

# 基于 HEXBS 算法的运动估计器的设计

肖 潇, 李 伟, 沈绪榜

(西安微电子技术研究所, 陕西 西安 710054)

**摘 要:** 运动估计是视频压缩中最重要的一环。文中讨论了运动估计的基本原理, 深入分析了 HEXBS 搜索算法及其特点与设计难点, 设计了一种满足 MPEG-4 SP@L1 标准的全并行结构的高速运动估计电路, 并通过 FPGA 验证, 系统时钟频率达到 30MHz, 性能达到了实时编码的要求。

**关键词:** 运动估计; MPEG-4 编码; HEXBS 算法; 视频压缩; 并行

**中图分类号:** TN919.81

**文献标识码:** A

**文章编号:** 1005-3751(2006)03-0017-03

## Design of Motion Estimation Circuit Based on HEXBS Algorithm

XIAO Xiao, LI Wei, SHEN Xu-bang

(Xi'an Microelectronic Technology Institute, Xi'an 710054, China)

**Abstract:** Motion estimation is the most important step in video compression. In this paper, analyzes the fundamental theory of motion estimation, uses the HEXBS search block-matching algorithm, a fast motion estimation algorithm for video coding, developed a fully parallel high speed motion estimation circuit which based on the MPEG-4 simple profile level1. The FPGA verification shows that the clock frequency is 30MHz and it reaches the demand of real-time decoding.

**Key words:** motion estimation; MPEG-4 encode; HEXBS algorithm; video compression; parallel

### 0 引 言

块匹配运动估计是视频编码标准 MPEG-1/2/4 和 H.261/263/264 的重要组成部分。块匹配运动估计的目的在于去除两个相邻视频帧之间的时间冗余, 降低编码比特率。块匹配算法就是从参考帧中寻找与当前块匹配最佳的块, 匹配通常使用计算简单的最小化匹配准则 SAD。最佳匹配块与当前块在  $x, y$  方向上的距离组成当前块的运动矢量。

过去, 大多数运动估计器都采用全搜索(FS)的块匹配方式, 因为其算法简单、数据流规则、控制简单、易于实现, 但是, 其计算量很大, 难于实现实时视频压缩处理。为了降低视频编码中运动估计的高运算复杂度, 人们提出了许多有效的快速运动估计算法, 如: 三步搜索(TTS)<sup>[1]</sup>, 新三步搜索(NTSS), 四步搜索(4SS)<sup>[2]</sup>, 梯度搜索(BBGDS)<sup>[3]</sup>, 钻石搜索(DS)<sup>[4]</sup>。最近, 又提出了一种新的快速运动估计算法——六边形搜索(HEXBS)<sup>[5]</sup>, 它采用六边形搜索模式, 获得了比其他算法更好的视频压缩效果以及最低的运算复杂度, 性能卓越。

文中在 HEXBS 搜索算法的基础上, 设计了一种全流水并行的运动估计电路, 并对 PE 做了改进, 提高了处理

能力。在 Synplify Pro 7.3.1 下综合, 用 ALTERA 公司 APEX20KE EP20K1500E 实现, 占用了 12504 个 LUT (Look Up Table), 系统时钟频率达到 30MHz, 满足 MPEG-4 SP@L1 (176×144, 15fps) 实时编码的要求。

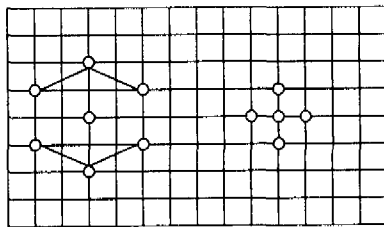
### 1 HEXBS 算法分析

在块匹配法中, 最小绝对值差(SAD)是常用的衡量最优匹配的匹配准则, 定义如下:

$$SAD(i, j) = \frac{1}{M \times N} \sum_{m=1}^M \sum_{n=1}^N |f_k(m, n) - f_{k-1}(m + i, n + j)|$$

上式中,  $(i, j)$  为位移矢量,  $f_k$  和  $f_{k-1}$  分别为当前帧和上一帧图像的灰度值, 在某一  $(i, j)$  处  $SAD(i, j)$  达到最小时, 则该点就是要找的最优匹配点。

HEXBS 算法就是一种基于此匹配准则的六边形搜索模式的快速运动估计算法<sup>[6]</sup>, 由两种搜索模式构成: 大六边形搜索模式和星形搜索模式, 如图 1 所示。



