

西安电子科技大学

学位论文独创性（或创新性）声明

秉承学校严谨的学风和优良的科学道德，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中所罗列的内容以外，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对本研究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，本人承担一切的法律责任。

本人签名： 陈佳

日期 2011.3.7

西安电子科技大学

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。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。同时本人保证，毕业后结合学位论文研究课题再撰写的文章一律署各单位为西安电子科技大学。

本人签名： 陈佳

日期 2011.3.7

导师签名： 阿伯12

日期 2011.3.7

摘 要

随着网络宽带的增加和信息处理技术的进步,人们对多媒体业务的需求越来越广泛。视频编码技术作为多媒体技术中重要的组成部分,早已经成为国内外研究和工业应用的热点之一。许多视频编码方案也在不断的被提出和应用,而 H.264 是目前最有效的编解码标准。

H.264 在大幅度提高编码效率的同时,因为复杂的运算对 H.264 的嵌入式应用带来了巨大障碍。因此如何对拥有大量复杂度编码算法的 H.264 编码器进行编码功能正确性的验证,以及验证编码器是否具有高效率的编码性能是一个重要的研究课题。

因此本论文设计并选用了某新型验证平台,采用软硬件协同的验证方法,对某 H.264/AVC 视频编码核进行层次化的验证。

本论文主要研究了 SoC 的软硬件协同设计与验证技术,构建了适合于验证该芯片的两种验证平台,即虚拟原型验证平台和 FPGA 原型验证平台,这两种验证平台优势互补,可以提高验证的质量和效率。

基于上述验证平台的搭建,本论文从主观和客观质量评定角度出发,结合 H.264/AVC 视频编码核的算法以及实现结构,对编码核做了相关的模块级和系统级验证,重点验证此编码核功能的正确性和性能的高效性。

关键词: H.264 编码协议 SoC 软硬件协同验证 H.264/AVC 视频编码核

Abstract

Along with the increase of broadband network and information processing technology progress, people needs more and more wide multimedia business. Video coding technology as multimedia technology is an important part of research at home and abroad, have already become one of the hottest and industrial applications. Many video coding schemes are proposed and used in constant, and H.264 is the most effective decoding standards.

H.264 improve the coding efficiency in substantially, but also because of complex operation for H.264, embedded application brings huge obstacles. Therefore how to validate the H.264 encoder, which encoding algorithm has large computation complexity, and if the encoder is highly efficient coding coder performance, all these are important research topics.

Therefore, this dissertation designs and choses a certain new validation platform, using hardware and software collaborative verification method, to one H.264/AVC video coding core for hierarchical validation.

This dissertation mainly studies the SoC of software and hardware co-design and verification technologies, constructs two kinds of validation platform suitable for the chip, namely the virtual prototyping validation platform and FPGA prototype validate platform, these two kinds of validation platform complementary advantages, can improve the quality and efficiency of verification.

Based on the above validation platform, from the subjective and objective quality evaluation Angle, this dissertation introduces the validation from module level and system level validation, combining h.264 / AVC video coding algorithm and the realization of the core configuration of coding core. The correctness of the coding core functions and properties of high efficiency are demonstrated finally.

Keywords: H.264 coding protocol SoC hardware and software collaborative verification H.264/AVC video coding core

目 录

第一章 绪 论.....	1
1.1 研究背景及意义.....	1
1.2 选题背景及意义.....	1
1.3 国内外研究现状.....	2
1.4 论文的研究内容及结构安排.....	3
第二章 H.264/AVC 编码芯片的简介.....	5
2.1 H.264/AVC 编码芯片的系统架构.....	5
2.2 H.264/AVC 编码芯片的片上资源.....	5
2.2.1 PowerPC440 处理器.....	5
2.2.2 PCI 主机接口	6
2.2.3 SRAM 控制器	6
2.2.4 IDE 记录接口	7
2.2.5 以太网接口.....	7
2.2.6 向量中断控制器.....	7
2.2.7 串口控制器.....	8
2.2.8 实时时钟.....	8
2.2.9 GPIO	8
2.3 H.264/AVC 视频编码核.....	9
2.3.1 分层结构.....	9
2.3.2 支持的图像帧类型.....	10
2.3.3 支持的图像帧结构.....	10
2.3.4 档次与级别.....	11
2.3.5 视频压缩编码方法.....	12
2.3.6 H.264/AVC 视频编码核主要技术.....	15
第三章 H.264/AVC 编码芯片的验证系统.....	17
3.1 验证规划.....	17
3.2 验证流程.....	17
3.3 IP 级验证	19
3.4 软硬件协同验证.....	20
3.5 虚拟原型验证平台设计.....	21
3.5.1 虚拟平台的验证环境及相关工具.....	22

3.5.2 新型平台的快速模式.....	22
3.5.3 新型平台慢速模式.....	22
3.5.4 虚拟验证平台的搭建.....	23
3.5.5 验证方法.....	23
3.6 FPGA 原型验证平台设计	24
3.6.1 FPGA 平台的验证环境及相关工具	24
3.6.2 PFGA 验证平台的搭建	24
3.6.3 验证内容.....	25
第四章 H.264/AVC 视频编码核的验证.....	27
4.1 H.264/AVC 视频编码核结构.....	27
4.1.1 算法结构.....	27
4.1.2 功能实现结构.....	28
4.2 编码核层次化验证.....	32
4.2.1 模块级验证.....	32
4.2.2 系统级验证.....	35
4.3 编码核功能验证.....	37
4.3.1 基于帧类型的验证.....	37
4.3.2 基于视频格式的验证.....	41
4.3.3 编码核功能验证方案.....	43
4.4 编码核性能验证.....	46
4.4.1 H.264/AVC 视频编码核质量评估方法.....	46
4.4.2 基于编码模式的验证.....	47
第五章 总结与展望.....	57
致 谢.....	59
参考文献.....	61
作者在读期间的科研工作及研究成果.....	63

第一章 绪论

1.1 研究背景及意义

视频信息和多媒体信息在 Internet 网络和移动网络中的处理和传输已成为了当前我国信息化中的热点技术, 视频信息具有直观、确切、高效、广泛等优势, 但是视频信息量太大, 要使视频得到有效的应用, 必须首先解决视频压缩编码问题, 其次解决压缩后视频质量保证的问题。

本论文验证了一款基于 H.264 协议的 SoC(System On a Chip)编码芯片, 其中 SoC 是指基于嵌入式核的系统芯片。本论文待验证的 SoC 编码芯片采用软硬件协同的设计和验证方法, 焦点关注系统整体的功能实现及性能, 利用 IP 核可复用技术, 在一个芯片上实现信号采集、转换、存储、处理和 I/O 等复杂的系统功能, 实现降低系统的功耗、改善系统的整体性能、减少系统开发成本等。以 SoC 为基础的信息产品已经渗透到了人们的日常生活中, SoC 是集成电路产业出现的重大技术突破, 是当今超大规模集成电路的发展趋势, 也是 21 世纪集成电路技术的主流, 它为集成电路产业提供了前所未有的广阔市场和重要的发展机遇。

1.2 选题背景及意义

采用 SoC 技术开发 H.264 编码芯片, 是因为 SoC 不仅能够缩小系统的体积、降低成本, 满足市场对小型化、低功耗设备的要求; 另一方面更为突出的是 SoC 能够通过利用 IP 复用及深亚微米技术, 缩短开发周期, 加快了产品上市时间。据统计, 验证过程占据了整个设计周期的 50% 到 80%。因此, 在验证 SoC 芯片过程中, 如何缩短验证时间、提高验证效率和质量加快芯片的上市时间也是一项艰巨的任务。

软硬件协同验证技术的产生正好改变了这种不利局面。它可以在软硬件协同设计的同时, 就通过软硬件之间交互信息来实现软硬件协同验证, 它是在流片前发现设计缺陷和错误的最后机会, 所以对于整个项目的成败而言, 验证的作用至关重要。因此, 对高效率的系统验证技术的开发和研究, 具有十分重要的意义。

同时本论文选用 H.264 编码核作为验证对象是因为 H.264 与 H.263 或 MPEG-4 相比, 在同样质量下 H.264 的数码率能降低一半左右; 或者说在同样码率下, 其信噪比明显提高。

具体讲, 首先是 H.264 具有超高的压缩率, 其压缩率为 MPEG-2 的 2 倍以上,

MPEG-4 的 1.5 至 2 倍, 在相同画质、相同容量的情况下, 采用 H.264 编码技术可比目前的 DVD 光盘多保存 2 倍以上时间的影响, 因此更节省带宽和存储空间。

其次, H.264 可在所有码率(含低码率)条件下提供较高质量的视频图像。

但是, H.264 的高性能是以实现的高复杂度为代价的。编码端计算的复杂度约是 H.263 的 3 倍, 约是 MPEG-4 的 4-5 倍; 在解码端计算的复杂度约是 H.263 的 2 倍, 约是 MPEG-4 的 2-3 倍。因此, 在高性能处理器上实现 H.264 算法, 也即 H.264 的嵌入式应用也是研究的热点^[8]。

因此, 如何对 H.264 的嵌入式应用进行验证是一个前景广阔又具有挑战性的任务。

1.3 国内外研究现状

软硬件协同验证技术是指在最终硬件没有准备好之前进行软件和硬件的协同验证, 其目的是使软件设计者在硬件完成之前接触到硬件模块, 在设计的早期就验证系统软硬件的正确性。直到近些年随着 SoC 技术的发展, 软硬件协同验证技术才得到更多的关注和重视, 并得到发展。

目前, 软硬件协同验证技术对低层次仿真的研究比较成熟, 其一般模式是硬件 RTL 描述、微处理器模型及软件调试环境的协同仿真。它在未生产出硬件之前对虚拟模型进行调试, 验证 SoC 软硬件接口功能和时序, 从而对包含硬件在内的整个系统进行功能验证, 节约了准备硬件平台时间, 可以对整个系统尽早的进行验证。由于考虑软件实际运行情况, 验证应更接近实际应用环境, 更容易发现设计中的问题, 检查软硬件协同工作的状况, 减少了设计中的盲目性, 尽早改正系统的缺陷, 改进系统的功能及性能, 避免早期设计错误, 降低设计风险。

软硬件协同设计与验证作为 SoC 的支持技术, 在理论上和技术上还在不断地发展和完善中。SoC 软硬件协同验证技术发展至今, 已有相当丰富的技术积累, 但是随着 SoC 设计技术的日趋成熟与功能的不断完善与强大, 复杂度的提高, 对 SoC 协同验证技术的发展也有了越来越高的要求^[3]。

基于 H.264 嵌入式的快速发展, 对 H.264 编码器的验证也进入到了白热化的状态。

美国高等电视系统会议和日本无线电工业和事务协会都准备把 H.264 作为地面便携式数字电视广播的编码方式。欧洲数字电视广播标准化团体也正在将 H.264 作为数字电视的一种编码方式来采用。

国内市场上很多 H.264 的芯片产品很多主要应用于基线层面(Base Line), 见表 1.1。基线层面的主要技术为图像只含有 I 画面, P 画面, 系统内有环路滤波, 1/4 帧间预测, 4:2:0 YUV 格式输入, 基于 VLC 的纠错编码, 弹性宏块指令等。而本文

所要验证的H.264/AVC编码芯片则致力于主要层面（Main Profile），主要层面是在基线层面基础上加入了CABAC运算编码技术和基于双向预测的B画面，滤波（接口）等技术。因此，对本芯片的验证很具有挑战性。

表 1.1 国内外相关成品

国内/外	国内			国外	
公司	华为海思	上海杰得微	成都数视微	FUJITSU	TI
处理器	ARM	ARM	处理器接口	ARM	ARB+DSP
码流标准	Base Line	Base Line	Base Line	High Profile	
分辨率	640x480	640x480	1280x720	1920x1088	
码率	最大 8Mbps		64Kbps~16Mbps		

1.4 论文的研究内容及结构安排

本论文研究了H.264标准协议的算法实现，分析了软硬件协同验证技术，解析了编码核软件部分在PowerPC440处理器中的实现，最后重点分析了H.264/AVC视频编码核的验证。论文的结构安排如下：

第一章分析了H.264以及SoC软硬件协同设计与验证的研究意义和发展现状，并指出了论文的主要研究内容及结构安排。

第二章介绍了H.264/AVC编码芯片，分析了芯片的系统架构及片上资源，并重点解析了编码核部分。

第三章介绍了H.264/AVC编码芯片的验证系统，分析了验证平台的设计及搭建，包括虚拟原型验证平台和FPGA原型验证平台。

第四章介绍H.264/AVC视频编码核的结构、软件实现以及验证方法，分析了对编码核的模块级和系统级的验证，以及重点从功能和性能两个方面展开对编码核详细的验证。

第五章对整篇论文的研究工作做出总结，并对有待进一步研究的问题和后续工作进行展望。

第二章 H.264/AVC编码芯片的简介

2.1 H.264/AVC编码芯片的系统架构

本论文待验证的H.264/AVC编码芯片是基于H.264协议标准，集成了PowerPC440处理器、PCI主机接口、片上SRAM、向量中断控制器、外部总线控制器、I2C总线控制器、串口控制器、实时时钟、通用输入输出接口、DMA控制器、以太网终端、IDE记录接口和H.264/AVC视频编码核等资源的片上系统，是一款可以实现实时高清视频编码的高集成度系统芯片。该芯片可实现H.264/AVC（又称MPEG4-Part10）标准中所定义的满足Main Profile的高清视频编码、MPEG2-TS流打包、以太网协议栈解析、以及文件系统管理等功能，其系统结构如图2.1所示。

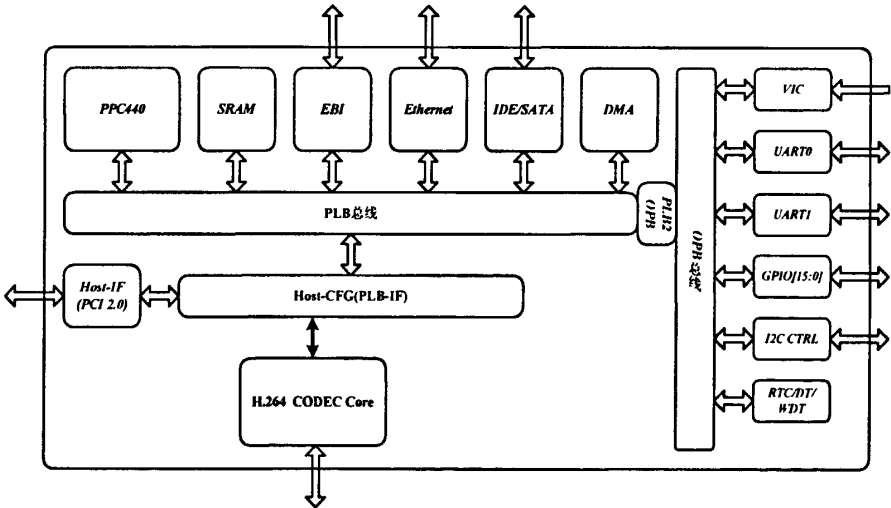


图 2.1 H.264/AVC 编码芯片的系统结构图

2.2 H.264/AVC编码芯片的片上资源

2.2.1 PowerPC440 处理器

IBM PowerPC 440 CPU核是IBM系列中一个32位RISC PowerPC嵌入式处理器核，运算速度达到667MHz，完美的实现了IBM先进的90纳米铜CMOS技术。440内核充分发挥了Book E增强型POWER架构的高可扩展性和灵活性，并专为嵌入式应用进行了优化。

可授权的嵌入式内核集成了超标量的7级管线，支持在每个时钟周期内处理两

组指令缓存和数据缓存, 一个JTAG端口, 跟踪式FIFO, 多个计时器和一个内存管理单元(MMU)。其通过在辅助处理器单元(APU)接口加装浮点单元, 满足需要浮点运算硬件支持的应用。此外, 高达256KB SRAM的L2缓存接口可提供额外的片内存储容量。

PowerPC440内核的突出性能、功耗规格以及设计特性将它推向了高性能片上系统(SOC)设计的领导地位。PowerPC440核将性能和无与伦比的微处理器的特性与灵活性, 低功耗以及嵌入式CPU核的模块性有效地结合在一起, 主要被设计用来在最大性能和大量外设集成中的应用。

PowerPC片上系统主要包含一个中央处理器PowerPC440、两条总线PLB和OPB。PLB总线用于连接PowerPC440和BRAM(Block RAM)等高速设备, BRAM相当于本片上系统的内存, 由FPGA内逻辑搭建而成。若应用程序很大, BRAM不够使用, 则可在PLB总线上挂接存储控制器和外部RAM, 而OPB总线则将挂接低速外设。通过总线桥连接到PLB总线, 实现与处理器的通信。处理器与外设的通信都是通过对外设地址的读写操作完成的, 每个外设都分配到一段独一无二的地址空间, 以实现处理器对外设的识别和访问。

PowerPC440核的目标市场部分包括: 数字照相机、视频游戏、机顶盒和英特网应用等消费应用; 激光打印机、thin-client系统等办公自动化产品; RAID控制器、路由器、ATM交换机、网络卡等存储及网络产品。

用PowerPC440核做嵌入式设计, 其开发工具包括C/C++编译器, 调试器, 总线功能模型, 软硬件协同仿真环境以及实时操作系统。

2.2.2 PCI 主机接口

PCI总线是当今流行的高速、高性能局部总线, 是一种不依附于某个具体处理器的局部总线。从结构上看, PCI是在CPU和原来的系统总线之间插入的一级总线。PCI主机接口主要实现了PCI主机和PowerPC440到H.264/AVC编码芯片内部寄存器的访问及数据通信。

编码芯片的PCI接口完全支持PCIv2.2协议, 32位数据/地址总线, 总线频率支持33MHz和66MH, 支持突发和单拍模式访问。

2.2.3 SRAM 控制器

SRAM控制器是视频核与外部存储器SRAM的接口, 访问数据宽度可配置, 当SRAM数据线扩展使能寄存器的使能位被使能时, 数据宽度被配置为128位, 该位未使能时, 数据宽度为64位, 该位的复位值为0, 默认为64位数据线。最大可以支

持256Mbit的访问空间。

2.2.4 IDE 记录接口

IDE接口控制器完成对外部IDE设备的读写操作。根据寄存器配置,将编码产生的TS流保存到外部的IDE设备。支持FAT32文件系统。

2.2.5 以太网接口

以太网接口模块支持10Mbps/100Mbps半双工和全双工操作,一端与PLB总线相连,一端与PHY相连。主要功能是完成以太网帧的发送和接收。其结合片上处理器和以太网协议栈嵌入式应用软件,完成发送数据帧的调度、接收数据帧的帧头分离、接收数据帧的数据部分校验和计算、UDP协议类型的提取,为片上处理器进行协议栈处理提供必要信息。

