

H.264/AVC视频编码器的DSP实现

黄勇坚 王亚丽 王洪君

作为新一代的多媒体应用视频编码标准, H.264/AVC (以下简称H.264) 采用了许多不同于以往标准的先进技术, 在编码效率和性能大幅提高的同时, 增强了错误恢复及网络自适应等功能, 在广播电视、视频存储与回放、视频会议等领域具有广泛的应用前景。但H.264 编码性能的提高是与其计算复杂度的明显增加为代价的。在硬件资源有限的嵌入式环境下开发具有实时编码功能的视频编码器对于推动H.264标准的应用具有重要意义。

TMS320DM642 (以下简称DM642) 是TI公司开发的基于C64x内核的高性能数字多媒体处理器。本文介绍了H.264 “baseline” 编码器的DSP实现及优化方法, 源码采用H.264/AVC 源代码x264的编码部分。最终获得了一个复杂度较低、编码效率较高的嵌入式实时H.264编码器方案。

H.264编码标准特点

H.264视频编码标准吸收了以往编码方案的优点, 在视频压缩性能上获得了很大提高, 极大地降低了发送视频图像所需要的带宽, 能使编码的视频显示以灵活的方式应用于多样化的网络环境。与以往的编码标准相比, 其主要特点如下:

(1) 提高压缩编码效率。与H.263+ 或者MPEG-4的简单档次 (simple profile) 相比, 在相同视频图像质量情况下, H.264能够节省50%左右的码率。

(2) 高质量的视频图像。H.264在各种码率下能够提供稳定的视频质量, 特别是在低码率条件下, 也能够保持较高质量的视频图像。

(3) 错误恢复功能。H.264提供了包传输网中处理包丢失所需的工具和易误码的无线网中处理比特误码的工具, 确定了如何在包丢失和移动通信崩溃的情况下进行错误恢复, 有效避免先前的错误会持续影响后面的图像质量。

(4) 网络友好性。H.264标准中采用面向网络传输的结构和语法, 增强了标准的网络适应能力, 通过引入面向IP (internet protocol) 包交换的编码机制, 实现视频数据在网络中的分组传输, 对于不同的内容采用分割 (data partition) 编码传输, 使H.264的文件能够容易的在不同网络上传输。

TMS320DM642特点

TMS320DM642是美国德州仪器公司开发的第二代高性能超长指令字结构 (VLIW) 的定点DSP处理器, 具有8个独立的功能单元和

H.264/AVC与以往视频编码标准相比, 编码效率和编码性能大幅提高, 但计算复杂度也明显增加。H.264 “baseline” 编码器DSP移植及优化后, H.264编码器效率获得了很大提高, 达到实时编码的要求, 同时能获得较好的视频图像质量。

64个32位通用寄存器,在8个功能单元里扩展了专门用于视频/图像处理的指令集,提高了视频处理的性能和指令结构的并行性;在600MHz的时钟频率下,DM642的峰值处理速度达到4800MIPS(每秒百万条指令);DM642片内采用两级存储器结构,并具有丰富的片上外围接口,如10/100Mbit/s以太网接口、三个可配置的视频端口和一个64位的外部存储器接口等。DM642的强大处理和接口能力使它非常适合基于IP和无线网络的音视频传输、安全监控等视频/图像处理领域的应用。

H.264编码器的DSP优化

x264源码不能未经任何修改简单的移植到DM642开发平台上,因为:(1)原x264代码的测试平台为通用PC,代码主要用C语言编写,未结合DSP的结构和指令集特点,简单移植到DSP上不能运行;(2)x264中的编码算法未经优化,运算量巨大,无法达到系统实时的要求。因此,H.264编码器在DM642硬件开发平台上的具体实现,必须结合DSP的特点对其存储空间分配、数据传输策略和编码器核心算法进行优化,提高程序运行的并行性。

DSP存储空间分配优化

DM642片内采用两级存储器结构,其中L2存储器可以配置成S

RAM(可片内寻址),也可以配置成Cache/SRAM混合模式。由于片内存储器的存取速度远快于片外存储器,应尽量将程序、数据等各类信息存放在片内存储器。但H.264编码器对内存的需求相当大,要将所有的代码和数据都放入片内是不可能的。当然也可以利用DM642的两级Cache机制,将代码和数据都放到片外,但这样做的缺点:一是CPU每次请求新数据都要从片外存储器读入,这样会造成Cache频繁地读缺失而使CPU产生阻塞,运行效率降低;二是过多占用外部存储空间;三是没有充分发挥DM642的功能特点,不利于编码器性能的进一步提高。所以必须对片内存储空间进行优化,最终的优化方案为:

