

ZYNQ 主芯片结合 TLV320AIC23 语音芯片实现语音采放

张来洪, 吴道远, 卢继川

(武汉中原电子集团有限公司 研发中心, 湖北 武汉 430000)

摘要: 详细阐述 ZYNQ7000 主芯片结合 TLV320AIC23 语音芯片在 Linux 系统下基于 ALSA 实现语音采放的技术原理, 包括硬件设计、VIVADO 工程设计、系统及驱动、应用程序等, 展示测试方法和效果。该技术方案可直接移植到电台数字板的实际工程中, 具有工程实用性。

关键词: ZYNQ7000; TLV320AIC23; ALSA; 语音采放

中图分类号: TN709

文献标识码: A

DOI: 10.16157/j.issn.0258-7998.222667

中文引用格式: 张来洪, 吴道远, 卢继川. ZYNQ 主芯片结合 TLV320AIC23 语音芯片实现语音采放[J]. 电子技术应用, 2022, 48(9): 127-130, 139.

英文引用格式: Zhang Laihong, Wu Daoyuan, Lu Jichuan. Implementation of voice capture and voice play with the combination of ZYNQ and TLV320AIC23[J]. Application of Electronic Technique, 2022, 48(9): 127-130, 139.

Implementation of voice capture and voice play with the combination of ZYNQ and TLV320AIC23

Zhang Laihong, Wu Daoyuan, Lu Jichuan

(WUHAN ZHONGYUAN Electronics Group Co., LTD R&D Center, Wuhan 430000, China)

Abstract: Technology principles of implementation of voice capture and voice play in Linux operation system using ALSA framework with the combination of ZYNQ and TLV320AIC23 were introduced in this paper, including technology principles such as hardware design, VIVADO project design, system and drivers, applications. The verification methods and results were showed as well. This scheme has obvious practical engineering significance because it can be easily transplanted into practical engineering such as digital processing board of tactical radio.

Key words: ZYNQ7000; TLV320AIC23; ALSA; voice capture and voice play

0 引言

ZYNQ7000 系列芯片是 Xilinx 提供的全可编程片上系统(System on Chip, SoC)芯片, 完美地将 ARM 与 FPGA 融合在一起, ARM 称为处理系统(Processing System, PS)侧, FPGA 称为编程逻辑(Programmable Logic, PL)侧, 既发挥了 ARM 基于操作系统开发应用程序的优点, 又发挥了 FPGA 进行实时信号处理和实现高速接口的优点, PS 侧自带了通用异步收发传输器(Universal Asynchronous Receiver Transmitter, UART)、集成电路总线(Integrated Circuit, IIC)等控制器, PL 侧还能通过 FPGA 的逻辑资源实例化知识产权核(Intellectual Property Core, IP 核)额外扩展集成电路内置音频总线(Inter IC Sound, IIS)等控制器, PS 侧与 PL 侧通过高级扩展接口(Advanced Extensible Interface, AXI)总线接口实现信息的交互, 这些优点使得 ZYNQ 芯片作为电台数字板的主芯片时充分发挥了优势。语音功能作为电台的基本功能, 在数字板上实现语音的采放是进行语音远距离传输的前提。TLV320AIC23 是一

款高性能低功耗立体声音频编解码器, 可以在 8 000 Hz~96 000 Hz 采样率下提供 16、24、32 等不同位数的采样数据, 非常灵活好用。本文详细阐述 ZYNQ 主芯片结合 TLV320AIC23 语音芯片在 Linux 系统下基于高级 Linux 声音架构(Advanced Linux Sound Architecture, ALSA)进行脉冲编码调制(Pulse Code Modulation, PCM)编程实现语音采放的技术原理, 展示了测试效果, 此方案可以移植到电台项目中, 作为电台的语音采集和播放方案, 具有实际工程意义。

1 硬件设计

本文所述 ZYNQ 型号为 xc7z030fbg676-2, ZYNQ7000 系列芯片的 PS 侧是一致的, 都是双核 CortexA9, 仅 PL 侧资源不一样, 芯片封装不一样。ZYNQ 与 TLV320AIC23 的主要连接关系如图 1 所示, 此处只画出两芯片之间的连接关系, 芯片自身所接的外围电路不画出。

ZYNQ 作为主芯片, 向 TLV320AIC23 提供 12.288 MHz 的主时钟信号 MCLK, TLV320AIC23 基于此主时钟运行;

