

无 SRAM 的 H.264/AVC 去块效应滤波器

李健 乔飞 罗嵘 杨华中
(清华大学电子工程系 北京 100084)

摘要: 针对 H.264/AVC 中的去块效应滤波器, 该文提出了一种新的滤波处理顺序, 能够显著减小片上数据缓存容量, 并以此为基础设计了一种去块效应滤波器的 VLSI 硬件新结构。该结构利用数据复用机制减少对片外存储的访问量、节省处理时间, 同时不使用片内 SRAM, 将对片内 SRAM 的访问降为 0。仿真结果显示, 该电路在工作频率为 100MHz 时对 HDTV 能较好地实现实时滤波; 在 0.18 μm 工艺下, 综合后的等效逻辑门数只有 16.8k。

关键词: H.264/AVC; 去块效应滤波器; 数据复用; SRAM(静态随机存储器); VLSI 设计

中图分类号: TN47; TN713

文献标识码: A

文章编号: 1009-5896(2008)08-2012-05

A SRAM-less Deblocking Filter in H.264/AVC

Li Jian Qiao Fei Luo Rong Yang Hua-zhong

(Department of Electronic Engineering, Tsinghua University, Beijing 100084, China)

Abstract: A new VLSI architecture of deblocking filter is developed for H.264/AVC system. In the presented architecture, a novel filter scheduling is proposed to reduce the size of local data buffer, and an enhanced data reuse technology is adopted to reduce the number of external memory access, thus the speed of filtering process is significantly improved as well. What's more, this architecture employs no on-chip SRAM, so there is no on-chip SRAM access. Simulation results show that the new filter can support real-time deblocking for HDTV video application when it works at 100 MHz. The synthesized logic gate count is only 16.8k with 0.18 μm CMOS technology.

Key words: H.264/AVC; Deblocking filter; Data reuse; SRAM (Static Random Access Memory); VLSI design

1 引言

H.264/AVC^[1]作为新一代的视频编码标准, 由 JVT (Joint Video Team)组织于 2003 年公布。与当前流行的编码标准 MPEG-2 相比, 由于采用了多参考帧的帧间预测、帧内预测、改进的熵编码算法及去块效应滤波等新技术, 在相同图像质量的情况下, 可以节省超过 50%的比特率^[2]。其中, 去块效应滤波器在提升性能上发挥了重要作用, 一方面可以平滑由于变换、量化及运动补偿引入的块效应, 改善图像质量; 另一方面可以使用滤波后的帧作为后续预测的参考帧, 从而减少残差数据量, 提高压缩率。文献[3]结果表明, 单独采用去块效应滤波技术后, 可以减少 9%的比特率。

去块效应滤波器对提高压缩率的贡献是以增加算法复杂度为代价的, 其计算复杂度约占解码器总运算量的 1/3^[4]。H.264/AVC 标准中, 去块效应滤波器需要对每个 4 \times 4 块的所有边界进行滤波操作, 这就意味着每个 4 \times 4 块中的数据都会被滤波 4 次, 因此需要大量的存储访问及滤波计算。文献[5]采用 H.264/AVC 标准中规定的基本滤波顺序, 同时使用片内 SRAM 缓存滤波中间结果, 以减少对片外存储的访问。

文献[5]的不足是没有利用相邻数据的相关性, 从而导致较慢的滤波速度及较大的片内 SRAM。针对以上问题, 有研究者提出了若干改进的滤波结构^[6-11], 这些改进主要集中在两个方面: 一是通过改进滤波顺序, 利用数据的相关性实现数据复用, 尽量减小片内 SRAM^[6,7,10,11]; 二是通过增加片内 SRAM 容量, 缓存更多的滤波数据, 以提高滤波速度^[8,9]。有些结构^[9,11]在采用上述改进方法的基础上, 使用流水线技术, 进一步提高滤波速度。

