

一种低功耗色彩空间转换模块的 FPGA 实现

沈 斌, 张多利, 何亚军

SHEN Bin, ZHANG Duo-li, HE Ya-jun

合肥工业大学 微电子设计研究所, 合肥 230009

Institute of VLSI Design, Hefei University of Technology, Hefei 230009, China

E-mail: shenbin1598@163.com

SHEN Bin, ZHANG Duo-li, HE Ya-jun. Low-power FPGA implementation of color space conversion. *Computer Engineering and Applications*, 2008, 44(22): 84-86.

Abstract: In this paper a low power design for a FPGA implementation of a color space converter (from RGB to YCbCr) is presented. First, the Color Space Converter (CSC) using FMC algorithms is developed, then it is hand optimized in YCbCr space. The proposed architecture is a low power and area optimized implementation. It can save more than 30% dynamic power consumption in low sample rate.

Key words: color space conversion; Color Space Conversion (CSC); dynamic power consumption; low power

摘 要: 介绍了一种视频压缩系统中使用的色彩空间转换(从 RGB 到 YCbCr)模块的 FPGA 实现。首先, 使用 FMC 算法实现了色彩空间转换模块, 并基于色彩空间转换模块在视频压缩系统中的应用特点提出了一种改进方式, 在该方法中使用了少量的单元实现了面积优化和功耗优化。在低采样率的模式下可以减少 30% 的动态功耗。

关键词: 色彩空间转换; CSC; 动态功耗; 低功耗

DOI: 10.3778/j.issn.1002-8331.2008.22.025 **文章编号:** 1002-8331(2008)22-0084-03 **文献标识码:** A **中图分类号:** TN402

1 介绍

低功耗已经成为集成电路设计中最重要因素之一, 尤其是在直接由电池供电的移动设备中。在数码相机、数码摄像机等视频系统中都要用到色彩空间转换电路(Color Space Conversion, CSC)。虽然 CSC 是整个视频压缩系统中很小的一部分, 但是在工作时 CSC 要对 RGB 空间每一个像素进行处理, 在经过优化的解码器中由 YCbCr 计算 RGB 所消耗的能量可以占到解码系统整体功耗的 40%^[1], 所以优化 CSC 对视频压缩系统的整体还是很有意义的。

由于人眼视觉系统的特点及视频压缩的实际应用需要, 在 YCbCr 空间中的采样最常用的是 4:2:0 和 4:2:2 的采样率^[2], 如图 1 所示。本文从减少电路硬件资源和降低电路的翻转率出发, 对 CSC 中的乘加运算进行了改进, 使其使用尽可能少的硬件资源并在 YCbCr 空间实现了资源共享, 从而可以在低采样率时, 减少电路不必要的翻转, 减少 30% 以上的动态功耗。

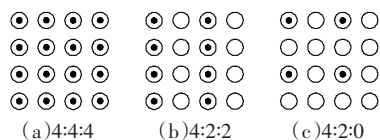


图 1 YCbCr 采样的不同模式

2 色彩空间

自然界中的任何一种颜色都可以由 RGB 三种基色构成, 它们构成一个三维的矢量空间。RGB 由于其设备的独立性, 成为计算机图像处理中使用最普遍的颜色空间。

YUV 模型的优点是它的亮度信号(Y)和色度信号(U、V)相互独立, 也就是说由它们构成的单色图可以单独编码、处理。由于这个特点, 在计算机视频应用中经常使用 YUV 颜色空间。

因为现有的视频采集和显示设备都采用 RGB 信号驱动, 而在数字图像的生成、存储、处理时需要使用 YUV 空间, 所以需要在 RGB 与 YUV 颜色空间之间作不同的处理和转换。

RGB 向 YCbCr 颜色空间转换的公式如下:

$$D = \begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = C \times \begin{pmatrix} R \\ G \\ B \end{pmatrix} = T \times S \quad (1)$$

其中 T 为 3×3 的常数系数矩阵, 在 JFIF 中定义为

$$\begin{pmatrix} 0.299 0 & 0.587 0 & 0.144 0 \\ -0.168 7 & -0.331 3 & 0.500 0 \\ 0.500 0 & -0.418 7 & 0.081 3 \end{pmatrix}, \text{用等式表示如下}$$

$$\begin{aligned} Y &= 0.299R + 0.587G + 0.114B \\ Cb &= -0.168 7R - 0.331 3G + 0.500B \\ Cr &= 0.500R - 0.418 7G - 0.081 3B \end{aligned} \quad (2)$$

