APB 主要负责连接一些系统低速的外围设备,相对于 AHB 协议比较简单,与 AHB 之间通过 AHB_to_APB bridge 相连。

AHB (Advanced High-Performance Bus) System由Master, Slave, Infrastructure 三部分组成。整个AHB bus上的传输都是由master 所发出的,由slave负责回应。而 infrastructure 由arbiter, master to slave multiplexor, slave to master multiplexor, decoder, dummy slave, dummy master 所组成。

§ 5.2 AHB总线的传输过程

AMBA AHB 总线是一个支持多 master 和多 slave 的总线。其传输的基本过程如下:

首先,多个 master 中的某一个 master 向 arbiter 发出 HBUSREQ 请求信号,然

后由 arbiter 判决是否把总线交给那个 master。如果准备把总线交给这个 master 则给这个 master 发出 HGRANT 确认信号, 同时 multiplexor 根据 arbiter 给出的这个 master 的序号完成 master 到 slave 的总线路由。

在得到 arbiter 的确认后, 这个 master 首先在地址和控制总线上发送地址信息和一些关于本次操作的控制信息, 然后下一个周期在数据总线上发送该地址对应的数据, 与此同时在地址总线上发送下一个数据对应的地址, 依此类推, 总之要保证数据对应的地址要提早一个时钟周期出现, 以此给 slave 有时间来决定是否接受数据或者有无错误发生。同时 decoder 根据 master 发送的地址信息找到它要进行操作的 slave 是哪一个, 并且通过选通信号激活对应的 slave。而 multiplexor 则根据 decoder 的判断结果, 完成 slave 到 master 的总线路由。

被选中的 slave 也要根据它接收的地址信息和控制信息来决定返回的信息(即 HREADY, HRESP[1:0])给 master, 告诉 master 传输能否顺利进行, 是否有错误发生。如果 master 得到肯定的答复则继续传输, 如果不是的话, master 则根据具体的反馈情况, 延长或者取消此次传输。图 5.1 是 AHB 的一次基本传输的时序关系图。