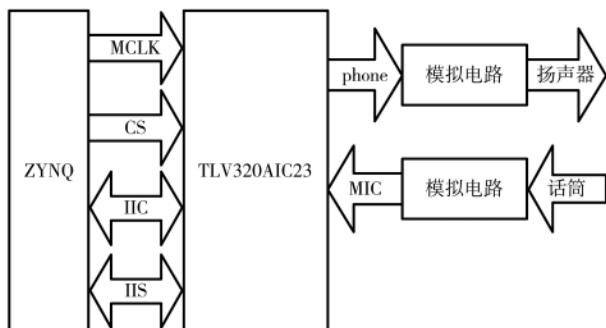


图1 ZYNQ与TLV320AIC23连接关系

ZYNQ向TLV320AIC23提供片选功能CS;ZYNQ通过IIC协议对TLV320AIC23进行配置,例如音量、采样率、量化位数等。ZYNQ通过IIS协议与TLV320AIC23进行语音样点的传输。IIS主要有LR_CLK左右声道选择时钟信号线、B_CLK比特时钟信号线、DATA串行数据线,工程中可进一步将DATA细分为DATA_O数据输出信号线和DATA_I数据输入信号线。LR_CLK和B_CLK由ZYNQ提供,LR_CLK用于区分此时的语音所属的左右声道,在LR_CLK变化后的B_CLK的第二个及之后的上升沿交互数据,B_CLK的一个上升沿传输一位数据,先传输高位再传输低位,这样允许收发采用不一样的数据位数,在工程中,收发的数据位数应尽量设计为相同。

2 VIVADO 工程设计

根据硬件设计,基于VIVADO18.3搭建工程,对ZYNQ的PS进行配置,例如时钟、复位、调试UART控制器等,确保PS部分能正常工作,添加一系列IP核用于实现语音功能。连接模块间的搭建完成的工程如图2所示。

ZYNQ7 Processing System描述了PS侧的配置,例如内存、调试串口控制器、网口控制器等的配置、DMA的

配置、PS-PL时钟的配置等。AXI Interconnect是PS侧与PL侧的AXI接口模块,axi_i2s_adi是进行串并转换实现IIS时序的IP核,且具备先入先出(First In First Out, FIFO)功能用于缓存语音样点。Clocking Wizard是时钟分频模块,将PS向PL提供的100 MHz时钟分频为12.288 MHz,此12.288 MHz的时钟是TLV320AIC23的主时钟,也是axi_i2s_adi模块的数据收发的时钟。AXI IIC是利用PL资源生成的IIC控制器,ZYNQ通过此IP核产生IIC时序。Constan是常量信号产生模块,用于产生片选信号。concat是信号合并模块,用于将多个中断信号源合并后接到PS模块,此处合并AXI IIC产生的中断信号。

PS侧使能了两路DMA资源,axi_i2s_adi模块的DMA_REQ_TX信号接PS的DMA1_REQ信号,axi_i2s_adi模块的DMA_REQ_RX信号接PS的DMA2_REQ。

axi_i2s_adi模块对外提供连接有BCLK_O、LR_CLK_O、LR_CLK_I、SDATA_O、SDATA_I等信号,连同时钟分频模块提供的MCLK_O信号一起组成IIS总线,实现IIS时序,这些信号从ZYNQ出来后最终连接到TLV320AIC23的管脚,用于ZYNQ和TLV320AIC23之间的语音样点的传输。

搭建完VIVADO工程,为axi_i2s_adi模块和AXI IIC模块分别设置AXI地址,axi_i2s_adi地址为0x60000000,AXI IIC地址为0x41600000,然后结合硬件原理图为外部信号做管脚约束,最后生成bit流文件。

3 系统启动及驱动程序

将VIVADO生成的bit流文件和hardware信息导出到SDK,SDK下生成FSBL作为一级引导程序,在Ubuntu虚拟机下使用XILINX厂家提供的arm-xilinx-linux-gnueabi-gcc交叉编译器配置并编译2016.07版的uboot作为二级引导程序,将FSBL、bit、uboot合成BOOT.BIN烧写进板子,启动进入uboot阶段。

