

快速二维离散余弦变换的 VLSI 设计^{*}

朱竹青, 郭裕顺^{*}, 秦 兴

(杭州电子科技大学 微电子 CAD 研究所, 浙江 杭州 310018)

摘要:针对多标准图像视频编码器中二维离散余弦变换的复用问题, 设计了适用于 MPEG-4, H. 263, JPEG, H. 264 High Profile 编码器芯片的快速 8×8 2-D DCT IP 核, 并完成了 RTL 设计、仿真和 FPGA 原型验证。通过引入 H. 264 High Profile 8×8 整数变换矩阵和增加缩放模块, 完成了多个标准中 2-D DCT 的复用。经 Xvid MPEG-4 编解码器验证, 满足精度要求。设计采用流水线技术, 并优化了速度和频率, 在 90 nm CMOS 工艺下频率达到 840 MHz。研究表明, 该技术能很好地应用于多标准编码器中。

关键词:离散余弦变换; 多标准图像与视频编码; JPEG; MPEG-4; H. 264

中图分类号: TH164; TN402

文献标识码: A

文章编号: 1001-4551(2010)07-0069-05

VLSI design of high-speed 2-D DCT

ZHU Zhu-qing, GUO Yu-shun, QIN Xing

(Institute of Microelectronics CAD, Hangzhou Dianzi University, Hangzhou 310018, China)

Abstract: Aiming at the problem of reusing 2-D DCT in the multi-standard coder, the design of a high-speed 8×8 2-D DCT IP core was presented, which could be used for the coding in MPEG-4, H. 263, JPEG and H. 264. The reuse of the design for multi-standard was realized by the adoption of an 8×8 integer transformation matrix from H. 264 High Profile and a compensation module. The pipeline structure was employed to improve the speed and frequency of the circuit. The design was prototyped using FPGA in the Xvid MPEG-4 codec. The results indicate that the design can be applied to multi-standard coder very well.

Key words: discrete cosine transform (DCT); multi-standard image and video coding; JPEG; MPEG-4; H. 264

0 引 言

自 1974 年 N. Ahmed 等人提出了离散余弦变换 (Discrete Cosine Transform, DCT) 算法后, 很多学者对 DCT 算法的实现进行了研究。陈禾等人^[1]总结了前人的研究, 将 DCT 算法的结构归纳为 5 类, 其中基于乘法器的 DCT 结构一直是很多学者研究的对象, 具有代表性的有 W. H. Chen、LLM^[2]、AAN^[3]等算法。LLM 算法将 1-D DCT 所需的乘法器个数减小到理论最小值 11 个。

在编解码器应用方面, Xvid 的 MPEG-4 编解码器使用 LLM 算法的 12 个乘法器形式, 而很多 JPEG 编码器用 AAN 算法。李晗等人^[4]利用加法和移位器代替

乘法器, 在提高频率上取得了比较好的效果, 但是即使将其运用到 LLM 算法中, 2-D DCT 仍需要使用 184 个加法器。

本研究通过引入整数变换矩阵和增加缩放模块, 设计面积小、速度快、频率高的且可复用多个标准的 2-D DCT。

1 离散余弦变换 DCT 原理

离散余弦变换 DCT, 是指对一帧图像以块为单位 (通常是 8×8 或 4×4 像素块), 通过 2 维 DCT 变换将图像从空间域变换到频域的过程, 目的在于去除像素间的相关性。

二维 $N \times N$ 图像块的 DCT 变换的定义^[5]如下:

收稿日期: 2009-11-18

基金项目: 浙江省科技计划重大资助项目 (2008C01049)

作者简介: 朱竹青 (1986-), 男, 江西赣州人, 主要从事视频变换编解码的 ASIC 设计方面的研究. E-mail: zhuzhuqing@live.cn

通信联系人: 郭裕顺, 男, 教授, 硕士生导师. E-mail: ysguo@hdu.edu.cn

$$F(u, v) = c(u) \times c(v) \times$$

$$\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \left(f(x, y) \cos \left[\frac{\pi(2x+1)u}{2N} \right] \cos \left[\frac{\pi(2y+1)v}{2N} \right] \right) \quad (1)$$

$$u, v = 0, 1, \dots, N-1$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 12/8 & 10/8 & 6/8 & 3/8 & -3/8 & -6/8 & -10/8 & -12/8 \\ 1 & 1/2 & -1/2 & -1 & -1 & -1/2 & 1/2 & 1 \\ 10/8 & -3/8 & -12/8 & -6/8 & 6/8 & 12/8 & 3/8 & -10/8 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 6/8 & -12/8 & 3/8 & 10/8 & -10/8 & -3/8 & 12/8 & -6/8 \\ 1/2 & -1 & 1 & -1/2 & -1/2 & 1 & -1 & 1/2 \\ 3/8 & -6/8 & 10/8 & -12/8 & 12/8 & -10/8 & 6/8 & -3/8 \end{bmatrix} \quad (2)$$

