

文章编号: 1672-2892(2010)05-0582-07

基于 FPGA 的音频监控系统的设计与实现

王 龙, 陈晓光

(复旦大学 通信科学与工程系, 上海 200433)

摘 要: 广播电台信号的监控是播出安全的重要保障。为了实现对电台音频信号的远程和本地监控, 提出了利用 FPGA 来处理数字音频信号, 并将处理的结果网络化传输的系统方案。文中具体阐述了系统中的音频采集、音频分析、音频切换、网络传输模块和软件的设计, 并给出了系统实物图和实际测试结果。测试表明, 该系统工作可靠, 性能稳定, 减轻了电台听音人员的负担。

关键词: 音频监控; 现场可编程门阵列; 自动切换; 网络传输

中图分类号: TN931.3

文献标识码: A

Designing and implementing audio monitoring system with FPGA platform

WANG Long, CHEN Xiao-guang

(Department of Communication Science and Engineering, Fudan University, Shanghai 200433, China)

Abstract: Radio broadcast signal monitoring is an important guarantee for the safety of broadcasting. A systematic solution to process digital signal based on FPGA platform is presented, aiming for the audio remote and local monitoring. Specially addressed in the paper includes audio signal acquisition, analysis, switching, network transmission modules and software designs. Also, the picture of actual object and test results are given. Test shows that the system is reliable and of stable performance, greatly reducing the burden of the radio listener in the broadcast station.

Key words: audio monitoring; Field-Programmable Gate Array; automatic switching; network transmission

数字信号处理手段多种多样, 而常用方式主要包括: 专用集成电路、数字信号处理器和可编程逻辑器件等。其中, 可编程逻辑器件又以高并行处理性、高可配置性等特点而逐渐受到青睐^[1]。目前, 发展速度最快的是可编程逻辑器件家族中的现场可编程门阵列(FPGA), 正由于此, 研究开发以 FPGA 为处理核心的音频处理系统就显得尤为重要。

本文以嵌入在 FPGA 中的软核处理器 MicroBlaze 为中心, 构建了一套完整的音频采集、分析和传输的嵌入式解决方案。由于采用了可编程逻辑器件, 整个方案具有极大的灵活性, 可以在该平台上进进一步扩展或者修改功能, 而不需要修改硬件平台。相对于其他用 DSP(Digital Signal Processing)或者 ASIC(Application Specific Integrated Circuit)器件实现的系统而言, 该系统具有较高的实时性, 且运行速度快, 克服了这些器件在处理多路音频时存在的并行处理性能差、系统升级和在线配置不灵活等缺点。

1 系统设计

MicroBlaze 是一款由 Xilinx 公司开发的嵌入式处理器软核, 采用 32 位 RISC(Reduced Instruction Set Computer)优化结构, 用以在 FPGA 上开发嵌入式工程。在 FPGA 内部集成 MicroBlaze 软核处理器, 这极大地增加了 FPGA 开发的灵活性和高度可配置性。可以根据设计的需要, 对其进行裁减, 用最少的资源完成设计的需要。它可以采用总线接口 PLB(Processor Local Bus), FSL(Fast Simplex Link)与外设或内存进行数据交互^[2]。

如图 1 所示, 系统采用了嵌入在 FPGA 中的 MicroBlaze 软核处理器作为主控制器, 以总线的形式访问各种外设。主要是以 FSL 总线接收 4 路音频数据; 以 PLB 总线访问 Bram, 接收来自音频分析模块产生的结果, 并将

音频切换模块需要的选择信号写入Bram中,供切换模块访问;并实现PLB总线与网络处理模块的访问,以实现实时音频数据和音频处理结果的网络化传输和显示。

2 硬件设计

2.1 音频接收模块

模拟音频信号经过 A/D 采样器 CS4392 后,产生 Left-justify 格式的数字音频格式。Left-justify 音频格式^[3]如图 2 所示,它和 I²S 总线基本一致,有 3 根信号线:

