一种适用于 H.264/AVC 宏块级反变换编码的 IP 核设计

张益林[1)] 徐 雄[1)] 杨宇红[2)]

[1)](上海交通大学电子工程系,上海 200240) [2)](上海交通大学电子系图像通信与信息处理研究所,上海 200240)

摘 要 本文提出了适用于 H.264/AVC 宏块级反变换的 IP 核完整设计方案。首先,使用改进的 T 型结构同步宏块中的 3 种不同变换和反 Zig-Zag 扫描。然后,对 Hadamard 反变换模块采用了时分复用存储器模块的设计方案,降低了系统时延;再利用 IDCT 矩阵运算可分离的特点,减少了 IDCT 模块资源消耗;最后,给出了以 Xilinx Viretex2 系列 XC2V6000 为目标器件的综合结果。仿真结果表明,该设计能够正确支持 1080i 50Hz 高清码流的实时解码。

关键词 反 DCT 变换 Hadamard 反变换 T 型结构 IP 核

中图法分类号:TP301.6 **文献标识码**:A **文章编号**:1006-8961(2008)10-2019-04

Design of a Macroblock Level Inverse Transform IP Core for H.264/AVC

ZHANG Yi-lin[1)], XU Xiong[1)], YANG Yu-hong[2)]

[1)](*Department of Electronic Engineering, Shanghai Jiaotong University, Shanghai* 200240)

[2)](*Institute of Image Communication and Information Processing, Department of Electronic Engineering, Shanghai Jiaotong University, Shanghai* 200240)

Abstract A high throughput inverse transform IP core for H.264/AVC was proposed in this paper. The improved T architecture was presented to synchronize three different transforms and inverse Zig-Zag scan module. By applying time multiplexing buffer management to inverse Hadamard transform, we efficiently reduce its latency. Separability property of IDCT is also utilized to minimize its area. At last, the results of synthesis are given with Xilinx Virtex2 while XC2V6000 as the target device. The simulation performance shows that the design can effectively support the real-time decoding of 1080i 50Hz HD stream.

Keywords inverse DCT, inverse Hadamard transform, T architecture, IP core

1 引 言

由 ITU 和 ISO 组成的 JVT(joint video team)所提出的 H.264/AVC 是目前最新的视频编解码标准。多帧参考,1/4 像素精度的运动估计,整数 DCT 变换,环内滤波等先进技术[1]的引入大大提高了它的编码效率,但是也使计算复杂度空前增加。权威测试结果表明:H.264/AVC 压缩效率是 MPEG-2 的 2 倍。解码复杂度约为 MPEG-2 的 4 倍,是 MPEG-4VSP 的 2 倍[2]。巨大的计算量对实时解码提出了挑战。硬件设计尤其是 ASIC 具有计算速度快、低成本的优点,使用硬件实现实时解码成为了大家的共识。在这种背景下,本文提出了适用于 H.264/AVC 宏块级反变换编码的 IP 核完整设计。

2 H.264/AVC 宏块级反变换编码原理

对于一个 H.264/AVC 宏块的残差数据,编码器提供了 3 种不同的变换工具用于其编码过程:4×4

收稿日期:2008-07-11;**改回日期**:2008-07-23

第一作者简介:张益林(1983~),男。上海交通大学电子系电路与系统专业硕士研究生。研究方向为视频与图像编解码、数字图像处理。E-mail:kenter@sjtu.edu.cn

DCT 变换,4×4 Hadamard 变换以及 2×2 Hadamard 变换。DCT 变换接近 K-L 变换但计算复杂度更低,它用于将 4×4 块内的能量使用较少的低频系数表征;Hadamard 变换则用于减少不同块之间的 DC 系数信息冗余。由于这 3 种变换同时存在于宏块中,为了处理经过 H.264/AVC 编码的码流,解码器在宏块级必须支持这 3 种变换的反变换。表 1 列举了解码器端对不同编码块所采用的变换工具。

表 1　反变换方式选择

Tab.1　Usage of inverse transform tools

数据类型	反变换方式
亮度/色度块系数	4×4 反 DCT 变换
亮度块的 DC 系数	4×4 Hadamard 反变换
色度块的 DC 系数	2×2 Hadamard 反变换

