

文章编号:1671-6833(2008)01-0048-04

## 基于改进的分布式结构的二维 DCT 硬件实现

刘雄飞, 马剑钊, 盛利元

(中南大学 物理科学与技术学院, 湖南 长沙 410083)

**摘 要:** 设计了一个适合 JPEG 图像压缩系统的二维离散余弦变换模块, 采用行列分离的方法, 首先设计了一维余弦离散变换单元, 该单元采用作者提出的改进的有符号分布式算法结构实现, 在硬件实现上可以明显提高吞吐率. 然后复用该单元完成二维离散余弦变换的 FPGA 设计. 在所选器件 EPF10K100EQC208-1 综合后显示, 一维余弦离散变换单元的最高频率可达到 104.17 MHz, 满足 JPEG 图像压缩系统的高吞吐率要求.

**关键词:** 现场可编程门阵列; 离散余弦变换; 联合图像专家组; 分布式算法

**中图分类号:** TN 911.73

**文献标识码:** A

### 0 引言

离散余弦变换(DCT)广泛应用于各种图像压缩编码标准中, 它实现图像数据从空间域到频率域的变换<sup>[1]</sup>, 以减少图像数据的相关性, 以便于后续的压缩处理. DCT 变换的硬件实现一般采用行列分离算法实现<sup>[2]</sup>, 行列分离算法是利用一维 DCT 变换单元实现二维 DCT 变换. 常用的一维快速 DCT 算法有<sup>[3]</sup>: Chen et al 算法、Lee 算法、Loffler 算法. 其中在 Loffler 快速算法中, 计算 8 点一维 DCT 只需要 11 个乘法器和 29 个加法器. 但是这些算法都不能避免乘法器的使用. 为了避免乘法运算, 在设计一维 DCT 变换单元时, 提出了改进的分布式算法结构, 虽消耗的 FPGA 逻辑单元增加了, 但明显提高了系统最高时钟频率.

近几年来, 现场可编程门阵列(FPGA)器件的容量越来越大, 价格越来越低, 利用 FPGA 的硬件可重配置特点, 可以给设计带来很大的灵活性. 笔者采用 FPGA 实现二维 DCT 变换, 采用行列分离的算法, 利用改进的有符号分布式算法实现 8 点一维 DCT 变换单元, 然后复用 8 点一维 DCT 变换单元实现  $8 \times 8$  点的二维 DCT 变换.

### 1 离散余弦变换算法

8 点的一维 DCT 变换定义如下:

$$Y(k) = \frac{1}{2}c(k) \sum_{n=0}^7 x(n) \cos\left[\frac{(2n+1)k\pi}{16}\right],$$
$$k = 0, 1, \dots, 7 \quad (1)$$

式中:  $c(k) = 1, k = 1, 2, \dots, 7; c(0) = \frac{1}{\sqrt{2}}$ .

根据余弦函数的周期性和对偶性, 对上式整理后, 写成矩阵的形式为

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

式中:  $C(i) = \frac{1}{2} \cos\left(\frac{i\pi}{16}\right), i = 1, 2, \dots, 7$ .

式(2)写成矩阵形式为  $Y = CX$ . 如果直接计算上式的话, 可以看出, 8 点一维 DCT 的计算量为 64 次乘法和 56 次加法, 其运算量相当大. 为了减少运算量, 将式(2)进行分解, 得到奇偶分离的两个矩阵:

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

收稿日期: 2007-10-16; 修订日期: 2007-12-11

基金项目: 国家自然科学基金资助项目(60672041)

作者简介: 刘雄飞(1960-), 男, 湖南望城人, 中南大学教授, 硕士, 主要从事数字信号处理的 FPGA 实现, E-mail:

x.f.liu@163.com.

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

分解后的乘法运算量减少了一半,在硬件实现上可以明显节省资源。

8×8 点的二维 DCT 变换定义如下:

$$Y(u, v) = \frac{1}{4}c(u)c(v) \sum_{i=0}^7 \sum_{j=0}^7 x(i, j) \cdot \cos\left[\frac{(2i+1)u\pi}{16}\right] \cos\left[\frac{(2j+1)v\pi}{16}\right] \quad (4)$$

其中:  $i, j, u, v = 0, 1, \dots, 6, 7$ .

$$c(u) = \begin{cases} \frac{1}{\sqrt{2}}, & u = 0 \\ 1, & \text{其它} \end{cases}; \quad c(v) = \begin{cases} \frac{1}{\sqrt{2}}, & v = 0 \\ 1, & \text{其它} \end{cases}$$

可以看出式(4)可分解为两个一维 DCT 变换:

$$Y(u, j) = \frac{1}{2}c(u) \sum_{i=0}^7 x(i, j) \cos\left[\frac{(2i+1)u\pi}{16}\right] \quad (5)$$

$$Y(u, v) = \frac{1}{2}c(v) \sum_{i=0}^7 x(u, j) \cos\left[\frac{(2j+1)v\pi}{16}\right] \quad (6)$$

