



国家人力资源和社会保障部 信息专业技术人才知识更新工程（“653工程”）指定教材
国家工业和信息化部
全国高等职业教育“十一五”计算机类专业规划教材

C / C + + D U O M E I T I K A I F A
A N L I S H I Z H A N

C/C++多媒体开发 案例实战

丛书编委会



中国电力出版社

<http://jc.cepp.com.cn>



国家人力资源和社会保障部 信息专业技术人才知识更新工程（“653工程”）指定教材
国家工业和信息化部
全国高等职业教育“十一五”计算机类专业规划教材

C / C + + D U O M E I T I K A I F A
A N L I S H I Z H A N

C/C++多媒体开发 案例实战

丛书编委会



中国电力出版社

<http://jc.cepp.com.cn>

内容提要

本书是一本使用 C/C++ 语言进行多媒体编程的书，其内容重点放在代码优化上面，也就是如何将相对简单的原理性代码优化为具有实用价值的、高效、稳定的程序。书中的代码大部分是 C 代码，MMX、SSEZ 的版本都在本书所附的光盘中的工程范例里面。

本书内容共有 12 章，分别介绍了数字图像、数字视频的一些基本概念和常用的编码解码过程，以及后期处理的重要函数。章节安排如下：第 1 章为程序优化，主要介绍程序优化的基本原则和方法；第 2 章为数字图像、数字视频基础；第 3 章为 get_bits 函数；第 4 章为 CABAC 过程；第 5 章为 IDCT 反离散余弦变换；第 6 章为空间预测；第 7 章为矩阵置换；第 8 章为去隔行；第 9 章为图像处理；第 10 章为 MMX 整数运算在音频解码中的使用；第 11 章为大整数运算；第 12 章为多线程开发。

本书主要作为信息专业技术人才知识更新工程指定教材，也可作为普通高等学校计算机及相关专业的教材。同时，除了适合从事多媒体开发的程序员作为参考材料以外，也可作为计算机相关专业硕士生的有关多媒体的补充材料。

图书在版编目 (CIP) 数据

C/C++ 多媒体开发案例实战 / 《国家人力资源和社会保障部、国家工业和信息化部信息专业技术人才知识更新工程 (“653 工程”) 指定教材》编委会主编. —北京：中国电力出版社，2008

国家人力资源和社会保障部、国家工业和信息化部信息专业技术人才知识更新工程 (“653 工程”) 指定教材

ISBN 978-7-5083-7214-3

I. C… II. 国… III. C 语言—程序设计—高等学校：技术学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2008) 第 126307 号

书 名：C/C++ 多媒体开发案例实战

出版发行：中国电力出版社

地 址：北京市三里河路 6 号

邮政编码：100044

电 话：(010) 68362602

传 真：(010) 68316497, 88383619

服务电话：(010) 58383411

传 真：(010) 58383267

E-mail: infopower@cepp.com.cn

印 刷：北京丰源印刷厂

开本尺寸：185mm×260mm 印 张：26.25 字 数：595 千字

书 号：ISBN 978-7-5083-7214-3

版 次：2008 年 9 月北京第 1 版

印 次：2008 年 9 月第 1 次印刷

印 数：0001—3000 册

定 价：39.80 元 (含 1CD)

敬告读者

本书封面贴有防伪标签，加热后中心图案消失
本书如有印装质量问题，我社发行部负责退换

版权专有 翻印必究

专家指导委员会

- 郭贺铨 中国工程院副院长 中国工程院院士
- 王 越 中国科学院院士 中国工程院院士
- 何积丰 中国科学院院士
- 潘云鹤 中国工程院院士
- 顾冠群 中国工程院院士
- 卢锡城 中国工程院院士
- 张乃通 中国工程院院士
- 李乐民 中国工程院院士
- 沈昌祥 中国工程院院士
- 方滨兴 中国工程院院士
- 张尧学 中国工程院院士 教育部高等教育司司长
- 高新民 国家信息中心原主任
- 魏 卓 人力资源和社会保障部专业技术人员管理司副司长
- 陈 冲 中国软件行业协会理事长
- 牛 晋 公安部信息通信局副局长
- 邓寿鹏 中国信息化推进联盟常务副理事长 原国务院发展研究中心局长
- 李明树 中国科学院软件研究所所长
- 陈 钟 北京大学软件与微电子学院院长
- 吴世忠 中国信息安全产品测评认证中心主任
- 王行刚 中国科学院计算技术研究所首席科学家
- 刘玉珍 工业和信息化部电子人才交流中心主任

