

文章编号:1000-582X(2008)04-0432-04

图像压缩编码在 DSP 上的实现与优化

曾 浩,高秀娟,谭晓衡,冯文江

(重庆大学 通信工程学院,重庆 400030)

摘 要:利用嵌入式系统完成 MPEG-4 图像编码时,由于编码运算复杂度高,而硬件运算速度又有限制,因而实现实时编码具有较大困难。通过分析 MPEG-4 图像编码器的运算量分布情况,结合 TMS320C6416 的结构特点,可以综合使用程序结构优化、应用层优化、代码优化的方法,把开源的 XVID 的 C 代码移植到 C6416 的过程中,实现程序整体优化。该方法相对于传统的单一优化,能更好提高程序的并行性和存储器的访问效率,从而在基于 DSP(digital signal processor)的嵌入式系统中实现了 MPEG-4 图像实时编码。

关键词:图像;编码器;移植;优化

中图分类号:TN919

文献标志码:A

Creation and optimization of image compression encoder by digital signal processor

ZENG Hao, GAO Xiu-juan, TAN Xiao-heng, FENG Wen-jiang

(College of Communication Engineering, Chongqing University, Chongqing 400030, P. R. China)

Abstract: When using an embedded system to complete MPEG-4 encoder, there are many difficulties in encoding in real time due to encoding algorithm complexity and hardware speed limit. Many optimization technologies were proposed in MPEG-4 encoder when transplanting C program of open source XVID to TMS320C6416. These include program structure optimization, application layer optimization, and code optimization. These technologies were based on the structure of an MPEG-4 encoder and TMS320C6416 characteristics. The optimization could improve program speed in parallel operation and effectual memory access. Experimental results indicate the optimized encoder can operate well in real time.

Key words: image; encoder; transplants; optimization

MPEG-4 编码器是由 ITU-T 的视频编码专家组(VCEG)及 ISO/IEC 的移动图像专家组(MPEG)大力发展研究的新一代基于对象的视频压缩标准^[1]。目前基于 PC 平台的 MPEG-4 视频编码器在互联网的多媒体业务方面已有较多的应用,但运用于数码摄像机、无线 3G 视频业务等嵌入式系统时,由于硬件资源的限制,需要进行各种优化才能满足

实时性的要求^[2-4]。

1 C6416 的主要结构特性

TMS320C6416 数字信号处理器,其主频 500 MHz,采用超长指令字(VLIW)结构。拥有 2 个各自独立的数据通道,64 个 32 bit 字长的通用寄存器,8 个独立功能单元,6 个算术运算单元 ALU。

收稿日期:2007-12-10

基金项目:国家自然科学基金资助项目(60672097);重庆大学自然科学基金资助项目(2007CD056)

作者简介:曾浩(1977-),男,重庆大学博士,主要从事通信、图像信号处理研究。

谭晓衡(联系人),男,重庆大学副教授,(E-mail)txh@ccee.cqu.edu.cn。
欢迎访问重庆大学期刊网 <http://qks.cqu.edu.cn>

支持每时钟周期单 32 bit、双 16 bit 或者 4 个 8 bit 算术运算。2 个乘法器支持每时钟周期 8 个 16×16 bit 乘法运算,或者 8 个 8×8 bit 乘法运算。采用 L1/L2 两级高速缓存结构,128 kbit 程序高速缓存 L1P、128 kbit 数据高速缓存 L1D、8 Mbit 内存 L2。

另外,TMS3206416 在完全兼容 C6000 软件资源的基础上,扩展了 88 条多媒体处理指令,这些指令能够更方便快速地执行图像处理中的算法。C6416 强大的包处理能力可同时处理 2 个 16 位数据或 4 个 8 位数据,并且还提供了 STDW、LDDW 等双字存取指令。这些特性和它的高速处理能力使其非常适合用于实时实现 MPEG-4 编码器。

2 MPEG-4 编码器的移植

2.1 算法简介

基于 PC 平台的 MPEG-4 标准软件主要有 DIVX 和 XVID,采用了 XVID1.1.0 版本作为实现基础。MPEG-4 编码器算法结构主要框架是:通过 main 函数进入编码程序,然后调用 xvid_encore() 进行每一帧的编码,最后通过调用 MBCoding() 实现每个宏块编码,从而形成码流。MPEG-4 标准规定了符合这个标准的码流应该是按什么形式组织,而且这个码流中很多地方是可选的,如一个码流可以支持 $1/4$ 像素运动补偿,也可以不支持。所以编码器可以是符合当前标准码流的一个子集,而算法也是开放的,因此具有很大的灵活性和优越性。

