

## 一种新型 2-DCT/IDCT 结构的设计与实现

傅宇卓<sup>1</sup>, 王嘉芳<sup>2</sup>, 胡铭曾<sup>3</sup>

(1. 上海交通大学计算机系, 上海 200030; 2. 黑龙江大学计算机系, 黑龙江哈尔滨 150080;

3. 哈尔滨工业大学 320 信箱, 黑龙江哈尔滨 150001)

**摘 要:** 本文根据 MPEG-2 视频编码的特点, 设计了仅由一个 1-DCT 核完成的 2-DCT/IDCT 结构, 该结构的转换矩阵通过 SRAM 实现, 具备双端口的输入输出, 数据吞吐率较高, 能够有效节省芯片面积. 1-DCT 核由 7 个乘法器组成, 乘法器可以根据计算速度的快慢灵活设计. 为了解决双端口无冲突的存储访问, 提出了一个数据排列方案. 由于乘法器的乘数之一为常数, 我们设计了一种常数修改方案能够有效的降低乘法器的硬件开销. 该 2-DCT/IDCT 结构通过了 FPGA 验证, 具有较强的工程实用价值.

**关键词:** 2-DCT/IDCT; 乘法器; FPGA 验证

**中图分类号:** TP302.2 **文献标识码:** A **文章编号:** 0372-2112 (2002) 12A-2126-04

### The Design and Implementation of a Novel 2-DCT/IDCT Architecture

FU Yu-zhuo<sup>1</sup>, WANG Jia-fang<sup>2</sup>, HU Ming-zeng<sup>3</sup>

(1. Computer Department, Shanghai JiaoTong University, Shanghai 200030, China;

2. Computer Department, Heilongjiang University, Harbin, Heilongjiang 150080, China;

3. P. O. BOX 320, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China)

**Abstract:** The paper gives a 2-DCT/IDCT architecture implemented by 1-DCT core according to the characteristic of MPEG-2 video encoder. The architecture's transform matrix is implemented by SRAM which has dual ports for input/output. It has the advantage of high data throughput rate and low chip area. The 1-DCT core is compared with seven multipliers which could be designed according to the computing speed. The paper gives a data arrangement scheme in order to avoid storage confusion of dual ports. What's more, a novel multiplier architecture is given by modifying constant precision which can reduce the hardware cost efficiently. The architecture is validated by FPGA which proves its engineering value.

**Key words:** 2-DCT/IDCT; multiplier; FPGA validate

## 1 引言

在图像压缩的众多正交变换中, DCT 变换是十分有效的一种, 其变换效果非常接近正交变换中的最佳变换 K-L 变换, 实现方法亦有多种 VLSI 结构设计可供选择. 正因如此, DCT 变换为 JPEG、H263、MPEG-1 及 MPEG-2 等众多标准<sup>[1][2]</sup>所采用. MPEG-2 标准已经成为数字电视、视频点播、机顶盒等众多视频热点产品的工业标准. 研究 MPEG-2 中 2-DCT/IDCT 的实时结构, 无疑是十分必要的.

行列分解结构中最常见的是 DA (Distribute Arithmetic) 结构<sup>[3]</sup>, DA 结构的优点是充分利用了 DCT 的乘法运算中的一个乘数为定常数的特点, 将乘法器改造成位串式的查表运算, 这样一来有效降低了乘法器的开销. 但是其缺点也十分明显: 这种结构芯片面积随内部字长呈指数增长, 而运算仍属于位串式的, 若要提高运算速度, 只能靠增加 ROM 数完成, 代价很高; 另外 DA 结构的部件不能为 DCT、IDCT 所公用, 这也正是现有的 DA 结构多用于解码器上的原因所在.

本文设计的 2-DCT/IDCT 结构面向的目标是 MPEG-2 的视频编码器, 采用硬连乘法器的结构设计, 这种结构可以比 DA 方法少用一个乘法器, 更主要的是通过提高乘法器的吞吐率来提高整个 DCT 的运算速度, 而且本结构可以完成 DCT/IDCT 两项功能. 继续提高 DCT 计算速度是完全必要的: (1) 为节省芯片面积往往只用一个一维 DCT 核, 这个 DCT 核要通过两次运算以完成一个完整的二维运算; (2) 既要完成二维 DCT 运算, 又要完成二维 IDCT 运算<sup>[2]</sup>; (3) 应用 MP@HL 标准的诸如 HDTV 类产品将要求完成更多宏块变换.