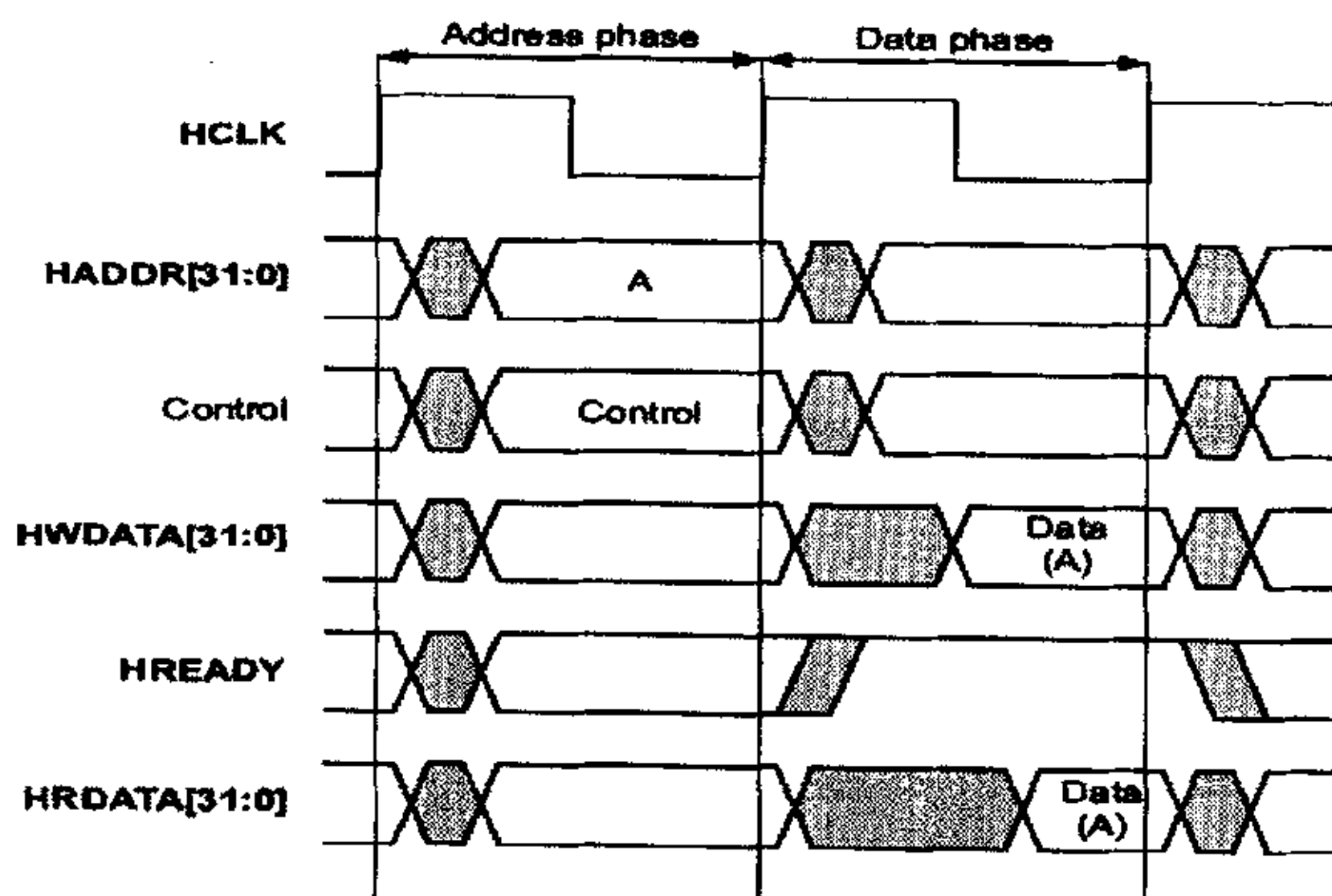


图 5.1 AHB 一次基本传输的时序图

§ 5.3 AHB 总线的硬件实现

本次设计采用的是一个 master 和两个 slave 的系统方案(如图 5.2)。

