

一种 AMR 语音编码器的 VLSI 设计及 FPGA 实现

郑海东, 王明江, 王进祥, 崔浩林, 商 迪

(哈尔滨工业大学 微电子中心, 黑龙江 哈尔滨 150001)

摘 要: AMR(自适应多速率)语音编码标准由于其低码率 and 高质量, 在通信和多媒体领域得到广泛应用. 针对 AMR 语音编码标准的算法特点, 提出了一种“音频 DSP 软核 + 硬件加速器”的 VLSI 实现结构. 这种结构能够有效地实现编码算法, 同时达到低成本、低功耗的要求. 综合后的电路在满足语音编码实时性的要求下能工作在 50MHz 频率以下, 功耗仅有不到 50mW, 和 78 k 门的芯片面积. 最后通过 FPGA 整体验证, 证明这种方案是可行有效的. 同时优化后的音频 DSP 软核和硬件加速器中的模块复用设计, 使得本设计方案对 G.7xx 系列编码算法具有通用和可移植性.

关键词: AMR; VLSI; DSP 软核; 硬件加速器; FPGA

中图分类号: TN4

文献标识码: A

文章编号: 1000-7180(2010)02-0017-05

VLSI Design and FPGA Implementation of AMR Speech Encoder

ZHENG Hai-dong, WANG Ming-jiang, WANG Jin-xiang, CUI Hao-lin, SHANG Di

(Microelectronics Centre, Harbin Institute of Technology, Harbin 150001, China)

Abstract: AMR(Adaptive Multi Rate)speech encoder have application in communication and multi media field widely for lower rate and higher quality. This paper presents a VLSI structure of “Audio DSP soft core + Hardware accelerator” based on AMR speech coding algorithm. This structure can complete encoder algorithm with meeting the demand of low cost and low power. The circuit can work steadily at 50MHz, and only have 50mW power and 78k chip area. This design was implemented on FPGA device, and the experimental results indicate that the structure is feasible. Meanwhile, as the design in this structure can also be applied in many other speech coding algorithms like G.7XX serial.

Key words: AMR; VLSI; DSP soft core; hardware accelerator; FPGA

1 引言

AMR(Adaptive Multi Rate)是 3GPP 颁布的第三代移动通信语音编码标准, 目前应用在中国 3G 标准 TD-SCDMA 中^[1]. AMR 及其改进方案(AMR-WB)随后被推广到 VoIP 和因特网、第 3 代移动通信、PSTN 高质量音频会议、音频流、ISDN 宽带电话、ISDN 可视电话和电视会议等众多语音通信领域^[2].

AMR 自适应多速率编码器支持八种速率: 12.2kb/s、10.2kb/s、7.95kb/s、7.40kb/s、6.70kb/s、5.90kb/s、5.15kb/s 和 4.75kb/s^[3]. 与传统窄带语音编码标准 G.729 相比, 算法复杂度和运算量有较

大增加, 编码器所需的运算量为 G.729 编码器的 3~5 倍. 流行的语音编码技术通常采用 DSP 编码实现^[4], 由于 AMR 的编码算法复杂度是 G.729 的 3~5 倍, 采用普通 DSP 实现比较困难, 且采用一个高性能的 DSP 实现, 会大大增加产品的成本. 对于移动通信、ISDN 宽带电话、可视电话等通讯领域, 提供的产品往往是终端产品, 因此提供低成本、低功耗、高性能的 AMR 编码方案具有非常大的意义.

2 AMR 编码的总体架构设计

2.1 AMR 语音算法及性能分析

AMR 主要采用代数编码激励线性预测(ACELP)技术, 功能包括九大部分: 预处理、线性预

测(LPC)分析和量化、开环基音分析、脉冲响应的计算、目标响应的计算、自适应码本搜索(闭环基音分析)和增益控制、代数码本的结构和搜索、自适应码本增益和固定码本增益的量化、修改存储器. 其中 LPC 分析完成的主要功能是获得 10 阶 LPC 滤波器的 10 个系数,并将它们转化为线谱对参数 LSF,以及对 LSF 进行量化;基音搜索包括了开环基音分析和闭环基音分析两部分,以获得基音延迟和基音增益这两个参数;代数码本搜索则是为了获得代数码本索引和代数码本增益,还包括了对码本增益的量化

