

# 高吞吐率低时延图像 DCT 处理器设计\*

刘思军<sup>1,2</sup>, 秦明伟<sup>1</sup>, 刘多强<sup>1,2</sup>

(1.西南科技大学 信息工程学院, 四川 绵阳 621010; 2.中国直升机设计研究所, 江西 景德镇 333000)

**摘要:** 针对高分辨率、高帧率图像实时压缩问题, 设计了一种应用于高速图像 JPEG 压缩编码系统的离散余弦变换(DCT)处理器。设计的 DCT 处理器基于 Virtex-7 系列 FPGA, 充分利用并行和流水线处理技术, 采用基于蝶形流图结构的行列分解算法, 实现了快速二维离散余弦变换(2D-DCT)。为了提高数据吞吐率, 设计了双核 DCT 处理单元, 可同时处理 16 个像素, 从整体上提高处理速度和降低时延。板级测试表明, 高速图像 DCT 处理器数据计算结果正确, 在 200 MHz 系统时钟下, 吞吐率高达 3 GB/s, 此时平均每帧图像处理时间不超过 10 ms, 实现了高速图像的实时处理。

**关键词:** 图像压缩; DCT; FPGA; 并行流水结构; 高吞吐率; 低时延

中图分类号: TN911; TP335

文献标识码: A

DOI: 10.16157/j.issn.0258-7998.200814

中文引用格式: 刘思军, 秦明伟, 刘多强. 高吞吐率低时延图像 DCT 处理器设计[J]. 电子技术应用, 2021, 47(9): 69-74.

英文引用格式: Liu Sijun, Qin Mingwei, Liu Duoqiang. Design of high throughput rate low latency image DCT processor[J]. Application of Electronic Technique, 2021, 47(9): 69-74.

## Design of high throughput rate low latency image DCT processor

Liu Sijun<sup>1,2</sup>, Qin Mingwei<sup>1</sup>, Liu Duoqiang<sup>1,2</sup>

(1.School of Information Engineering, Southwest University of Science and Technology, Mianyang 621010, China;

2.China Helicopter Design and Research Institute, Jingdezhen 333000, China)

**Abstract:** Aiming at the high-resolution and high-frame rate image real-time compression problem, a discrete cosine transform processor for high-speed image JPEG compression coding system was designed. The designed discrete cosine transform(DCT) processor is based on the Virtex-7 series FPGA, which makes full use of parallel and pipeline processing technology, and implements a fast two-dimensional discrete cosine transform (2D-DCT) by using a matrix-like decomposition algorithm based on the butterfly flow graph structure. In order to improve the data throughput rate, a dual-core DCT unit is designed to process 16 pixels at the same time, which improves the processing speed and reduces the delay as a whole. The board test shows that the calculation results of high-speed image DCT processor are correct. Under the 200 MHz system clock, the throughput rate is up to 3 GB/s, and the average image processing time per frame is no more than 10 ms, realizing the real-time processing of high-speed images.

**Key words:** image compression; discrete cosine transform(DCT); FPGA; parallel pipeline structure; high throughput; low latency

### 0 引言

DCT 变换运算量大, 是图像处理中计算复杂、耗时长 的运算单元。目前学界提出了两种快速 DCT 变换算 法: 一类是寻求类似于 FFT 的蝶形算法来计算 DCT<sup>[1]</sup>, 另一类是根据 DCT 变换的规律寻求快速算法<sup>[2]</sup>。在第二 类算法中, 最常用的快速算法是行列分解法, 该算法最 初由 Chen 等人提出<sup>[3]</sup>。典型的图像 DCT 处理器的输入 端采用串行输入机制, 在进行 DCT 变换前进行串并转 换<sup>[4]</sup>, 吞吐率不高, 耗时长, 实时性差, 无法应用于高分

辨率、高帧率视觉测量场景。

针对高速大容量图象的处理, 马林<sup>[5]</sup>等人针对 2 048× 2 048 像素、帧频为 150 f/s 的高速图像数据设计了存 储与实时显示系统, 便于延长记录时间和显示; 杨志 勇<sup>[6]</sup>等人针对星载图像高速大容量存储的文件化坏 块管理进行了设计。本文从图像压缩变换角度延长 记录时间和节省数据存储空间, 针对高速风洞试验 中视觉测量设备产生的分辨率可达 5 120 像素×5 120 像素、帧率达 80 f/s 以上的高分辨率、高帧率海量图 像数据的实时压缩问题, 研究设计了一种应用于高速 图像 JPEG 压缩编码的高吞吐率、低延时的 DCT 处 理器。