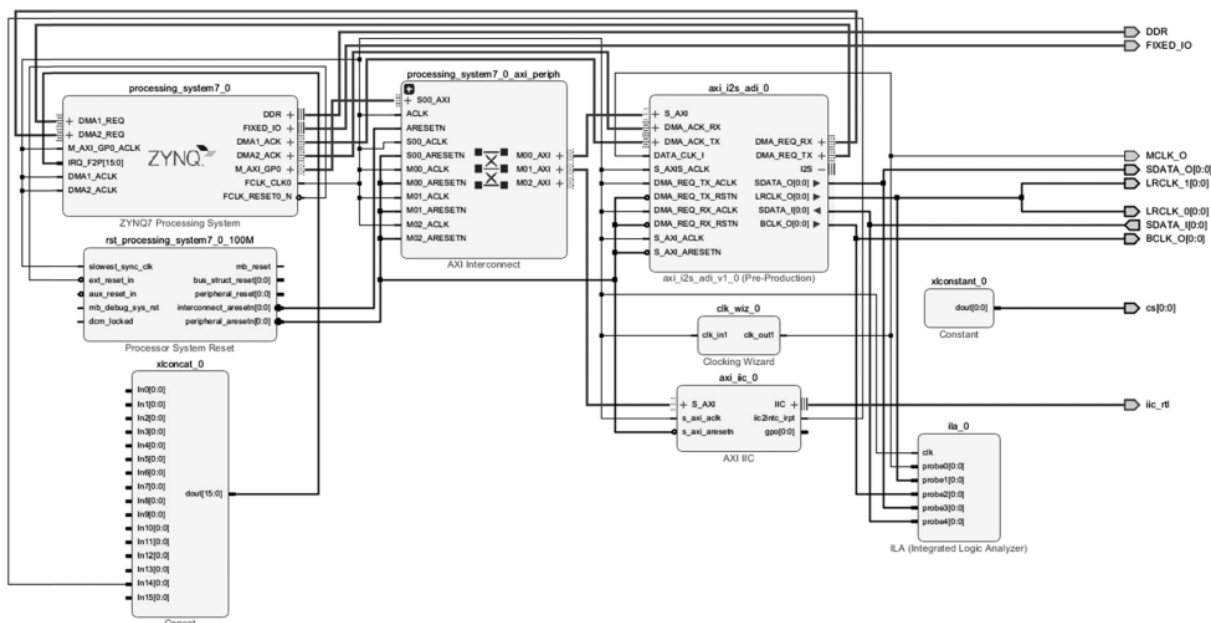


图2 VIVADO工程的模块连接图

配置 4.6.0 版的 Linux 内核,在配置选项中添加 AXI IIC 的驱动、TLV320 系列编解码器的驱动、ALSA for Soc audio 组件的支持等,交叉编译生成 Linux_uImage 内核镜像文件。

修改 Linux 源码目录子目录下的 dts 文件,除了描述 PS 侧资源外,额外添加声卡功能的描述片段,然后编译生成 devicetree.dtb 设备树文件。设备树中描述声卡功能的片段结构如下:

```
fpga-axi@0{
    audio_clock { ... };
    i2c0@41600000 { ...
        sound0-codec@1a { ... };
    };
    axi-i2s0@0x60000000 { ... };
    sound0 {
        simple-audio-card,cpu { ... };
        simple-audio-card,codec { ... };
    };
};
```

此处只展示片段结构,省略冗长的细节字段。一级节点 fpga-axi 包含整个与声卡有关的描述,二级节点 audio_clock 描述的是 12.288 MHz 的时钟,二级节点 i2c0 描述的是 AXI IIC 控制器,AXI IIC 控制器下挂的三级节点 sound0-codec 描述的是 TLV320AIC23,二级节点 axi-i2s0 描述的是 axi_i2s_adi 模块,以上节点描述了 VIVADO 工程中的具体配置;接下来的二级节点 sound0 描述了构造出来的具体声卡功能,即一个简易模式的声卡节点,包含了两路信息流,一路连到 axi_i2s_adi 模块,一路连到 sound0-codec。

与设备树中声卡节点匹配的内核源码主要有 DMA 的驱动程序源码 pl330.c、语音芯片的驱动程序的源码 tlv320aic23.c、axi_i2s_adi 的驱动程序的源码 axi-i2s.c。这些驱动程序源码在 XILINX 厂家提供的内核代码里已经存在,设备树中的片段将这些代码有机地结合起来,形成特定的功能。可以结合工程实际对这些源码进行修改,例如修改 tlv320aic23.c 文件里的 tlv320aic23_codec_probe 函数,使得在 Linux 系统启动阶段加载声卡节点驱动时即可对 TLV320AIC23 进行初步设置。

