

二维 DCT 算法及其精简的 VLSI 设计

陈 伟, 卢贵主, 郑灵翔

(厦门大学 电子工程系, 福建 厦门 361005)

摘 要: 采用了快速算法, 并通过矩阵的变化, 得到了一维离散余弦变换 (Discrete Cosine Transform, DCT) 的一种快速实现, 并由此提出一种精简的超大规模集成电路 (Very-large-scale integration, VLSI) 设计架构. 使用了一维 DCT 的复用技术, 带符号数的乘法器设计等技术, 实现了二维 DCT 算法的精简的 VLSI 设计. 实验结果表明, 所设计的二维 DCT 设计有效, 并能够获得非常精简的电路设计.

关键词: DCT; VLSI; 乘法器; 精简

中图分类号: TP301.6

文献标识码: A

文章编号: 1000 - 7180(2008)03 - 0107 - 04

2-D DCT Algorithm and Its Reduced VLSI Design

CHEN Wei, LU Gui-zhu, ZHENG Ling-xiang

(Department of Electronics Engineering, Xiamen University, Xiamen 361005, China)

Abstract: This paper presents a reduced VLSI architecture of one-dimensional discrete cosine transform (1-D DCT) algorithm by introducing a fast algorithm with a matrix transform. At the same time, the design makes use of lots of important technology, such as the reusing of 1-D DCT, the canonic signed digit multiplier design, and so on. A 2-D DCT VLSI design is finally realized. Experiments show that the design of the 2-D DCT is effective and can reach a reduced circuit design.

Key words: DCT; VLSI; multiplier; reduced

1 引言

在数码相机的图像压缩方面, 2-D DCT 作为关键模块使用. 另外, 在数字图像处理、实际的视频压缩中, 2-D DCT 也有着极其重要的作用. 目前的视频编码国际标准的基本方法, 都是采用了基于 DCT 变换的混合编码方法.

DCT 算法的 VLSI 设计一直是一个研究的热点. 人们提出了各种各样的快速算法来实现, 比较著名的有 Lee 算法、Chen 算法、AA & N 算法. 它们对普通的行列分解算法作了一些改进. 这些算法可以采取并行化 (Concurrent Execution), 流水线 (Pipelining Execution) 处理技术, 在视频压缩领域得到应用. 另外, 采用了 1-D DCT 复用设计方法, DCT 算法也应用于数码相机的图像压缩单元^[1].

在文献[2]中提到的一种采用查找表的 VLSI

设计, 但是占用资源比较多. 文中采用了 Li 算法^[3], 通过矩阵形式变化, 提出了一维 DCT 的 VLSI 的设计架构. 这种架构相对于其他算法的 VLSI 架构, 结构更加精简, 易于实现. 由于算法中的乘法可以改写为固定乘数形式, 从而可以用一种精简的带符号数 (Canonic Signed Digit, CSD) 乘法器来设计. 同时, 通过 1-D DCT 的复用又得到了 2-D DCT 的设计架构.

2 1-D DCT 的快速算法及其精简 VLSI 架构

2.1 一维 DCT 快速算法

以 $x(n)$ 表示 N 个有限的一维实数信号系列的集合, $n = 0, 1, \dots, N - 1$, 则一维 DCT 为 $Y(k)$, $k = 0, 1, \dots, N - 1$.

在 Li 算法^[3]中, DCT 的计算可以分离成计算奇数项的离散余弦变换值和偶数项的离散余弦变换

收稿日期: 2007 - 05 - 15

值. N 项数据的偶数项离散余弦变换值进一步通过变换, 转化为求 $N/2$ 项数据的离散余弦变换, 因而进一步可以转为计算奇、偶项的离散余弦变换值.

对应于 $N = 8$ 的情况, 由输入 $x(i)$ 与 $y(i)$ 的关系:

$$y(i) = x(n_i) - x(n_{(N/2)+i}),$$

可以得到如下的对应关系:

$$\begin{bmatrix} Y(1) \\ Y(3) \\ Y(7) \\ Y(3) \end{bmatrix} = \begin{bmatrix} C(0) & C(1) & C(2) & C(3) \\ C(1) & C(2) & C(3) & C(4) \\ C(2) & C(3) & C(4) & C(5) \\ C(3) & C(4) & C(5) & C(6) \end{bmatrix} \begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ y(3) \end{bmatrix} \quad (1)$$