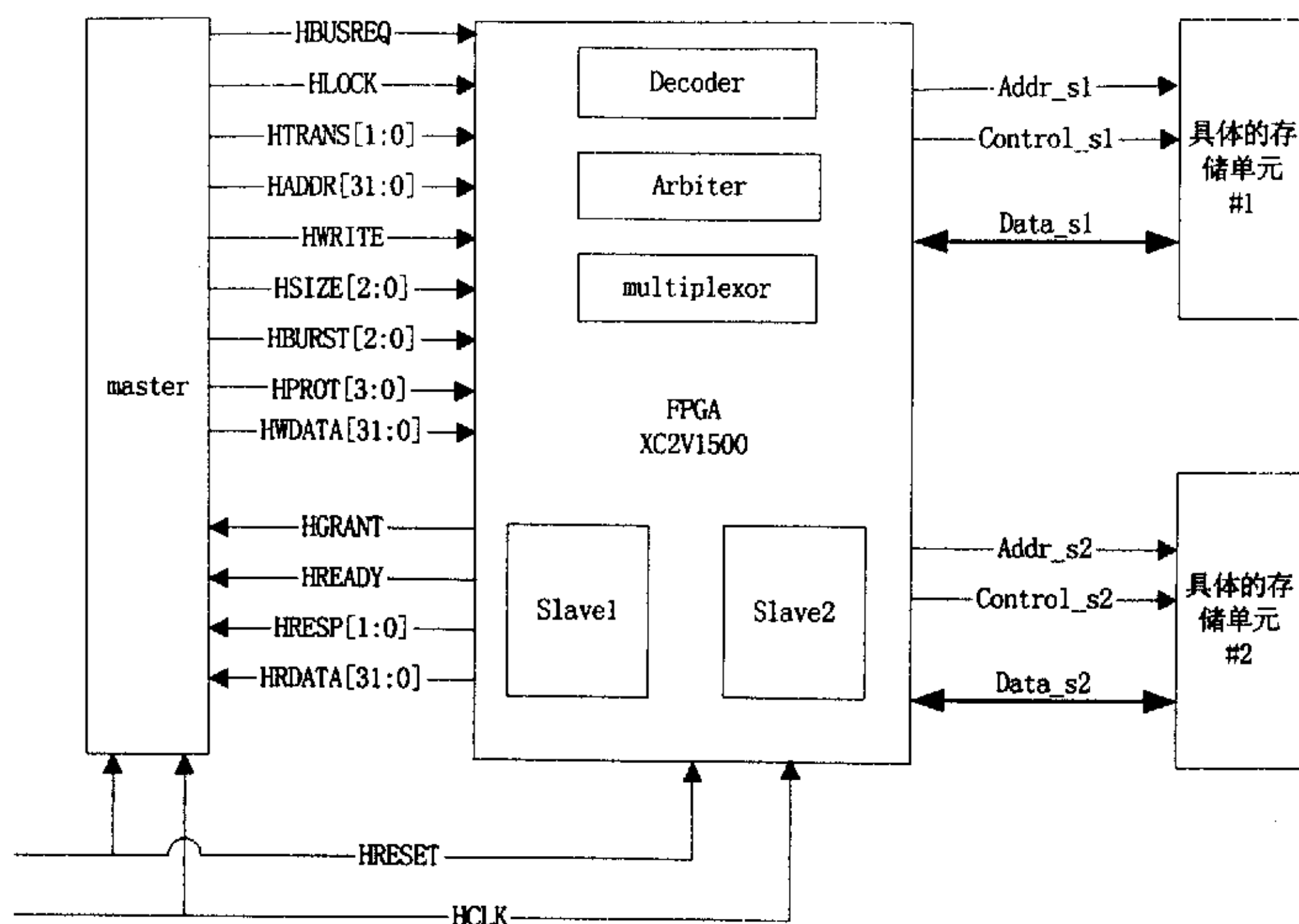


图 5.2 AHB 系统方案图

其中 Master 是 ARM 核处理器，具体的存储单元视具体情况而定，相应的他们的控制和地址，数据信号也要作相应的调整。

FPGA内部模块的具体连接关系如下图:

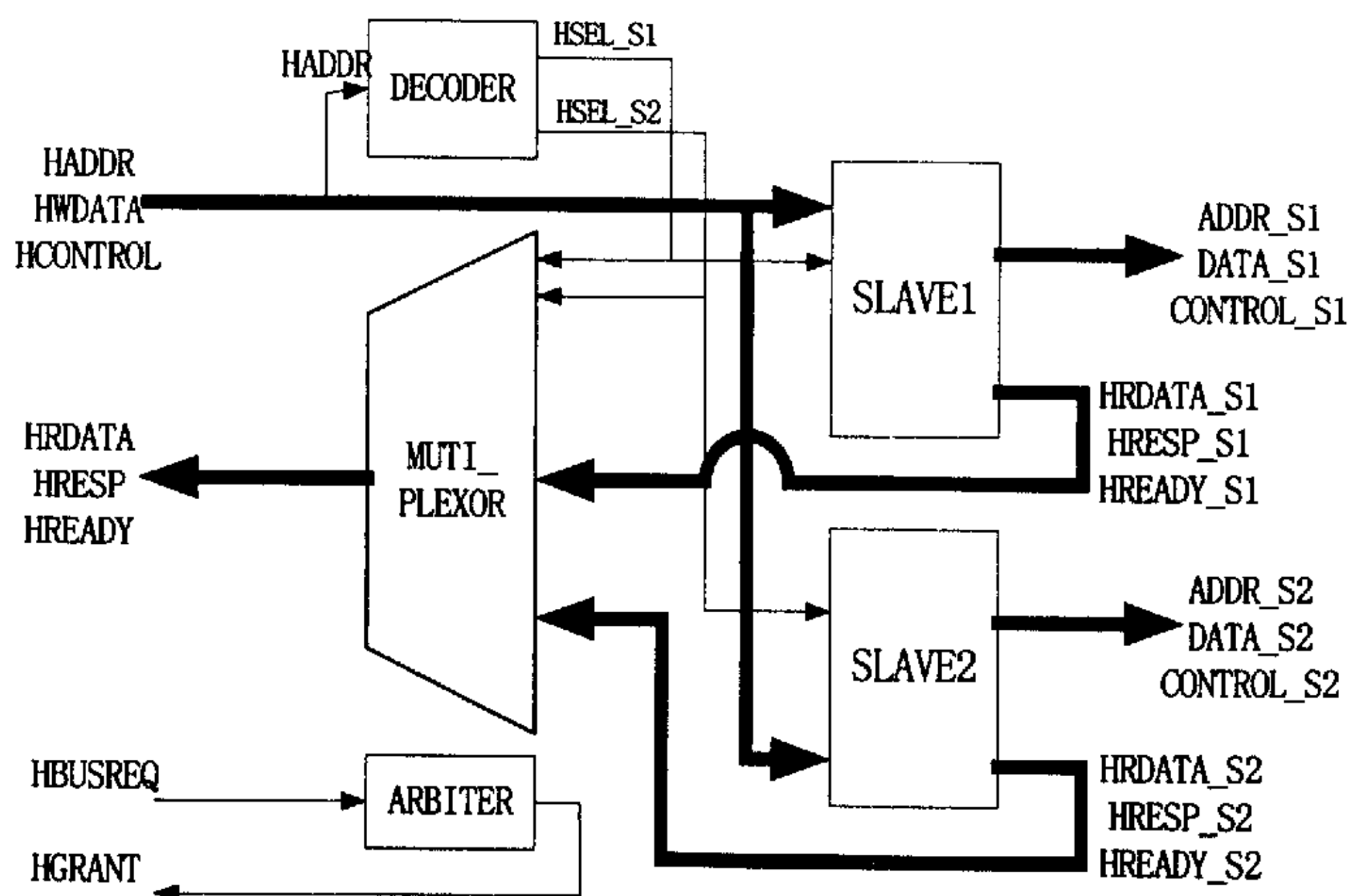


图5.3 FPGA中的各模块间具体连接关系

功能模块的具体描述如下:

(1)slave1, slave2: 负责接收 master 规定格式的数据信号, 另外还要向具体的

存储单元中写入相应格式的数据；或是反向的传输，即由 master 提供的地址信息，从 slave 读取数据后送回到 master。另外，slave 还要根据具体情况向 master 发送一些反馈信号，来表明此次传输的状态。

(2)Arbiter 负责向 master 传送总线确认信号(HGRANT)，向 slave 传一些内部信号。AHB 之所以需要 arbiter，是因为它支持多个 master，因此需要 arbiter 来仲裁。本设计中，只有一个 master 所以可以使得仲裁算法很简单，即只要 master 提出总线请求，即给出确认。

(4)Decoder 负责根据 master 传送的地址数据来判断需要对哪一个 slave 进行操作，向特定的 slave 发送选通信号。

(3)Multiplexor 负责 slave 和 master 之间的总线路由。因为本次设计只用了一个 master，所以由 slave 到 master 的 multiplexor 就省略了，只有 slave 到 Master 方向的 multiplexor。它根据 Decoder 给出的 HSELx 信号，知道那一个 slave 被选择，把选通的 slave 发出的信号连接到 master 上。

结束语

整个视频解码器是一个十分复杂的硬件系统。本文首先详细讨论了 MPEG-4 标准, 及标准所建议的解码模型, 在此基础上提出了基于 FPGA 和 ARM CPU 的 MPEG-4 解码系统的, 把整个系统细分成各个功能模块, 并且详细的讨论了各个模块的功能及其硬件实施方案。