2.2 算法的移植

算法移植就是将在 PC 机上的程序移植到 DSP (digital signal processor) 平台上,并针对 DSP 特点及具体应用而进行改写,以使其能达到实时要求。一般移植,主要需完成两个工作。

第一,消减冗余代码。XVID1.1.0 参考版本为了以后扩展需要,留了很多接口,也包含了很多冗余代码,如大量 trace 信息、assert 信息和 printf 函数,这些信息在 DSP 终端设备上对实现 MPEG-4 算法没有任何必要。另外,参考软件中通过读取文件来实现系统参数的配置非常耗时,不利于编码器的实时实现,所以预先先把视频文件放入片外 SDRAM 中,编码时使用独立于 CPU 工作的 EDMA 进行数据搬移。同时,SNR 计算为浮点运算,在定点 DSP 上实现也很费时,所以计算信噪比 SNR 等工作也没有必要由 DSP 来同步完成,可通过后继计算得到。最后,程序中存在很多没必要的判断跳转,这些判断既影响了程序并行化的执行,又浪费了 DSP 有限的片内空间,在移植代码时削减了这些不必要的判断,大大减少了代码空间,提高了程序运行速度。

第二,重新编写原程序的头文件。由于 C6416 DSP 的 CCS 代码开发环境具有比 VC 更为严格的编译环境,所以如果将在 PC 机上运行的程序简单照搬到 DSP 上是无法编译的,如在 VC 下的 int64 在 DSP 中应使用 long 代替。此外 CCS 中不需要像 VC 那样添加头文件,系统会自动添加。

3 MPEG-4 编码器的优化

把 MPEG-4 编码器移植到 DSP 端,从而实现实时编码的要求,优化工作是十分繁重的。一般的优化顺序是先做 DSP 端的程序结构优化,再做 MPEG-4 算法级优化,然后是应用层优化,最后做代码级优化,该过程如图 1 所示。算法级优化中快速算法的实现会降低图像质量,而底层的优化工作不会对图像质量产生影响。

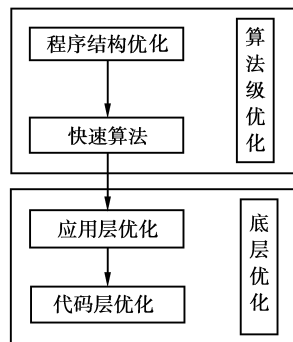


图 1 编码器优化流程

3.1 应用层优化

3.1.1 编译器优化

在 DSP 开发环境 CCS 下设置编译选项,利用编译器对程序进行自动优化。

-o3 选项表示可得到最高程度的优化。编译器将执行各种优化循环的方法,比如软件流水、循环展开和 SIMD(单指令多数据流)等技术。

-pm 选项表示程序级优化,编译器在编译时可以从整个程序的角度来优化程序。

-mt 选项编译器说明代码中没有使用混迭技术。对于汇编优化器,明确告知程序中不存在存储器相关性,即允许编译器在无存储器相关性假设下优化。

在使用 -o3 选项进行优化编译时,尽量联合使用 -pm 选项。实际测试表明,-o3 选项的打开对程序的优化相当显著。如果程序满足无存储器相关性的假设,打开 -pm 和 -mt 选项可以使编码速度提高 3~4 倍。

3.1.2 Cache 优化策略

在优化过程中,需要合理配置 L2 的 Cache 和片内 SRAM 的大小。配置的总原则是将尽量多的关键

数据分配在片内,Cache 越大越好。对于不同的应用需要用不同的配置。最优配置需要在开发中根据经验和实际的测试结果进行选择,同时需要合理组织缓冲区数据,减少内存访问冲突。实际测试表明把 L2 的 256 kB 配置成 Cache,剩余的 768 kB 作为 ISRAM 使用。L2 缓存设置时使用 CSL 中的 Cache 动态配置或使用 BIOS 静态配置。动态配置如下:

```
CSL_init();
CACHE_clean(CACHE_L2ALL, 0, 0);
CACHE_setL2Mode(CACHE_256KCACHE);
CACHE_enableCaching(CACHE_EMIFA_CE00);
```

