

# 一种高效多标准视频解码器架构研究与设计

刘慧超<sup>1,2</sup>, 王志君<sup>1</sup>, 梁利平<sup>1†</sup>

(1. 中国科学院微电子研究所, 北京 100029; 2. 中国科学院大学, 北京 100049)

**摘要:**针对目前视频解码器实现方案存在的灵活度低、开发周期长、不能适应快速变化的算法升级等问题,提出一种面向多种视频编解码标准的通用视频解码器架构设计方案.采用软硬件协同设计方法,基于可编程同构多核处理器+协处理器的硬件架构,同构多核处理器采用指令级和任务级并行加速,协处理器采用硬件定制单元实现矢量加速,同时利用分布式片上便笺式存储器(Scratchpad Memory, SPM)代替数据 Cache 实现高效的数据存储系统,以应用广泛的 H.264 视频标准为验证实例.实验结果表明,基于本文所提架构实现的 H.264 视频解码器高效可行,平均并行加速比为 9.12,相比于传统多核并行解码算法提高了 1.31 倍.

**关键词:**多标准;视频解码器;可编程;协处理器;便笺存储器;H.264 解码器;架构设计

**中图分类号:**TN919.3

**文献标志码:**A

## Research and Design of an Efficient Multi-standard Video Decoder Architecture

LIU Huichao<sup>1,2</sup>, WANG Zhijun<sup>1</sup>, LIANG Liping<sup>1†</sup>

(1. Institute of Microelectronics, Chinese Academy of Sciences, Beijing 100029, China;

2. University of Chinese Academy of Sciences, Beijing 100049, China)

**Abstract:** Aiming at the problems such as low flexibility, long development cycle, and incapability of adapting to rapidly changing algorithms for current video decoder implementation solutions, a generic video decoder architecture design scheme for multiple video coding and decoding standards is proposed using software and hardware collaboration. The methodology of the design is based on a programmable homogeneous multi-core processor and coprocessor hardware architecture. The homogeneous multi-core processor uses instruction-level and task-level parallel acceleration. The coprocessor uses a hardware customization unit to achieve the vector acceleration, while it uses distributed on-chip scratchpad memory instead of data cache to achieve an efficient data storage system. Taking the H.264 video standard widely used as an example, the experimental results show that the H.264 video decoder based on the architecture proposed in this paper is highly efficient and feasible, with an average speed-up of 9.12, which is 1.31 times better than the traditional multi-core parallel decoding algorithm.

**Key words:** multi-standard; video decoder; programmability; coprocessor; scratchpad memory; H.264 decoder; architectural design

\* 收稿日期:2018-11-10

基金项目:中国科学院知识创新项目(KGCX2-YW-134), Chinese Academy of Sciences Knowledge Innovation Project of China (KGCX2-YW-134)

作者简介:刘慧超(1991—),男,河南长垣人,中国科学院微电子研究所博士研究生

† 通讯联系人, E-mail: liangliping@ime.ac.cn

视觉是人们感知和认知外部世界的主要途径. 实验心理学家赤瑞特拉通过大量实验证实人类获取信息的 80% 以上都是来自视觉<sup>[1]</sup>. 信息化时代下, 与视觉相关的应用广泛地存在于通信、多媒体消费、安防监控、抗震救灾等领域. 视频应用的一项关键技术就是视频编解码, 自 20 世纪 80 年代开始, 先后出现了一系列国际化视频编解码标准, 如 H.26x 标准、MPEG-x 标准、H.264/AVC 标准、AVS 标准以及最新的 HEVC 标准等. 针对不同的视频标准, 不同应用环境下如何快速实现一个实时的视频解码器成为研究热点. 目前比较常见的实现方案有: 1) 基于 ASIC 专用集成芯片实现<sup>[2-3]</sup>方案, 已商业化的有富士通的 MB91696AM、Sigma Designs 的 SMP8630 和国产代表华为海思的 Hi3535, 虽性能稳定、功耗低, 但研发周期长、灵活度低, 当算法升级或变化时不能及时修正; 2) 基于 DSP 处理器实现<sup>[4-5]</sup>方案, 以 TI 的 DM64x 系列为代表, 还有 Philips 的 PNX1500 系列、ADI 的 Blackfin 处理器, 虽功能强大、性能可靠, 但功耗较大、启动速度慢; 3) 基于 FPGA 硬件实现<sup>[6-7]</sup>方案, 通过自主设计或购买商业 IP 实现. 首先商业 IP 费用高、灵活度差且关键模块为黑盒、调试风险大, 核心技术受制于人. 自主研发 IP 相对灵活、可定制且代码可见, 但实现整个解码系统任务复杂, 且稳定性有待测试、验证任务繁重; 4) 基于 CPU+GPU 的系统实现<sup>[8-9]</sup>方案, 需要使用高级编程语言, 门槛高、线程调度复杂、同步开销大. 近年来, 随着集成电路技术的快速发展, 开始出现基于多核处理器和 SoC 架构实现的视频解码器<sup>[10-13]</sup>.