化^[5]. 由于 3GPP 组织提供了标准未优化的 C 代码,所以可以对 AMR 标准算法进行模拟以得到其性能参数,模拟环境都是在 LINUX 环境,用 GCC 编译. 表 1 列出了算法中主要模块的数据(由于篇幅关系只列出 12.2kb/s 模式的数据,编码总 WMOPS 为 14.27). 由表 1 可知,整个编码算法复杂度在 15WMOPS 以下. 其中 LPC 分析,基音搜索和码本搜索的运算量占了整个编码运算的很大部分,表 2 是对主要模块运算量比例的统计.

表 1 12.2kb/s 模式主要模块编码算法复杂度 (WMOPS)

LPC 分析模块	Levinson	Az_lsp	Q_plsf_5	Autocorr	各功能块合计	
复杂度	0.25	0.92	1.35	0.46	2.98	
开环基音搜索模块	Weight_ai	Syn_filt	Residu	Pitch_ol		
复杂度	0.07	0.19	0.18	2.01	2.45	
闭环基音搜索模块	Pred_lt_6	Convolve	Residu	Syn_filt	Pitch_fr_6	
复杂度	0.25	0.44	0.27	0.53	1.26	2.75
固定码本搜索模块	Syn_filt	Cor_h_x	Cor_h	Search_10i40		
复杂度	0.21	0.48	1.28	2.73	4.70	

表 2 主要模块算法复杂度比例

编码模块	详细算法	比例/%
线性预测	LPC 分析	20.9
基音搜索	开环基音分析	17.2
	闭环基音分析	19.3
码本搜索	固定码本搜索	32.9
总计		90.3

由表 2 可知三个主要模块占了整个编码算法的 90.3%,如果能对此部分进行优化或硬件实现,将大大提升编码速度.

2.2 AMR 语音编码器体系结构

编码器体系结构的提出依赖于表 1 和表 2 中能分析得出来的数据. 首先编码器的算法复杂度低于 15WMOPS,因此可以选用一个工作频率较低的 DSP 软核来实现,并根据 AMR 标准的精度要求对 DSP 软核进行优化. 其次由于 DSP 软核的低频和 AMR 的实时性要求可以考虑用硬件加速器的方法来加速运算比例最大的三个模块,这样速度就可以提高将近 70%. 目的是要实现高性能、低成本、低功耗,因此最后的体系结构是一个“优化音频 DSP 软核+硬件加速器”的方案. 图 1 就是编码器的体系结构.

在图 1 中,Input control 模块主要负责输入的同步和缓存以及语音输入模式的选择(RECODER/PLAY),Output control 模块主要负责编码数据的打

包成帧和传输. MEMORY 分为两部分, RAM 和 ROM. 其中 RAM 用来做数据读写缓存区, ROM 用来做语音编码算法的常数表. 语音编码算法的运算量较小部分由 DSP 软核通过读取 ROM 里的汇编程序实现,而基音搜索模块和码本搜索模块等由专门的硬件加速器通过 RTL 的形式实现.

3 AMR 语音编码的主要模块设计

根据图 1 中的总体结构,模块设计主要分为三部分. 第一部分为 DSP 软核部分的指令扩展及优化设计,特别是把一个通用的 DSP 优化成满足 AMR 标准数据位宽和精度,同时又有低功耗少面积的专用音频 DSP 软核. 第二部分针对音频 DSP 软核的算法汇编实现及其调整优化. 第三部分是基音搜索和码本搜索硬件加速器的 RTL 实现.

3.1 DSP 软核的指令扩展及优化

文中的 DSP 软核是一个通用 32 位的 DSP,5 级流水,最高工作频率是 110MHz. 由于 AMR 语音编码的标准 C 程序中有很多自定义的运算函数,使用频率非常高. 因此如果用通用的 DSP 来运行程序的时候占用指令周期比较多,汇编程序的实现也比较复杂,同时由于数据处理时的溢出和饱和等因素,导致最后结果的精度不够和对 C 程序实现效果逼近不够好. 基于上述的三个原因,对 DSP 进行指令扩展及优化. 表 3 是 AMR 部分中自定义的使用频繁