丛书编委会

主 任 李建伟 逢积仁

副 主 任 黄雪峰 邸卫民 吕振凯 杨敬杰 杨功元 王建良 冯玉东
张志平 杨文利 李 新 李 缨 徐 峰 敖广武 陈 红
张学金 商 桑 赵耀培

委 员 (按拼音排序)

鲍金龙	曹素丽	陈国浪	陈青华	陈小中	陈月波	程 云
崔爱国	崔守良	丁 倩	丁荣涛	丁银军	杜少杰	杜文洁
范荣真	房振文	顾 爽	和海莲	胡新和	槐彩昌	嵇新浩
吉高云	李道旺	李 华	李立功	李 敏	李 霞	李英明
李艳玮	李玉清	林志伟	刘红军	刘 静	刘俊英	刘 颖
毛书朋	聂庆鹏	乔国荣	邱春民	荣 音	宋林林	滕红军
田文浪	涂 刚	王灿伟	王春燕	王 磊	王丽芬	王 盟
王赵慧	王志新	魏建明	温丹丽	谢建华	谢 菁	辛颖秀
徐长安	徐春华	徐 伟	严春风	阎 琦	杨光洁	叶若芬
叶展翔	于 畅	袁胜昔	翟鹏翔	张爱华	张洪明	张 琳
张兴科	张云鹏	张 震	赵思宇	郑伟勇	周国亮	周连兵
周瑞华	朱红祥	朱元忠				

本书编委会

主 编 鲍金龙

副 主 编 马宇飞 吉高云 徐 峰 王 磊

参 编 邢燕军 隋治强 王英合 王素倩

丛书编委会院校名单

(按拼音排序)

保定电力职业技术学院	日照职业技术学院
北京电子科技职业学院	山东电子职业技术学院
北京工业职业技术学院	山东济宁职业技术学院
北京建筑工程学院	山东交通职业学院
北京市经济管理学校	山东经贸职业学院
北京市宣武区第一职业学校	山东省工会管理干部学院
滨州职业学院	山东省潍坊商业学校
渤海大学高职学院	山东丝绸纺织职业学院
沧州职业技术学院	山东信息职业技术学院
昌吉职业技术学院	山东枣庄科技职业学院
大连工业大学职业技术学院	山东中医药高等专科学校
大连水产学院职业技术学院	沈阳师范大学职业技术学院
东营职业学院	石家庄邮电职业技术学院
河北建材职业技术学院	苏州建设交通高等职业技术学校
河北旅游职业学院	苏州托普信息职业技术学院
河南工程学院	天津铁道职业技术学院
河南农业职业学院	潍坊职业学院
湖北省仙桃职业学院	温州职业技术学院
嘉兴职业技术学院	无锡南洋职业技术学院
江门职业技术学院	武汉软件工程职业学院
江苏财经职业技术学院	咸宁职业技术学院
江苏常州工程职业技术学院	新疆农业职业技术学院
金华职业技术学院	新余高等专科学校
莱芜职业技术学院	兴安盟委党校
辽宁机电职业技术学院	浙江金融职业学院
辽宁金融职业学院	浙江商业职业技术学院
辽宁经济职业技术学院	郑州电力高等专科学校
辽宁科技大学高等职业技术学院	中国农业大学继续教育学院
青岛滨海学院	中国青年政治学院
青岛酒店管理职业技术学院	中华女子学院山东分院
青岛职业技术学院	淄博职业学院

丛 书 序

自 20 世纪 90 年代以来,伴随着信息技术创新和经济全球化步伐的不断加快,全球信息化进程日益加速,中国的经济社会发展对信息化提出了广泛、迫切的需求。党的十七大报告做出了要“大力推进信息化与工业化融合”,“提升高新技术产业,发展信息、生物、新材料、航空航天、海洋等产业”的重要指示,这对信息技术人才提出了更高的要求。

