

# 基于 PLB 总线的 H.264 整数变换量化软核的设计

吴从中, 项磊, 蒋建国

(合肥工业大学 计算机与信息学院, 安徽 合肥 230009)

**摘要:** 提出了在 FPGA 上实现 H.264 中整数变换量化的方法, 设计了基于动态数据宽度和流水线技术的软核(IP), 在处理速度和硬件资源方面分别进行优化。此软核作为 PowerPC 的一个硬件加速模块在 Xilinx Virtex-II PRO 中进行了验证。实验表明, 在目前较难使用软件方法实现高分辨率图像实时编码的情况下, 本文设计的软核能够提供 2110M Pixels/s 的编码速率, 完全适应实时编码。

**关键词:** H.264; 整数变换量化; 动态数据宽度; 流水线; PLB; 软核

## Hardware implementation of a PLB IP for H.264 integer transform and quantization

WU Cong Zhong, XIANG Lei, JIANG Jian Guo

(School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, China)

**Abstract:** This paper proposes a method of realizing the integer discrete transform and quantization blocks from H.264 on FPGA and a design of IP based on dynamic data width and pipelining technology, which are optimized for processing speed and hardware resources. As a hardware accelerating module of PowerPC, this IP is verified on a Virtex-II PRO FPGA. Under the current condition that it is difficult to use software methods to achieve real-time encoding of high-resolution images, the design presented in this paper can provide 2110M Pixel/s encoding rate, which fully adapts to real-time coding.

**Key words:** H.264; integer transform and quantization; dynamic data width; pipeline; PLB; IP

H.264 以其高复杂度为代价获得了优异的编码效率, 其中除部分控制流程的复杂模块外, H.264 中的很多模块适合用硬件实现。应用中通常使用 CPU+FPGA 结构, 将耗时较多的模块用 FPGA 实现, CPU 仅负责一些低复杂度的算法和编码流程参数的设置。

参考文献[1]、[4]、[5]介绍了整数变换量化的硬件实现, 但没有考虑数据在处理过程中的宽度问题, 因此会造成 FPGA 资源的浪费。本文充分考虑了数据在处理过程中的动态范围, 使用更少的 FPGA 资源来实现 H.264 中的整数变换量化模块。

结合实际应用, 文中对该软核在计算速度和硬件资源方面分别做了优化。经过速度优化的软核性能明显优于文献[1]、[4]、[5]中的设计, 经过消耗资源优化的软核也完全能够胜任高分辨率的实时编码。

### 1 整数变换

在完成帧内和帧间预测以后, 需要对图像参差数据进行整数变换和量化, 使图像数据的能量集中到一小部分系数上, 进一步降低码流速率。

#### 1.1 整数变换原理

H.264 中对图像参差进行二维 DCT 变换, 表达式为:

$$Y = AXA^T \quad (1)$$

其中:

$$A_{ij} = \begin{cases} \sqrt{\frac{1}{N}} & i=0; j=0, 1, \dots, N-1 \\ \sqrt{\frac{2}{N}} & i=1, 2, \dots, N-1; j=0, 1, \dots, N-1 \end{cases}$$

$X$  为输入数据,  $A$  为变换矩阵,  $Y$  为变换结果。H.264 对  $4 \times 4$  的图像块进行操作, 则相应的  $4 \times 4$  DCT 变换矩阵  $A$  为:

$$A = \begin{bmatrix} a & a & a & a \\ b & c & -c & b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix}$$

$$a = \frac{1}{2} \quad b = \sqrt{\frac{1}{2}} \cos\left(\frac{\pi}{8}\right) \quad c = \sqrt{\frac{1}{2}} \cos\left(\frac{3\pi}{8}\right)$$

# 嵌入式技术 Embedded Technology

防止解码后的数据失配, H.264 对 4×4 DCT 中的 A 进行改造, 采用整数 DCT 技术, 取  $b = \sqrt{\frac{2}{5}}$ , 则(1)式可分解为:

$$Y = (CXC^T) \otimes E \quad (2)$$

其中:

$$C = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & 2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \quad E = \begin{bmatrix} a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \\ a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \end{bmatrix}$$

与 E 的乘法被归纳到量化运算中, 这样(CXC<sup>T</sup>)中只剩下整数的加法、减法和移位运算, 因此可以大大降低硬件实现的复杂度, 变换结果最多只需要 16 位的数据。