- 1) 串行时钟 SCLK(Serial Clock),也叫位时钟(BCLK),即对应数字音频的每一位数据,SCLK 都有 1 个脉冲。SCLK 的频率=2×采样频率×采样位数。
- 2) 帧时钟 LRCK(Left Right Channel Clock)(也称 WS),用于切换左右声道的数据。LRCK 为“1”表示正在传输的是左声道的数据,为“0”则表示正在传输的是右声道的数据。LRCK 的频率等于采样频率。
- 3) 串行数据 SDATA(Serial Data),就是用二进制补码表示的音频数据。

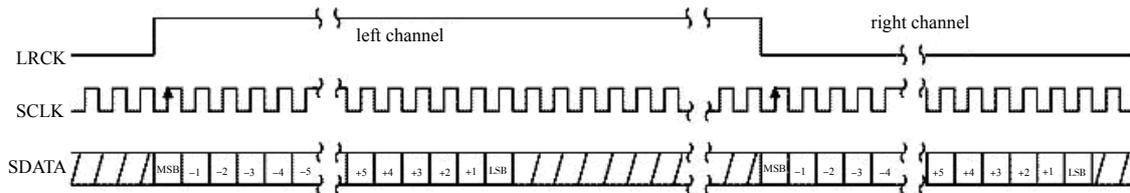


Fig.2 Left-justify audio format
图 2 Left-justify 音频格式

2.2 FSL 接口模块

FSL 总线是一个基于 FIFO(First In First Out)的单向点对点通信总线,主要用于 FPGA 的 2 个模块间进行快速通信。FSL 接口的 I/O 信号如图 3 所示,FSL 总线有一个主设备端和一个从设备端,其结构类似于一个 FIFO,主设备端是 FIFO 的数据写入端,从设备端是 FIFO 的数据读出端。

该接口的主要特点^[4]: 1) 单向的点对点通信; 2) 非共享的无仲裁通信机制; 3) 支持控制位与数据分离的通信; 4) 基于 FIFO 的通信模式; 5) 可配置的数据宽度; 6) 高速的通信性能(独立运行达到 600 MHz)。

FSL 接口模块作为主设备端,向 FSL 总线写数据;MicroBlaze 作为从设备端,以中断方式(当 FSL_M_FULL 为高电平时产生中断)从 FSL 读取数据,读取指令为 put 和 get 指令。

来自音频接收模块的并行数据需要以恒定的速率(48 kHz)存入 FSL,而 MicroBlaze 处理器是在 FSL 总线的 FIFO 数据满时通过响应中断请求来读取数据。这里产生了一个时间裕量的问题,即 MicroBlaze 必须在 1 个时钟周期内(48 kHz)响应中断,否则 FIFO 溢出,出现数据丢失。这对需要响应多路音频信号中断的 MicroBlaze 来说,条件比较苛刻。

FSL 接口模块实现了如下功能:在 FSL 总线接口之前加入一级 FIFO,来自音频接收模块的并行数据先写入该 FIFO。当 FIFO 满时,一次性将该 FIFO 中的数据以系统时钟速率写入到 FSL 总线的 FIFO 中。对 MicroBlaze 来说,响应时间裕量从 1/48 ms 增加到 FIFO 的深度/48 ms。合理设置 FIFO 的深度就能给予处理器足够的时间来响应中断。