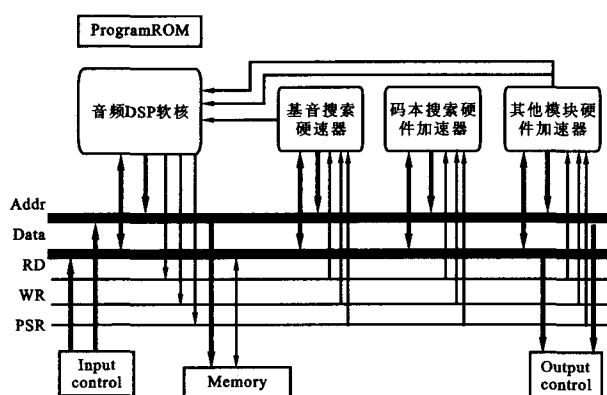


图1 AMR 编码器体系结构

的运算函数及其权重(数字表示),同时也是要增加的指令.这里的权重是结合实现流程把每个函数转换为基本运算的等效指令条数.在这些函数中,一半左右权重是大于2,其中个别如 `norm_s` 需要15个基本运算指令实现,而 `div_s` 和 `norm_1` 则分别需要18个和30个指令.在编码算法中,这些运算函数作为基本运算单元被频繁调用,因此通过硬件设计增加的同样功能的指令只需一个指令一个周期就能完成运算,这样不仅使汇编代码变得简洁,同时也大大提高了程序运行的速度.为了节省面积,这里指令扩展必须尽量复用原有的一些基本模块,同时要调整流水线控制器和译码结构等模块.

表3 部分常用运算函数及其权重

<code>add</code>	1	<code>L_add</code>	2	<code>L_shift_r</code>	3
<code>sub</code>	1	<code>L_sub</code>	2	<code>L_abs</code>	3
<code>abs_s</code>	1	<code>L_add_c</code>	2	<code>L_sat</code>	4
<code>shl</code>	1	<code>L_sub_c</code>	2	<code>norm_s</code>	15
<code>shr</code>	1	<code>L_negate</code>	2	<code>div_s</code>	18
<code>extract_h</code>	1	<code>L_shl</code>	2	<code>norm_1</code>	30
<code>extract_l</code>	1	<code>L_shr</code>	2	<code>L_extract</code>	3
<code>mult</code>	1	<code>mult_r</code>	2	<code>Div_32</code>	43
<code>L_mult</code>	1	<code>shr_r</code>	2	<code>Mpy_32</code>	5
<code>negate</code>	1	<code>shift_r</code>	2	<code>Mpy_32_16</code>	3

3.2 AMR 编码的汇编实现及优化

由于选用的 DSP 工作频率比较低,所以为了使得编码算法能更好地运行来满足实行性,这里选择用手工编写纯汇编的方式.对于选择哪些模块用汇编实现,本设计参考表1和表2中的数据来得出结论.对于预处理、VAD、帧相关处理、后处理等复杂度所占比例比较小和基音搜索、码本搜索中运算量比较小的子模块部分,这些部分都可以选择用 DSP 汇编来实现.汇编编写的步骤分为两部分:首先是数据空间的分配.数据分为常量和变量,变量又分为全局变量和局部变量,在分配空间的时候要分开用不

同方法处理.其次是各个函数和功能模块汇编的实现,为了能与 C 代码容易对比和调试,这里使用了与 C 同样的文件名和函数名.

汇编编写完之后必须对其进行优化以得到最好的运行速度和最简洁程序.汇编的优化需很好的理解本设计中有的 DSP 软核结构和指令特性.文中针对优化的 DSP 使用了以下几种优化方案:

① 循环的展开:在循环次数比较多的函数里展开,以此来减少循环判断和跳转.

② 循环的合并:将多个没有数据关联的循环合并,这样既可以减少代码量,同时也可以减少循环判断和跳转.

③ 使用块操作指令:编码算法中很多用循环的方法来操作多个连续的存储空间,用 DSP 特有的块操作指令可以减少指令数.

④ 使用优化的指令:这里包括用表3中的新增指令来完成原来算法中定义的运算函数,用 `mveb[x]()`, `mvib[x]`(将判断和移动指令结合)赋值和移动,等等诸如上述的方法.