2 二维 DCT 的 VLSI 结构优化设计

2.1 算法优化

整数矩阵的引入,一方面有效地减少计算量,另一方面由于 $\otimes E_f$ 的运算被放到量化端和量化同时考虑,其矩阵系数由量化步长 QP 的大小决定。所以 H. 264 标准对于 DCT 的处理和别的标准不同,很难用于其他标准中,即不具有普遍性。笔者考虑将 C_f 矩阵引入,得到 E_f 矩阵:

$$E_f = \begin{bmatrix} a & b & c & b & a & b & c & b \\ b & d & e & d & b & d & e & d \\ c & e & f & e & c & e & f & e \\ b & d & e & d & b & d & e & d \\ a & b & c & b & a & b & c & b \\ b & d & e & d & b & d & e & d \\ c & e & f & e & c & e & f & e \\ b & d & e & d & b & d & e & d \end{bmatrix} \quad (3)$$

其中,a,b,c,d,e,f都是浮点数,由于在 VLSI 中运算不方便,将其转化为定点数,经过扩展放大 2^{16} 得到 $a = 8\ 192, b = 7\ 740, c = 10\ 486, d = 7\ 346, e = 9\ 777, f = 13\ 159$ 。

如果直接乘以缩放矩阵式(3)的 64 个元素需要 64 个乘法,耗费时钟多,面积大。本研究考虑引入流水线技术,具体有以下两种方案:

(1) 通过在乘法器前面增加选择器可以压缩到 8 个乘法器。例如,如果 2-D DCT 按列输出,则每一列相应的元素都乘以上面矩阵中对应列的相应元素,由于对称性,可以在让每个乘法器在 3 个数之间选择,送出乘数,进行相乘,如第 1 个乘法器在 a,b,c 之间切换,如果当前输入列为第 1 列或者第 4 列,则输出到第 1 个乘法器的一个乘数为 a,另一个乘数为相应列的第

写成矩阵形式为 $F = A_f A^T$,其中 A 矩阵为浮点数矩阵,在 H. 264 High Profile 标准中,将 A 阵进行整数化,矩阵形式等效为: $Y = (C_f X C_f^T) \otimes E_f$,其中 C_f 矩阵^[6]为:

1 个数据,如果是第 2、4、6、8 列,则选择相应的乘数为 b,如果是第 3、7 列则选择乘数为 c。同理,其他 6 个乘法器类似。

(2) 如果对频率要求比较高,考虑到乘法器的延迟比较大,将乘法器用加法器和移位寄存器替代。根据 3 种乘数并行计算出乘法结果。例如,由于第 2 维 DCT 送出的某一列的第 1 个像素结果,有可能进行的乘法是“ $\times a$ ”、“ $\times b$ ”、“ $\times c$ ”。为了取代这个乘法器,先用加法器和移位器取代这些乘数,如: $b = 2 \ll 13 - 2 \ll 9 + 2 \ll 6 - 2 \ll 2$ 。这样可以通过加法和移位来计算当前数据经过 3 种乘法之后的结果。通过计数当前列数即可选出最后的结果。由于每个系数最多有 3 个加法器即可实现,因此,方案 1 需要的一级乘法器延迟可以变成 2 级加法的延迟,提高了工作频率。取代 8 个乘法器所需加法器数量如表 1 所示。

由表 1 可知,每组改成 2 级加法提高工作频率后,用 66 个加法器取代 8 个乘法器。针对不同的应用需求(频率、面积)可以考虑不同的方案。

表 1 方案 2 所需加法器数量

系数	需要次数	共需加法器数量
a	2	0
b	4	12
c	6	18
d	4	12
e	6	18
f	2	6
总计		66

2.2 算法验证

对以上算法进行 C 语言建模,并将其移植到 Xvid 的 MPEG-4 的开源代码^[7]编码器的 DCT 中。在不改变其他模块(包括 IDCT)的前提下,通过输入不同的 YUV 原始视频序列(格式为 4:2:0),先用标准代码进