本文提出一种采用改进的滤波顺序的滤波结构, 可以适用于片外存储为 DDR SDRAM (Double Data Rate Synchronous Dynamic Random Access Memory)的情况; 并且可以实现数据的完全复用, 将片内 SRAM 减小到 0; 同时达到较高的滤波速度。仿真综合结果表明, 与文献[5]中的结果相比, 硬件资源的消耗减少 20%以上, 同时滤波速度提高 47%。

本文第 2 节介绍改进的去块效应滤波器的 VLSI 结构; 第 3 节通过仿真和综合实验, 论证新结构在硬件消耗、滤波速度及计算复杂度等方面的改进; 第 4 节总结全文并给出相关的结论。

2 去块效应滤波器结构

图 1 所示是传统的使用片内 SRAM 的去块效应滤波器

2006-12-30 收到, 2007-04-27 改回

国家自然科学基金重大研究计划(90207001)和 863 计划项目(2006AA01Z224)资助课题

的一般结构,各种具体实现的差别主要体现在片内 SRAM 容量、寄存器组大小及数据流控制等方面。在这种结构中,片内 SRAM、移位寄存器组和转置寄存器组是不可缺少的:片内 SRAM 用于缓存滤波的中间结果;移位寄存器组负责向边界滤波器直接提供数据;垂直边界滤波和水平边界滤波使用的数据格式不同,需要使用转置寄存器调整格式。

本文提出的符合 H.264/AVC 标准的去块效应滤波器总体结构如图 2 所示,图中实线为数据流方向,虚线为控制流方向。与传统的结构(图 1)相比,片内存储部分没有使用 SRAM 及转置寄存器,仅有移位寄存器组。该结构可分为片内存储器、数据通路控制和边界滤波器 3 大部分。片内存储器包括 2 个 16×32 bit 的寄存器组及 1 个 4×32 bit 的寄存器组,其中 16×32 bit 的寄存器组用于寄存当前宏块内及当前宏块左侧的数据,记为 RegP 和 RegQ; 4×32 bit 的寄存器组用于寄存当前宏块上侧所需数据,记为 RegT。在不改变滤波结果的前提下,通过重新安排滤波顺序及合理的数据存储组织,仅使用上述的 3 个寄存器组即可实现数据的完全复用。下面详细介绍各部分的具体实现方法。

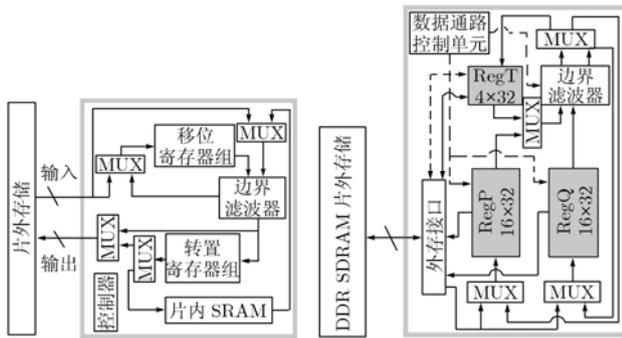


图 1 传统的使用片内 SRAM 的结构框图

图 2 本文提出的去块效应滤波器 VLSI 结构

2.1 改进的滤波顺序

H.264/AVC 标准中,去块效应滤波是基于 4×4 块进行的。标准中规定了相应的滤波顺序,如图 3(a)所示:先从左向右对块的垂直边界进行水平滤波,然后从上向下对块的水平边界进行垂直滤波。文献[5]采用这种基本的滤波顺序,同时为减少对片外存储的访问,使用片内 SRAM 缓存滤波的中间结果。

H.264/AVC 标准中规定的这种滤波顺序,为了减少对片外存储的访问,必须采用较大的片内 SRAM 缓存滤波的中间结果,而较大的片内 SRAM 一方面会增大面积,同时频繁的片内 SRAM 访问也会加大功耗。针对以上问题,有研究者提出了若干改进的滤波顺序^[7,10];在不改变滤波结果的前提下,可以增加数据的复用性,减少片内 SRAM 的面积和对片内 SRAM 的访问次数。