\* 基金项目: 国家重大科学仪器设备开发基金资助项目(2016YFF0104006); 西南科技大学研究生创新基金项目(20yex0063)

1 DCT 算法

1.1 DCT 变换定义

若  $\{X(m)|m=0, 1, \dots, N-1\}$  是对带宽有效信号  $x(t)$  取样得到的数据序列, 共  $N$  个样值, 其 1D-DCT 定义为<sup>[7]</sup>:

$$Y(u) = c(u) \sqrt{\frac{2}{N}} \sum_{m=0}^{N-1} X(m) \cos \frac{(2m+1)u\pi}{2N} \quad (1)$$

式中,  $Y(u)$  为第  $u$  个离散余弦变换系数,  $u$  为广义频率变量,  $u=0, 1, \dots, 7$ 。当  $u=0, c(u) = \frac{1}{\sqrt{2}}$ ; 其他情况下,  $c(u)=1$ 。

1D-DCT 的定义可推广到 2D-DCT。设一个大小为  $N \times M$  的像素块,  $\{X(i, j)|i=0, 1, \dots, M-1; j=0, 1, \dots, N-1\}$  为二维图像信号数据矩阵, 其二维离散余弦变换定义为<sup>[8]</sup>:

$$Y(u, v) = \frac{2}{\sqrt{MN}} c(u)c(v) \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} X(i, j) \cos \frac{(2i+1)u\pi}{2M} \cos \frac{(2j+1)v\pi}{2N}$$

$$= \sqrt{\frac{2}{N}} c(v) \sum_{j=0}^{N-1} \left[ \sqrt{\frac{2}{M}} c(u) \sum_{i=0}^{M-1} X(i, j) \cos \frac{(2i+1)u\pi}{2M} \right] \cos \frac{(2j+1)v\pi}{2N} \quad (2)$$

1.2 DCT 变换快速算法

若直接按照式(2)计算一个  $8 \times 8$  的图像矩阵, 则一次 2D-DCT 变换共需要 4 096 次乘法和 4 032 次加法, 不论是通过软件还是硬件实现该算法, 都需要消耗大量资源。观察定义式(2)可知, 2D-DCT 变换具有正交性和对称性, 基于此规律, 很多 2D-DCT 快速算法被提出。为方便描述, 将  $N=8$  的 1D-DCT 变换的定义式重写如下:

$$Y(u) = \frac{c(u)}{2} \sum_{i=0}^7 X(i) \cos \frac{(2i+1)u\pi}{16} \quad (3)$$

若将式(3)写成矩阵的形式, 则可表示为:

$$Y = CX \quad (4)$$

$$C(i, j) = \frac{c(i)}{2} \cos \frac{(2j+1)i\pi}{16} \quad (5)$$

其中,  $i=0, 1, 2, \dots, 7; j=0, 1, 2, \dots, 7$ 。

根据式(5)和系数约束条件, 可得到相应的系数矩阵:

$$C(i, j) = \begin{bmatrix} c(4) & c(4) \\ c(1) & c(3) & c(5) & c(7) & -c(7) & -c(5) & -c(3) & -c(1) \\ c(2) & c(6) & -c(6) & -c(2) & -c(2) & -c(6) & c(6) & c(2) \\ c(3) & -c(7) & -c(1) & -c(5) & c(5) & c(1) & c(7) & -c(3) \\ c(4) & -c(4) & -c(4) & c(4) & c(4) & -c(4) & -c(4) & c(4) \\ c(5) & -c(1) & c(7) & c(3) & -c(3) & -c(7) & c(1) & -c(5) \\ c(6) & -c(2) & c(2) & -c(6) & -c(6) & c(2) & -c(2) & c(6) \\ c(7) & c(5) & c(3) & c(1) & c(1) & c(3) & c(5) & c(7) \end{bmatrix} \quad (6)$$

其中,  $c(n) = \frac{1}{2} \cos \frac{n\pi}{16}$ 。

式(6)第一行作了数据上的处理, 根据式(6)和系数约束条件, 可计算出第一行为定值, 刚好等于  $c(4)$ 。

设  $X=[X(0), X(1), X(2), X(3), X(4), X(5), X(6), X(7)]$  为输入的一行信号序列,  $Y=[Y(0), Y(1), Y(2), Y(3), Y(4), Y(5), Y(6), Y(7)]$  为 1D-DCT 变换后输出的一行信号序列。根据式(4)和式(6), 可得:

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

式(7)进行展开计算后, 利用对称性进行奇偶分解, 可得到:

$$\begin{bmatrix} Y(0) \\ Y(2) \\ Y(4) \\ Y(6) \end{bmatrix} = \begin{bmatrix} c(4) & c(4) & c(4) & c(4) \\ c(2) & c(6) & -c(6) & -c(2) \\ c(4) & -c(4) & -c(4) & c(4) \\ c(6) & -c(2) & c(2) & -c(6) \end{bmatrix} \begin{bmatrix} X(0)+X(7) \\ X(1)+X(6) \\ X(2)+X(5) \\ X(3)+X(4) \end{bmatrix} \quad (8)$$

$$\begin{bmatrix} Y(1) \\ Y(3) \\ Y(5) \\ Y(7) \end{bmatrix} = \begin{bmatrix} c(1) & c(3) & c(5) & c(7) \\ c(3) & -c(7) & -c(1) & -c(5) \\ c(5) & -c(1) & c(7) & c(3) \\ c(7) & -c(5) & c(3) & -c(1) \end{bmatrix} \begin{bmatrix} X(0)-X(7) \\ X(1)-X(6) \\ X(2)-X(5) \\ X(3)-X(4) \end{bmatrix} \quad (9)$$

根据上式计算 8 点 1D-DCT 只需要 32 次乘法和 32 次加法。

由  $c(n) = \frac{1}{2} \cos \frac{n\pi}{16} (n=0, 1, 2, \dots, 7)$  代入式(7), 得到系数矩阵  $C$  (保留到小数点四位):

$$C = \begin{bmatrix} 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 \\ 0.4904 & 0.4157 & 0.2778 & 0.0975 & -0.0975 & -0.2778 & -0.4157 & -0.4904 \\ 0.4619 & 0.1913 & -0.1913 & -0.4619 & -0.4619 & -0.1913 & 0.1913 & 0.4619 \\ 0.4147 & -0.0975 & -0.4904 & -0.2778 & 0.2778 & 0.4904 & 0.0975 & 0.4157 \\ 0.2778 & -0.3536 & -0.3536 & 0.3536 & 0.3536 & -0.3536 & -0.3536 & 0.3536 \\ 0.1913 & -0.4904 & 0.0975 & 0.4157 & -0.4157 & -0.0975 & 0.4904 & -0.2778 \\ 0.1913 & -0.4619 & 0.4619 & -0.1913 & -0.1913 & 0.4619 & -0.4619 & 0.0913 \\ 0.0975 & -0.2778 & 0.4157 & -0.4904 & 0.4904 & -0.4157 & -0.2778 & -0.0975 \end{bmatrix}$$

式(8)、式(9)两式展开,可以得到如图1所示的1D-DCT算法流程图。

2D-DCT的矩阵表示为:

$$Y = CXC^T \tag{11}$$

其中,C为DCT的系数矩阵,C<sup>T</sup>为C的转置。令Z=CX,则:

$$Y^T = [CXC^T]^T = CX^TC^T = C[CX]^T = CZ^T \tag{12}$$

所以8×8的图像矩阵的2D-DCT变换分为两步:首先计算Z=CX,得到1D-DCT的结果(逐行计算),第二步是将中间结果Z进行转置后作为第二次1D-DCT变换的输入,再次计算1D-DCT(逐列计算),得到的结果为图像块的2D-DCT变换结果Y的转置Y<sup>T</sup>,基于该思想即可实现DCT变换的行列分解快速算法。

在对2D-DCT变换硬件设计时,1D-DCT一般采用分布式算法(Distributed Arithmetic, DA)<sup>[9-10]</sup>或者流图算法(Flow-Graph Algorithm, FGA)<sup>[11]</sup>实现。DA算法是一种基于查找表和累加器计算结果的位级重构算法,其基本思想是通过查找ROM表,利用ROM和累加器代替乘法器<sup>[12-13]</sup>。DA算法并行处理时需要使用大量ROM资源,查询后经过累加移位等操作,控制变得复杂。由于对ROM的频繁访问和位串行实现算法结构的采用,数据

处理速度受到限制,难以获得很高的时钟频率<sup>[14]</sup>。本文采用基于图1所示的流图算法,计算过程采用流水线处理方式,提升处理速度,减少运算时间。

### 2 架构设计

#### 2.1 一般DCT变换结构

根据行列分解算法可知,将2D-DCT分解为两次1D-DCT变换。行列变换法实现2D-DCT有两种硬件实现结构:分时循环结构和全流水结构。前者行列变换采用同一个1D-DCT变换单元,其结构如图2所示,该结构虽然节约资源,但是限制了2D-DCT变换的速度,实时性差。全流水结构如图3所示,采用两个1D-DCT变换核分别处理行变换和列变换,采用乒乓操作进行中间数据的存储和转置操作,实现流水线处理。

