

# 一种新的用于 H. 264/AVC 的运动估计 VLSI 结构

郑兆青<sup>1)</sup> 桑红石<sup>1)</sup> 赖晓玲<sup>1)</sup> 沈绪榜<sup>1),2)</sup>

<sup>1)</sup>(华中科技大学图像识别与人工智能研究所 武汉 430074)

<sup>2)</sup>(西安微电子技术研究所 西安 710054)

**摘 要** 提出了一种新的用于 H. 264/AVC 的分数运动估计 VLSI 结构. 首先改进分数运动估计算法的迭代顺序, 将 1/2 像素精度和 1/4 像素精度的串行搜索改为同时进行, 设计出一种空间上具有更高并行度的 VLSI 结构; 另外该结构不需要输入和输出 1/2 插值数据, 因此减少了存储带宽需求. 该结构计算 H. 264/AVC 中一个宏块 1/4 像素精度的 41 个运动向量需要 1344 个时钟周期. 在 0.18 $\mu$ m CMOS 工艺下作了逻辑综合和仿真, 结果表明时钟频率为 147MHz 时, 面积为 276k 门, 能够满足 SDTV(1280 $\times$ 720)@30Hz 的视频编码需要. 相比现有结构, 该结构降低了存储访问带宽需求, 提高了数据吞吐率.

**关键词** H. 264; VLSI 结构; 块匹配; 运动估计; 视频编码

中图法分类号 TP311

## A New VLSI Architecture for Fractional Motion Estimation of H. 264/AVC

ZHENG Zhao-Qing<sup>1)</sup> SANG Hong-Shi<sup>1)</sup> LAI Xiao-Ling<sup>1)</sup> SHEN Xu-Bang<sup>1),2)</sup>

<sup>1)</sup>(Institute for Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology, Wuhan 430074)

<sup>2)</sup>(Xi'an Microelectronics Technology Institute, Xi'an 710054)

**Abstract** This paper presents a new VLSI architecture for fractional motion estimation of H. 264/AVC. The first time changed loop order of fractional motion estimation algorithm, from 1/2-pixel and 1/4-pixel sequential search to parallel search. This improvement results in designing a high parallel VLSI architecture. And this architecture reduces the requirements for memory access because it doesn't need to input and output 1/2 interpolation data. It takes 1334 cycles to find 41 motion vectors of one macro-block with 1/4-pixel precision. This architecture has been designed and synthesized in HJIC 0.18 $\mu$ m technology. When operating at 147MHz, it consists of 276k gates and can meet SDTV(1280 $\times$ 720)@30Hz video coding requirements. Compared with other architectures, it reduces requirements for memory access and improves data throughput.

**Keywords** H. 264; VLSI architecture; block matching; motion estimation; video coding

## 1 引 言

H. 264/AVC 视频编码标准是由 ITU-T 和 ISO/IEC 联合开发, 相比以前的视频压缩标准, 压缩效率至少提高了两倍, 但是大大增加了计算复杂

度, 特别是运动估计采用了一些新特性, 如 7 种块尺寸、1/4 像素精度和多参考帧等<sup>[1-2]</sup>. 实验结果表明运动估计占 H. 264/AVC 整个编码过程 60%~80% 的时间<sup>[3]</sup>. 对 baseline level2 的 CIF 格式, 针对 5 个参考帧, 搜索范围为  $\pm 16$ , 利用 JM7.3 进行实验, 结果表明分数运动估计占用整个运动估计 45%

收稿日期: 2006-05-22; 最终修改稿收到日期: 2007-07-05. 本课题得到湖北省自然科学基金(2006ABA087)资助. 郑兆青, 男, 1975 年生, 博士, 研究方向为图像处理、视频编码和 VLSI 结构. 桑红石(通信作者), 女, 1970 年生, 博士, 副教授, 主要研究方向为图像处理、计算机结构. E-mail: sanghs@gmail.com. 赖晓玲, 女, 1982 年生, 硕士, 研究兴趣为 VLSI 设计. 沈绪榜, 男, 1933 年生, 教授, 博士生导师, 中国科学院院士, 长期致力于嵌入式计算机和芯片的研究.

的时间,而 PSNR 能够提高 4+dB.但是分数运动估计的编码过程与以前的标准不同,必须与整数运动估计宏块流水处理,用专用模块实现.因此,为了提高编码器的实时性,必须利用 VLSI 结构来开发算法的并行性.目前文献中整数运动估计 VLSI 结构已经能够满足高性能视频编码要求,本文研究如何用 VLSI 结构来加速分数运动估计模块,避免分数运动估计成为运动估计模块的整体性能瓶颈.