交叉编译 1.20.1 版本的 busybox,制作成 uramdisk.image.gz 根文件系统,在根文件系统中添加脚本语句用于生成声卡的文件节点。

```
mkdir /dev/snd
ln -s /dev/controlC0 /dev/snd/controlC0
ln -s /dev/pcmC0D0c /dev/snd/pcmC0D0c
ln -s /dev/pcmC0D0p /dev/snd/pcmC0D0p
ln -s /dev/timer /dev/snd/timer
```

在 uboot 阶段分别将 Linux_uImage 内核镜像、devicetree.dtb 设备树、uramdisk.image.gz 根文件系统三组件加

载到内存指定地址,然后启动 Linux 操作系统。

在正常启动 Linux 系统的基础上,可以看到系统启动过程中的打印信息,

```
ALSA device list:
#0: 60000000.axi-i2s0-tlv320aic23-hifi
```

这说明 Linux 系统已经识别了声卡设备,证明 VIVADO 工程、Linux 操作系统、驱动功能都是正常的。但若需声卡正常工作,还需要库文件的支持,以及应用程序的运行。

配置并交叉编译 1.1.3 版本的 ALSA 库源码,交叉编译最终得到整个 ALSA 库的生成文件夹,内含 Linux 系统下与 ALSA 声卡功能有关的动态库、静态库、头文件等。将 ALSA 库的生成文件夹下的内容拷贝到数字板的根文件系统下的相应目录下,通过 aplay -l 命令证明 ALSA 正常工作。

```
List of PLAYBACK Hardware Devices
card 0: D60000000axi2s
device 0: tlv320aic23-hifi-0 []
Subdevices: 1/1
Subdevice #0: subdevice #0
```

4 应用程序

为了使用应用程序验证声卡是否正常工作,可以用 ALSA 库的生成文件夹提供的若干应用程序,也可以自行编写实用性更强更灵活的程序,基于 ALSA 的 PCM 编程有相对固定的操作步骤,关键函数如下。

调用 snd_pcm_open 函数分别打开两个句柄,一个用于采样,一个用于放音。调用 snd_pcm_hw_params_set_access 函数为句柄设置语音样点的模式,此处设置为 SND_PCM_ACCESS_RW_INTERLEAVED 即读写交替模式。调用 snd_pcm_hw_params_set_channels 函数为句柄设置语音声道,设置为 1 即单声道。调用 snd_pcm_hw_params_set_format 函数为句柄设置语音样点的格式,此处设置为 SND_PCM_FORMAT_S16_LE,即每个语音样点用 16 bit 表示。调用 snd_pcm_hw_params_set_rate_near 函数为句柄设置采样率,此处设置为 8 000,即每秒采集或播放 8 000 个样点。调用 snd_pcm_prepare 使句柄准备工作。

调用以上配置函数时,应用程序进一步调用 ALSA 库里的函数,然后调用操作系统内核里的函数,通过层层调用,最终达到驱动层,一方面通过 AXI IIC 的驱动程序来配置 TLV320AIC23 语音芯片,一方面更新了 axi_i2s_adi 模块的工作状态,将以上各函数整个封装后成为初始化函数。也可以通过应用程序访问 AXI IIC 的驱动程序,将 TLV320AIC23 的各寄存器的值配置为项目所需的值。TLV320AIC23 寄存器的配置如表 1 所示。

调用 snd_pcm_writew 函数向放音句柄传递语音信息的变量,即可达到放音的效果。

调用 snd_pcm_readi 函数向采样句柄传递语音信息的变量的指针,即可达到采音的效果。

表 1 TLV320AIC23 各寄存器的配置

地址	功能	配置值
0	设置左声道输入音量	0x17
1	设置右声道输入音量	0x17
2	设置左声道输出音量	0xf7
3	设置右声道输出音量	0xf7
4	选择模拟信号模式	0x14
5	选择数字信号模式	0x0
7	设置数字音频格式	0x12
8	设置采样率	0x0c
9	数字接口激活	0x1
10	复位语音芯片	复位时置 0