文献[7]中改进的滤波顺序如图 3(b)所示。在这种滤波

顺序下,相邻 4×4 块间的滤波中间结果可以复用,而不必存入片内 SRAM。因此,对片内 SRAM 的访问次数减少为文献[5]的一半。基于文献[7]的工作,文献[10]提出的滤波顺序,可以实现对数据的完全复用,如图 3(c)所示。文献[10]所需片内 SRAM 仅为文献[7]的 1/10,对片内 SRAM 的访问下降为文献[7]的 1/3。

以上的滤波顺序都是基于 4×4 块的,每次从片外存储中读取少量数据存入寄存器组中,因此并不适用于片外存储是 DDR SDRAM。因为 DDR SDRAM 的读写需要一定的建立时间,一次读多组数据平均每组数据所需的时钟周期数会远小于一次读一组数据所需的时钟周期数。在本设计的 DDR SDRAM 仿真测试中发现:一次读 4 组数据,需要 24 个时钟周期;一次读 8 组数据,需要 32 个时钟周期;而一次读 16 组数据,则仅需 42 个时钟周期。所以,在片外存储为 DDR SDRAM 时,需一次读入多组数据。另外,上述几种滤波顺序都无法消除片内 SRAM 的存在。

图 3(d)是本文提出的适用于片外存储为 DDR SDRAM 时的滤波顺序。使用这种滤波顺序,可一次读入 8 组/16 组数据;可将片内 SRAM 降为 0,并实现数据的完全复用。处理过程如下:滤波首先从垂直边界 1~4 开始,一次读入 16 组数据,将(b,f),(c,f),(d,f)和(e,f)中的数据存入 RegP,将(b,g),(c,g),(d,g)和(e,g)中的数据存入 RegQ,然后对 RegP 和 RegQ 中的数据执行垂直边界滤波操作,滤波的中间结果仍然存放在 RegP 和 RegQ 中;当执行垂直边界 5~8 及水平边界 9~12 的滤波时,可从 RegQ 中获得(b,g),(c,g),(d,g)和(e,g),而不必从片内 SRAM 中读入。

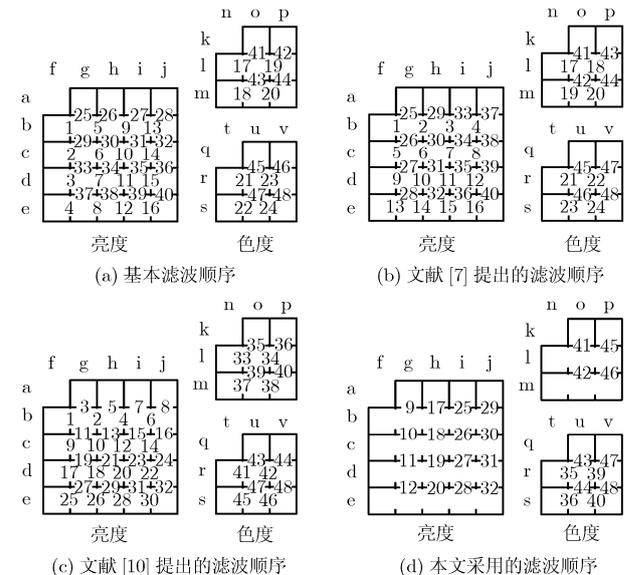


图 3 宏块的滤波顺序

2.2 新的数据存储组织

在前面介绍的几种滤波结构中,为了实现数据复用,需

要使用片内 SRAM 缓存滤波后的中间结果。而本文提出的这种新的滤波数据组织, 无需额外片内存储, 即可实现完全的数据复用。

下面以亮度分量为例来说明本文提出的这种新的数据存储组织:

假设 RegP 存放图 3(d) 中 5~8 边界左侧的待滤波数据, RegQ 存放 5~8 边界右侧的待滤波数据。

首先执行垂直边界滤波操作: 从 RegP 和 RegQ 寄存器组的上端向边界滤波器提供数据(p0~p3, q0~q3), 在提供数据的同时, 将寄存器组中的数据向上移动一个像素单元; 滤波后的中间结果通过回传路线传到寄存器组的下端, 存储在寄存器组的最下面; 具体操作如图 4 所示。这个操作重复 16 次, 完成 5~9 边界的垂直滤波。

