

# 基于 DM642 的 SMV 实现\*

王小兵, 黄冰, 胡巍, 胡斌

(桂林电子科技大学 信息与通信学院, 广西 桂林 541004)

**摘要:** 介绍了变速率语音算法 SMV 的 DSP 实时实现技术。根据标准 SMV 的要求优化算法的定点 C 程序, 减少占用空间, 改善码本的搜索时间, 最终在 DM642 硬件开发平台上运行成功, 语音质量符合通信要求。

**关键词:** 变速率语音算法; SMV; DM642; 语音质量

中图分类号: TN912.32

文献标识码: A

文章编号: 1674-7720(2010)10-0060-04

## Implementation of SMV based on DM642

WANG Xiao Bing, HUANG Bing, HU Wei, HU Bin

(Institute of Information and Communication, Guilin University of Electronic Technology, Guilin 541004, China)

**Abstract:** This paper describes a variable-rate speech algorithm of SMV and real-time implementation technology of DSP. According to the standard requirements of SMV, the optimized fixed-point C code reduces the occupied memory space and improves the codebook search time. Finally SMV runs successfully on the hardware development platform of DM642. The voice quality of SMV meets the communication requirements.

**Key words:** variable-rate speech algorithm; SMV; DM642; voice quality

SMV 是由 3GPP2 组织在 2001 年时提出的 3G 移动通信标准<sup>[1]</sup>, 该算法实现了更高的通信语音质量和更低的占用带宽。SMV 非常适合应用于 CDMA-2000, 并且也能取代 UMTS 系统中的 AMR 声码器, 还能在 3GPP 和 3GPP2 语音网络间提供无缝协作。

### 1 SMV 编解码原理

SMV 基于码本激励线性预测(CELP)方法而设计<sup>[2]</sup>。输入的语音先要经过预处理(包括静音增强、高通滤波、噪声抑制、自适应倾斜滤波), 抑制语音中的背景噪声, 为后面的编码过程作准备。接下来是帧级别处理:(1)线性预测(LPC)和开环基音分析, 利用语音样点间的短时相关和长时相关获得所需参数;(2)帧分类算法对每一帧进行分析, 确定语音的类型;(3)速率判决算法(RDA)根据语音特性和工作模式选择一种编码速率。由编码原理可知, 信号修正部分参与了以上 3 步的决策过程。信号修正完成之后, 若语音帧具有长时相关性, 则归类为类型 1; 若无长时相关性, 则归类为类型 0。SMV 支持 4 种码率和 6 种网络控制操作模式。

SMV 解码结果相对简单, 由四部分组成: 接收端对码流解码, 获得固定码本和固定码本增益、自适应码本和自适应码本增益。它们合成的 LPC 激励信号作用于 LPC 合成器滤波器, 产生合成语音。当有错误出现时, 采用差错隐藏算法处理。

### 2 DM642 平台解决方案

DM642 是德州仪器公司推出的高性能数字多媒体处理器, 主频在 720 MHz 下处理速度达到 5 760 MIPS。超长指令字(VLIW)的 DM642 核具 64 个 32 位字长的通用寄存器, 8 个独立的功能单元(.L1、.L2、.S1、.S2、.D1、.D2、.M1 和 .M2), 2 个寄存器文件(A 和 B)和 2 个数据交叉通道(1X 和 2X)。这些硬件资源对等分配为两个相同的组, 每组占用一个数据交叉通道。DM642 每周期能够提供 4 个 16 位 MAC, 两级缓存: 一级程序缓存 L1P 是一个 128 Kbit 的直接映射缓存, 另一级数据缓存 L1D 是一个 128 Kbit 双路结合设置缓存。L2 缓存器能被配置成映射存储器、高速缓存或者两者结合。其片上外设资源非常丰富。

\* 基金项目: 广西自然科学基金资助项目(0640167)