此外,为了减少 0 系数的传输,编码器还采用了 Zig-Zag 扫描对系数进行重列,这意味着在解码器端必须提供反 Zig-Zag 扫描将系数位置复原。

3　改进的 T 型结构

变换工具的多样性和码流的复杂性使设计一种高效的结构同步不同的工具显得十分必要。文献[3]提出了 T 型结构同步不同的变换模块,该结构为并行结构,吞吐率高,而且具有数据输出率恒定的优点。该 T 型结构的基本假设是模块的输入特性确定,即对于任意宏块编码模式均有相同数量的输入样点。这个假设对于编码器是合适的,但解码器接收的码流会由于编码模式是否为 Intra16×16 而改变:如果为 Intra16×16,码流中将会出现额外的 4×4 DC 系数块,反之则否。为了继承 T 型结构的优点并使其支持解码器设计,本设计对文献[3]中的 T 型结构重新设计,如图 1 所示。

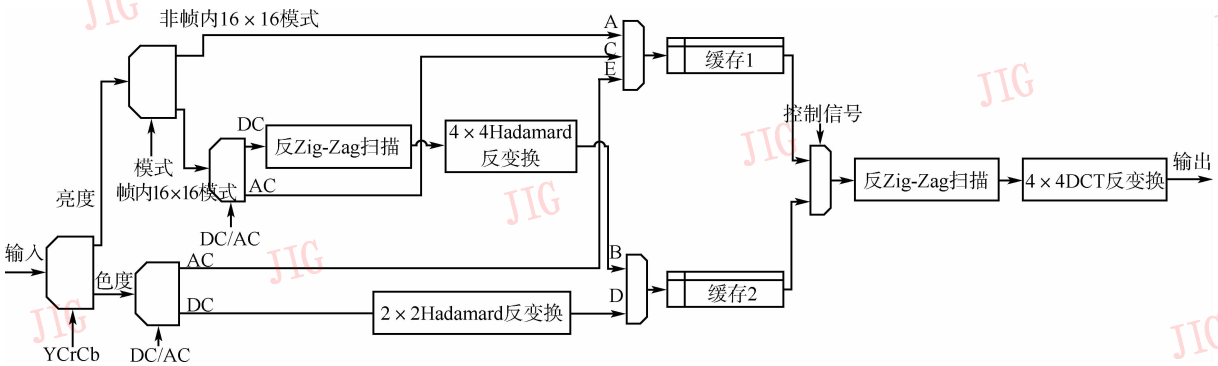


图 1　改进的 T 型结构

Fig.1　Improved T architecture

本设计与文献[3]结构不同之处在于反 Zig-Zag 模块的加入,解码器需要根据数据的类型决定何时进行反扫描。如果是 DC 系数块,那么直接进行反 Zig-Zag 扫描,以便进行 Hadamard 反变换;如果是 AC 系数块,那么需要将数据经过多路选择器,使 AC 系数与 DC 系数融合之后才能进行反 Zig-Zag 扫描。因此解码器中需要使用两个反 Zig-Zag 扫描模块分别用于 Intra 16×16 编码后 DC 系数反扫描和融合后 4×4 数据块反扫描。

除了加入反 Zig-Zag 模块以适应解码器的功能需求之外,本设计还对 T 型结构进行了进一步优化:文献[3]的设计对每个数据支路均采用单独的 buffer 进行数据同步;本设计通过分析各数据支路的特点,将先进先出缓存进行时分复用,减少了资源消耗。以 A,C,E 数据流为例说明该时分复用的理论根据:A,C 数据支路是根据编码模式是否采用 Intra 16×16 而区分,两种模式不可能共存于一个宏块,因此 A,C 数据支路共用缓存 1 是允许的;为了进一步提高资源利用率,我们将 A/C 与 E 支路数据也进行了复用,原因在于当 E 中的色度 AC 分量到达缓存时,亮度数据已经处理完毕,此时多路选择器将 E 选通而禁止 A,C 将不会导致缓存内数据被破坏(已无数据)。基于以上理由,我们将深度为 27 的缓存 1 复用于 A,C,E 支路。同理,B,D 亦可共用深度为 16 的寄存器组缓存 2。文献[3]中的方案实现类似功能却用了 851 个 LUT(look-up-table),可见本方案明显减少了缓存消耗。

4 反 DCT 变换结构

根据 H.264/AVC 标准[1]，反 DCT 变换所采用的结构为 $\boldsymbol{X}=\boldsymbol{CYC}^{\mathrm{T}}$。利用矩阵变换：