为贯彻落实科教兴国和人才强国战略,进一步加强专业技术人才队伍建设,推进专业技术人才继续教育工作,人力资源和社会保障部组织实施了“专业技术人才知识更新工程(‘653 工程’)”,联合相关部门在现代农业、现代制造、信息技术、能源技术、现代管理等 5 个领域,重点培训 300 万名紧跟科技发展前沿、创新能力强的中高级专业技术人才。工业和信息化部与人力资源和社会保障部在 2006 年 1 月 19 日联合印发《信息专业技术人才知识更新工程(“653 工程”)实施办法》(国人部发[2006]8 号),对信息技术领域的专业技术人才培养进行了部署和安排,提出了要在 6 年内培养信息技术领域中高级创新型、复合型、实用型人才 70 万人次左右。

作为国家级人才培养工程,“653 工程”被列入《中国国民经济和社会发展第十一个五年规划纲要》和《2006—2010 年全国干部教育培训规划》,成为建设高素质人才队伍的重要举措。

本系列教材作为“653 工程”指定教材,严格按照《信息专业技术人才知识更新工程(“653 工程”)实施办法》的要求,以培养符合社会需求的信息专业技术人才为目标,汇聚了众多来自信息产业部门、著名高校、科研院所和知名企业的学者与技术专家,组成强大的教学研发和师资队伍,力求使教材体系严谨、贴近实际。同时,教材采用“项目驱动”的编写思路,以解决实际项目的思路和操作为主线,连贯多个知识点,语言表述规范、明确,贴近企业实际需求。

为了方便教师授课和学生学习,促进学校教学改革,提升教学质量,本系列教材不仅提供教师授课所用的教学课件、习题和答案解析,而且针对教材中所涉及的案例、项目和实训内容,提供了多媒体视频教学演示课件。另外,在教学过程中,随时可以登录教师之家——中国学术交流网(www.jiaoshihome.cn),寻求教学资源的支持,我们特别为每一本教材设置了针对教师授课和学员学习的答疑论坛。同时,本套教材举办“有奖促学”活动,凡购买本套教材,学习完后,举一反三创作出个人作品,上传至教师之家——中国学术交流网,每个学期末将根据创作内容和网站点击率综合评选一次,选出一、二、三等奖和纪念

奖，并在假期中颁发奖项。

学员学习本系列教材后经考核合格，可以申请“专业技术人才知识更新工程（‘653 工程’）培训证书”。该证书可以作为专业技术人员职业能力考核的证明，以及岗位聘用、任职、定级和晋升职务的重要依据。

我们希望以本系列教材为载体，不断更新教学内容，改进教学方法，搭建学校与企业沟通的桥梁，大力推进校企合作、工学结合的人才培养模式，探索一条充满生机和活力的中国信息技术人才培养之路，为建设社会主义和谐社会提供坚强的智力支持和人才保证。

丛书编委会

本书序

DVD、EVD、蓝光、高清晰数字电视、流媒体、视频网站、播客、3G、TD-CDMA、视频电话、视频聊天、CMMB 手机电视、CGI 电影动画，听着这些熟悉的科技词汇，感受数字化浪潮的扑面而来。而在绚丽多彩的数字媒体技术后面，是广大科技工作者的辛苦工作和牺牲付出。对于每一个想从事数字媒体开发的程序员来说，用 C/C++ 做音视频开发，不仅仅是入门困难，而且关于这些底层技术的文档和书籍又是少之又少，更有数不清的技术难题和知识产权的壁垒。因此这本书的编写成功是相当及时并令人感到兴奋的。

