

文章编号：1005-8451(2011)10-0017-04

基于TMS320C672x的C语言源代码优化

李 红，杜普选

(北京交通大学 国家电工电子教学基地，北京 100044)

摘要：本文以TMS320C672x为平台，分析数字信号处理器(DSP)芯片的特点，阐述程序开发和C语言源代码优化的流程，深入探讨代码编写过程中的3种主要优化方法。对优化前、后的程序运行时间进行测量与比较，并对优化结果进行评价。结果表明，针对不同的运算任务采用恰当的优化方法能极大地提升程序的运行效率。

关键词：数字信号处理器；C语言；优化；效率

中图分类号：U29 : TP391 **文献标识码：**A

Optimization of C Language code based on TMS320C672x

LI Hong, DU Pu-xuan

(National Teaching Center of Electrical and Electronic, Beijing Jiaotong University, Beijing 100044, China)

Abstract: This paper briefly analyzed the characteristics and development process of the DSP chip based on the platform of TMS320C672x, described the process of program development and code optimization, deeply discussed the method of optimization using DSP library. The running time of program was compared and tested before and after optimization, the results of these optimizations were summarized. The results showed that it could greatly enhance the efficiency of program operation if appropriate optimization method was chosen for different computing tasks.

Key words: DSP; C Language; optimization; efficiency

DSP—数字信号处理器，在当下的科研、工程、教学等领域有着极为广泛的应用。在现代DSP的开发过程中，C语言逐渐成为主要的开发语言，但由于DSP的特殊结构，使得C编译器在DSP开发过程中的效率较低，编译产生的汇编代码有较大的冗余，导致DSP的运算能力难以得到充分发挥。因此，对C语言源代码的优化成为DSP软件开发过程中一个至关重要的环节。本文针对TMS320C672x系列DSP，研究相关的C语言代码优化问题。

1 TMS320C672x的程序开发

随着DSP应用的日趋复杂，汇编语言程序在可读性、可修改性、可移植性和可重用性上的缺点日益突现，同时汇编语言是一种非结构化的语言，对于大型的结构化程序设计已经逐渐难以胜任，这就要求我们采用更高级的语言去完成这一工作。而在高级语言中，C语言无疑是高效、灵活的。但是由于DSP的特殊结构使得该平台上的C语言编

译器无法充分发挥DSP器件的性能优势，因此有必要根据DSP的特性对C语言代码进一步优化，满足其在日益广泛的应用场合下的效率要求。

TMS320C672x程序开发过程可以简单概括为以下步骤：

(1) 编辑C语言源程序，将编辑好的源程序用C Compile进行编译，每一个源文件经过编译后产生一个*.obj目标文件，目标文件采用COFF的格式。

(2) 使用链接器，链接生成的目标文件，产生一个*.out的可执行文件，这个过程完成了地址的重分配和解析外部引用。

(3) 得到可执行文件之后就可以进行调试。可用软件仿真器在PC机上对指令和运行时间进行精确仿真，或者用硬件仿真器在目标板上进行调试。

(4) 调试通过后即可下载到目标板进行独立运行。

程序开发流程初步完成后，为了考虑效率的问题，要对程序进行代码优化处理，判断所作的优化是否能够提高运行效率，图1给出了程序开发和优化的流程，本文主要讨论第2阶段优化C语言源代码的3种优化方法。

收稿日期：2010-12-12

作者简介：李 红，在读硕士研究生；杜普选，副教授。

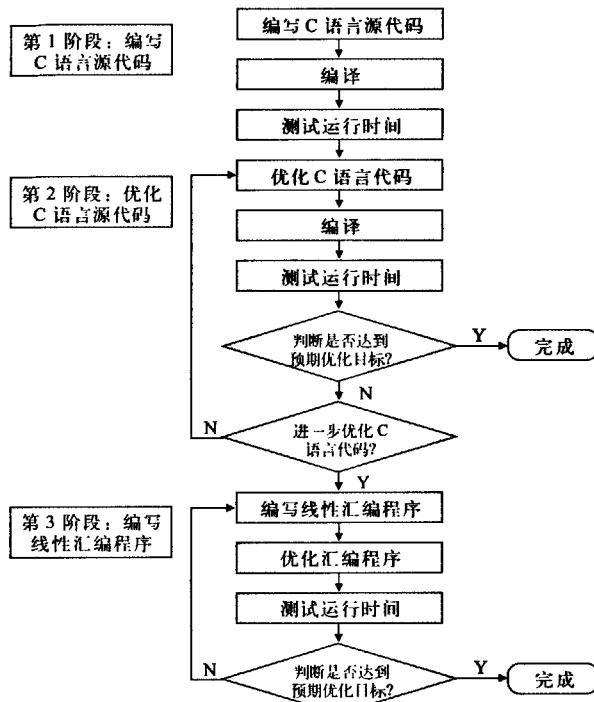


图1 程序开发和优化流程图

2 C语言源代码优化方法

2.1 手工优化

通过对代码的分析得出能提高程序效率的相关程序编写的技巧，可以减少运算时间。

(1) 为了提高算法的效率，将计算时的参数转换成查找表格或常数数值，使运行时的计算转化为编译时的计算，这样减少了运行时计算的时间开销，在比较耗时的运算（如浮点除法）中，更能体现优势。