行编码,再用改进后的代码进行编码,分别通过 Xvid 的 MPEG-4 解码器进行解码。最后分别将解码得到的 YUV 文件和原始 YUV 文件,在亮度 Y、色度 Cb 和 Cr 分量上进行 PSNR 值对比。统计每个序列 300 帧中每种成分的均值和所有序列在 Y、Cb、Cr 上的均值。本研究所用 YUV 原始视频序列均为标准测试序列,图像大小为 CIF,比特率设为 512 Kbps 时统计所得结果如表 2、表 3 所示。

表 2 方案 1 与 Xvid 算法的编码视频 PSNR 值 (dB) 对比

	Y0	Y1	Cb0	Cb1	Cr0	Cr1
1	36.44	36.41	39.43	39.51	41.31	41.33
2	30.49	30.5	40.97	40.96	43.02	42.97
3	32.61	32.51	39.58	39.58	39.28	39.29
4	35.7	35.66	41.34	41.32	42.46	42.5
5	30.8	30.8	34.85	34.85	34.66	34.68
6	29.82	29.84	33.93	33.94	36.23	36.24
7	29.67	29.68	32.7	32.72	34.7	34.69
8	31.15	31.15	37.84	37.74	38.86	38.8
9	33.63	33.64	36.79	36.69	38.39	38.45

表 3 方案 2 与 Xvid 算法的编码视频 PSNR 值 (dB) 对比

	Y0	Y2	Cb0	Cb2	Cr0	Cr2
1	36.44	36.41	39.43	39.52	41.31	41.34
2	30.49	30.5	40.97	40.98	43.02	42.96
3	32.61	32.51	39.58	39.56	39.28	39.29
4	35.7	35.67	41.34	41.33	42.46	42.49
5	30.8	30.8	34.85	34.85	34.66	34.68
6	29.82	29.83	33.93	33.94	36.23	36.27
7	29.67	29.68	32.7	32.73	34.7	34.71
8	31.15	31.15	37.84	37.74	38.86	38.84
9	33.63	33.63	36.79	36.82	38.39	38.47

YUV 序列 1 ~ 9 视频序列分别是 akiyo, coast-guard, container, mother_daughter, stefan, tempete, flower, vectra, footbal, 帧数为 300。表中 Y0, Cb0, Cr0 分别代表 Xvid 标准算法编码所得视频序列的亮度 Y 和色度 Cb, Cr 分量的 PSNR 值,而 Y1, Cb1, Cr1, Y2, Cb2, Cr2 分别为方案 1 和方案 2 在各分量的 PSNR 值。

两种方案相对于 Xvid 标准算法信噪比变化如表 4 所示,数值如果为正,表明 PSNR 值有所改善,如果为负,表明 PSNR 值有所损失。由表 4 可以看到本研究的两种方案图像质量损失不超过 0.015 dB,而且方案 2 显示出本研究算法在色度分量上分别改善了 0.004 dB 和 0.015 dB。经过大量统计,结果表明本研究提出的算法保证了图像质量,精度满足要求。

表 4 两种方案相对于 Xvid 标准算法信噪比变化

信噪比变化均值/dB	亮度 Y	色度 Cb	色度 Cr
方案 1	-0.013 333	-0.013 333	0.004 44
方案 2	-0.014 444	0.004 444	0.015 55

3 二维 DCT 的硬件实现

3.1 整体结构

本研究采用行列分解法进行二维 DCT 的设计,其硬件实现框图如图 1 所示。

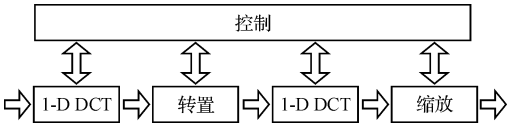


图 1 2-D DCT 硬件实现框图

整个 2-D DCT 采用流水线设计,每个时钟输入为 8 个数据,输出也是 8 个数据,中间除了流水深度需要的时钟外,不需要任何多余的等待,速度快。同时可以根据不同的应用需求以及对频率和速度要求更改流水节拍,实现速度和频率的最优化。

3.2 一维 DCT 的实现

这里一维 DCT 要实现的是式(2)中的 C_f 矩阵,其硬件实现如图 2 所示。硬件实现的时候可以设计成 4 级流水或 2 级流水。

8×8 块的一行数据经过 4 级加法器之后输出。如果对频率要求比较高,考虑用 4 级流水,图 2 的 1、2、3、4 分别为 1D-DCT 的对应 4 级加法,在第 3 级的加法中上半部分的数据需要进行寄存以保证数据的同步。如果需要时钟周期少,则用 2 级流水,图中 5 和 6 分别对应 2 级加法。图中的“ $\gg$ ”为右移操作。