3.3 硬件加速器模块的设计

在硬件设计时,针对算法需要对硬件实现单元进行合理的资源调度和分配,合理使用 ASAP(as soon as possible)和 ALAP(as last as possible)两种调度方法.得到性能和面积的满足需要,设计硬件加速器必须遵守一个原则:算法核心的瓶颈部分要用硬件加速,而对一些运算部分硬件实现提高不多的要用汇编实现,尽量减少面积.因此下面的设计方案都是在用 DSP 和汇编仿真之后,找出算法的计算瓶颈处来硬件加速,并不是整个模块都用硬件实现.

基音搜索主要分两部分:开环和闭环.由于开环部分算法比较简单,计算量比较小,通过汇编实现既可,用硬件实现速度提高不明显而面积上得不偿失.

式(1)是闭环搜索的主要算法核心公式部分,其中 k 的范围是在 18~143 之间,再加上每个不同 k 值是要计算 40 次彻底自相关和互相关,其汇编实现时计算量和计算周期是非常大的.

$$R(k) = \frac{\sum_{n=0}^{39} x(n)y_k(n)}{\sqrt{\sum_{n=0}^{39} y_k(n)y_k(n)}},$$

$$y_k(n) = y_{k-1}(n) + u(-k)h(n) \quad (1)$$

针对算法特点,设计了图2所示的硬件结构,采用3级流水,32位乘法器通过改进 booth 编码和 wallace 树结构实现,32位加法器采用超前进位结构. SRAM 是图1中的 MEMORY 中的 RAM,由于

编码算法中的存在很多的自相关运算,如果从一个 SRAM1 中读取两个相同的数据需要两个周期.在这里综合考虑速度和面积,用增加一个小的 SRAM2,当做自相关运算时用输入缓存控制模块把相同数据复制一份到 SRAM2,这样就可以只用一个周期完成输入,在计算量大的时候效果非常明显.

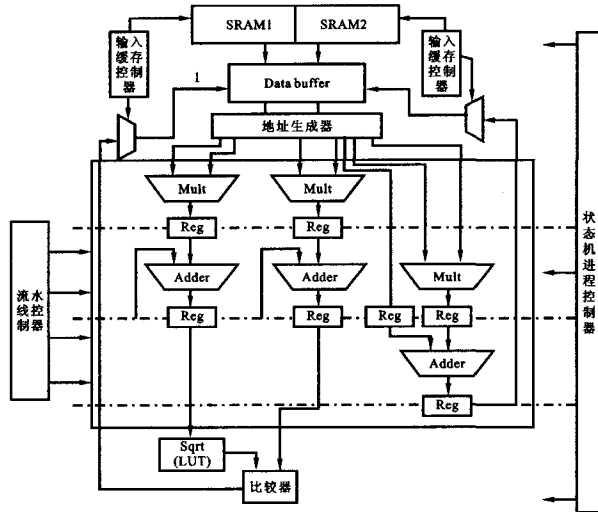


图2 基音搜索硬件加速器结构

为以后的复用乘法器节省面积考虑,把 MAC 运算分成乘和加两部分,用流水方式实现.式(1)中分子自相关和分母互相关分别用两个乘法器来并行计算, $y(n)$ 的更新是通过前一个周期数据的迭代来完成,因此在流水线上要比前两项计算慢一个周期.开方用查表 ROM 的方式实现,速度快但是面积较大.由于开方运算在整个算法中用到比较多,可以用分时复用于多个硬件加速器模块的方法来减小其带来的面积影响.除法运算在这里用比较分子分母的大小及两则差值的方法间接实现. Data Buffer 中暂存相关待操作数,操作中间数据以及操作结果,其数据的读写由输入缓存控制器来控制.

码本搜索的主要算法核心公式如式(2)所示,具体的硬件结构如图3所示.

$$\begin{cases} x_2(n) = x(n) - \hat{g}_k y(n), n = 0, \dots, 39 \\ y(n) = v(n) * h(n) \\ A_k = \frac{(C_k)^2}{E_{Dk}} = \frac{(d'c'_k)^2}{c'_k \Phi c_k} \end{cases} \quad (2)$$

码本搜索与基音搜索是在不同时刻启动的模块,因此为了节省面积需要对两个模块的基本模块进行复用.码本搜索硬件加速器的基本功能模块乘法器、加法器、开方查表运算、比较器与基音搜索硬件加速器复用.由于各模块之间的数据关联性,这里用了3个乘法器和5级流水的方式实现.这里有3

个加法器在3个流水级,其数据输入都不同,因此实际实现是用一个加法器分时复用3个周期.5个流水级中的前3个跟基音搜索模块中的流水复用, Data buffer 也互相共用,剩余的设计也尽量通用化以备复用.当然由于复用,必须在这些模块中增加组合逻辑来进行复用控制,这也是复用带来的逻辑复杂度和面积牺牲.