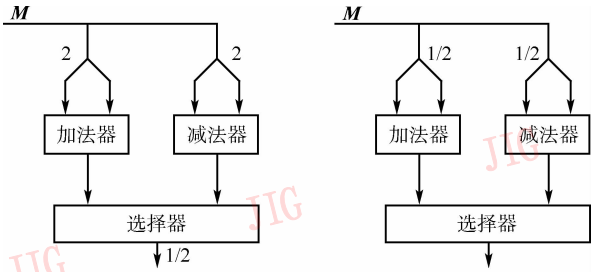
$$\boldsymbol{X}=\boldsymbol{CYC}^{\mathrm{T}}\Rightarrow\boldsymbol{M}=\boldsymbol{CY}^{\mathrm{T}},\boldsymbol{X}=\boldsymbol{CM}^{\mathrm{T}} \tag{1}$$

式中，$\boldsymbol{Y}$ 是 IDCT 的输入，$\boldsymbol{C}$ 为 IDCT 的系数矩阵，$\boldsymbol{M}$ 是本设计的中间数据，$\boldsymbol{X}$ 为反变换的最终结果。将式(1)中的 $\boldsymbol{M}$ 和 $\boldsymbol{Y}$ 当作输入，那么求 $\boldsymbol{X}$ 就简化成了两次相同的矩阵运算。这意味着我们在硬件设计时可以考虑只采用一个矩阵运算单元 $\boldsymbol{X}=\boldsymbol{CM}^{\mathrm{T}}$，通过复用该单元，将所需的资源减少了一半。另外根据变换的对称性，我们可以将 $\boldsymbol{X}=\boldsymbol{CM}^{\mathrm{T}}$ 进一步分解[5]：

$$\begin{aligned}\begin{bmatrix}X_0\\X_1\end{bmatrix}&=\begin{bmatrix}1&1\\1&-1\end{bmatrix}\begin{bmatrix}M_0\\M_2\end{bmatrix}+\frac{1}{2}\begin{bmatrix}2&1\\1&-2\end{bmatrix}\begin{bmatrix}M_1\\M_3\end{bmatrix}\\&=A\begin{bmatrix}M_0\\M_2\end{bmatrix}+\frac{1}{2}B\begin{bmatrix}M_1\\M_3\end{bmatrix}\end{aligned} \tag{2}$$

$$\begin{aligned}\begin{bmatrix}X_3\\X_2\end{bmatrix}&=\begin{bmatrix}1&1\\1&-1\end{bmatrix}\begin{bmatrix}M_0\\M_2\end{bmatrix}-\frac{1}{2}\begin{bmatrix}2&1\\1&-2\end{bmatrix}\begin{bmatrix}M_1\\M_3\end{bmatrix}\\&=A\begin{bmatrix}M_0\\M_2\end{bmatrix}-\frac{1}{2}B\begin{bmatrix}M_1\\M_3\end{bmatrix}\end{aligned}$$

根据式(2)，可以将 4×4 的矩阵运算简化为 2×2 的运算，从而减少硬件消耗。本系统 IDCT 设计与文献[4]的不同之处在于矩阵运算单元，如图 2 所示。



(a) 文献[4]矩阵运算单元架构　(b) 本设计矩阵运算单元架构

图 2　矩阵运算单元架构

Fig. 2　The architecture of matrix unit

5 Hadamard 反变换结构

文献[1]中，Hadamard 反变换由矩阵运算 $\boldsymbol{X}=\boldsymbol{HYH}^{\mathrm{T}}$ 得到。事实上，第 4 节提供的反 DCT 变换结构仍然可以应用于 Hadamard 反变换，但是由于 Hadamard 反变换矩阵系数的特点，我们决定采用一种更为高效的结构。将 Hadamard 反变换矩阵展开，得到表 2 所示的四级运算结构。