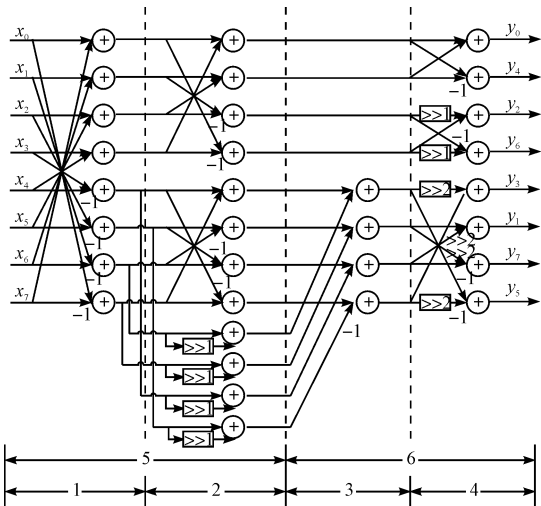


图 2 一维 DCT 硬件实现

为了保证数据不溢出,对于第 1 级一维 DCT 输入的数据精度为 9 bit,输出数据精度为 13 bit。对于第 2 级一维 DCT 输入的数据精度为 13 bit,输出为 17 bit。

3.3 转置的实现

为了保证速度,流水作业不能被打断,如果用单口 RAM 乒乓操作则实际所需要的 RAM 数为 16 个深度为 1、宽度为 104 bit 的 RAM,所以无论从 RAM 的产生还是面积的角度来考虑都不现实。A. Aggoun 等人^[8]使用 2 组三角矩阵堆实现了矩阵转置的功能,通过控制 8 个 8 选 1 选择器来连接,这种结构需要 72 个寄存器,流水的深度为 9 clk。李哈等人实现 4×4 块只用了 16 个寄存器,节省了面积和时钟。

本研究使用 64 个寄存器的寄存器组进行转置的硬件实现方案,如图 3 所示。图中每个方框(如 r00)都是由一个选择器和一个寄存器组成。举例说明,如果输入的数据是按照 8×8 的列顺序输入,依次是 $x_0, x_1, \dots, x_7$ 。在第 1 个时钟将 $x_0 \sim x_7$ 依次存入寄存器 r70, r60, $\dots$, r00 中,第 2 个时钟,第 2 列的寄存器将第一列 r70 ~ r00 的数据依次存入 r71 ~ r01 中,而 r70 ~ r00 的数据还是从输入端取数据存入。如此执行下去,直到第 8 个时钟,整个 8×8 块的数据已经存入寄存器组中,此时寄存器 r77, r76 $\dots$ r70 的值已经有效,依次从中取出数据,经过选择器按顺序排好输出,此时输出的是上个矩阵的第 1 列。在第 9 个时钟, r77 ~ r70 从 r67 ~ r60 中取数据并且更新寄存,而 r67 ~ r60 从 r57 ~ r50 中取数据并且寄存,其他行的数据类似操作,唯独 r07 ~ r00 分别从输入端取数据 $x_0 \sim x_7$ 并且寄存。第 16 个时钟时第 2 个 8×8 块的数据已经寄存完毕,可以依次从 r77 ~ r07 的寄存器中取数据。如此流水下去,便完成了行列转换,流水深度为 8 clk,中间不需要等待,每个时钟每个寄存器都处于工作状态,提高了寄存器的利用率,并且节省了面积,提高了速度。