3.1.3 使用 DMA 技术

利用 DMA 技术完成在片外 SDRAM、片内 ISRAM 和 Cache 之间转移数据的任务和数据重排等功能,提高数据传输的效率和内部存储空间的利用效率。DMA 不需要 CPU 的介入,这样可以节省 CPU 取数据所花费的时间。使用了 TI 的 CSL 支持功能,它有专门的 DMA 模块,便于对 DMA 的各个存储器控制,还有一个 DAT 模块,使用 DMA 进行存储器间的数据传输。使用函数 DAT_copy()和 DAT_fill()就象常用的内存操作 memcpy()、memset()一样,只需要在 API 接口指出源地址、目的地址和长度,或其他的维数属性等即可,而不需要再去管理具体的寄存器^[5]。下面的代码就是从 SDRAM 中视频序列(格式 144 × 176 的 QCIF)中,利用 EDMA 搬移一帧数据的例子。

```
DAT_open(DAT_CHAANY, DAT_PRI_HIGH, DAT_OPEN_2D); id = DAT_copy(rawbuf1, in_buffer, 38016);
```

```
DAT_wait(id);
```

3.2 代码级优化

代码级优化分为 C 语言级和汇编语言级,这里主要讨论汇编语言级的程序优化方案。通过使用 CCS 特有的 profile clock 工具,对 MPEG-4 算法各部分耗时情况进行测试,测试结果如图 2 所示。从中可看出运动估计补偿,量化和 DCT 模块耗时较多,因此对其进行重点优化。

3.2.1 使用图像库

TI 提供了基于 C64x 的图像库 IMGLIB,库中都是图像处理常用的函数,而且他们都是 C 可调用的,汇编优化好的,执行效率非常高,因此尽量用库函数^[6]。只是具体形式需要根据实际情况进行改动。为了避免替换漏项,应该直接在 fdct.c 文件中进行改写,把里面的内容去掉,接写入库函数 IMG_fdct_8x8((short * const)dct_data,1)。另外,XVID

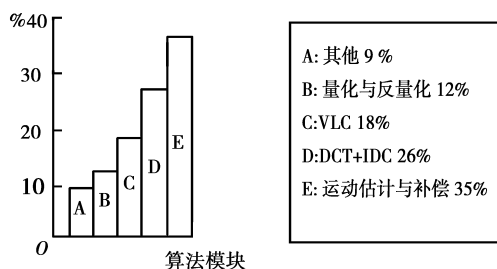


图 2 耗时测试结果图

代码中函数使用了函数指针,也可利用其进行替换。实验结果表明,替换后函数执行比替换前的效率提高了 20 倍。

3.2.2 改写成线性汇编或汇编

线性汇编语言是为了简化 C6000 汇编语言程序而开发设计的。与标准汇编语言相比,采用线性汇编语言进行编程不需要考虑并行指令的安排、指令延迟和寄存器的使用^[7]。上述工作由汇编优化器自动完成,产生的代码效率可以达到人工编写代码效率的 95%~100%^[8]。如果编写线性汇编语言仍不能满足性能要求,可再运用纯汇编语言编写相关的代码。这时需要充分考虑 C6000DSP 结构,尽可能并行其中的非相关语句,从而进一步减少代码的执行时间和提高程序的性能。针对那些耗时却没法用库函数代替的函数,应改用这种方式进行处理。以 SAD16 为例,对其使用线性汇编进行改写,下面是改写为线性汇编的部分代码。

```
Loop:
LDDW    *cur++[cur_stride],    c_d2:c_d1
LDDW    *ref++[ref_stride],    r_d2:r_d1
SUBABS4 c_d1,    r_d1,          tmp1
SUBABS4 c_d2,    r_d2,          tmp2
DOTPU4  tmp1,    four_1,        sad1
DOTPU4  tmp2,    four_1,        sad2
ADD     sad1,    sad,            sad
ADD     sad2,    sad,            sad
[ctr] SUB ctr,    1,             ctr
[ctr] B    Loop
```

从以上线性汇编画出相关图可知,迭代间隔至少是 2 个时钟周期,就可以用 4 个 LDDW 指令,每个周期使用 2 个 LDDW 指令,这样循环次数为 16 次。再考虑到交叉通路的影响,由于 1 个时钟周期内 A 侧和 B 侧的功能单元只能交叉 1 次,所以使用 4 个 LDDW 指令时,循环迭代间隔 2 已经不够用了,此时循环迭代间隔应为 3。通过手动汇编可得模迭代间隔表(见表 1),汇编程序可参考此表进行编写。