写成矩阵形式为  $Y = CXC^T$ , 其中式(5)为行变换, 式(6)为列变换。令  $Z = CX^T$ ,  $C$  表示系数矩阵, 那么 8×8 点的二维 DCT 变换写成矩阵形式为:  $Y = CZ^T$ , 这样二维 DCT 变换就可以通过两个一维 DCT 变换单元实现。

## 2 算法的 FPGA 实现

基于以上的分析,2 维 DCT 算法的硬件架构如图 1 所示:

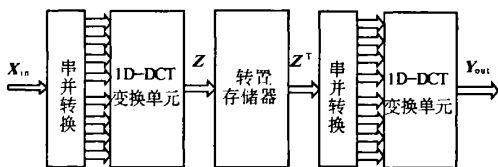


图 1 2D-DCT

Fig.1 2D-DCT

串并转换单元完成串行数据到并行数据的转换,每 8 个有效数据到来后输出 8 个并行数据一次。1D DCT 变换单元完成 8 点的一维 DCT 变换,运算结果串行输出给转置存储器。转置存储器对前面 1D DCT 变换单元的运算结果进行转置。最后一级 1D DCT 变换单元从转置存储器读取转置后的数据进行处理,处理后的结果即为二维 DCT 的变换结果。因此,一维 DCT 的设计成了关键。

### 2.1 一维 DCT 变换的硬件实现

考虑到当数据位宽比较大时,乘法器将消耗

大量的硬件资源以及时间。为了提高运算速度,笔者采用分布式算法<sup>[4]</sup>(distributed arithmetic)实现一维 DCT 变换。考虑式(7):

$$y = \langle c, x \rangle = \sum_{n=0}^{N-1} c(n) \times x(n) \quad (7)$$

其中,  $c(n)$  为已知系数,  $x(n)$  为采样值。在有符号 DA 算法中,  $y$  可以表示为

$$y = -2^B \times x_B(n) + \sum_{b=0}^{B-1} x_b(n) \times 2^b \quad (8)$$

利用分布式算法,式(8)可写为

$$y = \sum_{n=0}^{N-1} c(n) \times -2^B \times x_B(n) + \sum_{b=0}^{B-1} 2^b \times \sum_{n=0}^{N-1} f(c(n), x_b(n)) \quad (9)$$

由于  $x_b(n)$  只取值 0 或 1,所以以上结果只是系数  $c(n)$  的简单相加,然后移位累加。在实现方式上,可利用查找表 LUT 实现,向量  $x = [x_b(0) \ x_b(1) \ \dots \ x_b(N-1)]$  作为查找表的地址。由式(3)可知,每次  $y$  值的计算只涉及到 4 个系数,所以查找表的入口地址为 4 位,总共需要计算 8 个  $Y$  值,1D-DCT 的结构设计如图 2 所示。

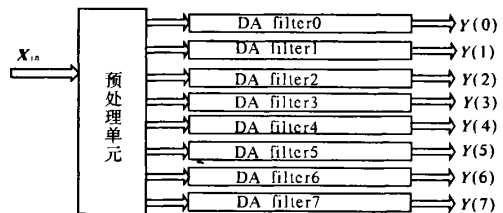


图 2 DA 分布式的 1D-DCT

Fig.2 1D-DCT based on DA

每个 DA filter 都是基于查找表的方式实现的,在具体实现 DA filter 时,考虑到如果输入数据位宽为  $N$  时,则至少需要  $N$  次查询循环。为了减少查询次数,笔者提出了有符号改进型的算法硬件结构,如图 3 所示。改进后,每次只需要查询 4 次就可以得到结果,节省了时钟周期。这里,输入数据宽度为 9 位,最高位的符号位单独作为一个查找表的入口地址。

这样,当串行数据经并行处理后,首先计算  $x(0) + x(7), x(1) + x(6), x(2) + x(6), x(3) + x(4), x(0) - x(7), x(1) - x(6), x(2) - x(6), x(3) - x(4)$  的值作为查找表(DALUT)的入口参数。经 1D-DCT 计算后的  $Y(0), Y(1), Y(2), Y(3), Y(4), Y(5), Y(6), Y(7)$  并行输出。最后在 Altera 公司的 FPGA 开发工具 QuartusII6.0 上综合,所选器件为 EPF10K100EQC208-1。与文献

[2]的综合结果比较,如表1所示.通过比较可以看出,在资源增加较少的情况下,系统时钟提高了近7倍.仿真的结果表明,该1D-DCT单元的功能完全正确.与C语言计算的实际结果对比,如表2所示.