## 网络与通信 Network and Communication

### 2.1 硬件结构

在 SEED-VPM642 系统中, McASP 同时接入 4 路音频输入和输出, 协议采用 BurstFrame-SyncMode。TLV320AIC23B 采用立体声输入方式, 设置采样率为 8 kHz, 提供 16 bit 的采样值, ADC 和 DAC 的信噪比分别可以达到 90 dB 和 100 dB。DM642 使用 I<sup>2</sup>C 总线读写 AIC23B 的控制寄存器。I<sup>2</sup>C 总线根据设备地址区分访问设备, 不能同时连接 4 个 AIC23B 芯片所以 I<sup>2</sup>C 总线以 GP0[0] 作总线切换控制。McASP 工作在从模式, 只能有一片 AIC23B 工作在主模式, 其他 3 片工作在从模式。硬件平台中, 4 路音频的帧同步信号、发送与接收时钟都是共用的。第一路音频原理图如图 1 所示, 其他 3 路音频结构递推即可。

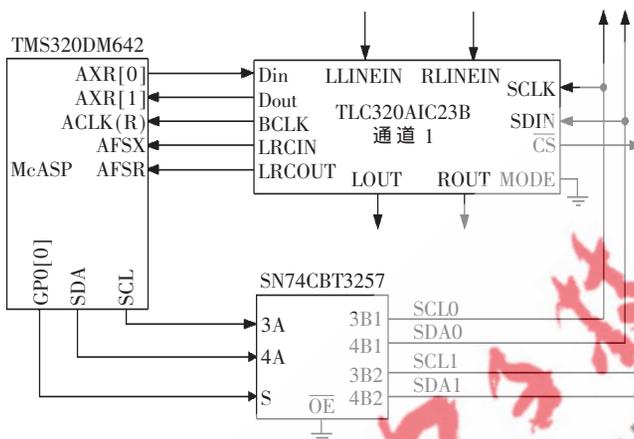


图 1 第一路音频连接示意图

### 2.2 音频设备驱动程序结构

DM642 硬件平台采用 TI 公司的 DDK(Drive Developer's Kit), 简化了音频驱动程序的开发难度, 也提高了驱动程序的移植性。DDK 是 CSL(Chip Support Library) 的补充, 定义了标准模型和一系列的 API。DDK 实际上是 IOM Mini-drive 的一部分, 处在两层器件驱动模型的 Mini-drive。IOM 分层驱动模型如图 2 所示。



图 2 IOM 分层驱动模型

Class-drive 层可以设置为 DSP/BIOS 的 GIO、SIO/DIO 或者 PIP/PIO 类驱动。Class-drive(类驱动)层向上兼容, 为应用程序提供通用的 API, 与器件相对独立, 完成缓

冲区的管理和请求同步等功能; Mini-drive(微型驱动)层向下兼容, 直接控制底层硬件设备, 提供外围器件寄存器的配置和初始化等控制。

语音编解码芯片 AIC23B 的 Mini-drive 函数表结构为: IOM\_Fxns EVMDM642\_EDMA\_-AIC23\_FXNS={&md-BindDev, &mdControlChan, mdCreatChan, &mdDeleteChan, &mdSubmitChan, &mdUnBindDev}。Mini-drive 函数表结构体说明: mdBindDev 函数用于将驱动程序绑定某一器件, mdUnBindDev 函数用于将驱动程序脱离绑定某一器件, mdCreatChan 函数用于创建一个数据通道, mdDeleteChan 用于删除一个数据通道, mdControlChan 用于控制一个数据通道。

### 3 SMV 算法移植

SMV 算法首先在 VC6.0 中仿真成功, 但是要移植到 DSP 上, 需要针对 DM642 的内核结构和平台配置做具体的适应性修改<sup>[3-4]</sup>。

#### 3.1 AIC23B 的控制寄存器设置

AIC23B 的驱动程序的入口必须加入到 DSP/BIOS 的配置文件中, 以建立数据管理联系。在 DSP/BIOS 中注册 IOM\_Mini-drive, 首先在 Device Drive 的 User-Define Devices 中创建设备对象 UDEV, 然后修改设备对象属性。配置如下:

- (1) Init function: SEEDVPM642\_EDMA\_-AIC23\_init, 初始化函数;
- (2) Function table ptr: SEEDVPM642\_ED-MA\_AIC23\_Fxns, 此表映射对象为 AIC23B;
- (3) Function table type: IOM\_Fxns, 设备驱动类型;
- (4) Device id: McASP 对应一块 AIC23B, 设置 id 为 0, 将会传递给 mdBindDev 的参量 devid;
- (5) Device params ptr: AIC23B 参数的入口指针默认为 0x00000000, 传递给 mdBindDev 的参量 devparams;
- (6) Device global data ptr: 指向 Mini-driver 的全局数据地址, 传递给 mdBindDev 的参量 devp, 其默认值为 0x00000000。