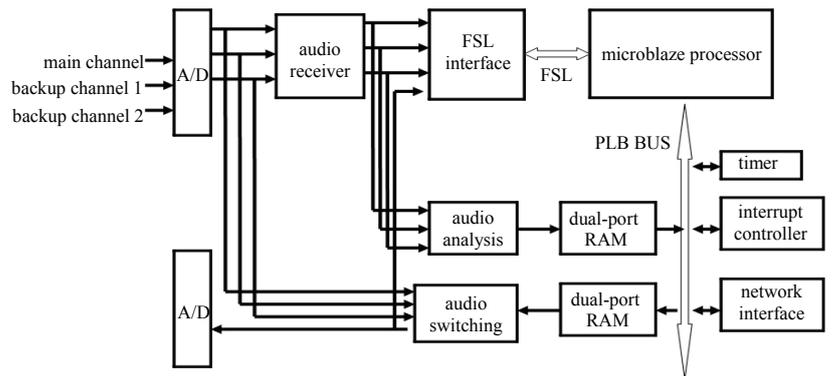


Fig.1 System function block diagram
图 1 系统功能框图

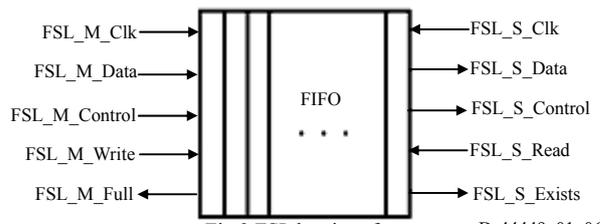


Fig.3 FSL bus interface
图 3 FSL 总线 IP 接口

Ds44449_01_0625

2.3 音频分析模块

对广播电台音频信号的监控,涉及到对音频信号的有无以及输出信号和原输入信号一致性的检测。这就需要音频信号进行处理分析。在本系统中,音频信号的有无通过计算接收信号的 RMS(Root Mean Square)值来判断,而输入输出的一致性主要是通过检测信号的频谱来实现。

2.3.1 RMS 值的计算

信号的 RMS 值实际就是有效值,用一组统计数据的均方根值来表示。对于离散信号,其表达式为:

$$RMS = \sqrt{\frac{x_1^2 + x_2^2 + \dots + x_N^2}{N}} \quad (1)$$

由于开方运算需要消耗较多的逻辑资源,为了便于 FPGA 的实现,将其写成分贝表示的形式:

$$RMS_{dB} = 10 \lg(x_1^2 + x_2^2 + \dots + x_N^2) - 10 \lg N \quad (2)$$

这样 RMS 值的求取关键转化为求取离散数据的平方和,对于这类乘累加运算, FPGA 芯片内部有大量的乘累加资源块,可以用硬件资源来实现。对于这类算法比较简单的数字信号处理, Xilinx 公司针对 FPGA 的实现推出了 system generator 开发环境,可以快速地将算法转换成模块形式,在不降低可靠性的情况下,免去代码编写和调试的繁琐过程,快速实现功能,有效减少开发成本,并缩短开发周期^[5]。

调用 Xilinx blockset 中的 mult 和 accumulator 模块,搭建如图 4 所示的算法模型,实现输入信号平方和的计算。在这里,每次取 512 个采样点进行平方平均,因此每进行 512 次累加以后,累加器需要进行复位,以进行一次运算。输入为 24 位的音频数据,运算后产生 64 位的数据,将 64 位的数据写入到双端口的 Bram 中,微处理器 MicroBlaze 以一定的时间频率访问 Bram 中的数据,并通过式(2)计算出各个通道的 RMS 值。该值反映了该通道信号的幅度,由此可以做出通道信号有无的判断。

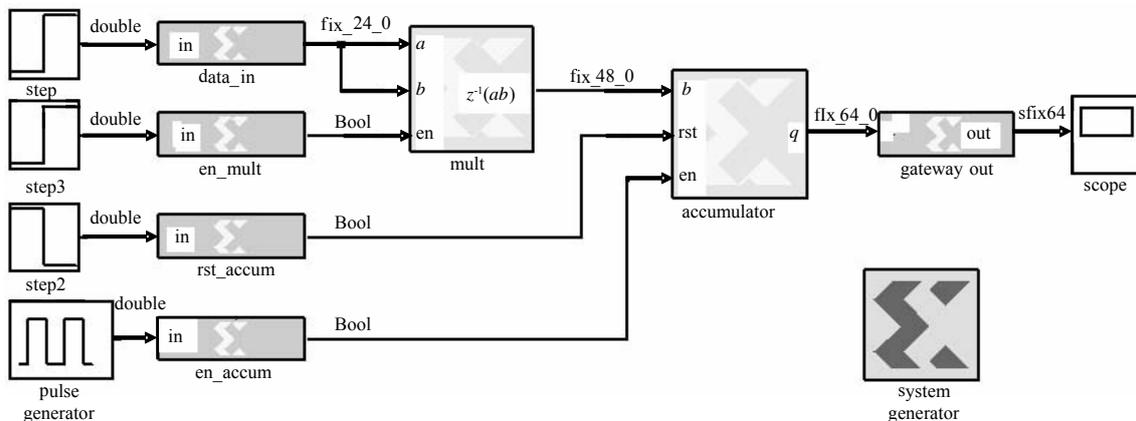


Fig.4 Calculation model for sum of squares
图4 平方和计算模型

2.3.2 信号频谱的计算

傅里叶变换用于将信号由时域转化到频率域,由此可以得到信号的频谱。对于离散信号,其变换为:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N} \quad (3)$$

