文章编号:1001 - 2486(2006)01 - 0068 - 05

一种基于 Wallace 树的分散式 DCT/IDCT 体系结构*

黎铁军,王爱平,李思昆

(国防科技大学 计算机学院,湖南 长沙　410073)

摘　要:提出了一种新的基于 Wallace 树的分散式 DCT/IDCT 体系结构。它不依赖于 ROM 和乘法器,用面积开销低的加法器、移位器和 4 - 2 压缩器,实现了乘法密集的 DCT/IDCT 算法。该体系结构在 SMIC 0.18μm 工艺上进行了设计和综合,可以达到 100Mpixels/s 的吞吐率,只消耗了 36 141 个晶体管和 1024bits 转换存储器,时序—面积性能较已有的体系结构有了显著的改善。

关键词:MPEG;DCT;IDCT;Wallace 树;体系结构

中图分类号:TP391　　**文献标识码**:A

A New DCT/IDCT Architecture Based on Wallace Trees

LI Tie-jun, WANG Ai-ping, LI Si-kun

(College of Computer, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: This paper proposes a Wallace-tree based new DCT/IDCT architecture, which does not depend on ROM and multipliers any more, but utilizes low cost adders, shifts and 4 - 2 compressors to implement the multiplication-dense DCT/IDCT algorithm. Designed and synthesized by SMIC 0.18 μm technology, the architecture could achieve 100Mpixels/sec throughout with cost of only 36 141 transistors and 1024 bits transform memory. As a result, a far better performance of time series and space is obtained than that of the existing architecture.

Key words: MPEG; DCT; IDCT; Wallace tree; architecture

在视频图像处理中,离散余弦变换(DCT)及其反变换(IDCT)能够有效地减少空间冗余,实现数据压缩,因此它们已经成为众多国际标准(如 MPEG、JPEG 和 H.26x)[1] 的核心部分。然而,DCT/IDCT 的计算复杂度很高,是实时系统的一个性能瓶颈。为满足实时性的要求,有必要用硬件的方法实现 DCT/IDCT 算法[2-5]。

1　硬件实现的数学推导

1.1　DCT/IDCT 计算

二维 DCT/IDCT 可以采用行列分解的算法实现,这种方法规则性强,宜于用硬件实现。在该方法中,二维 DCT 和 IDCT 变换被分解为两个一致的一维 DCT 和 IDCT 变换,公式如下:

$$Y(k) = \sqrt{\frac{2}{N}} \sum_{i=0}^{N-1} C(k)\, x(i) \cos \frac{(2i+1)k\pi}{2N}, \quad x(i) = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} C(k)\, Y(k) \cos \frac{(2i+1)k\pi}{2N} \tag{1}$$

其中 $x(i)$ 是输入数据,$Y(k)$ 是变换数据,i, k 是 $[0, N-1]$ 范围内的整数,C 是 DCT 常系数,定义如下:

$$C(0) = \sqrt{\frac{1}{2}}, \quad C(k) = 1 \quad (k \neq 0) \text{ 且 } k = 1, \cdots, N-1 \tag{2}$$

计算一维 DCT/IDCT 仍然要进行 8×8 的矩阵运算,考虑到运算矩阵是奇行、偶行对称,为了利用这种对称性,可以分解奇行和偶行的运算。分解后的一维 DCT 如下:

* 收稿日期:2005 - 09 - 10
基金项目:国家自然科学基金资助项目(90207019);国家 863 高技术研究发展计划基金资助项目(2002AA1Z1480)
作者简介:黎铁军(1977—),男,博士生。

$$\begin{cases}\begin{bmatrix}Y(0)\\Y(2)\\Y(4)\\Y(6)\end{bmatrix}=\begin{bmatrix}a&a&a&a\\c&f&-f&-c\\a&-a&-a&a\\f&-c&c&-f\end{bmatrix}\begin{bmatrix}x(0)+x(7)\\x(1)+x(6)\\x(2)+x(5)\\x(3)+x(4)\end{bmatrix}\\\begin{bmatrix}Y(1)\\Y(3)\\Y(5)\\Y(7)\end{bmatrix}=\begin{bmatrix}b&d&e&g\\d&-g&-b&-e\\e&-b&g&d\\g&-e&d&-b\end{bmatrix}\begin{bmatrix}x(0)-x(7)\\x(1)-x(6)\\x(2)-x(5)\\x(3)-x(4)\end{bmatrix}\end{cases} \tag{3}$$