目前文献中有几个针对 H. 264/AVC 分数运动估计的 VLSI 结构.文献[4]利用一个  $16\times 16$  宏块的整数运动向量(MV)计算  $92\times 92$  搜索范围内所有块尺寸的 41 个分数精度 MV.由于实际上整数搜索后的 41 个 MV 可能向任何地方,而这种方法限制了搜索范围,因此会降低率失真性能,尤其是对于有剧烈运动的视频序列.文献[5]利用先求 41 个整数 MV,然后再插值  $1/2$  像素计算 41 个半像素 MV,最后再插值  $1/4$  像素计算 41 个  $1/4$  像素 MV.这种顺序方法的特点是在计算  $1/2$  像素 MV 后还要保留  $1/2$  像素的插值结果,用于后面  $1/4$  像素插值,导致控制复杂和浪费存储带宽.文献[6]也利用层次搜索设计了一个 4 并行的分数运动估计 VLSI 结构,但是同样存在层次搜索方法的缺点.文献[7-8]分别提出了  $1/4$  像素插值器结构,其中文献[7]属于串行结构,每次插出一个整数点的周围分数像素,速度慢,而文献[8]属于并行结构.文献[9]提出了一种低空间复杂度的  $1/4$  插值方法.这些插值结构主要是从解码器应用考虑,难以与运动估计的处理单元相配合.

存储访问量是分数运动估计的最大问题,存储带宽的增加要求使用高性能的存储器,这在一定程度上增加了存储器代价,另一方面存储量的增加也在使耗费在访问存储器上的功耗大大增加,从而提高了整体系统的功耗.本文通过改变分数运动估计迭代循环顺序,一次性插值出所有  $1/2$  像素点和  $1/4$  像素点,直接送入 49 个运算单元完成  $1/4$  像素精度全搜索.这种方法的特点是提高了计算的并行性,降低了存储访问带宽.

## 2 H. 264/AVC 分数运动估计过程

### 2.1 树形结构运动估计

H. 264/AVC 采用树形结构运动估计,这种方法支持变块尺寸,范围从  $4\times 4$  到  $16\times 16$ .对于亮度分量,每个  $16\times 16$  宏块(MB)可以有  $16\times 16, 16\times 8, 8\times 16$  或者  $8\times 8$  块尺寸.  $8\times 8$  块能够进一步划分

为  $8\times 4, 4\times 8$  或者  $4\times 4$  块尺寸(见图 1).树形结构运动估计提供了选择块尺寸的灵活性,另外,也允许组合不同的块尺寸来匹配图像序列中不同形状的对象.因此,树形运动补偿方法的编码效率明显好于以前的标准.

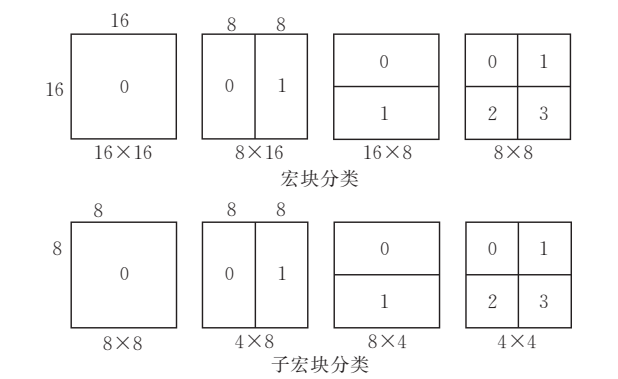
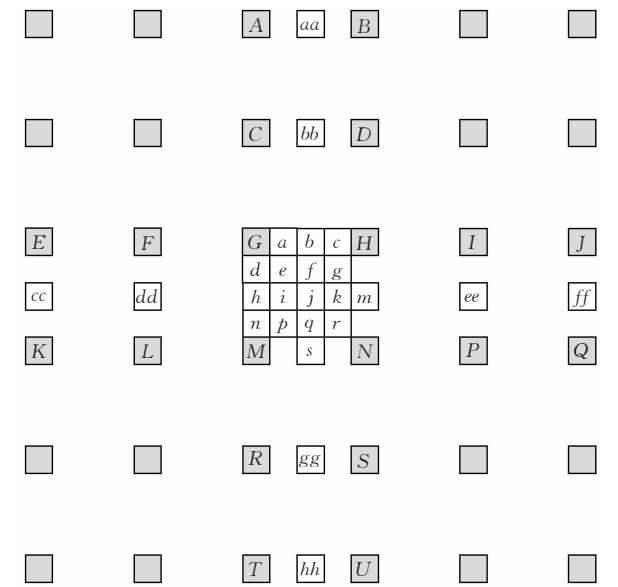


图 1 树形运动估计划分

### 2.2 分数像素运动向量

帧间编码宏块的每一个子宏块都由一个 MV 描述.当前帧的一个子宏块的 MV 通过对参考帧中同样尺寸的一个区域预测得到,两个 MV 的偏移量被编码和发送. H. 264/AVC 中 MV 的偏移量可以达到  $1/4$  像素精度,从而消除了更多的残差,提高了运动补偿的效率.

为了计算分数精度的 MV,需要对搜索区中原本不存在的分像素数据进行插值.图 2 表示了分数像素插值方案中的像素位置.大写字母表示整数像素,小写字母表示分数像素.