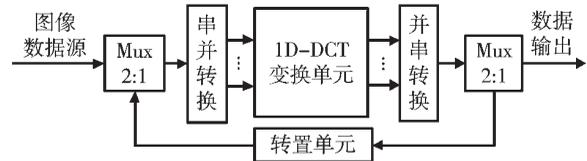


图2 分时循环DCT变换结构

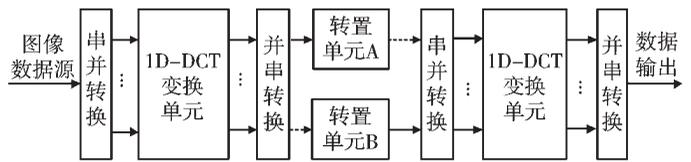


图3 全流水线DCT变换结构

对于这两种DCT变换结构,输入端数据经过8个时钟完成串并转换后,再将8个像素并行送入1D-DCT变

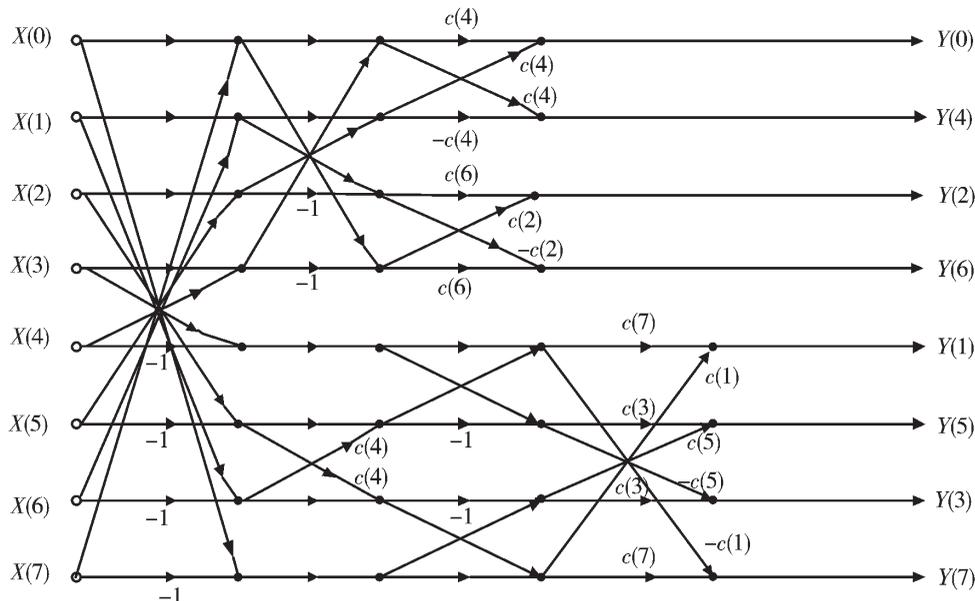


图1 1D-DCT算法流程图

换单元,这样的结构能够处理低速图像,吞吐率受限。例如,一个DCT处理器在200 MHz的工作时钟下,每个时钟输入一个8 bit像素,则吞吐率为200 MB/s。但对于图像源的数据率可达2 GB/s的高速图像处理系统,为了保证图像数据在处理过程中不丢失,则必须保证图像处理系统的吞吐率可达2 GB/s以上,因此必须考虑设计新的DCT处理器结构。

2.2 DCT变换结构设计

整个DCT处理器设计结构如图4所示,主要分为三部分:读控制器、数据预处理、DCT算法处理单元。读控制器从图像行缓存模块FIFO阵列中读取数据,然后传输至DCT算法处理模块完成后续操作。前级的图像行缓存的数据源按照Camera Link通信协议<sup>[15]</sup>从外部模块写入,包含4个视频数据信号:帧有效信号FVAL、行有效信号LVAL、数据有效信号DVAL和数据DATA。从图4中可以看出,整个设计架构是一种并行流水线处理结构,设计了两个DCT变换核,以此提高处理器吞吐率。