分解后的一维 IDCT 如下:

$$\begin{cases}\begin{bmatrix}x(0)\\x(1)\\x(2)\\x(3)\end{bmatrix}=\begin{bmatrix}a&c&a&f\\a&f&-a&-c\\a&-f&-a&c\\a&-c&a&-f\end{bmatrix}\begin{bmatrix}Y(0)\\Y(2)\\Y(4)\\Y(6)\end{bmatrix}+\begin{bmatrix}b&d&e&g\\d&-g&-b&-e\\e&-b&g&d\\g&-e&d&-b\end{bmatrix}\begin{bmatrix}Y(1)\\Y(3)\\Y(5)\\Y(7)\end{bmatrix}\\\begin{bmatrix}x(7)\\x(6)\\x(5)\\x(4)\end{bmatrix}=\begin{bmatrix}a&c&a&f\\a&f&-a&-c\\a&-f&-a&c\\a&-c&a&-f\end{bmatrix}\begin{bmatrix}Y(0)\\Y(2)\\Y(4)\\Y(6)\end{bmatrix}-\begin{bmatrix}b&d&e&g\\d&-g&-b&-e\\e&-b&g&d\\g&-e&d&-b\end{bmatrix}\begin{bmatrix}Y(1)\\Y(3)\\Y(5)\\Y(7)\end{bmatrix}\end{cases} \tag{4}$$

其中

$$[a,b,c,d,e,f,g]=\frac{1}{2}\left[\cos\frac{\pi}{4},\cos\frac{\pi}{16},\cos\frac{\pi}{8},\cos\frac{3\pi}{16},\cos\frac{5\pi}{16},\cos\frac{3\pi}{8},\cos\frac{7\pi}{16}\right]$$

从式(3)、(4)中可以看出,一维 DCT 和 IDCT 的计算中有三个基本的 4×4 矩阵乘计算结构,式(5)给出了这三个矩阵乘。

$$\begin{cases}\begin{bmatrix}Y_0\\Y_1\\Y_2\\Y_3\end{bmatrix}=\begin{bmatrix}b&d&e&g\\d&-g&-b&-e\\e&-b&g&d\\g&-e&d&-b\end{bmatrix}\begin{bmatrix}x_0\\x_1\\x_2\\x_3\end{bmatrix},\\\begin{bmatrix}Y_0\\Y_1\\Y_2\\Y_3\end{bmatrix}=\begin{bmatrix}a&c&a&f\\a&f&-a&-c\\a&-f&-a&c\\a&-c&a&-f\end{bmatrix}\begin{bmatrix}x_0\\x_1\\x_2\\x_3\end{bmatrix},\\\begin{bmatrix}Y_0\\Y_1\\Y_2\\Y_3\end{bmatrix}=\begin{bmatrix}a&a&a&a\\c&f&-f&-c\\a&-a&-a&a\\f&-c&c&-f\end{bmatrix}\begin{bmatrix}x_0\\x_1\\x_2\\x_3\end{bmatrix}\end{cases} \tag{5}$$

为了利用矩阵乘中常系数的特点,最大程度地发掘 DCT/IDCT 乘法运算的相容结构,将式(5)矩阵乘变换为以下形式:

$$[Y_0\quad Y_2\quad Y_3\quad Y_1]=[b\quad d\quad g\quad e]\begin{bmatrix}x_0&-x_1&-x_3&-x_2\\x_1&x_3&x_2&x_0\\x_3&x_2&x_0&-x_1\\x_2&x_0&-x_1&-x_3\end{bmatrix} \tag{6}$$

$$[Y_0\quad Y_2\quad Y_3\quad Y_1]=[c\quad f]\begin{bmatrix}x_1&x_3&-x_1&-x_3\\x_3&-x_1&-x_3&x_1\end{bmatrix}+[a\quad a]\begin{bmatrix}x_0&x_0&x_0&x_0\\x_2&-x_2&x_2&-x_2\end{bmatrix} \tag{7}$$

$$[Y_1 \quad Y_3] = [c \quad f]\begin{bmatrix} x_0 - x_3 & -(x_1 - x_2) \\ x_1 - x_2 & x_0 - x_3 \end{bmatrix}, \quad [Y_0 \quad Y_2] = [a \quad a]\begin{bmatrix} x_0 + x_3 & -(x_1 + x_2) \\ x_1 + x_2 & x_0 + x_3 \end{bmatrix} \tag{8}$$