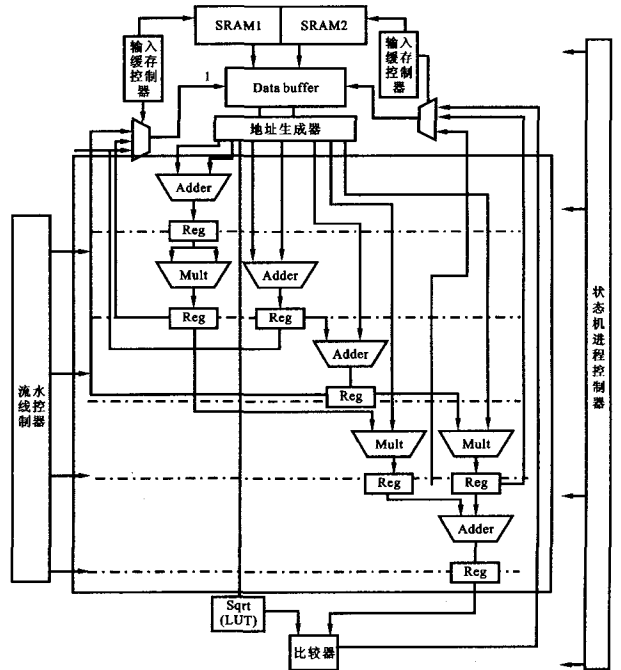


图3 码本搜索硬件加速器结构

利用其中的基本模块又设计了四个类似的加速器,分别是互相关模块、开环搜索自相关模块、矢量量化量化相关模块和矩阵量化相关模块.最后利用这些硬件加速器相似的结构和相同的数据通路设计了一种复用结构来获得最小的面积.如图4所示,其中的 Datapath 模块由于在各个加速器里是相同的

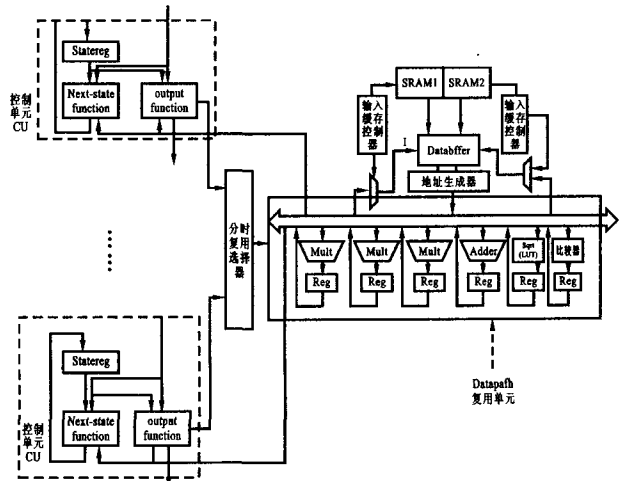


图4 硬件加速器复用结构

所以复用,每次分时复用只需切换不同的状态控制单元(CU)即可,从而达到节省面积目的^[6].

由于设计中有 3 个乘法器并行,也就是有 3 条并行处理线,再加上对其数据读取上做两个 SRAM 的改进,以及硬件结构在做循环算法时的可知性,从而减少指令比较判断周期.因此硬件加速器能使这些模块速度提高 3 倍以上,相对于汇编代码减少了将近 70% 的指令周期.而码本搜索和基音搜索占整个算法 70% 左右的计算量,因此对整体运算速度能提高最少 1 倍左右.