## 2 二维 DCT/IDCT 的硬件结构

利用上一节介绍的行列变换法得到的 2-DCT/IDCT 部分的详细设计见图 1. 本设计使用一个一维 DCT 核完成二维 DCT/IDCT 运算, 即每个 2-DCT 运算要通过两次一维 DCT 核完成. 因此其设计功能上有一次 DCT、二次 DCT、一次 IDCT、二次 IDCT 共四种功能.

整个 DCT/IDCT 结构包括 1-DCT 核、行列变换存储器. 其

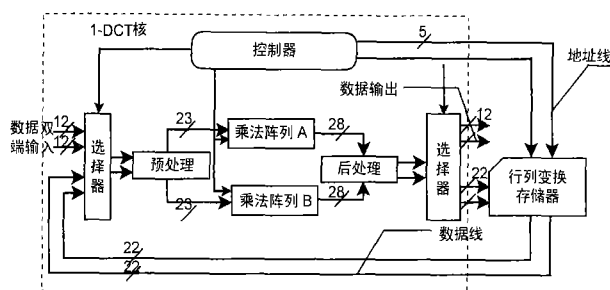


图 1 2-DCT 结构图

工作过程如下:经过滤波的原始图像(9 位数据)<sup>[2]</sup>送入 DCT 核,得到的结果存入行列变换存储器;待一个宏块完成 1-DCT 运算之后,从行列存储器调入 DCT 核进行第二次核运算.输出即为最终结果.对于 *B* 帧的差分图像,其量化结果送出做下面的量化和熵编码.*I*、*P* 帧图像除了送出做熵编码外,还要进行反量化和 IDCT, IDCT 的过程与 DCT 的运算过程一样,只是要把 DCT 核的功能切换成 IDCT. IDCT 的结果送帧缓存供运动估计使用.

## 2.1 1-DCT/IDCT 核的结构设计

DCT 核完成两个  $4 \times 4$  和  $4 \times 1$  矩阵矢量积,该核由两个乘法阵列和两个重排序组成,乘法阵列 *A* 完成一个矩阵矢量积.乘法阵列 *B* 完成另外一个矩阵矢量积.预处理完成输入数据的预排序,输出数据则对同时得到的 8 个数据结果进行缓存输出.

**2.1.1 预处理、后处理功能** 在进行子阵乘之前要先做一次加法,预处理的功能便是完成这四对数的加法,由于预处理的输入来自存储器,而且是每拍两个数据,因此预处理的输入是规则的 0、2、4、6 和 1、3、5、7.显然为了得到所需要的输出序列要在预处理内缓存部分数据.

同理在进行子阵乘之后还需再做一组加法,而且其结果要按规整顺序输出到存储器,后处理便是完成这一工作.

预处理工作于 DCT 方式时(见图 2(a)),端口一、端口二的数据输入并非都是按照从 0 到 7 的顺序,这样做的目的是为了减少预处理的缓存,使得预处理只需要一个加法器、一个减法器就可以了;后处理在做 IDCT 运算时,实际的输出次序并未像图 3(b)所示的那样,为了只使用一个加法器、一个减法器送出结果并且不附带额外的选择控制,端口一送出次序为:  $t_0 + t_1, t_2 + t_3, t_4 + t_5, t_6 + t_7$ ;端口二送出的次序为  $t_0 - t_1, t_2 - t_3, t_4 - t_5, t_6 - t_7$ ,这样就无需控制加法器做加法还

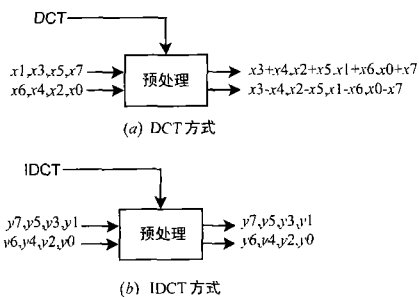


图 2 预处理功能图

是做减法.当然这样一来,数据存入变换矩阵的地址计算将不完全是顺序的了.