本文通过分析不同视频标准间的共性, 创新性地提出了一种基于同构多核处理器+协处理器的通用视频解码器架构, 并以 H.264 视频解码器为实例进行验证. 采用媒体运算指令并行加速和硬件协处理器矢量加速, 实验结果表明本文所提架构可行有效, 便于软硬件升级.

## 1 多标准视频编解码算法分析

### 1.1 视频编解码标准发展历程

视频编解码标准自 20 世纪 80 年代起至今, 不断完善和发展. 自 H.261 标准开始, 普遍采用包括基于运动补偿的帧间预测、离散余弦变换 (Discrete Cosine Transform, DCT)、量化、zig-zag 扫描和熵编码等编码方法. 这些技术组合在一起形成了沿用至今的混合编码框架. 图 1 为混合编码器核心结构框图.

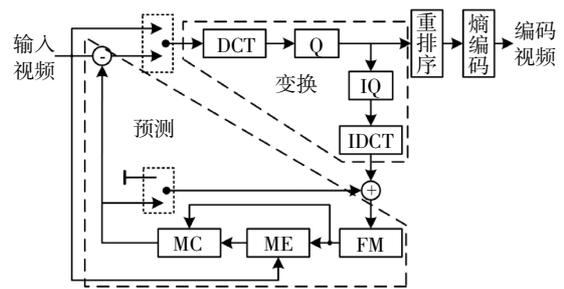


图 1 混合编码器框图

Fig.1 The block diagram of hybrid encoder

混合编码架构主要包括预测编码和变换编码两大部分. 其中, 变换编码包括 DCT 和 IDCT (Inverse Discrete Cosine Transform, 反离散余弦变换)、Q (Quantization, 量化) 和 IQ (Inverse Quantization, 反量化). 输入视频后的减法器、运动估计 (Motion Estimation, ME) 和运动补偿 (Motion Compensation, MC) 属于预测编码部分. 编码视频输出前的熵编码是一种变长编码, 可进一步提高混合编码的压缩效率. 重排序过程使得量化后的 DCT 非零系数集中, 减少统计事件的个数, 进一步增加熵编码的压缩率.

### 1.2 不同标准编解码算法对比分析

本文以最新的 AVS2、H.264 和 HEVC 3 种视频标准为例, 重点分析对应视频解码算法在结构上的共性和差异. 整体上, 3 种标准均采用混合解码架构, 如图 2 所示, 解码器包括熵解码、重排序、IQ、IDCT、帧间预测、帧内预测和环路滤波, 但各模块内部算法细节却存在较大差异.

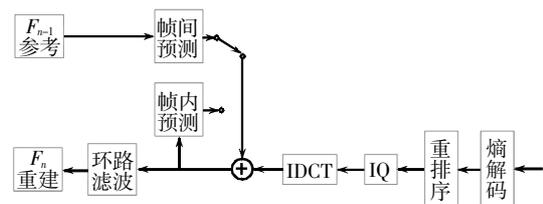


图 2 混合解码器框图

Fig.2 The block diagram of hybrid decoder

#### 1.2.1 帧间预测

AVS2、H.264 和 HEVC 标准都支持变块大小的运动补偿和四分之一精度像素插值, 但插值使用的滤波器阶数和滤波系数均不相同. 此外, 运动矢量 (Motion Vector, MV) 虽均由相邻块 MV 预测而来, 但具体的预测策略也各不相同.

首先, AVS2 和 HEVC 的分像素插值算法类似. 亮度分量, HEVC 对半像素位置采用 8 抽头滤波器, 1/4 和 3/4 像素采用 7 抽头滤波器; AVS2 统一采用

8 抽头滤波器,只是分像素位置不同,对应滤波系数不同.对色度分量而言,两者均采用 4 抽头滤波器,且根据分像素位置采用不同的滤波系数.具体如表 1 所示.

表 1 亮度插值滤波器系数比较

Tab.1 Comparison of luminance interpolation filter coefficients

| 亚像素位置 | 视频标准 | 滤波器系数                     |
|-------|------|---------------------------|
| 1/4   | AVS2 | {-1,4,-10,57,19,-7,3,-1}  |
|       | HEVC | {-1,4,-10,58,17,-5,1}     |
| 1/2   | AVS2 | {-1,4,-11,40,40,-11,4,-1} |
|       | HEVC | {-1,4,-11,40,40,-11,4,-1} |
| 3/4   | AVS2 | {-1,3,-7,19,57,-10,4,-1}  |
|       | HEVC | {1,-5,17,58,-10,4,-1}     |

与 AVS2 和 HEVC 相比,H.264 标准对亮度分量半像素点采用 6 抽头滤波器插值得出,滤波系数为 {1,-5,20,20,-5,1},1/4 和 3/4 像素点则利用相邻整像素和/或分像素的线性内插得出,计算量较低.相应地,色度分量通过相邻整像素的双线性内插得出.