3 详细设计

3.1 读控制器设计

读控制器完成从图像缓存器中读取图像数据,读数据过程通过一个状态机实现。图4中的FIFO\_A阵列和FIFO\_B阵列构成乒乓缓存,每个FIFO阵列由8个FIFO组成,用于缓存8行图像。系统启动后,读控制器首先从8个FIFO均已写满的FIFO\_A阵列中的第一个FIFO开始读取一个128 bit有效数据,第二个时钟时在状态机的切换控制下读FIFO\_A阵列的第二个FIFO。以此类推,

读了第8个FIFO后又回到读第1个FIFO,如此循环操作,读空FIFO\_A阵列,接着以同样的方式读FIFO\_B阵列。

3.2 数据预处理

该模块主要功能是将每次输入的128 bit数据分割为16个数据段,每个数据段即为一个8 bit像素值,然后将每个像素值都减去固定值128后转变为有符号数输出,送入DCT变换处理单元。数据传输至DCT处理单元时,将输入数据高位分割出来经过有符号化处理后的8个数据发送至DCT\_A模块,将低位分割出来经过有符号化处理后的8个数据发送至DCT\_B模块,16个数据并行输出。

3.3 DCT变换单元设计

从图4所示的架构图可以看出,DCT变换单元分为DCT\_A和DCT\_B两个结构完全相同的模块,数据处理时相互独立,完全并行操作。因DCT\_A模块与DCT\_B模块结构完全相同,在此仅给出DCT\_A模块的设计结构,如图5所示。在输入数据有效时,同时输入8×8图像块的一行8个有符号数据,然后依次完成一维DCT变换、转置、第二次一维DCT变换。该模块一个时钟并行处理8个输入数据,完成DCT计算后,同时输出8个计算结果。

由于FPGA处理浮点数将消耗大量资源,需要将浮点数转化为定点整数进行运算。在设计中,考虑到计算精度,将变换系数扩大4 096倍,然后取整作为固定系数参与运算。结合式(9),扩大4 096倍并四舍五入取整后的 $c(1)\sim c(7)$ 依次为2 009、1 892、1 703、1 448、1 138、

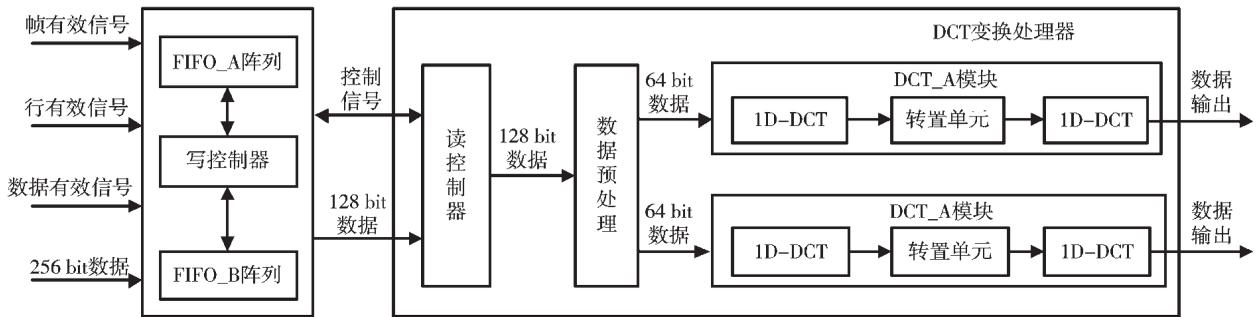


图4 2D-DCT处理器设计结构

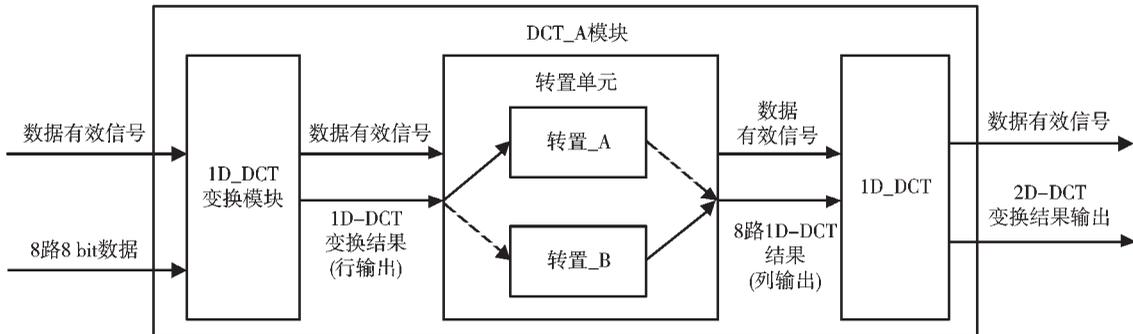


图5 DCT\_A模块设计结构图

784、400。根据式(9)和式(10),采用3个处理步骤完成1D-DCT运算。实现1D-DCT变换的具体过程可以通过表1说明。在计算过程中,一共使用24个乘法器、28个加法器。3个步骤构成流水线处理结构,对于一个 $8 \times 8$ 图像子块的1D-DCT变换,只需要11个时钟即可完成。

