

## 基于粗粒度可重构阵列结构的多标准离散余弦变换设计

陈锐<sup>①②</sup> 杨海钢<sup>\*①</sup> 王飞<sup>①</sup> 贾瑞<sup>①②</sup> 喻伟<sup>①②</sup>

<sup>①</sup>(中国科学院电子学研究所 北京 100190)

<sup>②</sup>(中国科学院大学 北京 100190)

**摘要:** 在视频信号的编解码流程中, 离散余弦变换(DCT)是一个至关重要的环节, 其决定了视频压缩的质量和效率。针对 $8\times 8$ 尺寸的2维离散余弦变换, 该文提出一种基于粗粒度可重构阵列结构(Coarse-Grained Reconfigurable Array, CGRA)的硬件电路结构。利用粗粒度可重构阵列的可重配置的特性, 实现在单一平台支持多个视频压缩编码标准的 $8\times 8$  2维离散余弦变换。实验结果显示, 这种结构每个时钟周期可以并行处理8个像素, 吞吐率最高可达 $1.157\times 10^9$ 像素/s。与已有结构相比, 设计效率和功耗效率最高可分别提升4.33倍和12.3倍, 并能够以最高30帧/s的帧率解码尺寸为 $4096\times 2048$ , 格式为4:2:0的视频序列。

**关键词:** 粗粒度可重构阵列; 视频压缩; 离散余弦变换; 功耗效率

中图分类号: TN402

文献标识码: A

文章编号: 1009-5896(2015)01-0206-08

DOI: 10.11999/JEIT140104

## Design of Multi-standard Discrete Cosine Transform Based on Coarse-grained Reconfigurable Array

Chen Rui<sup>①②</sup> Yang Hai-gang<sup>①</sup> Wang Fei<sup>①</sup> Jia Rui<sup>①②</sup> Yu Wei<sup>①②</sup>

<sup>①</sup>(Institute of Electronics, Chinese Academy of Sciences, Beijing 100190, China)

<sup>②</sup>(University of Chinese Academy of Sciences, Beijing 100190, China)

**Abstract:** Discrete Cosine Transform (DCT) plays an important role in the codec process of video signals, and has a significant influence on the compression efficiency and quality. In this paper, a Coarse-Grained Reconfigurable Array (CGRA) based hardware architecture is proposed for 8-point 2D DCT. Through the reconfiguration of coarse-grained reconfigurable array, the proposed architecture is capable of supporting  $8\times 8$  2D discrete cosine transform of the multiple video compression coding standards in a single platform. The experimental results show that the proposed architecture is able to parallel process 8 pixels in a cycle, and the throughput achieves up to  $1.157\times 10^9$  pixels per second. The design efficiency and power efficiency is about 4.33 times and 12.3 times higher than existing works respectively. Moreover, the proposed architecture can support real-time decoding of  $4096\times 2048$  at 30 fps (4:2:0) video sequences.

**Key words:** Coarse-Grained Reconfigurable Array (CGRA); Video compression; Discrete Cosine Transform (DCT); Energy efficiency

### 1 引言

在视频序列的编解码流程中, 离散余弦变换(Discrete Cosine Transform, DCT)需要占用大量时间, 如何通过硬件加速执行DCT从而提高编解码的效率显得非常有必要。并且, 随着视频编码标准的增多, 对于单一平台支持多种标准的需求逐渐突显出来, DCT硬件的可重构设计逐渐成为学术研究的热点。然而, 在通过硬件设计加速执行DCT

的同时, 还需要考虑由此带来的能效问题, 特别是对于一些需要电池支持的手持设备, 能效问题更显得突出, 因此, 需要一种兼顾能效与性能的平台, 实现对不同视频压缩编码标准的支持。

视频压缩标准中采用的DCT可以分为两类: 2维离散余弦正变换(2D Forward DCT, 2D-FDCT)和2维离散余弦逆变换(2D Inverse DCT, 2D-IDCT)<sup>[1]</sup>。2D-IDCT/FDCT的整个计算过程一般会被拆成3部分: 基于行/列的1D-IDCT/FDCT、矩阵转置和基于列/行的1D-IDCT/FDCT。根据这3部分的实现方式, 2D-IDCT/FDCT的硬件电路结构可以分为: 流水线结构和时分复用结构<sup>[2]</sup>。前者可