**2.1.2 乘法阵列的结构** 乘法阵列完成  $4 \times 4$  与  $4 \times 1$  的矩阵矢量乘,其结构由两个分别计算常数系数矩阵 *P*、*Q* 见表 1、2 的并行乘法列组成(见图 4(a)、图 4(b)).  $p^*$  表示固定系数为 *p* 的乘法器,从左至右的累加器中保存的是矩阵 *P*、*Q* 中第 1 行至第 4 行矢量与像素矢量的乘加结果.

图 3 后处理功能图

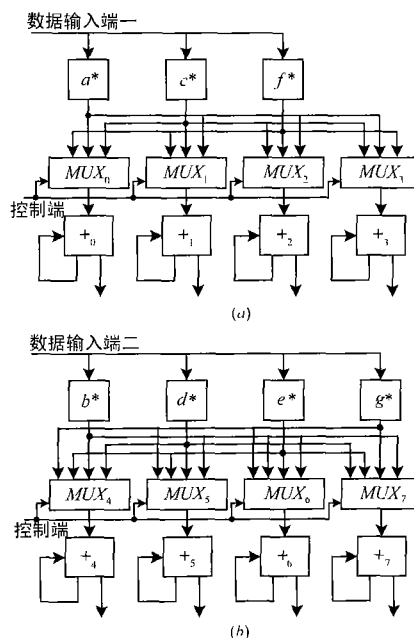


图 4 一维 DCT 核的乘法阵列。(a)乘法阵列 A;(b)乘法阵列 B

两个乘法阵列每隔 4 拍可计算 8 个乘加结果.每个累加器的加数控制由多路选择器控制.其控制表格见表 1、2.

表 1 乘法阵列 A 控制表

	MUX <sub>0</sub>	MUX <sub>1</sub>	MUX <sub>2</sub>	MUX <sub>3</sub>
1	a/a	c/a	a/a	f/a
2	a/c	f/f	-a/-f	-c/-c
3	a/a	-f/-a	-a/-a	c/a
4	a/f	-c/-c	a/c	-f/-f

表 2 乘法阵列 B 控制表

	MUX <sub>0</sub>	MUX <sub>1</sub>	MUX <sub>2</sub>	MUX <sub>3</sub>
1	b	d	e	g
2	d	-g	-b	-e
3	e	-b	g	d
4	g	-e	d	-b

注: a/c 表示做 DCT 运算时选择 *a*, 做 IDCT 运算时选择 *c*

下面以乘法器阵列 *A* 计算 IDCT 时, MUX1 连接的累加器输出来说明其工作原理(假设阵列工作为 4 拍,输入端输入的数据依次为  $x_1, x_2, x_3, x_4$ ):

第一拍选择常数 *a* 乘法器,其运算结果是  $a \times x_1$ ;

第二拍选择常数 *f* 乘法器,最终的运算结果是  $a \times x_1 + f$

$\times x_2$ ;

第三拍选择常数  $a$  乘法器的结果, 控制累加器作减运算, 最终结果为  $a \times x_1 + f \times x_2 - a \times x_3$ ;

第四拍选择常数  $c$  乘法器的结果, 控制累加器作减运算, 最终结果为  $a \times x_1 + f \times x_2 - a \times x_3 - c \times x_4$ ;

第四拍后, 加法器 1 中流出的计算结果为待变换数据与矩阵  $P$  相乘的第二项结果。

同理, 自第 0 个加法器到第 3 个加法器, 计算的是公式矩阵乘的前 4 项结果; 自第 4 个加法器到第 7 个加法器计算的是第二个矩阵乘的结果。

**2.1.3 常数乘法器降低硬件开销的方法** 乘法阵中的基本部件是乘法器, 由于乘法器要求在一个节拍内完成一次乘法, 因此对乘法器的结构设计取决于计算节拍的快慢<sup>[4]</sup>。另外一个问题是乘法器所占用的硬件资源占了 DCT 结构的绝大部分, 如何降低其硬件开销呢?