4 实验结果及分析

文中 DC 综合用的工艺库是和舰 0.18 μm 工艺库(10 μm^2 约为 1 门),FPGA 验证板用的是 Xilinx

的 spartan XC3000 芯片.测试序列是由 3GPP 提供的标准测试序列,以此来对对比标准.对 DSP 优化前和优化后 DC 综合的面积结果分别为 8049k 门和 8594k 门.因此在优化以后仅仅增加了 545 门,同时 DSP 能稳定地运行在 110MHz 频率上,设计达到了优化的目的.汇编程序未优化时在音频 DSP 软核上功能仿真结果与测试序列标准对比正确.汇编优化后时序满足,编码器能工作在 100MHz 的频率上,但功耗将近 100mW,这不满足本设计低功耗的要求.最后再加上硬件加速器之后,系统能稳定工作在 50MHz,时序分析时 slack 为 5.15,功耗降到 50mW,面积在 78k 门左右,性能和成本基本达到设计要求.表 4 是最后 DC 综合后的结构.

表 4 AMR 编码器 DC 综合结果 (μm^2)

	音频 DSP 软核	硬件 加速器	片上存储器				其他 模块	
			Sram1	Sram2	程序 ROM	常数 ROM		I/O buffer
面积	85946	51328	319313	123258	107234	76417	15665	15318
总计	784488(约 78k 门)							

语音编码如果不受回波影响且有好的质量其每帧算法延迟要小于 20ms.在本实验中首先输入两帧简单重复帧,然后输入测试帧,其仿真结果如图 5 所示:第三行波形为 AMR_data_in,第六行波形为编码后写到存储器 AMR_wdata.两者波形延迟 9.5ms,这就是说每帧算法延迟为 9.5ms,不仅满足实时性要求,而且还有可以进一步降低工作频率的空间.最后为了使设计更有说服力,把整个编码器 DEMO 到 FPGA 上验证,语音质量的 MOS 评分在 3.7 左右,满足了设计要求.

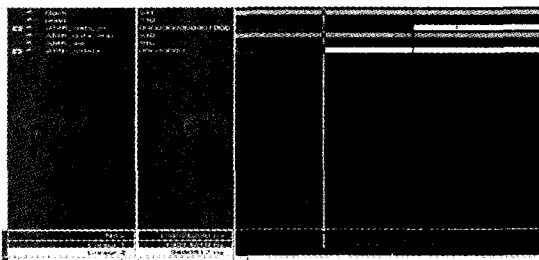


图 5 输入输出仿真图

最后,由本设计的综合结果可知片上存储器在整个设计中占了 80%,因此设计还有很大的优化空间.由于本设计语音编码实时性余量还很大,理想情况下可以将 DSP 工作在 25MHz 左右频率下.通过 DMA 或者 CACHE 的方式将片上存储器移到片外由 FLASH 和 SDRAM 实现,通过进一步优化数据

读写和硬件加速器,可以使面积减少到 25K 门以下,具有更低的功耗同时能满足编码的实时性要求和可懂度高的语音质量.

5 结束语

本设计严格按照数字 IC 前端设计的流程来设计 VLSI,然后针对各个具体模块进行详细设计和优化,以及不同方法之间的设计结果对比,以此来达到设计的最优.用 DC 综合软件映射到和舰 0.18 μm 工艺库上,并对其作静态时序分析.最后在 FPGA 进行了整个语音编码器的整体验证.实验结果数据证明本设计是可行的,满足了低成本、低功耗、高性能的设计目标.由于本设计中优化的 DSP 指令对 ITU 的其他语音编码标准通用,同时硬件加速器的设计方案也可以复用,因此本设计存在一定的可扩展性和可移植性.

参考文献:

- [1] Huang Tao, Li Xiaowen, Li Shitao. AMR codec mode selection in TD-SCDMA[C]// Cross Layer Design, IW-CLD '07. China: Jinan, International Workshop on Publication, 2007.