2.2.6 向量中断控制器

PowerPC440 支持两种中断,非关键中断 IRQ 与关键中断 FIQ。向量中断控制器(VIC)提供 H.264/AVC 编码芯片内部中断源与外部中断源到处理器中断的路由和控制。

VIC 共支持 32 路中断源,每个中断源在寄存器中的对应位与中断源的序号相同,如中断源 0 对应寄存器中的位 0,依此类推。

每路中断源的中断方式及中断极性都可编程,且可受软件控制产生软中断。

FIQ 和 IRQ 具有不同的中断路由路径,即 VIC 可以同时产生 FIQ 中断和 IRQ 中断,但如果同时有多个 IRQ 中断源有效,VIC 会在产生 IRQ 中断时依据优先级排序的结果处理中断。向量中断 0~15 的优先级依次降低,非向量 IRQ 中断优先级最低。FIQ 中断拥有最高优先级,可以打断任何中断。大部分情况下系统只有一个 FIQ 中断,这样当 FIQ 中断到达 CPU 之后,CPU 可直接跳转到对应的 FIQ 中断服务程序去执行服务程序,效率高。

VIC 最多支持 16 路向量中断,所有向量中断的优先级都比非向量 IRQ 中断的优先级高,向量中断有优先级(0~15),当高优先级中断在服务时,低优先级中断被屏蔽,只有当高优先级中断服务完,中断控制器才会将低优先级中断报给 CPU,当低优先级中断在服务时,来了高优先级中断,高优先级向量中断可打断低优先级中断,直到高优先级中断服务完再继续执行低优先级中断,实现嵌套;所有的向量中断都可以打断非向量 IRQ 中断,每一个向量中断具有唯一的向量地址。

所有的非向量 IRQ 中断具有一个入口地址和一样的优先级, 当一个非向量 IRQ 中断在服务时, 会屏蔽其他的非向量 IRQ 中断, 但如果有一次 CPU 响应非向量 IRQ 中断时同时有多个非向量 IRQ 申请, 则此时需要软件来判断服务顺序。

VIC 支持特权模式, 提供对 VIC 寄存器的访问保护。

2.2.7 串口控制器

串口控制器 (UART) 包含一个标准的通用异步收/发器接口, 软件编程和目前流行的串口器件。UART 对来自调制解调器或其它串口器件的数据进行串并转换, 同时对 CPU 发送给串口器件的数据进行并串转换。在数据发送过程中, 并行数据被写入 UART 的发送保持寄存器或者发送 FIFO (在 FIFO 模式下), 该数据随后被按照编程设定的格式串行发送; 在数据接收过程中, UART 将接收到的串行数据转换成并行形式存入接收缓冲寄存器或者接收 FIFO (在 FIFO 模式下), 以供处理器读取。CPU 可以随时查询 UART 的状态信息。UART 主要用于以微处理器为基础的系统, 系统应用框图如图 2.2 所示。



图 2.2 系统应用框图

2.2.8 实时时钟

实时时钟能够提供百分之一秒、十分之一秒、秒、分、小时、星期、日期、月份和年份的 BCD 数据, 当实际数据与闹钟 (报警) 数据相同时, 可以产生中断; 支持 ms 级节拍中断, 支持闰月功能, 为实时操作系统提供时间基准。

2.2.9 GPIO

通用输入输出控制器 (GPIO Controller) 模块是 H.264/AVC 编码芯片中非常重要的一种 I/O 接口, 具有灵活、可配置性好、硬件代价小等优点, 通用 I/O 接口可以控制 16 个 IO 引出端为输入或输出, 图 2.3 为其在系统中应用的结构。

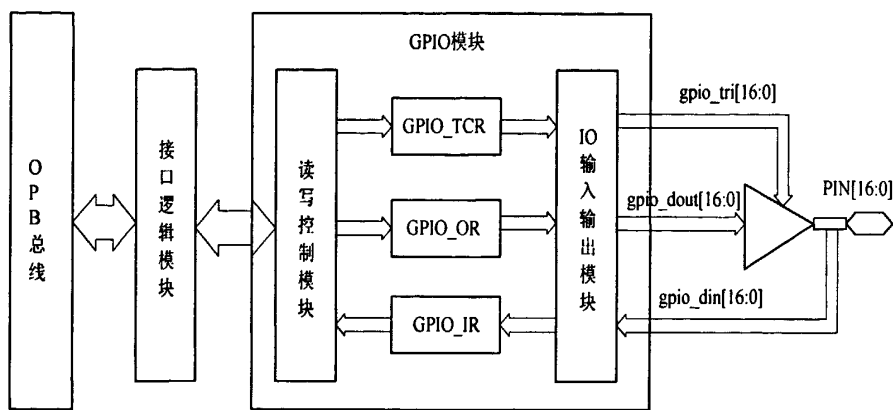


图 2.3 GPIO 逻辑结构图

2.3 H.264/AVC视频编码核

H.264/AVC视频编码核作为最重要的模块挂接在整个系统上，完成编码芯片的视频编码功能。

2.3.1 分层结构

H.264的功能分为两层：视频编码层（VCL：Video Codig Layer）负责高效的视频内容表示，网络提取层（NAL：Network Abstraction Layer）以网络所要求的恰当的方式对数据进行打包和传送。VCL数据即编码处理的输出，它表示被压缩后的视频数据序列。在VCL数据传输或存储之前，这些编码的VCL数据，先被映射或封装进NAL单元中。较高的视频压缩比和良好的网络亲和性是制定H.264的主要目标。只有将网络适应层和视频编码层紧密结合起来，才能获得最佳的传输性能。

每个NAL单元包括一个原始字节序列负荷（RBSP）、一组对应于视频编码数据的NAL头信息。NAL单元序列的结构如图2.4所示^[2,13]。

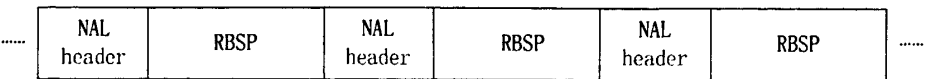


图 2.4 NAL 单元序列

网络层提取层（NAL）负责使用下层网络的分段格式来封装数据，包括组帧、逻辑信道的信令或发序列结束信号等。NAL以NALU（NAL Unit）为单元来支持编码数据在基于分组交换技术网络中的传输，这有利于数据经打包后在网络中传输。对于面向比特流和面向数据包的传输，NALU采用统一的数据格式，每个NALU包

含单个字节的包头和多个字节的数据，包头信息包含存储标志和类型标志。存储标志用于只是当前数据是否属于被参考的帧，从而便于服务器根据网络的拥塞情况进行丢弃；类型标志用于指示图像数据的类型。例如，NAL支持视频在电路交换信道上的传输格式，支持视频在Internet上利用RTP/IP传输的格式。NAL的任务是提供适当的映射方法将头部信息和数据映射到传输协议上。这样，在分组交换传输中可以消除组帧和重同步开销。

2.3.2 支持的图像帧类型

H.264/AVC 视频编码核运动估计涉及到三种帧类型：I 帧、P 帧和 B 帧。只包含帧内编码的帧称为 I 帧，对 I 帧或 P 帧进行前向预测得到的帧称为 P 帧，对 I 帧或 P 帧进行前后双向预测得到的帧称为 B 帧。

2.3.3 支持的图像帧结构

H.264/AVC 视频编码核支持多种分辨率的视频图像格式，这些视频数据一般是以 YUV 格式存储的，将其亮度信号和色度信号分开存储。编码核支持的视频源格式为 4:2:0，同时支持逐行扫描和隔行扫描的视频序列。对于视频源格式 4:2:0，如图 2.5 所示，亮度信号按照图像格式本身的分辨率进行采样，色度块在 x 轴和 y 轴的采样率均为亮度信号采样率的一半^[1,19]。

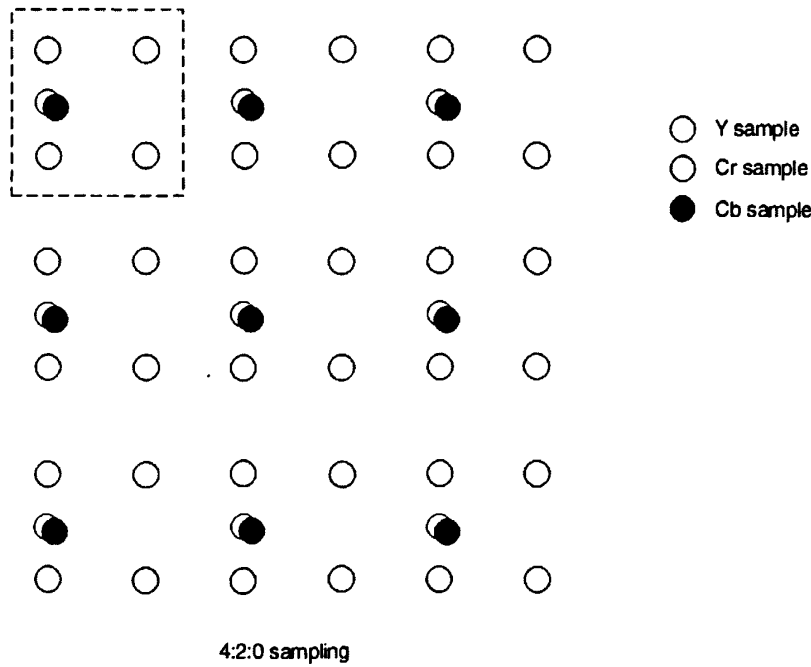


图 2.5 4:2:0 的 yuv 视频源格式

视频输入的图像都被分为 16*16 大小的宏块，每个宏块包含 4 个 8*8 的 Y 子块（16*16）和 2 个 8*8 的 UV 子块，如图 2.6 所示。若干个宏块可以组成宏块组，称为片（slice）。设片的目的是为了限制误码的传输与扩散，所以应该编码片间保持独立。

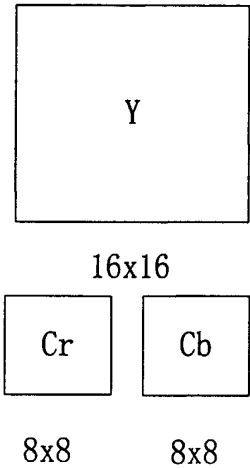


图 2.6 YUV 子块

2.3.4 档次与级别

档次（Profile）规定了编码核所采用的编码工具与算法组合，每一个档次都支持一组特定的编码功能。H.264 标准共定义了三个档次，即基本档次（Baseline profile）、主要档次（Main profile）、扩展档次（Extended profile），其中每个档次支持一组特定的编码功能，并支持一类特定的应用。

基本档次利用 I 帧和 P 帧支持帧内和帧间编码，支持利用基于上下文的自适应的算术编码（CAVLC）。主要用于“视频会话”，如会议电视，可视电话，远程医疗、远程教学等。

主要档次支持隔行视频，采用 B 帧的帧间编码和采用加权预测的帧内编码，支持利用基于上下文的自适应的变长编码进行的熵编码（CABAC）。主要用于消费电子应用，如数字电视广播，数字视频存储等。

扩展档次支持码流之间有效的切换（SP 和 SI 片）、改进误码性能（数据分割），但不支持隔行视频和 CABAC。主要用于网络的视频流，如视频点播。

图 2.7 为 H.264 各个档次具有的功能，扩展档次包括了基本档次的所有功能，而不能包括主要档次的。基本档次的复杂度最小，主要档次更强调压缩编码效率，扩展档次则将基本档次的健壮性与高效率编码、网络传输的健壮性结合起来^[1,20]。

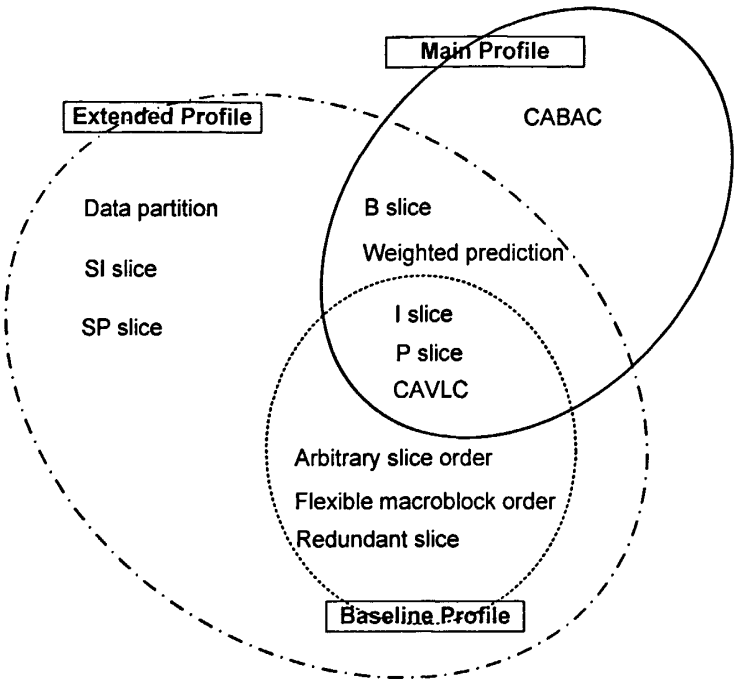


图 2.7 H.264 的档次

级别（Level）定义了视频编码核的性能，对每一个级别的取值范围都作了限制。这些限制主要根据图像分辨率与帧率而定。

2.3.5 视频压缩编码方法

视频编码的主要目的是去除相关性，在满足一定质量（信噪比的要求或主观评价得分）的条件下，以较少比特数表示视频图像或视频图像中所包含信息的技术。在信息论术语中，称为信源编码。广泛应用于视频图像数据压缩、视频图像传输和特征提取等方面。其基本原理为：减少视频图像中多余信息，以缩短传输无用信息时间，利用图像固有统计特性进行编码；利用人们视觉心理特性进行编码。常用方法有预测编码、变换编码和混合编码等。H.264/AVC 视频编码核采用的是预测和变换的混合编码法。

(1)预测编码

预测法是最简单和实用的视频压缩编码方法，这时压缩编码后传输的并不是像素本身的取样幅值，而是该取样的预测值和实际值之差。大量数据统计表明，同一副图像的邻近像素之间有着相关性，或者说像素值相似。邻近像素之间发生突变或“很不相似”概率很小。而且同帧图像中邻近行之间对应位置的像素之间也有较强的相关性。进一步说明，由于图像的误差信号方差相对图像信号本身方差较小，

其量化器的动态范围可以缩小,相应的量化分层数目就可以减小,每个像素的编码比特数也显著下降,而且不致视频质量明显降低,达到视频压缩的目的。

(2)变换编码

绝大多数图像有一个共同的特征:平坦区域和内容缓慢变化区域占据一副图像的大部分,而细节区域和内容突变区域则占小部分。可以说图像中直流和低频区占大部分,高频区占小部分。这样,空间域的图像变换到频域或所谓的变换域,会产生的一些相关性很小的变换系数,并可对其进行压缩编码,即所谓的变换编码。变换中有一类叫做正交变换,可用做图像编码。自1968年利用快速傅里叶变换(FFT)进行图像编码以来,出现了多种正交变换编码方法,如K-L变换、离散余弦变换(DCT)等等。其中,编码性能以K-L变换最理想,但缺乏快速算法,且变换矩阵随图像而异,不同图像需计算不同的变换矩阵,因而只用来参考比较。DCT编码性能最接近于K-L变换,具有快速算法,广泛应用于图像编码。

变换编码实现比较复杂,预测编码的实现相对容易,但预测编码的误差会扩散。以一行为例,由于后面像素以前面像素为参考,前面像素的预测误差会逐步向后面像素扩散。而在二维预测时,误差会扩散至后面几行,形成区域误码。相比而言,变换编码则不会误码扩散,其影响只限制在一个块内,而且反变换后误码会均匀分散到块内各个像素上,对视觉无甚影响。

两者各有优缺,在实现中往往采用混合编码方法,即对图像先进性带有运动补偿的帧间预测编码,再对预测后残差信号进行DCT变换。这种混合编码方法已成为许多视频压缩编码国际标准的基本框架。

(3)熵编码

如果要求编码过程中不丢失信息量,即要求保存信息熵,这种信息保持编码叫熵编码。熵编码是无损压缩编码方法,它生成的码流可以经解码无失真地恢复出原数据。

熵的大小与信源的概率模型有着密切的关系,各个符号出现的概率不同,信源的熵也不同。当信源中各事件是等概率分布时,熵具有极大值。信源的熵与其可能达到的最大值之间的差值反映了该信源所含有的冗余度。信源的冗余度越小,即每个符号所独立携带的信息量越大,那么传送相同的信息量所需要的序列长度越短,符号位越少。因此,数据压缩的一个基本的途径是去除信源的符号之间的相关性,尽可能地使序列成为无记忆的,即前一符号的出现不影响以后任何一个符号出现的概率。

在视频编码中,熵编码把一系列用来表示视频序列的元素符号转变为一个用来传输或者存储的压缩码流。输入的符号可能包括量化的变换系数、运动向量、标记(在序列中用来表示重同步位的点)、头(宏块头、图像头、序列头等)以及附加信息。

对于不同的句法元素采用了不同的编码方式,包括:expGolomb、CAVLC、CABAC。其中 expGolomb 和 CAVLC 分别编码宏块层预测量化信息和残差数据,CABAC 与前者是并行关系,完成对宏块层所有数据的编码,在程序中通过配置可以二者选其一使用。

算术编码(CABAC)的思想是用 0 到 1 的区间上的一个数来表示一个字符输入流,它的本质是为整个输入流分配一个码字,而不是给输入流中的每个字符分别指定码字。算术编码是用区间递进的方法来为输入流寻找这个码字,它从于第一个符号确定的初始区间(0 到 1)开始,逐个字符地读入输入流,在每一个新的字符出现后递归地划分当前区间,划分的根据是各个字符的概率,将当前区间按照各个字符的概率划分成若干子区间,将当前字符对应的子 2 区间取出,作为处理下一个字符时的当前区间。到处理完最后一个字符后,得到了最终区间,在最终区间中任意挑选一个数作为输出。解码器按照和编码相同的方法和步骤工作,不同的是作为逆过程,解码器每划分一个子区间就得到输入流中的一个字符。

在 H.264 的 CAVLC(基于上下文自适应的可变长编码)中,通过根据已编码句法元素的情况动态调整编码中使用的码表,取得了极高的压缩比。CAVLC 用于亮度和色度残差数据的编码。残差经过变换量化后的数据表现出如下特性:4*4 块数据经过预测、变换、量化后,非零系数主要集中在低频部分,而高频系数大部分是零;量化后的数据经过 zig-zag 扫描,DC 系数附近的非零系数值较大,而高频位置上的非零系数值大部分是+1 和-1;相邻的 4*4 块的非零系数的数目是相关的。CAVLC 充分利用残差经过整数变换、量化后数据的特性进行压缩,进一步减少数据中的冗余信息,为 H.264 卓越的编码效率奠定了基础。

