

# 基于 FPGA 的 H. 264 变换量化的 高性能的硬件实现

刘海鹰 张兆杨 沈礼权

(上海大学教育部新型显示技术及应用重点实验室, 上海 200072)

**摘要** 针对 H. 264 视频标准中一个功能频繁调用的变换量化模块, 提出了一种高性能的 FPGA 硬件实现方法, 并完成了其硬件原型的设计。该硬件原型包含了从残差形成到宏块重建的变换量化全过程, 其可以构成 DSP 的协处理器, 用于完成 H. 264 实时编解码。该硬件原型根据算法特点和数据流特点, 采用了流水线控制策略和分时复用技术, 同时合理利用 FPGA 片内资源, 从而提高了系统性能。仿真结果表明, 该设计能满足高清数字视频的实时处理应用。

**关键词** H. 264 整数变换 分级量化 流水线 分时复用

**中图分类号**: TN919. 81 **文献标识码**: A **文章编号**: 1006-8961(2006)11-1636-04

## A High-performance Hardware Implementation of the H. 264 Transformation and Quantization Oriented to FPGA

LIU Hai-ying, ZHANG Zhao-yang, SHEN Li-quan

(Key Laboratory of Advanced Display and System Application, Ministry of Education, Shanghai University, Shanghai 200072)

**Abstract** This paper presents a high-performance FPGA hardware implementation of the H. 264 transformation and quantization. The hardware prototype is composed of the whole processes from obtaining residual error to Macro-Block reconstruction, and it can be used as a co-processor of DSP to fulfill H. 264 real-time CODEC. Based on the characteristics of algorithms and data flow, the hardware prototype adopts pipeline strategy and time division multiplexing(TDM) technology, and utilizes FPGA dedicated resources reasonably, that enhances the performance of the hardware prototype greatly. The simulation results show that the design can satisfy the real-time constraints required by HDTV applications.

**Keywords** H. 264, integer transformation, scalable quantization, pipeline, time division multiplexing(TDM)

## 1 引言

当前, H. 264 已由研究领域过渡到应用领域。由于 H. 264 优异的编码效率是以前其高复杂性为代价的, 因而成为其实时应用的瓶颈。在诸如视频监控等低分辨率的应用中, 主要采用 DSP (digital signal processor) 来实现 H. 264 编解码实时化, 但在诸如 HDTV (high definition television) 等高分辨率的应用中, 鉴于性价比, 及系统复杂度等多种因素, 单纯依靠 DSP 阵列已不再适合实现 H. 264 实时编解

码。故较为通行的做法是采用 DSP + FPGA 的解决方案, 也即为 H. 264 标准中频繁调用的或者复杂耗时的模块设计专门的 FPGA (field programmable gate array) 硬件加速引擎, 来作为 DSP 的协处理器, 以达到实时编解码的目的<sup>[1]</sup>。

变换量化模块是 H. 264 中一类频繁调用的模块。在 H. 264 中, 变换编码和量化过程中采用了整数类 DCT (discret cosine transform) 变换, 并将尺度调整融合到量化过程中。由于变换过程仅通过加法和移位操作完成, 用整数运算代替浮点运算, 这不仅提高了变换速度, 也解决了反变换的失配问题, 同时由

**基金项目**: 国家自然科学基金重点项目(60332030)和上海市科委重点攻关项目(055115008)

**收稿日期**: 2006-06-30; **改回日期**: 2006-08-01

**第一作者简介**: 刘海鹰(1976~), 女, 1998 年获安徽大学学士学位, 现在上海大学通信与信息工程学院攻读硕士学位。主要研究方向为数字视频编码和传输。E-mail: eaglecrown@163.com

于量化过程用乘法和移位操作代替除法运算,因而适合用硬件实现。

在文献[2][3]给出的变换量化模块的硬件设计中,采用了流水线来提高数据吞吐量,但由于其仅包含了变换量化过程的部分功能,结构不完整,如果要作为 DSP 的协处理器则需添加大量额外的运算逻辑和接口逻辑。同时,工作频率较低,无法适应高分辨率数字视频的实时应用。本文给出了一种基于 FPGA 的 H.264 变换量化模块的硬件原型设计,其能够在资源消耗不明显增加的情况下,工作在尽可能高的时钟频率上,适合高分辨率视频应用。

## 2 算法原理

H.264 标准中,变换编码是基于  $4 \times 4$  块,并采用以下新的变换核<sup>[4]</sup>:

$$Y = CXC^T \otimes E$$

$$= \begin{pmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} [X] \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \\ a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \end{bmatrix} \end{pmatrix} \quad (1)$$