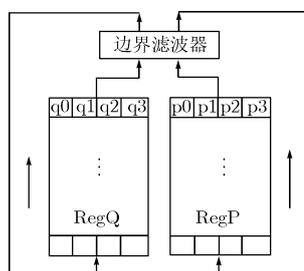


图4 垂直边界滤波操作示意图

垂直边界滤波操作完成后, 其滤波结果存储于 RegP 和 RegQ 中, 而 RegP 中存放的数据恰好是图 3(d) 中 9~12 边界水平滤波所需要的数据。然后用 RegP 和 RegT 中的数据完成 9~12 水平边界滤波操作。水平边界滤波操作的中间结果仍然存放在 RegP 和 RegT 中, 过程和垂直边界滤波类似。水平边界滤波完成后, RegT 和 RegP 中的数据是已完成全部滤波操作(垂直和水平)的数据, 可写入片外存储。然后重新读入下一个 16×32 bit 的数据, 开始下一轮的垂直、水平边界滤波操作。

对于色度分量, 同样遵循以上算法流程。

2.3 数据通路控制

本设计的控制单元, 尽量做到模块化、规则化, 大大降低硬件的复杂度; 另外, 在做垂直/水平边界滤波的同时, 读/写上侧相邻宏块中的 4×4 块滤波数据(图 3(d) 中的(a, g),(a, h),(a, i)和(a, j)), 通过并行操作, 减少一定的时钟周期开销。

数据通路控制单元用一个有限状态机控制边界滤波器, RegP, RegQ 和 RegT 的同步, 共有 5 个不同的状态。各状态如下:

(1) 准备垂直边界滤波数据, 首先判断是否是宏块的边界, 若是, 则读入该边界左右两侧数据分别存入 RegP 和 RegQ; 若不是, 则读入该边界右侧数据存入 RegQ;

(2) 对 RegP 和 RegQ 中的数据做垂直边界滤波, 同时读入上侧相邻宏块 4×4 块的数据存入 RegT; 滤波的中间结果

仍然存放在 RegP 和 RegQ 中, 具体数据存储组织参见 2.2 节;

(3) 对 RegP 和 RegT 中的数据做水平边界滤波; RegT 中的数据滤波完成后, 即可马上写入片外存储, 与后面的水平边界滤波操作并行执行;

(4) 将 RegP 中的数据写入片外存储;

(5) 将 RegQ 中的数据复制到 RegP 中。

完成上面的 5 步操作后, 即完成了一组垂直边界与水平边界的滤波操作。对同一宏块的其余边界执行类似的操作, 即可完成一个宏块的滤波操作。

2.4 边界滤波器

边界滤波器是电路中实际执行滤波运算的电路单元, 输入输出都是 4 个像素点(32 bit)。通过向边界滤波器提供相应的输入数据, 垂直边界和水平边界滤波可以共用此电路。本设计中的边界滤波器和 H.264/AVC 标准中规定的滤波操作一致, 这里不再赘述。

3 仿真与综合结果

对本文提出的去块效应滤波器进行可综合的 RTL 级建模, 并使用 Modelsim 进行仿真验证。测试数据从 JVT 的官方参考软件^[12]中导出, 本设计的滤波器输出结果与文献^[12]中的滤波结果进行对比, 达到比特级的全部匹配, 由此可以证明本设计的功能完全正确。图 5 为滤波器功能仿真测试过程模型。