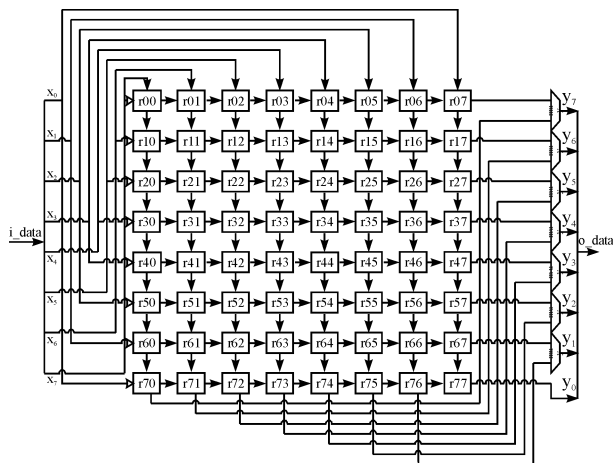


图 3 转置的硬件实现

3.4 缩放模块的实现

(1) 方案 1:使用乘法器。

采用方案 1 的缩放模块的硬件实现如图 4 所示,使用 8 个乘法器来实现。通过 8 个 3 选 1 选择器分别选择不同的乘法器系数, control 通过计数确定当前输出的数据是 8×8 块中的第几列。如果是第 1、5 列,则 8 个选择器依次选择 8 192, 7 740, 10 486, 7 740, 8 192, 7 740, 10 486, 7 740。如果是第 2、4、6、8 列,即依次选 7 740, 7 346, 9 777, 7 346, 7 740, 73 46, 9 777, 7 346。如果是第 2、4、6、8 列,则依次选择 10 486, 9 777, 13 159, 9 777, 10 486, 9 777, 13 159, 9 777。本方案的延时主要是 1 级乘法器。

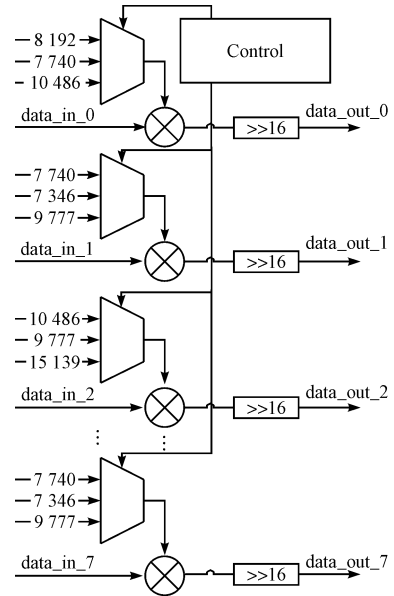


图 4 采用方案 1 的缩放模块的硬件实现

(2) 方案 2:使用加法器和移位器代替乘法器。

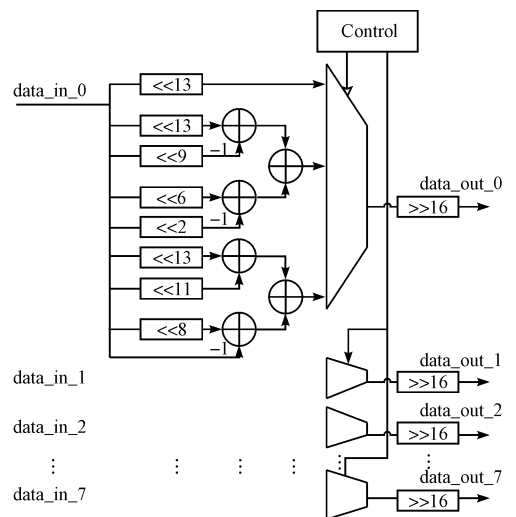


图 5 采用方案 2 的缩放模块的硬件实现

采用两级的加法器和移位器代替乘法器,提高电路的工作频率,如图 5 所示。第 1 个数据经过并行的 3 路输入,每一路都由移位器和加减法器组成,通过

control 模块对输入的数据所在 8×8 块的列数进行选择,以面积换速度,通过 3 路并行工作,最后控制选择输出。本方案的延时主要在 2 级加法器。

两种方案都需进行缩放,由于这些系数都经过放大 2^{16} 倍来保证中间计算值的精度,最后 DCT 结果需要经过右移 16 bit,需要注意数据的截断。

3.5 性能对比

本研究用 Verilog HDL 语言描述电路,分别用 VCS 和 DC 进行了仿真和综合,下面将本研究与其他文献所设计的 2-D DCT 进行性能对比。2-D DCT 算法使用乘法器和加法器个数对比如表 5 所示。

表 5 2-D DCT 算法使用乘法器和加法器个数对比			
算法	乘法器/个	加法器/个	备注
LLM	24	64	
AAN	10	58	另外需要乘以一个缩放矩阵
文献[4]算法 应用于 LLM 中	0	184	
本研究(方案 1)	8	64	
本研究(方案 2)	0	130	

由表 5 可见,本研究提出的 2 种方案计算复杂度比其他文献都小。其中,将文献[4]应用于 LLM 的 24 个乘法器和 64 个加法器 2D-DCT 方案需要 184 个加法器。若应用于理论上最低乘法器(11 个)的结构时,也需要 168 个加法器。AAN 算法中 1D-DCT 蝶形中用了 5 个乘法器,但每次一维 DCT 后要乘以 8 个缩放系数需要额外的 6 个乘法器,或二维 DCT 后乘以一个缩放矩阵,对于除 JPEG 外的其他标准不适用。