CABAC 的优点是可以简单地通过更新符号概率表来实现对信源统计特性的自适应。通过对不同上下文用不同的概率表也可以容易地实现条件编码。对于哈夫曼编码,则不得不基于更新的概率表重新设计码字,或对不同的上下文设计多个码表。由于较高的编码效率和易于自适应,只要所涉及的计算量是能接受的,无疑算术编码比哈夫曼编码是一种更好的选择。

CABAC 中内建了由大量实验统计而得到的概率模型。在编码过程中,CABAC 根据当前所要编码的内容以及先前已编码好内容,动态地选择概率模型来进行编码,并实时更新相对应的概率模型。并且,CABAC 在计算量和编码速度上进行了优化,用了量化查表、移位、逻辑运算等方法代替复杂的概率估计和乘法运算。在实际应用中,CABAC 与其它主流的熵编码方式相比有更高的编码效率,用一组质量在 28~40dB 的视频图像做测试,应用 CABAC 可使比特率进一步提高 9%~14%。

2.3.6 H.264/AVC 视频编码核主要技术

H.264/AVC 视频编码核加大了预测部分的比重,通过减少预测误差而提高编码效率。其有别与基于其他标准的视频编码核的主要技术有:

- (1)采用 4×4 像素块的整数变换,反变换过程中没有匹配错误问题;
- (2)运动补偿块大小可采用可变形式,从 16×16 到 4×4 中选择,采用不同块大小的运动向量预测可比单独使用 16×16 块预测提高大于 15% 的编码效率;
- (3)运动向量精度可达 $1/4$ 像素;
- (4)采用多参考帧进行帧间编码,这比单独参考帧方法节省 5%~10% 的码率,并且有利于码流的错误恢复;
- (5)为消除块效应,采用基于 4×4 块边界的去块效应滤波器,提供了图像的主观质量;
- (6)采用CAVLC或CABAC编码,其中后者可节省约10%的码率。

第三章 H.264/AVC编码芯片的验证系统

本论文除了实现 H.264 编码芯片的所有功能，同时也非常关注如何评估、验证和集成多个现有的软硬件模块，以满足应用需求。H.264 编码芯片的设计方法以组装为基础，更深入系统级设计和软硬件并行设计，并在设计流程的所有层次上进行验证。本文在验证过程中，不仅保证单个模块的功能得到完全验证，同时还从系统的角度，在整个芯片的水平上做软硬件协同验证，保证 H.264/AVC 编码芯片的功能得到了充分验证。

3.1 验证规划

验证规划是验证工作的说明书，也是最终验证报告的一部分，其定义如何来验证一个设计。一个完整的验证规划主要包括对模块和顶层的测试策略，组成 Testbench 各个组件的定义和规范，验证工具和流程，仿真环境的定义和搭建，关键的验证点，以及验证工具结束的标准。

系统级行为模型创建之后就需要建立测试平台和测试向量对模型进行验证。在创建测试平台的同时，也要开发设计对应的系统程序并对行为模型进行测试，而不是等到真正程序开发出来之后再进行测试。这样，当 RTL 代码产生之后，会有大量的测试向量、软件可用来进行测试。

一个高质量的验证规划使得验证工程师可以更早地开发标准测试程序环境。这种并行的开发验证环境，能尽早给验证团队一个明确的目标，也是保证验证可重用（re-used）的关键。为了得到一个高质量的验证规划，验证工程师要正确和充分地理解设计需求和规范，要与设计工程师及时地交互，这样才能保证验证计划的易读、易用和可重用。因此可以说，一个好的验证规划可以有效提高验证效率，缩短开发周期，在 SoC 开发中有着重要的意义。

3.2 验证流程

在 H.264/AVC 编码芯片的验证工作中，为了可以完整全面可靠的验证各个功能模块，验证工作贯穿整个设计周期，从最初的 IP 设计开始到芯片设计完成。在整个过程中验证工作大致分为系统集成前的 IP 级验证，系统集成后的模块级验证，系统集成后的系统级验证，芯片 FPGA 原型验证以及系统应用验证五个阶段。

第一阶段：系统集成前的 IP 级验证，包括片上总线接口的功能验证，各 IP 和

模块的所有功能点在这一层得到充分验证。对 IP 和模块的验证主要使用 HDL 语言, 开发 testbench、testcase 和 bfm 给设计施加激励并观察其响应来进行。

第二阶段: 系统集成后的模块级验证, 这一阶段主要验证的对象是各模块的功能点, 但是与集成前验证不同的是, 这个阶段的验证环境是集成后的系统环境, 采用软硬件协同验证方法。测试输入是通过嵌入式处理器执行测试程序来完成的, 测试结果的收集也是通过处理器来输出到外设或其功能仿真模型的。

第三阶段: 系统集成后的系统级验证, 这一阶段验证的重点不再是各个模块的某个功能点是否正确, 而在于通过小系统的运行来验证 H.264/AVC 编码芯片的系统级特性, 如互联、流控制及模块间的互操作性等是否正常。这一阶段的内容主要包括系统各模块之间互操作的功能验证及基于操作系统的验证。

第四阶段: 芯片 FPGA 原型验证, 这阶段的验证需要开发芯片 FPGA 原型验证板, 主要由 FPGA 芯片和一些必要的外部设备构成。这一阶段的验证工作相对比较重要, 整个系统的运行非常接近于芯片的实际工作环境, 验证工作的重点集中在系统级特性上面, 如外部设备之间的访问是否正常, 系统性能是否达到设计指标, 应用软件是否能够正确运行等。

第五阶段: 系统应用验证, 主要是指将整个系统置于真实的工作环境, 开发真实的应用程序, 按照将来系统应用中的实际需求, 对整个系统进行验证。

以上这五个阶段在一定程度上采取并行推进的方式展开。第一阶段 IP 验证是伴随着 IP 设计开始的, 由 IP 设计人员根据该 IP 设计为每个功能编写详细的 IP 验证规范, 提交给相应的 IP 验证人员对其设计进行验证。系统集成阶段开始于关键 IP 设计验证完成并系统总线调试通过之后, 一旦开始集成第二阶段的模块级验证工作就可以开始着手进行。在这一阶段的系统集成人员从 IP 验证规范中提取适合于当前系统的验证项目形成详细的模块级验证规范提交给软件人员开始该阶段的验证工作。随着 IP 级设计验证工作的不断完成, 系统也日益完善, 这时可以根据系统集成情况开始部分简单的系统功能验证, 当系统所有 IP 均设计完成时系统集成工作也接近尾声。与此同时第三阶段的系统级验证工作也即将完成。这样的验证流程使得设计和验证工作基本同步进行在最大程度上缩短了项目周期, 并且提高验证效率并保证验证的全面性。

本文文所设计的验证平台是基于 PowerPC440 处理器的某新型验证系统, 主要用于 H.264/AVC 编码芯片的功能验证。

功能验证是验证中最复杂、工作量最大的部分。功能验证有 3 个关键点, 分别是语言、平台和方法。硬件验证语言最通用的是 Verilog HDL, 软件验证语言是标准 C 语言。本文所设计的新型验证平台, 可以自动生成 Testbench 环境, 提供验证的可重用性, 从而有效缩短验证时间。

图 3.1 就是 H.264/AVC 编码芯片设计及验证的流程图。

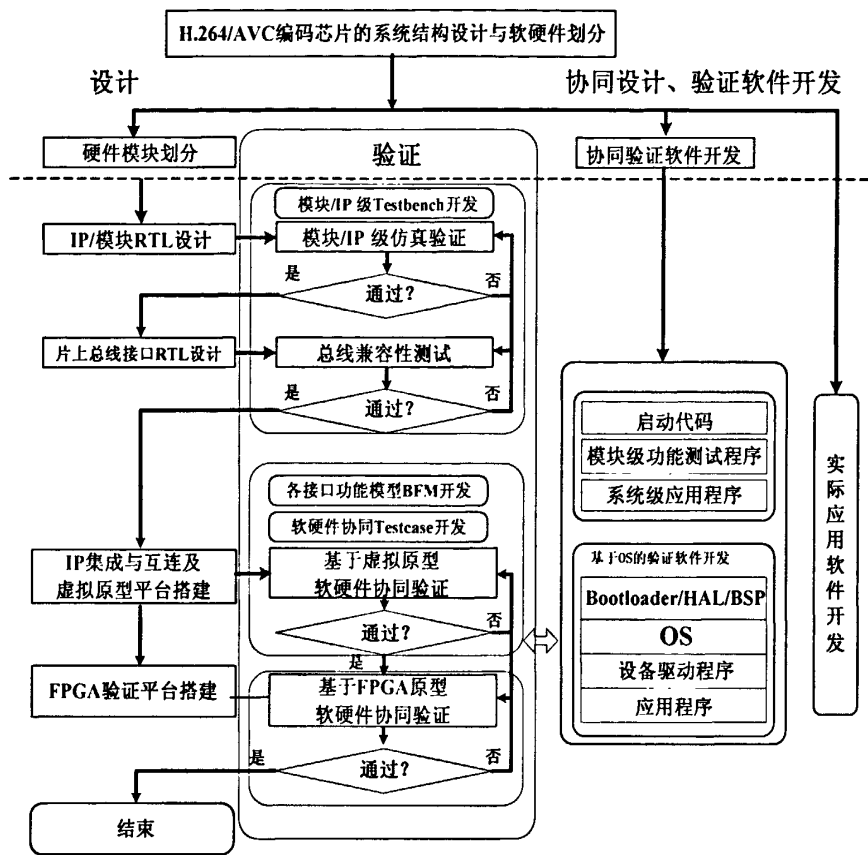


图 3.1 H.264 编码芯片系统设计验证流程图

3.3 IP级验证

IP 核级验证是系统级验证的先决条件,验证 IP 的目的在于保证 IP 功能和时序的正确性。IP 核的验证由 3 个主要阶段构成:子模块的验证,IP 核的验证和建立原型。

子模块验证的目的就是要达到足够高的功能覆盖率,如果子模块本身有一些功能错误,则整个 IP 的验证将变得非常困难和耗时。对于这些难验证的子模块,需要先用简单的测试平台验证基本功能,同时详细检查子模块的接口模块,最后在 IP 核级验证时对子模块进行全面详细的验证。

本文所述的 IP 级验证在各 IP 模块代码设计完成后,首先对各模块进行独立的功能验证。根据对 IP 功能的理解和可能的各种应用环境,编写 testbench、testcase 以及 bfm 功能模型来产生测试激励,模拟各种应用环境来验证各 IP 模块的功能,确保各 IP 模块功能的正确性。在各 IP 模块的测试验证过程中还包括对代码风格检查,尤其是对跨时钟域检查。该测试验证环境是所有验证里最灵活和快速的,可以通过编写代码模拟各种外部情况激励。不过该测试的模型和代码都是理想情况

下的，而且主机接口操作时序也是按照个人理解编写的，因此具有一定的主观性。

IP 核充分验证之后，需要进行接口验证。IP 核间的接口一般都有规则结构，如地址总线 and 数据总线等。现在 SoC 的 IP 核之间一般通过片上总线（如 PLB/OPB 等）互连。

3.4 软硬件协同验证

SoC 软硬件协同验证是一种在硬件设计确认制造之前，验证软件能在设计硬件上正确运行的过程。硬件在进行设计验证的同时，可以及时与软件之间进行数据交换。软件验证通过编译器、调试器和仿真器来建立系统中处理器的软件仿真环境，它能够运行汇编后指令，并产生指令运行结果。通过对运行结果的分析，就可以看出应用程序是否正确运行，并通过调试器对程序进行调试。

硬件方面，将软件调试验证正确的应用程序作为测试向量加载到硬件测试文件（testbench）中，进行硬件仿真，结果正确后依次进行综合、布局布线，检查布局布线后时序和逻辑是否正确，最终采用硬件加速器（FPGA）完成整个验证过程。

协同验证过程的基本框架如图 3.2 所示。

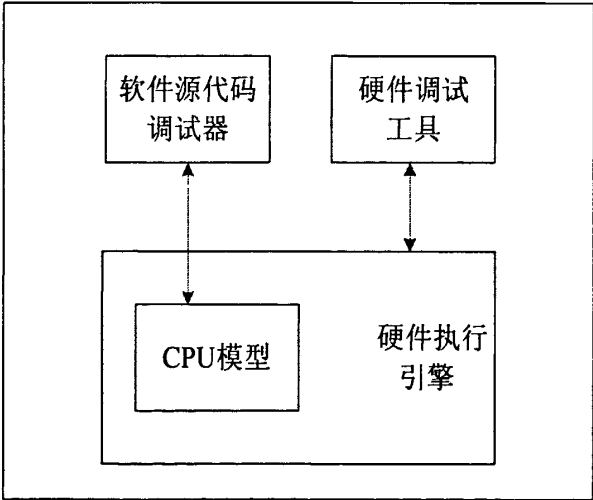


图 3.2 软硬件协同验证过程

软硬件协同验证不同于传统验证中软硬件模块分开验证，即可以避免两方面验证不在同一个环境下进行。首先，不会因为软硬件模块不在同一个验证环境中产生不一致性；同时，不需要额外增加硬件模拟软件环境和软件模拟硬件环境的仿真模型；另外，因为仿真模型有别于实际的代码，可以避免在最后整合时产生不可预期的错误，而这种错误会经常导致整个设计的反复。

因此，采用软硬件协同验证可以使设计者在设计的早期发现系统、软硬件划分、

软硬件之间的接口以及设计中的其他问题，在硬件原型构造之前，就能在一个纯“软”的环境中对系统进行软硬件集成测试。如果发现硬件有问题，可以通过改变硬件设计加以解决，而不是在软件中解决有关硬件的问题，提高了一次流片成功的可能性。同时因为协同验证是在硬件尚未流片时进行的，采用软硬件协同验证，可以在较早阶段进行软硬件集成与整合，缩短上市时间。

3.5 虚拟原型验证平台设计

H.264/AVC 编码芯片的功能验证平台有虚拟原型验证平台和FPGA 原型验证平台两种，虚拟原型验证平台优点是可以仿真模拟出芯片的实际工作条件，仿真出被测设计在各种实际激励下的响应，方便除错，但速度较慢；FPGA 的优点是速度快，可以大量运行和调试软件，但可观测性较差；两者结合可以得到很好的效果与效率的平衡。

虚拟原型验证平台通过开发各种验证组件并同 RTL 代码集成到一起，来模拟 H.264/AVC 编码芯片实际的硬件工作情况。同时在软件开发环境中开发相应的测试程序，将编译好的软件的二进制码加载到系统原型的逻辑存储器中，从而使得处理器核可以从逻辑存储器中读取指令并执行，达到协同验证的目的。利用这种方法，系统软件和硬件都在与实际相近的状态下运行，从而保证验证的准确性和可靠性^[18]。图 3.3 是虚拟原型验证平台。

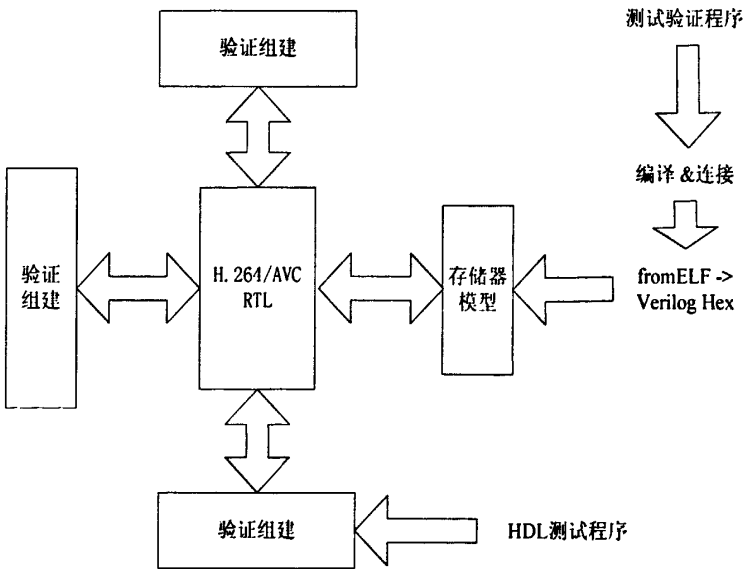


图 3.3 虚拟原型验证平台示意图

虚拟原型验证平台在应用中具有以下优点：可以多个 IP 核协同验证，并行控制各个模块；建立了一个统一的可配置的系统测试环境，提高了验证代码的可重

用性；验证平台可以在门级仿真和后仿真及直至不同层次混合仿真时复用。

3.5.1 虚拟平台的验证环境及相关工具

在 H.264/AVC 编码芯片验证中，要快速高效的验证编码芯片，并提高动态仿真验证的自动化，加快 H.264/AVC 编码芯片的开发，就迫切需要一个软硬件协同验证的验证系统。

本文所设计的验证系统是基于 PowerPC440 处理器的新型验证平台。在虚拟平台下使用的验证工具和验证环境如表 3.1 所示：

表 3.1 虚拟验证平台测试工具、环境

验证环境	虚拟原型验证平台、LINUX
仿真验证工具	QuestaSim6.4
程序开发工具	GUN
硬件描述语言	VHDL, Verilog HDL
软件语言	C, PowerPC ASSEMBLE language