## 1.2 整数变换硬件结构

本设计 H.264 中的 4×4 整数变换采用蝶形快速算法, 如图 1 所示。首先对 4×4 块的每一行做一维整数变换, 然后再对行变换结果做列的一维整数变换, 最终得到 4×4 的整数变换结果。图 1 模块需要 32 个加法器和 32 个减法器, 这样在一个时钟周期内就可以完成一个 4×4 块的整数变换。

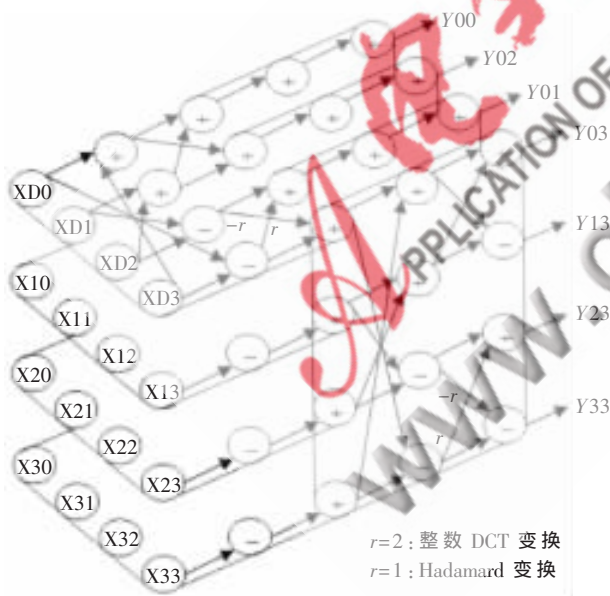


图 1 4×4 快速整数变换

整数变换通常的做法是全部使用 16 位加法器和减法器, 这样可以简化设计。然而在实际应用中, 输入的像素点范围是 [0, 255], 如果第一级变换就使用 16 位, 则数据宽度无疑会占用更多的硬件资源。根据分析可知, 计算一维行变换以后, 数据输出范围为 [-765, 1020], 只需使用 11 位二进制数表示。计算一维列变换以后, 数据输出范围是 [-3 060, 4 080], 需要使用 13 位二进制

数表示。因此整数变换模块的数据宽度可以确定为: 进行一维行变换时的加法器和减法器使用 11 位数据宽度, 一维列变换的加法器和减法器使用 13 位数据宽度。

由于数据单向流动的特点, 即没有反馈, 可以使用流水线提高系统性能。使用流水线时需要注意前后模块处理速率的匹配。如本模块中在行变换与列变换插入缓存构成前后模块, 前后模块用同一时钟, 并且整个模块的工作频率以最低工作频率的模块来确定。本设计用 1 级流水线来提高工作频率。如果流水线级数过多会消耗大量 FPGA 资源, 仿真实验证明, 此模块使用 1 级流水线时, 只增加极少 FPGA 资源。

## 2 量化

为了进一步降低图像传输码率, 需要对图像进一步压缩, 方法是使用变换编码及量化技术。

### 2.1 量化原理

H.264 中采用标量量化器。标量量化器的原理是:

$$FQ = \text{round}\left(\frac{y}{Q_{step}}\right) \quad (3)$$

其中,  $y$  为输入样本点编码,  $Q_{step}$  为量化步长,  $FQ$  为  $y$  的量化值。H.264 标准支持 52 个量化步长。量化的简化操作如下:

$$|Z_{ij}| = (|W_{ij}| \cdot MF + f) \gg q_{bits} \quad (4)$$

$$\text{sign}(Z_{ij}) = \text{sign}(W_{ij}) \quad (5)$$

$$q_{bits} = 15 + \text{floor}(QP/6) \quad (6)$$

其中,  $W_{ij}$  为输入样本点编码,  $MF$  是标准中定义的值,  $QP$  为量化系数,  $f$  为偏移量, 对帧内预测图像块  $f$  取  $2^{q_{bits}}/3$ , 对帧间预测图像块  $f$  取  $2^{q_{bits}}/6$ 。

### 2.2 量化器实现

量化器硬件结构如图 2 所示, 其结果是实现对式(4)和式(5)的组合电路。W 是需要量化的数据, MF 是根据 QP 和当前点位置在查找表中得到的值。f 和 qbits 是根据 QP 查表的输出。