乘法器的结构设计已经十分成熟<sup>[4~6]</sup>, 如何选择乘法器的结构可以直接交给 EDA 工具完成。由于 DCT 乘法器的乘数之一为常数, 关于常数字长的选择, 将从设计的 RTL 级影响乘法器的开销, 本节拟从乘法器中常数字长的选择来降低硬件开销。乘法阵列中共有 7 个乘法器, 每个乘法器中的常数都是小于 1 的小数, 通过乘以  $2^Q$  进行定标, 此时常数为  $Q$  位。由于进行二进制编码时会出现诸如:  $\cdots 00 \cdots 001$  或  $\cdots 011 \cdots 1111$  这样的情况, 如果将这样的情况改成  $\cdots 00 \cdots 000$  以及  $\cdots 100 \cdots 0000$ , 此时带来的误差为  $2^{-Q}$ , 当  $Q$  较大时, 这个误差是很小的, 但是此时常数项出现了大量 0, 无论采用什么样的乘法结构, 都将极大地降低其硬件开销。

表 3 是本文设计采用的常数值 (常数位数 16 位, 即  $Q = 16$ , 另一个乘数位长 22 位)、带来的误差、节省的硬件开销的一个列表。其中的硬件资源开销是利用 Max-Plus II 中的综合工具, 其单位为逻辑单元 (Logic Cell-LC)。有关乘法器的两个乘数字长的确定和常数修改带来的误差对全局误差的影响, 将另作讨论。

表 3 常数的改造误差及资源节省情况

常数 $\times 2$	原二进制表示	修改后表示	误差 $\times 2^{-16}$	占用的硬件资源		
				改造前 (LC)	改造后 (LC)	节省率 (%)
$a = \cos\pi/4$	0101101010000010	0101101010000000	2	240	189	21.2
$b = \cos\pi/16$	0111101100010101	0111101100010000	2	288	249	13.5
$c = \cos\pi/8$	0111011001000001	0111011001000000	1	254	201	20.9
$d = \cos 3\pi/16$	0110101001101101	0110101001110000	-3	303	235	22.4
$e = \cos 5\pi/16$	0100011100011100	0100011100100000	-4	249	219	12.0
$f = \cos 3\pi/8$	0011000011111011	0011000100000000	-5	270	155	42.6
$g = \cos 7\pi/16$	0001100011111000	0001100100000000	-8	218	146	33.0

从表 3 中可以看到, 应用这种方法, 硬件资源的节省是十分可观的, 7 个乘法器资源的平均节省了 23.4%。

## 2.2 DCT 控制器

控制器主要功能包括: ①复位整个 DCT 核; ②每 4 个周期

复位一次累加器; ③生成乘法器阵列所需要的 0 至 3 的译码信号; ④生成第一次 1-DCT 运算结束和第二次 1-DCT 即一个 2-DCT 运算结束的信号; ⑤生成行列变换存储器的读写时序 (包括读写信号和读写地址); ⑥生成从输出/行列存储器输出的选择信号、生成从片外/行列存储器输入的选择信号。

控制器由时钟控制每拍执行相应的动作, 从启动开始到第 69 拍完成一个完整的控制周期。下面是该控制器的接口描述:

```
entity producecontrol is
    port(reset: in std-logic; //复位信号
          start: in std-logic; //启动信号
          clock: in std-logic; //工作时钟
          DCT: in std-logic; //DCT/IDCT 方式选择
          controls: out std-logic-vector(1 downto 0);
          every4dataforMultiplier: out std-logic;
          firstcomputing, secondcomputing: out std-logic := '0';
          exchange: out std-logic;
          we: out std-logic;
          resetpostprocess: out std-logic;
          address0s, address1s: out std-logic-vector(4 downto 0);
          ReadyforoutputtoRAM: out std-logic);
end entity;
```

控制器中包括了所有 DCT 设计的控制信号: 复位信号、启动信号、工作时钟和 DCT/IDCT 方式选择, 加上两个 12 位数据输入端构成了全部设计输入端。

## 3 行列变换存储器设计

行列变换存储器将  $8 \times 8$  矩阵按行存、按列取, 多数实现由  $8 \times 8$  个寄存器完成, 这样做的好处是没有地址计算, 控制简单, 缺点是消耗大量面积和功耗。为此, 本设计采用 RAM 实现, 由于本结构的存数、取数为两个数据宽度, 我们用两个大小为 32 个字的 RAM 组成行列变换存储器, 其存储方式见表 4。