3.5.2 新型平台的快速模式

新型平台快速模式的设计是由一个虚拟的处理器总线功能模型来代替处理器执行新型平台的软件代码。快速模式的典型验证环境如图 3.4 所示。

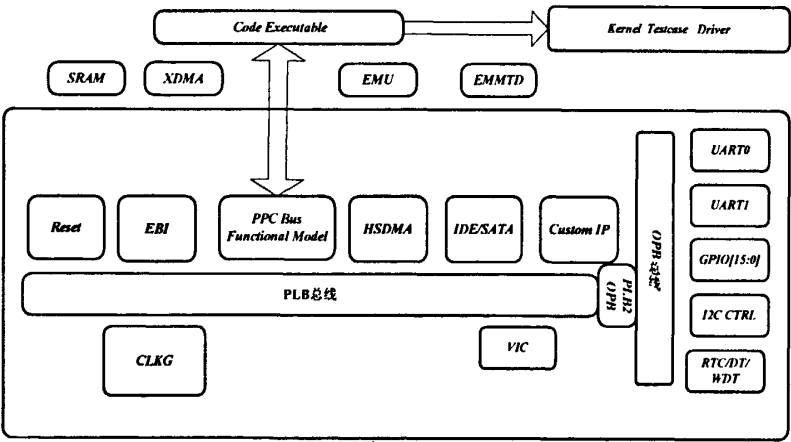


图 3.4 新型平台快速模式的典型验证环境

3.5.3 新型平台慢速模式

新型平台慢速模式的设计是由一个完全的处理器模型来执行新型平台的软件代码。慢速模式的典型验证环境如图 3.5 所示。

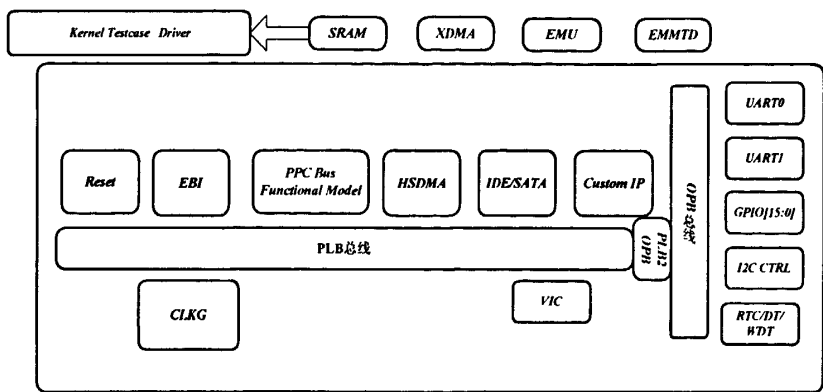


图 3.5 新型平台慢速模式的典型验证环境

3.5.4 虚拟验证平台的搭建

本文采用新型平台 SoC 系统级验证方法开发 H.264/AVC 编码芯片的虚拟原型验证平台，通过开发相应的验证组件来模拟 H.264/AVC 编码芯片实际工作时的周边元件，然后将这些验证组件、H.264/AVC 编码芯片的 RTL 代码和新型平台集成到一起；在新型平台上开发测试程序，将编译链接生成的可执行映像文件分别运行于新型的快速模式和慢速模式来完成验证项的测试。

在快速模式下，新型平台运行于 Unix 环境而非 H.264/AVC 编码芯片的仿真环境，使用 IC-SIM 管理新型平台的代码与挂接在 PLB 总线上用来替代处理器的总线功能模型间的接口，这种方法是最快的仿真模式，其缺点是在此模式下仿真平台设计中没有处理器，故系统验证精确度不高，适用于软件开发阶段。

在慢速模式下，新型平台的镜像被加载到存储器模型，使 H.264/AVC 编码芯片内部 PowerPC440 处理器模型可从中取指执行，从而达到模拟 H.264/AVC 编码芯片内部工作过程进行各项验证工作，是精确度最高、速度最慢的模式。

3.5.5 验证方法

本文根据虚拟平台的特点在该平台下进行部分模块级及系统级的验证。

模块级验证：基于新型验证系统的虚拟平台验证分为慢速与快速两种模式，根据两种模式的特性以及为了提高验证效率，先在快速模式下进行除了处理器模块之外的所有功能验证，确认功能正确后，在慢速模式下进行包括处理器模块在内的所有功能验证。在虚拟平台下的验证项内容根据系统集成的特性较系统集成前的 IP 验证内容略有裁剪。

系统级验证：系统验证项也在两个验证系统中均验证。

3.6 FPGA 原型验证平台设计

随着半导体制造技术的不断前进和相应的设计规模以及复杂度增加，使得传统的软件仿真工具已不可能完全解决功能验证的问题。而且一些需要处理大量实时数据的应用（如视频）也越来越多，因此要求能够在接近实时的条件下进行功能验证。

FPGA 验证成为 SoC 设计流程中重要的一个环节，一方面作为硬件验证工具，可以将所设计的 RTL 级代码综合实现后写入 FPGA 芯片进行调试检错；另一方面可以进行软件部分的并行开发，在验证板上检测驱动程序、启动操作系统。FPGA 验证的流程相当于一个 FPGA 设计的主要流程，它主要分为设计输入、综合、功能仿真（前仿真）、实现、时序仿真（后仿真）、配置下载、下载后板级调试检错这几个步骤。FPGA 验证能够保证仿真环境真实性的同时，以最快的速度发现可能存在逻辑功能和时序上的问题，是整个 SoC 设计中一个重要而且有效的必须环节，改进了 RTL 级设计代码，验证功能的正确和完整性，提高 SoC 流片成功率^[11]。

3.6.1 FPGA 平台的验证环境及相关工具

FPGA 平台下验证环境以及相关工具如表 3.2 所示：

表 3.2 FPGA 验证平台测试工具、环境

验证环境	FPGA 验证平台、LINUX
程序开发工具	GUN、Tornado
硬件描述语言	VHDL, Verilog HDL
软件语言	C, PowerPCASSEMBLE language
调试工具	RiscWatch

3.6.2 PFGA 验证平台的搭建

基于 PowerPC440 处理器的 FPGA 原型验证平台是将 H.264/AVC 编码芯片视频编码核和其他 IP 系统综合后的逻辑后下载到 FPGA 芯片中，H.264/AVC 编码芯片的输入输出通过开发板上的其它外设器件来完成。将测试程序烧入 FPGA 验证板上 FLASH 单元，FPGA 验证板上电后处理器从 FLASH 中取指执行；或通过仿真器（如 RiscWatch）将 ELF 格式（Excutive Linked File，即一种为 linux 系统所采用的通用文件格式，支持动态链接）可执行代码加载到片内 DPRAM 中，结合仿真器对测试程序进行在线调试，这样就达到与 H.264/AVC 编码芯片实际工作过程相同的效果。

3.6.3 验证内容

验证内容是按照第二章描述的芯片架构通过总线互联,将整个环境综合进 FPGA 所设计的。验证程序由 C 语言实现,加载进 FPGA 内部的 bram,然后通过外部实际电路配合产生或响应操作。本文通过与计算机、H.264/AVC 编码芯片进行通信,来检测各模块功能是否正确,在测试过程中对代码的时序检查也有一定的意义。通过实际电路和外部的真实环境和激励来验证代码,整个 FPGA 内的代码与真是芯片的架构基本一致,可以在真实应用环境中进行替代测试。而且验证速度快,可同时对功能和时序进行测试验证。

虚拟原型验证平台优点是它可以仿真模拟出芯片的实际工作条件,仿真出被测设计在各种实际激励下的响应,方便除错,但速度较慢;FPGA 的优点是速度快,可以大量运行和调试软件,但可观测性较差;两者结合可以得到很好的效果与速率的平衡^[18]。

第四章 H.264/AVC视频编码核的验证

本章将简述 H.264 的视频编码系统，分析 H.264/AVC 编码核所采用的是软硬件协同设计，针对软件所负责的功能进行基本描述。重点针对 H.264/AVC 视频编码核进行软硬件协同验证。首先将根据编码核编码时所需要共同工作的外围模块，比如中断控制器、PCI 接口等进行模块级验证，然后联合这些模块和编码核进行系统级验证。模块级验证是保证硬件上的每一个模块能够正常可靠的工作，是实现编码芯片能正常工作的基础。对编码核做系统级验证时，本文在保证编码核与外围模块能协同编码的同时，还会从编码算法、编码核功能、性能等多角度来保证编码核验证的完备性和编码结果的正确性。

4.1 H.264/AVC视频编码核结构

4.1.1 算法结构

H.264 并不明确地规定一个编解码器如何实现, 而是规定了一个编码完成的视频比特流的句法, 和该比特流的解码方法。H.264/AVC 视频编码核的功能组成如图 4.1 所示:

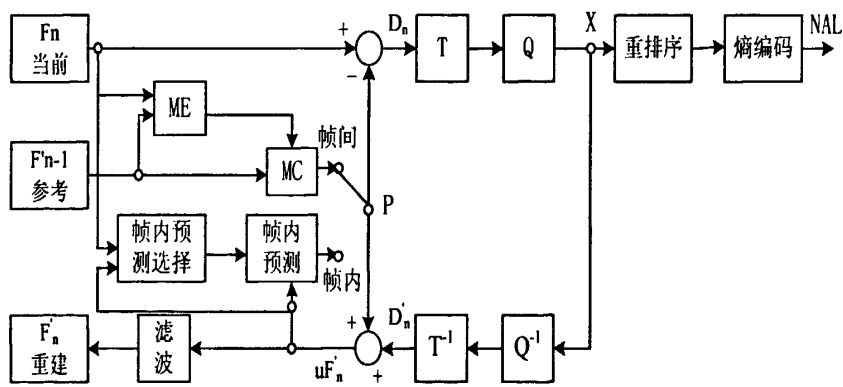


图 4.1 H.264/AVC 视频编码核的算法结构

从上图可见, H.264 和基于以前的标准(如 H.261、H.263、MPEG-1、MPEG-4)中的编码核功能块的组成并没有什么区别,主要的不同在于各功能块的细节。由于视频内容时刻在变化,有时空间细节很多,有时大面积的平坦。这种内容的多变性就必须采用相应的自适应的技术措施;由于信道在环境恶劣下也是多变的,例如互联网,有时通畅,有时不畅,有时阻塞,又如无线网络,有时发生严重衰落,有时衰耗较小,这就需要采取自适应方法对抗这种由于信道畸变而产生的不

良影响^[8]。

由图 4.1, 输入的帧或场 F_n 以宏块为单位被编码核处理。首先, 按帧内或帧间预测编码的方法进行处理。

如果采用帧内预测编码, 其预测值 PRED (图中用 P 表示) 由当前片中已经编码部分经预测得到; 如果采用帧间预测编码, P 是由当前片中前面已编码的参考图像 F'_{n-1} 经运动补偿 (MC) 后得出。为了提高预测精度, 从而提高压缩比, 实际的参考图像可在过去或未来 (指显示次序上) 已编码解码重建和滤波的帧中进行选择。

预测值 PRED 和当前块相减后, 产生一个残差块 D_n 。经块变换、量化后产生一组量化后的变换系数 X, 再经熵编码, 与解码所需的一些头信息 (如预测模式量化参数、运动矢量等) 一起组成一个压缩后的码流, 经 NAL (网络自适应层) 供传输和存储用。

正如上述, 为了提供进一步预测用的参考图像, 编码核必须有重建图像的功能。因此必须使残差图像经反量化、反变换后得到的 D'_n 与预测值 P 相加, 得到 uF'_n (未经滤波的帧)。为了去除编码解码环路中产生的噪声, 为了提高参考帧的图像质量, 从而提高压缩图像性能, 设置了一个环路滤波器, 滤波后的输出 F'_n 即重建图像可用作参考图像。

对 H.264 来说, 整个运算量在各部分的分配如下: 帧间预测和帧内预测占 60%~70%, 整数变换约占 10%, 熵编码约占 20%, 环路滤波占 10%^[1,7]。

4.1.2 功能实现结构

对于基于 H.264 协议的视频编码系统, 如果单纯用软件实现整个编码过程, 会由于整个编码系统算法的高复杂度, 以及大量的数学运算, 使得软件的计算能力差, 速度慢, 而且容易造成总线拥挤。如果单纯用硬件实现, 视频压缩实现的复杂度过高, 缺乏灵活性, 并且容错能力差, 完成整个编码系统的效率也低。所以为了缩短整个编码的时间, 提高编码系统的工作效率, H.264/AVC 编码核采用软硬件协同设计, 将软件中耗费时间和资源较多的模块用硬件来实现, 而软件主要实现对视频编码核管理、速率控制和参考帧管理等, 硬件主要实现运动估计、宏块编码、熵编码和去块去块效应滤波等, 其结构框图如图 4.2 所示。其中, FW 就是编码核的软件部分。

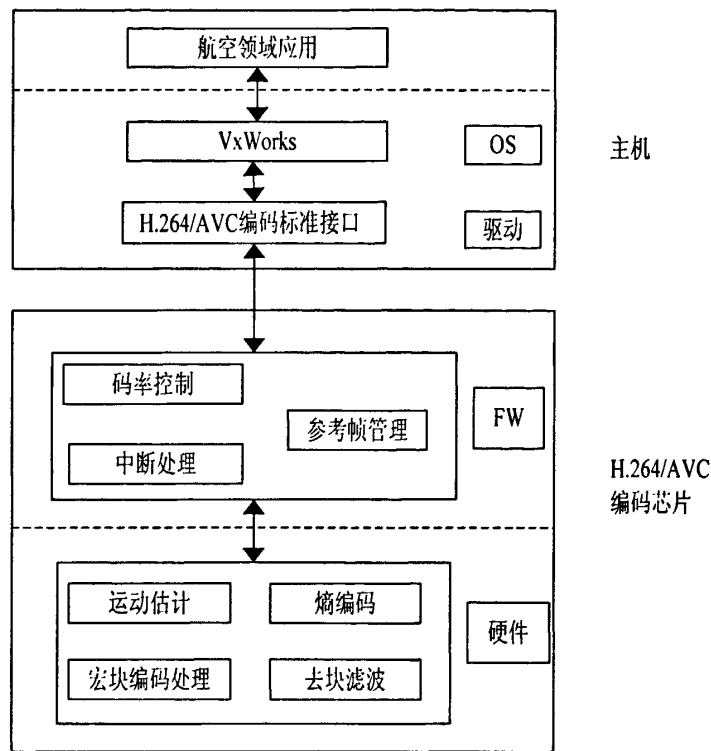


图 4.2 H.264/AVC 编码核软硬件功能划分

H.264/AVC 视频编码核分为三层：主机驱动、FW 固件以及视频核硬件部分。其中，主机驱动模块是为用户提供视频顶层接口；FW 固件模块由 PowerPC440 处理器执行，辅助硬件编码核实现编码过程；硬件层主要实现了运动估计、宏块编码、熵编码、去块滤波等相关功能。

H.264/AVC 视频编码核的在 FPGA 上实现的如图 4.3 所示。

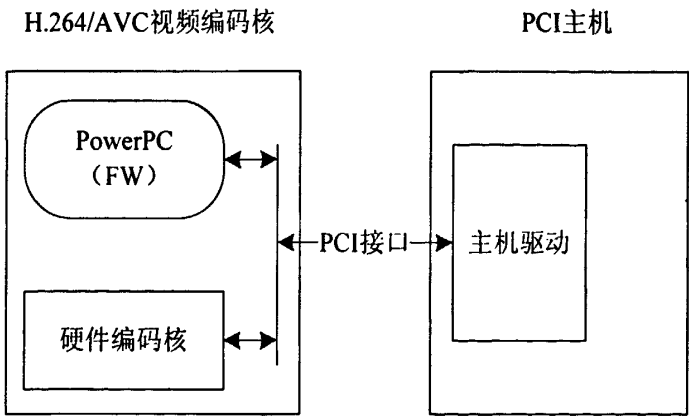


图 4.3 H.264/AVC 编码核 FPGA 实现图

(1)主机接口的实现

主机接口的程序主要是通过 PCI 接口实现的，其通过 PCI 接口将上层用户应用

的参数配置及控制信息传给 FW 模块，另外负责也是从 PCI 接口将 FW 接口提交的信息传送给上层应用，主要实现视频编码过程中的参数配置、视频输入接口和码流输出接口的配置、待编码数据的输入、编码后的码流输出、编码处理的启动以及整个编码处理的控制。

因此对 PCI 接口模块级验证的完备性和正确性是编码核能实现编码功能的基础。

(2)FW 固件的实现

FW 与硬件共同完成 H264 的编码功能。FW 接收主机传送的信息并进行处理，主要完成编码控制、码率控制、参考帧管理、中断处理等功能协同硬件编码。

FW 固件是协同硬件编码核共同编码的软件模块，主要功能包括获取码流基本参数，配置设备驱动，分配内存，配置硬件，以及获取新帧数据。具体流程如图 4.4 所示。

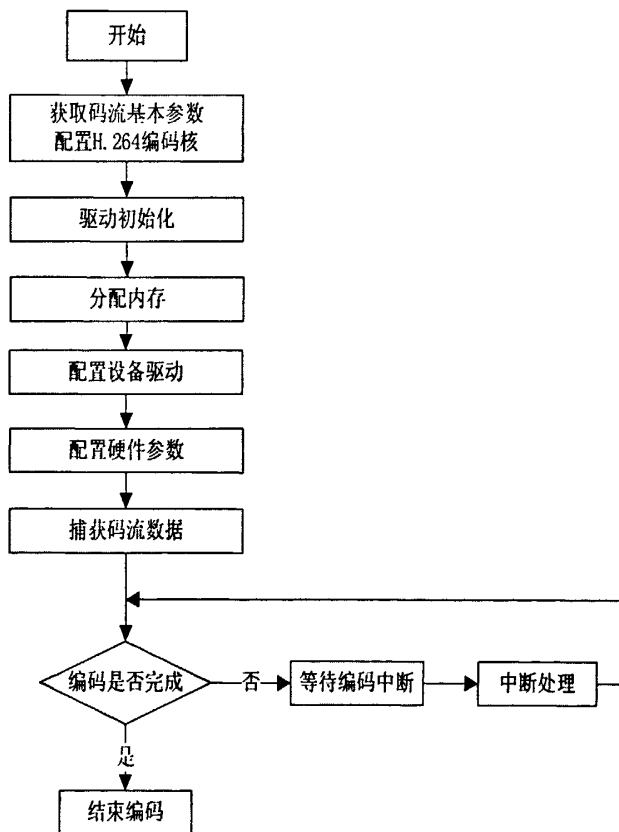


图 4.4 H.264/AVC 编码核 FW 固件流程图

从上图可以看出，FW 是一个中断驱动软件，此模块是用中断的方式与硬件编码核进行通信一起完成编码功能。编码核编码时将一帧图像按像素分为相应的宏块行，编码核以宏块行为单位进行编码，当一行宏块编码完成后硬件编码核发出中断给中断控制器，中断控制器判断是否正确中断源，PowerPC440 处理器接收到

此中断后，进入到中断服务程序对此宏块行结束中断进行处理。当一帧所有的宏块行都处理完成，同样硬件发出图像结束中断，处理器处理此中断。当一帧编码完成后，硬件编码核会先后发送刷新缓冲区中断和码流输出中断，处理器依次处理完成之后处理就继续等待接收新帧中断。以往反复，直到所有待处理的视频图像都编码完成，其流程图如图 4.5 所示。