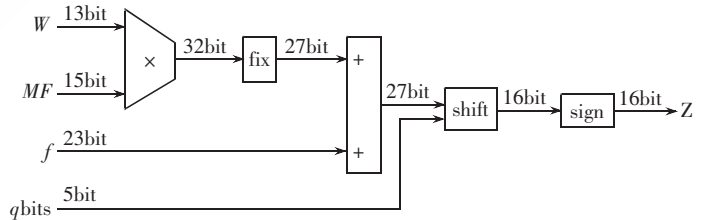


图 2 量化器硬件结构

为了硬件资源消耗最少, 需要分析数据在计算过程中的动态范围, 使用最少的数据宽度来表示传输的数据, 从而使用最少的硬件资源来实现组合运算逻辑。W 是整数变换结果, 根据整数变换部分的分析, 其取值范围是 [-3 060, 4 080], MF 最大取 13 107, 乘法器输出动态范围是 [-40 107 420, 53 476 560], 至少用 27 位二进制数表示。本量化器的乘法器使用 Verilog 中的 18×18 硬件

# 嵌入式技术 Embedded Technology

乘法器,因此在输入加法器之前需要将数据宽度调整为27bit。加法器输出的27位数据通过移位模块调整为16位数据宽度。

综合考虑工作频率和资源占用率,设计中使用2级流水线来提高性能。

## 2.3 量化器优化

本量化器一次只能量化一个点,无法满足高质量图像编码的要求。针对高质量图像编码对此量化器进行速度优化,即将 $n$ 个量化器并联,一个时钟周期计算 $n$ 个点。 $n$ 根据实际需要进行选择。考虑到与变换模块的速度匹配,选择16个量化器并联。

## 3 软核设计及测试

### 3.1 PLB总线的软核结构

PLB (Processor Local Bus) 总线是IBM开发的一种高性能片上总线,主要应用于PowerPC405处理器系统中,它支持32位、64位和128位数据宽度。本设计使用64位总线宽度,最大速度传输速率达800Mb/s。

如图3,PLB的软核设计分为总线接口和H.264整数变换量化模块两部分。PLB总线接口是与硬件体系结构相关的部分,它是整数变换量化模块与PLB总线上其他设备进行交互的桥梁。整数变换量化模块与体系结构无关,它也可以被移植到ARM体系结构中。

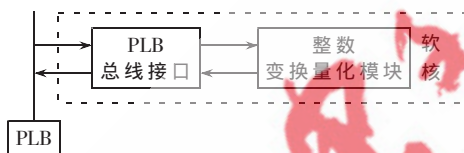


图3 PLB软核结构

整数变换量化模块的硬件结构如图4。寄存器组包含9个32bit的寄存器,0、1、2、3为输入的4×4参差数据寄存器;5、6、7、8为输出数据寄存器;4为控制寄存器,包括go、done、reset、intra、DCT、ZSCAN、QP、datacount,分别对应启动、完成、复位、帧内/帧间、DCT/Hadamard变换、Z扫描输出、量化级数、数据输出计数器。第二个模块对寄存器组中数据进行4×4整数变换,使用1级流水线。第三个模块对输入的整数变换结果按QP进行量化,量化模块中的 $MF$ 、 $f$ 、 $qbits$ 使用FPGA中的查找表保存。数据输出模块受ZSCAN位控制,当ZSCAN=1时,输出数据寄存器中的数据按Z扫描排列;ZSCAN=0时,按矩阵排列。

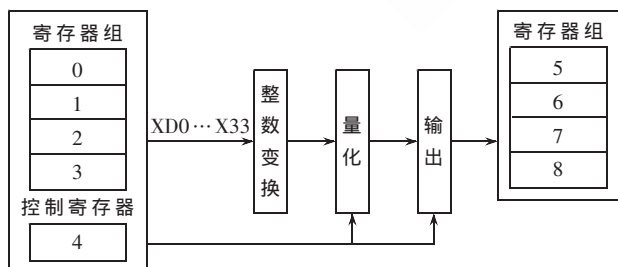


图4 整数变换量化模块硬件结构

### 3.2 软核的测试

首先使用IBM的CoreConnect工具和Modelsim6.0a来仿真设计的软核并使用虚拟平台进行测试,仿真的目的是保证PLB总线接口能被处理器正确访问。仿真测试平台结构如图5所示。通过虚拟处理器将虚拟内存中的数据写入待测软核,然后读取待测软核中的数据并判断是否正确。

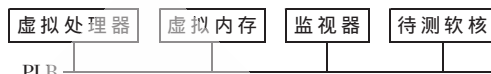


图5 仿真测试平台

仿真通过以后,再将软核集成到系统中,以便验证软核在实际系统中的工作是否符合要求。验证平台系统结构如图6,软核验证使用PC机和目标板相结合的方法,目标板使用Xilinx公司XUP Virtex-II PRO开发板,内部含有两个PowerPC内核。