其中,  $C$  是核矩阵,  $E$  是尺度调整矩阵,  $a = 1/2, b = \sqrt{2/5}$ 。核变换 ( $CXC^T$ ) 仅用加法和移位运算就能实现,且是短字长运算<sup>[5]</sup>。

由于变换块尺寸从  $8 \times 8$  降至  $4 \times 4$ ,虽减轻了振铃效应,但能量集中性变差,故 H.264 采用分级变换方案。对色度块和帧内编码的  $16 \times 16$  亮度块,每个  $4 \times 4$  子块经过第 1 级整数变换后,抽取其 DC 系数组成一个新的  $4 \times 4$  亮度块(或  $2 \times 2$  色度块),再对其进行一次哈达码变换,以进一步去除 DC 系数的相关性,这不仅能量更集中,且编码效率更高。

H.264 采用分级量化器,从而避免了除法和浮点运算。由于整数变换后的尺度调整可在量化过程中实现,故算法复杂度降低。残差系数量化过程可用下式实现<sup>[4]</sup>:

$$|Z_{i,j}| = (|W_{i,j}|MF_{i,j} + f) \gg k$$

$$k = 15 + \text{floor}(QP/6) \quad (2)$$

$$\text{sign}(Z_{i,j}) = \text{sign}(W_{i,j})$$

其中,  $Z_{i,j}$  为残差系数经变换量化后的值,  $W_{i,j}$  是核变换后的矩阵元素;  $QP$  是量化参数;  $k$  代表右移位,等效为除以  $2^k$ ;  $f$  用来四舍五入取整,取值范围为  $[0, 0.5] \times 2^k$ ,一般对帧内块取  $2^k/3$ ,帧间块取  $2^k/6$ ;  $MF_{i,j}$  是乘数因子,其取值取决于  $QP$  和变换系数的位置  $(i, j)$ 。  $MF_{i,j}$  取值如表 1 所示。

表 1 乘数因子  $MF$   
Tab.1 Multiplication factor  $MF$

QP	(i, j)位置乘数因子		
	(0,0), (2,0), (2,2), (0,2)	(1,1), (1,3), (3,1), (3,3)	其他
0	13 107	5 243	8 066
1	11 916	4 660	7 490
2	10 082	4 194	6 554
3	9 362	3 647	5 825
4	8 192	3 355	5 243
5	7 282	2 893	4 559

## 3 硬件原型设计

### 3.1 DCTQ 硬件结构

H.264 的变换过程根据残差数据类型的不同分成了 3 类,其中包括对帧内  $16 \times 16$  模式亮度宏块(I16YMB)的  $4 \times 4$ DC 系数的变换和对色度宏块(chrominance macro block, CMB)的  $2 \times 2$ DC 系数的变换,以及对所有其他类型的  $4 \times 4$  块的变换。在本文所实现的 H.264 DCT 变换量化的 DCTQ (discrete cosine transform and quantization)硬件原型中,可根据算法特点分成 6 个模块,其中包括  $4 \times 4$  整数变换及反变换、哈达码变换及反变换、量化及反量化,而且只要修改 DCTQ 参数设置,就可以灵活地处理各种类型残差宏块。DCTQ 模块的外部接口如图 1 所示。

DCTQ 对宏块数据的处理流程如图 2 所示。图

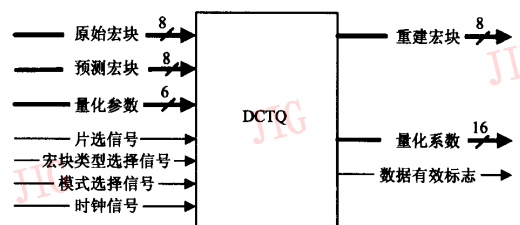


图 1 DCTQ 模块接口关系

Fig.1 DCTQ module interface block diagram

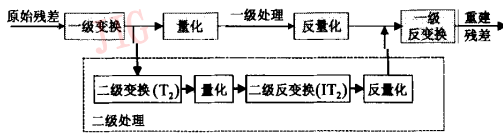


图 2 DCTQ 模块数据流图

Fig. 2 DCTQ module dataflow block diagram

中灰色标记的模块为一级处理模块,所有类型的数据都要经过一级处理,其中包括一级变换( $T_1$ )、量化( $Q$ )、反量化( $IQ$ )和一级反变换( $IT_1$ );虚线框内的模块为二级处理模块,仅用于处理 CMB 或 I16YMB 的 DC 系数。根据数据流的特点,对分级处理中的量化和反量化模块分时复用,这样可以减少资源消耗,同时,为避免过长的路径延迟造成工作频率下降,故不再对其他模块进行分时复用。

