文章编号:1002-8692(2008)09-0021-03

H.264/AVC 编解码器在 TMS320C6400 上的优化实现

·实用设计·

彭 晔，韩 峥，唐 昆，崔慧娟
（清华大学 微波与数字通信技术国家重点实验室，北京 100084）

【摘 要】分析了基于 TMS320C6400 平台的 H.264/AVC 编码器运算量在各个部分的分布，并且将其中运算量较大的部分进行线性汇编，分析了这些部分在进行线性汇编时需要注意的技术细节，能够提高线性汇编的并行程度。

【关键词】TMS320C6400；H.264/AVC 标准；线性汇编
【中图分类号】TN918.81 【文献标识码】A

Optimization of H.264/AVC Codec Based on TMS320C6400

PENG Ye, HAN Zheng, TANG Kun, CUI Hui-juan
(*State Key Laboratory of Microwave and Digital Communications, Tsinghua University, Beijing 100084, China*)

【Abstract】 Data computing with the standard of H.264/AVC codec based on TMS320C6400 is analysed in this paper, and the C codec is changed into linear assembly language where there is a large amount of computing. And the technical details about the linear assembly language is discussed, some methods to improve the parallel processing is presented.

【Key words】 TMS320C6400; H.264/AVC; linear assembly language

1 引言

H.264/AVC[1]标准是由国际标准化组织及国际电工委员会(ISO/IEC)下属的运动图像专家组(MPEG)和国际电信联盟(ITU)下属的视频编码专家组(VCEG)联合的视频联合工作组(JVT)共同制定的。作为最新一代的编码技术，H.264/AVC 分为网络提取层(NAL)和视频编码层(VCL)，使得其有很好的网络适应性。1982 年，TI 公司推出了 TMS320 系列的第一款 DSP 芯片 TMS32010。至今，TMS320 已经在许多领域得到了应用。其中，由于 TMS3200C6000 系列的 DSP 具有高效的 C 编译器，使得其对于不同的产品具有广泛的应用[2]。

2 TMS320C6400 简介

TMS3200C6000 系列 DSP 的中央处理单元 CPU 的结构有：1）程序取指单元；2）指令分配单元，高级指令打包(仅 C64)；3）指令译码单元；4）2 个数据通路，每个有 4 个功能单元；5）32 个(C64x 有 64 个)32 位寄存器；6）控制寄存器；7）控制逻辑、测试、仿真和中断逻辑。

C6400 的数据通路有两个通用寄存器组 A 和 B。在 C64x 中，A 和 B 分别有 32 个寄存器 A0~A31，B0~B31，每个通用寄存器都是 32 位。

C6400 的数据通路中有两组功能单元，每组有 4 个，共 8 个功能单元，分别是.L1，.S1，.D1，.M1，.L2，.S2，.D2，.M2 单元，其中，数字为 1 的功能单元对 A 寄存器组进行操作，数字为 2 的功能单元对 B 寄存器组进行操作。L，S，D，M 是可以进行不同操作的功能单元。在数据通路中，还有一对交叉通路 A 和 B，A 通路通过 1X 可以访问 B 寄存器组的数据，同样，B 通路可以通过 2X 交叉通路访问 A 寄存器组。因此，在每个时钟周期中，DSP 最多只能完成两次访问对方寄存器组的操作。

在 C6400 中，允许用户把程序写成线性汇编来对程序进行优化。在线性汇编中，不必要制定在常规 C6400 汇编代码中必须制定的全部信息[3]。用户可以用编译器制定以下信息：1）并行指令；2）潜在的流水；3）寄存器的使用；4）功能单元的指定。

使用 DSP 的编译器 CCS 的 profile 功能，可以对编码器的各部分的运算量进行测试，编码器的编码选项有：1）I P P P P…..；2）无宏块提前退出；3）参考帧数：1；4）QP：I 帧 28，P 帧 28；5）P 帧中存在帧内预测；6）格式：cif。