表 1 模迭代间隔表

单元名称	周期编号				
	0	3	6	9	12
.D1	LDDW (c1)	LDDW (c1)	LDDW (c1)	LDDW (c1)	LDDW (c1)
.D2	LDDW (r1)	LDDW (r1)	LDDW (r1)	LDDW (r1)	LDDW (r1)
.M1			DOTPU4 (sad_1)	DOTPU4 (sad_1)	DOTPU4 (sad_1)
.M2			DOTPU4 (sad_2)	DOTPU4 (sad_2)	DOTPU4 (sad_2)
.S2			B	B	B
.L1X			SUBABS4 (d_d3)	SUBABS4 (d_d3)	SUBABS4 (d_d3)
.L2X			SUBABS4 (d_d4)	SUBABS4 (d_d4)	SUBABS4 (d_d4)

单元名称	周期编号				
	1	4	7	10	13
.D1	LDDW (c2)	LDDW (c2)	LDDW (c2)	LDDW (c2)	LDDW (c2)
.D2	LDDW (r2)	LDDW (r2)	LDDW (r2)	LDDW (r2)	LDDW (r2)
.M1			DOTPU4 (sad_3)	DOTPU4 (sad_3)	DOTPU4 (sad_3)
.M2			DOTPU4 (sad_4)	DOTPU4 (sad_4)	DOTPU4 (sad_4)
.S1X				ADD (sad_2)	ADD (sad_2)
.S2X					ADD (sad_3)

单元名称	周期编号				
	2	5	8	11	14
.D1				ADD (sad_1)	ADD (sad_1)
.D2		SUB (cntr)	SUB (cntr)	SUB (cntr)	SUB (cntr)
.L1X		SUBABS4 (d_d1)	SUBABS4 (d_d1)	SUBABS4 (d_d1)	SUBABS4 (d_d1)
.L2X		SUBABS4 (d_d2)	SUBABS4 (d_d2)	SUBABS4 (d_d2)	SUBABS4 (d_d2)
.S2				ADD (sad_4)	ADD (sad_4)

表 2 是各种方式下的耗时对比表。由此可以看到执行时间的改善效果。表中的时间是查看 profile 中的 cycle cpu 得到的,不包括程序调度,数据等待等。也可以使用 cycle total 来查看综合效果,但有时 cycle cpu 是几十,但是 cycle total 可能是几百甚至上千。这种情况主要是 CPU 要到片外存储器读取数据或代码,因此把能放到调用最频繁的程序和常用的数据放到片上。具体实现分 C 环境和汇编环境。在汇编或线性汇编中做如下定义:

数据段: .usect “section name”,size in bytes
 程序段: .sect “section name”
 在 C 程序中定义如下:

数据段: # pragma DATA_SECTION(symbol, “section name”);
 程序段: # pragma CODE_SECTION(symbol, “section name”);
 使用上面的两条指令,就可以定义用户的程序段和数据段。这些定义完成后,需要在连接命令文件中,用 SECTIONS 伪指令指定他们在存储器上的存放位置。

表 2 函数周期数对比表

程序	周期数	
	不使用 -o3	使用 -o3
原 C 代码	5 486	360
线性汇编代码	896	98
汇编代码	60	60

4 实验结果

采用目标帧率 256 kbs/s,2 个相邻的关键帧之间参数设置为 10 帧,输入序列为 QCIF 格式(176×144),标准序列大小为 300 帧作为视频源,实验结果如表 3。

表 3 3 种标准测试序列测试性能

序列名称	平均 PSNR /db	总编码长/B	平均每帧长度 /B	每帧平均消耗时间 /ms	帧率/帧·s ⁻¹	压缩比
Container	26.20	418 391	1 399	36.37	26.95	27.26
Hall	25.78	428 017	1 426	390.30	25.48	26.65
Coastguard	17.34	457 749	1 525	54.78	24.16	24.91

从表 3 可以看出,如果综合采用一系列优化方式,效果十分明显,该编码器在保持了很高的图像质量和较低的码率的同时,完全可以实现 QCIF 格式图像在嵌入式系统中的实时性编码需求。

5 结 语

针对 TMS320C6416 的嵌入式图像处理系统,充分利用 DSP 适用于快速实现数字信号处理算法的结构优点,综合应用程序结构优化、应用层优化、代码优化的方法,克服了单一优化的不足,提高了系统运算速度,从而在嵌入式系统中,实现了实时编码。