通过以上变形后，DCT/IDCT 变换最终被简化为三种常系数（$[b\ d\ g\ e]$，$[c\ f]$和$[a\ a]$）的向量内积运算和一些加法运算，其中常系数都为正常数。向量内积的输入顺序也具有一定的循环移数特性，可以通过寄存器间的循环移数实现不同变换输入的产生。为了进一步减少乘法操作，本文将变换系数放大$2\sqrt{2}$倍。因为$2\sqrt{2}a = 1$，所以经过系数变换后可以减少由 a 引入的乘法操作。在二维变换后，通过简单的右移操作就能消除系数放大对结果的影响。

1.2 常系数内积运算的硬件实现推导

为了与输入寄存器循环移数相配合，希望在一个周期内计算出常系数（$[b\ d\ g\ e]$ 和$[c\ f]$）的向量内积，为此，考虑以下形式的内积运算：

$$Y = \sum_{i=1}^{k} A_i X_i \tag{9}$$

其中 A_i 是正值常系数，X_i 是输入变量。常系数为正值，可以用二进制原码表示如下：

$$A_i = \sum_{j=N}^{M} b_i^j 2^j \tag{10}$$

其中 $b_i^j = 0$ 或 1，N 为最小有效位，M 为最大有效位。不像传统的变量分散式算法，本文是将系数进行分散。$(M - N + 1)$是算法的系数精度。通过系数分散，式(9)可以变换为

$$Y = [2^N \quad 2^{N+1} \quad K \quad 2^M]\begin{bmatrix} \sum_{i=1}^{k} b_i^N X_i \\ \sum_{i=1}^{k} b_i^{N+1} X_i \\ M \\ \sum_{i=1}^{k} b_i^{M-1} X_i \\ \sum_{i=1}^{k} b_i^M X_i \end{bmatrix} \tag{11}$$

通过式(11)变形后，常系数的向量内积运算被转换为部分积的移位求和运算，其中部分积加的操作数个数 $k = 4$ 或 $k = 2$，可以通过子表达式共享来减小硬件开销 部分积的移位求和操作数个数等于系数精度$(M - N + 1)$，系数精度即操作数个数大于等于 13。如果用移位器和加法器实现部分积的移位求和运算，不仅硬件开销大，而且在一个周期内也难以完成。本文采用 Wallace 树结构，可以在一个时钟周期内完成部分积的移位求和操作，而且比加法器和移位器节省了大量硬件开销。

2 DABWT 体系结构设计

2.1 总体结构

图 1 给出了本文提出的基于 Wallace 树的分散式 DCT/IDCT 体系结构，它由五级流水栈构成。第一级流水栈是一个串入并出单元（Serial Input Parallel Output，简称 SIPO），将串行输入的数据转换为 8 个一组的并行数据，用于后续的并行处理，而且隔离了输入和处理单元，使得在处理数据的同时还可以接收输入数据。第二级流水栈是桶形移数寄存器组，包括两个单元：偶数桶形单元（Even Barrel，简称 EB）和

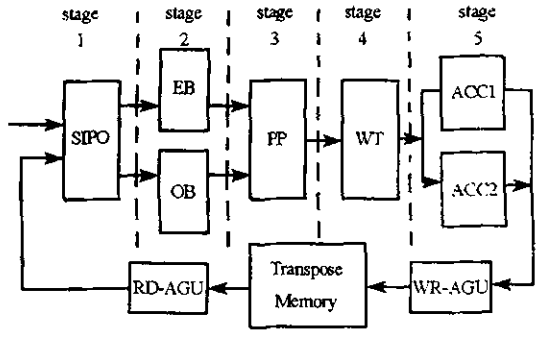

图 1 基于 Wallace 树的分散式 8×8 DCT/IDCT 体系结构

Fig.1 The 8×8 DCT/IDCT distributed architecture based on Wallace trees

奇数桶形单元(Odd Barrel,简称 OB),在一次外部输入后,能自动产生连续的四个偶数和四个奇数变换输入,最大程度地重用了数据。第三级流水栈是部分积产生单元(Partial Product,简称 PP),主要通过共享子表达式产生部分积。第四级流水栈是 Wallace 树单元(Wallace Tree,简称 WT),用于累加部分积,计算向量内积。第五级流水栈是累加(Accumulator,简称 Acc)和写回单元,用于产生和保存最终结果。