表1 1D-DCT变换单元数据处理过程

第一步	第二步	第三步
$add0=X(0)+X(7)$	$q0=sub0 \times c(1)+sub1 \times c(7)$	$Y(0)=q8 \times c(4)+q10 \times c(4)$
$add1=X(3)+X(4)$	$q1=sub0 \times c(7)-sub1 \times c(1)$	$Y(2)=q9 \times c(2)+q11 \times c(6)$
$add2=X(1)+X(6)$	$q2=sub2 \times c(3)+sub3 \times c(5)$	$Y(4)=q8 \times c(4)-q10 \times c(4)$
$add3=X(2)+X(5)$	$q3=sub2 \times c(5)-sub3 \times c(3)$	$Y(6)=q9 \times c(6)-q11 \times c(2)$
$sub0=X(0)-X(7)$	$q4=sub0 \times c(5)+sub1 \times c(3)$	$Y(1)=q0+q2$
$sub1=X(3)-X(4)$	$q5=sub0 \times c(3)-sub1 \times c(5)$	$Y(3)=q5-q6$
$sub2=X(1)-X(6)$	$q6=sub2 \times c(7)+sub3 \times c(1)$	$Y(5)=q4-q7$
$sub3=X(2)-X(5)$	$q7=sub2 \times c(1)-sub3 \times c(7)$	$Y(7)=q1-q3$
	$q8=add0+add1$	
	$q9=add0-add1$	
	$q10=add2+add3$	
	$q11=add2-add3$	

在表1中,add0、add1、add2、add3、sub0、sub1、sub2、sub3为1D-DCT第一步计算的结果,将第一步的结果作为第二步的输入得到第二步的运算结果,即 $q0$ 、 $q1$ 、 $q2$ 、 $q3$ 、 $q4$ 、 $q5$ 、 $q6$ 、 $q7$ 、 $q8$ 、 $q9$ 、 $q10$ 、 $q11$ 的值,再将第二步的结果作为第三步的输入,从而完成1D-DCT的计算。两次1D-DCT变换结构相同,只需要将相关的数据端口和片内寄存器适配对应的数据位宽。为了减小计算误差,第一次1D-DCT变换的结果直接送入转置单元,在第二次1D-DCT变换输出时,通过截断丢弃低位保留高位的操作方法,将运算得到的结果缩小 $2^{24}$ 倍,同时完成四舍五入取整输出。

### 3.4 转置单元设计

通常中间结果用RAM存储起来,然后通过读地址操作完成行列转换,但考虑到处理速度和操作灵活性,本设计利用寄存器组完成转置操作。为了减少图像块处理时间等待消耗,采用乒乓操作技术,设计两个寄存器

组实现连续读写,提高吞吐率和实现低时延。每个寄存器组由64个寄存器组成,构成 $8 \times 8$ 的寄存器阵列,用于缓存 $8 \times 8$ 图像块的1D-DCT变换得到的64个数据结果。8个时钟后,64个寄存器已经完成数据寄存,然后按照 $8 \times 8$ 矩阵的列顺序读出。

## 4 实验与分析

### 4.1 MATLAB平台算法仿真

在进行DCT变换处理器用FPGA进行算法硬件实现前,通过MATLAB完成相应的算法开发和数值分析。在MATLAB程序中,读取一幅分辨率为 $512 \times 512$ 的灰度图像,将无符号的像素值转化为有符号数,经过 $8 \times 8$ 分块,然后根据DCT算法实现DCT变换。为方便显示计算结果,以原始灰度图像前4个 $8 \times 8$ 图像块的DCT变换结果为例,给出MATLAB平台计算DCT变换的结果,如图6所示。

	1	2	3	4	5	6	7	8
1	-923	-139	129	-113	93	-71	47	-24
2	-33	45	-40	34	-27	20	-13	6
3	0	0	0	0	0	0	0	0
4	-56	78	-73	66	-56	44	-30	15
5	0	0	0	0	0	0	0	0
6	22	-31	30	-27	23	-18	13	-6
7	0	0	0	0	0	0	0	0
8	-24	34	-32	29	-24	19	-13	7