(a) 大六边形搜索模式 (b) 星形搜索模式

图 1 HEXBS 算法中的两种搜索模式

收稿日期: 2005-08-23

**作者简介:** 肖 潇(1982—), 男, 湖南娄底人, 硕士研究生, 研究方向为视频压缩和嵌入式微处理器体系结构; 沈绪榜, 博士生导师, 中国科学院院士, 研究方向为嵌入式计算机体系结构与设计。

算法首先以当前需匹配块的坐标为中心,采用六边形搜索模式,由参考宏模块构成一个大六边形,比较此 7 个块的 SAD 值,如果中心块 SAD 值为最小,则进入星形搜索模式;如果中心块的 SAD 值非最小,则以 SAD 值最小的块为中点,继续采用六边形搜索模式,直到中心块的 SAD 值为最小时,进入星形搜索模式。星形搜索模式则是以六边形模式产生的块为中心构成一个星形,比较此 5 个块的 SAD 值,SAD 值最小的块为最终匹配块。

从 HEXBS 算法的搜索过程,可以得到 HEXBS 算法在运动估计中探测点个数计算公式:

$$N_{\text{HEXBS}} = 7 + 3(n - 1) + 4$$

其中  $n$  为大六边形搜索模式使用的次数。

HEXBS 搜索算法与全搜索法相比,在计算量上有了显著降低,采用  $-7/7$  窗口,用全搜索法进行一次估计需要做  $15 \times 15 = 225$  个块的比较,而经过实验统计<sup>[5]</sup>,HEXBS 算法的计算量下降了 90% 以上,且画面质量相差无几,它是目前快速算法中性能最为优越的一种,因此采用此算法作为运动估计器设计的搜索方法。

分析 HEXBS 算法,计算量的大幅下降以及很好的画面质量,带来的是控制上花销的增加,首先,由于算法每一次搜索的起点都依赖前一步搜索产生的结果,这样数据流很不规则,只有第一步是 7 个固定位置的数据装入片上进行计算,后一步需要依赖前一步的结果,需装入数据的位置不固定,也就是要解决寻址问题,即如何产生需要访问的数据的地址。其次,HEXBS 算法之所以降低了计算量是因为其重复利用了前一步计算过的块,当在大六边形模式中,除了第一次要装载 7 个块的数据外,其他几步只要装载 3 个块的数据,这就要求考虑如何去设计电路结构,动态地分配片上存储器,调入需要的数据。最后,一对宏模块的 SAD 值计算实际就是 256 个像素对 SAD 值的累加,如果用传统的方法,进行像素对的减,绝对值运算,最后累加,则产生 MAD 值的周期太长,能否找到一种更优的结构,简化这个过程的运算且压缩这个过程的执行周期,这样就可以提高整个电路的吞吐率。

综合以上优点和难点,文中提出以下的电路结构来实现 HEXBS 搜索算法。

## 2 运动估计器电路的设计

### 2.1 算法的映射与电路基本结构

文中提出了一种 7 PE(运算单元)的并行运算电路结构,如图 2 所示。上一帧的图像数据都存储在片外的 SDRAM 中,运动估计器通过寻址,将片外 SDRAM 中存储的参考宏模块调度到片上的分布式本地存储器中,然后 7 个 PE 并行地调度片上存储器中的 7 个参考宏模块做 SAD 值计算,并将各个参考宏模块的 SAD 值输入到比较器中比较,产生一个最优匹配点,并将信息传给地址产生器,地址产生器根据这个最优匹配点计算出下一步的搜索起点的坐标地址。依以上步骤进行 HEXBS 运算,直到产