(2) 在C语言中，对数组特别是对多维数组的寻址是非常耗时的，因此，在程序中尽量使用低维的数组，并且进一步将其指针化，那么在运算过程中，采用指针的C程序不用计算数组地址，只需将指针递增，由于省去了每次重新计算地址的过程，运算工作量会大大减少，可读性也增加。

(3) C672x编译器提供的intrinsic内联函数代替复杂的C代码也可以快速优化C语言代码，提高代码的效率，intrinsic用前下划线表示使用时用调用函数的方法调用它。例如，一个饱和加法的C代码：

```
int sadd(int a,int b)
{
}
```

```

int result;
result=a+b;
if(((a^b)&0x80000000)==0)
{
    if((result^a)&0x80000000)
    {
        result=(a<0)?
0x80000000:0x7fffffff;
    }
}
return(result);
}

```

执行这段代码需要多个时钟周期，但是如果使用内联函数：result=_sadd(a,b)就大大提高了它的效率，只需一条指令即可完成。

2.2 编译器优化

在TMS320C672x编译器中提供了不同种类的若干等级的自动优化选项，如下：

-on：使能软件流水和其他优化方法。共有4个级别。-o1、-o2、-o3、-o4。优化器默认为-o2，它对文件级别进行最强的优化。

-pm：指定其他文件可以调用当前文件中定义的函数和变量。共有4个级别：-op0、-op1、-op2、-op3。

-mt：使能编译器假设程序中没有数据存储混淆，可进一步优化代码。

-mg：使能分析优化代码。

-ms：确保不产生冗余循环，从而减小代码的尺寸。

-k：保留汇编文件以便检查。

-mu：禁止软件流水。

-mh：允许投机执行。

-mx：使能软件流水循环重试，该选项基于循环次数对循环使用多个方案，以便选择最佳方案。

软件流水可以用来安排循环指令，使这个循环的多次迭代并行执行。在编译时使用-o2和-o3选项，编译器可以对循环代码实现软件流水。代码中的循环对执行速度起重要作用，因此改进代码性能，主要就是改进代码中的循环程序。循环次数是一个循环中最重要的因素，减少循环次数将最大限度地减少该循环的指令。软件流水的循环一

般采用递减方式对循环进行计数。优化器将优化前的递增指令 `for(i=0;i<N;i++)` 优化成递减指令 `for(i=N;i!=0;i--)`。这样优化的目的是使优化器在优化之前就知道循环的次数，以便进行循环的拆开等优化工作，产生更快更紧凑的代码。如果编译器不能确定一个循环执行的循环次数，它将产生一个冗余的、不进行流水的循环，当运行的循环次数小于最小循环次数时，冗余的不流水循环被执行。否则，执行软件流水形式的循环。如果编译器不能确定循环的最小次数，则总是不执行软件流水，从而大大降低程序运行的效率。

下面分析一段 FFT 函数的编译器优化，这段函数段包括了采样，加窗等工作。

```
void main()
{
    .....
    msamp ();
    haming(INPUT,8192);
    msamp2();
    /* 产生旋转因子 */
    gen_w_r2(w, 8192);
    DSP_bit_rev(w, 8192 >> 1);
    DSP_FFT(fft, w, 8192);
    DSP_bit_rev(fft, 8192);
    .....
}
```

对这段程序进行 -on 优化分析，用 DSP 环境 CCS 中自带的剖析工具“profiler”中的 clock 工具观察运行所需的时钟周期，没有经过优化时运行需要 17,918,185 个时钟周期（见图 2），分别对它进行 -o0,-o1,-o2 级别的优化，可以得到 -o0 级别的优化需要 11,423,076 个时钟周期（见图 3），-o1 级别的优化需要 9,712,159 个时钟周期，-o2 级别的优化需要 11,423,427 个时钟周期。相比没有经过优化时效率有了很大的提高，-o0 级别提高了 36%，-o1 级别提高了 46%，但是 -o2 级别的优化效率只提高了 36%，发生这个现象是由于在个别情况下使用这个选项优化程序可能会出现错误，可能是在优化循环，组织流水线的时候发生错误。如果有这种现象出现可以同时使用 -g 选项，程序优化就不会出现错误，但是优化效果会下降。尽管如此，选择合适的优化选项还是可以

① : 17,918,185 Ln 1, Col 1

图 2 未优化的运行时钟周期

① : 11,423,076 Ln 20, Col 30

图 3 -o0 级优化后运行时钟周期

大大减少程序运行的时钟周期，对提高效率有显著的作用。

2.3 库函数优化

DSP 库函数的核心实际上是一系列经过手工优化的汇编程序代码，这些代码封装在后缀名为 *.lib 的文件中，可用于完成各种运算。它们对外是不可见的。这些函数可被 C 程序调用。由于经过了手工优化，它们的效率都非常高。一个完整的 DSP 库函数通常由 lib 文件夹、include 文件夹和其它辅助文件组成。其中 lib 文件夹用于存放 *.lib 文件，其内部封装着手工优化的汇编程序代码，是一个 DSP 库函数的核心部分。