其基本的算法结构仍然为乘累加单元,其实现方式与上面所讲的 RMS 值的计算方法相似,在此不再重复,只是列出其在 Simulink 下的建模模型,如图 5 所示。

采用模块 FFT v5_0,实现快速傅里叶变换。由于一个 FFT 占用的资源比较多,而且一共需要 8 个 FFT(Fast Fourier Transformation)单元(4 路音频,每路 2 个声道),而 FPGA 上面的乘累加单元有限,故采用 2 种方法来减少该资源的使用:一是点数 N 取得合适,因为只需看出信号频谱的趋势,不需要很高的准确度,取点数 $N=64$,在采样率为 48 kHz 情况下,其频率分辨率为 $48/64=750$ Hz;二是将 4 路音频以时分复用的方式使用 FFT 单元,4 路音频的左声道分时使用 1 个 FFT,右声道分时使用 1 个 FFT,这样在满足功能需求的同时,减少了资源占用率。

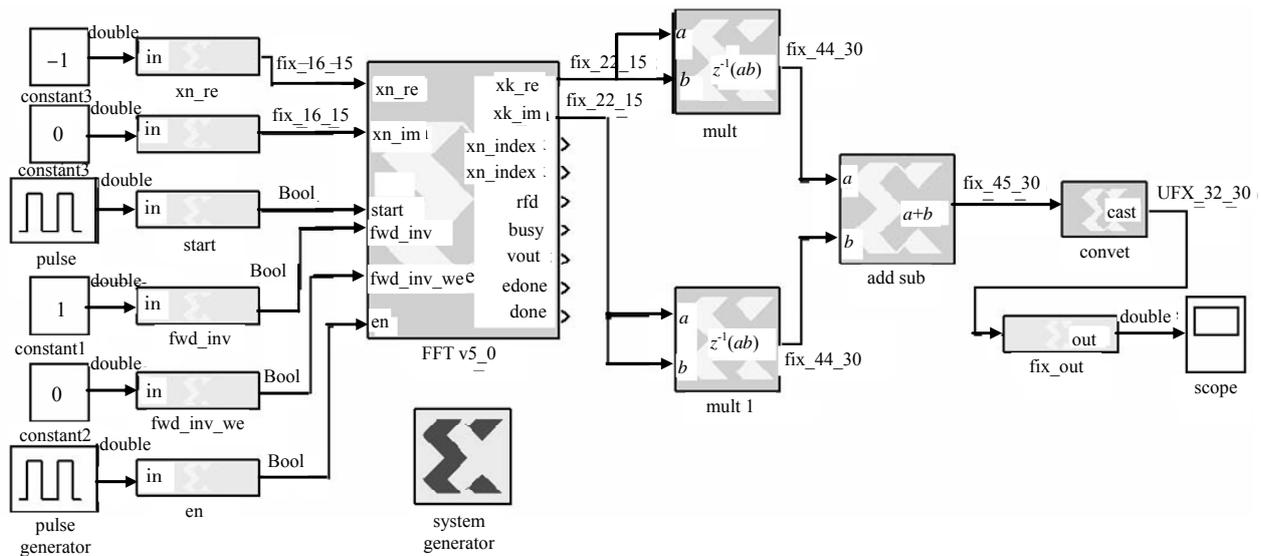


Fig.5 FFT calculation model
图 5 FFT 计算模型

2.4 音频切换模块

有了对音频信号的分析，音频的自动切换则容易实现。音频信号切换模块实际上是一个三选一的选择器，从主路、备用路 1 和备用路 2 中选择一路输出。其中，主路的优先级>备用路 1>备用路 2。默认情况下，选择主路输出，当 MicroBlaze 检测到主路信号的 RMS 值低于某一值(-100 dB，可配置)，并且持续时间超过某一时间(避免因音频中的停顿而误切换)，就认为主路出现故障，选择器选择备用路 1 输出，当备用路 1 出现故障时，选择备用 2 路输出。若优先级高的一路输入恢复时，切换回该路输出。