AIC23B 参数的设置采用默认结构函数 EVMDM642\_EDMA\_AIC23\_DEFAULT\_DEVP-ARAMS 进行配置, 此结构体仅有成员函数 AIC23\_DEFAULTPARAMS\_EVMDM642 需要修改。此函数也为结构体, 包含了 AIC23B 的 Register0~9 十个控制寄存器, 修改 Register7 为 16 bit 采样精度和 Register8 为 8 kHz 的采样频率。因为仅修改头文件中的宏达不到修改目的, 因此必须将驱动的头文件(.h)和执行文件(.c)一起在 CCS 中编译生成归档文件(.l64), 同时也需要设置 McASP 为从工作模式, 第一路音频的 AIC23B 为主模式, 其他三路为从模式。然后在 CMD 文件中调用归档文件(c6x1x\_edma\_mcasp.l64 和 seedvpm642\_edma\_aic23.l64)即可。

#### 3.2 SIO 流单任务模型

DSP/BIOS II 提供了处理数据流的两种基本结构: 流

## 网络与通信 Network and Communication

I/O(SIO)和数据管道(PIP)。二者都可以提供可靠确定的方法处理音频数据,且与通道数量无关。本文选择 SIO 流和 DIO 适配器组合模式,同时采用 SIO 的发放/回收流传输模型(SIO\_issuse 和 SIO\_reclaim 分别用于发放和回收)。函数 SIO\_issuse 用来传递缓冲区给流,但不返回任何缓冲区,且流在不产生阻塞时将控制权交给应用程序;函数 SIO\_reclaim 用来请求流返回一个缓冲区,如果没有可返回的缓冲区,流就会阻塞任务的执行直到有可返回的缓冲区或者超时。发放/回收模型不会自动创建缓冲区,需要通过函数 MEM\_alloc 来分配,每一个数据流都声明了 3 个缓冲区。这种乒乓缓冲结构能适应大数据量传送,接收和发送缓冲区不是数据拷贝,而是缓冲区指针的交换。4 路音频采集通道首先形成 4 路不同的 SIO 输入流,所有数据都通过 SIO 进入任务处理程序 TSK;数据得到 TSK 的处理,然后经由 4 路不同的 SIO 输出流输送到音频输出硬件。SIO 的单任务模型如图 3 所示。

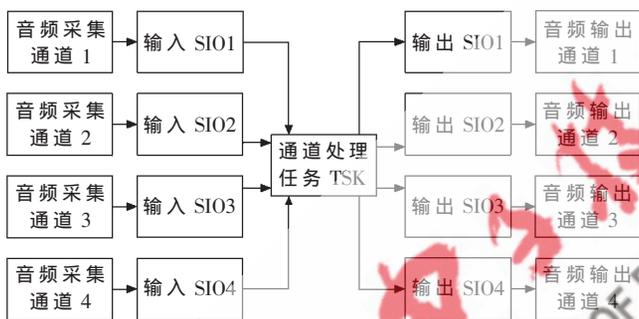


图 3 SIO 的单任务模型

### 4 SMV 算法程序的优化

SMV 算法的 C 语言程序来源于 3GRP2 官方网站,采用最新版本 V3.9 的定点代码<sup>[5-6]</sup>。SMV 算法复杂且计算量巨大,增加了实时应用的硬件成本。SMV 代码优化主要考虑运行速度、程序占用空间和数据占用空间,但是运行速度与程序占用空间、数据占用空间相矛盾。故有必要针对 DM642 处理器进行代码优化,采取一定策略来折中运行速度和硬件资源。