2014-01-15收到, 2014-05-28改回

国家自然科学基金(61204045, 61271149)和中国科学院、国家外国专家局创新团队国际合作伙伴计划资助课题

\*通信作者: 杨海钢 yanghg@mail.ie.ac.cn

以实现流水执行，因此速度很快，后者速度较慢，但是所需的硬件资源较前者少。在这两种硬件电路架构的基础上，近年来有许多相关文献通过算法优化或者结构优化来降低 1D-FDCT/IDCT 硬件实现的难度。按照支持标准的数量，可以将这些优化方法划分为两类：(1)只针对一种标准进行的设计优化<sup>[2-4]</sup>；(2)针对多种标准的设计优化<sup>[5-8]</sup>。为了提升 DCT 硬件实现的设计效率和能量效率，本文提出一种基于粗粒度可重构结构的 DCT 硬件电路结构，与已有结构的不同之处在于：(1)2D-IDCT/FDCT 的硬件实现的架构不再采用流水线结构或者时分复用结构，而是基于粗粒度可重构阵列结构 (Coarse-Grained Reconfigurable Array, CGRA)<sup>[9,10]</sup>；(2)通过定制 CGRA 的互连网络，省去了矩阵转置所需存储器或者寄存器阵列，而且矩阵转置无需占用时钟周期；(3)整个设计形成一种“阶段级”的流水线结构，能够流水处理 8×8 尺寸大小的像素块。在 SMIC 130 nm 标准单元工艺库下的仿真和综合结果显示，本文提出的结构每个时钟周期能够并行处理 8 个像素，而吞吐率最高可达  $1.157 \times 10^9$  像素/s。与已有结构相比，本文提出的结构设计效率最高提升 4.33 倍，最低提升 77.5%，而能量效率最高提升 12.3 倍，最低提升 14.6%，并且能够以 30 帧/s 的帧率解码尺寸为 4096×2048，格式为 4:2:0 的视频序列。

论文的章节安排如下：第 2 节介绍 2D-IDCT/FDCT 的原理；第 3 节陈述本文提出的硬件电路结构的基本思路；第 4 节给出详细的硬件电路设计；第 5 节是实验结果与数据分析；最后是总结。

## 2 2D-IDCT/FDCT 原理

式(1)和式(2)分别给出了 2D-IDCT/FDCT 的原理，从它们的组成上可以看出二者在计算方式上有很大的相似性，二者的硬件设计一般可以实现共用。在式(1)和式(2)中，矩阵  $C$  是系数矩阵，不同视频标准的系数值不同，但系数在矩阵中的摆放位置和符号相同。以 8×8 2D-IDCT 为例，系数矩阵中系数摆放位置和符号关系如图 1(a)中的  $C_{8 \times 8}$  所示，表 1 列出了不同视频压缩标准的系数比较。

从式(1)或式(2)也可以看出，2D-IDCT/FDCT 的整个计算过程可以拆成 3 部分：基于行/列的 1D-IDCT/ FDCT、矩阵转置和基于列/行的 1D-IDCT/FDCT。而 1D-FDCT/IDCT 可以进一步简化，以 1D-IDCT 为例，如式(3)所示，根据文献[11]提出的算法，系数矩阵  $C_{8 \times 8}$  可以进行矩阵分解为两个子矩阵  $C_{4 \times 4}$  和  $V_{4 \times 4}$  与两个稀疏矩阵  $P_0$  和  $P_1$ ，如图 1(b)~图 1(e)所示。稀疏矩阵  $P_1$  用于调整输入数据的顺序，无需任何计算。稀疏矩阵  $P_0$  用于将两个子矩阵与输入数据的乘积结果通过加减运算合并。

$$Y = CX C^T = C(CX^T)^T \quad (1)$$

$$X = C^T Y C = C^T (C^T Y^T)^T \quad (2)$$

$$C_{8 \times 8} X_{1 \times 8}^T = P_{0,8 \times 8} \begin{bmatrix} C_{4 \times 4} & 0 \\ 0 & V_{4 \times 4} \end{bmatrix} P_{1,8 \times 8} X_{1 \times 8}^T \quad (3)$$

## 3 基本思路

本文提出的电路结构的基本思路是在以下 4 个

表 1 不同视频编码标准采用的系数比较以及常系数乘法的实现方式(系数值：采用移位和加/减法实现系数值)

系数	MPEG-1/2/4	AVS	VC-1	H.264	HEVC
a	362, <<7 + <<7 + <<6 + <<5 + <<3 + <<1	8, <<3	12, <<3 + <<2	8, <<3	64, <<6
b	502, <<7 + <<7 + <<7 + <<6 + <<5 + <<4 + <<2 + <<1	10, <<3 + <<1	16, <<4	12, <<3 + <<2	89, <<6 + <<4 + <<3 + <<0
c	426, <<7 + <<7 + <<7 + <<5 + <<3 + <<1	9, <<3 + <<0	15, <<4 - <<0	10, <<3 + <<1	75, <<6 + <<3 + <<1 + <<0
d	284, <<7 + <<7 + <<4 + <<3 + <<2	6, <<2 + <<1	9, <<3 + <<0	6, <<2 + <<1	50, <<5 + <<4 + <<1
e	100, <<6 + <<5 + <<2	2, <<1	4, <<2	3, <<1 + <<0	18, <<4 + <<1
f	473, <<7 + <<7 + <<7 + <<6 + <<4 + <<3 + <<0	10, <<3 + <<1	16, <<4	8, <<3	83, <<6 + <<4 + <<1 + <<0
g	196, <<7 + <<6 + <<2	4, <<2	6, <<2 + <<1	4, <<2	36, <<5 + <<2
时钟周期	4	1	1	1	2

$$\begin{aligned}
 \mathbf{C}_{8 \times 8} &= \begin{bmatrix} a & b & f & c & a & d & g & e \\ a & c & g & -e & -a & -b & -f & -d \\ a & d & -g & -b & -a & e & f & c \\ a & e & -f & -d & a & c & -g & -b \\ a & -e & -f & d & a & -c & -g & b \\ a & -d & -g & b & -a & -e & f & -c \\ a & -c & g & e & -a & b & -f & d \\ a & -b & f & -c & a & -d & g & -e \end{bmatrix} & \mathbf{P}_{0,8 \times 8} &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \\
 \mathbf{C}_{4 \times 4} &= \begin{bmatrix} a & f & a & g \\ a & g & -a & -f \\ a & -g & -a & f \\ a & -f & a & -g \end{bmatrix} & \mathbf{V}_{4 \times 4} &= \begin{bmatrix} -e & d & -c & b \\ -d & b & -e & -c \\ -c & e & b & d \\ -b & -c & -d & -e \end{bmatrix} & \mathbf{P}_{1,8 \times 8} &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \\
 \text{(a)系数矩阵} & & \text{(b)矩阵 } \mathbf{P}_0 & & \text{(c)简化矩阵 } \mathbf{C}_{4 \times 4} & & \text{(d)简化矩阵 } \mathbf{V}_{4 \times 4} & & \text{(e)矩阵 } \mathbf{P}_1
 \end{aligned}$$