(1)  $1/2$  像素插值方法  
 $1/2$  像素通过水平和垂直使用一维有限响应

(FIR)滤波器得到. H. 264/AVC 定义了 6 抽头滤波器的系数为  $(1, -5, 20, 20, -5, 1)/32$ . 每个  $1/2$  像素按如下方法产生:

位置  $b$  的  $1/2$  像素通过对最近的水平整像素, 即  $E, F, G, H, I$  和  $J$  应用 6 抽头滤波器计算得到. 同样位置  $h$  的  $1/2$  像素通过对最近的垂直整像素, 即  $A, C, G, M, R$  和  $T$  应用 6 抽头滤波器计算得到. 这些计算过程通过式(1)和式(2)描述:

$$b = (E - 5F + 20G + 20H - 5I + J + 16)/32 \quad (1)$$

$$h = (A - 5C + 20G + 20M - 5R + T + 16)/32 \quad (2)$$

位置  $aa, bb, s, gg$  和  $hh$  的  $1/2$  像素计算方法同位置  $b$ . 类似的, 位置  $cc, dd, m, ee$  和  $ff$  的  $1/2$  像素计算方法同位置  $h$ .

位置  $j$  的  $1/2$  像素通过在水平方向对  $cc, dd, h, m, ee$  和  $ff$  位置的像素使用 6 抽头滤波器计算得到, 或者在垂直方向对  $aa, bb, b, s, gg$  和  $hh$  位置的像素使用 6 抽头滤波器计算得到. 计算过程通过式(3)和式(4)描述:

$$j = (cc - 5dd + 20h + 20m - 5ee + ff + 16)/32 \quad (3)$$

$$j = (aa - 5b + 20b + 20s - 5gg + hh + 16)/32 \quad (4)$$

### (2) $1/4$ 像素插值方法

当  $1/2$  像素得到后, 可以通过在垂直、水平或者对角线方向使用线性插值得到  $1/4$  像素. 位置  $a, c, d, n, f, i, k, q, e, g, p$  和  $r$  的  $1/4$  像素如式(5)~式(16)计算:

$$a = (G + b + 1)/2 \quad (5)$$

$$c = (H + b + 1)/2 \quad (6)$$

$$d = (G + h + 1)/2 \quad (7)$$

$$n = (H + h + 1)/2 \quad (8)$$

$$f = (b + h + 1)/2 \quad (9)$$

$$i = (h + j + 1)/2 \quad (10)$$

$$k = (j + m + 1)/2 \quad (11)$$

$$q = (j + s + 1)/2 \quad (12)$$

$$e = (b + h + 1)/2 \quad (13)$$

$$g = (b + m + 1)/2 \quad (14)$$

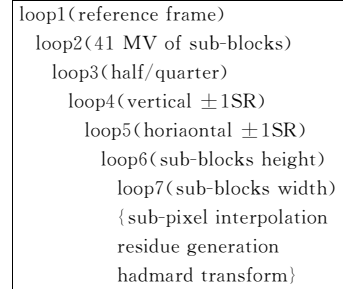
$$p = (h + s + 1)/2 \quad (15)$$

$$r = (m + s + 1)/2 \quad (16)$$

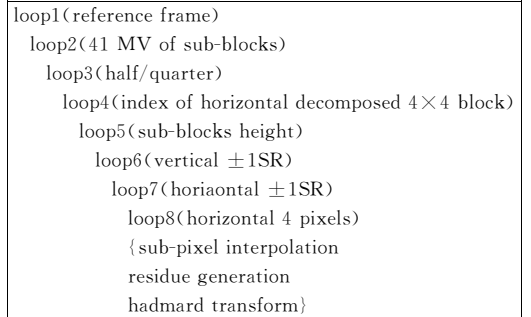
## 2.3 改进的分数运动估计过程

H. 264/AVC 首先根据 41 个最佳整数 MV 位置寻找 41 个最佳  $1/2$  像素 MV, 在此基础上再寻找

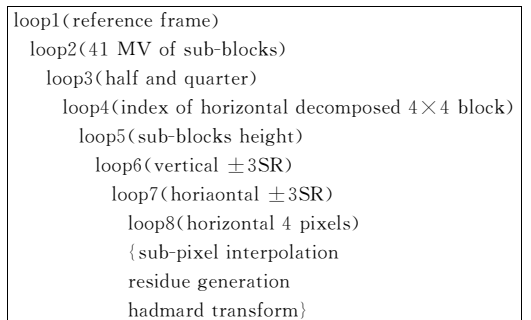
41 个最佳  $1/4$  像素 MV. 整个分数运动估计过程可以分解成 7 个循环, 如图 3(a). loop1 和 loop2 是来自不同参考帧, 不同块类型的子块. Loop3 按照先  $1/2$  像素精度, 再  $1/4$  像素精度的顺序处理每个子块. 接下去的两个循环包含水平和垂直  $\pm 1$  搜索范围(SR)内的 9 个搜索点, 最后两个循环处理每个子块中的每个像素. 这些循环的核心过程包括插值、残差产生和哈达马变换. 与残差产生相比, 插值是水平和垂直两个方向的 6 抽头 FIR 处理, 哈达马变换是 2-D  $4 \times 4$  运算. 为了满足实时处理, 必须对一些循环在硬件上并行处理. 从这个循环过程可以看出, 不同的块尺寸, 难以用一个硬件模块满足所有需要. 图 3(b)是文献[5]修改后的并行过程, loop6 和 loop7 变为并行处理, 表示 9 个搜索点并行运算. 在 H. 264/AVC 中,  $4 \times 4$  是最小的子块, 同时也是计算差的变换的绝对值和(SATD)的最小单元, 因此将所