DSP 数据处理过程中，卷积、FFT、FIR 滤波等操作频繁出现，所以使用 DSP 库函数来完成程序开发，将会极大地提高整个程序的效率并简化编程。

以 FFT 的库函数使用为例讨论库函数的优化效果。

选取 DSP 库函数中的基 2 时间抽取 “DSPF_sp_cfftr2_dit()” 完成 FFT 运算。它的函数形式是：

```
void DSPF_sp_cfftr2_dit(float *x,float *w,
short n)
```

其中 x 为输入复数组，w 是旋转因子，n 为输入数组的长度。

程序中的输入选用正弦函数。程序选段如下：

```
#include "DSPF_sp_cfftr2_dit.h"
.....
void main()
{
    .....
    /* 实部 */
    x[2 * i] = (sin(2 * PI * F1 * i / N));
    /* 虚部 */
}
```

```

x[2 * i + 1] = 0.0;
.....
/* 产生旋转因子 */
gen_w_r2(w, N);
/* 旋转因子倒叙 */
bit_rev(w, N >> 1);
DSPF_sp_cfftr2_dit(x, w, N);
.....

```

分别对使用库函数和不使用库函数做不同点数的FFT运算，分析运行的时间，得到表1相关数据。从表1可以看出，使用库函数的效率远远高于使用源函数，节省的时钟周期数随着输入序列长度的变化而变化，最多能相对节省96.5%的时钟周期数，少的也能节省85.5%的时钟周期数，能十分有效地提高编程效率，节省运行时间。

表1 库函数和源函数时间分析

	128点	256点	512点	1024点
源函数周期数	155 344	223 888	290 064	422 928
库函数周期数	5,465	12,377	27,737	61,529
库函数节省时钟周期 (%)	96.5%	94.5%	90.4%	85.5%

3 结束语

用C语言开发DSP芯片大大提高了程序的可读性和可移植性，对于系统的改进和升级换代也

(上接P16)

可以将自己的学习心得上传到服务器供所有学生分享。多媒体播放模块的主要功能是实现将视频和音频实时发送到所有的学生端。

3.2.4 单独交流模块

单独交流模块其实是用于单个的学生和老师交流，其本质也是语音传输，只是此时不再是一对多，而且传输的方向是双向的，所以其相关技术也与课堂教学模块一样，不再赘述。

4 结束语

本文通过对现有语音教学系统的调查研究，总结其优点和缺点，设计和实现了一种基于嵌入

带来了极大的便利。随着DSP承担的运算任务复杂度增大，如何对C语言源代码优化、使DSP能更充分利用资源成为软件开发过程中的关键环节。本文在DSP TMS320C672x平台上，通过手工优化、编译器优化和库函数优化3种途径对代码进行了优化，从对结果的分析可以看出，这3种优化方法可以有效节省程序运行时间，对提高系统的实效性、可读性和快速性起到重要的作用，可为执行大规模复杂算法提供有效的时间保障。

参考文献：

- [1] 汪安民, 张松灿, 常春藤. TMS320C6000DSP实用技术与开发案例[M]. 北京: 人民邮电出版社, 2008.
- [2] 高学强, 王玉晶, 康锡章. 基于TMS320C67xx DSP的软件开发及优化[J]. 现代计算机, 2003 (11).
- [3] 杨光宇, 高晓蓉, 王黎, 王泽勇. 基于TI C6000系列DSP的C/C++程序优化技术[J]. 现代电子技术, 2009 (8): 55-59.
- [4] 刘朝晖, 郑玉墙. 用C语言进行DSP软件设计的优化考虑[J]. 空军雷达学院学报, 2001 (2): 49-52.
- [5] 阳明晔, 张志勇. 基于TMS320C6000系列DSP的C代码优化方法研究[J]. 微处理机, 2003 (2): 59-64.
- [6] 任志考. 用C语言进行DSP软件设计的优化方法[J]. 青岛科技大学学报, 2003 (24): 102-104.

责任编辑 杨利明

式技术的无线语音教学系统，该系统使得语音教学不再局限于固定的场所，可以灵活选择教学场所，另外该系统具有成本低、灵活性好、通用性好以及功能丰富等特点。

参考文献：

- [1] 赵莹莹, 张兰芬. 基于RTP协议的音频传输技术的研究与实现[J]. 消费电子, 2006 (10): 30-38.
- [2] 潘桐. G.726语音压缩算法在低码率下的改进与实现[J]. 微计算机应用, 2009, 30 (4): 76-80.
- [3] 赵代强. 基于数字语音交换技术的多方会议系统[J]. 计算机工程, 2004, 30 (13): 87-88.

责任编辑 杨利明