图 1 8×8 IDCT 简化过程中涉及的矩阵

想法的基础上构建:

(1)利用 CGRA 在加速执行计算密集型应用方面的优势,实现基于 CGRA 的 2D-DCT,提升计算效率。2D-DCT 中包含了大量的算术运算,属于计算密集型应用,而粗粒度可重构阵列结构在加速执行计算密集型应用方面的优势非常突出。1 行像素的 1D-DCT 可以通过 1 行的处理单元(Processing Element, PE)来实现,8×8 的 PE 阵列可以同时处理 8 行,如图 2 所示。假设 1 行像素的 1D-DCT 所需时钟周期数为  $N$ ,那么整个 8×8 尺寸大小像素块的 1D-DCT 执行时间只需要  $N$  周期即可完成。

(2)通过定制 CGRA 的互连网络,可以消除矩阵转置所需的存储器或者寄存器阵列。2D-DCT 的矩阵转置一般是通过存储器或者寄存器阵列实现

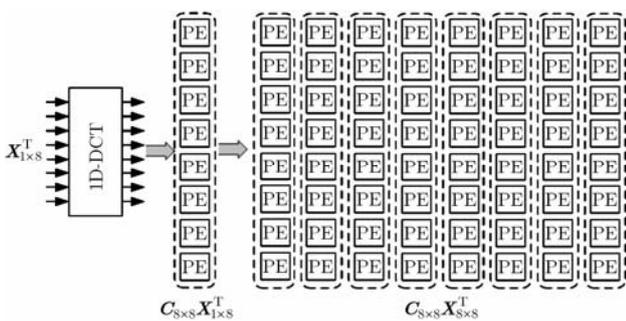


图 2 8×8 尺寸的 1D-DCT 映射到 CGRA 示例

的,而存储器或者寄存器阵列会导致较大的面积开销,并且需要占用多个时钟周期才能完成矩阵转置,同时访问存储器也会产生较高的功耗。如果在 CGRA 阵列中引入行总线和列总线,每行 PE 的计算结果传到行总线,并且第 1 行处理单元的行总线和第 1 列的处理单元的列总线连接,第 2 行处理单元的行总线和第 2 列的处理单元的列总线连接,以此类推,而在映射时采用时间映射,即每行负责 1 行像素的处理,那么处理完的数据传到行总线,而在下 1 个周期读入列总线的数据,即可实现矩阵转置。这样设计的优势在于:矩阵转置无需占用额外的时钟周期;矩阵转置无需借助于存储器或者寄存器阵列,因此可以省去这一部分的面积开销和功耗。

(3)通过定制处理单元的硬件电路结构,节省面积开销。DCT 在计算过程中涉及到大量的乘法运算,如果直接通过乘法器来实现乘法运算,那么整个阵列将需要 64 个乘法器,这将造成巨大的面积开销。为了节省面积开销,可以以移位和加法的方式代替乘法。

(4)引入“阶段级”的流水线结构,进一步提升计算效率,并降低对带宽和 I/O 数目的需求。在 CGRA 上并行执行 8 行数据的 1D-DCT 之前,需要同时将 8 行的数据准备好并传输给每个 PE,因此 CGRA 本身对带宽和 I/O 数目的要求比较高。如图 3 所示,如果将 2D-DCT 的整个计算过程按阶段划

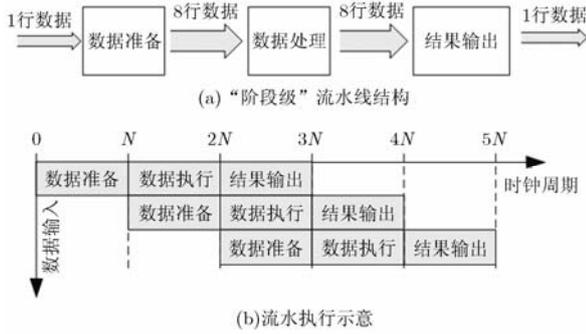


图 3 本文提出的“阶段级”流水线结构

分，每个阶段所需的时钟数相等，并且等前一阶段的数据完全准备好之后才传给后一阶段，那么可以将整个计算过程理解为一个包含 3 级的流水线结构。这样设计不仅可以降低对输入输出数据带宽和 I/O 数目需求，而且利用流水线结构在加速执行方面的特性，可以进一步提升整个结构的计算效率。