结合 snd_pcm_writeti 和 snd_pcm_readti 即可实现对外语音回环的效果。

调用 snd_pcm_writeti 函数时,应用程序进一步调用 ALSA 库里的函数,然后调用操作系统内核里的函数,通过层层调用,最终达到驱动层。通过使用 DMA 通道,将内存中的信息搬移给 axi_i2s_adi 模块的发 FIFO,然后由 axi_i2s_adi 产生 IIS 时序,将发 FIFO 里的语音样点传输给 TLV320AIC23,然后由 TLV320AIC23 将数字量转化为模拟量,这是放音方向。对于采音方向,TLV320AIC23 将模拟量转化为数字量,然后将语言样点传输给 axi_i2s_adi 模块的收 FIFO,等到应用程序调用 snd_pcm_readti 函数时,应用程序进一步调用 ALSA 库里的函数,然后调用操作系统内核里的函数,通过层层调用,最终达到驱动层,通过使用 DMA 通道,将 axi_i2s_adi 模块的收 FIFO 里的信息搬移到内存中的,最终到达应用层。

5 测试

先测试放音功能,运行初始化函数后,通过传递表示不同频率的正弦波样点的数组变量给 snd_pcm_writeti

函数,来播放不同频率的正弦波单音,例如每 8 个样点形成一个正弦周期,则对应 1 000 Hz 的单音,通过示波器观测数字板上 TLV320AIC23 的 PHONE 脚,可以看到示波器观测的波形与代码变量的理论值一致,说明声卡放音功能正常。

再测试采音功能,运行初始化函数后,在 while 循环中成对地依次调用 snd_pcm_readti 和 snd_pcm_writeti,实现对外语音回环的效果,用信号源接到 TLV320AIC23 的 MIC 脚,通过示波器观测 TLV320AIC23 的 PHONE 脚,信号源分别输出不同频率的正弦波,可以看到示波器观测的波形与信号源的设置值一致,说明声卡采音功能正常。

验证了放音功能和采音功能后,可以进一步通过人声来实测喊话自回环功能,给数字板接上话柄扬声器,对着话柄喊话,能在扬声器听到自己所喊声音,说明喊话自回环功能正常。

各项测试结果如图 3 所示,测试结果与理论一致。

6 结论

本文详细阐述了 ZYNQ 结合 TLV320AIC23 实现语音采放功能的原理,对于实际电台的数字板的 VIVADO 工程,只需添加与波形有关的其他模块即可,即对于语音功能,本文所述的技术方案是可以直接移植使用的,在应用程序里采集到的语音,递交给波形模块进行传输,在接收方收到后,再在应用程序里进行放音,即实现了电台里的话音功能,若需优化用户体验,还可以再加提示音等辅助功能。

参考文献

[1] 何宾.Xilinx zynq-7000 嵌入式系统设计与实现[M].第 2 版.北京:电子工业出版社,2019.
[2] 宋宝华.Linux 设备驱动开发详解[M].第 3 版.北京:机械工业出版社,2015.