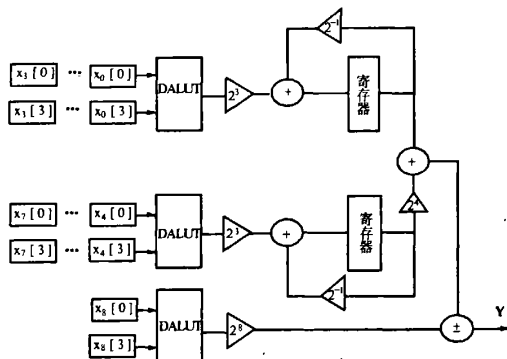


图3 DA filter的实现

Fig. 3 The implementation of DA filter

表1 综合结果对比

Tab.1 The compare of results

方法	逻辑单元	周期/ns	$f_{max}/\text{MHz}$
1D-DCT	2 652	9.60	104.17
1D-DCT <sup>[2]</sup>	2 052	73.2	13.66

表2 计算结果对比

Tab.2 The compare of results

采样值	-76	-73	-67	-62	-58	-67	-64	-55
理论值	-184	-14	-5	-6	7	-3	0	2
仿真值	-185	-15	-6	-7	7	-4	-1	2

从表2中可以看出,仿真结果与实际的DCT值最大误差也不超过 $\pm 1$ .该误差主要是由硬件截位和有限字长引起的.

## 2.2 转置存储器的设计

笔者利用乒乓操作技术实现转置存储器的设计.乒乓操作需要两个RAM,当对一个RAM进行写操作的时候,可以同时另一个RAM进行读操作.假设前64个周期写满第一个RAM,第65个

时钟开始写第二个RAM,同时开始从第一个RAM里读取数据,读取顺序是按列进行的.这样,在花费64个周期后,读完第一个RAM里的数据,此时第二个RAM也已经写满.然后开始读第二个RAM的数据,同时写第一个RAM……,依此进行下去.这样处理的好处,避免了读取数据时不能写入数据,提高了数据吞吐率.基于乒乓操作<sup>[5]</sup>的转置存储器的结构,如图4所示.

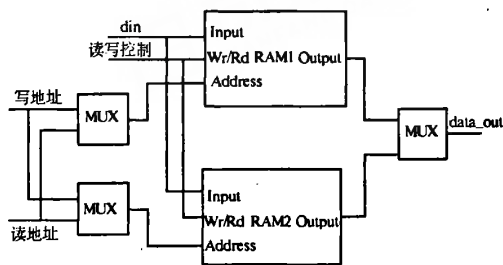


图4 转置存储器的结构

Fig. 4 The architecture of transpose-RAM

## 2.3 2D-DCT的FPGA实现

通过IP核复用的思想,使用两个1D-DCT运算单元实现2D-DCT变换.由于第一个1D-DCT的输出数据的宽度为11位,所以第二个1D-DCT的输入数据位宽应改为11位.整个2D-DCT变换单元如图5所示.接口信号说明如下,CLK:系统时钟;RST:系统复位;din:输入数据;det\_out:输出数据;rdy\_out:输出数据有效.

最后给出测试矩阵A如下:

$$A = \begin{bmatrix} -76 & -73 & -67 & -62 & -58 & -67 & -64 & -55 \\ -65 & -69 & -62 & -38 & -19 & -43 & -59 & -56 \\ -66 & -69 & -60 & -15 & 16 & -24 & -62 & -55 \\ -65 & -70 & -57 & -6 & 26 & -22 & -58 & -59 \\ -61 & -67 & -60 & -24 & -2 & -40 & -60 & -58 \\ -49 & -63 & -68 & -58 & -51 & -65 & -70 & -53 \\ -43 & -57 & -64 & -69 & -73 & -67 & -63 & -45 \\ -41 & -49 & -59 & -60 & -63 & -52 & -50 & -34 \end{bmatrix}$$

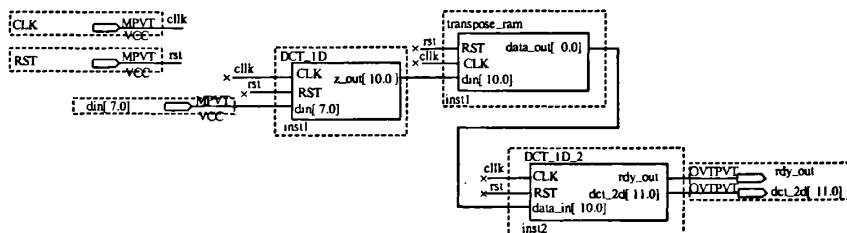


图5 2D-DCT变换单元

Fig. 5 The unit of 2D-DCT

对该设计进行验证. 编写 C 程序计算 A 的二维离散余弦变换, 得到结果

$$B = \begin{bmatrix} -415 & -29 & -62 & -25 & 55 & -20 & -1 & 3 \\ 7 & -21 & -62 & 9 & 11 & -7 & -6 & 6 \\ -46 & 8 & 77 & -25 & -30 & 10 & 7 & -5 \\ -50 & 13 & 35 & -15 & -9 & 6 & 0 & 3 \\ 11 & -8 & -13 & -2 & -1 & 1 & -4 & 1 \\ -10 & 1 & 3 & -3 & -1 & 0 & 2 & -1 \\ -4 & -1 & 2 & -1 & 2 & -3 & 1 & -2 \\ -1 & -1 & -1 & -2 & -1 & -1 & 0 & -1 \end{bmatrix}$$