由于整个数据处理具有“单向流”特点,即没有反馈或迭代运算,因此各硬件模块的管理和调度可采用流水线控制策略来提高系统性能。流水线设计的关键,一方面在于整个设计时序的合理安排;另一方面要注意前后级接口之间数据流量的匹配,统筹考虑各个操作环节间的数据流量,当前后级之间数据流速不匹配时,可以通过串并转换等手段将数据分流,或者在前后级间加入适当容量的缓存。

DCTQ 的时序开销如图 3 所示(以 I16YMB 为例)。流水线上的数据包为  $4 \times 4$  子块,考虑到一级反变换必须等待二级 DC 系数处理完成才能进行,故仅在当前宏块的第 1 个  $4 \times 4$  子块反变换结束后,

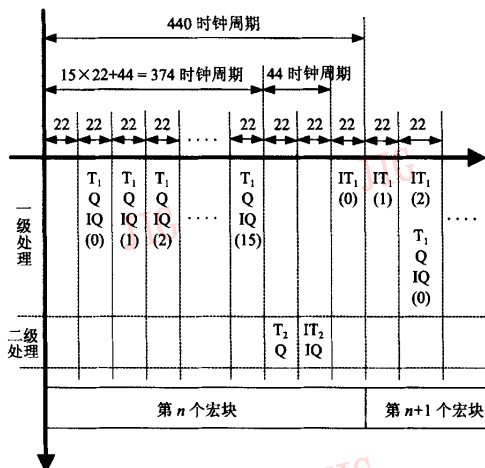


图 3 帧内  $16 \times 16$  模式亮度宏块时序开销

Fig. 3 The clock cycle overhead for I16YMB

并开始重建时,新的宏块数据才能进入 DCTQ。在同一时刻,一级处理各模块之间可以并行工作,即对第  $n$  个宏块和第  $n+1$  个宏块的不同数据包并行处理,充分利用各模块的处理能力来提高数据吞吐量;而二级处理模块则需要根据数据类型来激活。这样,DCTQ 对一个宏块的处理时间为 440 个周期(以 I16YMB 为例)。为了确保数据流量的匹配,需要在流水线上的前后级间插入缓存,一般对于容量较小的缓存可以直接用逻辑单元实现,而当缓存容量较大时,为了减少逻辑资源的消耗,则可以利用 FPGA 内部的 RAM block 实现。例如,在本结构中调用了两个 RAM,分别用于存储宏块内全部  $4 \times 4$  子块的 DC 系数以及 AC 系数的反量化结果。

### 3.2 整数变换模块

整数变换模块采用行列分离法来实现 2 维整数变换,即先逐行做 1 维变换,然后对变换结果转置后再进行一次 1 维变换,即得到 2 维变换结果。整数变换框图如图 4 所示,变换时,每个数据包中的数据按扫描顺序输入,每个时钟输入一个残差数据,每读入一行数据后,即开始做 1 维变换。1 维变换采用蝶形结构的快速算法<sup>[5]</sup>。如果用 CPU 实现一个 1 维变换需要两次加法,两次移位及一次减法操作,共需 5 个指令周期,而在 DCTQ 中充分利用 FPGA 丰富的逻辑资源,则上述操作可以在一个时钟周期内完成,从而减少了流水线上的操作环节。考虑到数据流的单向性,图 4 中的转置寄存器和输出寄存器是分时复用的,这样可以有效减少资源消耗。

由于反变换过程类似,而哈达码变换仅权重因子有所变化,因此它们的实现过程不再详细介绍。

### 3.3 量化模块

根据前面介绍的量化公式(式(2)),量化和反量化模块只需要乘加操作和移位操作就可以实现,其中乘法操作可用 FPGA 内部的专用硬核(如 Altera 公司 StratixII 系列中的 DSP block)实现,这样既节约了逻辑资源,又能提供高性能。另外,量化和

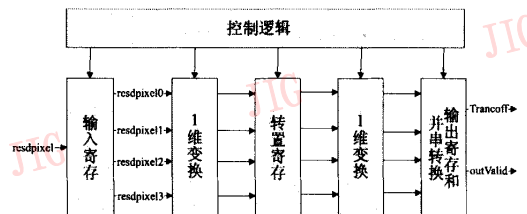


图 4  $4 \times 4$  整数变换硬件结构

Fig. 4  $4 \times 4$  integer transformation hardware architecture

反量化模块中各需要调用一个 ROM,分别用于存放正反量化中的乘数因子。量化表设计成二维数组,其检索地址  $(i, j)$  与量化参数  $QP$  及变换系数的位置  $pos$  的映射关系为

$$\begin{aligned} i &= \text{mod}(QP, 6), QP = 0, 1, \dots, 51 \\ j &= (pos \& 1) + ((pos \gg 2) \& 1), pos = 0, 1, \dots, 15 \end{aligned} \quad (3)$$