3 种视频标准虽然采用的分像素插值算法不同,但都可以归结为像素点积和均值计算两种算子,有利于采用高性能乘累加指令编程实现.

关于运动矢量预测算法,由于策略不同,所以这部分需针对特定标准采用软件或硬件方式独立实现.

### 1.2.2 帧内预测

帧内预测充分利用了帧内像素的空间相关性,通过邻近已编码或解码块的重建值进行预测.H.264 帧内预测算法,4×4 和 16×16 亮度块分别拥有 9 种、4 种预测模式,色度分量以 8×8 为预测单元,拥有 4 种预测模式.AVS2 和 HEVC 与 H.264 类似,但比 H.264 预测方向更精细更灵活,分别为亮度预测块定义了 33 种和 35 种不同的预测模式,色度预测块均定义了 5 种预测模式.但从算法结构来看,虽然 3 种视频标准预测模式总数不同,但可分为以下 3 类: Planar 模式、DC 模式和角度模式.

### 1.2.3 环路滤波

视频压缩编码是一种有损压缩,编码中的量化和计算误差会给重建后的图像造成不可恢复的失真.因此自 H.263 和 MPEG-2 标准开始,在视频编

解码器中均加入了环路滤波处理,以改善图像质量.然而,不同视频标准中的环路滤波算法又存在些许差别.以 H.264 为例,采用自适应去块效应滤波器,对所有的 4×4 块边界和样点值进行边界强度判断和自适应滤波处理,整个过程几乎涉及重建图像所有像素点,计算复杂度高.

AVS2 和 HEVC 均相对于 H.264 环路滤波技术做了精简,主要体现在以下 3 个方面:

1) 亮度和色度分量均以 8×8 块为单位进行边界滤波,相比于 H.264,滤波边界数大大减少;

2) 边界滤波时,每条边界两边最多各修正 3 个像素点,使得 8×8 块之间相互独立,可以并行处理,而 H.264 只能串行操作;

3) 可以先对整幅图像进行垂直边界滤波,然后进行水平边界滤波,而 H.264 必须以宏块为单位交替执行垂直边界滤波和水平边界滤波.

此外,在去块效应滤波器后,AVS2 和 HEVC 又增加了不同的补偿技术,AVS2 引入了自适应样点偏移和样本补偿滤波,而 HEVC 则引入了样点自适应补偿技术,用于抑制 DCT 变换后高频交流系数量化失真造成的振铃效应.

### 1.2.4 熵解码

预测+变换编码消除信源的时间和空间冗余,经量化处理后得到预测残差变换系数,连同其他控制信息和标识信息等句法元素,利用熵编码方法进一步压缩,消除码字之间的冗余.H.264、AVS2 和 HEVC 分别采用了不同的熵编码策略,如表 2 所示.

表 2 不同视频标准熵编码算法对比

Tab.2 Comparison of entropy coding algorithms in different video standards

| 熵编码算法   | H.264 | AVS2 | HEVC |
|---------|-------|------|------|
| CAVLC   | 支持    | 支持   | 不支持  |
| CABAC   | 支持    | 支持   | 支持   |
| 指数哥伦布编码 | 支持    | 支持   | 支持   |

熵编解码算法基于信息熵原理实现数据的无损编解码,算法按照信息比特完成运算,计算粒度最小,分支复杂.通用处理器运算粒度为字或字节,基于通用处理器实现的熵编解码算法无论速度和功耗均不具有优势.以熵解码过程为例,目前的研究主要分两种途径实现:其一,实现完整的句法解析过程<sup>[14]</sup>;其二,仅实现单纯的熵解码单元,通过调用解

码单元实现句法解析<sup>[15]</sup>. 采用第2种途径时,单一句法元素的解析由硬件完成,而码流结构的多分支控制由主控方实现,影响主控方的执行性能. 相反,采用第1种途径时,码流结构控制和句法元素解析均由硬件实现,缓解主控方压力,同时不会带来巨大的硬件开销.

总结不同视频编解码算法特点,有以下共性:1)统一的混合编解码架构,功能模块和输入输出在算法级是一致的,这为采用统一的视频解码体系架构提供了基础. 2)不同功能模块内部细节差异较大,但计算过程具有很大相似性,如分像素插值、帧内预测和环路滤波,可由高性能乘累加单元编程实现. 以上特点为本文解码器架构提供了基础.