(a) 原始的分数运动估计的 7 个循环



(b) 文献[3]中使用的利用  $4 \times 4$  分解和垂直集成的方法



(c) 本文修改的分数运动估计过程

图 3

有块尺寸分解为  $4 \times 4$  块来处理,利用垂直集成的方法避免垂直方向重复插值,这样不同的块尺寸就可以用统一的硬件模块处理. 由于这种方法还是先半像素搜索,再  $1/4$  像素搜索,因此需要存储  $1/2$  像素以便用于  $1/4$  像素插值,增加了存储带宽需求. 本文一次性把所有分数像素,即  $1/2$  和  $1/4$  像素点同时插值出来,然后同时处理一个整数 MV 周围的 48 个搜索点,见图 3(c),这里 loop6 和 loop7 变成并行处理水平和垂直  $\pm 3SR$ ,包括整数搜索中心,共有 49 个搜索点;loop8 同图 3(b)依然是水平并行处理 4 个像素,所有的块尺寸都被分解为  $4 \times 4$  基本单位来处理,这些技术避免输入输出  $1/2$  插值数据带来的存储带宽需求,同时减少控制复杂度.

3 提出的分数运动估计 VLSI 结构

3.1 整体 VLSI 结构

本文提出的分数运动估计结构是基于修改后的分数运动估计循环过程. 对 H. 264 中 7 种不同的块类型进行运动估计时,都将每 4 列划分为一个整体来连续处理. 以  $8 \times 8$  块类型为例,首先完成左边 4 列( $4 \times 8$  块)的连续计算,然后再累加右边的 4 列( $4 \times 8$  块)的计算结果. 分数运动估计的整体 VLSI 结构如图 4 所示,包括插值器(interpolation)、处理单元阵列(pu array)、累加单元(satd accumulator)和比较器(comparator tree). 插值单元负责插值出搜索所有的  $1/2$  像素和  $1/4$  像素. 处理单元阵列用

于计算匹配代价 SATD. 累加单元负责累加不同块尺寸的 SATD. 比较器用于比较 SATD 挑选出最佳 MV. 该结构串行处理不同的块尺寸,一个处理单元每个周期并行处理 4 个分数像素,共有 49 个处理单元分别对应 49 个搜索点. 下面分别介绍各个模块的结构.

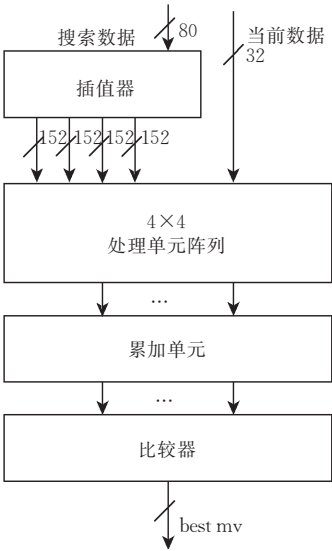


图 4 分数运动估计结构

3.2 插值器结构

用硬件实现编码器的时候,由于不可能将整帧数据放在片内存储器中,因此插值时需要将用到的参考帧数据从片外存储器中读入. 进行分数运动估计时,根据整数 MV 决定插值位置,再由插值位置决定读入的参考帧数据. 由于该结构是一次性插值