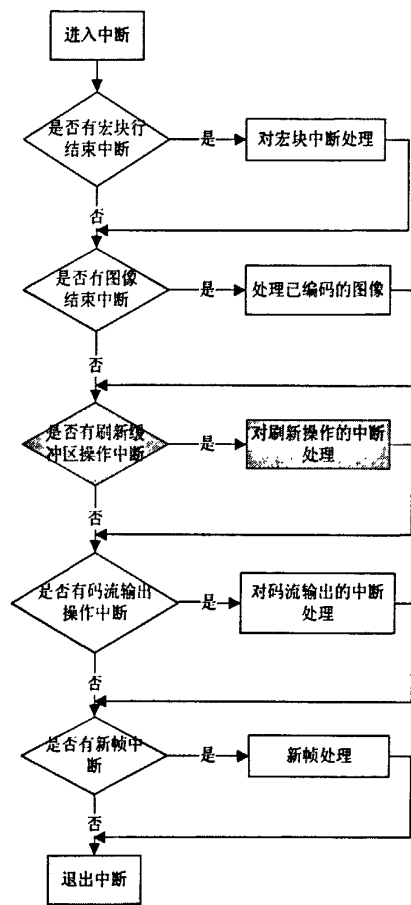


图 4.5 中断处理的流程图

如图 4.6 所示是编码核的仿真波形，从图中可以看出编码核编码的流程，其中 row_done 是宏块行结束中断信号，pic_done 是图像结束中断信号，hep_flush_done 是刷新缓冲区中断信号，hep_so_done 是码流输出中断信号，vi_new_frame 是接收新帧中断信号。

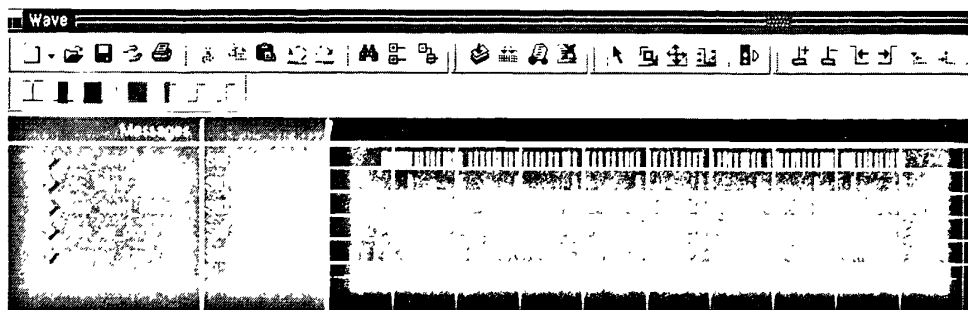


图 4.6 编码核仿真波形

4.2 编码核层次化验证

在分析 H.264/AVC 视频编码核的编码原理以及功能实现结构后,就需要对编码核周边 IP 以及编码核系统进行软硬件协同验证。结合第三章所介绍的软硬件验证环境以及验证流程,首先就是编写并测试 IP 和编码核的系统初始化程序。系统运行环境初始化用来初始化硬件设备、设置堆栈,建立内存空间的映射图,从而将系统的软硬件环境带到一个合适的状态,完成系统的启动程序后,然后开发验证测试程序。除处理器核外,要根据规范开发出每个模块相应的测试程序。所有的验证程序都必须在虚拟原型验证平台和 FPGA 原型验证平台下进行验证,保证验证的可靠性和充分性。测试软件开发过程中,本文主要针对与编码核紧密相关的中断控制器和与主机相连的 PCI 的测试程序,然后再开发 H.264/AVC 视频编码核相关程序。

4.2.1 模块级验证

在 H.264/AVC 编码芯片的软硬件协同验证中,测试软件的编辑和编译在 PowerPC440 处理器集成开发环境的验证平台中进行,软件调试用集成的调试环境 RiscWatch 进行。

模块级验证的对象是 IP/模块的各功能点,主要关注系统集成、互连后片上各个功能模块是否能够正常工作。模块级的验证必须覆盖各个模块的所有功能,必须保证硬件上任何一个模块能够正常可靠的工作,它是系统级验证的基础。模块级验证的测试输入是通过处理器执行测试程序来完成的,测试结果是通过处理器来输出到其功能仿真模型或外设的。

(1) 中断控制器的协同验证

中断控制器接收来自中断源的中断信号,并判断出中断信号各自的优先级,按优先级发报给 PowerPC440,PowerPC440 通过查询相应的寄存器确定中断源的中断服务的入口地址,从而执行对应的中断服务程序。PowerPC440 支持两种中断,

非关键中断 IRQ 与关键中断 FIQ, 硬件编码核与 FW 固件通信所发的中断属于 IRQ 中断。

在对中断模块进行测试验证时, 首先通过配置中断控制器的寄存器对相应中断进行初始化, 然后进行程序设计使其产生相应的中断, 最后是编写相应的中断服务程序, 将设计好的测试程序在开发环境中经过链接、编译后加载到实际的验证环境下运行。当中断完成时, 通过串口或通用 I/O 接口打印提示信息和观测一些相关寄存器的值来判断每路中断是否能正常的发生、相应的中断服务程序是否被正确执行及正常返回。在验证编码核时, 本文着重关注 FW 和硬件编码核的中断源, 设计了相应的中断服务程序去测试中断能否正常发生、处理及返回。

(2) PCI 接口的协同验证

PCI(Peripheral Component Interconnect)总线是由 Intel 公司首先推出的一种局部总线。它定义了 32 位数据地址总线, 并且可以扩展为 64 位, 支持突发读写操作, 同时可以支持多组外围设备。

本文所设计的 PCI 主机接口主要实现了编码芯片与 PCI 总线的连接, PCI 接口逻辑包含了 2 种不同类型的通道, Register 类型的从通道 0 (Target0), FIFO 类型的从通道 1 (Target1)。Register 类型从通道 0, 实现了 PCI 主机和 PowerPC440 到编码芯片内部寄存器的访问及数据通信。FIFO 类型从通道 1, 用于测试模式, 实现了 PCI 主机与芯片内部视频、音频输入接口和码流输出接口模块之间的数据交换。

PCI 接口的基本特征:

1. 完全支持 PCIv2.2 协议;
2. 32 位数据/地址总线;
3. 总线频率支持 33MHz 和 66MHz;
4. PCI 配置寄存器只能由 PCI 主机配置;
5. 目标接口支持 FIFO 通道类型和 Register 类型。

在当前的 PC 体系结构内, 几乎所有外部设备采用的各种各样的接口总线, 均是通过桥接电路挂接在 PCI 系统内的。在这种 PCI 系统中, Host/PCI 桥称为北桥, 连接主处理器总线到基础 PCI 局部总线, H.264/AVC 视频编码核就相当于一个从设备挂接在 PCI 总线上。

PCI 有三个相互独立的物理地址空间: 除了设备存储器地址空间、I/O 地址空间之外, 为支持 PCI 设备系统资源的自动配置, 还定义了配置空间。配置空间是 PCI 所特有的一个物理空间, 如表 4.1 所示。

表 4.1 PCI 配置空间

DW	Byte3		Byte2		Byte1		Byte0		Addr
0	Device ID				Vendor ID				00
1	Staus				Command				04
2	Class Code						Revision ID		08
3	BIST		Header Type		Latency Timer		Cache Line		0C
4	Base Address Register 0(BAR0)								10
5	Base Address Register 1(BAR1)								14
6	Base Address Register 2(BAR2)								18
7	Base Address Register 3(BAR3)								1C
8	Base Address Register 4(BAR4)								20
9	Base Address Register 5(BAR5)								24
10	Cardbus CIS Pointer								28
11	Subsystem Device ID				Subsystem Vendor ID				2C
12	Expansion ROM Base Address								30
13	Reserved (Capability List)								34
14	Reserved								38
15	Max Lat		Min lat		Interrupt Pin		Interrupt Line		3C

值得注意的是，PCI 接口的验证环境不同于第三章的验证平台，因为上述环境下的 IP 验证都是由处理器 PowerPC440 访问的。对于 PCI 接口的验证，其主设备是 PCI 主机。由于 FPGA 板卡插不到 PC 机的 PCI 插槽，所以解决方法是在 FPGA 板卡后背一个主机板 MPC8245，由 MPC8245 的处理器作为 PCI 主机对本文所设计的 PCI 接口进行验证，验证环境是 VxWorks 操作系统和 Tornado 验证平台。

本文选用 VxWorks 作为验证 PCI 的操作系统，是因为 VxWorks 是运行在目标机上的高性能、可裁减的，目前世界上用户数量最大的嵌入式实时操作系统。VxWorks 被广泛的应用在通信、航天、航空、军事等高精尖及实时性要求极高的领域中，如通信卫星、弹道制导、飞机导航等。嵌入式实时操作系统负责实时嵌入式系统的所有软硬件资源的分配、调度工作、控制和协调并发活动，如内存管理、任务调度、同步机制、异常和中断处理、任务间通信等，具有一般操作系统的基本功能。同时它也必须体现其实时处理和嵌入式系统特征。嵌入式操作系统具有自引导性、实时性、健壮性、可移植性、小规模性等特点。

嵌入式实时操作系统在目前的嵌入式应用中用得越来越广泛，尤其在功能复杂、系统庞大的应用中显得愈来愈重要。

Tornado 集成开发环境提供了高效明晰的图形化的实时应用开发平台，它包括一套完整的面向嵌入式系统的开发和调测工具。

具体操作是将 VxWorks 映像文件先下载到目标板上，启动该映像，即启动 VxWorks 操作系统。先使用烧写工具将 Bootloader 烧写到 flash，然后利用 Bootloader 下载内核映像，再运行 VxWorks 内核映像就可以在 Shell 看到如图 4.7 出现的信息。

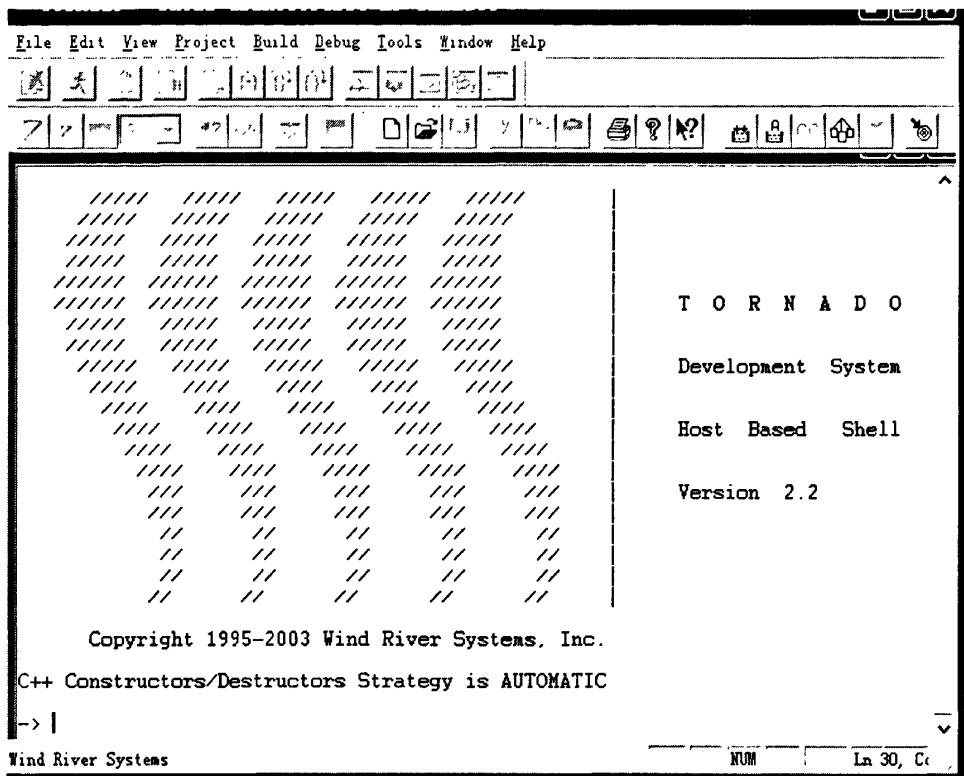


图 4.7 VxWorks 启动界面

在完成环境搭建及软件调试之后，就可以验证 PCI 接口。

本文从以下几个角度对 PCI 接口进行验证：

- 1. PCI 接口配置寄存器的验证：PCI 设备必须支持配置寄存器中的供应商 ID、设备 ID、指令和状态区。
- 2. T0/T1 通道读写验证：PCI 主机通过 T0 通道对编码核进行配置，通过 T1 通道灌入视频编码数据 and 取出编码核编好码流的输出。
- 3. PCI 核做目标时对校验和错误的检测功能验证：当 PCI 核做目标时，检测到校验和错误后在状态寄存器中能否正常响应。

以上的验证项都是保证 H.264/AVC 视频编码核能正常工作的基础。

4.2.2 系统级验证

系统级验证的重点不再是各 IP/模块的某个独立的功能点是否正确，而是各个模块之间的互动操作，通过整个系统的各个功能模块协同运行验证。对于 H.264/AVC 视频编码核的功能实现，从两条通路进行验证，分别是协同用于调试的 PCI 接口，和用于实际应用的视频输入接口和码流输出接口。

(1)编码核协同 PCI 主机

本节将协同 PCI 主机对视频编码核进行系统级验证，验证平台是 Tonado 环境和第三章所述的新型验证平台。

其中 PCI 主机上的测试程序运行在 Tonado 环境下，处理器是 MPC8245 主机板上的 CPU，而 FW 固件代码运行在新型平台，处理器就是高性能的 PowerPC440。

PCI 主机程序负责配置视频编码核、导入视频数据以及导出码流数据，其难点就是 PCI 主机在灌入数据的同时还要兼顾将编好的数据取出，否则如果码流取不出来会导致缓冲区堵塞，编码核会停止编码。

本文在解决上述难题时使用的是中断机制，即编码核如果编好码流则发中断通知 PCI 主机取码流，进视频新帧时也是 PCI 等待编码核发相应中断。

(2)编码核协同外围接口

除了 PCI 总线接口，H.264/AVC 视频编码核还有其他三部分外围接口：视频输入接口，存储器控制接口和码流输出接口,如图 4.8 所示。

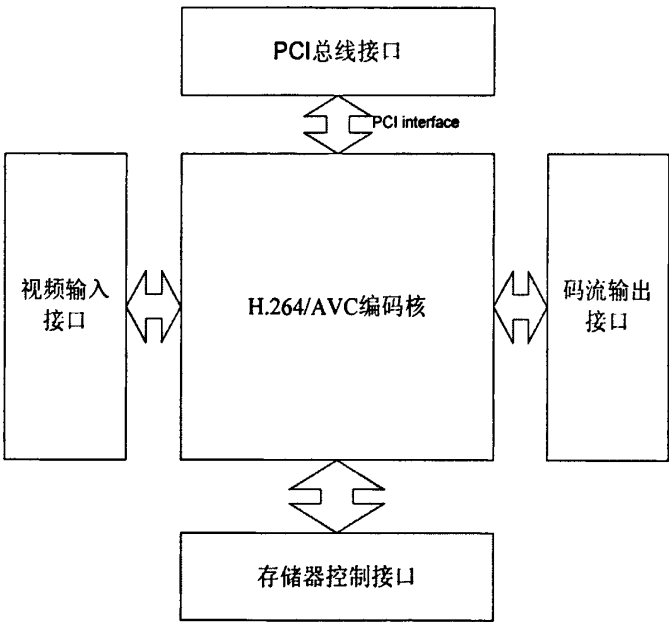


图 4.8 H.264/AVC 外围接口

上图的视频输入接口负责根据当前编码的需要，从存储原始图像的 YUV 文件中读取原始图像数据，并将数据按照既定接口时序发送到 H.264/AVC 视频编码核。编码核在编码、传输时需要对数据进行暂时存储，因此其过程中的视频数据就通过存储器控制接口写入外部存储器。而码流输出接口是收集编码核输出的码流，将码流输入到解码器或者通过协议栈发送到互联网上，亦或者存为文件放在硬盘上。

本文在对视频输入接口、码流输出接口和存储器控制接口协同编码核进行系统级验证时，所有的测试程序都运行在第三章所述的新型验证平台，处理器就是高性能的 PowerPC440。

本文从应用角度出发，进行了多种视频格式、多种码流输出方式的验证，其编

出的码流完全符合 H.264 协议标准。

4.3 编码核功能验证

对于 H.264/AVC 视频编码核功能正确性的验证方法, 本文将编码核输出的码流与 JVT 的 H.264 标准进行比较, 即如果编码正确, 则经过标准解码器解码后可得到正确的解码图像。

H.264/AVC 编码核可以实现符合《H.264/AVC 通用视听业务的先进视频编码》标准的 Main Profile 的视频编码, 支持电视制式和 VESA 制式两种类型的视频编码图像, 支持从 352x288 到 1920x1088 的多种分辨率。

4.3.1 基于帧类型的验证

(1) 帧结构

H.264/AVC 将画面分为构成方式不同的三种帧: 帧内编码 I 帧, 前向预测编码 P 帧和双向预测编码 B 帧。

H.264/AVC 视频编码核中, 已编码图像存储在参考缓冲区供帧间宏块的运动补偿预测使用。参考图像缓冲区通常由 H.264/AVC 视频编码核发送的 IDR (瞬时解码器刷新) 编码帧刷新。当解码器接收到 IDR 帧时, 立即将缓冲区中的图像标为“非参考”, 并将所有的参考帧队列丢弃。因此, 编码视频序列的第一幅图像都是 IDR 帧, 而 IDR 帧属于 I 帧。

I 帧用原始数据直接编码, 不与其它画面比较。所以, I 帧的编码量信息和数据量都很大, 但可以独立还原画面。如果视频画面中出现紊乱如马赛克等, 只要有 I 帧出现就能中止错误的画面。

P 帧用邻近的 I 或 P 帧做为基准进行运动预测, 只记录当前画面与基准的不同之处。所以, 编码效率较高, 数据量小, 但还原画面就需要前面的基准帧数据。

B 帧可以同时用前面和后面的帧作为比较基准进行运动预测。编码压缩率最大, 数据量最小, 而且不会传递错误。但需要附加存储器, 时间上有迟后, 还不能作为以后的预测基准。

两个 I 帧之间的帧序列(包括 P 帧或者 P 帧和 B 帧)即为一个图片组 GOP(Group of Pictures)。一个 GOP 就是一组连续的画面, I 帧是一个完整的画面, 而 P 帧和 B 帧记录的是相对于 I 帧的变化, 若基准画面已经有错, 那么这个画面上的错误会被传下去, 直到 I 帧到来。因此, GOP 的选定会影响编码质量, 不要选太长^[14,25]。

(2) 对于 I 帧编码的验证

H.264/AVC 视频编码核对于 I 帧编码的基本流程是:

Step1: 进行帧内预测, 决定所采用的帧内预测模式。

Step2:像素值减去预测值,得到残差块。

Step3:对残差进行变换和量化。

Step4:变长编码和算术编码。

Step5:重构图像并滤波,得到的图像作为其它帧的参考帧。

H.264/AVC 视频编码核不同于基于其他协议标准的视频编码核之一是在帧内编码 I 帧中,又加入了帧内预测编码技术,即解码时可用周围数据的差分值来重构画面,在视频压缩中通常采用帧内预测技术去除图像的空间冗余度。在以往 H.263+、MPEG-4 等视频压缩编码标准中,帧内预测被引入变换域。H.264/AVC 视频编码核则采用了更为精确和复杂的帧内预测方式,参考预测块左方或者上方已编码块的邻近像素点被引入空间域。但是,如果参考预测块是帧间编码宏块,该预测会因参考块的运动补偿引起误码扩散。所以,参考块通常选取帧内编码的邻近块。

在帧内预测模式中,预测块是基于已编码重建块和当前块形成的。帧内预测(包括亮度预测和色度预测)以特定大小的块作为基准单元,从周围像素中预测编码化的模板。H.264 的基准块大小分 4×4 和 16×16 两种, 4×4 亮度子块有 9 种可选预测模式(0:垂直,1:水平,2:DC,3:下左对角线,4:下右对角线,5:右垂直,6:下水平,7:左垂直,8:上水平),独立预测每一个 4×4 亮度子块,适用于带有大量细节的图像编码; 16×16 亮度块有 4 种预测模式(0:水平、1:垂直、2:DC、3:plane),预测整个 16×16 亮度块,适用于平坦区域图像编码;色度块也有 4 种预测模式,类似于 16×16 亮度块预测模式。编码核通常选择使预测块和编码块之间差异最小的预测模式。因为原图像左右、上下都有一定的相关性,不同部分的差分值总比不进行帧内预测的原始值要少,故量化后的编码量就减少了,而且高频成分也比原图像为少。这样,相对于直接对该帧编码而言,可以大大减小码率。

H.264 采用拉格朗日率失真优化(RDO)进行最优化编码模式选择,通过遍历所有可能的编码方式,然后选择最小率失真代价模式作为帧内预测模式。

本文通过验证工作中大量的数据统计以及相关资料查找,发现 4×4 亮度子块的后 4 种可选预测模式(5:右垂直,6:下水平,7:左垂直,8:上水平)被用来选择进行帧内预测编码的概率还不及 5%,而且对于每一个宏块都要遍历所有的预测模式,这样大大降低了编码核的工作效率。因此,H.264/AVC 视频编码核只进行前 5 种帧内预测模式,并且编码核的编码正确性不受影响,编码效果也完全在可接受范围内。

因此可以通过配置 GOP 进行全 I 帧的验证,此时所有帧都进行帧内预测编码,编码速度快,但是编出的码流压缩比小,文件最大。

(3)对于 P 帧编码的验证

H.264/AVC 视频编码核对于 P 帧编码的基本流程是:

Step1:进行运动估计, 计算采用帧间编码流程编码模式的率失真函数值。P 帧只参考前面最近解码的 I 帧或 P 帧, 称为前向预测。

Step2:进行帧内预测, 选取率失真函数值最小的帧内模式与帧间模式比较, 决定采用哪种编码模式。

Step3:计算实际值和预测值的差值。

Step4:对残差进行变换和量化。

Step5:熵编码, 如果是帧间编码模式, 编码运动矢量。

一般而言, 帧间预测编码编码效率比帧内更高。有人测得, 对缓慢变化 256 级灰度的黑白图像序列, 帧间差超过阈值 3 的像素不到一帧像素的 4%; 对剧烈变化 256 亮度值的彩色电视序列, 帧间差超过阈值 6 的像素平均只占一帧的 7.5%。因此, 通过帧间预测就可以很好的利用时间相关性进行视频压缩。

在帧间预测编码中, 由于活动图像邻近帧中的景物存在着一定的相关性。因此, 可将活动图像分成若干块或宏块, 并设法搜索出每个块或宏块在邻近帧图像中的位置, 并得出两者之间、空间位置的相对偏移量, 得到的相对偏移量就是通常所指的运动矢量, 得到运动矢量的过程被称为运动估计。

将运动矢量和经过运动匹配后得到的预测误差共同发送到解码端, 在解码端按照运动矢量指明的位置, 从已经解码的邻近参考帧图像中找到相应的块或宏块, 和预测误差相加后就得到了块或宏块在当前帧中的位置。

在细节比较多的帧中如果选择较大的块尺寸, 意味着用于表明运动矢量和分割区域类型的比特数会少些, 但是运动压缩的冗余度要多一些; 如果选择小一点的块尺寸, 那么运动压缩后冗余度要少一些, 但是所需比特数要比较多。因此必须要权衡块尺寸选择上对压缩效果的影响, 一般对于细节比较少、比较平坦的区域选择块尺寸大一些, 对于图像中细节比较多的区域选择块尺寸小一些。

如图 4.9 所示, P 帧就是用 I 帧或 P 帧经前向预测帧间编码的。

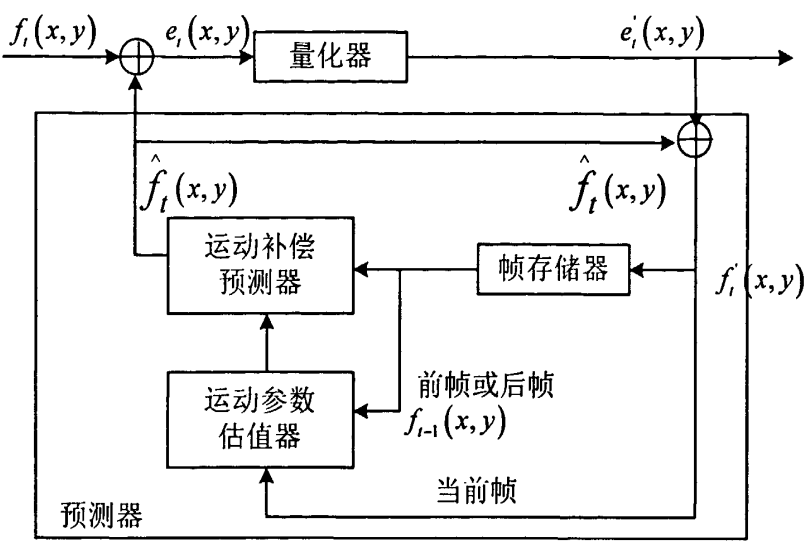


图 4.9 单向预测帧间编码

通过运动估计可以去除帧间冗余度，使得视频传输的比特数大为减少，因此，运动估计是视频压缩处理系统中的一个重要组成部分。

验证 P 帧的编码是将 GOP 配置进行 IP/IPPPP 等不同的帧序列，发现一个 GOP 中如果 P 帧越多，编码越慢，但是编出的.264 视频文件会越小。

(4)对于 B 帧编码的验证

H.264/AVC 视频编码核高性能的体现之一就是编码核支持 B 帧编码。B 帧编码与 P 帧的唯一区别是，在运动估计时，B 帧图像使用两帧图像作为预测参考，即双向预测。其中一个参考帧在显示顺序上先于编码帧，另一帧在显示顺序上晚于编码帧，B 帧的参考帧在任何情况下都是 I 帧或 P 帧。由于 B 帧本身不用于参考预测的基准，所以误差不会传递。

B 帧帧间预测时，利用前向参考帧预测当前帧称为前向运动补偿，利用后向参考帧预测当前帧称为后向运动补偿，利用前后向同时预测的就称为双向预测运动补偿，如图 4.10 所示。双向预测在实时通信中是不能应用的，例如会议电视、可视电话等，因为后向预测在当前帧之后进行，会引入编码时延。

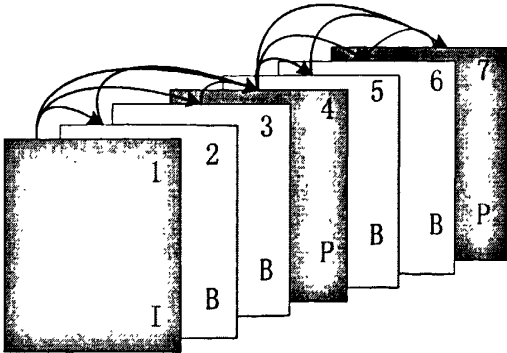


图 4.10 双向预测

本文通过实验发现，由于 B 帧编码的复杂度最高，所以当 一个 GOP 中 B 帧数目越多时，编码完成的码流文件压缩比也越大，即文件比较小，但是编码的用时时间也越长。

如表 4.2 所示，当一个 GOP 中只有一个 P 帧时，B 帧数目越多编码核的压缩效率也越大，相比如果 GOP 中没有 B 帧时，编码效率随 P 帧数改变。可以证明，虽然 B 帧的算法复杂度高，但是 B 帧的压缩效率却很好。

表 4.2 基于帧结构的实验数据

GOP	原始文件	全 I 帧	IP	IPPPP	IBP	IBBP
文件大小	21.7M	2.5M	1.1M	611.5k	879.9K	721.7K
压缩效率		8.7 倍	19.7 倍	36.3 倍	25.3 倍	30.7 倍

4.3.2 基于视频格式的验证

(1)视频信号的采样

运动图像是由每秒若干帧静止图像构成，对视频信号的采样可以通过逐行扫描或者隔行扫描。

如果通过逐行扫描，那么得到的信号是一帧图像。通常，我国采用的 PAL 制彩色电视的帧频是每秒 25 帧，美日等采用的 NTSC 制彩色电视则为每秒 30 帧。如果是会议电视、可视电话等运动量不大的视频信号，帧频也可取 15~20 帧/秒，但低于 15 帧/秒的视频质量不高。

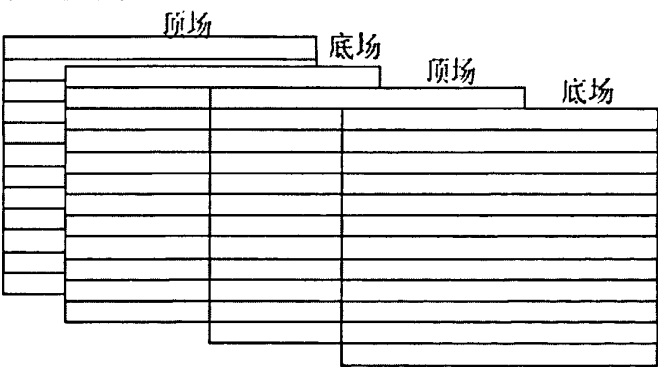


图 4.11 场结构

如果通过隔行扫描（奇偶数行），如图 4.11 所示，一帧图像就被分成了两场，每场由若干行组成，奇数行和偶数行各构成一场，它们分别为顶场和底场。通常 PAL 制的场频为 25HZ，NTSC 制场频为 30HZ。

对于帧频和场频的由来，是最早由于抗干扰和滤波技术的限制，电视图像的场频通常与电网频率（交流电）相一致。于是根据各地交流电频率不同就有了欧洲和中国等的 PAL 制的 50HZ 和北美等 NTSC 制的 60HZ。但是现在没有这样的限制

了，帧频可以和场频一样，甚至场频更高。

(2)输入的视频格式

H.264/AVC 视频编码核支持的视频输入格式包括电视制式从标清（576i）到高清（1080p）和电脑图像制式从 VGA（480）到 UXGA（1200）的一系列分辨率，具体如表 4.3 所示。

表 4.3 视频输入格式

分辨率	帧频率 (Hz)	扫描格式	支持的标准
1920x1088	30	隔行	SMPTE 274M
1920x1088	25	隔行	SMPTE 274M
1920x1088	30	逐行	SMPTE 274M
1920x1088	25	逐行	SMPTE 274M
1280x720	60	逐行	SMPTE 296M-2001
1280x720	50	逐行	SMPTE 296M-2001
720x576	25	隔行	ITU-R BT.656-4
720x480	30	隔行	ITU-R BT.656-4
1600x1200	60		VESA
1280x1024	60		VESA
1024x768	60		VESA
800x600	60		VESA
640x480	60		VESA

(3)帧编码和场编码

视频的一场或一帧可用来产生一个编码图像，视频帧也分为两种类型：连续或隔行的视频帧。隔行视频帧将一帧分为两个隔行的场，显然场内邻行之间的时间相关性较强，对活动量较大的运动图像则宜采用场编码方式。而逐行视频帧内邻行之间的空间相关性较强，因此活动量较小或静止的图像宜采用帧编码方式。

如图 4.12 和图 4.13 分别是帧编码和场编码的仿真波形，从波形可以看出在帧编码情形下，当编码核将一帧中所有宏块行编码完成后，产生图像结束中断并冲刷缓冲区以及输出编好的码流，并进下一帧新数据；而在场编码情形下，编码核将一帧图像分为奇偶两场，编码核按场编码，当一场的所有宏块行编码完成后产生一次图像结束中断，而只有当两场都完成后，编码核才会冲刷缓冲区输出码流，及进入新帧。

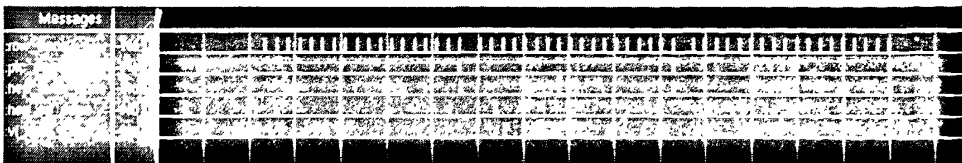


图 4.12 帧编码的仿真波形

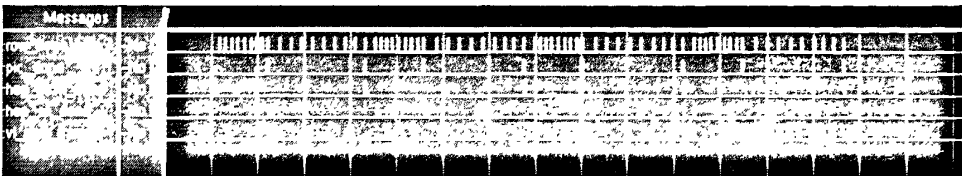


图 4.13 场编码的仿真波形

4.3.3 编码核功能验证方案

(1)仿真验证

H.264/AVC 视频编码核的仿真验证平台可模拟芯片的 SOC 架构, 通过 Host 配置不同的编码参数, 完成对视频编码核功能、接口等正确性验证。其仿真平台由两部分构成: Host&FW 的仿真和硬件层的仿真。

在芯片正常工作状态下, Host 程序主要用于在初始化阶段配置编码参数, 这部分程序通常运行在 Host PC 中, 通过 PCI 接口将编码参数配置到编码核中。FW 程序在芯片工作时运行在 PowerPC440 处理器中, 通过 PLB 高速总线与硬件逻辑进行通信。

H.264/AVC 视频编码核是软硬件协同编码, 视频编码软件常驻内存, 和硬件共同完成视频编码功能。但由于软件具有易实现且调试代价小的特点, 设计的第一步就是完全用软件模拟硬件编写一个 C 模型编码器, 其架构、算法和 H.264/AVC 视频编码核完全一样, 如图 4.2 所示。如果 C 模型编码器编码出来的*.264 视频文件可以用标准解码器解码, 那么可以说明 H.264/AVC 视频编码核基本算法原理是正确的。

验证完成 C 模型后, 就需要设计和代码编写 Verilog HDL 模型, 用真正的硬件和软件协同提供编码效率。

在验证 H.264/AVC 视频编码核功能的正确性时, 采用 C 模型和 Verilog HDL 模型协同验证的方式, 验证方案如图 4.14 所示。

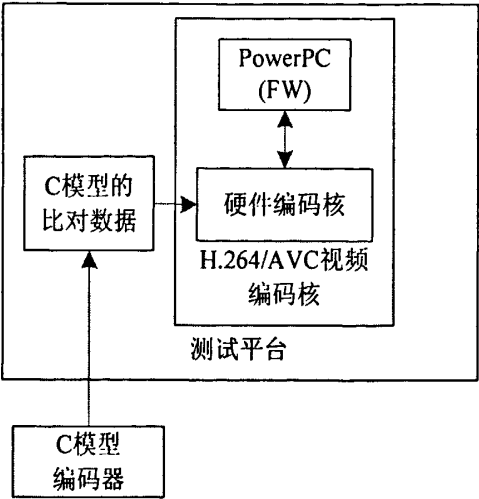


图 4.14 H.264/AVC 视频编码核验证方案

本文采用了大量的视频序列对 H.264/AVC 视频编码核功能进行仿真验证，其测试步骤如下：

Step1:随机地选择测试参数，组合为当前测试用例的测试参数组，参数组中包含的参数如表 4.4 所示。

表 4.4 H.264/AVC 视频编码核功能测试参数列表

测试参数	说明
视频序列分辨率	352x288、720x576、1280x720、1920x1088
帧场模式	帧模式/场模式
B 帧个数	0/1/2
码率模式	CBR、VBR

Step2:用 C 模型按照选定的测试参数进行编码，得到最终的编码码流以及各模块的测试验证数据。

Step3:将 C 模型编码输出的码流作为 H.264 标准解码软件 JM 的测试输入，对码流进行解码，验证码流一致性。如果 JM 能够正确地解码，说明编码码流符合 H.264 标准，否则说明有错误。

Step4:将 C 模型编码输出的验证数据作为 H.264/AVC 视频编码核的输入，运行 H.264/AVC 视频编码核，用来验证 H.264/AVC 视频编码核的正确性。如果 H.264/AVC 视频编码核输出的数据与 C 模型输出的数据完全一致，说明 H.264/AVC 验证平台编码正确，否则说明编码有错误。

仿真结果的正确性验证主要包括以下两个方面：

1.软硬件一致性。

验证 C 模型生成的编码码流与硬件逻辑生成的编码码流是否一致。

本文通过查看编码过程中是否有出错停止的情况判断编码过程中是否有软硬

件不一致的情况。当设定的编码帧数编完之后，可通过将 C 模型生成的码流与硬件生成的码流做二进制比对来验证软硬件是否一致。

2.软硬件编码与标准参考软件的一致性。

验证码流是否能够用 JM 顺利解码，解码后的解码图像与编码时的重建图像是否一致。

本文用 JVT 官方发布的软件 JM 对码流进行解码，验证码流是否满足与标准的一致性。解码后图像可与编码重建图像进行二进制比对，验证编码重建图像与解码图像的一致性。

H.264/AVC 视频编码核的验证结果如表 4.5 所示：

表 4.5 H.264/AVC 视频编码核的验证仿真结果

分辨率	帧编码/场编码	B 帧个数	码率模式	仿真结果	
				软硬件一致性	JM 一致性
1920x1088	帧编码	0/1/2	CBR	一致	一致
	场编码	0/1/2	VBR	一致	一致
1280x720	帧编码	0/1/2	VBR	一致	一致
	场编码	0/1/2	VBR	一致	一致
720x576	帧编码	0/1/2	CBR	一致	一致
	场编码	0/1/2	VBR	一致	一致
352x288	帧编码	0/1/2	CBR	一致	一致
	场编码	0/1/2	VBR	一致	一致