## 2 多标准视频解码器体系架构

### 2.1 流水线的视频解码器架构

#### 2.1.1 流水线任务划分

如前文所述,不同视频标准对解码系统的可编程性提出了很高的要求. 本文结合视频解码器特点,提出一种多标准通用的多核流水解码体系架构. 系统由两部分组成:专用协处理器 ECore 和高性能同构4核处理器 Core0~Core3. 采用软硬件协同的设计方法,由专用协处理器实现熵解码等计算粒度小、计算不规整、分支复杂的解析过程,高性能同构4核处理器实现帧间预测、帧内预测和环路滤波等计算密集、计算规整的解码过程. 如图3所示. 混合解码结构中,变换解码完成对残差系数的反量化和反变换,得到像素残差;预测解码通过帧间或帧内预测算法计算得到像素预测值,两个过程相互独立. 预测值和残差叠加得到像素重建值. 考虑到当前块进行帧内预测时需要参考已解码相邻块像素,为避免流水线出现反馈环路,将帧内预测与 IQ、IDCT 和重建映射到流水线的同一级.

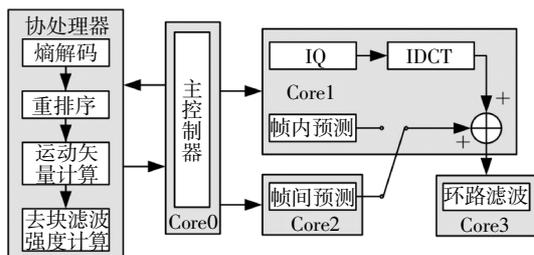


图3 多核视频解码器系统架构

Fig.3 Multi-core video decoder system architecture

#### 2.1.2 存储结构

对于多媒体流数据的处理面临两个关键问题:其一,流数据吞吐量;其二,流数据时间局部性差,每个元素的生命周期短. 传统多核解码器采用共享存储器和片上私有 Cache 的两级存储结构,共享存储器结构使得多核之间访存冲突严重,高速缓存 Cache 利用局部性原理实现快速访存,恰恰与流数据局部性差的特点相违背,导致 Cache miss 严重,不仅影响整个系统的吞吐率,而且会造成大量的功耗损失. 本文采用分布式片上便笺存储器作为数据缓存单元,如图4所示,多核处理器均配置一块支持多核并行访问不同存储体的片上 SPM 存储器,开辟乒乓结构实现核与核之间数据通信,有效解决传统多核解码器对共享存储器的访问冲突,降低 Cache miss 造成的性能损失,提高多核解码效率.

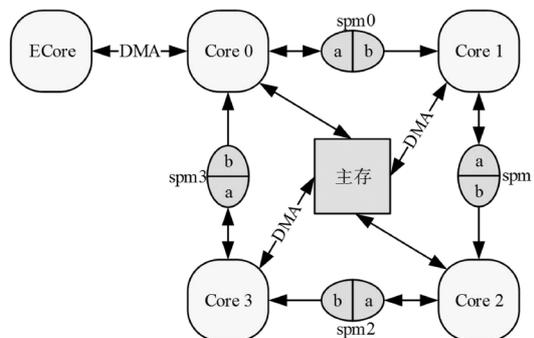


图4 流水线视频解码器存储架构

Fig.4 Storage architecture of pipelined video decoder

此外,视频解码器中,帧间预测和环路滤波两部分需要频繁的与外部存储器进行数据通信. 其中,帧间预测过程需要读取片外参考帧像素进行插值运算,环路滤波完成后需要将滤波结果写回到位于片外存储器的解码帧缓存空间. 本文采用基于宏块组(此处的宏块定义是针对 H.264 标准的,与 HEVC 和 AVS2 中的编码单元 CU 是一个概念)滤波结果写回与参考帧预取,并由 DMA(Direct Memory Access, DMA)方式实现快速访存.

1)基于宏块组滤波结果写回:文中采用基于宏块组的功能并行解码算法,宏块组的大小根据解码视频分辨率进行选取. 选取依据是:根据对应分辨率下视频帧每行包含的宏块数,选择其对应的一个因子数作为宏块组的大小. 当前宏块组解码完成后,可以按照二维结构顺序写回帧缓存空间,而不用考虑跨行存储问题,整个宏块组只需启动3次DMA搬运过程(亮度和色度分开存储),相比于以宏块为单位写回,可以大大降低DMA调用次数. 同时,基于

宏块组的存储策略,每次只需将当前解码完成的宏块组最右边界拷贝到下一个待解码宏块组的左边界上,相比于传统的基于宏块级并行解码,可以有效避免宏块组内相邻宏块边界数据的拷贝过程,避免了基于宏块行或更高级别并行对存储器资源占用率高的劣势。

2)参考帧预取:采用窗口式并行读取方式,降低访问片外存储器的频次.根据宏块不同预测模式,找出宏块内所有分割块参考数据在参考帧中水平和垂直方向的最大最小坐标位置,确定宏块的参考数据范围,如图 5 所示。