输出选择信号由 MicroBlaze 计算产生，并在该值改变时写入双端口 Bram，音频切换模块周期性地读入该 Bram 中的选择信号的值，并做相应的切换响应，从而实现故障的检测和自动切换。

2.5 网络发送模块

为了实现音频数据和音频处理结果的网络化传输，需要在系统中实现 TCP/IP 协议，并对音频数据进行 IP 封装。在本系统中，采用了硬件芯片 W5300 来实现音频数据的 IP 化，并采用 UDP(User Datagram Protocol)协议来传输实时音频数据。网络发送模块的功能就是用来实现外围芯片 W5300 和 MicroBlaze 处理器的接口连接。

W5300 是一款高性能的以太网芯片。W5300 支持固件的 TCP/IP 协议；支持 8 个独立端口同时工作；高速网络数据传输，速度可达到 50 Mbps；支持混合网络 TCP/IP 协议栈(软件/硬件 TCP/IP 协议栈)；支持自动极性变化(MDI/MDIX)；支持 8/16 位数据总线等。该器件的高速和高性能，可广泛应用于 IPTV(Internet Protocol Television)、视频监控、可视 IP 电话和大屏幕显示等领域^[6]。

如图 6 所示，FPGA 作为主控制器，需要提供访问 W5300 的接口信号，包括：CS(片选信号)、RD(读命令)、WR(写命令)、INT(中断请求信号)、ADDR[9:0](地址总线)、DATA[15:0](数据总线)。可以发现，其接口比较简单，没有太多的控制信号，因此采用已有的 IP 核来实现，这样更加方便可靠。尽管 Xilinx 公司提供的 IP 核 XPS_MCH_EMC(外围存储控制器)是用来访问外围 SRAM 和 FLASH，但是巧妙地利用它提供的接口信号 Mem_ADDR,Mem_DQ,Mem_WEN,Mem_OEN,Mem_CEN，能够实现对 W5300 的访问。

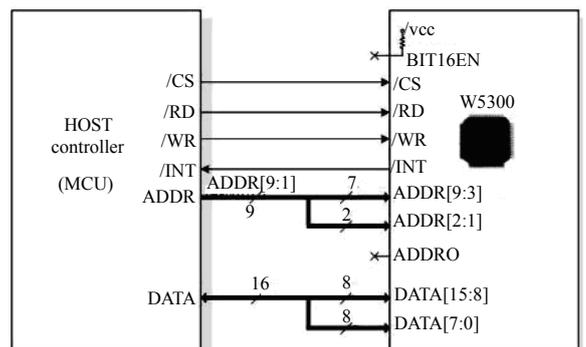


Fig.6 Interface between FPGA and W5300
图 6 FPGA 与 W5300 的连接

2.6 系统硬件实物

实际电路板如图 7 所示。

3 软件设计

在 MicroBlaze 处理器上进行软件开发，其流程和其他的嵌入式处理器相似，一般都采用 C 语言开发。在 EDK(Embedded Development Kit)中，每一个外设 IP 模块都有自己的软件函数库。利用 LibGen 工具，将所需外设函数库的头文件添加到工程中，通过调用这些函数可以操作和控制这些外设^[7]。

首先在主程序中完成初始化，包括定时器、FSL 和 W5300 的初始化。定时器的初始化包括设置中断方式，定时控制字的写入；FSL 的初始化主要为 FSL 总线添加中断；W5300 的初始化包括各个寄存器的配置，如源 IP、端口号、目的 IP、端口号、缓冲器大小、运行协议等。

主程序采用循环等待的方式，流程见图 8。定时器每隔 200 ms 中断一次，中断时，使得变量 en_rms、en_fft 置为 1，从而在主程序中完成一次 4 个通道 rms 的计算、4 个通道 fft 值的读取，结果都写入 W5300，以 TCP/IP 协议传输到网络中，以便于通过软件显示。FSL 总线采用中断方式接收数字音频数据。为了防止 FIFO 的溢出而造成数据的丢失，该中断的优先级高于定时器。在 FSL 中断中，一次性读取 FIFO 中的全部数据，并将结果写入 W5300。最后根据 rms 值的计算结果，根据通道切换的原则，将控制输出的选择字写入双端口 Bram 中。