综合各个部分的分析，可以看出，在编码过程中，运算量比较大的部分有 SATD 的计算、SAD 的计算、DCT 变换、量化、反量化等部分。测试结果如表 1 的 A 列所示。

3 各部分分析、优化方法及测试结果

3.1 SATD，SAD 和 DCT 等部分的线性汇编优化

这部分的线性汇编优化中，有下面几点需要注意：

1）C6400 中，DSP 对函数的调用方式如下：对于一

个函数(参数 a,参数 b,…)来说,参数 a 存放在寄存器 A4,参数 b 存放在 B4,如此类推分别是 A4,B,A6,B6,A8,B8,…,其中,如果参数 a 的位数大于 32 位,那么将使用 A5:A4 进行这一个参数的存放。

2) SATD 的计算、SAD 的计算和 DCT 变换主要是进行两个像素块之间的运算,所以本质上这些计算是处理两个同样大小的数组。在原编码器中的 C 语言函数通常不会考虑接口参数顺序,有可能使得在程序编译以后并行执行的效果很差,导致程序并行程度不高的结果,所以为了节省运算时间,需要重新排列参数顺序,比如 sub4 ×4 _dct (int16_t dct [4] [4],uint8_t*pix1,int i_pix1, uint8_t*pix2,int i_pix2)这个函数的接口中的参数 dct[4][4] 的首地址存放在寄存器 A4 中,*pix1 的地址存放在寄存器 B4 中,i_pix1 的值存放在寄存器 A6 中,*pix2 的地址存放在寄存器 B6 中,i_pix2 的值存放在寄存器 A8 中。由于在 B 数据通路单元只有一个交叉通路,所以在读入 pix1 和 pix2 数据的时候只能依次进行,无法并行进行,故可以将接口改成 sub4 ×4_dct (uint8_t*pix1, uint8_t*pix2,int i_pix1,int i_pix2,int16_t dct [4][4]),这样函数的接口中的参数 *pix1 的地址存放在寄存器 A4 中,*pix2 的地址存放在寄存器 B4 中,i_pix1 的值存放在寄存器 A6 中,i_pix2 的值存放在寄存器 B6 中,dct [4][4] 的首地址存放在寄存器 A8 中,这样在读数据的时候,就可以使得 A 和 B 数据通路单元分别进行自己的操作,提高了并行度。

3) 在 TMS3200C6400 的芯片上,由于数据读入的原则是数据读入后的第 4 个时钟周期才可以使用,所以,在并行处理数据的时候,尽量不要处理近 3 个周期内读入的数据,如果强制进行的话,编译器会自动在处理数据之前加入若干次 nop 的操作,以符合其芯片的处理规则。这样,就将会浪费 3 个时钟周期。

同时,在读入数据以后,尽量展开一些内层的循环,这样可以有效地进行并行处理,节约时钟周期,比如对一个 4×4 的块进行操作,如果在 C 语言的函数中,可能会进行一个时间复杂度为 n^2 的操作。由于在 C6400 的线性汇编中,一共有 64 个寄存器可用,所以可以把循环展开,尽量充分使用寄存器,以达到更好的并行处理的目的。

4) 在寄存器中,由于只有一对数据交叉通路,所以将循环展开把数据读入寄存器中时,需要根据后续对数据处理的操作来安排存放于哪个寄存器组之中,使得尽可能减少数据交叉通路的使用。因为在 SATD,SAD,DCT 中,后续的对数据处理的操作主要是加、减、乘和移位,在芯片的.M,.S,.L,.D 功能单元中,.S,.L,.D 单元皆可以进行加减的操作,所以如果使用合理的话,其时钟周期可以达到以前的 1/6。

在这些部分中,由于有时需要进行乘 2 的操作,所以,可以并行使用.M 单元进行乘 2 操作,.S 单元进行 1 位位移操作,.L 和.D 单元进行自相加操作,都可达到乘 2 操作的目的。如果仅仅使用.M 单元进行乘 2 操作,只能使这一段的时钟周期提高为原来的 1/2。但是如果将位移和相加的操作同时使用,并行度将会大大提高,使得时钟周期变为原来的 1/8。