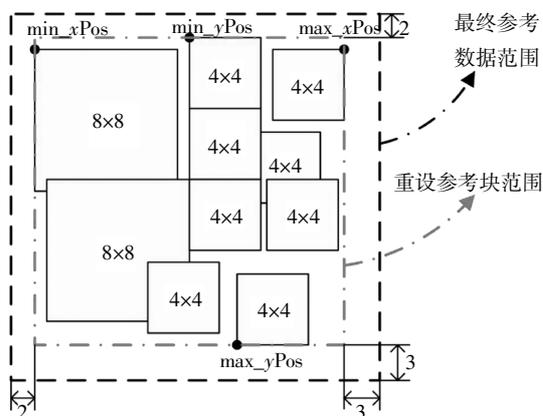


图 5 基于窗口式的参考像素预取  
Fig.5 Window-based reference pixel prefetch

以 H.264 标准为例,亮度分量读取范围为  $(min\_xPos-2, min\_yPos-2) \sim (max\_xPos+3, max\_yPos+3)$ ,色度分量读取范围为  $(min\_x/2, min\_y/2) \sim (max\_x/2+1, max\_y/2+1)$ 。同时,采用乒乓双缓存单元,当前宏块帧间预测与下一个宏块参考数据预取并行执行,最大限度降低帧间预测模块的数据等待。

### 2.1.3 流水线同步与负载平衡

为保证同构多核处理器与协处理器之间相互协调工作,进一步减少同步等待造成的周期损耗,设计实现环形队列和乒乓缓存单元.如图 6 所示,首先在 Core0 片上 SPM 空间开辟一个长度为 32 个单元的环形队列 loopbuf0,用于接收来自协处理器解码后的中间数据,Core0 将接收到的数据“按需分配”打包发送到 3 个从核各自的环形队列 loopbuf1、loopbuf2 和 loopbuf3 上.当 loopbuf0 满时,Core0 不再响应协处理器的读请求,此时协处理器需等待;同理,当 loopbuf1、loopbuf2 或 loopbuf3 任意一个为满时,Core0 需等待,只是,由于 Core3 处于流水线末端,所以 Core0 只需判断 loopbuf3 的状态是否为满即可。

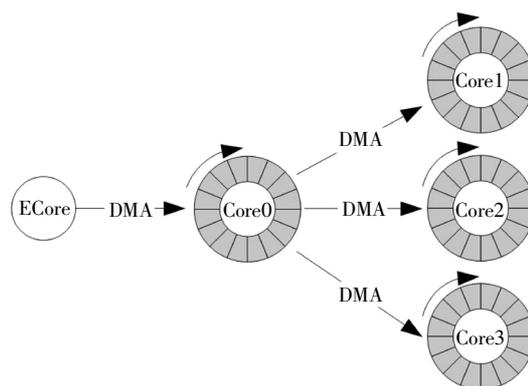


图 6 核间同步环形队列  
Fig.6 Loop queue for inter-core synchronization

除环形队列外,负责解码的 3 个从核之间利用上一节所述的乒乓缓存单元进行数据通信.为实现同一时刻相邻 2 个核不同时访问一个缓存单元,文中每个核均维护一个选择信号 bufsel,值为 0 对应 a 单元,为 1 对应 b 单元,bufsel 的值也会第一时间更新到对应核的邮箱中.当某个核开始对下一个缓存单元解码前,只需判断当前选择信号与相邻核选择信号是否一致,若不一致则继续,否则等待。

基于以上两种同步机制,可以很好地实现核与核之间的协同工作。

## 2.2 协处理器设计

本文对熵解码、反扫描和运动矢量与滤波强度计算采用硬件定制的协处理器进行加速,如图 7 所示。

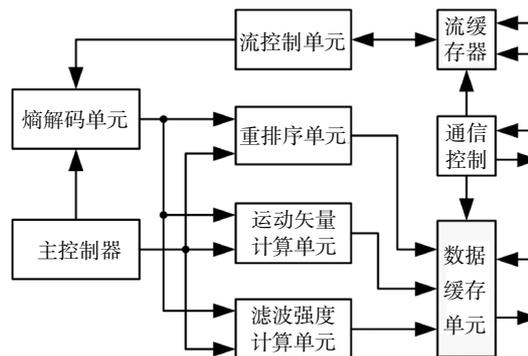


图 7 协处理器结构图  
Fig.7 Coprocessor structure diagram

协处理器除包括熵解码、反扫描等功能单元外,还有通信控制单元、码流读写控制单元和码流缓存单元和解码结果缓存单元,实现与核 0 的数据交互.采用 FPGA 实现的协处理器具有 3 点优势:1) 可以快速地实现细粒度运算;2) 相比于通用处理器的分支预测,可以高效地实现分支复杂的运算过程;3) 对

于不同视频标准中熵编解码过程的差异,可以很容易实现协处理器在线升级.关于这部分的具体实现不在本文所述范围内.本文中,协处理器在FPGA上实现,工作频率为50 MHz,可以满足H.264标准1080P解码要求.