视频解码器需要在短时间内处理大量的数据。因为处理的数据量很大, 视频数据都是放在片外的存储器中, 所以视频处理器和外部 RAM 就需要一个高效的接口来连接它们, 满足处理速度的要求。通过对 MPEG-4 数据流的研究, 并且结合具体的存储器的特点, 我们为解码器选用 SDRAM 作为外部存储的单元, 并且提出了一种总线控制方案, 使得解码器能够及时的处理各种码流。解码时总控制模块, 结合显示模块提供的同步信号, 把整个总线分成三个小的时间片, 分别完成: 读显示数据, 填充并且读参考帧数据, 写入解出的宏块数据, 使得整个解码系统的数据流通畅。

在测试的时候, 我们用 Momusys 的编码软件对多个标准测试序列进行编码, 然后让解码系统显示解码效果, 都达到了较高的主观质量。

在设计的时候我们还考虑到以后的兼容问题, 即为了能使解码器的 IP 核能够很方便的应用到其他的系统中, 和其他 IP 核结合起来, 不需要对核心作太多的修改, 我们还设计了一套 AMBA 总线, 作为 SoC 系统中的系统总线。

如今, 视频解码器的应用范围越来越广, 其中很大一部分就是便携设备, 如 PDA 等。因此除去性能方面要使得存储器能够满足高带宽的要求, 能耗问题就突显成为一个不容回避的问题, 所以我们下一步的任务就是尽可能的降低存储器的能耗。这就需要研究 SDRAM 或者其他存储器的特点, 在存储算法上作相应的变化, 降低能耗。拿 SDRAM 为例, 行激活(row-activation)和列激活(column-activation)这两个操作占能耗的很大一块。而通过一种新的存储方案完全可能使得上述的操作显著的减少, 从而达到降低能耗的目的, 这也就是我们下一步的主要目标。

致谢

首先，深深的感谢我的导师吴成柯教授。在研究生学习阶段，他给予我悉心的指导和帮助。吴老师渊博的知识、严谨的治学态度、一丝不苟的工作作风、平易近人的品德以及对国内外学术动态敏锐的洞察力，都使我受益匪浅。此外，吴老师为我们创造的自由和民主的学术氛围，为我顺利完成学业提供了良好的条件。从吴老师身上感受到的精神和学习到的知识将使我终生受用，并对我今后工作产生有益的影响。

非常感谢我在求学期间，王养利老师、王军宁老师、李云松老师、肖嵩老师等所提供的帮助。他们扎实的理论基础和丰富的实践经验，都使我受益匪浅。

感谢张方、邓稼先、杜建超、赵波、张建龙、庄怀宇、肖江、刘凯、方勇、刘侍刚、王海、纪小明、常骊波、王信、石迎波、吴涛、韩伟、戴洪帅、王柯俨、曹斌、郭高波、王晓方、宋锐、陈晨、朱东巍、封颖、虞新阳、卢冀等对我的关心、帮助，这使得我在读研期间能有一个宽松愉快的工作环境。与他们的合作，使我学到了好多东西。

深深感谢含辛茹苦养育我、为我的成长倾注了无数心血并不断鼓励和支持我上进的父母亲以及给我无私照顾、关心和指导的哥哥，他们无私的爱、深切的期望是我漫漫求学及人生征途中的强大精神支柱和动力源泉。