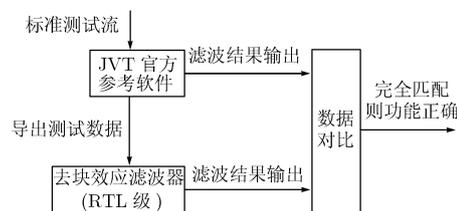


图5 滤波器功能测试过程模型

采用 SMIC 0.18 CMOS 工艺对本文提出的 H.264/AVC 标准中的去块效应滤波器进行综合。仿真及综合结果见表 1。

由于参考文献中使用的片外存储基本上都是 SRAM, 为了在相同条件下进行比较, 本文的仿真数据是使用片外 SRAM 时的结果。表 1 列举了本设计和其它设计的比较结果。与去块效应滤波器的经典实现文献^[5]相比, 综合后的等效逻辑门减少 20% 左右, 同时滤波速度提高 47% 左右, 另外, 本设计不含片内 SRAM, 而文献^[5]的等效逻辑门中并不包括片内 SRAM, 因此, 与文献^[5]相比, 本设计在硬件资源消耗和滤波速度方面都取得了极大的改进。文献^[8,9]是典型的通过增加片内 SRAM, 缓存大量数据来提高滤波速度的设计, 这两篇文献的滤波速度比本文快 20% 左右, 但综合后的等效逻辑门比本文增加了 17~20%, 若解码 HDTV 格式的图像, 使用的片内 SRAM 在 100kbit 左右, 若按照 1bit 的片内

表1 与其它设计的结果比较

	文献[5]	文献[6]	文献[7]	文献[8]	文献[9]	文献[10]	文献[11]	本文
周期数/宏块	614	446	N/A	250	214 或 246	300	192	324
4×4 像素阵列数	4	8	2	4	2	2	2	9
流水线级数	N/A	N/A	N/A	N/A	5	N/A	4	0
工艺类型(μm)	0.25	0.25	0.18	0.18	0.18	0.25	0.13	0.18
适用的片外存储器类型	SRAM	SRAM	SRAM	SRAM	SRAM	SRAM	SRAM	SRAM SDRAM
等效逻辑门(k)	20.66	24	9.16	19.64	20.9	13.41	7.5	16.8
片内 SRAM	2P 96×32 2P 64×32	DP 64×32 2×2P 96×32	1P 80×32	2×1P 96×32 1P (2×FW) ×32	1P 96×32 2P 32×32 1P (1.5×FW) ×32	2P 16×32	DP 32×32	0

注(1)DP: 有两个读写端口的双端口 SRAM; 2P: 有一个读端口一个写端口的双端口 SRAM; 1P: 有一个读写端口的单端口 SRAM; (2)等效逻辑门不包括片内 SRAM; (3)FW 代表帧的宽度(FW=Frame Width); (4)1 个 4×4 像素阵列包含 4×4×8bit

SRAM 等价于 1 个等效逻辑门计算, 其总的硬件资源的消耗是本文的 7 倍左右。因此, 与文献[8,9]相比, 本文的设计在速度和面积方面取得了良好的折衷。文献[11]是目前做的最好的一个设计, 其滤波速度很快并且硬件资源消耗极小, 但在片内 SRAM 资源严重受限或片外存储为 DDR SDRAM 的情况下, 文献[11]的结构无法直接应用, 这时就显示出本文片内 SRAM 为 0, 接口适合片外存储为 DDR SDRAM 的优势了。

SRAM 虽然速度快, 但是其价格高、功耗大, 在实际应用中有一定的局限性。DDR SDRAM 具有价格便宜、功耗较低等优点, 是片外存储的良好选择, 但因速度稍慢, 制约了其在实时滤波处理中的应用。本文提出的这种滤波顺序及结构, 非常适合片外存储为 DDR SDRAM 的情况, 可有效减少对片外存储的访问次数, 提高滤波速度, 使 DDR SDRAM 能够有效的用于实时滤波处理。在片外存储使用 DDR SDRAM 的情况下, 本设计每宏块所需周期数为 474, 对 HDTV(1920×1088)格式的图像进行滤波, 处理速度为 25.8 帧/秒, 可以达到良好的实时滤波处理, 因而具有较高的实用价值。