### 2.3 多核调度

本文采用一种简单易实现的调度方法.首先,核0负责解码器的初始化,然后协处理器开始工作,此时核1~核3均处于休眠状态.然后,利用前文所述的环形队列结构,当核0用于接收协处理器解码数据的环形队列非空时,通过多核中断顺序启动核1~核3.之后,核间按照前文所述的流水线同步机制开始协同解码,不再需要核0干预.

## 3 实例验证

本文以应用广泛的H.264视频解码器为例,对提出的通用视频解码器架构进行了实例验证.验证平台采用中国科学院微电子研究所完全自主研发设计的IME-Diamond同构多核处理器<sup>[16-17]</sup>和Altera公司的型号为Cyclone IV EP4CE115F23I7的FPGA芯片.IME-Diamond是一款高性能32位嵌入式定点处理器,具有CPU和DSP功能融合性架构.其中CPU支持嵌入式操作系统和各种通信协议处理的控制,DSP支持无线宽带通信和多媒体等高计算密集度的数字信号处理.采用VLIW静态多发硬件架构,支持最大8条指令并行执行.单核片上SPM为128 kB;指令高速缓存I-Cache大小为16 kB;数据高速缓存D-Cache大小为16 kB.拥有强大的DMA功能,支持多核SPM之间以及与外部Memory之间数据的全交叉通信,可实现视频数据的1D-2D、2D-1D和2D-2D多模态传输等.

本文采用IME-Diamond处理器作为视频解码器架构中的通用同构处理器Core0~Core3,基于FPGA设计实现可升级的协处理器.

IME-Diamond处理器提供了丰富的高性能多媒体运算指令,表3列举了其中的一小部分.

采用IME-Diamond处理器提供的多媒体运算指令,可以加速实现诸如分像素插值、IDCT和边界滤波等计算密集的运算过程.此处以 $4 \times 4$ 块半像素插值为例,简要说明其在IME-Diamond处理器上的实现过程.

表3 IME-Diamond指令简集

Tab.3 Instructions list for IME-Diamond

| 指令        | 功能描述                      |
|-----------|---------------------------|
| vpixdprb  | 四组8 bit × 16 bit 点积运算     |
| vpixdprba | 带累加的四组8 bit × 16 bit 点积运算 |
| vecreord  | 多模式矩阵转置、重排                |
| veexdet16 | 16位数的半向量奇偶位抽取交叉点积         |
| vbytadd   | 四组8 bit 像素或补码数加法          |
| vshf      | 32位数寄存器移位或两组16位数移位        |
| vmovn     | 非零条件赋值                    |
| vmovz     | 零条件赋值                     |

H.264标准对亮度分量半像素点采用6抽头滤波器滤波得出,滤波系数为 $\{1, -5, 20, 20, -5, 1\}$ ,本文将其扩展为8系数滤波器,扩展后的滤波系数为 $\{0, 1, -5, 20, 20, -5, 1, 0\}$ ,利用IME-Diamond提供的vpixdprb和vpixdprba两条指令实现8像素点积运算.如图8所示.

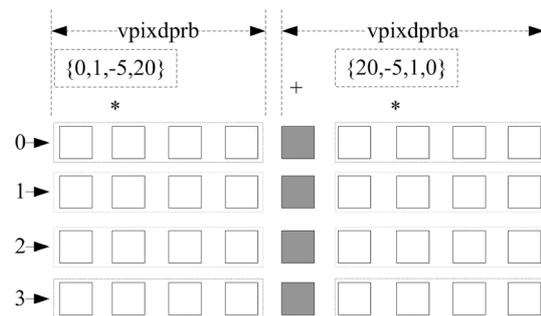


图8  $4 \times 4$ 块半像素插值实现

Fig.8 Half-pixel interpolation for  $4 \times 4$  block

图8中,白色方块代表整像素参考样点,灰色方块为待插值的半像素点.8抽头滤波计算分为前后两部分,前四点乘累加运算由vpixdprb指令完成,后四点乘累加运算及与前四点结果的累加由vpixdprba指令完成.由于不同行插值过程相互独立,且IME-Diamond处理器支持最大8条指令并行,所以每次可以并行计算 $4 \times 4$ 块的四行像素插值运算,若不考虑数据准备过程,两个时钟周期可以完成4个半像素点的插值运算.同理,IDCT和边界滤波等运算也可以按照上述类似的方法实现,不再赘述.