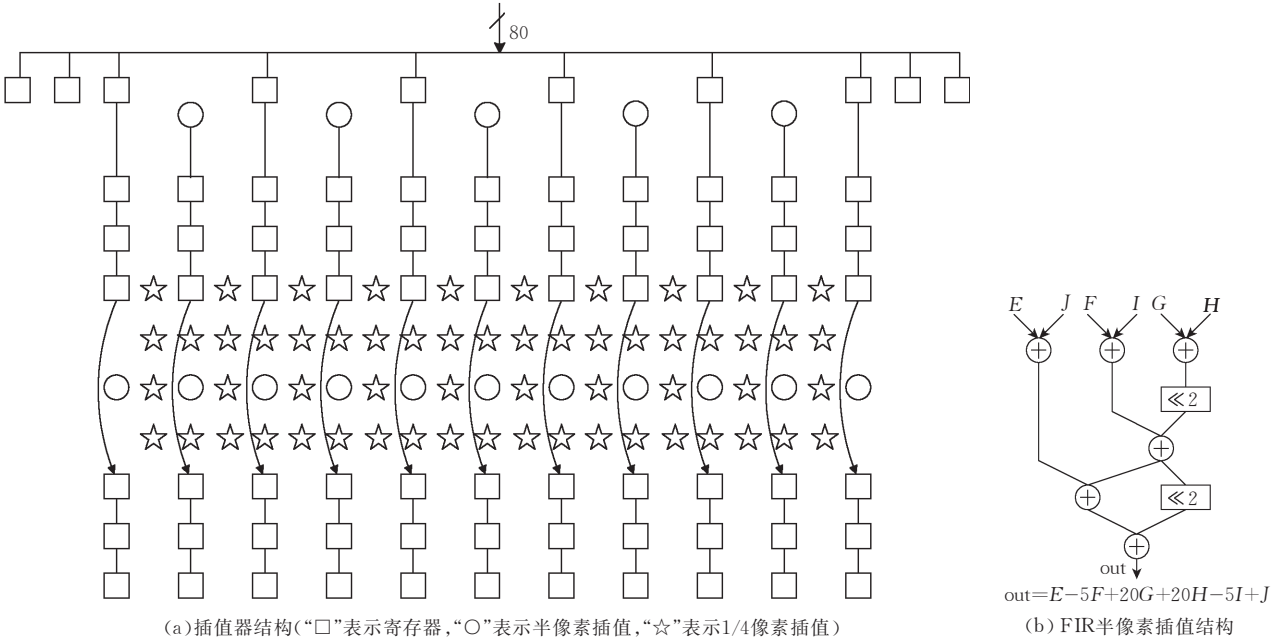


图 5

出所需的 1/2 和 1/4 像素,相比文献中先把 1/2 像素插值出来,计算 1/2 像素精度 MV,再利用同样的方法进行 1/4 像素插值和计算 1/4 像素精度 MV 的串行方法,显然降低了存储访问需求.如为了插值 1/4 像素不需要重复读入整像素和 1/2 像素,从而节约片外存储器的带宽.

插值器结构如图 5(a)所示.每个周期输入一行 10 列整数像素,用 5 个 FIR 滤波器产生相应的 5 个水平 1/2 像素.这些整数像素和 1/2 像素被送入垂直移位寄存器组中,每个周期下移一个寄存器.当垂直 6 个寄存器被缓冲满后,由 11 个水平 FIR 滤波器产生相应的 11 个垂直 1/2 像素,同时完成 1/4 像素插值,4 个整像素的下半部的所有分数像素都能够产生,共计  $19 \times 4$  个数据;随后这些插值数据被送入处理单元阵列中.该插值器每个周期完成一行 4

个整像素下半部分分数像素的插值,加上数据充满流水线需要延迟,完成一个  $4 \times 4$  块插值共需 11 个周期.对于垂直相邻的  $4 \times 4$  块,只需连续输入即可,而不需重新等待流水线充满,既减少了延迟,又提高了插值数据重用.通过这种方法可以减少 26% 的时钟周期.对于水平相邻的  $4 \times 4$  块,需要重新开始插值过程.图 5(b)给出了用于产生 1/2 像素的 FIR 滤波器的实现方式.根据 H. 264/AVC 标准,1/2 像素用相邻的 6 个数据滤波产生,  $E, F, G, H, I, J$  表示相邻的 6 个数据.该结构只需要移位器和加法器,容易实现.1/4 像素插值由于采用线性插值,比较简单,这里不再陈述.

3.3 处理单元阵列结构

与整数运动估计仅仅计算两个块的差的绝对值和(SAD)不同,分数运动估计采用 SATD 作为匹配

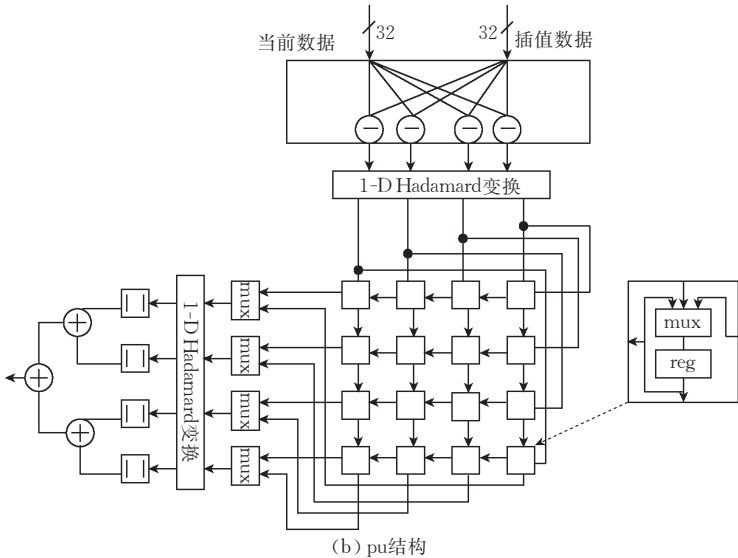
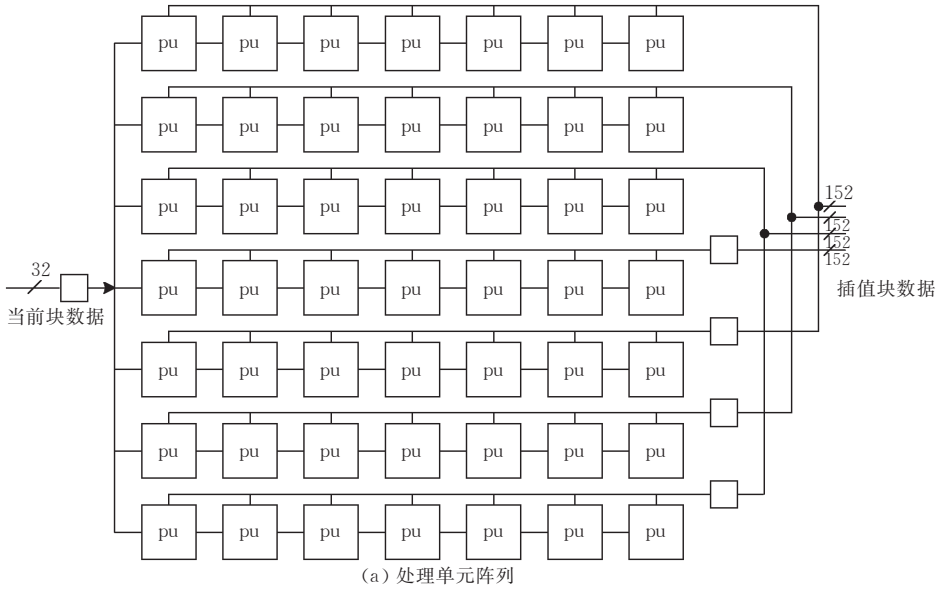