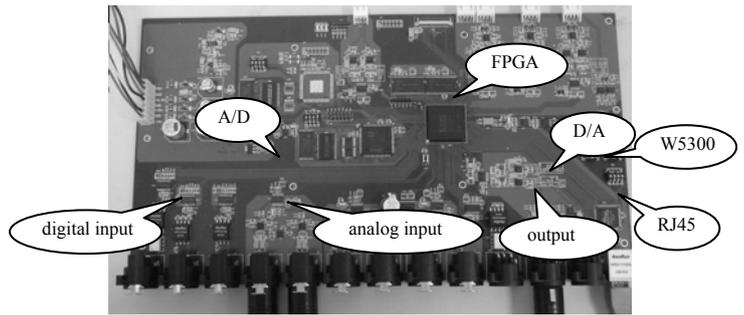


Fig.7 Picture of actual object
图 7 系统实物图

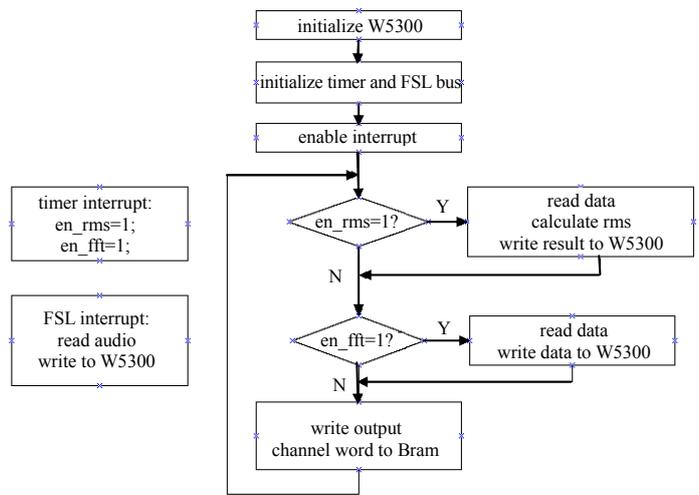


Fig.8 Process flow
图 8 主程序流程

4 系统实现

选用 Xilinx 公司 Spartan 系列的 XC3SD1800A 实现。将经 ISE 综合、实现产生的 bit 文件下载到目标板上，并运行。

RMS							
ML	-25	MR	-25	B1L	-45	B1R	-45
B2L	-112	B2R	-112	0L	-25	0R	-45

Fig.9 RMS values of four channels
图 9 4 路音频 RMS 值

4.1 音频的 RMS 值

在电脑终端，采用 UDP 协议接收来自 W5300 网络接口的 UDP 数据包，根据不同的端口号区分 4 路音频数据，并显示。结果如图 9 所示，当主路输入测试信号为幅度 1 dBu(相当于-25 dB)的正弦波时，计算得到的主路 RMS 值为-25 dB。

4.2 音频的自动切换

如图 10 所示，当主路有信号时，输出为主路；当主路没有信号时，切换到备用路输出，说明切换能够自动进行。

4.3 音频信号的频谱

输入信号分别为 1 kHz,5 kHz,10 kHz 的正弦波时，对应的频谱如图 11 所示。系统的采样率为 48 kHz，FFT 的计算点数为 64，频谱的分辨力为 0.75 kHz。当输入 1 k,5 k,10 k 的正弦波时，其频谱的峰值分别在第 2,7,14 点。

由于实际广播音频信号的 RMS 值、频谱是不断变动的，不便于定量分析，上述结果是在输入信号为标准的正弦波的情况下测得。目前，该目标板已经在电台中进行实际测试，运行稳定，较好实现了音频监控的功能。