- Cologne, Germany. New York: ACM, 2005: 31-42.
- [4] CHEBROLU K, RAMAN B, SEN S. Long-distance 802.11b links: performance measurements and experience [C]//Proceedings of the 12th Annual International Conference on Mobile Computing and Networking, Los Angeles, California, USA. New York: ACM, 2006: 74-85.
- [5] ALICHERRY M, BHATIA R, LI L. Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks [C]//Proceedings of the 11th Annual International Conference on Mobile Computing and Networking, Cologne, Germany. New York: ACM, 2005: 58-72.
- [6] KODIALAM M, NANDAGOPAL T. Characterizing the capacity region in multi-radio multi-channel wireless mesh networks[C]//Proceedings of the 11th Annual International Conference on Mobile Computing and Networking, Cologne, Germany. New York: ACM, 2005: 73-87.
- [7] ADYA A, BAHL P, PADHYE J, et al. A multi-radio unification protocol for IEEE 802.11 wireless networks, MSR-TR-2003-44 [R]. Washington: Microsoft Research, 2003: 1-13.
- [8] DRAVES R, PADHYE J, ZILL B. Routing in multi-radio, multi-hop wireless mesh networks[C]//Proceedings of the 10th Annual International Conference on Mobile Computing and Networking, Philadelphia, Pennsylvania, USA. New York: ACM, 2005: 114-128.
- [9] PIRZADA A A, PORTMANN M, INDULSKA J. Evaluation of multi-radio extensions to AODV for wireless mesh network [C]//Proceedings of the International Workshop on Mobility Management and Wireless Access. New York: ACM, 2006: 45-51.
- [10] HU Y, YANG S B, WANG D P. SMETT: a new routing metric for multi-radio and multi-channel wireless mesh network[C]//IEEE Communications Society, the 2nd IEEE International Conference on Wireless Communications, Networking and Mobile Computing, Wuhan, China. Wuhan: Wuhan University, 2006: 22-24.
- [11] SHROFF N, KWON S. Paradox of shortest path routing for large multi-hop wireless networks[C]//INFOCOM 2007, 26th IEEE International Conference on Computer Communications, Anchorage AK, USA. Anchorage: IEEE INFOCOM, 2007: 1001-1009.

(编辑 赵 静)

(上接第 435 页)

参考文献:

- [1] SIKORA T. The MPEG-4 video standard verification model[J]. IEEE Trans. on Circuits and Systems for Video Technology, 1997, 7(1):19-31.
- [2] 许小东,徐佩霞.基于 TMS320DM642 的视频解码系统优化[J].数据采集与处理,2005,20(1):91-95.
XU XIAO-DONG, XU PEI-XIA. Optimization of Video Decoder System Based on TMS320DM 642[J]. Journal of Data Acquisition & Processing, 2005, 20(1):91-95.
- [3] GUO H, SHENG T, SUN W. Cache optimization for an embedded MPEG-4 video decoder[C]. Nanning China: The 8th International Conference on Signal Processing, 2006.
- [4] 胡君红,江浩洋,樊荣.一种基于 C64x 的多 DSP 实时图像处理系统[J].计算机工程,2006,32(11):268-270.
HU JUN-NHONG, JIANG HOA-XING, FAN RONG. A C64x-based Multi-DSP Real-time Image Processing System [J]. Computer Engineering, 2006, 32(11):268-270.
- [5] 薛漪,马伍新.基于 TMS320DM642 芯片 H.264 编码器设计[J].计算机工程与设计,2005,26(5):1158-1160.
- XUE YI, MA WU-XIN. Design of H.264 coder based on TMS320DM642 chip[J]. Computer Engineering and Design, 2005, 26(5):1158-1160.
- [6] DENOLF K, VLEESCHOUWER C, TURNEY R. Memory centric design of an MPEG-4 video encoder[J]. IEEE Trans. on Circuits and Systems for Video Technology, 2005, 15(5):609-619.
- [7] 费伟,朱向军,裘塞海.基于 C64x DSP 的 MPEG-4 视频编码器算法设计及优化[J].信号处理,2007,23(2):256-261.
FEI WEI, ZHU XIANG-JUN, QIU SAI-HAI. Algorithms Design and Optimization of MPEG-4 Video Encoder Using C64x DSP[J]. Signal Processing, 2007, 23(2):256-261.
- [8] 王钢,王世刚.基于 TMS320DM642 DSP 的 MPEG-4 视频编码器优化[J].吉林大学学报:信息科学版,2006,24(6):590-593.
WANG GANG, WANG SHI GANG. MPEG-4 Video Encoder Realization and Optimization Based on TMS320DM642 DSP[J]. Journal of Jilin University, 2006, 24(6):590-593.

(编辑 陈移峰)