### 4 详细设计

基于第 3 节的分析，本文设计了一款基于 CGRA 的 DCT 硬件电路结构。如图 4 所示，本文提出的硬件电路结构包括 4 部分：CGRA、数据准备模块、数据输出模块和时钟分频模块。数据准备模块为 CGRA 准备数据，CGRA 处理输入的数据，处理完成之后传送给数据输出模块。整个结构的时钟信号由时钟分频模块提供。以下几节先详细介绍如何针对 DCT 定制 CGRA，之后再简单介绍其他模块。

#### 4.1 针对 DCT 定制 CGRA 的结构

为了进一步降低硬件实现的复杂度，式(3)可以概括为 4 个步骤：

- 步骤 1 载入数据；
- 步骤 2 调整输入数据的顺序，按照奇偶划分；
- 步骤 3 输入数据与分别与两个子系数块矩阵相乘；

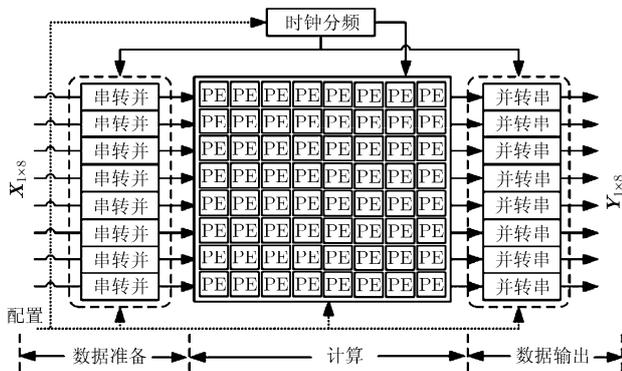


图 4 本文提出的硬件电路结构框图

步骤 4 对乘法结果进行加减处理。

其中，步骤 1 载入的数据既可以是数据总线输入的数据，也可以是矩阵转置之后的数据，步骤 2 和步骤 4 需要有特定的互连线支持，步骤 3 和步骤 4 决定了处理单元的计算功能。以上这些是每行处理单元应该支持的功能。1D-FDCT 可以得到类似的结果，唯一不同的是这些步骤的顺序。以下几节主要针对 IDCT，而 FDCT 可以通过调整这些步骤的顺序得到。

**4.1.1 定制每行处理单元支持的互连线** 根据以上的分析，每行处理单元应该支持的互连线可以划分为 3 类，如图 5 所示。

**I 类：**用于支持稀疏矩阵  $P_1$  调整输入数据的顺序。假设输入 1 行 8 个数据分别为  $x_0 \sim x_7$ ， $x_0$  对应于  $PE_0$ ， $x_1$  对应于  $PE_1$ ，以此类推，那么 1 行处理单元之间的互连线应该包括以下 8 根互连线，即  $PE_0 \rightarrow PE_0$ ， $PE_1 \rightarrow PE_4$ ， $PE_2 \rightarrow PE_1$ ， $PE_3 \rightarrow PE_5$ ， $PE_4 \rightarrow PE_2$ ， $PE_5 \rightarrow PE_6$ ， $PE_6 \rightarrow PE_3$ ， $PE_7 \rightarrow PE_7$ 。通过这 8 根互连线可以完成 1D-IDCT 的第 2 步，也就是按照奇偶顺序调整输入数据。

**II 类：**用于支持稀疏矩阵  $P_0$  进行加减运算。 $P_0$  需要 8 根互连线，具体为： $PE_0 \rightarrow PE_7$ ， $PE_1 \rightarrow PE_6$ ， $PE_2 \rightarrow PE_5$ ， $PE_3 \rightarrow PE_4$ ， $PE_4 \rightarrow PE_3$ ， $PE_5 \rightarrow PE_2$ ， $PE_6 \rightarrow PE_1$  和  $PE_7 \rightarrow PE_0$ ，通过这 8 根互连线即可满足  $P_0$  矩阵在加减运算方面的需求。

**III 类：**除了  $P_0$  和  $P_1$  需要定制互连线之外，系数矩阵的乘法也需要定制互连线。一般来讲，系数矩阵的计算采用  $4 \times 4$  的蝶形运算来实现，但是这样的设计需要过多的互连线支持，互连线增多对应着面积开销增大。在本文提出的设计中，矩阵乘法是

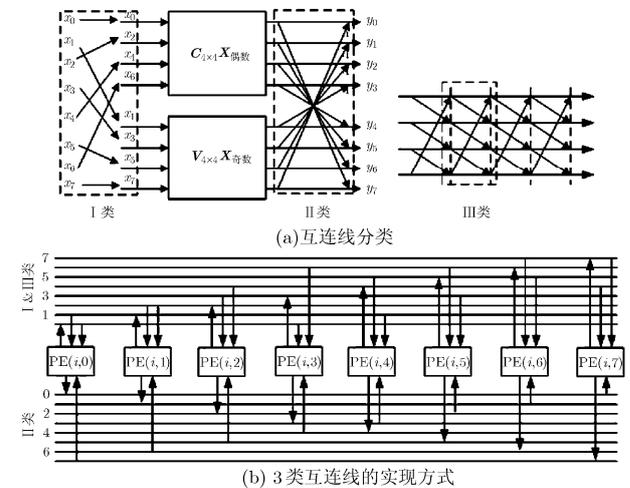


图 5 针对 DCT 为每行处理单元定制的 3 类互连线

通过累加的方式实现的,而数据则是通过循环传递的方式,这种方法需要的互连线较少。对于  $C_{4 \times 4} X_{\text{偶数}}$ ,需要循环传递  $x_0, x_2, x_4, x_6$ ,因此需要互连线  $PE_0 \rightarrow PE_3, PE_1 \rightarrow PE_0, PE_2 \rightarrow PE_1, PE_3 \rightarrow PE_2$ 。同样,对于  $V_{4 \times 4} X_{\text{奇数}}$ ,需要互连线  $PE_4 \rightarrow PE_7, PE_5 \rightarrow PE_4, PE_6 \rightarrow PE_5, PE_7 \rightarrow PE_6$ 。