综上所述,如果不计算函数接口处的跳出跳入操作所耗费的时钟周期,线性汇编后的程序所耗费的时钟周期将会在原时钟周期的 1/8~1/6。

其中 SATD,SAD,DCT 在线性汇编后,所占的比重大大减小,其中 SATD 从 37%减到 8%,SAD 从 14%减到 2%,DCT 从 11%减到 2%,具体数据见表 1 的 B~D 列。

3.2 量化、反量化部分的线性汇编优化

尽管 H.264/AVC 在 DCT 的变换以及量化和反量化中,采取了整型变换的措施,使得量化和反量化中的除法化为乘法和位移,大大节省了数据处理的时间。但是在运算过程中还是存在少量的除法。在 H.264/AVC 的标准中,量化的 QP 值每增大 6,量化值变为原值的 2 倍,所以这里出现了需要对量化 QP 值除 6 并且取余数的操作,由于在量化和反量化中的除数仅仅是一两个个别的数字(3 或者 6),所以在此可以将除法转化为乘法,先乘以一个确定的整数,然后再进行位移。这样仅仅 2 个时钟周期就可以完成这一除法。

同时,对于求余运算,可以使用上述方法求得的商来求取余数,也仅仅是一个相乘后再相减的过程,2 个时钟周期就可以完成。由于原 C 程序中存在了除法的运算,所以线性汇编完成之后的函数所消耗的时钟周期是原函数时钟周期的 1/10 左右。

在解码器端,反量化的优化与编码器端相同。表 1 的 E 列和 F 列即为量化反量化后的结果。可以看出来,量化部分从优化前的 10%的比重降到优化后的 4%,反量化部分从优化前的 3%降到优化前的 1%。

4 实验结果

对于 H.264/AVC 标准的编码过程中,虽然有些部分运算量占有一定的比重,但是由于其主要的运算消耗在判断、选择以及调用别的函数等方面,所以线性汇编后的结果不能产生太大的效果,并且考虑到代码长度、调用次数等各方面的原因,所以其线性汇编的价值就显得更小了,完全可以直接使用 C 代码。

经过测试,DCT 部分的函数时钟周期降为原来的 1/6 左右;SATD 部分的时钟周期降为原来的 1/7 左右;SAD 部分的时钟周期降为原来的 1/7 左右;量化部分的时钟周期降为原来的 1/14~1/12;反量化部分的时钟周期降为

原来的 1/7~1/4。

将所有线性汇编过后的函数加入原编码器，其运算量分部如表 1 的 G 列所示。在编码器的速度方面，其编码的速度可以达到其优化前的 2~4 倍。

表 1 编码器运算量分布

编码步骤	不同情况所占比例/%						
	A	B	C	D	E	F	G
编码器初始化	4	6	5	5	5	4	11
量化	10	15	12	11	4	11	8
运动估计运动补偿	7	11	8	8	9	7	19
帧内 SATD 计算	37	8	40	40	43	37	14
帧内模式选择	7	10	8	8	8	7	18
反量化	2	3	3	2	3	1	2
帧间运动矢量计算	0	0	0	0	0	0	0
插值	8	11	9	8	9	8	19
DCT变换	11	16	13	2	2	11	5
宏块/pps/sps/nal/编码	0	0	0	0	0	0	0
帧间 SAD 计算	14	20	2	16	17	14	4
CABAC	0	0	0	0	0	0	0

5 小结

由于 H.264/AVC 标准以其大运算量来换取高压缩比，所以对于 TMS320C6400 平台上的 H.264/AVC 的编解码器而言，使用线性汇编优化其大运算量的部分，减少运算时钟周期是十分必要的。本文的指导思想和方法可以应用于基于 TMS320C6400 平台的其他领域的开发。

参考文献

[1] ITU-T Rec H.264/ISO/IEC14496-10 AVC (Doc JVT -G050), Draft ITU-T recommendation and final draft international standard of joint video specification[S].2003.