本书的主编鲍金龙同志是一个相当勤奋的程序员。在 2000 年的时候，担任阜国数字有限责任公司的技术总监期内，带领技术团队组织科技攻关，每天只休息不足 6 小时，保证了 EVD 国标的开发、制定顺利完成。2001 年到金山软件有限责任公司担任金山影霸的技术总监，负责新一代金山影霸播放器的开发工作，总是随身带着笔记本电脑，随时键入代码，甚至在公司开会的休息时间内也会工作几分钟。2002 年 6 月发布了金山影霸 2003，这个产品带有绚丽的外观设计和强大的 DVD 特色功能，被媒体评为当年最出色的国产工具软件之一。2004 年以后，鲍金龙同志继续在 MPEG2、MPEG4、H.264 及后续数字音视频压缩技术方向上探索学习，发表了《影海风雷播放器》、《影海风雷媒体中心》、《影海风雷流媒体》等多个软件产品，其中《影海风雷 2006》是当年第一个支持多核个人电脑处理器的解压软件，获得广泛好评。可以说，在音视频开发上鲍金龙同志是一个经验相当丰富的专家，是程序员实战中的权威。相信这本书一定会给在音视频开发方向上的广大读者带来切实有用的经验和技能。



金山软件副董事长 雷军



前 言

这是一本关于使用 C/C++ 语言进行多媒体编程的书，其内容重点放在代码优化上面，也就是如何将相对简单的原理性代码优化为具有实用价值的、高效、稳定的程序。

书中提供的程序涉及了众多的数字图像处理、数字视频处理、数字信号处理等基础理论知识，以及多媒体技术标准如 MPEG2、MPEG4、H.264 等。具有这方面的知识背景当然是非常有益的。但是算法不是本书讨论的中心，对程序员来说，面对的往往不是算法的问题，而是把已有的算法转化为程序的问题。因此使用本书的程序员朋友大可以先动手解决实际问题，有时间的话再逐步了解有关背景知识。

对于所有的实现过程，基本上都提供了 C 语言、MMX、SSE2 优化等两到三个版本。由于使用汇编语言的代码过于庞大，因此相当一部分 MMX、SSE2 优化的版本并没有在书中罗列，而是放在了本书所附光盘的工程范例中。但是在书中至少给出了一种实现，通常是 C 语言版本。因此本书在脱机的情况下基本上也可以使用。

所有的工程范例都是在 Visual C++ 6.0 环境下编译的。需要注意的是需要安装补丁 SP5，并最好安装有 DirectX 9。使用更高版本的开发环境，由于类型检查更为严格的关系，代码可能需要做一些修改。

本书内容共有 12 章，分别介绍了数字图像、数字视频的一些基本概念和常用的编码解码过程，以及后期处理的重要函数。第 1 章为程序优化，介绍程序优化的基本原则和方法。第 2 章为数字图像、数字视频基础。第 3 章为 get_bits 函数。第 4 章为 CABAC 过程。第 5 章为 IDCT 反离散余弦变换。第 6 章为空间预测。第 7 章为矩阵置换。第 8 章为去隔行。第 9 章为图像处理。第 10 章为 MMX 整数运算在音频解码中的使用。第 11 章为大整数运算。第 12 章为多线程开发。

熟练地掌握 C/C++ 语言及相应的开发工具是参考本书的必要前提，本书适用于从事多媒体开发的程序员，也可作为普通高等学校计算机及相关专业的教材和参考材料，以及计算机相关专业硕士生的多媒体专业的补充材料。

对于书中存在的不足和范例工程中可能有的错误，希望程序员朋友们予以批评指正。

编 者

2008 年 5 月



目 录

丛书序

本书序

前 言

第 1 章 程序优化1

- 1.1 确定需要优化的重点——性能瓶颈1
- 1.2 循环展开2
- 1.3 分支程序4
- 1.4 乘法和除法、定点和浮点7
- 1.5 表格的使用7
- 1.6 SIMD 技术的使用13
- 1.7 数据纠错55

第 2 章 数字图像、数字视频基础57

- 2.1 RGB 位图和 RGB 格式57
- 2.2 RGB 16、24、32 的转换60
- 2.3 YUV 格式71
- 2.4 YUV 格式之间的转换77
- 2.5 RGB 到 YUV 的转换80
- 2.6 YUV 到 RGB 的转换89
- 2.7 数字视频压缩和相关标准91