(1) L2设置为128k SRAM+128k Cache模式。

(2) 利用CCS编译器提供的指令语#pragma CODE_SECTION和DATA_SECTION将调用比较频繁的代码和数据指定在片内,将调用比较少的代码和数据放在片外。

(3) 栈区大小可以通过CCS中的View|Memory观测其空间占用情况而进行调整。

(4) 将重建帧(本编码器仅1帧参考帧)放在片内,而将原始的YUV视频数据放在片外,使用

EDMA进行搬移。

数据传输策略优化

在视频编码过程中,外部存储器和内部存储器间存在大量的数据传输,数据传输任务占据了CPU大量的时间。利用DM642的EDMA控制器与CPU并行工作的特点对数据传输策略进行优化,可以“隐藏”大部分的数据传输时间。

Ping-Pong缓冲是一种用于协调EDMA与CPU并行处理的技术。ping和pong是两个用于存放输入和输出数据流的缓冲区。当EDMA从ping缓冲区读入或搬出数据时,CPU对pong缓冲区中的数据进行处理。当两者均完成操作后交换缓冲区。通过这种方式,逐步完成大量数据的处理。Ping-Pong缓冲原理如图1所示。

利用Ping-Pong缓冲可以对H.264编码器的输入接口进行优化。将ping和pong两个缓冲区以1帧(QCIF格式)的数据量为单位进行设置,这样即减少了对片内存储空间的占用,片内和片外数据交换的频率又不至于过于频繁。

算法的DSP优化

本编码器主要对运动补偿中的亚像素内插算法进行了优化。H.264标准采用1/4像素精度的运动矢量。半像素点的亮度预测值由6抽头滤波器内插得到,而1/4像素

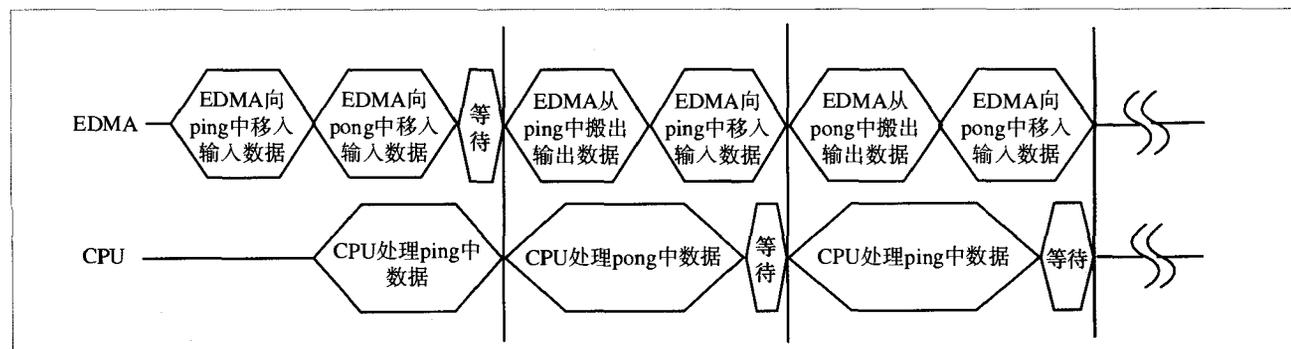


图1 Ping-Pong缓冲原理图

ELECTRONIC TECHNOLOGY 应用设计

点则是由整像素点和半像素点经过线性内插求得。采用6抽头滤波器进行半像素插值可以改善编码效果,但提高并不明显,而数据的读取和运算量却增加了很多,算法的高复杂性也不利于指令的并行执行。为了提高编码效率,在进行亚像素插值运算时可以使用双线性内插来代替二维6抽头滤波。双线性

8bit数Y1和Y2。

(2) 使用右移拼接指令_shrmb对Y1和Y2进行打包数据处理:Y1右移8位,再把Y2的最低字节续为其最高字节,形成新字Y3。

(3) 使用_avg4求出Y1和Y3对应的4个无符号字节的平均值。

同理,可以采用该算法快速计算出垂直半像素点的内插值(V)。

以进一步提高代码的运行效率。C6000系列DSP支持线性汇编语言,线性汇编语言简化了C6000汇编语言程序的开发。与标准汇编语言相比,采用线性汇编语言进行编程不用考虑并行指令的安排、指令延迟和寄存器的使用,以上工作均由汇编优化器自动完成,而且所产生的代码效率可以达到手工汇编效率的95%~100%。