图6 MATLAB平台计算DCT变换结果

### 4.2 FPGA板级调试与分析

DCT处理器在Vivado集成环境上设计完成,当仿真验证通过后,下载至FPGA板卡上进行板级调试。为了测试图像处理系统的实际工作情况,DCT处理器与图像采集、缓存和图像 $8 \times 8$ 分块等功能模块进行合并,完成图像采集到变换的处理过程。板级测试时,硬件管理器实时捕获的待测信号如图7和图8所示。从图7可以看出,图像数据从输入DCT单元到输出,只需要16个时钟周期,完成一个 $8 \times 8$ 图像块的2D-DCT变换,只需要24个时钟周期。图8是将数据波形观察窗口进行了放大,便于观察DCT处理器的计算结果。

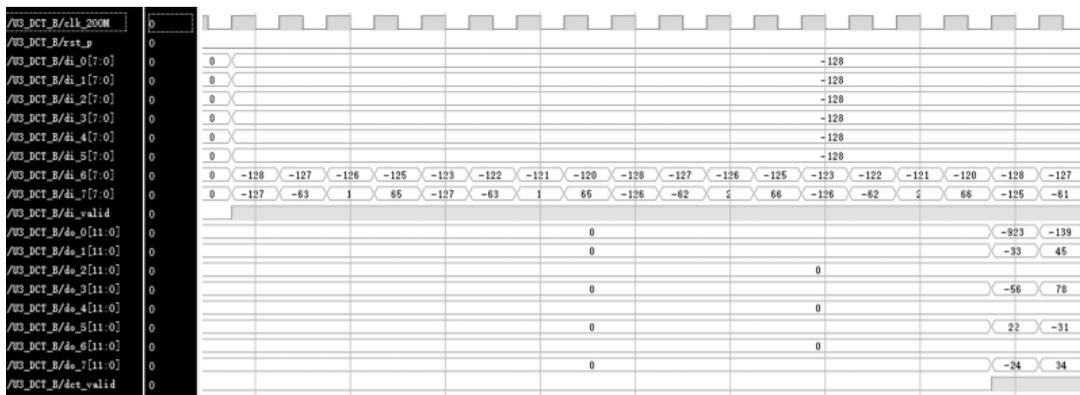


图7 ILA捕获DCT\_B部分数据和时序

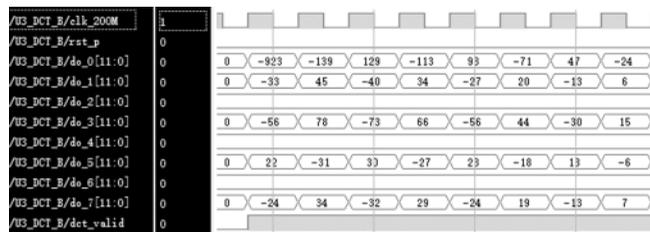


图8 FPGA 计算 DCT 变换结果

从图6和图8中可以看出,DCT变换在MATLAB平台上的仿真结果与在FPGA中实际计算结果完全一致。在设计验证过程中,针对不同的数据源进行重复性测试,实验结果表明,在图像分辨率为 $5120 \times 5120$ 、帧率为 $116 \text{ f/s}$ 、数据率为 $3 \text{ GB/s}$ 的连续帧图像数据源下,处理过程中数据无丢失,平均每帧图像的处理时间不超过 $10 \text{ ms}$ 。针对不同的数据率图像源进行测试,经过多次重复实验,得到DCT变换单元的性能测试结果,如表2所示。

表2 DCT 变换单元性能测试结果

分辨率/ppi	帧率/(f/s)	数据率/(GB/s)	时钟频率/MHz	单帧处理时间/ms
5120×5120	116.1	3	200	8.633
5120×5120	103.8	2.72	200	9.66
5120×5120	85.58	2.24	200	11.7

## 5 结论

本文所设计的基于FPGA的高速图像DCT处理器能够实时处理高分辨率、高帧率时序图像。采用并行流水结构设计,在DCT处理器中设计两个DCT变换核,可同时处理16个像素,实现了数据率高达 $3 \text{ GB/s}$ 图像DCT变换,此时平均单帧图像处理时间在 $10 \text{ ms}$ 以内,满足高吞吐率、高精度、低时延要求。设计的DCT处理器具备灵活性和可移植性,可根据需要选择相应模块集成到图像JPEG压缩编码器中,实现高速图像实时压缩,具有较大的应用价值。