2.2 部分积产生

表 1 给出了式(11)中部分积的产生方法。其中的变换系数是放大 $2\sqrt{2}$ 倍后的值,系数精度为 13 位,采用二进制原码表示。为了最小化计算量,通过共享多个部分积计算中的公共部分达到减少总操作数和硬件资源的目的。A ~ I 表示部分积,表 1 中列出了式(3)和(4)中部分积的公共子表达式和替换后的部分积表达式。

通过分析公共子表达式,奇数部分积只需要 5 个加法操作和一个移位操作就能够完成,其中移位操作可以通过硬线连接直接实现,不需要额外的逻辑门。而 DCT 和 IDCT 的偶数部分积的输入形式不同,它们的部分积产生形式也不同,但最大也只需要 5 个加法器就能实现。这样,部分积计算过程总共需要 10 个加法。

表 1 基于公共子表达式的部分积

Tab.1 Partial products based on common subexpressions

输入变量		系数	系数的二进制原码表示	公共子表达式	部分积表达式
even	even0	a	1.0000000000000	A = even0 + even1	A.0000000000000
	even1	a	1.0000000000000		
	even2	c	1.0100111001111	B = even0 − even3, C = even1 − even2, D = B + C, E = even2 + even3, F = even2, G = even3	IDCT:FGF00EFE0GFFFE DCT:B.CB00DBD0CBBBD
	even3	f	0.1000101010001		
odd	odd0	b	1.0110001100010	A = odd0 + odd1, B = odd0 + odd2, C = odd1 + odd2, D = odd1 + odd3, E = odd2 + odd3, F = odd0 << 2, G = odd3, H = odd2, I = odd1	A.GEAFDCBDHFEI0
	odd1	d	1.0010110100000		
	odd2	g	0.0100011010100		
	odd3	e	0.1100100100100		

2.3 Wallace 树单元

图 2 给出了部分积求和的 Wallace 树结构。Wallace 树结构主要被用于实现快速乘法器[7],在本文中被用于汇总式(11)和表 1 中的部分积。Wallace 树由基本的 4 – 2 压缩器(4 – 2 Compressor)构成,可以同时完成四个输入和一个串行进位的加法。与加法树相比,Wallace 树有三个好处:首先,一个基本的 4 – 2 压缩器仅含 6 个逻辑门和最大 3 级异或门延时。其次,Wallace 树的关键路径级数与输入的个数有关,与输入的位宽无关,具有很好的精度扩展性。而加法树的级数不仅与输入个数相关,还和每个输入的位宽相关。最后,Wallace 树比加法树结构规则,适合物理布局布线。例如,对 16 个部分积的加法,每个部分积为 18 位,如果采用加法树,需要 4 级加法器,每级加法器关键路径 5 级门,共 20 级门;但如果采用图 2 所示的 Wallace 树,则只需要 3 级 4 – 2 压缩器和 1 级加法器,共 14 级门。

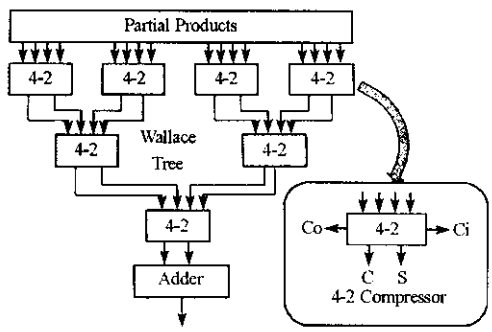

图 2 Wallace 树单元

Fig.2 A Wallace tree

3 模拟和比较

首先用 C 语言对提出的 DCT/IDCT 体系结构进行建模和模拟,以验证位宽精度,以及数据流和控制流的正确性。通过模拟,该体系结构最终采用了 16 位数据宽度(转置矩阵单元宽度)。表 2 列出了该体

