基于TI DM642的一种H.264插值快速实现方法

陈晨航　卢官明

(南京邮电大学通信与信息工程学院,江苏 南京 210003)

摘　要:H.264视频编码标准正在得到越来越广泛的运用。在H.264中,插值和运动估计的计算复杂度之和大概占到整个视频编码的50%以上。因此,如何降低插值和运动估计的计算复杂度是快速实现H.264的关键所在。针对DM642的硬件架构,尝试了一种新的插值数据结构。经过优化,对于插值过程本身,以及到后来计算运动矢量时的数据缺失引起的CPU数据挂起,都降低到了比较理想的水平。

关键词:H.264插值;DM642;CPU数据挂起

中图分类号:TN922.11　　**文献标识码**:A

0　引言

近年来,随着TI的高性能定点处理能力的C6000系列芯片的推出以及H.264在引入整数量化和DCT之后,使C6000和H.264很自然地结合在一起。事实上,现在很多视频会议的终端设备供应商都已经或则正在实现基于TI DM642的H.264视频编解码。在视频会议上,一般使用的是H.264基本层(baseline),CIF及4:2:0的格式图像。在H.264中,运动搜索的1/4像素精度是其图像质量比以往H.26x系列视频编解码标准更好的重要因素,但同时也大大增加了实现的复杂度。首先是,参考图像的1/4插值像素点的实现。插值出来的像素总数是原来的16倍,数据量增加的同时,也造成了很大的CPU数据挂起。其次,造成计算运动矢量,运动搜索的时候,CPU的数据挂起(CPU DATA STALL)。所以,这两方面是在DM642上快速实现H.264的难点所在。本文尝试提出的新数据结构,经过测试序列的验证,在这两方面都收到比较理想的效果。

1　TI DM642的两级CACHE和DMA技术

在DM642的内核上,使用的是两级高速缓存(简写为L1,L2),如图2所示。L1又可以分为16KB的L1P和16K的L1D,256K的L2是SRAM和CACHE可配置的。而片外存的空间则通常要大的多。

L2负责与片外数据进行通讯,L1则与CPU直接相连。以读数据为例,CPU是按如下过程读到数据的。在CPU读取某一个地址空间的数据时(比如片外数据),先检查在寄存器里是否有该数据,若有,则直接读取。若没有,则到L1D里面查找,如果找到了,这就叫做一次L1D读命中,否则,则称为一次L1D读缺失。如果在L1D里找不到,则到L2里面查找。如果找到,则称为一次L2读命中,若不然,则称为一次L2读缺失。那么最后只能到外存读取数据了。所以,在DM642上,数据会缓存在两级CACHE里面,按照LRU策略进行更新。CPU在不同的地址空间获取数据所使用的周期数是大不相同的。表格1是CPU的读写各个地址空间的周期数。

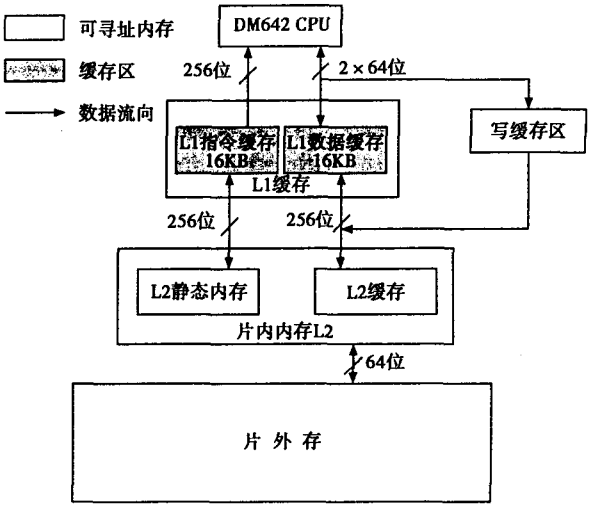

图1　DM642的两级缓存结构

表1　不同情况下的CPU读取数据的所用的周期数(单位:cycles)

寄存器	1
L1D读命中	1
L1D读缺失,L2D读命中	6~8
L1D读缺失,L2D读缺失(片外取数据)	200~300

由此可见,在CPU和外存之间频繁地读写数据,会造成极大的CPU拥塞。而在H.264的插值中,最后一共形成了16帧的数据,对于CIF格式的图像来说,亮度分量的大小是1.7 MB。当然,这么大量的数据不可能全部放置在片内。而在做运动估计的时候,需要在这16个参考帧里得到最佳匹配块。从而,需要CPU频繁地到外部存储器读取数据。虽然鉴于DM642的两级CACHE技术,不需要每次获取数据都到片外得到,但由这部分引起的CPU拥塞依然很大,对foreman序列来说,按照300帧的测试结果,大概是每帧6.8M cycles。因此,针对DM642的缓存策略,本文提出了一

收稿日期:2006-10-18　修回日期:2006-11-29

第一作者　陈晨航　男　24岁　硕士

种新的插值后的数据重新编排顺序,按照新的数据结构,可以减少CPU直接到外存取数据,从而减少了CPU DATA STALL。考虑到了DM642是64比特位(8个字节)可寻址的,新数据结构将两行同类型的8个像素(8个字节)编排在一起,这样做更利于实现。