序列测试的结果表明 H.264/AVC 视频编码核工作正常，编码输出的码流与 JVT 的 H.264 标准一致，采用标准解码器解码后可得到正确的解码图像。从而可以说，H.264/AVC 视频编码核支持 352x288 到 1920x1088 多种格式的视频序列，并且可以准确无误地完成视频编码的功能。

(2)FPGA 平台验证

FPGA 具有高性能、灵活性、低开发成本，不会过时、低成本的结构化 ASIC 移植途径等特点，而理想的编码核结构应具有高性能、灵活性、方便更新、低开发成本等特点，并且随着应用的成熟和产量的提高，需要能够提供低成本移植的途径。因此，FPGA 非常适合视频和图像处理的体系结构，所以本文采用 FPGA 来实现图像的压缩编码。由于获取图像的摄像头输出的是 PAL 制式的模拟视频信号，FPGA 不能直接处理该信号，所以需要加入视频解码芯片将 PAL 制式的电视信号转换成 FPGA 可以处理的视频信号。在变换、编码和传输时需要对数据进行暂时

的存储，所以需要大容量且有较快读写速度的存储器。FPGA 平台方案设计如图 4.15 所示。

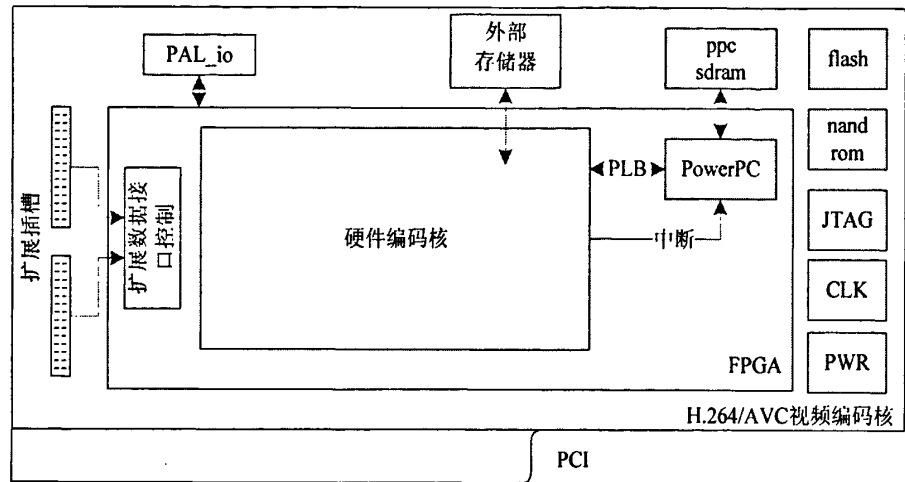


图 4.15 FPGA 平台设计方案

只有当 H.264/AVC 视频编码核仿真平台的验证足够充分后，才可以进行 FPGA 平台的验证。因为在 FPGA 平台不会进行与 C 模型输出码流的数据比对，也没有波形可以进行错误分析，惟一的验证依据就是编码核编出的码流用 JM 能够正确解码，即说明编码码流符合 H.264 标准。

4.4 编码核性能验证

4.4.1 H.264/AVC 视频编码核质量评估方法

对压缩后的视频质量评估是一件困难的工作。大体上，可分为主观视频质量评定和客观视频质量评定两种估计方法。

(1) 主观质量的评定

由于个人的视觉系统（HVS）不尽相同，对视频内容的熟悉程度也不一样。为了减少主观随意性，在对视频图像主观评定前，选若干名专家和“非专家”作为评分委员，共同利用五项或七项评分法对同一种视频图像进行压缩编码结构图像评定。最后按加权平均法对该压缩后的图像质量进行主观评定，如表 4.6 所示：

表 4.6 主观评价分数标准

CCIR 五级评分等级	评分等级	高清晰度采用七级评分等级	评价
	7	不能察觉任何图像损伤	特别好
	6	刚能察觉有图像损伤	相当好
优	5	不同程度的察觉, 轻度损伤	很好
好	4	有损伤, 但不令人讨厌	好
稍差	3	有令人讨厌损伤	稍差
很差	2	损伤令人讨厌, 但尚可忍受	很差
劣	1	非常令人讨厌损失, 无法观看	劣

测试方法可用随机次序请评委观察比较原始图像和压缩编码后的图像。国际上称为 DSCQS 的测试系统如图 4.16 所示。其中 A 为原始图像, B 为编解码后的图像, 以任意的 A、B 次序让评委打分评定。

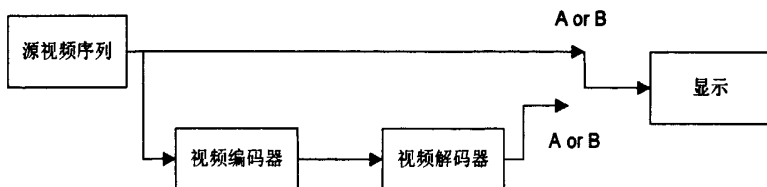


图 4.16 DSCQS 测试系统

(2)客观质量的测量

主观的视频质量评分更接近人的真实视觉感受, 当需耗费人力和时间, 成本较高。客观质量的测定方法速度快, 易实行, 但往往不会太符合人眼的视觉感受, 只能说大体上的质量。客观质量测定方法应致力于改进其测试标准和测试方法, 使其符合人的视觉感受。

最常用的测试标准是峰值信号与噪声之比 (PSNR):

$$PSNR_{db} = 10 \lg(2^n - 1)^2 / MSE$$

其中 MSE 为原始和编解码后图像之间的均方误差, $(2^n - 1)^2$ 为图像中最大可能的信号值平方, n 为表示每个像素的比特数。

一般讲, PSNR 愈高视频质量愈高; 反之亦然, 但实际上有时并非如此^[1,15]。

4.4.2 基于编码模式的验证

为了测试视频编码器视频编码系统的性能, 本文采用主观和客观质量评定相结合的方法, 将标准 JM 编码器和 H.264/AVC 视频编码核进行对比, 并通过两种编码模式 (CBR: 恒定目标码率/VBR: 可变目标码率) 进行测试。

码率是指数据传输时单位时间传送的数据位数, 单位一般是 kbps 即千位每秒。

单位时间内码率越大, 精度越高, 处理出来的文件就越接近原始文件, 但是文件体积与取样率是成正比的, 所以几乎所有的编码格式重视的都是如何用最低的

码率达到最少的失真，围绕这个核心衍生出来的 CBR（固定码率）与 VBR（可变码率）。

其中，码率计算公式：文件体积=时间 X 码率/8，时间单位是秒。举例，D5 的碟，容量 4.3G，考虑到音频的不同格式，占用一定的空间，姑且算为 600M，视频文件应不大于 3.7G，视频长度 100 分钟（6000 秒），计算结果：码率应为 4900K。

(1)CBR 模式

CBR(Constant Bit Rate)即固定码率，就是静态（恒定）比特率的意思，CBR 是一种固定采样率的压缩方式。CBR 编码指的就是编码器的输出码率应该是固定值。当在一个带宽受限的信道中进行多媒体通讯的时候 CBR 是非常有用的，因为这时候受限的是最高码率，CBR 可以更好的利用这样的信道。但是 CBR 不适合进行存储，因为 CBR 将导致没有足够的码率对复杂内容部分进行编码，从而导致质量下降，同时 CBR 编码也会在简单的图像部分浪费一些码率。当编码方案输出的是可变长码字时，编码核很难做到完美的 CBR，即使编码核可以通过调整量化参数来部分解决这个问题，但同时也影响了编码质量。简单讲，CBR 模式的优点是压缩快，能被大多数软件和设备支持，缺点是占用空间相对大，效果并不十分理想。

本文对 H.264/AVC 视频编码核分别对 CIF（352x288），SD（720x576）和 HD（1920x1088）三种视频序列进行了 CBR 模式下的帧/场编码测试，并将测得结果数据与标准 JM 编码器测得结果比对，如表 4.7、表 4.8 和表 4.9。

表 4.7 CBR 模式的 CIF(352x288)序列测试结果

序列	目标码率	JM(帧)		H.264/AVC (帧)		JM(场)		H.264/AVC (场)	
		psnr	bitrate	psnr	bitrate	psnr	bitrate	psnr	bitrate
CIF	0.5Mbps	32.43	1208	26.94	524	32	1932	23.98	515
	0.7Mbps	32.39	1206	28.54	731	32	1932	25.62	725
	0.9Mbps	32.9	1251	29.85	941	31.84	1895	26.58	1025
	1.1Mbps	33.86	1351	30.94	1143	31.69	1868	27.32	1102
	1.3Mbps	34.57	1447	31.59	1366	31.66	1855	29.24	1358
	1.5Mbps	35.67	1599	32.63	1577	32.28	1921	30.28	1586

根据表 4.7 的数据生成的 PSNR-Bitrate 对比如图 4.17、图 4.18 所示。

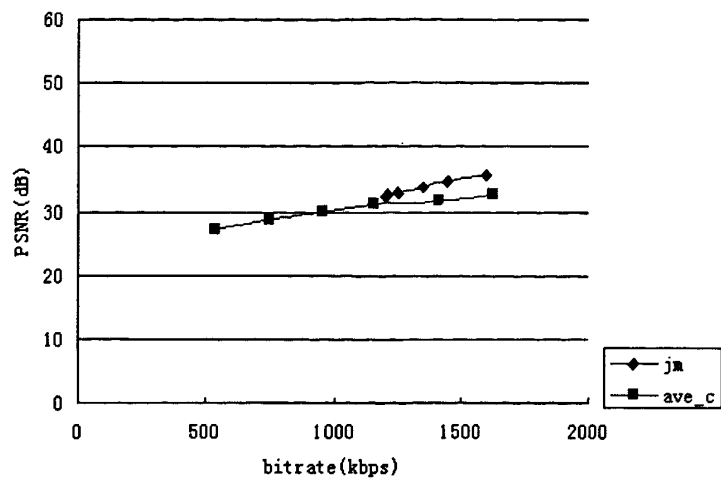


图 4.17 CBR 模式下的 CIF 序列帧模式测试

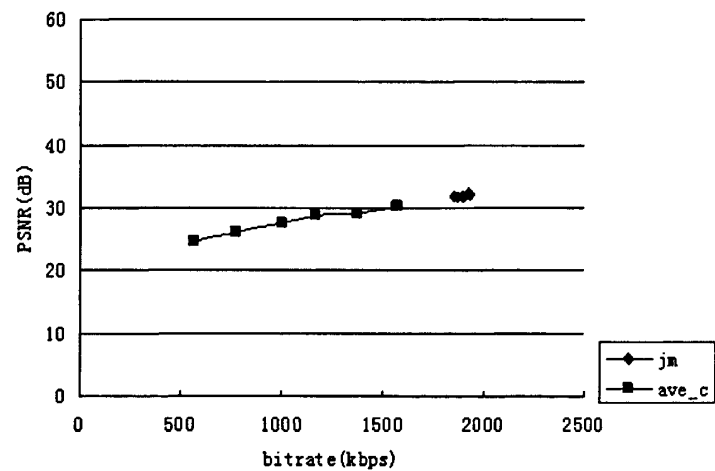


图 4.18 CBR 模式下的 CIF 序列场模式测试

表 4.8 CBR 模式的 SD(720x576)序列测试结果

序列	目标码率	JM(帧)		H.264/AVC(帧)		JM(场)		H.264/AVC(场)	
		psnr	bitrate	psnr	bitrate	psnr	bitrate	psnr	bitrate
S D	2Mbps	47.96	2186	46.2	1953	46.92	2139	45.44	1953
	2.8Mbps	48.73	3009	47.49	2760	47.96	2909	46.87	2757
	3.6Mbps	49.24	3842	48.26	3475	48.58	3685	47.68	3466
	4.4Mbps	49.71	4681	48.8	4276	49	4460	48.32	4420
	5.2Mbps	50.09	5524	49.65	5389	49.35	5249	48.91	5351
	6Mbps	50.46	6408	49.98	6219	49.62	6051	49.26	6182

根据表 4.8 的数据生成的 PSNR-Bitrate 对比如图 4.19、图 4.20 所示。

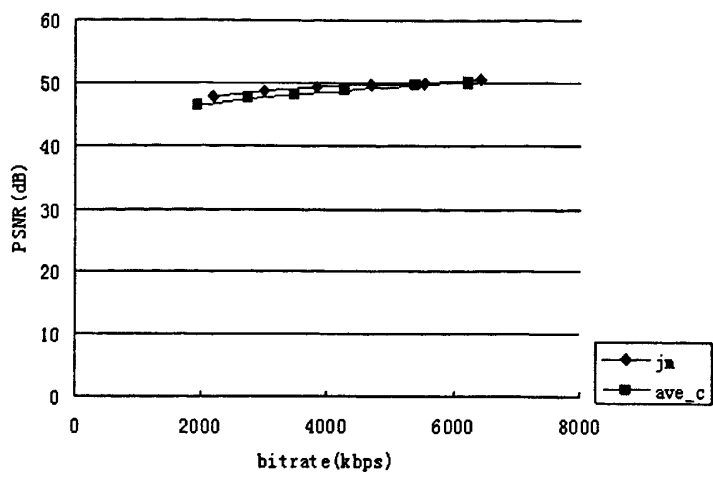


图 4.19 CBR 模式下的 SD 序列帧模式测试

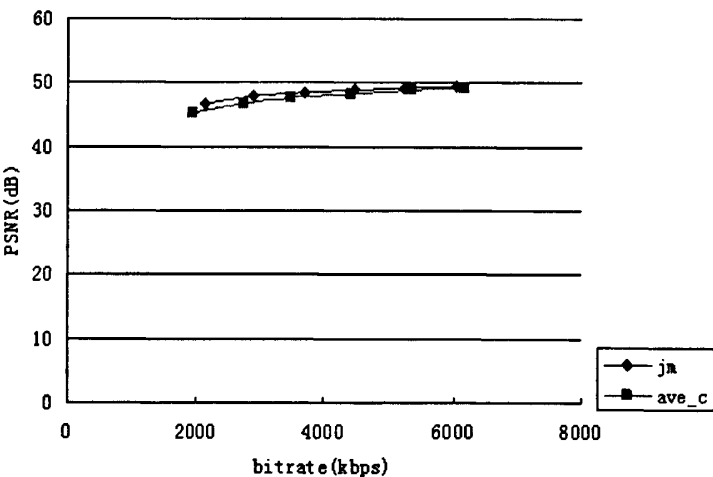


图 4.20 CBR 模式下的 SD 序列场模式测试

表 4.9 CBR 模式的 HD(1920x1088)序列测试结果

序列	目标码率	JM(帧)		H.264/AVC(帧)		JM(场)		H.264/AVC(场)	
		psnr	bitrate	psnr	bitrate	psnr	bitrate	psnr	bitrate
H D	10Mbps	42.97	11088	40.69	9993	40.69	15570	39.28	9780
	14Mbps	42.97	11088	40.69	9993	40.69	15570	39.28	14258
	18Mbps	44.3	14480	42.53	14428	41.48	16364	41.08	16535
	22Mbps	45.12	18009	43.68	18298	43.43	19462	41.75	22075
	26Mbps	45.87	21914	44.32	22007	44.48	22974	42.86	26137
	30Mbps	46.52	26290	44.76	27024	45.18	26541	44.03	29626

根据表 4.9 的数据生成的 PSNR-Bitrate 对比如图 4.21、图 4.22 所示。

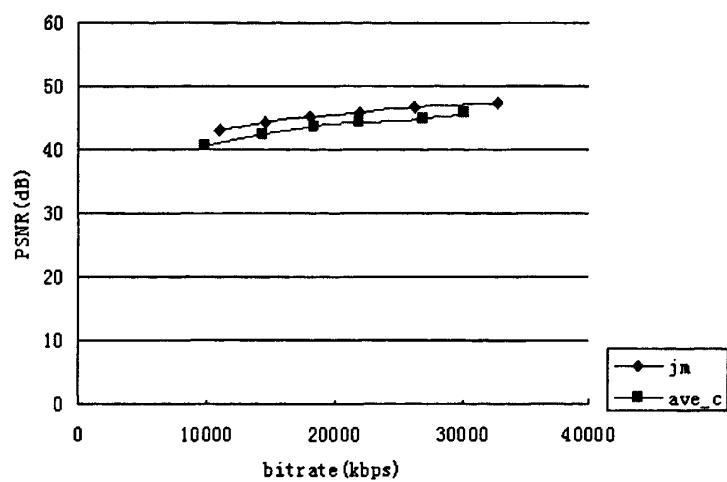


图 4.21 CBR 模式下的 HD 序列帧模式测试

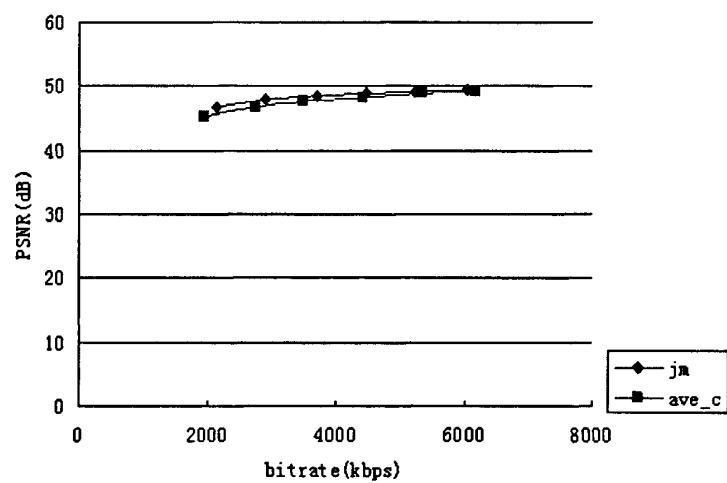


图 4.22 CBR 模式下的 HD 序列场模式测试

通过实验发现，在 CBR 模式下，H.264/AVC 视频编码核和 JM 的编码性能在大多数情况下相近。因为编码核对码率控制算法进行了改进，能更好地控制实际码率，使码率控制的精度在 10%之内，但 JM 的码率控制算法无法在低码率情况下正常工作，实际码速率与目标码速率有很大的差距。

1. VBR 模式

VBR(Variable Bit Rate)即动态比特率，也就是非固定的比特率，这是以质量为前提兼顾文件大小的编码方式。VBR 模式编码可以让视频图像的复杂部分用高比特率编码，简单部分用低比特率编码，通过这种动态调整编码速率的方式，进一步得到编码质量和文件体积之间的平衡。其优点是可使视频质量达到要求，但缺点是编码时无法估计压缩出来的文件体积大小。