4 结束语

本文实现了 H.264/AVC 中去块效应滤波器的 VLSI 设计。在不使用片内 SRAM 的前提下, 利用相邻 4×4 像素块间数据的依赖关系, 合理组织滤波顺序及数据存储, 实现数据的复用, 使读写片外 DDR SDRAM 的次数减半, 从而减少滤波处理的周期数, 同时降低滤波器操作的复杂度。与以往的设计相比, 不但减少了对片外存储的访问次数, 而且将

对片内 SRAM 的访问降为 0, 有助于降低功耗。仿真及综合结果表明, 本设计的显著优点是硬件资源消耗少, 控制电路简单, 滤波速度较快等。这些优点使其能较好的用于 HDTV(1920×1088 @ 26fps)的实时滤波。

参考文献

- [1] ITU-T Rec.H.264 ISO/IEC 14496-10:2003, Draft ITU-T Recommendation and Final Draft International Standard of Joint Video specification.
- [2] Wiegand T, Sullivan G J, and Bjøntegaard G, *et al.* Overview of the H.264/AVC video coding standard. *IEEE Trans. on Circuits and Systems for Video Technology*, 2003, 13(7): 560-576.
- [3] List P, Joch A, and Lainema J, *et al.* Adaptive deblocking filter. *IEEE Trans. on Circuits and Systems for Video Technology*, 2003, 13(7): 614-619.
- [4] Horowitz M, Joch A, and Kossentini F, *et al.* H.264/AVC baseline profile decoder complexity analysis. *IEEE Trans. on Circuits and Systems for Video Technology*, 2003, 13(7): 704-716.
- [5] Huang Yu-Wen, Chen To-Wei, and Hsieh Bing-Yu, *et al.* Architecture design for deblocking filter in H.264/JVT/AVC. *Multimedia and Expo, 2003.ICME '03, Los Alamitos, California, 2003: 693-699.*
- [6] Sheng Bin, Gao Wen, and Wu Di. An implemented architecture of deblocking filter for H.264/AVC. *Image Processing, 2004.ICIP '04, Piscataway, Singapore, 2004: 665-668.*

- [7] Cheng Chao-Chung and Chang Tian-Sheuan. An hardware efficient deblocking filter for H.264/AVC. *Consumer Electronics*, 2005.ICCE, Nanyang, Singapore, 2005: 235-236.
- [8] Liu Tsu-Ming, Lee Wen-Ping, and Lin Ting-An, *et al.* A memory-efficient deblocking filter for H.264/AVC video coding. *Circuits and Systems*, 2005.ISCAS, Kobe, Japan, 2005: 2140-2143.
- [9] Shih Shen-Yu, Chang Cheng-Ru, and Lin Youn-Long. A near optimal deblocking filter for H.264 advanced video coding. *Asia and South Pacific Design Automation Conference*, Yokohama, Japan, 2006: 170-175.
- [10] Cheng Chao-Chung, Chang Tian-Sheuan, and Lee Kun-Bin. An in-place architecture for the deblocking filter in H.264/AVC. *IEEE Trans. on Circuits and Systems*, 2006, 53(7): 530-534.
- [11] Khurana G, Kassim A A, and Chua Tien Ping, *et al.* A pipelined hardware implementation of in-loop deblocking filter in H.264/AVC. *IEEE Trans. on Consumer Electronics*, 2006, 52(2): 536-540.
- [12] JVT. H.264/AVC Reference Software JM8.6 [EB/OL]. <http://ip.hhi.de/>, 2003.
- 李 健: 男, 1982 年生, 硕士生, 研究方向为视频解码和 VLSI 电路设计.
- 乔 飞: 男, 1977 年生, 助理研究员, 研究方向为低功耗集成电路关键技术、低功耗媒体处理芯片设计和低功耗传感器关键技术.
- 罗 嵘: 女, 1970 年生, 副教授, 研究方向为 VLSI 设计技术、嵌入式系统设计、电子设计自动化等.
- 杨华中: 男, 1967 年生, 教授, 研究方向为微系统芯片的新结构、面向通信和媒体处理的芯片设计、微系统芯片的综合与验证和模拟及混合信号系统设计等.