性能测试

为了验证H.264编码器优化后的性能,我们选取了3个QCIF格式的标准视频序列进行了仿真测试,包括Foreman、Carphone和News序列。其中Foreman和Carphone序列具有相对大的运动幅度,而News序列运动幅度较小。取QP=24,编码60帧。测试结果如表1所列。表中列出了编码帧率和亮度分量平均峰值信噪比Y-PSNR的前后变化值。从表中可以看出,编码器可对序列进行实时编码,而Y-PSNR的损失不超过0.2dB。Y-PSNR的下降主要是因为亚像素插值使用了双线性内插,高频分量被滤除的缘故,但对图像的主观质量影响不大。

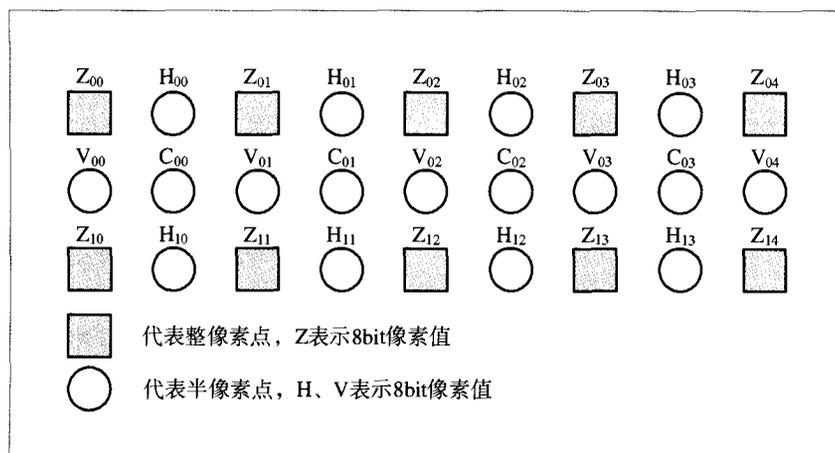


图2 双线性内插原理

内插原理如图2所示。

图中半像素点(如 H_{00} 、 H_{01} 、 V_{00} 、 V_{01})可以通过相邻两个整像素取平均值得到,如 H_{00} 、 V_{00} 点计算如下:

$$\begin{aligned} H_{00} &= (Z_{00} + Z_{01}) / 2 \\ V_{00} &= (Z_{00} + Z_{01}) / 2 \end{aligned} \quad (1)$$

C_{00} 点也可以通过整像素点 Z_{00} 、 Z_{01} 、 Z_{10} 、 Z_{11} 线性插值得到:

$$C_{00} = (Z_{00} + Z_{01} + Z_{10} + Z_{11}) / 4 \quad (2)$$

DM64x提供了丰富的用于视频/图像处理的扩展指令集,可以对双线性内插运算进行进一步的指令优化,提高代码运行的并行性,获得更高的编码效率。利用扩展指令集可以实现双线性内插的快速算法,以下给出了水平方向内插快速算法的具体步骤:

(1) 使用两个_amem4指令从存储器中分别读取2个连续的4x

对角线上的内插值(C)可以通过对垂直内插得到的像素值再进行一次水平内插得到。程序的DSP优化

DSP的程序优化技术包括C语言程序优化和汇编语言程序优化。C语言程序优化的主要方法

序列	耗时 (s)	帧率 (fps)	ΔY -PSNR/dB
Foreman	1.94	30.99	0.17
Carphone	1.84	32.69	0.03
News	1.50	39.92	0.12

表1 优化后的编码器性能测试

有:尽可能的使用C6000编译器提供的内联函数、数据打包处理技术和软件流水技术等。对调用比较频繁的函数可以采用汇编语言进行重写,如运动估计中的匹配误差函数SAD和整数变换中关键函数sub4x4_dct和add4x4_idct等,可

本文介绍了H.264视频编码器在存储空间分配、数据传输、半像素插值和程序等方面的优化方法。优化后的编码器编码效率获得了很大提高,对于QCIF格式的视频序列每秒钟能够处理30~40帧,同时获得了较好的视频图像质量。□