基金项目: 国家自然科学基金(the National Natural Science Foundation of China under Grant No.60576034); 安徽省自然科学基金(the Natural Science Foundation of Anhui Province of China under Grant No.070412031)。

作者简介: 沈斌(1982-), 男, 硕士研究生, 主要研究领域为大规模集成电路设计。

收稿日期: 2007-10-15 **修回日期:** 2007-12-29

3 精度控制

由颜色空间转换公式可见,RGB 和 YCbCr 之间是一种非整数比例的和或差的关系,系数的精度过小会导致结果出现误差,而精度过大又会占用过多的硬件资源消耗更多的能量,而且所提高的精度并不会被采样到,是没有意义的。所以选取合适的位数使用适当的精度对于减小功耗还是相当重要的。

(1)在计算误差允许范围内,采用合适的比例因子,将小数放大成整数,为便于处理,设置放大比例为 2^m ,定义为放大比例因子。这里选择 $M=8$,即放大 256 倍,对转换公式中的乘数进行近似处理。

M 取 8 时转换公式如下:

$$Y = \frac{77R + 150G + 29B}{256}$$

$$Cb = \frac{-43R - 85G + 128B}{256}$$

$$Cr = \frac{128R - 107G - 21B}{256}$$

(2)使用近似的乘数完成乘运算,并对结果进行四舍五入,为防止计算过程中,中间结果超过最大绝对值,设计中还要对其进行限制,以作为出现上溢的处理。

(3)将计算结果除以 2^m ,还原 YCbCr 的真实值。

近似算法的误差与性能分析:表征颜色空间转换算法误差的是整数乘法计算结果与小数乘法计算结果的绝对误差及其积累。由于模块输出结果是整数,且取值区间为 $[-128, 127]$,因此定义误差域值为 0.500。当计算结果的绝对误差小于该误差域值时,标识近似算法的误差在允许范围之内;否则将引起 YCbCr 计算结果的差异。

由颜色空间转换算法可见,采用整数乘法近似的计算误差与放大比例因子的选取密切相关。 M 值过小,近似结果的准确度得不到保障,甚至会出现颜色失真, M 值过大,又会出现超出人眼区分范围,同时降低频率等性能指标。表 1 给出了 M 分别取 7、8、9 时,YCbCr 的最大理论绝对误差。理论绝对误差的计算公式如下:

$$ERROR_{D_i} = \sum_{j=1}^3 abs[C_{ij} - \frac{int(2^M \cdot C_{ij})}{2^M}] S_j \quad (3)$$

其中, $abs[]$ 表示的是取绝对值函数; $int()$ 表示的是取整函数。

表 1 不同放大比例因子下 YCbCr 的最大误差

	$M=7$	$M=8$	$M=9$
Y 的最大绝对误差	0.816	0.454	0.228
Cb 的最大绝对误差	0.813	0.187	0.187
Cr 的最大绝对误差	0.813	0.187	0.187

在本设计中,选取放大比例因子 M 为 8,即放大 256 倍,YCbCr 的最大理论绝对误差均小于误差域值 0.500。

4 电路设计和结构优化

注意到转换的系数矩阵是常数矩阵,使用通用的乘法器可以具有高的灵活性,在改变精度或对系数进行调整时,对电路进行相应的改动很容易。但是却并不具有面积和功耗的优势,同时算法的精度和系数相对稳定在设计当中一般不会改变。所以使用分布式算法对其进行优化还是很有必要的。

4.1 已有方法

方法 1 使用加法器和线移位的方式实现变量和固定常数

的乘法操作,如图 2 所示。

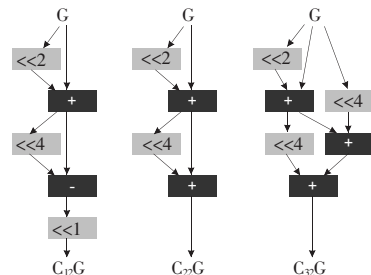


图 2 使用线移位和加法器实现的常数乘

实现 CSC 最少需要 23 个加/减法器。注意到 RGB 信号分别要与 3 个不同的常数相乘,而在这种方法中将它们孤立的对待,所以还可以减少加法器的个数。

方法 2 使用 FMC(Fused Multiplication Circuit)方式实现^[9],在这里使用 Hcub^[10]算法。注意到转换公式中 RGB 信号分别要与 3 个不同的常数相乘,在此过程中其中间计算结果可能出现冗余,可以通过共享的方式减小电路。图中“<<”表示左移位,“+”和“-”表示相加减。