#### 4.1 编译器的优化选项和内联函数

在 DSP 处理器性能不断提高的同时,编译器的优化效率也获得了很大的进步。首先利用软件 CCS3.1 的编译选项进行优化,采用编译选项 -O3 和 -pm(文件级和程序级优化)有很好的优化效果。C6000 编译器提供的内联函数可以快速优化 C 代码,调用时必须包含头文件 intrinsic.h。内联函数实际上为 DSP 的特殊函数,是用汇编语言编写的一些常用的基本运算。在源代码的基本运算函数前加下划线就表示可以直接内联为 C6000 的函数,尽可能多地令语句使用内联函数,从而极大地减小执行时间。

#### 4.2 优化循环体

DM642 具有 8 个并行运算单元,非常有利于循环体

的优化。循环体优化的思路是将循环体内部的运算放到循环体外部执行。由于编译器不能自动分解循环体,如果充分利用 DM642 内部的指令缓存,就要主动分解为小循环。多循环比较复杂,且在优化器优化时只能在最内层循环中形成一级流水线,无法充分利用 C6000 的多级流水线,但是将其拆成单层循环后,优化就可以按照单层循环来执行。循环体的优化还要充分利用数据打包技术,但是数据地址要对齐 32 bits。

#### 4.3 流水线技术

流水线把指令分为预取址、取址、译码、取数、执行和缓冲回写 6 个操作阶段。每条指令的不同操作阶段可以同时进行,在相同时钟频率下,采用流水线作业减少了每条指令的执行时间,提高了处理速度。软件流水将循环体内部指令进行部分重叠(也是对循环的展开)。直接的循环体展开能减少循环开销,而软件流水最主要的优势是产生代码量小、能减少循环执行时间,实际中常将这两种优化措施结合使用。流水线原理如图 4 所示,在第六个指令周期中,有 6 条指令同时执行。



图 4 DSP 的六级流水线操作

#### 4.4 填充延迟时隙

DSP 执行一些指令时需要一定的延迟时隙才能生效。由于这个原因,程序如果都按照常规执行,中间就会有很多的延迟时隙。如果可以在延迟时隙时也执行操作,就可以节约很多的时间。填充延迟时隙是解决这个问题的方法。

例图 5(a)中,执行 LDH 指令后有 4 个延迟时隙,可以把后面的 2 条指令 SUB.L1 B0, 1, B0 和 [B0] B.S1 LOOP 放在指令 LDH 后面,这样仅在 NOP 的两个周期之

<pre> MVK S1B0, B0 LOOP: LDH D1*AS++, A2 LDH D1*AS++, A3 NOP4 MPY M1x A2, A3, A4 NOP ADD L1 A4, A6, A6 SUB L1 B0, 1, B0 [B0] B S1 LOOP NOP5 STH D1A6, *A7 </pre>	<pre> LOOP: LDH D1*AS++, A2       LDH D1*AS++, A3       SUB L1 B0, 1, B0       [B0] B S1 LOOP       NOP2       MPY M1x A2, B3, A4       NOP       ADD L1 A4, A6, A6 </pre>
--	--

(a) 汇编源代码

(b) 修改后的代码

图 5 汇编源代码和修改后的代码

## 网络与通信 Network and Communication

后就可以执行乘法运算。由于跳转指令提前,它的延迟时隙中也在执行操作。这样执行循环的周期将大大减少,程序如图5(b),符号||表示并行执行。需要注意的是,填充延迟时隙一般需要改变指令执行的顺序,在调整时需要特别注意不能改变程序本身的功能。经过填充延迟时隙后,完成一次循环的时间减少到8个周期。

### 4.5 用指针移动代替数组更新

SMV算法中有很多有限冲激响应滤波器(FIR)运算,如果对其优化可以起到很好的效果,下面以10阶FIR为例:

$$h_{out}(n) = h_{in}(n) + \sum_{k=1}^{10} a_k h_{in}(n-k) \quad (1)$$

每次循环时,FIR滤波器要用新的输入值更新,源代码通过数组的更新来实现。以FIR滤波器为例,用10阶数组Fir\_a[10]存放滤波器的输出,每计算一次就要将该数组的前9个元素向后移一位,Fir\_a[0]存入当前输入值,共需要更新9次。现在将数组Fir\_a[10]加大到Fir\_a[70],设一个指针\*Fi\_a指向该数组,初始位置指向Fir\_a[60],用指针的向前滑动代替数组的右移更新。这样信号每通过一次滤波器就可以减少 $60 \times 9 = 540$ 次数据搬运。在SMV算法中,存在着大量的滤波、卷积算法,可采用同样的方法解决滤波器数组值更新的问题。