式中, $C(i) = \cos((2 / 4 N) 3^i)$, 根据 Lemma1^[1], 有 $C((N/2) + i) = -C(i)$, 变化上式得到

$$\begin{bmatrix} Y(1) \\ Y(3) \\ Y(7) \\ Y(5) \end{bmatrix} = \begin{bmatrix} C(0) & C(1) & C(2) & C(3) \\ C(1) & C(2) & C(3) & -C(0) \\ C(2) & C(3) & -C(0) & -C(1) \\ C(3) & -C(0) & -C(1) & -C(2) \end{bmatrix} \begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ y(3) \end{bmatrix} \quad (2)$$

进一步变化得到

$$\begin{bmatrix} Y(1) \\ Y(3) \\ Y(7) \\ Y(5) \end{bmatrix} = \begin{bmatrix} y(0) & y(1) & y(2) & y(3) \\ -y(3) & y(0) & y(1) & y(2) \\ -y(2) & -y(3) & y(0) & y(1) \\ -y(1) & -y(2) & -y(3) & y(0) \end{bmatrix} \begin{bmatrix} C(0) \\ C(1) \\ C(2) \\ C(3) \end{bmatrix} \quad (3)$$

其中

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ y(3) \end{bmatrix} = \begin{bmatrix} x(0) - x(7) \\ x(1) - x(6) \\ x(4) - x(3) \\ x(2) - x(5) \end{bmatrix},$$

$$C(i) = \sqrt{\frac{2}{8}} \cos((2 / 4 N) 3^i), N \text{ 取 } 8, i = 0, 1, 2, 3.$$

偶数项部分有

$$\begin{bmatrix} Y(2) \\ Y(6) \end{bmatrix} = \begin{bmatrix} y(0) & y(1) \\ -y(1) & y(0) \end{bmatrix} \begin{bmatrix} C(0) \\ C(1) \end{bmatrix} \quad (4)$$

其中

$$\begin{bmatrix} y(0) \\ y(1) \end{bmatrix} = \begin{bmatrix} (x(0) + x(7)) - (x(3) + x(4)) \\ (x(1) + x(6)) - (x(2) + x(5)) \end{bmatrix},$$

$$C(i) = \sqrt{\frac{2}{8}} \cos((2 / 4 N) 3^i), N \text{ 取 } 4, i = 0, 1;$$

$$\begin{bmatrix} Y(0) \\ Y(4) \end{bmatrix} = \frac{1}{2\sqrt{2}} \begin{bmatrix} y(0) & y(1) \\ y(0) & -y(1) \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad (5)$$

其中

$$\begin{bmatrix} y(0) \\ y(1) \end{bmatrix} = \begin{bmatrix} (x(0) + x(7)) + (x(3) + x(4)) \\ (x(1) + x(6)) + (x(2) + x(5)) \end{bmatrix}.$$

2.2 一维 DCT 精简的 VLSI 架构

对于 $N = 8$ 的 DCT, 由式 (3) ~ (5), 设计中提出一种 VLSI 设计架构, 如图 1 所示.

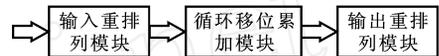


图 1 $N = 8$ 的一维 DCT 架构

各个模块的作用: 输入重排列模块完成数据的重新排列、加减; 循环移位累加模块完成数据的循环寄存, 并且具有反相器单元, 输出重排列模块对输出的数据顺序输出. 输入的数据 $x(n)$, 其中 $n = 0 \dots 7$, 经过模块处理, 8 个周期后依次输出 $Y(0)$ 、 $Y(4)$ 、 $Y(2)$ 、 $Y(6)$ 、 $Y(1)$ 、 $Y(3)$ 、 $Y(7)$ 、 $Y(5)$.

其中的关键模块是循环移位累加模块, 如图 2 所示.

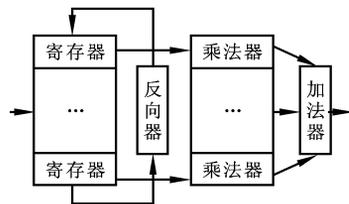


图 2 移位累加模块的架构

在实际的电路中, 需要一些控制端对数据流加以控制, 以完成所需要的移位并反向的工作. 输入数据在计数器的控制下, 在第 6 个时钟周期开始输入经过前处理的数据, 经过 4 个时钟周期完成 4 个数据的输入寄存. 在接下来的 4 个时钟周期内, 模块的控制输入端中断输入数据. 同时存储的 4 个数据完成移位并反向的功能. 从数据的移入到数据完成循环反向共需要 8 个时钟周期. 每 8 个时钟周期的前 4 个时钟周期完成数据的输入, 后 4 个时钟周期完成数据的循环移动.