图 6

标准. 为了统一比较整数搜索点和周围的 48 个分数搜索点的 SATD, 需要 49 个处理单元并行运算. pu 阵列结构如图 6(a) 所示, 由  $7 \times 7$  个 pu (processing unit) 组成. 当前块数据通过广播方式送到每个 pu 锁存, 每周送入 4 个像素, 因此当前块数据总线为  $32(4 \times 8)$  bits. 同时, 插值器产生的搜索区数据也送到相应的 pu 中. 由于插值器每个周期只能产生  $19 \times 4$  个搜索区数据, 不足以进行一次匹配, 因此对下面四行 pu 增加了流水线延迟寄存器. 每次产生的  $19 \times 4$  个搜索区数据既送到下面四行的流水线延迟寄存器中, 也送到上面三行的 pu 中. 这样, 在首次经过一个周期的延时后, 开始流水式的最佳分数 MV 匹配. 由于送到每一行 pu 的搜索区数据是 19 个, 因此参考块数据总线的宽度为  $152(19 \times 8)$  bits. 一旦每个 pu 接受到自己所需要的当前块数据和搜索区数据, 那么就开始完成一个搜索点的计算, 共 49 个 pu 同时完成 49 个搜索点的计算, 这里每个 pu 的计算都是以垂直 4 列像素作为计算整体.

pu 结构如图 6(b) 所示. 它每次完成一行 4 个像素的残差产生和哈达马变换 (Hadamard transform), 具有 4 倍的并行, 顺序处理来自子块的分解的  $4 \times 4$  块. 一行 4 个当前块数据被广播到 pu 中, 插值器产生的 4 个参考块数据也被送入相应的 pu. 首先对应像素进行减法运算, 接着对差值进行水平哈达马变换, 送入转置寄存器向下移位, 每周期下移一位, 经过 4 个周期后, 转置寄存器中的数据开始从右向左移位, 完成垂直哈达马变化. 经过求和后送入下一个累加器模块. 转置寄存器每 4 个周期切换一次数据流方向, 这样保持运算完全流水.

累加器用于累加垂直相邻的  $4 \times 4$  块的 SATD. 当一个块尺寸垂直方向大于 4 时, 累加器连续累加即可; 当水平方向大于 4 时, 由于插值流水线要重新启动, 需要一定延迟, 因此累加器要保持以前的值, 等结果可用时再启动累加器工作. 这样能够完成不同块尺寸的 SATD 的计算. 累加器模块共有 49 个累加器并行处理 49 个搜索点的 SATD 的累加.

比较器由 48 个比较单元组成, 通过比较 49 个搜索点的 SATD 决定最佳  $1/4$  精度 MV, 最佳分数

MV 通过寄存器输出. 每个比较器结构如图 7 所示, 通过比较两个 SATD 值, 根据大小控制多路选择器输出相应的 MV.

4 工作过程

为了描述提出的分数运动估计 VLSI 结构工作过程, 以  $4 \times 8$  块尺寸搜索匹配作为例子:

(1) 每个时钟周期输入一行 10 个搜索区数据, 第 7 个周期时插值器中的寄存器已经充满, 可以同时进行  $1/2$  像素和  $1/4$  像素插值. 由于使用六抽头 FIR 滤波器进行插值, 因此要插值  $4 \times 8$  块尺寸, 那么垂直方向需要输入有效数据  $14(3+8+3)$  行, 共需要  $14 \times 10$  像素输入, 这样就能得到整个  $4 \times 8$  块的所有的  $1/2$  和  $1/4$  像素.

(2) 第 9 个周期 pu array 开始工作, 来自插值器的 19 列插值像素被分配到 7 行的 7 列 pu 阵列中, 每个 pu 仅仅使用 19 列中的 4 列像素用于计算, 具体的数据分配从插值示意图上很容易看出每个 pu 所需要的像素列, 因为每个 pu 对应一个搜索点. 转置寄存器开始向下移动数据, 经过 4 个周期后再进行水平移动数据. 即第 9~12 周期, 第一个  $4 \times 4$  块完成水平哈达马变换, 13~16 个周期第一个  $4 \times 4$  块完成垂直哈达马变换, 第二个  $4 \times 4$  块完成水平哈达马变换.