**4.1.2 消除矩阵转置存储器** 矩阵转置就是将矩阵中第  $i$  行的数据转换为第  $i$  列的数据。如图 6(a)所示,将第  $i$  ( $i \in [0, 7]$ , 从上到下)行处理单元的数据同时输出到第  $i$  根 H 总线上,同时将第  $i$  根 H 总线连接到第  $i$  ( $i \in [0, 7]$ , 从左到右)根 V 总线上,而 V 总线作为 1 列处理单元的 1 个输入,在基于列的 1D-IDCT 的第 1 步时载入 V 总线输入的数据。通过 H 总线和 V 总线即可实现矩阵转置,无需占用额外的时钟周期数,而由此导致的面积开销与存储器或者寄存器阵列相比是很小的。

**4.1.3 定制处理单元的功能** 处理单元的内部结构可以按照功能划分为 3 个部分:

(1)输入数据选择:如图 6(b)所示,输入数据的来源有 6 种:数据输入总线, V 总线, I 类互连线输入的数据, II 类互连线输入的数据、内部寄存器反馈的数据和 III 类互连线输入的数据。数据输入总线和 V 总线输入的数据是第 4.1 节中 1D-IDCT 计算过程中的第 1 步载入的数据, I 类总线传递的数据是 1D-IDCT 的第 2 步调换顺序的数据, II 类互连线则是传递 1D-IDCT 的第 3 步累加所需的数据, III 类互连线输入的数据则是用于完成 1D-IDCT 的第 4 步。

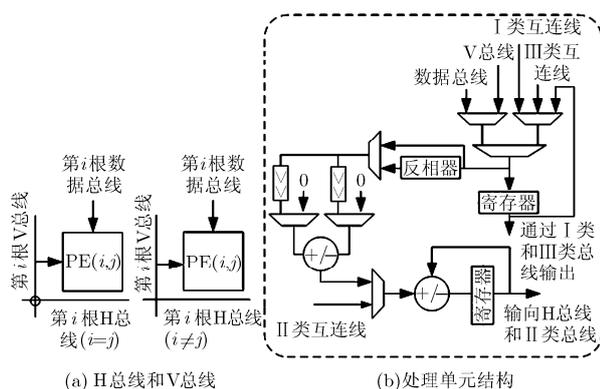


图 6 针对 DCT 定制的互连网络以及处理单元结构

(2)常系数乘法器:系数矩阵  $C_{4 \times 4}$  和  $V_{4 \times 4}$  与  $X$  的乘法需要用到很多系数乘法和加/减法运算,而系数乘法可以用移位和加法来实现以节省面积开销。表 1 为不同视频标准的系数乘法实现方式的比较。从表 1 可以看出,视频标准 VC-1, AVS 和 H.264 只

需要两个移位器和一个加/减法即可实现常系数乘法,而较老的 MPEG-1/2/4 以及仍处于开发状态的 HEVC 所需的移位器和加法器则较多。为了节省常系数乘法导致的面积开销,本文规定每个常系数乘法器中只包含两个移位器(最多支持左移 7 次)和一个加/减法器,而对于需要多个移位器和加法器才能实现常系数乘法的视频标准,则通过多个周期重复调用常系数乘法器来实现。从表 1 中可以看出完成一个常系数乘法最多需要 4 个时钟周期。

(3)累加器和数据输出:累加器用于系数矩阵乘累加的计算,对应于 1D-IDCT 的第 3 步。累加所需的数据通过 II 类互连线循环传递。累加完成之后,需要对  $C_{4 \times 4} X_{\text{偶数}}$  和  $V_{4 \times 4} X_{\text{奇数}}$  的计算结果进行加减运算,即第 4 步,以结束 1D-IDCT 的计算。第 4 步所需的数据需要通过 III 类互连线输入。1D-IDCT 完成之后输出数据。若是基于行的 1D-IDCT,那么输出数据传递到 H 总线上,如果是基于列的 1D-IDCT,输出数据传给输出模块处理。

## 4.2 “阶段级”流水线结构

本文提出的硬件电路结构构建了一种“阶段级”流水线结构。通过这种结构,可以实现流水执行  $8 \times 8$  尺寸大小的 2D-IDCT/FDCT。整个计算过程可以划分为 3 个阶段:数据准备、计算和数据输出阶段,而每个阶段对应 1 个流水级,每个阶段所需的时钟周期均一致,这样可以实现平衡流水。而且,这种设计可以降低对输入/输出带宽和 I/O 数目的需求,同时也可以以  $8 \times 8$  尺寸大小的块为单位加速完成整个视频序列的处理。