本节用测试 CBR 同样的三种视频序列（CIF（352x288），SD（720x576）和

HD (1920x1088)) 对 H.264/AVC 视频编码核进行 VBR 模式的帧/场编码测试, 并将测得结果数据与标准 JM 编码器测得结果进行比对。

表 4.10 VBR 模式的 CIF(352x288)序列测试结果

序列	QP	JM(帧)		H.264/AVC (帧)		JM(场)		H.264/AVC (场)	
		psnr	bitrate	psnr	bitrate	psnr	bitrate	psnr	bitrate
C I F	15	47.19	6923	44.32	6729	47.16	8330	43.7	7735
	25	38.34	2257	36.42	2555	37.9	3028	35.27	3015
	30	33.94	1106	32.55	1453	33.24	1565	31.12	1653
	35	30.07	545	29.21	853	29.11	773	27.63	903
	40	26.72	275	26.24	469	25.59	374	24.68	480
	51	11.26	13	20.83	181	11.26	15	19.76	226

根据表 4.10 的数据生成的 PSNR-Bitrate 对比如图 4.23、图 4.24 所示。

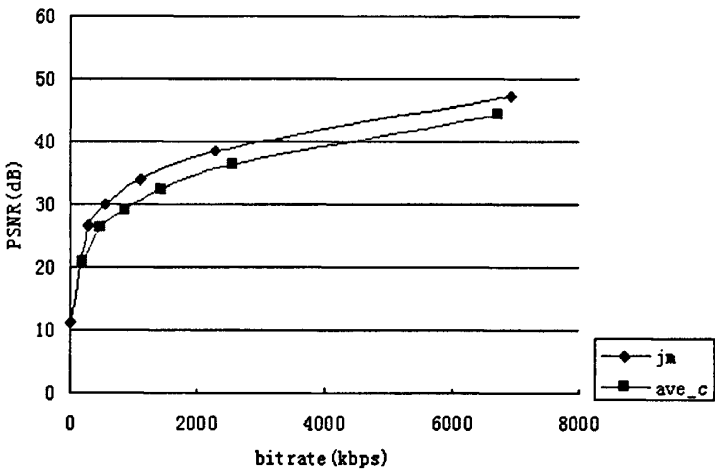


图 4.23 VBR 模式下的 CIF 序列帧模式测试

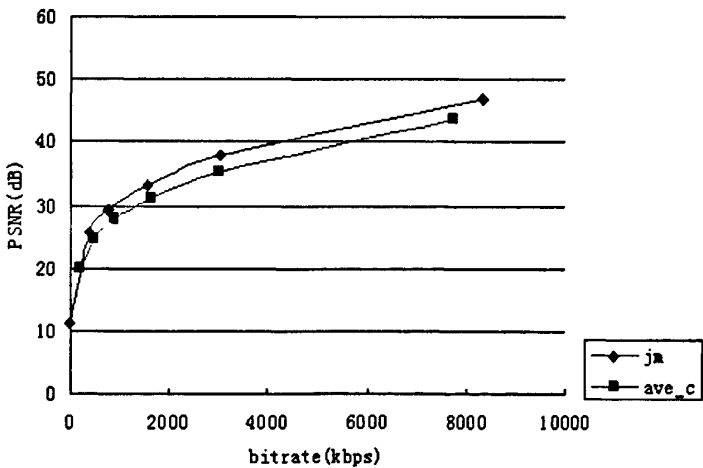


图 4.24 VBR 模式下的 CIF 序列场模式测试

表 4.11 VBR 模式的 SD(720x576)序列测试结果

序列	QP	JM(帧)		H.264/AVC (帧)		JM(场)		H.264/AVC (场)	
		psnr	bitrate	psnr	bitrate	psnr	bitrate	psnr	bitrate
S D	15	50.09	4305	48.82	3593	49.6	4890	48.12	3826
	25	45.55	837	44.72	1114	44.63	1026	43.46	1218
	30	42.47	396	42.04	746	41.3	497	40.57	819
	35	39.4	199	39.37	584	38.06	244	37.6	639
	40	36.46	106	36.77	507	35.23	125	35.1	562
	51	9.41	25	31.85	396	9.41	30	30.3	483

根据表 4.11 的数据生成的 PSNR-Bitrate 对比如图 4.25、图 4.26 所示。

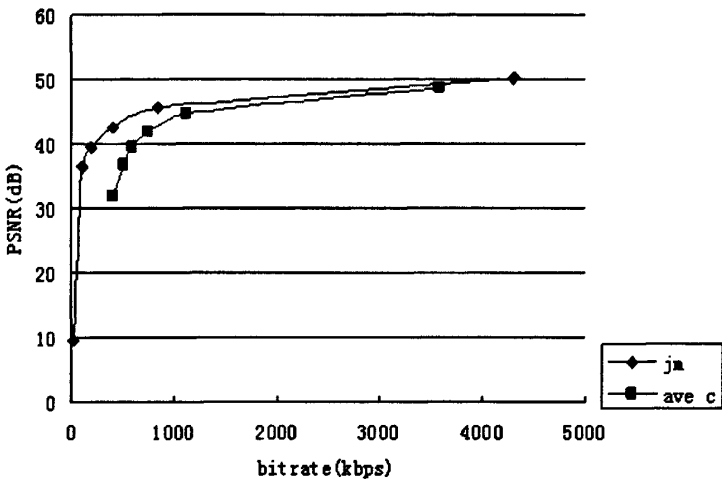


图 4.25 VBR 模式下的 SD 序列帧模式测试

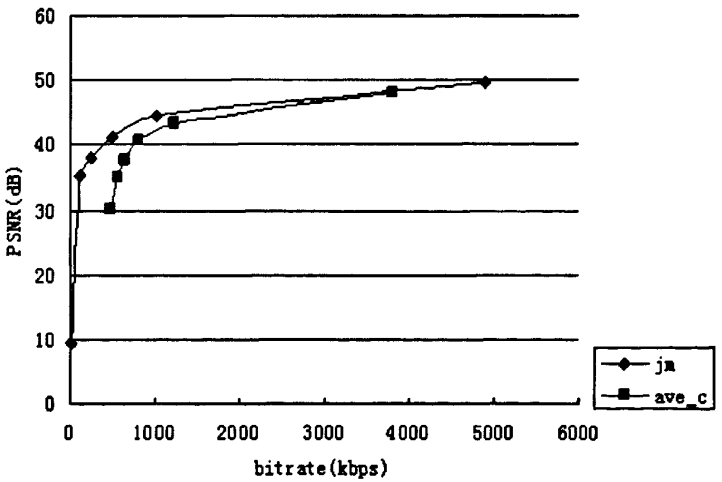


图 4.26 VBR 模式下的 SD 序列场模式测试

表 4.12 VBR 模式的 HD(1920x1088)序列测试结果

序列	QP	JM(帧)		H.264/AVC (帧)		JM(场)		H.264/AVC (场)	
		psnr	bitrate	psnr	bitrate	psnr	bitrate	psnr	bitrate
H D	15	49.28	49989	48.73	47958	48.71	56151	47.55	48550
	25	43.64	11607	43.02	14104	42.76	12913	41.6	13913
	30	41.04	5687	39.92	8085	39.94	6338	38.72	7990
	35	38.5	3141	36.79	5340	37.25	3439	35.95	5301
	40	35.88	1835	33.75	4011	34.5	1974	33.17	4060
	51	15.27	443	29.83	3180	15.27	508	29.09	2969

根据表 4.12 的数据生成的 PSNR-Bitrate 对比如图 4.27、图 4.28 所示。

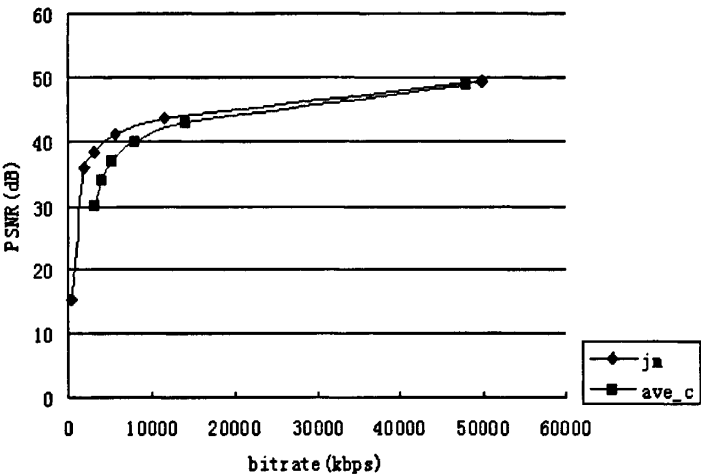


图 4.27 VBR 模式下的 HD 序列帧模式测试

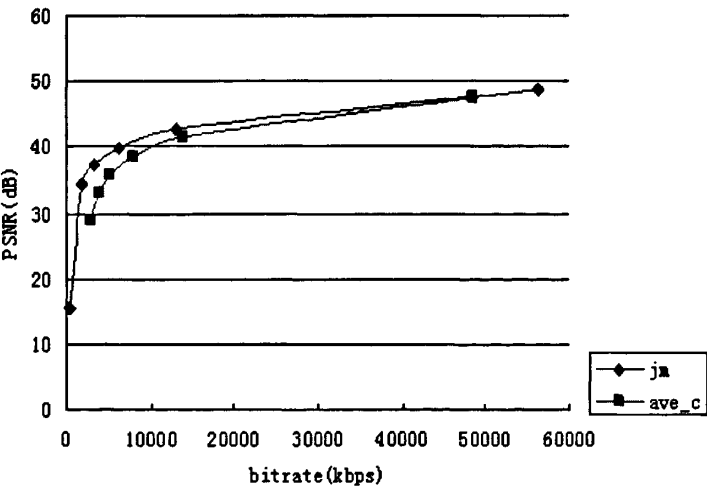


图 4.28 VBR 模式下的 HD 序列场模式测试

通过实验发现，在高码率时，H.264/AVC 视频编码核的编码性能与 JM 的编码性能基本相同；而在低码率时，JM 的编码性能稍优于编码核的编码性能，但是由于 JM 在低码率的时候采用了过多了 skip 模式和 direct 模式，所以导致 JM 的主观

视频质量相对于编码核大幅度下降。

第五章 总结与展望

设计验证是电子设计人员目前面临的主要挑战,无论是设计团队还是验证团队,都将超过 50%的时间用在纠错上,因此这一领域的技术进展将对缩短产品上市时间产生重大影响。功能性错误是造成设计重复修改的首要原因。用于查找这些错误的功能验证流程是设计流中目前面临的瓶颈。然而,验证技术的发展步伐已经落后于设计和制造能力,验证鸿沟在进一步扩大。这一验证鸿沟是限制设计人员充分发挥其生产力和设计能力的因素。为了弥合这一验证鸿沟,验证必须成为整体设计方法的一个内在组成部分。验证的目的是为了发现设计中的 Bug,而从芯片的整个设计流程来看,Bug 发现的越早越好修复,对芯片的影响也越小。因为要修改错误可能要一直追溯到芯片规范文档中,这样如果在物理验证时候发现的错误可能会造成整个芯片进行大的修改,从而可能导致设计流程重新进行一遍。在整个流程中越早验证 Bug 发现的就越多,而同时修复 Bug 的花费则是越早花费越少。

设计验证在设计周期中占了最大的一部分,这是人所共知的事实。然而,面对如此众多用于缓解该问题的现有技术,我们真正应该采用的是哪一项或哪几项?得到的答案通常并不简单明了,而且往往令人感觉含混不清和成本昂贵。虽然有些工具可以起到作用,但真正的问题是再造方法学,可是由于业界近几年并没有提出真正的新的验证方法。由于各种原因,对本文的课题研究还存在许多不足,例如很多时候还得依靠示波器和逻辑分析仪进行 FPGA 原型验证,更有一些还得依靠国外的工具来进行协助验证。在 FPGA 原型验证的软/硬件协同验证等方面还需更深入地研究,这一些也都是我今后研究工作的方向。

由于作者水平有限,存在很多不足之处,希望本论文的读者能够批评并予以指正,以便在以后的研究中予以改进和提高。

致 谢

在论文即将完成之际，谨向给予我关心和帮助的老师、同学、朋友及亲人表示最衷心的感谢！

衷心感谢我的导师陶海红教授！她在近三年时间里一直帮助我教导我，在学业上，生活上以及思想上都给予我巨大的帮助。她严谨的治学态度，谦逊的品德，渊博的知识使我终生受益，她无私的关怀使我度过了研究生的学习生活。特别是在本次论文写作中，给予了我非常细致而又耐心的指导。我在此向陶海红老师无私的教诲表示衷心的感谢。

感谢田泽研究员，感谢他给我提供了一个难得的学习环境，使我各个方面都得到了很大的提高，并感谢他对我的关心和指导。田老师扎实的理论功底、严谨的治学态度和活跃的创新思想使我受益匪浅。从田老师身上，我不仅学到了丰富的知识，更学到了治学的方法，这些收获，必将影响我的一生。

感谢我们项目组的田靖、安博峰、马超、刘练、刘航、张玢、王婷、刘宁宁等，在整个项目的设计和开发过程中，他们与我共同探讨，解决了很多问题，并且给予了我大量的帮助。

特别感谢黎小玉和王琦卉师姐，在我困惑的时候给我很多建设性的意见，以她们丰富的经验帮助我学业进步。

感谢我们宿舍的刘影、郑奔月、胡蓉在近三年的研究生学习生活中，她们给了我很多关心和帮助，度过了美好的学习时光。

深深感谢我的父母，他们无私的爱和深切的期望是我人生中强大的精神动力！

衷心感谢在百忙之中评阅论文和参加答辩的各位专家、教授！

向所有关心、帮助我的人致谢！

参考文献

- [1] 毕厚杰. 新一代视频压缩编码标准—H.264/AVC. 北京: 人民邮电出版社, 2005.
- [2] (英)Iain E.G. Richardson. H.264 和MPEG-4 视频压缩. 长沙: 国防科技大学出版社, 2005.
- [3] 田泽. 嵌入式系统开发与应用. 北京: 北京航空航天大学出版社, 2004.
- [4] H.264/MPEG-4 Part 10 White Paper. 2003.
- [5] Yao Wang, Jorn Ostermann and Ya-Qin Zhang. 视频信号处理与通信. 北京: 清华大学出版社, 2003.
- [6] JVT&ITU-T VCEG. Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification H.264(MPEG-4 Part10), 7-14 March, 2003.
- [7] 邓磊, 高文, 互铭曾等. AVC/H.264 标准中高效运动估计硬件结构设计, 高技术通讯, 2006, 16(10).
- [8] I TU -T H.241. Extended video procedures and control signals for H.300 series terminals. July 2003.
- [9] AD Inc. Blackfin processor instruction set reference.Rev 3.0. June 2004.
- [10] 熊明霞, 雷宏, 马小兵. FPGA片上PowerPC系统设计, 计算机测量与控制 2007.
- [11] 段然, 梁婕. SoC系统验证方法研究, 航天控制 2009.
- [12] 成运. 实时H.264 关键编码算法研究与实现, 国防科学技术大学硕士学位论文, 2006.
- [13] Wiegand Thomas, Sullivan Gary J, Bjontegaard Gisle,etal. Overview of the H.264/AVC Video coding standard.IEEE Transactions on Circuits and Systems for Video Technology, 2003, 13(7).
- [14] 方健. 新一代视频压缩标准算法和研究. 浙江大学博士学位论文, 2008.
- [15] 李军克. 基于FPGA的SoC/IP验证平台的设计与实现. 哈尔滨工业大学硕士学位论文, 2006.
- [16] 刘志刚. 基于FPGA的H.264 编码器的硬件的实现. 西安电子科技大学硕士学位论文, 2009.
- [17] 淮治华. 基于VxWorks的SoC协同设计与验证技术研究与应用. 西北大学硕士学位论文, 2009.
- [18] 郭慰萱. 基于DM642 的实时网络视频传输系统的研究与实现. 西安电子科技大学硕士学位论文, 2009.

-
- [19] 柳书涛. 基于H.264 视频加密算法的研究. 西安电子科技大学硕士学位论文, 2007.
- [20] 祁晓龙. 一种面向嵌入式的H.264/AVC快速运动估计算法. 西安电子科技大学硕士学位论文, 2007.
- [21] 姚晋鹏. SoC软硬件协同验证平台监控系统设计与实现及综合测试. 电子科技大学硕士学位论文, 2006.
- [22] 蒋成明. 基于H.264 的嵌入式视频采集系统的研究与应用. 江苏大学硕士学位论文, 2007.
- [23] 葛保建. 基于SOPC的软硬件协同设计平台的研究与实现. 武汉科技大学硕士学位论文, 2008.
- [24] Alexis M.Tourapis, Oscar C.Au,Ming L.Liou. Predictive motion vector field adaptive search technique (PMVFAST) - Enhancing block based motion estimation.In proceedings of Visual Communications and Image Processing 2001(VCIP-2001). San Jose, CA, Jan.2001.
- [25] 刘向阳, 方芳. 新的视频压缩编码标准H.264. 广播电视信息, 2005.
- [26] 胡兴军. 视频编码标准H.264 的技术革新及应用. 电子数字成像, 2009.
- [27] 葛先雷. 数字电视H.264 编码技术分析及应用展望. 山西电子技术, 2009.
- [28] 孙毅, 林涛, 林争辉. 新一代视频编码标准H.264/AVC的特点及应用. 广播电视信息, 2006.
- [29] Yu-Wen Huang, Bing-Yu Hsieh, et al. Analysis and reduction of reference frames for motion estimation in MPEG-4 AVC/JVT/H.264. ICASSP'03, Vol.3.
- [30] 杨洋, 宋锐, 吴成柯, 高玉娥, 张磊. H.264 硬件编码器设计, 电视技术, 2007.
- [31] MohammedE.AI-Mualla, Nishan Canagarajah and David R.Bull. Simplex minimization for multiple-reference motion estimation. ISCAS 2000.
- [32] 裴世保, 李厚强, 俞能海. 一种快速的H.264/AVC帧内预测模式选择算法. 计算机工程与应用, 2005 年, 第 10 期.

作者在读期间的科研工作及研究成果

在硕士研究生期间取得的研究成果如下：

一、 参加科研情况

1. 中航集团某研究所的“HK1553”的PCIMB板卡的驱动和可视化程序的开发。
2. 中航集团某研究所的“H.264/AVC音视频编码芯片”的开发。

H. 264/AVC视频编码核的验证技术研究

作者：[陈佳](#)
学位授予单位：[西安电子科技大学](#)

本文链接：http://d.g.wanfangdata.com.cn/Thesis_Y1865859.aspx