用这种架构实现一个 8 点的 1-D DCT 运算, 只需要 7 个乘法器和 16 个加法器. 因而, 这种 VLSI 架构具有精简的特点.

2.3 精简的乘法器设计

在信号处理中, 经常要用到乘法器和加法器等功能单元. 而乘法器往往占用的芯片资源比较多. 因

而设计 VLSI 时,在不特别影响性能的前提下,往往优先采用需要较少的乘法器的算法.同时,乘法器的设计算法也多种多样.通常一方面考虑的是乘法器的高速性能,这种的实现可以采用并行执行(Concurrent Execution),以及流水线执行(Pipelining Execution)来实现.另一方面考虑的是功能单元的精简的架构.在本设计中考虑后者.

乘法器的设计可以分为固定乘数的乘法器和未知系数的乘法器.实现方式上可以用移位相加乘法器、Baugh Wooley 补码阵列乘法器、分布式算法(Distributed Arithmetic, DA)乘法器、Sram 型乘法器等.

在设计中,考虑到 DCT 算法所使用的乘法器的乘数是固定不变的.文中采用了更为精简的带符号数(Canonic Signed Digit, CSD)乘法器^[4].数的 CSD 表示是一种用最少的非零比特来表示符号数的编码技术.

例如:二进制数 $1111 * A$,使用移位相加乘法器实现,共使用了 4 个移位单元.当采用 CSD 时,1111 可以表示为 10001,相当于 10000-00001,优化了两个移位寄存器.CSD 乘法器对于小数的情况,同样适用.对于精度很高的设计,这种架构可以有效地设计出精简的电路.

3 2-D DCT 的 VLSI 架构

目前各种优化的二维 DCT 快速算法,大都是基于行列分解的原理来实现.设输入为 f 的 $N \times N$ 的 DCT,其输出为 F .其中中间矩阵 Y 的列等于 f 的行的一维 DCT 的输出.二维 DCT 变换可以通过一维变换来计算,即先沿着 f 的行进行一维 DCT 计算,再沿着 Y 的转置的行进行一维 DCT 运算^[5].

文中提出了一种二维 DCT 的 VLSI 架构,如图 3 所示.通过一个控制单元控制输入的数据读取,以及各个模块的时序.一个 8×8 单元的数据输入时,在控制单元的控制下,模块先对每一个列的 8 个数据进行一维的 DCT 运算,经过 8 个时钟周期就可以得到一个列的 DCT 变换后的输出数据.之后,数据经过转置 RAM 模块的处理,实现了 8×8 的数据行列对调,这个过程需要经过 64 个时钟周期.转置 RAM 输出的数据,通过选择器,作为一维 DCT 的输入,进行了行的 DCT 变换.

转置 RAM 模块采用普通的 RAM 实现.它的读写共用一个地址,并且有一个读写控制信号.实现转置的原理是,读控制信号有效时,顺序输入 n ($0 \dots 63$) 作为地址;写控制信号有效时,输入地址为

$8m, 8m+1, 8m+2, 8m+3, 8m+4, 8m+5, 8m+6, 8m+7$ (其中 $m = 0 \dots 7$),那么经过 64 个时钟周期之后,从转置模块输出的数据就实现了转置.

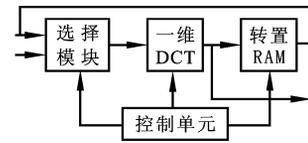


图 3 2-D DCT 的架构

4 实验结果及其分析

4.1 实验结果

在设计中,采用了 Verilog 硬件描述语言对各个模块进行描述.集成开发环境采用 ISE7.1i,仿真工具采用 ModelSim SE 6.1b.如图 4 所示,经过 2DCT 处理之后的输出结果为 dout,输出的顺序为 $Y(0)$ 、 $Y(4)$ 、 $Y(2)$ 、 $Y(6)$ 、 $Y(1)$ 、 $Y(3)$ 、 $Y(7)$ 、 $Y(5)$.经过重排之后的输出为 dout2out,标志信号为 done.硬件模块的输出结果与 Matlab 软件计算的结果进行比较,进一步验证了设计的正确性.