表 2　展开后的 Hadamard 反变换

Tab. 2　The expanded inverse Hadamard transform

$a_0=Y_0+Y_4$	$b_0=a_0+a_1$	$c_0=b_0+b_1$	$X_0=c_0+c_1$
$a_1=Y_8+Y_{12}$	$b_1=a_2+a_3$	$c_1=b_2+b_3$	$X_1=c_0-c_1$
$a_2=Y_1+Y_5$	$b_2=a_4+a_5$	$c_2=b_0-b_1$	$X_2=c_2-c_3$
$a_3=Y_9+Y_{13}$	$b_3=a_6+a_7$	$c_3=b_2-b_3$	$X_3=c_2+c_3$
$a_4=Y_2+Y_6$	$b_4=a_0-a_1$	$c_4=b_4+b_5$	$X_4=c_4+c_5$
$a_5=Y_{10}+Y_{14}$	$b_5=a_2-a_3$	$c_5=b_6+b_7$	$X_5=c_4-c_5$
$a_6=Y_3+Y_7$	$b_6=a_4-a_5$	$c_6=b_4-b_5$	$X_6=c_6-c_7$
$a_7=Y_{11}+Y_{15}$	$b_7=a_6-a_7$	$c_7=b_6-b_7$	$X_7=c_6+c_7$
$a_8=Y_0-Y_4$	$b_8=a_8-a_9$	$c_8=b_8+b_9$	$X_8=c_8+c_9$
$a_9=Y_8-Y_{12}$	$b_9=a_{10}-a_{11}$	$c_9=b_{10}+b_{11}$	$X_9=c_8-c_9$
$a_{10}=Y_1-Y_5$	$b_{10}=a_{12}-a_{13}$	$c_{10}=b_8-b_9$	$X_{10}=c_{10}-c_{11}$
$a_{11}=Y_9-Y_{13}$	$b_{11}=a_{14}-a_{15}$	$c_{11}=b_{10}-b_{11}$	$X_{11}=c_{10}+c_{11}$
$a_{12}=Y_2-Y_6$	$b_{12}=a_8+a_9$	$c_{12}=b_{12}+b_{13}$	$X_{12}=c_{12}+c_{13}$
$a_{13}=Y_{10}-Y_{14}$	$b_{13}=a_{10}+a_{11}$	$c_{13}=b_{14}+b_{15}$	$X_{13}=c_{12}-c_{13}$
$a_{14}=Y_3-Y_7$	$b_{14}=a_{12}+a_{13}$	$c_{14}=b_{12}-b_{13}$	$X_{14}=c_{14}-c_{15}$
$a_{15}=Y_{11}-Y_{15}$	$b_{15}=a_{14}+a_{15}$	$c_{15}=b_{14}-b_{15}$	$X_{15}=c_{14}+c_{15}$

表中，$\boldsymbol{Y}_i$ 为本模块输入，$\boldsymbol{a}_i$，$\boldsymbol{b}_i$，$\boldsymbol{c}_i$ 为流水线中间变量，$\boldsymbol{X}_i$ 为本模块输出。从表 2 看出，可以使用 4 级流水线结构的硬件实现方案。该设计最大的特色在于采用了时分复用的存储器管理方式。这样的结构有两个优点：

(1) 低资源消耗　在文献[3]中，每一级流水线均需要 16 个寄存器组成的寄存器组，共需 64 个寄存器；本设计中只需要 4+8+2+1=15 个寄存器即可完成。

(2) 低延时　文献[3]的方案每一级均需 16 个时钟的延时，总延时 4×16=64，而本设计延时仅为 18。

参考表 2，以第 1 级为例来说明本设计中所采用的时分复用存储器管理的过程：注意到，第 1 级输出均由时钟间隔为 4 的输入经过算术运算得到，例如：$\boldsymbol{a}_0=\boldsymbol{Y}_0+\boldsymbol{Y}_4$ 等。因此对于流水线的第 1 级结构，本设计采用一个深度为 4 的寄存器组。图 3 显示了第 1 级中的数据流情况：