通过这样的映射关系,就可以将量化表的尺寸由  $6 \times 16 \times w_{\text{bit}}$  缩小为  $6 \times 3 \times w_{\text{bit}}$ ,大大减少了存储资源消耗。其中,  $w_{\text{bit}}$  表示乘数因子的位宽,对正量化  $w_{\text{bit}}$  取 14,对反量化  $w_{\text{bit}}$  取 5。

## 4 仿真及实验结果

该原型用 VHDL 语言设计,并采用 Mentor-Graphics 公司的 ModelSim6.1 仿真,同时用 Precision2005.c 进行综合,其目标器件是 Altera 公司的 stratiXII 系列的 EP2S15 器件,布局布线采用 Altera 公司的 QuatusII6.0。表 2 列出了该硬件原型与文献[2,3]的性能对比数据。

表 2 性能对比数据  
Tab.2 Performance comparison

	文献[2]方法	文献[3]方法	本文设计方法
关键路径(ns)	28.3	7.73	6.802
时钟频率(MHz)	34.8	>120	147.02
功能	4×4 正变换量化	4×4 正/反变换	包含了从残差获取到 宏块重建的变换量化的 完整功能
处理时间	—	64 周期/子块	≤440 周期/宏块

由表 2 看到,与同类型设计<sup>[2,3]</sup>相比,DCTQ 模块包含了 H.264 变换量化部分的完整功能,且工作频率大幅提高。文献[2]中,对基于  $4 \times 4$  块的变换量化在高清格式视频应用中的时序需求进行了简要分析,其结论是建立在正向变换量化的关键路径决定了整个模块的实时性的基础上。但是,对于 H.264 变换量化这样的分析是不充分的,因为其没有考虑到反向变换量化部分以及对于 DC 系数的处理部分。事实上,由于 H.264 编码器中的数据单元是宏块,因此宏块的处理时间决定了整个模块的实时性。如前文所述,DCTQ 对一个宏块的最大

处理时间为 440 个时钟周期。

根据上面的分析,如果要完成对一个高清格式视频 ( $1280 \times 720$ ) 的实时处理(帧率为 60fps),则其所需的主时钟频率计算如下:

$$f_{\text{max}} = \frac{(1280 \times 720 \times 60) \times 440}{256 \times 1\,000\,000} = 95.04 \text{ MHz} \quad (4)$$

由表 2 可以看到,DCTQ 模块的主时钟频率达到 147.02MHz,完全能满足上述的高分辨率实时应用的时序要求,其不仅能保证系统稳定工作,并拥有足够的时序余量,具有较大的性能提升空间。

## 5 结论

本文介绍了一种基于 FPGA 的 H.264 变换量化的硬件原型的设计。在硬件原型中,首先采用流水线控制策略,使各硬件模块并行工作,充分利用各模块的处理能力,来提高系统的工作频率;其次,根据数据流的特点,对部分模块或寄存器由于采用了分时复用的技术,并合理利用了 FPGA 内部的各种 RAM block, DSP block 资源,从而降低了资源消耗。仿真结果表明,该硬件原型可以满足高清格式数字视频的实时应用。为进一步提高编解码的实时性,下一步工作需要 H.264 中最为复杂耗时的部分运动估计模块设计专门的硬件加速引擎。

### 参考文献(References)

- Xilinx Inc., Texas Instrument Inc.. DSP and FPGA complementary solutions for HD video infrastructure systems [EB/OL]. <http://www.tikorea.co.kr/seminar/tidc2006>
- Amer I, Badawy W, Jullien G. Hardware prototyping for the H.264  $4 \times 4$  transformation[A]. In: International Conference on Acoustics, Speech & Signal Processing [C], Montreal, Canada, 2004, 5: 77~80.
- Liu L Z, Qiu L, Rong M T, et al. A 2-D forward/inverse integer transform processor of H.264 based on highly-parallel architecture [A]. In: IEEE International Workshop on System-on-Chip for Real-time Applications [C], Alberta, Canada, 2004: 158~161.
- Iain E G Richardson. H.264/MPEG-4 Part 10 white paper-transform & quantization[EB/OL]. <http://www.vcodex.com>.
- Malvar Henrique S, Hallapuro Antti, Karczewicz Marta, et al. Low-complexity transform and quantization in H.264/AVC [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2003, 13(7): 598~602.