CSC 中的运算部分具体实现方式如图 3 所示。再对中间结果加减运算得到最终的值。共需要 17 个加/减法器。最长的路径需要经过 5 个加/减法器。

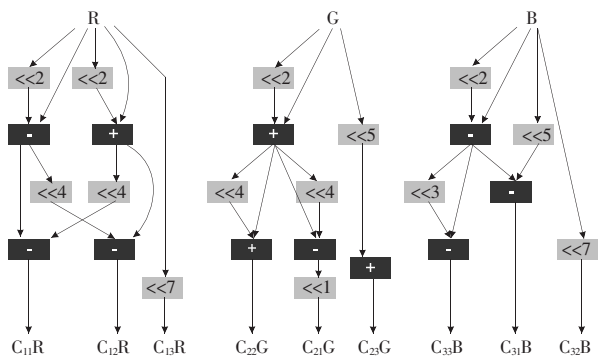


图 3 Hcub 算法实现的数据流图

考虑到人眼视觉系统的特点及视频压缩的实际应用需要,在 YCbCr 空间中的最常用的是 4:2:0 和 4:2:2 的采样率^[2]。很大一部分 CbCr 信号将不会被采样到,但是生成这些信号却消耗了能量。对于 8×8 的宏块,即使在 4:2:2 和 4:2:0 的采样模式下,同样需要完成 64 点色彩空间转换所需的乘法加法运算。对于 FMC 方法的实现来说电路中 17 组信号分别翻转 64 次,即 64×17 组次。

4.2 本文改进方法

方法 3 一种直接的方法是增加使能控制逻辑,使 CbCr 部分只对采样的点进行运算。但是上面的电路在 RGB 空间实现了共享,如图 4 所示,计算 Y 信号的电路和计算 CbCr 的电路融合在一起,彼此不独立,这样就无法让尽可能多的加法器停止翻转。在 4:2:2 和 4:2:0 的采样模式下分别翻转 64×13 和 64×11 组次。这种方法控制逻辑略显复杂,对于功耗的优化不十分明显。

方法 4 (YUVShare)考虑到上面方法中的不足,将乘加运算在 YCbCr 空间进行优化实现资源共享,并使 YCbCr 信号的生成完全独立,这样就可以在不需要 CbCr 信号时,完全阻止其工作,从而在低采样率时,减少电路不必要的翻转,达到降低功

耗的目的。

手动优化加法器使其在 YCbCr 空间实现加/减法的共享^[5],如图 5 所示。

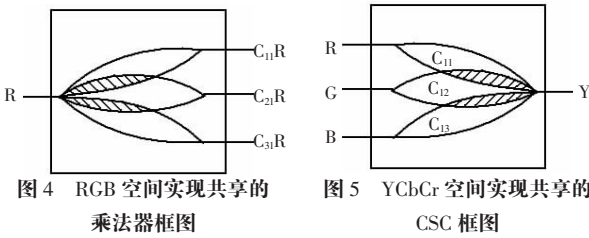


图4 RGB空间实现共享的乘法器框图

图5 YCbCr空间实现共享的CSC框图

数据流如图 6 所示, Y 与 CbCr 完全独立, 增加少量的采样控制逻辑, 就可以使其工作在不同地采样模式下, CbCr 部分的电路只在采样时工作, 对于未采样点可以完全关闭。可以根据需要在某些虚线处对其进行流水操作。总共需要 19 个加/减法器, 在 4:2:2 和 4:2:0 的采样模式下分别翻转 64×12 和 64×9 组次, 只占方法 2 中的实现的 12/17 和 9/17。