系结构精度的模拟结果,可以看出,它符合文献[8]描述的精度规范。

然后采用 Verilog HDL,对所提出的基于 Wallace 树的 DCT/IDCT 体系结构进行了设计实现,在 Modelsim 6.0 环境上对设计进行了模拟验证,并用 Synopsys 公司的 Design Compiler 和中芯国际(SMIC) 0.18μm1P5M 工艺库对设计进行了综合。表 3 给出了模拟和综合后的结果以及与其它三种的 DCT/IDCT 体系结构的比较。从结果来看,本文的设计估计频率可以达到 200MHz,在该频率下可以获得 100Mpixels/s 的吞吐率。该设计的硬件开销只有 36 141 个晶体管(9035 门),比已有的设计显著减少。

表 2 位宽精度标准[8]及本文 DCT/IDCT 模拟结果

Tab.2 The precision analysis of the proposed design for DCT/IDCT

项目	缩写	标准	本文体系结构
Pixel peak error	PPE	≤1	1
Pixel mean square error	PMSE	≤0.06	0.0140
Pixel mean error	PME	≤0.015	0.0121
Overall mean square error	OMSE	≤0.02	0.0021
Overall mean error	OME	≤0.0015	0.000042
All input data is 0	ALL INPUT 0	ALL OUTPUT 0	pass

表 3 本文与已有的 DCT/IDCT 体系结构的比较

Tab.3 The performance comparison of the proposed design and existing DCT/IDCT architectures

	Madisetti[2]	洪明吉[5]	Guo[6]	本文体系结构
实现方法	基于快速算法和硬线乘法器	基于 ROM 分散式算法	基于加法树	基于 Wallace 树的分散式算法
工艺	0.8μm	0.35μm	0.6μm	0.18μm
晶体管数	67 000	47 580	50 800	36 141
存储器	1024bits 转换存储器	1024bits 转换存储器和 512bits 只读存储器	1024bits 转换存储器	1024bits 转换存储器
吞吐率	100 Mpixels/s	35 Mpixels/s	66 Mpixels/s	100 Mpixels/s

4 结论

本文提出了一种新的基于 Wallace 树的分散式 DCT/IDCT 体系结构,它不依赖于 ROM 和乘法器,用面积开销低的加法器、移位器和 4-2 压缩器,实现了乘法密集的 DCT/IDCT 算法。该设计统一了每个行变换的结构,充分地共享了中间数据,加法器的数量只有 17 个。最终的硬件实现表明,该设计消耗了 36 141个晶体管和 1024bits 转换存储器,可以达到 100Mpixels/s 的吞吐率,使之适合于低功耗的实时应用。

参考文献:

[1] Rao K R, Hwang J J. Techniques and Standards for Image, Video and Audio Coding. Englewood Cliffs[M]. NJ: Prentice-Hall, 1996.

[2] Madisetti A, Wilson A N. A 100MHz 2-D 8×8 DCT/IDCT Processor for HDTV Applications[J]. IEEE Trans. on Video Technology, 1995,5(2): 158-164.

[3] Cho N I, Lee S U. DCT Algorithms for VLSI Parallel Implementation[J]. IEEE Trans. on ASSP., 1990,38(1): 121-127.

[4] Chang T S, Kung C S, Jen C W. A Simple Processor Core Design for DCT/IDCT[J]. IEEE Trans. on CSVT, 2000,10(3): 439-447.

[5] 洪明吉. 有效面积使用二维离散余弦转换及反转换架构之超大型集体电路实现[D]. 台湾逢甲大学自动控制工程学系,2001.

[6] Guo J I, Liu C M, Jen C W. The Efficient Memory-based VLSI Arrays for DFT and DCT[J]. IEEE Trans. on CAS-II, 1992,39(10): 723-733.

[7] Sun K H. Design and Implementation of a Module Generator for Low Power Multipliers[D]. Master's Thesis, Division of Electronics Systems, Depart. of Electrical Engineering, Linköping Univ., Sweden, Sept. 25, 2003.

[8] IEEE Standard Specifications for the Implementations of 8×8 Inverse Discrete Cosine Transform[S]. IEEE Std.: 1180-1190.