### 4.6 SMV算法运算量优化前后的对比

变速率语音编解码算法SMV经过C语言和汇编语言的优化、DSP/BIOS配置,已经成功移植到SEED-VPM642平台,实现了语音的变速率压缩和解压处理。经过优化之后,SMV程序的运行效率有所提高。优化效果如表1、表2所示。

表1 编码 enc\_smv\_frame\_fx 函数的运算量

MODE	最大加权运算量(WMPOS)		平均加权运算量(WMPOS)	
	优化前	优化后	优化前	优化后
0	31.29	27.35	23.84	19.03
1	31.63	27.50	22.51	18.14
2	31.27	27.12	22.41	18.03
3	32.16	27.89	22.45	17.85
4	31.91	28.13	24.09	19.19
5	31.93	28.15	22.72	18.23

表2 解码 dec\_smv\_frame\_fx 函数的运算量

MODE	最大加权运算量(WMPOS)		平均加权运算量(WMPOS)	
	优化前	优化后	优化前	优化后
0	3.92	2.73	3.18	2.12
1	5.02	3.92	3.46	2.22
2	5.05	3.95	3.66	2.41
3	5.05	3.98	3.73	2.54
4	5.05	3.90	3.80	2.72
5	5.05	3.91	3.81	2.70

编码部分配置的程序空间和数据空间分别为96KB和16KB;解码部分配置的程序空间和数据空间分别为32KB和16KB。

SMV算法运行在DM642平台时,其语音分析延时20ms、未来帧延时10ms、码本搜索延时3ms和信号修正累计延时2ms,故总延时35ms。在六种模式下,四通道的语音质量良好,且具有很好的噪声抑制能力。SMV算法处理带噪语音情况如图6所示。

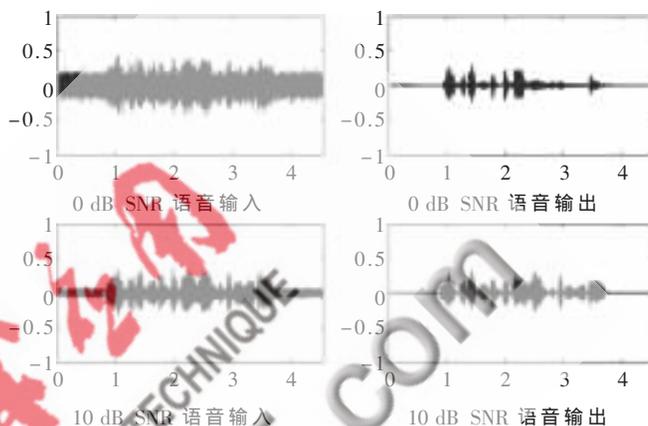


图6 0 dB和10 dB带噪语音处理前后对比

### 参考文献

- [1] GAO Yao, EYAL S, ADIL B, et al. The SMV algorithm selected by TTA and 3GPP2 for CDMA applications[S]. IEEE, Proc ICASSP-2001: 709-712.
- [2] 王炳锡,王洪.变速率语音编码[M].西安:西安电子科技大学出版社,2004.
- [3] 张太镒,任宏.基于TI DSP的通用算法的实现[M].北京:电子工业出版社,2006.
- [4] ERH W H, CYRIL S K, ANDREW T R, et al. Performance analysis of digital signal processors using SMV benchmark[J]. International Journal of Signal Processing, 2009.
- [5] 日本无线工业及商贸联合会. ARIB STD-T64-C.S0030-0 V2.0[EB/OL]. 2001.
- [6] 日本无线工业及商贸联合会. Software Distribution V3.5 for C.0030-0 V2.0 SMV Specification[EB/OL]. 2001.

(收稿日期:2009-12-19)

### 作者简介:

王小兵,男,1983年生,硕士研究生,主要研究方向:语音信号处理。

黄冰,男,1946年生,教授,主要研究方向:信号处理、计算机网络。

胡巍,男,1982年生,硕士研究生,主要研究方向:图像处理。