**4.2.1 时钟分频** 为了构建流水线结构,本文在电路结构中定义了两个时钟域:(1)低频时钟域,包括数据准备和数据输出模块;(2)高频时钟域,包括 CGRA。不同的视频压缩标准对应的高低频时钟域的时钟周期比不同,如表 2 所示(其中,  $P$  代表时钟周期)。在低频时钟域中,完成一个阶段需要 8 个周期,转换成时间之后等于高频时钟域完成一个阶段的时间,即每个阶段的执行时间是一致的,因此整个结构可以看成“平衡流水线”结构。

表 2 不同标准数据准备阶段周期的比较

	H.264/AVS/VC-1	HEVC	MPEG-1/2/4
完成 2D-DCT 的时钟周期数	12	20	36
高频时钟域的周期	$P$	$P$	$P$
高/低频时钟域周期比	$P/1.5P$	$P/2.5P$	$P/4.5P$

**4.2.2 数据准备和数据输出模块** 数据准备和数据输出模块由低频时钟控制，完成一个阶段需要 8 个时钟周期。数据准备模块由 8 个并行的串转并模块组成，每个周期载入 1 个像素值，8 个周期后拼接成 1 行的数据，然后发出“Ready”信号。数据输出模块由 8 个并行的并转串模块组成，在接收到 CGRA 发出的“Done”信号之后，将 CGRA 输出的数据载入到数据输出模块，然后通过 8 个并转串模块，每个周期输出 8 个数据，经过 8 个周期之后完成输出。

**4.2.3 功能可重构** 本文提出的硬件电路具有动态可重构的特性，具体体现在：(1)通过配置，处理单元阵列的功能重构，实现对不同视频标准的 DCT 的支持。不同标准下，矩阵系数不一致，完成一次常系数乘法的周期也不相同。(2)通过配置，处理单元每个周期的功能重构，实现不同时钟周期下的不同运算。不同标准下，常系数乘法所需的加减运算和移位操作不同，通过配置可选择运算类型和移位数量。(3)通过配置，处理单元之间的互连网络实现重构，实现 DCT 不同步骤下的数据交换和传输。不同步骤下，需要传递的数据不相同，通过配置互连网络实现数据的准确传递。为了阐明硬件电路可重构的特性，本文以在 CGRA 上映射 HEVC 的 DCT 为例描述电路重构的过程。

在 CGRA 上映射 DCT 时，每行处理单元完成 8 个像素的 1D-DCT，对于 HEVC，第 1 行第 1 个处理单元 PE(0,0)在 8 个周期内完成  $ax_0 + gx_6 + ax_4 + fx_2$  的计算，每两个周期完成 1 个常系数乘法，而常系数乘法是通过移位和加法实现，因此每个周期通过配置处理单元内部的移位器即可实现乘法。如图 7 所示，可以看出 PE 的功能每个周期都会重构 1 次，这些功能包括：移位，直传和移位加。而在计算过程中涉及的数据则通过配置 3 类互连线实现准确传递。

在 CGRA 上映射 2D-DCT 时，每行处理单元处理 1 行的像素，8 行的处理单元并行处理 8 行像

素，因此，在同 1 个周期时钟内，每行的配置是一致的，而整个结构可以理解为 1 个单指令多数据 (Single Instruction Multiple Data, SIMD)。因此，CGRA 每个周期所需的配置信息并不多，仅需 1 行处理单元的配置信息即可，8 行处理单元共用一组配置信息。

**4.2.4 整个流程** 本文提出的硬件电路结构的工作流程为：(1)数据准备阶段：利用 8 个低频时钟周期载入 1 个  $8 \times 8$  尺寸大小的像素块，载入完成之后发出“Ready”信号；(2)数据处理阶段：接收到“Ready”信号，将  $8 \times 8$  尺寸大小的像素块分配到每行的处理单元，然后执行基于行的 1D-DCT，之后通过 H 总线和 V 总线进行转置，转置之后再执行基于列的 1D-DCT，完成之后发出“Done”信号；(3)数据输出阶段：接收到“Done”信号，将 CGRA 的处理结果载入，然后每个周期输出 8 个数据，利用 8 个周期完成数据输出。

## 5 实验结果与分析

本文提出的硬件电路结构通过 Verilog HDL 语言描述，并在 SMIC 13 nm 标准单元工艺库下，通过 Synopsys EDA<sup>[12]</sup>工具链对整个设计进行了功能仿真、验证和综合。为了评估本文提出结构的性能优劣，本文挑选了 8 种设计作为参照，这些参照中既有支持单一标准的设计，又有支持多个标准的设计，表 3 给出了本文提出结构与这两类设计的比较结果。由于本文提出的结构中采用的常系数乘法所需的时钟周期是根据视频标准的不同而调整的，因此需要按照视频标准的不同划分为 3 种工作模式，然后分别于参考设计进行比较，这 3 种工作模式与视频标准的对应关系为：模式 1 针对视频标准 H.264/AVC, VC-1 和 AVS；模式 2 针对视频标准 HEVC；模式 3 针对视频标准 MPEG-1/2/4。从表 3 可以看出，与支持单一标准的参考设计相比，本文提出结构的面积开销处于劣势，但是功耗的优势较为明显，这主要是因为本文提出的结构无需矩阵