### 参考文献

- [1] NIKARA J A, TAKALA J H, ASTOLA J T. Discrete cosine and sine transforms—regular algorithms and pipeline architecture[J]. Signal Processing, 2006, 86(2): 230–249.
- [2] 唐永亮. 高速高精度离散余弦变换的设计与实现[D]. 天津: 天津大学, 2008.
- [3] CHEN W H, SMITH C H, FRALICK S C. A fast computa-

- tional algorithm for the discrete cosine transform[J]. IEEE Transactions on Communications, 1977, 25(9): 1004–1009.
- [4] 徐洁. 基于FPGA的JPEG图像压缩系统的实现[D]. 大连: 大连理工大学, 2013.
- [5] 马林, 李锦明, 张虎威, 等. 高速CMOS图像存储与实时显示系统设计[J]. 电子技术应用, 2017, 43(6): 7–10, 14.
- [6] 杨志勇, 董振兴, 朱岩, 等. 星载高速大容量存储器文件化坏块管理设计[J]. 电子技术应用, 2017, 43(6): 11–14.
- [7] 张春田, 苏育挺, 张静. 数字图像压缩编码[M]. 北京: 清华大学出版社, 2006.
- [8] DUHAMEL P, GUILLEMOT C. Polynomial transform computation of the 2D-DCT[C]//International Conference on Acoustics, Speech, and Signal Processing. IEEE, 1990: 1515–1518.
- [9] LEE Y, CHEN T, CHEN M, et al. A cost effective architecture for  $8 \times 8$  two-dimensional DCT/IDCT using direct method[J]. IEEE Transactions on Circuits and Systems for Video Technology, 1997, 7(3): 459–467.
- [10] 张慧慧. 基于FPGA的JPEG压缩编码器的设计与实现[D]. 太原: 中北大学, 2017.
- [11] 曹青云. JPEG图像压缩编码器的研究与设计[D]. 上海: 复旦大学, 2013.
- [12] BABIC D. Discrete cosine transform algorithms for FPGA device[M]. Zagerb, 2003.
- [13] YU S, SWARTZLANDER E E. DCT implementation with distributed arithmetic[J]. IEEE Transactions on Computers, 2001, 50(9): 985–991.
- [14] 来蒲军, 桑红石, 陈朝阳. 一种用于JPEG IP核的DCT设计[J]. 计算机与数字工程, 2007, 35(3): 148–150.
- [15] Automated Imaging Association. Specifications of the camera link interface standard for digital cameras and frame grabbers—v2.0[S]. 2012.

(收稿日期: 2020–08–05)

### 作者简介:

刘思军(1993–), 男, 硕士研究生, 主要研究方向: FPGA高速信号处理、图像算法处理。

秦明伟(1979–), 通信作者, 男, 博士, 副教授, 主要研究方向: 软件无线电与自适应通信、高速信号采集与实时处理、复杂算法硬件加速技术, E-mail: qmw\_qyq@163.com。

刘多强(1991–), 男, 硕士研究生, 主要研究方向: 图像算法处理。



扫码下载电子文档

(上接第68页)

构设计[J]. 数据分析与知识发现, 2019, 25(1): 63–70.

(收稿日期: 2020–10–16)

### 作者简介:

仲妍(1982–), 女, 博士, 助理研究员, 主要研究方向: 数

据工程。

钟艳玲(1979–), 通信作者, 女, 硕士, 工程师, 主要研究方向: 数据工程, E-mail: 331965476@qq.com。

郭文思(1983–), 女, 硕士, 工程师, 主要研究方向: 数据工程。



扫码下载电子文档

## 版权声明

经作者授权，本论文版权和信息网络传播权归属于《电子技术应用》杂志，凡未经本刊书面同意任何机构、组织和个人不得擅自复印、汇编、翻译和进行信息网络传播。未经本刊书面同意，禁止一切互联网论文资源平台非法上传、收录本论文。

截至目前，本论文已经授权被中国期刊全文数据库（CNKI）、万方数据知识服务平台、中文科技期刊数据库（维普网）、DOAJ、美国《乌利希期刊指南》、JST 日本科技技术振兴机构数据库等数据库全文收录。

对于违反上述禁止行为并违法使用本论文的机构、组织和个人，本刊将采取一切必要法律行动来维护正当权益。

特此声明！

《电子技术应用》编辑部

中国电子信息产业集团有限公司第六研究所