第 3 章 get_bits 函数95

- 3.1 get_bits 操作95
- 3.2 常见的 C 语言实现96
- 3.3 改进版本98
- 3.4 MMX 版本100
- 3.5 性能测试104

第 4 章 CABAC 过程108

- 4.1 CABAC 介绍108

- 4.2 CABAC 的解码器程序109

- 4.3 CABAC 的解码器优化112

- 4.4 C 代码113

- 4.5 MMX 优化115

第 5 章 IDCT 反离散余弦变换 126

- 5.1 二维反余弦变换的基本算法：
AAN 和 2D 算法126
- 5.2 快速算法和快速通道的结合136
- 5.3 快速通道的 MMX 优化157

第 6 章 空间预测 198

- 6.1 空间预测的基本方式198
- 6.2 MMX 技术在空间预测上的
应用199

第 7 章 矩阵置换 219

- 7.1 矩阵置换的 C 函数219
- 7.2 MMX、SSE2 优化221
- 7.3 性能测试226
- 7.4 使用矩阵转置进行位图的旋转227

第 8 章 去隔行 235

- 8.1 隔行图像毛刺现象的成因236
- 8.2 测试用例239
- 8.3 去隔行 API242

第 9 章 图像处理 260

- 9.1 RGB 色度、亮度的调整260
- 9.2 YUV 色度、亮度的调整269
- 9.3 图像的锐化275

9.4 图像的缩放.....	287
9.5 图像的旋转.....	310

第 10 章 MMX 整数运算在音频解码中的使用.....329

10.1 利用 MMX PMADDWD 指令实现高精度乘法的原理	329
10.2 程序实现	330

第 11 章 大整数运算.....337

11.1 大整数的表示.....	337
11.2 大整数加法.....	337
11.3 大整数减法和取反.....	342

11.4 大整数比较和移位	343
11.5 大整数乘法	346
11.6 大整数除法	357
11.7 大整数开平方	365
11.8 用十进制字符串表示二进制大整数	370
11.9 用 C++封装大整数类型	372
11.10 大数割圆法.....	386

第 12 章 多线程开发.....395

12.1 线程模型	395
12.2 设计原则	397
12.3 测试用例	399



1.1 确定需要优化的重点——性能瓶颈

一般来说，多媒体程序使用 C 语言开发，Visual C（简称 VC）环境下，具体适用于 SIMD 技术的函数，使用 ASM 函数，或者在 C 函数内部使用 `_asm{}` 语句块。

做多媒体开发的时候，一般情况下，我们很少从 0 开始写一个模块，基本上是由一个能初步运行的代码再作一些改动就可以了，这中间很重要的一部分工作就是性能优化。

作为商用软件，虽然性能指标很重要，但是开发代价也是必须要考虑的。

优化工作的基本思路，简单地讲，就是提升模块中运行时最为耗时的代码的执行速度，也就是把优化的重点指向性能瓶颈部分。所以开始动手优化之前的工作，或者说优化的第一步，就是要确定对哪里进行优化。对于整个模块来说，要确定耗时的函数；对于一个函数来说，要确定耗时的语句。

下面是 MPEG2 解码中常见的空间预测部分的 `pred_comp` 的部分代码。基本上我们目测就能发现需要优化的部分。详细了解 MPEG2 空间预测，请参考 ISO/IEC 13818-2, 7.6: Motion compensation。

```
void pred_comp_c (unsigned char *src, unsigned char *dst,
                  int lx, int w, int h, int x, int y, int dx, int dy,
int addflag)
{
    int xint, xh, yint, yh;
    int i, j;
    unsigned char *s, *d;

    /* half pel scaling */
    xint = dx >> 1;
    xh = dx & 1;
    yint = dy >> 1;
    yh = dy & 1;

    /* origins */
    s = src + lx*(y+yint) + (x+xint);
    d = dst + lx*y + x;

    //以上部分代码的运算量比重很低，对其进行优化对整体性能没有帮助

    if (!xh && !yh) {
        if (addflag) {
```