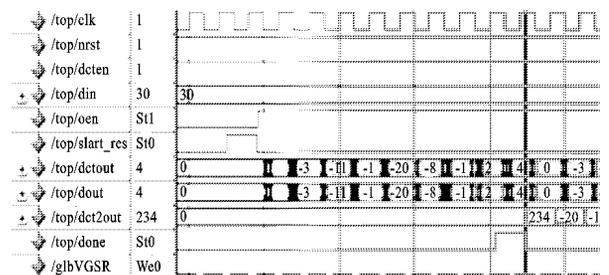


图 4 2-D DCT 的仿真结果

4.2 结果分析

采用 Spartan-3FPGA 系列的 XC3S1000 芯片.在 ISE7.1i 中完成设计模块的综合、布局布线,以及实现.从综合报告中,可以看出所使用的资源的情况:芯片使用 Slices 数目为 866,占总数的 11%;使用 Slice Flip Flops 数目为 474,占总数的 3%;使用 input LUTs 数目为 1535,占总数的 9%;使用 bonded IOBs 数目为 28,占总数的 16%.

详细的资源统计结果如下:

- (1) Design Statistics: IOs: 28;
- (2) Cell Usage Statistics: BELS: 2115; FlipFlops/Latches: 474; RAMS: 1; Clock Buffers: 1; IO Buffers: 27;
- (3) Macro Statistics: RAM: 1; Registers: 58; Multiplexers: 1; Tristates: 1; Xors: 27; Adders/ Subtractors: 22.

综合后可以看到,芯片的最高工作频率达到了 74.16MHz. 对于 4CIF 格式的视频输入,一帧图像的大小为 740 × 576,设帧频为 25 帧/s,每 64 个输入信号之间有一个间隔,那么对所设计的芯片的最低的时钟频率要求:

$$f_{\min} = \frac{740 \times 576}{8 \times 8} \times 1.5 \times (64 + 1) \times 25 \\ = 16.23\text{MHz},$$

可见,芯片也可满足一般的视频工作的需要.

5 结束语

文中研究了 2-D DCT 的 VLSI 架构,用 1-D DCT 的复用进行设计. 最终,在 FPGA 上得到了验证,并获得了很好的性能.

另外,在 1-D DCT 的实现中,设计选择了精简的快速算法并提出了一种的 VLSI 实现架构.

由于架构具有结构简单,设计精简的特点,因而易于复用. 它经过流水化或者并行化处理后,很容易就改造成一种高速的 2D-DCT 模块,从而可以在视频压缩领域得到应用.

参考文献:

- [1] Adamo O B, Mohanty S P, Kougiianos E, et al. VLSI architecture and FPGA prototyping of a digital camera for image security and authentication[C]// Proceedings of the IEEE Region 5 Technology and Science Conference. IEEE, San Antonio, 2006:154 - 158.
- [2] 饶海潮,郭立,黄征. IDCT IP 核的 VLSI 结构[J]. 微电子学与计算机,2004,21(8):132 - 134.
- [3] Li W. A new algorithm to compute the DCT and its inverse[J]. IEEE Transactions on Signal Processing, 1991, 39(6):1305 - 1313.
- [4] Uwe Meyer - Baese. 数字信号处理的 FPGA 实现[M]. 刘凌,译. 北京:清华大学出版社,2006:32.
- [5] 张欣. VLSI 数字信号处理设计与实现[M]. 北京:科学出版社,2003:157.

作者简介:

陈伟 男,硕士研究生. 研究方向为图像编解码的硬件实现、DSP 算法到 FPGA 的映射实现.

卢贵主 女,副教授,硕士生导师. 研究方向为算法到 FPGA 的映射实现、电路与系统研究、集成电路设计.

郑灵翔 男,讲师. 研究方向为嵌入式系统.

(上接第 106 页)

参考文献:

- [1] 张凯,李云岗. 基于 AMBA 总线的 DDR2 SDRAM 控制器研究与实现[J]. 微电子学与计算机,2005,22(9):117 - 122.
- [2] 刘军,郭立,段勃,等. AMBA2.0 总线 IP 核的设计与实现[J]. 微电子学与计算机,2005,22(6):145 - 148.
- [3] AMBA(tm) Specification(Rev 2.0)[EB/OL]. [2006 - 10 - 28] <http://www.arm.com>
- [4] MIPS(r) BusBridge(tm) 2 Module Users Manual[EB/

OL]. [2006 - 10 - 28] <http://www.mips.com>

作者简介:

封君 男,(1973 -),博士. 研究方向为 ASIC 芯片设计和 SoC 芯片设计和验证.

蒲智星 男,(1981 -),硕士研究生. 研究方向为 SoC 芯片设计.

王新安 男,(1963 -),博士,副教授. 研究方向为 SoC 设计和验证.