(3) 第 14 个周期累加器开始工作, 经过 8 个周期, 即第 21 个周期完成一个  $4 \times 8$  块的 SATD 的累加.

(4) 第 22 个周期并行比较器输出  $4 \times 8$  块尺寸的最佳 MV.

详细的数据流的调度和计算如表 1 所示, 阴影表示数据有效. 从表 1 中可以看出对于垂直相邻的块, 只需经过额外周期就可以输出 MV; 对于水平相邻的 4 列块, 要重复表 1 的过程.

表 1 提出的分数运动估计 VLSI 结构数据流						
时钟	插值	延迟寄存器	水平哈达马变换	垂直哈达马变换	累加器	比较器
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						

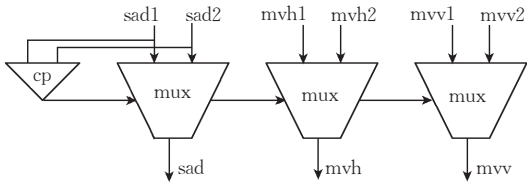


图 7 比较器原理图



(续 表)						
时钟	插值	延迟 寄存器	水平 哈达马变换	垂直 哈达马变换	累加器	比较器
12						
13						
14						
15	x					
16	x					
17	x					
18	x					
19						
20						
21						
22						

为了便于控制,每次计算前通过寄存器表示此次运算属于哪种块尺寸.对于垂直方向相邻的可以连续送入,通过控制累加器累加的次数来完成垂直相邻  $4\times 4$  的 SATD 计算,对于水平方向相邻的  $4\times 4$  需要重新启动新的计算流水线,如果垂直方向大于 4,那么需要在芯片内部记住上次累加的值,因此累加器要有控制信号使能信号,清除信号.当完成一个块模式的计算后使用清除信号,当一个块模式内部使用使能信号时,如果没有使能信号,而又没有清除信号,那么累加器保持运算结果.

无论哪种情况,累加的时间都是在输入开始后第 14 个周期.转置寄存器的多路选择器控制信号总是在第 9 个周期开始不断切换,如果多个垂直  $4\times 4$  相连,多切换几次控制信号.为了保持当前块数据连

续输入,当连续输入搜索区数据过 8 个周期开始输入当前块.从表 2 可以看出完成一个宏块的 41 个 MV 搜索需要 1344 个周期.

表 2 一个宏块的 41 个 MV 搜索所需时钟周期数

块尺寸	块尺寸标志	累加周期	块的数量	周期数/块	总周期
$4\times 4$	000	4	16	18	288
$4\times 8$	001	8	8	22	176
$8\times 4$	010	4	8	36	288
$8\times 8$	011	8	4	44	176
$8\times 16$	100	16	2	60	120
$16\times 8$	101	8	2	88	176
$16\times 16$	110	16	1	120	120
合计					1344

5 实现与性能比较

本文提出的分数运动估计 VLSI 结构使用 HJTC  $0.18\mu\text{m}$  工艺进行了逻辑综合,时钟频率可达 147MHz,面积为 276k 门.每秒可以处理 109K 宏块,能够满足 SDTV( $1280\times 720$ )@30Hz 单参考帧的视频编码需要.表 3 给出了各种 VLSI 结构性能比较,可以看出该结构数据吞吐率和处理能力得到了提高.如果想进一步提高该结构的性能,可以在关键路径上插入流水线寄存器.该结构的另外一个优点是不需要输入输出  $1/2$  插值数据,因此减少了存储带宽.

表 3 各种 VLSI 结构性能比较

	PE 数量	搜索方法	搜索点数	延迟(周期数)	工艺/ $\mu\text{m}$	频率/MHz	门数/k	(存储访问/宏块) /KByte	处理能力 /(KMB/s)
Chen <sup>[6]</sup>	$4\times 9=36$	hierarchy	18	2040	0.18	100	79	22.3	47
Rahman <sup>[4]</sup>	$8\times 16=128$	False Full	$32\times 32=1024$	$16\times 4\times 32=2048$	Virtex2 device	149	225*	—	72
López <sup>[5]</sup>	$8+8=16$	Hierarchy	16	3320	xc2v6000	54	—	22.3	16
本文	$4\times 49=196$	Full	49	1344	0.18	147	276	9.4	109

注: \*表示不包含插值模块.

6 结 论

H. 264/AVC 的分数运动估计能够有效提高预测精度,但是增加了计算复杂度,需要使用硬件提高处理速度.本文提出了一种新的用于 H. 264/AVC 的分数运动估计 VLSI 结构.该结构一次性插出所需的  $1/2$  像素和  $1/4$  像素,利用 49 个 4 并行处理单元结构并行处理.该结构使用 HJTC  $0.18\mu\text{m}$  工艺进行了逻辑综合,时钟频率可达 147MHz,面积为 276k 门.每秒可以处理 109K 宏块,能够满足 SDTV ( $1280\times 720$ )@30Hz 单参考帧的视频编码需要.该结构的优点是数据吞吐率和处理能力高,不需要存

储中间插值数据,相比以前的串行搜索结构,减少存储带宽需求,这也隐含着访问外部存储器的功耗大大减少.