## 4 性能仿真及比较

本文以IME-Diamond处理器为实验平台,以

H.264 传统单核软解码器为对照,以 H.264 官方提供的标准测试序列 akiyo、paris、soccer 和 duck 对基于本文提出的通用多标准视频解码器体系架构下实现的 H.264 多核解码器进行了验证测试.测试结果如表 4 所示.

表 4 不同分辨率视频序列测试结果

| 测试序列   | 分辨率   | 单核/fps | 多核+FPGA/fps | 加速比   |
|--------|-------|--------|-------------|-------|
| akiyo  | CIF   | 76     | 531         | 6.99  |
| paris  | CIF   | 44     | 338         | 7.68  |
| soccer | CIF   | 34     | 330         | 9.71  |
| duck   | 720 P | 2.4    | 27          | 11.25 |

从表 4 中数据可以看出,相比于单核解码器,本文提出的解码方案切实有效,最大加速比可达 11.25.表 5 为相关文献结果对比.

表 5 相关文献结果比较

| 平台     | 核数 | 主频/MHz | 加速比        |
|--------|----|--------|------------|
| 文献[12] | 8  | 2 670  | 7.60       |
| 文献[13] | 16 | 1 000  | 3.20~4.70  |
| 本文     | 5  | 530    | 6.99~11.25 |

文献[13]并行算法在配置为 16 核时的加速比为 3.20~4.70,中值为 3.95;与之相比,虽然本文使用的处理器核数目比较少,但却取得了高的加速比,中值为 9.12,平均提高了 1.31 倍.文献[12] 8 核并行算法虽然取得了 7.60 的加速比,但主频却是本文的 5.03 倍,尽管如此,相比之下,本文加速比平均提高了 20%.文献[12-13]均从软件层次出发,通过设计多线程算法实现线程级并行解码,相比之下,本文根据不同解码模块算法复杂度,将分支情况复杂、计算粒度低的熵解码等过程采用定制硬件协处理器进行加速,实现了更高的加速比.

此外,对于商业化代表海思 Hi35 系列视频解码 SoC 芯片,其解码核心为硬件固化的 HiVXE2.0 处理引擎,虽性能独具优势,但仍旧属于 ASIC 范畴,研发周期长,灵活度差.相比之下,本文提出的基于软硬件协同的视频解码器架构,无论从软件算法还是

协处理器单元结构均支持在线升级,可编程性强,为未来新标准视频解码器设计提供了新的方向.

功耗方面,由于不同实现方法所采用的硬件平台和软件环境不同,暂时无法给出准确的功耗对比结果.本文从自身出发,对系统架构修改前后的功耗情况做了简单的评估,修改前系统功耗约 1.7 W,其中内核功耗约 1.1 W;修改后系统功耗约 5.3 W,内核功耗约 3.6 W.总体上,相比于修改前的单核系统,修改后系统由 4 个处理器核与一个 FPGA 协处理器同时运行,导致系统功耗增加了两倍;但是,由于 FPGA 分担了部分负载,主系统单个内核平均功耗有所降低,约为 0.9 W.

## 5 结 论

本文提出了一种通用的支持多标准的视频解码器体系架构,并以 H.264 视频解码器为实例进行了验证.通过 FPGA 定制协处理器加速单元,实现熵解码、运动矢量和滤波强度计算;通过多核处理器并行实现计算密集的运动补偿、IDCT 和环路滤波模块;通过宏块组的存储结构,有效减少了 DMA 启动次数和相邻宏块边界数据的拷贝;利用分布式片上便笺存储器实现快速的核间数据通信.实验结果表明,本文提出的并行解码方案平均并行加速比为 9.12,比传统算法提高了 1.31 倍.同时,与现有视频解码器架构相比,具有很好的灵活性和可编程性,更能适应不断升级优化的视频解码算法.

## 参 考 文 献

- [1] TREICHLER D G. Are you missing the boat in training aids [J]. *File and Audio-Visual Communication*, 1967, 48(1):28-30.
- [2] 沈沙. H.264/HEVC 视频解码的 VLSI 结构及实现研究 [D]. 上海:复旦大学微电子学院, 2013:7-9.  
SHEN S. Research on VLSI architecture and implementation of H.264/HEVC video decoding [D]. Shanghai: School of Microelectronics, Fudan University, 2013:7-9. (In Chinese)
- [3] 郝秀丽. 基于 GF14 nm 工艺的 H.264 视频解码器综合与物理实现 [D]. 哈尔滨:哈尔滨理工大学应用科学学院, 2016:7-10.  
HAO X L. The physical implement and synthesis of H.264 video decoder based on GF14 nm technology [D]. Harbin: College of Applied Sciences, Harbin University of Science and Technology, 2016:7-10. (In Chinese)