生当前宏模块的运动矢量(MV)。由上文讨论得知,地址产生单元是设计中的最大难点,PE 的性能也是一个影响系统速度的重要因素。下文着重论述这两个重要部件的设计。

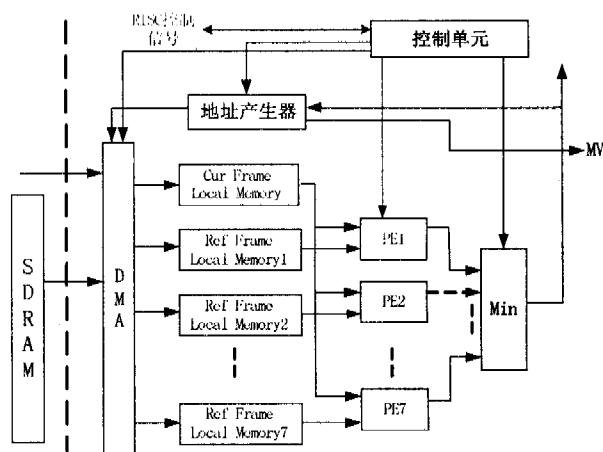


图 2 运动估计器的基本结构

### 2.2 地址产生单元的设计

由于 HEXBS 算法的每一步都要依赖前一步的计算结果,且搜索模式的不同也会产生不同的数据装载量,使得地址产生单元成为设计的难点。在本设计中,以块的坐标为基准组织计算,例如,当前要匹配块的坐标为 (12, 12),则步骤 1 中,搜索模式为大六边形,则需要到 SDRAM 中取坐标为 (12, 12), (12, 10), (10, 11), (10, 13), (12, 14), (14, 13), (14, 11) 的 7 个块并装载到片上的分布式存储器中去,同时将这 7 个坐标值按一定格式打入一个表 (Table) 中,表的结构共有 7 个字,每个字 16 位,其中 11~0 位分别装载当前片上存储器所存储待比较块的坐标值 ( $x, y$ ); 14~12 位存储此块在搜索模式中的位置,例如块在六边形正中,则标识为 1,正下标识为 5;第 15 位为中心点标识位,如是中心点,则标识为 1,每一个状态字分配给一个片上存储器,例如 (10, 11) 点在表中表示成: 0\_011\_001010\_001011。由于上文分析得知,大六边形搜索模式中有 4 个块的数据会被重复计算,设置这样一个表的结构,便于对片上存储器的动态调度,减少从外存调度的数据量。当这 7 个块与当前块分别通过数据路径计算时,会产生一个标号,传回给地址产生单元,地址产生单元则通过这个标号查表找到相对应的坐标及位置,这个坐标及位置信息就是代表这一步搜索产生的结果,并以此作为下一步的搜索起点;若进入步骤 2,则地址产生单元会根据前一步的结果及其位置信息产生下一步要装载数据块的坐标值,并在装填数据时将相关信息写入表中;若进入步骤 3,地址产生器依然通过查表来确定需要调度的块的坐标。在设计中,对每一步的控制都是通过一个相应的状态机来实现的,每一个状态机代表一个步骤,当为步骤 1 时,也就是状态机 1 计算出最优匹配块的中心块时,则激活状态机 3 运行,否则激活状态机 2 运行,状态机 1 结束,状态机 2 按大六边形模式计算直到产生中心块,激活状态机

3,本状态就结束了。状态机 3 则按星型模式产生结果。基本的结构框图如图 3 所示。

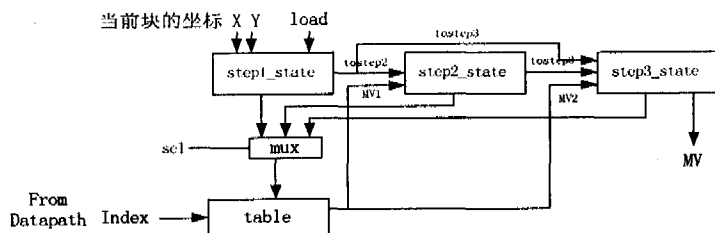


图 3 坐标计算

当然,在实际设计过程中,为加快计算的收敛,还引入了门限法技术,硬件开销很小,且对画面质量的影响也小。

### 2.3 PE 的设计

SAD 的计算公式就是先对像素对做绝对值减法,然后进行累加,实现很简单。但是,这样处理速度很慢,首先,需处理的像素对多,有 256 对,需要的周期比较长,且做绝对值减也是一个影响系统速度的步骤。如果能在一个步骤中同时做多个像素对的 MAD 计算,并将绝对值减消除掉,这样 PE 的运算速度将大为提高。在文献[6]中,提出了一种新的 MAD 计算方法,基本思想就是一次计算 16 对像素的 MAD 值,通过进位保留加法树累加,得出结果。主要步骤如下:

1)判断两个操作数,将小的操作数做位反操作。

2)所有的 32 个操作数传入 CSA(Carry Save Adder)树中进行累加。

3)将结果加以校正得出最终结果。

步骤 1)中,两个 8 位的操作数大小判断可以运用超前进位加法器的进位产生电路的原理实现,将操作数 A 先做位反,再将其和操作数 B 输入到进位产生电路,如果最高位有进位就说明  $B > A$ ,就将 A 取反。反之,就将 B 取反。