通过 QuartusII6.0 仿真计算矩阵 A, 得到仿真结果:

$$C = \begin{bmatrix} -419 & -29 & -62 & 25 & 56 & -19 & -2 & 3 \\ 6 & -21 & -62 & 8 & 12 & -7 & -6 & 6 \\ -46 & 8 & 76 & -25 & -31 & 10 & 7 & -5 \\ -50 & 13 & 35 & -16 & -9 & 6 & 0 & 2 \\ 11 & -8 & -13 & -1 & -1 & 1 & -3 & 1 \\ -10 & 2 & 3 & -3 & -1 & 1 & 2 & -1 \\ -4 & -1 & 2 & -2 & 2 & -4 & 1 & -2 \\ 0 & -1 & 0 & -2 & -1 & 0 & 0 & -1 \end{bmatrix}$$

其中, 矩阵 A 作为 2D-DCT 的输入数据 din 串行输入, 矩阵 C 为 det\_2d 的输出结果串行输出, 仿真波形如图 6 所示.

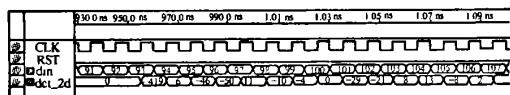


图 6 2D-DCT 仿真结果

Fig. 6 The simulative result of 2D-DCT

### 3 结论

笔者研究了二维 DCT 算法的 FPGA 实现, 并给出了主要模块的硬件结构. 提出了改进的分布式算法结构实现一维 DCT 变换, 大大提高了系统时钟频率. 转置存储器设计采用乒乓操作技术, 可以明显提高数据吞吐量. 在计算一维 DCT 的过程中, 首先要对系数进行整数化处理, 最后在结果中进行硬件截位处理. 该设计方案通过与文献[2]比较, 系统时钟提高了近 7 倍, 可以大大减少图像压缩的时间. 该模块可与 JPEG 编码器的其他模块集成, 实现一个完整的 JPEG 编码器.

### 参考文献:

- [1] XILINX. Virtex-II Platform FPGAS: Complete Data sheet [EB/OL]. [http://www.xilinx.com/application/web\\_de\\_v2/ds031.pdf](http://www.xilinx.com/application/web_de_v2/ds031.pdf), 2004-03-29.
- [2] AGOSTINI, L V, SARAIVA I. Pipelined Fast 2-D DCT Architecture for JPEG Image Compression[C]//Proceedings of Integrated Circuits and Systems Design: 2001, 14th Symposium on, 2001:226-231.
- [3] EL-BANNA H, EL-FATTAH A A. An Efficient Implementation of the 1D DCT using FPGA Technology [C]//Engineering of Computer-Based Systems: Proceedings of 11th IEEE International Conference and Workshop, 2004-12-27:356-360.
- [4] MEYEER-BAESE U. 数字信号处理的 FPGA 实现 [M]. 刘凌, 译. 北京:清华大学出版社, 2006.
- [5] 吴继华, 王诚. Altera FPGA/CPLD 设计: 高级篇 [M]. 北京:人民邮电出版社, 2005.
- [6] 全瑞阳, 耿永军, 朱雪芹. 基于离散余弦变换的数水印算法[J]. 郑州大学学报:理学版, 2005, 37(3):58-80.

## Hardware Implementation of 2D-DCT Based on Improved Distributed Architecture

LIU Xiong-fei, MA Jian-zhao, SHENG Li-yuan

(School of Physical Science and Technology, Central South University, Changsha 410083, China)

**Abstract:** This paper designs two-dimensional discrete cosine transform module for the JPEG image compression system. The architecture of 2D-DCT is based on row-column decomposition method. The paper firstly designs a 1D-DCT unit based on distributed architecture improved by us, which can improve throughput clearly in hardware implementation, and then reuses the unit to complete the FPGA design of 2D-DCT. Synthesis and simulation results of 1D-DCT unit based on EPF10K100EQC208-1 show that the design clock is up to 104.17MHz and can meet JPEG image compression system.

**Key words:** FPGA; DCT; JPEG; distributed algorithm