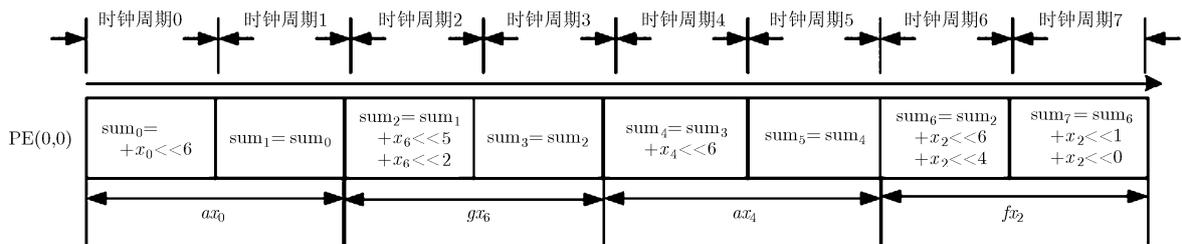


图 7 PE 单元的功能可重构示例

转置存储器,因此也就省去了这一部分造成的动态功耗;与支持多个标准的参考设计相比,本文提出的结构的功耗方面的优势显然要高于面积开销方面的优势。为了较公平地比较本文提出结构与参考设计的性能差异,本文借助于设计效率和功耗效率这两个指标来综合评价这些设计。

### 5.1 设计效率

$$\text{设计效率} = \frac{\text{吞吐量}(\times 10^6 \text{ 像素/s})}{\text{等效门数(门)}} \quad (4)$$

设计效率为吞吐量与等效门数的商,如式(4)所示,其与吞吐量成正比,与面积开销(等效门数)成反比。如图8所示,与只支持单一标准的参考设计相比,本文提出的结构的设计效率较低。这是因为只支持单一标准的参考设计功能单一,灵活性差,应用范围局限于一种标准,而本文提出的结构支持多种标准,灵活性较高,而灵活性的提高必然以面积增加为代价,因此设计效率较低。但是,与支持多种标准的参考设计相比,本文提出的结构的设计效率优势较明显。当处于模式1时,吞吐量达到最大值,此时设计效率最高,与其他结构相比,最多

提升4.33倍,最低提升1.57倍。模式2和模式3由于吞吐率的限制,设计效率的有所下降,但是依然比文献[8]至少提升77.57%。

$$\text{能量效率} = \frac{\text{吞吐量}(\times 10^6 \text{ 像素/s})}{\text{功耗(mW)} \times \text{时钟频率(Hz)}} \quad (5)$$

### 5.2 能量效率

能量效率是指每消耗单位mW的功耗处理的像素数目,计算方式如式(5)所示,其与吞吐量成正比,与功耗成反比。如图9所示,本文提出结构的能量效率优势较为明显,模式1时,与其他结构(包括支持单一标准和支持多个标准的参考设计)相比,本文提出的结构的能量效率最高提升12.3倍,最低提升14.6%,而在模式2和模式3时,由于吞吐率的限制,能量效率略有下降,但是依然比文献[2-4,7-8]至少提升77.7%。

$$\text{最高帧率} = \frac{\text{吞吐量}(\times 10^6 \text{ 像素/s})}{\text{视频宽度} \times \text{视频高度} \times \text{格式系数(像素)}} \quad (6)$$

### 5.3 视频序列的解码能力评估

为了评估本文提出的结构在实时解码视频序列

表3 本文提出的结构与参考设计的比较结果

	文献[4]	文献[5]	文献[11]	文献[6]	文献[7]	文献[8]	文献[9]	文献[10]	本文
工艺(nm)	180	130	180	180	180	180	180	180	130
等效门数( $\times 10^3$ )	17.7	29.3	5.1	47.1	39.8	95.1	15.8	39.4	42.6
时钟频率(MHz)	250.0	330.0	200.0	125.0	50.0	125.0	167.0	200.8	217.0
功耗(mW)	54.0	147.0	15.6	269.0	38.7	58.0@50MHz	32.0	29.9	13.5
并行处理速率(像素/周期)	4	8	4	8	8	8	1	1	8
吞吐量( $\times 10^6$ 像素/s)	1000.0	2640.0	800.0	1000.0	400.0	1000.0	167.0	200.8	1157.0/694.4/385.8
是否需要转置存储	是	是	否	是	是	是	是	是	否
DCT类型	2D-DCT	2D-DCT	2D-DCT	2D-DCT	2D-DCT	2D-IDCT	2D-DCT	2D-IDCT	2D-DCT
支持标准	单一	单一	单一	单一	多个	多个	多个	多个	多个

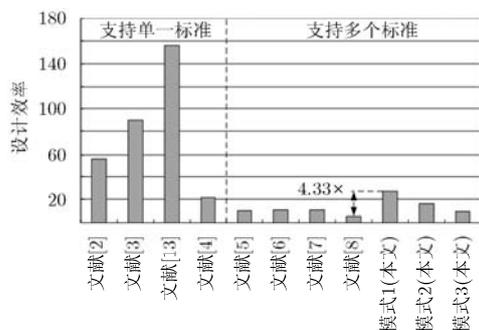


图8 设计效率的比较结果

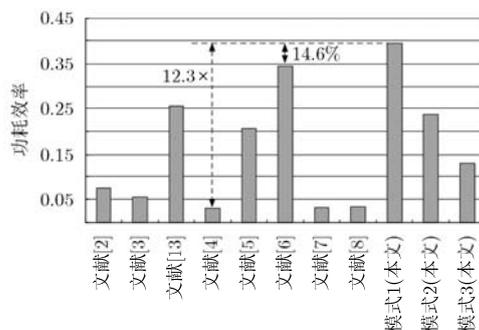


图9 能量效率的比较

方面的能力，给出了硬件电路解码视频序列时可以达到的最高帧率。最高帧率是指 1 秒钟内，实时解码某种格式的视频序列的帧数，如式(6)所示，其与吞吐率成正比，公式中的格式系数根据视频格式为 4:2:0 还是 4:4:4 分别设置为 1.5 和 3.0。表 4 给出了在不同视频尺寸下由式(6)计算得来的评估值，可以看出，本文提出的结构能够在任意模式下以 30 帧/s 的帧率，实时解码最大尺寸为 4096×2048，格式为 4:2:0 的视频序列。