感谢所有帮助过我的人，他们的帮助我将铭记在心。

参考文献

- [1] 吴成柯, 戴善荣, 陆心如. 图像通信. 西安: 西安电子科技大学出版社, 1996
- [2] 钟玉琢, 王琪, 贺玉文. 基于对象的多媒体数据压缩编码国际标准—MPEG4 及其校验模型. 北京: 科学出版社, 2000
- [3] 冈萨雷斯. 数字图像处理. 北京: 电子工业出版社, 2004
- [4] 侯伯亨, 顾新. VHDL 硬件描述语言与数字逻辑电路设计. 西安: 西安电子科技大学出版社, 2000
- [5] 张亮. 数字电路设计与 Verilog HDL. 北京: 人民邮电出版社, 2000
- [6] 崔之佑, 江春, 陈丽鑫. 数字视频处理. 北京: 电子工业出版社, 1998
- [7] 陈显治. 现代通信技术. 北京: 电子工业出版社, 2001
- [8] Robert D Turney .MPEG-4 内涵解析. 电子产品世界. 2003 年 10 月. pp40-45.
- [9] 陈飞. SDRAM 控制器的 VHDL 实现. 电子产品世界. 2002 年 12 月. pp74-77
- [10] 雷国平等, MPEG 标准发展和研究综述, 计算机工程, 2003 年 7 月第 29 卷, 第 12 期. pp1-3.
- [11] Lap-Pui Chau, Nam Ling. "A real-time realization of MPEG-4 video decoder". Circuits and Systems, 1999. ISCAS '99. Proceedings of the 1999 IEEE International Symposium on , Volume: 1 , 30 May-2 June 1999 , pp:222 - 225
- [12] Hansoo Kim, In-Cheol Park. "High-Performance and Low-Power Memory-Interface Architecture for Video Processing Applications". IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL.11, NO.11, 2001
- [13] D. Lei Lee. Scrambled Storage for Parallel Memory Systems. In Proc. IEEE International Symposium on Computer Architecture, pp 232-239, May 1988.
- [14] ISO/IEC JTC1/SC29/WG11 N3342. Overview of the the mpeg-4 Standard. Mar, 2000
- [15] ISO/IEC JTC1/SC29/WG11 Document N2552. Mpeg-4 Video Verification Model V18.1
- [16] ITU-T Recommendation H.263. Video Coding for low bit-rate communication. 1996
- [17] S.A. White, "Applications of distributed Arithmetic to digital sequence processing: A tutorial review" IEEE ASSP Mag., vol.6, pp5-19, July 1989.

- [18] Chris Dick, "Minimum Multiplicative Complexity Implementation of the 2-D DCT Using Xilinx FPGAs", Proc. Of SPIE's Photonics East'98 Configurable Computing: Technology and Applications. 1998
- [19] Vasudow Bhaskaram and Konstantinous Konstantinides, "Image and Video Compression Standards, Algorithms and Architectures", pp227-235, Kluwer Academic Publishers 1995.
- [20] "The Programmable Logic Data Book 2000", Xilinx, pp3-11, Feb, 2000.
- [21] "The Role of Distributed Arithmetic in FPGA-based Signal Processing".<http://www.xilinx.com>, 2001
- [22] Kun-Min Yang, Ming- Ting Sun, et al. "A Family of VLSI Designs for the Motion Compensation Block-Matching Algorithm". IEEE Transaction on Circuits and Systems, Vol.36, NO 10, October 1989.
- [23] Sungwook Yu, Earl E. Swartzlander "DCT Implementation with Distributed Arithmetic", IEEE Transaction On Computers, Vol.50.No.9 pp985-991,2001
- [24] Sun Xiantao, Wu Chengke "Efficient Method for Hardware-based DCT/IDCT Implementation", Proc. Of SPIE on Neural Network and Distributed Processing, 2001
- [25] A.Murat Tekalp, "Digital Video Processing", Prentice Hall,1995
- [26] L.V. Agostini, I.S. Silva, et al. "Pipelined Fast 2-D DCT Architecture for JPEG Image Compression", <http://inel.ieeeexplore.ieee.org>, 2001
- [27] ISSI IS42S16100 DATASHEET NOVEMBER. <http://www.issi.com>. 2001
- [28] AMBA™ Specification(Rev 2.0). <http://www.arm.com>. 1999
- [29] 孙先涛.基于硬件实现的 H263 编码系统. 西安: 西安电子科技大学硕士研究生学位论文, 2002
- [30] 王海.H.263 视频编码系统的研究与实现.西安: 西安电子科技大学硕士研究生学位论文, 2003
- [31] 纪小明.高效视频编解码系统的研究与硬件实现.西安: 西安电子科技大学硕士研究生学位论文, 2004

作者在读期间的研究成果

- [1] 程培星, 韩伟. MPEG-4 SP 级解码器中的 SDRAM 接口设计. 西安电子科技大学 2004 年研究生学术年会. page 281-285.