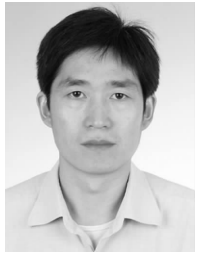
参 考 文 献

[1] Jiont Video Team. Draft ITU-T recommendation and final draft International standard of joint video specification. Geneva: ITU-T Rec H264 ISO/IEC 14496-10 AVC, 2003

[2] Wiegand T, Sullivan G J, Bjntegaard G et al. Overview of the H. 264/AVC video coding standard. IEEE Transactions on Circuits and Systems for Video Technology, 2003, 13(7):560-576

[3] Chen Zhi-Bo, Zhou Peng, He Yun. Fast Integer Pel and fractional Pel Motion estimation in for JVT//Proceedings of the

- Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, 6th Meeting, JVTf017. Awaji, Japan, 2002
- [4] Rahman C A, Badawy W. A quarter pel full search block motion estimation architecture for H. 264/AVC//Proceedings of the IEEE International Conference on Multimedia and Expo, IPiscataway, NJ 08855-1331, United States: IEEE Press, 2005; 414-417
- [5] López S, Tobajas F, Villar A et al. A quarter pixel precision motion estimation architecture for H. 264/AVC video coding//Proceedings of the VLSI Circuits and Systems II. Bellingham WA, WA 98227-0010, United States: SPIE Press, 2005; 174-184
- [6] Chen Tung-Chien, Huang Yu-Wen, Chen Liang-Gee. Fully utilized and reusable architecture for fractional motion estimation of H. 264/AVC//Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing. Piscataway, NJ 08855-1331, United States: IEEE Press, 2004; V9-V12
- [7] Deng Lei, Gao Wen, Hu MingZeng et al. An efficient VLSI architecture for MC interpolation in AVC video coding//Proceedings of the International Conference on Embedded Systems and Applications ESA'04, Proceedings of the International Conference on VLSI, VLSI'04. Bogart, GA 30622, United States: CSREA Press, 2004; 564-568
- [8] Dang Philip P. Embedded architecture for fast implementation of H. 264 subpixel interpolation//Sudharsanan S, Bove V M. Proceedings of the Embedded Processors for Multimedia and Communications II. San Jose, CA, USA: SPIE Press, 2005; 72-78
- [9] Wang Rong-Gang, Li Jin-Tao, Huang Chao, Zhang Yong-Dong. A sub-pixel motion compensation interpolation method and its high performance VLSI design. Chinese Journal of Computers, 2005, 28(12): 2052-2058(in Chinese)  
(王荣刚,李锦涛,黄晔,张勇东.一种分像素运动补偿插值滤波方法及高效 VLSI 实现.计算机学报, 2005, 28(12): 2052-2058)



**ZHENG Zhao-Qing**, born in 1975, Ph. D.. His research interests include image processing, video coding, VLSI architecture.

**SANG Hong-Shi**, born in 1970, Ph. D., associate pro-

fessor. Her research interests include computer architecture and digital image processing.

**LAI Xiao-Ling**, born in 1982, M. S.. Her research interest is VLSI design.

**SHEN Xu-Bang**, born in 1933, professor, Ph. D. supervisor, member of Chinese Academy of Sciences. He has long been engaged in the research of embedded computers and their chips.

## Background

The H. 264/AVC video compression standard, jointly developed by ITU-T and ISO/IEC, provides at least 2x compression improvement and substantial perceptual quality enhancement over all previous standards but significantly increases the computation complexity. In particular, the motion estimation results to be the most intensive task in the whole encoding process. Some new features of the standard that enable enhanced coding efficiency by accurately predicting the values of the content of a picture to be encoded are variable block-size, quarter-sample-accuracy and multiple reference picture for motion estimation and compensation. Experimental results have shown that motion estimation can consume 60% for 1 reference frame to 80% for 5 reference frames of the total encoding time of H. 264 codec. For this reason, in order to get real time performance from a H. 264 encoder, parallel processing must be exploited in the architecture.

41 fractional motion vectors (MVs) in one macro-block

(MB) are derived from one integer MV, in fact, 41 integer MVs may point to different positions. Because this method restricts search range, the performance of rate-distortion will decrease, especially for large motion video. In 41 half pixels MVs are derived from 41 integer pixels MVs, then 41 quarter pixels MVs are derived from 41 half pixels MVs. This hierarchical refining increases requirement of memory access bandwidth.

This paper proposes a new VLSI architecture for fractional motion estimation (FME) of H. 264/AVC. This architecture implements all fractional pixels interpolation and uses 49 4-parallel PEs. It can meet SDTV(1280×720)@30Hz video coding motion estimation requirements.

This project supported by the Hubei Natural Science Foundation of China(No. 2006ABA087).

The authors proposed integer motion estimation VLSI architecture for H. 264 before. These achievements help to design high performance H. 264 video codec.