在介绍新数据结构之前,先介绍一下DMA技术。利用DMA后台处理的特性,在CPU处理当前数据的时候,将下个循环CPU所要用到的数据搬移到片内,以使下一个循环中CPU直接L2 SRAM读命中。或者,在CPU处理当前循环的时候,将上一个循环的数据结果搬移到片外,这样就避免了CPU的回写数据拥塞。为此,我们需要在片内开辟称为PING-PONG缓存的SRAM。所谓PING-PONG缓存就是两块大小一样的内存,一块供DMA搬移使用,另一块供CPU使用,这样就可以避免数据的读写冲突,使CPU的写数据挂起减少到最小。在进行插值的过程中,我们可以使用DMA将当前循环中得到的像素值搬移到片外,而CPU则处理下个循环中的像素插值。

2　新数据结构

在JM的参考插值代码里,按照从左到右,从上到下的顺序,一共生成了16个独立的平面。一个平面代表了一个位置类型的像素集合。而在本文中,我们将整像素,1/2像素,1/4像素合在一起,将相邻两行的同类型像素合为一行。这样做的目的是在计算运动矢量,计算残差的时候,一个CACHE LINE包含了两行的数据,从而减少了近一半的CACHE MISS。

㊣表示整像素,▲,○,▼分别代表1/4像素点,横向插值1/2像素点,1/4像素点;◇,□,▽,△都是第二行的1/4像素点;●纵向插值1/2像素点,■第三行1/4像素点,◆交叉向插值1/2像素,★第三行3/4像素;☆,※,◎,⊙都是第四行1/4像素点。

新的数据编排流程如图2所示。其中1代表插值后同类型像素行的前一行,2代表后一行。省略号表示后面省去的像素点按照该两行编排的规律排列,直到这两行数据排列完毕为止。

1 ㊣㊣㊣㊣2㊣㊣㊣㊣ 1▲▲▲▲ 2▲▲▲▲ 1○○○○ 2○○○○ 1▲▲▲▲2▲▲▲▲……
1 ◇◇◇◇2◇◇◇◇ 1□□□□ 2□□□□ 1▽▽▽▽ 2▽▽▽▽ 1△△△△2△△△△……
1 ●●●●2●●●● 1■■■■ 2■■■■ 1◆◆◆◆ 2◆◆◆◆ 1★★★★2★★★★……
1 ☆☆☆☆2☆☆☆☆ 1※※※※ 2※※※※ 1◎◎◎◎ 2◎◎◎◎ 1⊙⊙⊙⊙2⊙⊙⊙⊙……

图2　新参考帧数据结构

3　实验结果

经过代码的优化和数据结构流程调整之后,在TI提供的集成开发环境CCS 3.1(Code Composer Studio)的软件仿真环境下,DM642主频为600MHz,外存频率133MHz。利用foreman,mobile.,paris作为测试序列,图像格式为CIF,额定码率768kps,帧频为25/秒,测试帧数均为300帧。测试出来的结果和JM参考代码的16个独立平面的C代码进行对比。结果如表2所示。

表2　测试结果对比

测试序列	Foreman	Mobile	paris	说明
优化后插值CPU数据挂起每帧	106K	106K	106K	单位均为cycles;K表示1000;M表示1000000
优化前插值CPU数据挂起每帧	5.5M	5.5M	5.5M	
优化后1/4插值总时钟每帧	1.58M	1.58M	1.58M	
优化前1/4插值总时钟每帧	9.60M	9.60M	9.60M	
优化后运动搜索残差CPU数据挂起每帧	3.66M	3.51M	3.34M	
优化前运动搜索残差CPU数据挂起每帧	6.83M	6.20M	6.15M	

4　结论

在表2中,在插值函数的优化上,新数据结构将总时钟周期将为原来的1/6,而在插值本身CPU的数据挂起上,更是降为到原来的2%。在计算运动搜索残差的CPU数据挂起上,三个序列差不到都降低到了原来的一半。可以看到,本文建议的新数据结构在三个测试序列在H.264视频编码标准两个实现关键点即1/4像素插值和运动估计上都取得明显的效果,为下一步的工作建立了良好的基础。

参考文献

[1] Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC), approved Output Document of JVT[Geneva 27 May 2003]

[2] TMS320C6000 DSP Cache User's Guide, Texas Instruments, spru656A,may 2003.

[3] 江思敏,刘畅.TMS320C6000 DSP应用开发教程[M].北京:机械工业出版社,2005.4.

A Fast Implementation Method of H.264 Interpolation Based on TI DM642

Chen Chen-hang　Lu Guan-ming

(*College of Communication and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing Jiangsu* 210003, *China*)

Abstract: The H.264/AVC video coding standard has being applied more and more recently. In H.264, the total calculation of interpolation and motion estimation consists 50% of all the coding. Therefore, these two aspects are key points of implementation of H.264 coding. According to the structure of TI's DM642, the paper proposes a new interpolated data structure. After optimizing, the process of interpolation and the CPU data stall aroused by data missing of motion estimation are both decreased to an acceptable level.

Key words: H.264 interpolation; DM642; CPU data stall