表 4 行列变换存储数据表

存储器 0	A00	A11	A02	A13	A04	A15	A06	A17
	A20	A31	A22	A33	A24	A35	A26	A37
	A40	A51	A42	A53	A44	A55	A46	A57
	A60	A71	A62	A73	A64	A75	A66	A77
存储器 1	A10	A01	A12	A03	A14	A05	A16	A07
	A30	A21	A32	A23	A34	A25	A36	A27
	A50	A41	A52	A43	A54	A45	A56	A47
	A70	A61	A72	A63	A74	A65	A76	A67

表格的行方向表示存储器的地址增加方向, 列方向地址加 8,  $A_{i,j}$  表示从后处理器中第  $i$  个周期流出的第  $j$  个数据。之所以这样安排数据, 目的就是要从后处理器中流出的两个数据可以一次写入行列变换存储器中同时没有访存冲突, 而又可以按照预处理器要求的顺序从中一次取出两个数据做预处理器输入同时没有访存冲突。

两个存储器的地址写入时生成次序如下 (假设存储器 0

的地址信号为 Address0、存储器 1 的地址信号为 Address1): 第一个周期的 4 拍: Address0 为 0、2、4、6, 而 Address1 为 1、3、5、7; 第 2 个周期的 4 拍 Address0 为 1、3、5、7, Address1 为 0、2、4、6; 第 3 个周期的 4 拍: Address0 为 8、10、12、14, Address1 为 9、11、13、15; ……第 8 个周期的 4 拍, Address0 为 25、27、29、31, Address1 为 24、26、28、30. 在读出时地址做 IDCT 运算时 Address0、Address1 变化相同, 其地址生成次序为: 第一个周期的 4 拍: 0、8、16、24; 第 2 个周期的 4 拍: 1、9、17、25; ……第 8 个周期的 4 拍: 7、15、23、31. 做 DCT 运算时, 为了减少缓存、降低控制的复杂性, 预处理单元要求的地址生成次序为: 第一个周期的 4 拍: Address0 为 0、8、16、24, Address1 为 24、16、8、0; 第 2 个周期的 4 拍: Address0 为 1、9、17、25, Address1 为 25、17、9、1; ……第 8 个周期的 4 拍: Address0 为 7、15、23、31, Address1 为 31、23、15、7.

#### 4 DCT/IDCT 核的工作频率和硬件规模的估算

由于只使用一个 1-DCT 核, DCT/IDCT 运算就无法象两个一维 DCT 核一起工作时那样进行流水<sup>[8,9]</sup>, 对这种结构的吞吐要求要重新进行计算. 按照 HDTV 的 MP&ML720/576/30 指标和 4:2:0 色度格式, 计算一个宏块图像的时间应小于:

$$T_B = \frac{64 \times 10^9}{720 \times 576 \times 30 \times 1.5 \times 2} = 1715 \text{ ns}$$

上式计算中由于 4:2:0 格式每两个亮度块要匹配一个色度块<sup>[2]</sup>, 故有一个 1.5 的系数, 另外由于 DCT/IDCT 核要通过两次以完成 2-DCT 和 2-IDCT 计算, 故有一个系数 2.

由前面计算一个宏块需要 69 拍, 由此估计 DCT/IDCT 核的工作周期  $t = T_B/69 = 24.8 \text{ ns}$ , 工作频率  $f = 1/t = 40.2 \text{ MHz}$ .

该 DCT/IDCT 核在 Altera FLEX10K100 上实现, 芯片利用率 83%, 因此 Altera 等效门数为  $158000 \times 83\% = 131, 140$ .

本文结构与其它结构的比较见表 5, 从表中可知, 本结构的面积时间平方积是较优的. 另外, Ma、Liu、Wang 设计的结构都是脉动式结构, 从表中可以看到, 其面积时间平方积较差.