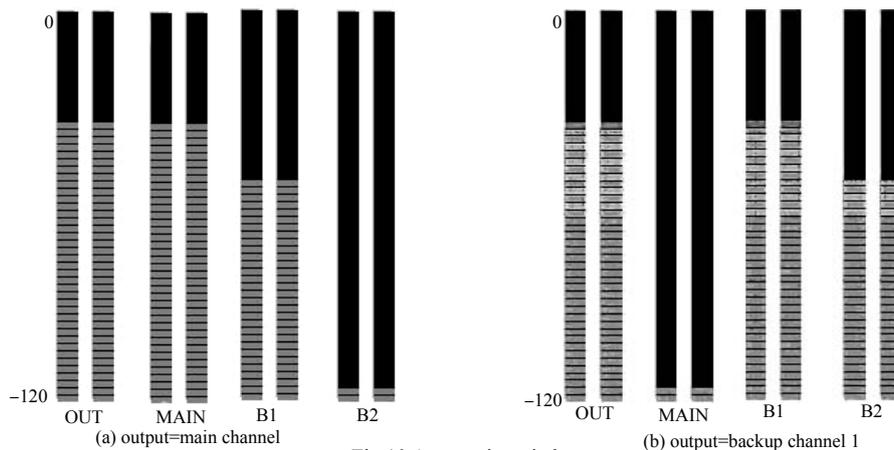


Fig.10 Automatic switch
图 10 自动切换

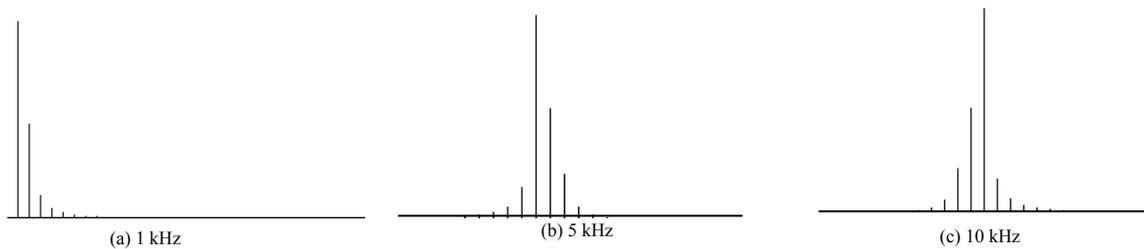


Fig.11 Frequency spectrum
图 11 频谱

5 结论

本文的研究内容主要是利用 FPGA 及其内嵌的软核处理器进行嵌入式开发,实现对电台音频信号的监控,不但完成了预期的目标功能,而且很大程度上减小了专用集成芯片的数量和 PCB 板的体积,具有很好的应用性。由于 FPGA 相对于其他处理器而言,有比较大的并行处理能力和灵活的接口设计上的优势,所以此音频处理器方案有:处理能力强、实现处理算法灵活、开发周期短和功耗低等优势。

同时也可以看到,文中对音频信号进行处理的算法都比较简单,且重复性高,使用 FPGA 进行开发具有较大的优势。然而在开发复杂的算法时,比如音频处理中的均衡、滤波、AGC 等,其计算和控制比较复杂,使用 FPGA 开发比较困难,而这恰好是 DSP 的优势。因此在以后的工作中,可以考虑采用 DSP+FPGA 结构。该结构最大的特点是结构灵活,有较强的通用性,适于模块化设计,从而能够提高算法效率;同时其开发周期较短,系统易于维护和扩展,适合于实时信号处理^[8]。

参考文献:

- [1] 袁晔. 基于FPGA的通用数字化音频处理平台的研究与实现[D]. 北京:北京邮电大学, 2008. (Yuan Ye. Research and Implementation of general digital audio processing platform based on FPGA[D]. Beijing:Beijing University of Posts and Telecommunications, 2008.)
- [2] 田耘,胡彬,徐文波. Xilinx ISE Design Suite 10.x FPGA 开发指南[M]. 北京:人民邮电出版社, 2008. (Tian Yun,Hu Bin, Xu Wen-bo. Xilinx ISE Design Suite 10.x FPGA Development Guide[M]. Beijing:Posts and Telecom Press, 2008.)
- [3] Cirrus Logic. 24-Bit,192 kHz Stereo DAC with Volume Control[Z]. CS4392 datasheet, 2002.
- [4] 叶肇晋,张稀楠,马磊. 基于 Xilinx FPGA 片上嵌入式系统的用户 IP 开发[M]. 西安:西安电子科技大学出版社, 2008. (Ye Zhaojin,Zhang Xinan,Ma lei. On-chip Embedded Systems User IP Development Based On Xilinx FPGA[M]. Xi'an:Xi Dian University Press, 2008.)

(下转第 597 页)