[2] Texas Instruments Incorporaterd. TMS320C6000 系列 DSP 的 CPU 与外设[M]. 北京：清华大学了版社，2007.

[3] Texas Instruments Incorporaterd.TMS320C6000 系列 DSP 编程工具与指南[M]. 北京：清华大学了版社，2006.

作者简介：

彭 晔(1983-)，硕士生，研究方向为视频压缩编码；

韩 峥(1981-)，博士生，研究方向为视频压缩编码；

唐 昆，教授，从事多媒体通信方面的研究；

崔慧娟，女，教授，从事多媒体通信方面的研究。

责任编辑：任健男 收稿日期：2008-07-11

(上接第 20 页)

到 8×8 寄存器组 0，块 5 的数据输出到 8×8 寄存器组 1。

3）对边界 3 进行滤波，块 4 的数据由 8×8 寄存器组 0 转置后输入滤波模块，块 1 的数据由 3×8 寄存器组 0 转置后输入滤波模块，滤波后块 1 的数据经 8×3 寄存器组转置输出到 FIFO 存储器，块 4 的数据输出到 8×8 寄存器组 0 转置后，前 5 行输出到 FIFO 存储器，后 3 行输出到 3×8 寄存器组 0。然后对边界 4 进行滤波，块 5 的数据由 8×8 寄存器组 1 转置输入滤波模块，块 2 的数据由 3×8 寄存器组 1 转置后输入滤波模块，滤波后块 2 的数据经 8×3 寄存器组转置输出到 FIFO 存储器，块 5 的数据输出到 8×8 寄存器组 1 转置后，前 5 行输出到 FIFO 存储器，后 3 行输出到 3×8 寄存器组 1。

4）对边界 5，6 进行滤波，与上面处理过程相同。

5）对边界 7，8 进行滤波，与上面处理过程相同。

6）对色度边界进行滤波，与亮度的处理过程相同。

4 实验结果

本文对上述方法在 Xilinx Vertex-4 LX200 进行了仿真和测试，采用 Synplify Pro 8.6.2 进行综合和采用 ModelSim SE 6.3a 进行仿真。假设 AVS-P2 硬件编码系统的工作频率为 100 MHz，要实现高清 1 920×1 088@30 f/s 实时编码，每个宏块的处理时钟数的上限为 408 个时钟。实验结果显示本文硬件滤波器最高工作频率可达 150 MHz，整个滤波器消耗约 37 000 逻辑门，完成一个宏块滤波需要 289 个时钟。采用多个 1 080i 视频序列进行测试，仿真结果与 AVS 参考模型的计算结果一致。

5 小结

本文提供一种 AVS 环路滤波的硬件实现方法，采用垂直滤波和水平滤波交叉进行以及利用寄存器组进行中间数据的暂存与转置，有效减少片内存储器的开销和片外存储器的访问，可实现快速滤波，满足实时高清编码的要求，可应用于 AVS 高清实时编解码的芯片设计。

参考文献

[1] GB/T 20090.2-2006，信息技术先进音视频编码第 2 部分：视频[S]. 2006.

[2]虞露，胡倩，易峰. AVS 视频的技术特征[J].电视技术，2005(7):8-11.

[3]梁凡. AVS 视频标准的技术特点[J].电视技术，2005(7):12-15.

[4] SHENG Bin, GAO Wen ,WU Di . A platform-based architecture of loop-filter for AVS [J].2004 7th International Conference on Signal Processing, 2004,1:571-574.

[5] CHEN Tung-chien, CHIEN Shao-yi, HUANG Yu-wen, et al.Analysis and architecture design of a HDTV720p 30 frames/s H.264/AVC encoder [J]. IEEE Trans. Circuits and Systems for Video Technology, 2006,16(6):673-688.

作者简介：

刘锦阳(1980-)，硕士生，主研视频压缩编码和 EDA 技术。

责任编辑：任健男 收稿日期：2008-07-22