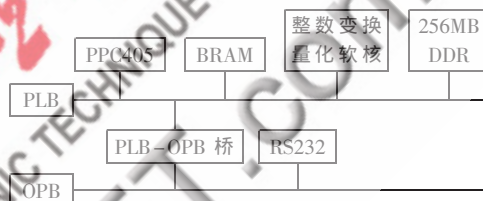


图6 验证平台结构

验证流程如图7。首先通过USB口下载目标板程序,程序开始运行以后通过RS232从上位机下载待编码的图像参差数据,下载的数据保存在目标板上的256MB DDR SDRAM中。数据下载完毕后,程序将待编码数据依次写入软核并启动转换,并将计算完的数据写入DDR SDRAM,待全部数据编码完毕,PowerPC将处理结果一起发送给上位机。上位机将结果与本机C代码执行结果相比较,最终确认软核是否正确工作。

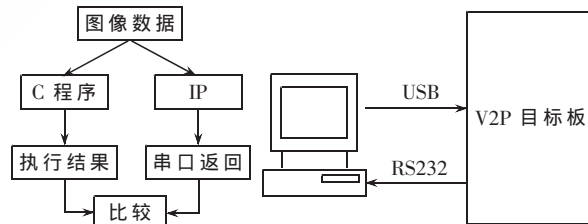


图7 验证流程

## 4 综合结果分析

本设计中的整数变化量化模块在Xilinx的XC2VP30 FPGA中进行综合。XC2P30含有13 696个Slices和136个18×18乘法器。综合工具使用Xilinx的ISE9.1。

整数变换模块综合结果如表1。

表1 整数变换模块综合结果

	Slices	关键路径/ns	Pixels/s
整数变换模块	357	7.357	2174M

# 嵌入式技术 Embedded Technology

对量化器在速度和硬件资源上分别做了优化,表 2 给出了两种优化的综合结果。

表 2 量化器综合结果

	Slices	关键路径/ns	Pixels/s
速度优化	1 268	7.581	2 110M
资源优化	126	6.779	147M

表 3 给出本设计整数变换量化模块与文献[1]的比较结果。对照发现速度优化软核编码性能远远高于文献[1]的设计,资源优化的软核性能也略高于文献[1]的设计。

表 3 与文献[1]比较

	文献[1]的方法	速度优化模块	资源优化模块
关键路径/ns	6.802	6.779	7.581
时钟频率/MHz	147.02	147.51	131.90
处理时间/(周期/宏块)	≤ 440	16	256
处理 1280×720, 60f/s 视频所需频率/MHz	95.04	3.456	55.296

软核综合结果如表 4。目前该软核中的 PLB 接口部分资源占用较大,后续工作对这部分进一步优化。

本文将 H.264 中的整数变换量化与微处理器系统相结合,针对不同的应用场合实现了两个基于 PLB 总线的 H.264 整数变换量化软核,并在 Xilinx XUP Virtex-II PRO 开发板中做了验证。实验结果表明,两个软核均能

表 4 软核综合结果

模块	占用 Slices	18×18 硬件乘法器	最长路径/ns
速度优先整数变换量化软核	2 382	16	7.946
资源优先整数变换量化软核	1 240	1	7.305
PLB 接口	757	0	7.236
验证系统	1 130	0	7.203

在系统中稳定工作,并能满足不同分辨率的实时应用。

## 参考文献

- [1] 刘海鹰,张兆杨,沈礼权.基于 FPGA 的 H.264 变换量化的性能高的硬件实现.中国图像图形学报,2006,11(11).
- [2] RAJA G, KHAN S, MIRZA M J. VLSI Architecture & Implementation of H.264 Integer Transform. IEEE, 2005.
- [3] LEE S, Kyeongssoon Cho. Implementation of an AMBA-Compliant IP for H.264 Transform and quantization. IEEE, 2006.
- [4] 楼剑,陆亮,虞露,等.H.264 变换和量化的分析.浙江大学学报,2004(5).
- [5] 毕厚杰.新一代视频压缩编码标准——H.264/AVC[M].北京:人民邮电出版社,2005.
- [6] IBM Inc, Xilinx Inc. 64-Bit Processor Local Bus Architecture Specifications. [http://www.xilinx.com/products/virtex/coreconnect/03\\_05\\_02\\_cc/CoreConnect/crcn.html](http://www.xilinx.com/products/virtex/coreconnect/03_05_02_cc/CoreConnect/crcn.html).

(收稿日期:2008-04-28)