ASIC 设计实现 2-D DCT 性能对比如表 6 所示^[9-10],本研究所设计的 2-D DCT 完成单个 8×8 像素块的周期被降到 20 个时钟频率能达到 510 MHz。在 90 nm CMOS 工艺下,DC 综合最高可以达到 840 MHz。若数据保持流水送入,则只需要经过 3.09 μs 可完成一帧高清 1 080 pixels 图像的数据处理,满足工程应用需求。

表 6 ASIC 设计实现 2-D DCT 性能对比				
作者	工艺	频率/MHz	周期/Clk	面积/ μm^2
KURODA T	0.3 μm	150	112	4 000 000
Rohini Krishnan 等	0.18 μm	150	38	834 000
李晗,等	0.18 μm	250	26	—
方案 1	90 nm	400	20	101 923
方案 2	90 nm	510	20	72 665
		840	25	106 016

FPGA 原型验证采用的器件为 Xilinx Virtex-5 XC5VLX220-1 FF1760,综合工具采用 Xilinx ISE。对于 2-D DCT,计算周期为 20 clk 时,方案 1 综合频率为 180 MHz,方案 2 为 200 MHz,周期为 25 clk 时方案 2 为 290 MHz。

4 结束语

本研究旨在通过设计适用于 MPEG-4, H. 263, JPEG, H. 264 等编码器芯片的快速 8×8 DCT 的 IP 核,研究并设计了适合于 VLSI 实现的快速 2-D DCT 结构,利用行列分解法,通过引入整数变换矩阵、增加缩放模块实现,经算法验证满足精度要求。转置的实现方案提高了寄存器的利用率,并且减小了流水深度。提供了 2 种实现方案实现缩放模块,可以根据不同的应用需求在面积、速度、频率之间权衡。采用流水线设计,并减小了流水深度,提高了速度和频率。具有灵活性和实用性。

用 Verilog HDL 进行硬件设计,并用 VCS 进行了仿真,在 90 nm CMOS 工艺下电路综合频率达到 840 MHz,FPGA 模拟验证频率达到 290 MHz,完全满足设计要求。与其他文献 2-D DCT 算法结构相比,本研究提出的结构具有面积小、速度快、频率高的特点。不仅可以很好地应用于 MPEG-4 和 H. 263 和 JPEG 等标准,还将 H. 264 High Profile 中的 8×8 整数 DCT 进行了复用,是一个基于多标准的 2-D DCT 的 IP 核,可以很好地应用于支持多标准的编码器中。

参考文献(References):

[1] 陈 禾,毛志刚,叶以正. DCT 快速算法及其 VLSI 实现[J]. 信号处理,1998,14(A12):62-70.

[2] LOEFFLER C, LIGHTENBERG A, MOSCHYTZ G S. Practical Fast 1-D DCT Algorithms with 11 Multiplications[C]//Int Conf Acoustics, Speech and Signal Processing. Glasgow: [s. n.], 1989:988-991.

[3] ARAI Y, AGUI T, NAKAJIMA M. A fast DCT-SQ scheme for images[J]. Transactions of Institute of Electronics, Information and Communication Engineers, 1988, E71(11):1095-1097.

[4] 李 晗,孙义和,向采兰. 二维离散余弦变换及其逆变换的 VLSI 实现[J]. 微电子学,2008,38(3):326-329.

[5] 毕厚杰. 新一代视频压缩编码标准—H. 264/AVC[M]. 北京:人民邮电出版社,2005:112-114.

[6] GORDON S, MARPE D, WIEGAND T. Simplified use of 8×8 transforms[S]. JVT Doc, JVT-K028, Munich,2004.

[7] XVID MPEG-4 VIDEO CODEC[DB/OL]. [日期不详]. www.xvid.org.

[8] AGGOUN A, JALLOH I. Two-dimensional DCT/IDCT architecture[J]. IEEE Proc. Computer Digital Technology,2003,150(1):2-10.

[9] KURODA T. A 0.9 V,150 MHz,10 mV,4 mm²,2-D discrete cosine transform core processor with variable threshold-voltage(VT) scheme[J]. IEEE J. SSC,1996,31(11):1770-1779.

[10] KRISHNAN R, GANGWAL O P, EIJNDHOVEN J, et al. Design of a 2D DCT/IDCT Application Specific VLIW Processor Supporting Scaled and Sub-sampled Blocks[C]. 16th International Conference on VLSI,2003.