//这个循环体是需要优化的重点

```

for(j=0; j<h; j++) {
    for(i=0; i<w; i++){
        d[i] = (unsigned int)(d[i]+s[i+1]>>1;
    }
    s+=1x;
    d+=1x;
}
//这个循环体是需要优化的重点
}
//以下代码略去
else{
}
}
//
}

```

实际应用中，比如开发一个 MPEG2 解码器，模块中的函数数量很大，用目测分析整个模块的性能瓶颈就达不到目的了。这时候使用一些性能分析工具，如 Intel 的 Vtune、VC++ (Visual C++的简称) 的 profile 工具等，能大致确定一下模块中哪些函数的运算比重比较大，进而对具体函数作进一步的分析。

图 1-1 是使用 Vtune 对某个解码器程序模块的分析结果，可以初步得到每个函数在模块中的运算比重，如函数 idct8x8aan 是 7.85%。

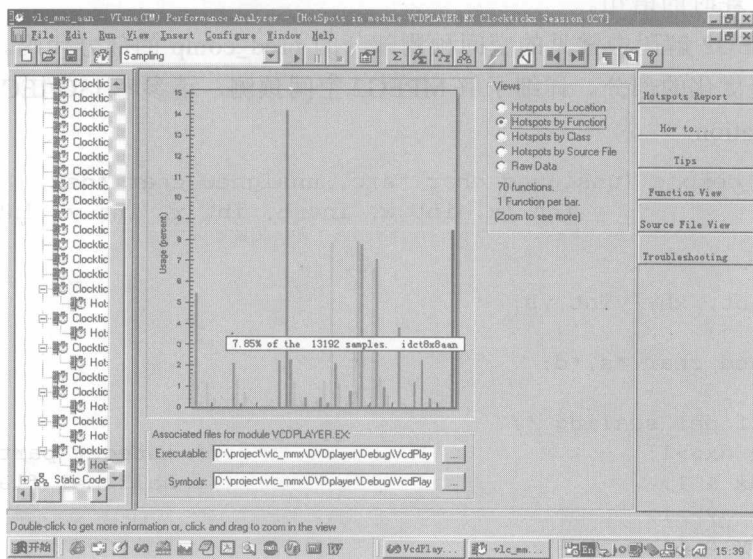


图 1-1 使用性能分析工具

1.2 循环展开

一般说来，如果循环体内 C 语言语句少于 10 行，汇编语句少于 20 行，那么循环就是不充分的，需要做循环展开。

以上面的 `pred_comp` 函数为例，显然循环体内语句过少：

```
for (j=0; j<h; j++) {
    for (i=0; i<w; i++){
        d[i]=(unsigned int)(d[i]+s[i]+1)>>1;
    }
    s+= 1x;
}
```

我们知道 MPEG2 空间参考块的宽度最小是 8，所以定义宏 `mc_avg_p1`：

```
#define mc_avg_p1(m) d[i+m]=(unsigned int)(d[i+m]+s[i+m]+1)>>1;
```

然后展开到 8：

```
for (j=0; j<h; j++) {
    for (i=0; i<w; i+=8) {
        mc_avg_p1(0)
        mc_avg_p1(1)
        mc_avg_p1(2)
        mc_avg_p1(3)
        mc_avg_p1(4)
        mc_avg_p1(5)
        mc_avg_p1(6)
        mc_avg_p1(7)
    }
    s+=1x;
    d+=1x;
}
```

更有效率的做法是，对求平均的算法做一点改进：

```
#define CAVGB(A,B) \
( ( (A|B) & 0x01010101 ) | (((A >> 1) & 0x7f7f7f7f) + ( (B >> 1) & 0x7f7f7f7f) ) )
```

然后再进行展开：

```
for (j=0; j<h; j++) {
    for (i=0; i<w; i+=8) {
        DWORD const A=(DWORD*)(s+i);
        DWORD const B=(DWORD*)(d+i);
        *(DWORD*)(d+i) = CAVGB(A,B);
        DWORD const C=(DWORD*)(s+i+4);
        DWORD const D=(DWORD*)(d+i+4);
        *(DWORD*)(d+i+4) = CAVGB(C,D);
    }
    s+= 1x;
    d+= 1x;
}
```