## 6 结束语

本文基于粗粒度可重构阵列结构，设计了一款支持多个视频压缩标准的 8×8 离散余弦变换的硬件

电路结构。利用粗粒度可重构阵列结构的可重配置的特性，根据不同的视频标准，调整工作模式，从而实现支持多个视频压缩标准的 2D-FDCT/IDCT。通过定制粗粒度可重构阵列结构的互连网络，本文提出的结构在矩阵转置时无需占用额外的时钟周期，也无需借助于存储器。而且，整个电路结构形成了一种“阶段级”流水线结构，能够实现流水处理 8×8 尺寸的像素块。实验结果显示，与已有支持多个标准的结构相比，设计效率最高可提升 4.33 倍，最低提升 77.5%，能量效率最高提升 12.3 倍，最低提升 14.6%，而且能够以最高 30 帧/s 的帧率实时处理尺寸为 4k×2k(4096×2048)，格式为 4:2:0 的视频序列。

表 4 本文提出结构的视频序列解码能力评估

视频序列	Full-HD	WQXGA	Quad Full-HD	4k×2k
尺寸	1920×1080	2560×1600	3840×2160	4096×2048
最高帧率(帧/s) (4:2:0) @217 MHz	模式 1	372	188	93
	模式 2	223	113	56
	模式 3	124	62	31

## 参 考 文 献

- [1] Richardson I E G. Video Codec Design: Developing Image and Video Compression Systems[M]. New York: John Wiley & Sons Ltd., 2002: 127-161.
- [2] Chen Y, Chang T, and Lu C. A low-cost and high-throughput architecture for H.264/AVC integer transform by using four computation streams[C]. Proceeding of the 13th International Symposium on Integrated Circuits (ISIC), Singapore, 2011: 380-383.
- [3] Michell J A, Solana J M, and Ruiz G A. A high-throughput ASIC processor for 8×8 transform coding in H.264/AVC[J]. *Signal Processing: Image Communication*, 2011, 26(2): 93-104.
- [4] Sun C C, Donner P, and Götze J. VLSI implementation of a configurable IP Core for quantized discrete cosine and integer transforms[J]. *International Journal of Circuit Theory and Applications*, 2012, 40(11): 1107-1126.
- [5] Huang C, Chen L, and Lai Y. A high-speed 2-D transform architecture with unique kernel for multi-standard video applications[C]. IEEE International Symposium on Circuits and Systems (ISCAS), Seattle, WA, USA, 2008: 21-24.
- [6] Fan C, Fang C, Chang C, et al. Fast multiple inverse transforms with low-cost hardware sharing design for multistandard video decoding[J]. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2011, 58(8): 517-521.
- [7] Chen Y and Chang T. A high performance video transform engine by using space-time scheduling strategy[J]. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2012, 20(4): 655-664.
- [8] Martuza M and Wahid K A. Implementation of a cost-shared transform architecture for multiple video codecs[J]. *Journal of Real-Time Image Processing*, 2012, 1-12, DOI: 10.1007/s11554-012-0266-5.
- [9] Choi K. Coarse-grained reconfigurable array: architecture and application mapping[J]. *IPSI Transactions on System LSI Design Methodology*, 2011, <http://dx.doi.org/10.2197/ipsjtsldm>. 4.31.
- [10] 魏少军, 刘雷波, 尹首一. 可重构计算处理器技术[J]. 中国科学: 信息科学, 2012, 42(12): 1559-1576.
- [11] Wei S J, Liu L B, and Yin S Y. Key techniques of reconfigurable computing processor[J]. *SCIENCE CHINA: Information Sciences*, 2012, 42(12): 1559-1576.
- [12] Chen W H, Smith C, and Fralick S. A fast computational algorithm for the discrete cosine transform[J]. *IEEE Transactions on Communications*, 1977, 25(9): 1004-1009.
- [13] Synopsys Corporation. Accelerate design innovation with design compiler: accelerate design innovation with design compiler[OL]. <http://www.synopsys.com/Tools/Implementation/RTLsynthesis/Pages/default.aspx>. January 2014.
- [14] Qi H, Wei C, and Tong J. A dynamically reconfigurable VLSI architecture for H.264 integer transforms[J]. *Chinese Journal of Electronics*, 2012, 21(3): 510-514.

陈 锐：男，1986 年生，博士生，研究方向为粗粒度可重构阵列结构。

杨海钢：男，1960 年生，研究员，博士生导师，研究方向为可编程芯片与系统、数模混合信号 SoC。

王 飞：男，1981 年生，副研究员，研究方向为数模混合信号可编程序片上系统。