(下转第 25 页)

算法利用接收信号的信息较少;另一方面原因,以式方式重构协方差矩阵,会使部分噪声混入信号空间。

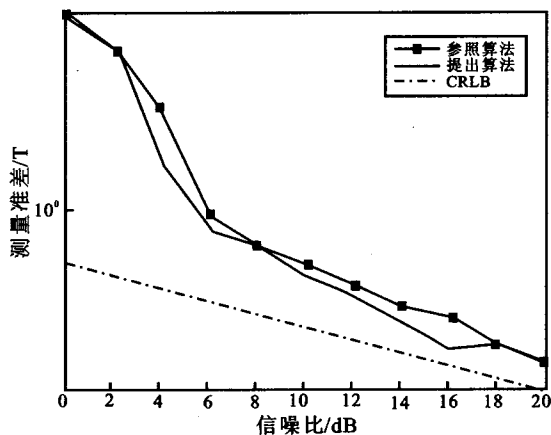


图2 测量标准差与信噪比关系

6 结束语

文中提出一种多径信号到达时延测量的方法。通过将时延测量问题转换成具有低通包络的正弦信号频率估计问题,重构信号矩阵,构造低阶协方差矩阵,然后应用 MUSIC 法解出每条多径信号的到达时延。由于新方法构造的协方差矩阵阶数比文献[3]的低得多,因而运算量比后者小两个数量级。计算机仿真结果表明新方法以稍微降低时延测量性能为代价换取运算量的大幅减小。

参考文献:

- [1] Vanderveen M C, Van der Veen A J, Paulraj A. Estimation of multipath parameters in wireless communications [J]. IEEE Transactions SP, 1998, 46(3): 682 - 690.
- [2] 郭立,王妙峰. 1.6Kb/s 类 MELP 语音压缩编码器的 FPGA 实现[J]. 小型微型计算机系统, 2008(29): 3551 - 6551.
- [3] Yang Jie, Yu Shengsheng, Zhao Mian. The implementation and optimization of AMR speech codec on DSP[J]. Intelligent Signal Processing and Communication Systems, 2007, 28(1): 365 - 367.
- [4] Kyung Jin Byun, Ik Soo Eo, Hee Bum Jeong. Real-time implementation of AMR and AMR-WB using the fixed-point DSP for WCDMA systems[J]. IEEE Tenth International Symposium, 2006(22): 1 - 6.
- [5] 3GPP TS26. 071. Adaptive multi-rate (AMR) speech

- [2] 郭莹,邱天爽. 基于分数低阶矩的 LETDE 时延测量算法[J]. 通信学报, 2006, 27(5): 12 - 17.
- [3] Ge Fengxiang, Shen Dongxu, Peng Yingning. Super-resolution time delay estimation in multipath environments [J]. IEEE Trans. circuits and systems, 2007, 54(9): 1977 - 1986.
- [4] 郭跃,王宏远,陈思捷. 阵列测向中的精确高速并行谱峰搜索算法[J]. 微电子学与计算机, 2007, 24(12): 50 - 54.
- [5] Waz M, Shan T J, Kailath T. Spatiotemporal spectral analysis by eigenstructure methods[J]. IEEE Trans on Acous speec and Signa, 1982, 32(4): 817 - 827.
- [6] 周隰,王宏远. 基于虚拟扩展的超分辨改进 MUSIC 算法及伪峰抑制[J]. 微电子学与计算机, 2007, 24(6): 5 - 7.
- [7] Wax M, Kailath T. Detection of signals by information theoretic criteria[J]. IEEE Trans. On ASSP, 1985, 33(4): 387 - 392.
- [8] Francesco Viola, William F Walker. A spline-based algorithm for continuous time-delay estimation using sampled data[J]. IEEE Transactions on Ultrasonio, 2005, 52(1): 80 - 93.

作者简介:

黎英云 男, (1974-), 硕士研究生, 讲师. 研究方向为数字信号处理、时差测量技术。

马洪 男, (1966-), 教授, 博士生导师. 研究方向为数字信号处理、软件无线电、无源定位技术。

朱光喜 男, (1945-), 教授, 博士生导师. 研究方向为移动通信、多媒体信息处理。

codec: general description[S]. Bruhn, Stefan, 2004.

- [6] 孙剑,郭立,林海涛,等. 先进音频编码中 MDCT 的新型递归结构及其 FPGA 设计[J]. 微电子学与计算机, 2008, 23(4): 61 - 63.

作者简介:

郑海东 男, (1984-), 硕士研究生. 研究方向为数字集成电路设计、语音编码实现技术。

王明江 男, (1969-), 副教授. 研究方向为多媒体芯片体系结构。

王进祥 男, (1968-), 教授. 研究方向为 VLSI/SoC 设计方法及其实现技术。

(上接第 21 页)