经过步骤 1)处理的操作数将被分为 16 组,输入到加法树中,进行累加,得出一个中间结果。这样一次需要做 32 个 8 位数的加法,为了加快加法操作的速度,采用了一种 4-2 压缩器为基本的加法器单元,Wallace 树型结构的加法树,从经过 Wallace 加法树,得到中间结果需要校正,加一个校正因子(这里取 16)<sup>[6]</sup>,然后取最后结果的低 13 位,这样就可以得出这 16 个像素对的 MAD 值了,也就是说,PE 可以一次快速运算宏模块中一行 16 个的像素的 MAD 值,这样,经过 16 个周期的累加就可以得到整个宏模块的 MAD 值了。

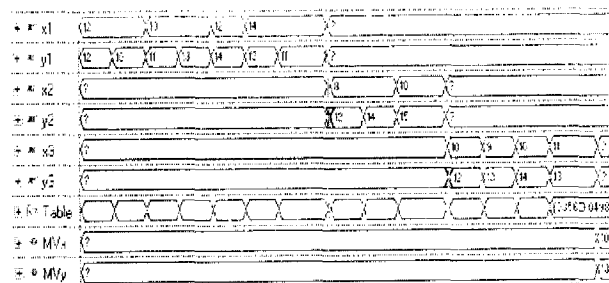
### 3 实验结果与性能分析

本设计在 ALDEC 公司的 Active-HDL 仿真环境下,用 Verilog 和原理图混合输入方式设计完成,并通过功能仿真,如图 4 所示的是坐标为 (12,12) 的宏模块匹配的部分仿真波形,计算出的运动矢量为 (10,13)。在 Synplify Pro7. 3. 1 下综合,用 ALTERA 公司 APEX20KE EP20K1500E 实现,占用了 12504 个 LUT (Look Up

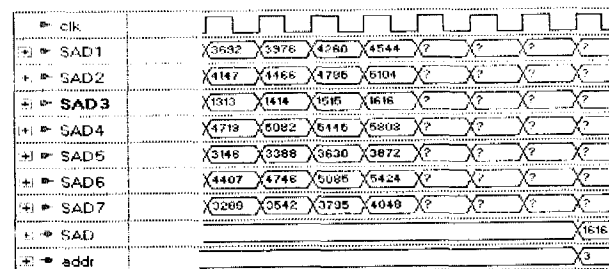
Table),系统时钟频率达 30MHz,满足 MPEG-4 SP@L1 (176×144,15fps)实时编码要求。

### 4 结束语

运动估计是视频压缩中很重要的一个环节,由于运动估计中有大量数据要处理,运动估计性能的好坏直接影响到编码器的编码质量和效率。文中设计的运动估计器采用了 HEXBS 搜索算法,并将其映射到一种 7PE 的并行结构上,通过 FPGA 验证,系统时钟频率达 30MHz,完全达到了实时编码的要求。



(a) 地址产生波形



(b) 块匹配中部分 SAD 值计算的仿真波形

图 4 坐标为 (12,12) 的宏模块匹配的部分仿真波形

### 参考文献:

- [1] Li R, Zeng B. A New Three-step Search Algorithm for Block Motion Estimation[J]. IEEE Trans Circuit Syst Video Technol, 1994(4): 438-442.
- [2] Po L M, Ma W C. A Novel Four-step Search Algorithm for Block Motion Estimation[J]. IEEE Trans Circuit Syst Video Technol, 1996(6): 313-317.
- [3] Liu L K, Feng E. A block-based Gradient Descent Search Algorithm in Video Coding[J]. IEEE Trans Circuit Syst Video Technol, 1996(6): 419-423.
- [4] Zhu S, Ma K K. A New Diamond Search for Fast Block-matching Motion Estimation[J]. IEEE Trans Image Processing, 2000(9): 287-290.
- [5] Zhu C, Lin X. Hexagon-based Search Pattern for Fast Block Motion Estimation[J]. IEEE Trans Circuit Syst Video Technol, 2002(12): 349-355.
- [6] Vassiliadis S, Hakkennes E, Wong S, et al The Sum-Absolute-Difference Motion Estimation Accelerator[A]. In Proceedings 24th Euromicro Conference Volume 2[C]. Vasterass Sweden: IEE INSPEC Pub, 1998. 559-566.