作者：程培星
学位授予单位：西安电子科技大学
被引用次数：2次

参考文献(32条)

1. 参考文献
2. 吴成柯, 戴善荣, 陆心如 图像通信 1996
3. 钟玉琢, 王琪, 贺玉文 基于对象的多媒体数据压缩编码国际标准— MPEG4及其校验模型 2000
4. 冈萨雷斯, 阮秋琦 数字图像处理 2004
5. 侯伯亨, 顾新 VHDL硬件描述语言与数字逻辑电路设计 2000
6. 张亮 数字电路设计与Verilog HDL 2000
7. 崔之估, 江春, 陈丽鑫 数字视频处理 1998
8. 陈显治 现代通信技术 2001
9. Robert D. Turney MPEG4内涵解析[期刊论文]-电子产品世界 2003(19)
10. 陈飞 SDRAM控制器的VHDL实现[期刊论文]-电子产品世界 2002(2B)
11. 雷国平, 周琨, 吉吟东 MPEG标准发展和研究综述[期刊论文]-计算机工程 2003(12)
12. Lap-Pui Chau, Nam Ling A real-time realization of MPEG-4 video decoder 1999
13. Hansoo Kim, In-Cheol Park High-Performance and Low-Power Memory-Interface Architecture for Vedio Processing Applications 2001(11)
14. D Lei Lee Scrambled Storage for Parallel Memory Systems 1988
15. ISO/IEC JTC1/SC29/WG11 N3342. Overview of the the mpeg-4 Standard 2000
16. ISO/IEC JTC1/SC29/WG11. Document N2552. Mpeg-4 Video Verification Model V18.1
17. ITU-T Recommendation H.263. Video Coding for low bit-rate communication 1996
18. S A White Applications of distributed Arithmetic to digital sequence processing:A tutorial review 1989
19. Chris Dick Minimum Multiplicative Complexity Implementation of the 2-D DCT Using Xilinx FPGAs 1998
20. Vasudow Bhaskaram, Konstantinous Konstantinides Image and Video Compression Standards, Algorithms and Architectures 1995
21. The Programmable Logic Data Book 2000 2000
22. The Role of Distributed Arithmetic in FPGA-based Signal Processing 2001
23. Kun-Min Yang, Ming-Ting Sun A Family of VLSI Designs for the Motion Compensation Block-Matching Algorithm[外文期刊] 1989(10)
24. Sungwook Yu, Earl E Swartzlander DCT Implementation with Distributed Arithmetic 2001(09)
25. Sun Xiantao, Wu Chengke Efficient Method for Hardware-based DCT/IDCT Implementation 2001
26. A Murat Tekalp Digital Video Processing 1995
27. L V Agostini, I S Silva Pipelined Fast 2-D DCT Architecture for JPEG Image Compression 2001
28. ISSI IS42S 16100 DATASHEET NOVEMBER 2001
29. AMBATM Specification (Rev 2.0) 1999
30. 孙先涛 基于H.263的视频编码系统硬件设计与实现[学位论文]硕士 2002
31. 王海 H263视频编码系统的研究与实现[学位论文]硕士 2003
32. 纪小明 高效视频编码系统研究与硬件实现[学位论文]硕士 2004

引证文献(2条)

1. 王国忠 宽带无线视频网络关键技术的研究[学位论文]硕士 2006
2. 林成保 基于TMS320DM642的MPEG-4视频解码器的研究和实现[学位论文]硕士 2006

本文链接：http://d.g.wanfangdata.com.cn/Thesis_Y695835.aspx