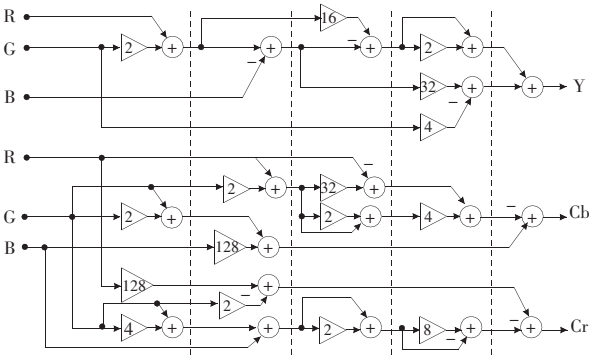


图6 YCbCr空间实现共享的CSC结构图

不同方法所占资源和路径深度定性比较如表 2。

表2 不同方法所占资源和比较

方法	资源	深度
通用	3 乘法器+6 加法器	3
方法 1	23 加法器	5
方法 2,3	17 加法器	5
方法 4	19 加法器	5

5 FPGA 实现及功耗分析

FPGA 的功耗主要有两个来源:

(1) 静态功耗来自于器件保持当前状态的漏电流。随着器件尺寸的减小和低电压的应用, 静止状态时器件沟道的亚域值漏电流增大, 导致静态功耗增加^[6]。

(2) 动态功耗来自于器件翻转时电路内部互连线和器件电容的充放电。动态功耗随着频率的增加而增加。计算公式如式 (4) 所示, C 为和电容大小相关的系数, V 为电路的工作电压, F 为翻转频率。在本文所使用的方法中, 通过减小部分电路的翻转次数, 达到减小动态功耗的目的。

$$Power_{\text{r}} = C \times V^2 \times F \quad (4)$$

本文中所涉及的不同方法的 CSC 均使用 Verilog 实现, 8 位输入输出 8 位系数。使用 CycloneII 器件, QuartusII 综合后的

网表加仿真生成的 asf 文件, 使用 QuartusII 自带的功耗分析工具 PowerPlay Power Analyzer^[7] 进行分析。

CSC 各种实现的结果如表 3 所示, 单元个数和延时信息由综合工具得到。

表3 不同方法 CSC 实现结果

方法	单元(LE)	延时/ns	功耗/nW
方法 1	255	9.02	14.53
方法 2	232	9.32	13.76
方法 4	226	10.00	13.25

注: 功耗表示逻辑单元的动态功耗, 是在频率为 100 MHz, 采样率为 4:4:4 模式下得到的结果, 不同采样率得到的结果如表 4。

表4 方法 4 在不同采样率下的比较

频率		4:4:4	4:2:1	4:2:0
100 M	组合逻辑/nW	13.25	10.32	9.07
	翻转率/(Mconv/s)	190.66	146.15	129.06
50 M	组合逻辑/nW	5.97	4.59	3.52
	翻转率/(Mconv/s)	85.36	63.75	50.58

6 结论

本文对 CSC 的不同实现方法进行了比较, 并提出了一种优化方法。

动态功耗和器件的翻转率相关, 同时翻转率和所施加的激励相关。这里的激励是随机函数生成的伪随机数, 因为对于不同方法的实现所使用随机种子是相同的, 所以激励是相同的, 也就是电路工作的环境是一样的。

因为使用的是随机函数, 导致翻转率较大, 而实际的工作情况是, RGB 空间的数据具有很强的相关性, 实际翻转率同施加随机数所得的翻转率相比较要小, 但是 YUVShare 的方法在低采样率时对于节省功耗的相对比例是不变的。

参考文献:

- [1] Bartkowiak M. Optimizations of color transformation for real time-video decoding[C]//Digital Signal Processing for Multimedia Communications and Services: EURASIP ECMCS 2001, Budapest, September 2001.
- [2] Richardson I E G. H.264 和 MPEG-4 视频压缩系统[M]. 欧阳合, 韩军, 译. 长沙: 国防科技大学出版社, 2004: 17.
- [3] Tummeltshammer P, Hoe J C, Puschel M. Time-multiplexed multiple-constant multiplication[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2007, 26(9): 1551-1563.
- [4] Gustafsson O, Dempster A G, Wanhammar L. Extended results for minimum-adder constant integer multipliers[C]//IEEE International Symposium on Circuits and Systems, 2002, 1: 73-76.
- [5] Holm K, Gustafsson O. Low-complexity and low-power color space conversion for digital video[C]//24th Norchip Conference, 2006: 179-182.
- [6] Khan M. Power optimization in FPGA designs[Z]. Synopsys Users Group-SNUG San Jose, 2006-03.
- [7] ALTERA. QuartusII Handbook, Volume 3, Altera Corporation [EB/OL]. (2007-05). <http://www.altera.com>.