表 5 表 5 不同的 2-DCT/IDCT 性能比较

	Totzek <sup>[7,8]</sup>	Ma <sup>[7,9]</sup>	Chiu-Liu <sup>[7,10]</sup>	Wang <sup>[7]</sup>	本文结构
计算时间	64	64	64	8	64
乘法器数	16	288	64	256	7
变换矩阵	有	无	无	无	有
常数 ROM	128	0	0	0	7
面积 × 时间 <sup>2</sup>	196608	5308146	262114	524288	163840

#### 5 小结

本文根据 MPEG-2 的特点, 设计了仅由一个 1-DCT 核完成的 2-DCT/IDCT 结构, 该结构具备双端口的输入输出, 数据

吞吐率较高. 2-DCT/IDCT 的转换矩阵通过 SRAM 实现, 能够有效节省芯片面积. 1-DCT 核由 7 个乘法器组成, 乘法器可以根据计算速度的快慢灵活设计. 为了解决双端口无冲突的存储访问, 提出了一个数据排列方案. 由于乘法器的乘数之一为常数, 设计了一种常数修改方案能够有效的降低乘法器的硬件开销.

该 2-DCT/IDCT 结构通过了 FPGA 验证: 工作频率 6.18 MHz、工作器件 Altera EPF10K100GC503-3、存储器占用 2176 位、使用 4362 个逻辑单元、PCB 板接口 ISA 总线.

#### 参考文献:

- [1] 杨品, 钟玉琢. MPEG-1 运动图像压缩编码标准 (ISO/IEC 11172) [M]. 北京: 机械出版社, 1995.
- [2] 钟玉琢, 祁卫. MPEG-2 运动图像压缩编码标准 (ISO/IEC 13812) [M]. 北京: 清华大学出版社, 1997.
- [3] 陈禾. IDCT 的结构设计 [D]. 哈尔滨: 哈尔滨工业大学, 1998.
- [4] David A Patterson. Computer Architecture A Quantitative Approach [M]. 北京: 机械工业出版社, 1999.
- [5] Joseph Y Lee, Hugh L Gravin, Charles W Slayman. A high-speed high-density silicon  $8 \times 8$ -bit parallel multiplier [J]. IEEE Journal of Solid-State Circuits, 1987, 22(1): 35 - 40.
- [6] N Nagamatsu, S Tanaka, J Mori, K Hirano. A 15ns  $32 \times 32$  bit CMOS multiplier with an improved parallel structure [J]. IEEE Journal of Solid-State Circuits, 1990, 25(2): 494 - 497.
- [7] U Totzek. Two-Dimensional DCT with Linear Arrays [M]. Proc. Int. Conf. Systolic Arrays. Kerry, Ireland. Hertfordshire: Prentice-Hall, 1989.
- [8] Chin-Liang Wang. High-throughput VLSI architectures for the 1-D and 2-D DCT [J]. IEEE Transactions on Circuits and Systems for Video Technology, 1995, 5(1): 31 - 40.
- [9] W Ma. 2-D DCT systolic arrays implementation [J]. Electron. Letters, 1991, 27(1): 201 - 202.
- [10] C T Chiu. Real-time parallel and fully pipelined 2-DCT lattice structures with application to HDTV systems [J]. IEEE Transactions on Circuits and Systems for Video Technology, 1992, 2(3): 25 - 37.

#### 作者简介:

傅宇卓 男, 1968 年 10 月生于哈尔滨市, 1990 年本科毕业于国防科技大学计算机系, 1997 年获得哈尔滨工业大学硕士学位, 2001 年获得哈尔滨工业大学计算机系博士学位, 研究方向: 高性能系统结构设计、超大规模集成电路设计、视频图像压缩专用 ASIC 设计. E-Mail: fuyuzhuo@ic.sjtu.edu.cn, Fyz@cs.sjtu.edu.cn.

王嘉芳 女, 1971 年 11 月生于哈尔滨市, 1992 年本科毕业于黑龙江大学计算机系, 2000 年获得哈尔滨工程大学计算机系硕士学位, 目前为哈尔滨工业大学在读博士研究生, 主要研究方向: 多媒体同步技术、超大规模集成电路设计技术、EDA 并行处理技术、高性能计算机系统结构设计.