- [4] 王珊珊. 基于 TI C6678 多核处理器的 HEVC 视频解码软件设计[D]. 杭州:浙江工业大学信息工程学院,2014:16—22.  
WANG S S. Design of HEVC video decoder based on TI C6678 multi-core processor [D]. Hangzhou: College of Information Engineering, Zhejiang University of Technology, 2014:16—22. (In Chinese)
- [5] 胡宏华, 湛德荣. 基于 DSP 的 H.264 解码器优化设计[J]. 中北大学学报(自然科学版), 2011, 32(6):763—767.  
HU H H, CHEN D R. Optimal design of DSP-based H.264 decoder [J]. Journal of North University of China (Natural Science Edition), 2011, 32(6):763—767. (In Chinese)
- [6] 马雨然. 基于 FPGA 的 H.264 解码器设计[D]. 成都:中国科学院光电技术研究院, 2017:1—4.  
MA Y R. Design of H.264 decoder based on FPGA [D]. Chengdu: Institute of Optics and Electronics, Chinese Academy of Sciences, 2017:1—4. (In Chinese)
- [7] 刘宁宁, 田泽, 许宏杰. H.264/AVC 解码芯片验证系统及验证策略[J]. 计算机技术与发展, 2014(5):191—194.  
LIU N N, TIAN Z, XU H J. Verification system and strategy of H.264/AVC decoder [J]. Computer Technology and Development, 2014(5):191—194. (In Chinese)
- [8] 潘俊夫. H.264 视频解码器宏块级并行实现与调度优化策略[D]. 武汉:华中科技大学计算机科学与技术学院, 2016:28—38.  
PAN J F. Macroblock level parallel implementation and its scheduling optimization strategy for H.264 decoders [D]. Wuhan: School of Computer Science & Technology, Huazhong University of Science & Technology, 2016:28—38. (In Chinese)
- [9] 杨杭军. 基于多核处理器的视频编解码并行算法研究 [D]. 南京:南京大学电子科学与工程学院, 2013:3—4.  
YANG H J. Video codec parallel algorithm based on multi-core processors [D]. Nanjing: School of Electronic Science and Engineering, Nanjing University, 2013:3—4. (In Chinese)
- [10] 潘竞. SOC 结构的 MPEG-4 编解码器设计[D]. 上海:上海交通大学软件学院, 2007:13—16.  
PAN J. The MPEG-4 codec design base on SoC [D]. Shanghai: School of Software, Shanghai Jiao Tong University, 2007:13—16. (In Chinese)
- [11] VU D, KUANG J L, BHUYAN L. An adaptive dynamic scheduling scheme for H.264/AVC decoding on multicore architecture [C]// IEEE International Conference on Multimedia and Expo. Melbourne, VLC, Australia: IEEE Computer Society, 2012:491—496.
- [12] RICHTER H, STABERNACK B. Architectural decomposition of video decoders for many core architectures [C]// Design and Architectures for Signal and Image Processing. Karlsruhe, Germany: IEEE, 2012:1—8.
- [13] BAAKLINI E, SBEITY H, NIAR S. H.264 macroblock line level parallel video decoding on embedded multicore processors [C]// Euromicro Conference on Digital System Design. Izmir, Turkey: IEEE Computer Society, 2012:902—906.
- [14] 林子明. 高帧率实时 HEVC 熵编解码关键技术研究[D]. 北京:中国科学院大学, 2015:78—80.  
LIN Z M. Key Research on entropy coding/decoding of HEVC in high frame rate real-time application [D]. Beijing: The University of Chinese Academy of Sciences, 2015:78—80. (In Chinese)
- [15] 俞政. 可扩展 64 核处理器关键技术研究——单核、加速器架构及 H.264 解码器实现 [D]. 上海: 复旦大学微电子学院, 2014: 51—52.  
YU Z. Research on key technologies of scalable 64-core processor-single core, accelerator-rich architecture and realization of H.264 decoder [D]. Shanghai: School of Microelectronics, Fudan University, 2014:51—52. (In Chinese)
- [16] 王志君, 梁利平, 吴凯, 等. 一种面向多媒体和通信应用的处理器指令集及架构实现 [J]. 湖南大学学报(自然科学版), 2014, 41(10):108—114.  
WANG Z J, LIANG L P, WU K, *et al.* Architecture and implementation of an application specific instruction set for multimedia and communication [J]. Journal of Hunan University (Natural Sciences), 2014, 41(10):108—114. (In Chinese)
- [17] 王志君, 梁利平, 洪钦智, 等. 一种 DSP 和通用 CPU 一体化的处理器架构及其 4 核实现[J]. 微电子学与计算机, 2014, 31(10): 32—38.  
WANG Z J, LIANG L P, HONG Q Z, *et al.* The architecture of an unified DSP plus general-purpose CPU and the implementation of a 4-core homogeneous processor [J]. Microelectronics & Computer, 2014, 31(10):32—38. (In Chinese)