1.3 分支程序

程序的分支太多，特别是很短的分支，会造成 CPU 指令流水线的停顿。如果能减少分支数量或者没有分支，程序在运行时效率就会得到改善。

下面的代码是从 MPEG2 intra-block 解码过程中提取的。简要地说这部分代码就是首先取得一个不超过 16 位的整数 code，然后根据 code 的数值确定下一步要使用的表格。

DCTtabnext、DCTtab0a、DCTtab0、DCTtab1a、DCTtab1、DCTtab2、DCTtab3、DCTtab4、DCTtab5、DCTtab6 都是数据结构 DCTtab 类型的全局常量列表。PPDCTtab 是指向 DCTtab 类型的指针。详细了解该过程，请参考 ISO/IEC 13818-2, 7.4: Inverse quantisation 和 Annex B: Variable length code tables。

```
code=ShowBits(16);
if(code>=16384 && ! nIntraVlcFormat){
    tab = &DCTtabnext[(code>>12)-4];
}else if (code>=1024) {
    if (nIntraVlcFormat) {
        tab = &DCTtab0a[(code>>8)-4];
    }else{
        tab = &DCTtab0[(code>>8)-4];
    }
}else if (code>=512){
    if (nIntraVlcFormat){
        tab = &DCTtab1a[(code>>6)-8];
    }else{
        tab = &DCTtab1[(code>>6)-8];
    }
}else if (code>=256){
    tab = &DCTtab2[(code>>4)-16];
}else if (code>=128){
    tab = &DCTtab3[(code>>3)-16];
}else if (code>=64){
    tab = &DCTtab4[(code>>2)-16];
}else if (code>=32){
    tab = &DCTtab5[(code>>1)-16];
}else if (code>=16){
    tab = &DCTtab6[code-16];
}else{
    g_bFaultFlag = 1;
    return;
}
```

从以上程序可以看到程序的分支相当多，从直觉上来看这部分代码就不是很好看。

由于计算复杂度的关系，很多时候分支程序不可避免。值得庆幸的是，上面的代码还是可以进行适当地改进的。

下面的代码使用了不是很大的表格，根据变量 code 查表，从而消灭了分支程序。

```

unsigned char g_byM2IntraCode1[2][256];
unsigned char g_byM2Code0[256];
unsigned char g_byM2Code1[256];
unsigned char g_byM2NonIntraSub[16] = { 16, 16, 16, 16, 16, 0, 8, 0, 4,
4, 4, 4, 4, };

```

```

PPDCTtab g_M2NonIntraTab[2][16] = {
{
(DCTtab (*)[16])&DCTtab6,
(DCTtab (*)[16])&DCTtab5,
(DCTtab (*)[16])&DCTtab4,
(DCTtab (*)[16])&DCTtab3,
(DCTtab (*)[16])&DCTtab2,
NULL,
(DCTtab (*)[16])&DCTtab1,
NULL,
(DCTtab (*)[16])&DCTtab0,
NULL, NULL, NULL,
(DCTtab (*)[16])&DCTtabfirst,
},

```

```

{
(DCTtab (*)[16])&DCTtab6,
(DCTtab (*)[16])&DCTtab5,
(DCTtab (*)[16])&DCTtab4,
(DCTtab (*)[16])&DCTtab3,
(DCTtab (*)[16])&DCTtab2,
NULL,
(DCTtab (*)[16])&DCTtab1,
NULL,
(DCTtab (*)[16])&DCTtab0,
NULL, NULL, NULL,
(DCTtab (*)[16])&DCTtabnext,
}
};

```

```

PPDCTtab g_M2IntraTab[2][16] = {
{
(DCTtab (*)[16])&DCTtab6,
(DCTtab (*)[16])&DCTtab5,
(DCTtab (*)[16])&DCTtab4,
(DCTtab (*)[16])&DCTtab3,
(DCTtab (*)[16])&DCTtab2,
NULL,
(DCTtab (*)[16])&DCTtab1,
NULL,
(DCTtab (*)[16])&DCTtab0,
NULL, NULL, NULL,
(DCTtab (*)[16])&DCTtabnext,
},

```