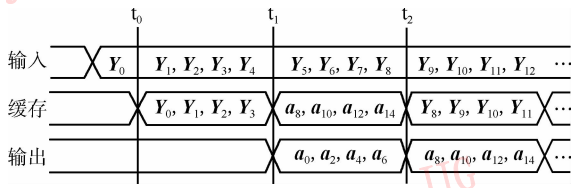


图 3　第 1 级数据流程图

Fig. 3　Data flow in the first stage of pipeline

图 3 中,$\boldsymbol{Y}_i$,$\boldsymbol{a}_i$ 分别为第 1 级输入,输出。首先将 $\boldsymbol{Y}_i$,$i=0,\cdots,3$ 存入寄存器组,在 t_1 时刻,$\boldsymbol{Y}_4$ 进入该子系统,此时可以开始进行计算,经过加法器和减法器处理的计算结果分别进入不同的数据处理过程:加法器结果:$\boldsymbol{a}_0$,$\boldsymbol{a}_2$,$\boldsymbol{a}_4$,$\boldsymbol{a}_6$ 直接作为第 1 级输出;而此时 $\boldsymbol{Y}_0,\cdots,\boldsymbol{Y}_3$ 将不再使用,因此可以将该存储器用于保存减法器的运算结果 $\boldsymbol{a}_8$,$\boldsymbol{a}_{10}$,$\boldsymbol{a}_{12}$,$\boldsymbol{a}_{14}$,这样就完成了对该寄存器组的第一次时分复用。在 t_2 时刻,为了保持第 1 级流水线输出的连续性,我们将 $\boldsymbol{a}_8$,$\boldsymbol{a}_{10}$,$\boldsymbol{a}_{12}$,$\boldsymbol{a}_{14}$作为第 1 级输出并且同时对该寄存器组执行写入操作,保存输入的 $\boldsymbol{Y}_8$,$\boldsymbol{Y}_9$,$\boldsymbol{Y}_{10}$,$\boldsymbol{Y}_{11}$,由于采用了 Verilog 硬件语言的非阻塞赋值(<=)操作,这样,同时进行写入和读出操作都不会对结果产生干扰。反复进行这一过程,就完成了第 1 级的处理。

由上述过程可见,本设计的第 1 级延时只有 5 个时钟,而文献[3]需要 16 个时钟的延时。因此,本设计具有低延时的特点。第 2,3,4 级的设计也采用了相同的时分复用思想。

6　实验结果

本设计采用 Verilog HDL 硬件设计语言实现,并用 ModelSim 进行功能仿真,仿真数据与 JM 生成的测试向量比较表明,本设计能够正确地对 H.264/AVC 宏块进行解码。以 Xilinx Virtex2:XC2V6000 为目标器件,使用 Synplify 工具进行综合验证并选中 FSM Explorer。结果显示,在目标综合频率 100MHz 情况下,最高工作频率可达 123.7MHz。综合结果如表 3 所示。

本设计的目标为支持 1080i 50Hz 高清码流的解码,假设码流的帧率为 30fps,并采用 4:2:0 抽样,为了满足实时解码,系统处理速率必须达到 $1\,920\times1\,080\times30\times1.5=93.3$M Samples/s。根据表 3 的综合结果,本设计处理速率为 123.7M Samples/s,达到了设计要求。

表 3　宏块变换综合结果

Tab.3 Macroblock transform synthesis results

性能指标	本设计综合结果
期望频率(MHz)	100
预计频率(MHz)	123.7
预计周期	8.083
总 LUT 量	2 931
总 Flip-Flop 数量	2 060

7　结　论

本文提出了对 H.264/AVC 中宏块变换解码提供完整支持的硬件设计方案,改进后的 T 型结构简化了变换控制流程,并且能够协同各个模块并行运算;通过对寄存器组采用时分复用的方案,减少了本设计的硬件资源消耗并降低了延时。综合结果表明,本设计完全可以满足 1080i 50Hz 的 H.264/AVC 高清解码。

参考文献(References)

1　ISO/IEC14496-10, Draft ITU-T Recommendation H.264 and Final Draft International Standard of Joint Video Specification [S].

2　Dufour Cecile, Le Maguet Yann. MPEG4 versus H.26L document VCEG-L31 [A]. In: Meeting of International Tele Communicotion Union Video Coding Expect Group [C], Eibsee, Germany, 2001.

3　Porto R, Porto M, Bampi S, *et al*. High throughput architecture for forward transforms module of H.264/AVC video coding standard [A]. In:14th IEEE International Conference on Electronics, Circuits and Systems[C], Marrakech, Morocco,2007:150 ~ 153.

4　Liu Ling-zhi, Rong Meng-tian, Jiang Li. A 2D forward and inverse integer transform processor based on highly-parallel architecture[J]. Journal of Shanghai Jiaotong University, 2004, **38**(12): 2048 ~ 2051.[刘凌志,戎蒙恬,姜黎. 一种并行结构的二维正/逆整数变换处理器[J].上海交通大学学报,2004,**38**(12): 2048 ~ 2051.]