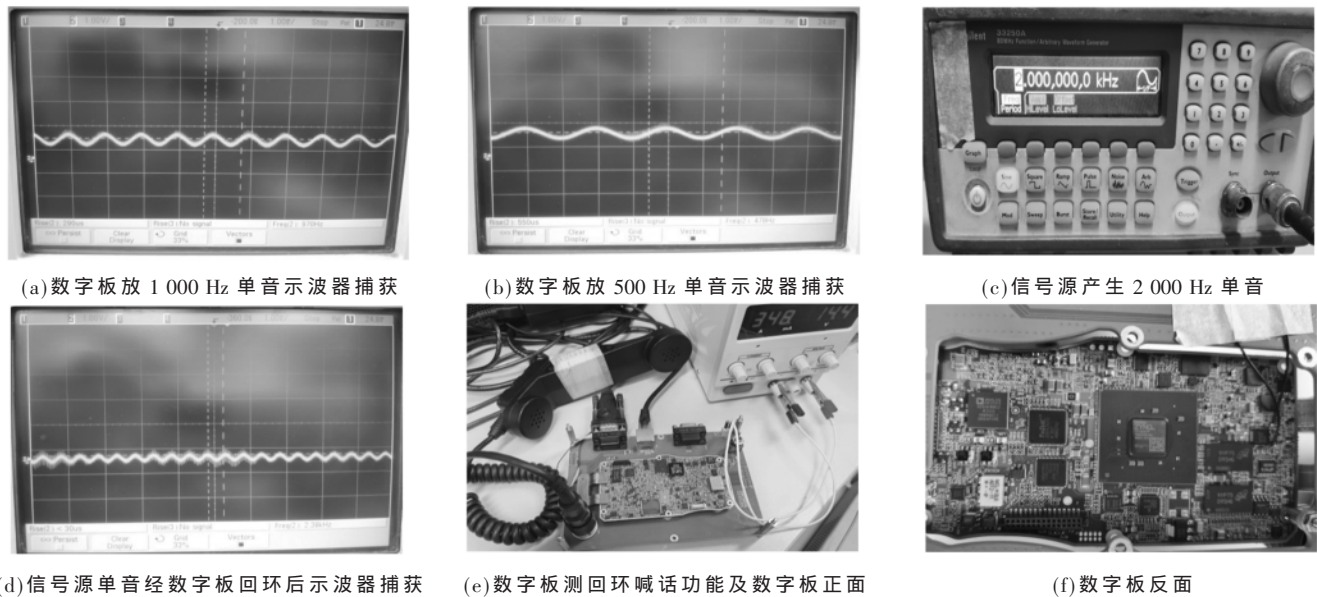


图 3 语音采放功能测试

(下转第 139 页)

曲线通带内信号幅值存在略低的情况,可能是受到测量仪器误差、滤波器内部电阻并非精密电阻等因素的影响。

系统自噪声也是滤波器系统中较重要的参数,反映系统本身对处理信号的影响。其测量方法如下:滤波器正常工作时将输入端接地,输出端连接到示波器,测得的交流有效值为噪声均方根值 V_{rms} ,由此可计算滤波器的自噪声水平,如式(5)所示:

$$V_{noise} = \frac{V_{rms}}{A} \quad (5)$$

式中 A 为系统的实际增益。测量结果如表 7 所示。

表 7 自噪声测试

频率/Hz	V_{p-p}/mV	$V_{rms}/\mu V$
10 000	3.264	575
100 000	3.670	569
5000 000	4.080	515
1 000 000	4.670	489

从表 7 结果可以观测到滤波器的噪声峰峰值低于 4.670 mV,等效输入噪声低于 575 μV ,在滤波器工作过程中对信号影响较小。

4 结论

本文设计实现了一种带宽可调的程控滤波器,研究了电路设计和程控方法,完成了带通可调的低通滤波器硬件电路模块设计和滤波器控制程序设计,并搭建实验测试平台,设计实验并进行测试。通过实验数据可知,该低通滤波器通带在 10 Hz~40 kHz 可调,且截止频率的误差控制在 0.45% 以内,实现低通滤波功能,满足设计需求,适用于噪声测量与谐波失真测量的实际工作。

参考文献

- [1] 王海亮.四通道程控滤波器的设计与实现[D].哈尔滨:哈尔滨工程大学,2014.
- [2] 严小黑,慕文静.层叠型小型化 5G FR2 频段滤波器设

计[J].通信技术,2021,54(7):1768-1773.

- [3] 余晖冬,阮维剑,侯柳英.基于 ADS 的小型化切比雪夫带通滤波器的设计[J].科学技术创新,2021(20):147-148.
- [4] 吴星翰,许双婷,王杨,等.阻尼高通滤波器及其在高压直流系统中的应用[J].高电压技术,2021(8):1-9.
- [5] 任乐,徐金平.具有超宽带吸收特性的 Ku 波段带通滤波器设计[J].微波学报,2021(6):15-20.
- [6] 钟天云.基于 Multisim 的有源滤波器的设计[J].电子技术,2021,50(7):44-45.
- [7] 张斌,李磊,王元源,等.超宽带四通道开关滤波器的设计[J].火控雷达技术,2021,50(2):88-92,102.
- [8] 王万生,关恩明.18 阶巴特沃兹有源低通滤波器设计[J].黑龙江科技信息,2015(21):57.
- [9] 朱思浩,罗显志.可配置八阶巴特沃斯低通滤波器的设计[J].湖北大学学报(自然科学版),2021,43(3):276-281.
- [10] KARANTZALIS P.Free FilterCAD 3.0 software designs filters quickly and easily[J].Electrical Design News,2000,(25):A1-A2.
- [11] 单财良,鲁千红,罗玉文,等.基于 LTC1064 的多功能程控滤波器设计[J].空军雷达学院学报,2010,24(1):47-50.
- [12] 曹彪,梁伟明.频带可变开关电容滤波器设计[J].集成电路通讯,2016(1):4-7.
- [13] 王文胜.基于 MAX264 的程控低通滤波器设计[D].成都:电子科技大学,2010.

(收稿日期:2021-12-03)

作者简介:

杨春玲(1965-),通信作者,女,博士,教授,主要研究方向:信号检测与处理、高精度仪器仪表设计、大规模集成电路实现,E-mail:1300802724@qq.com.

张扬(1999-),男,硕士研究生,主要研究方向:电子系统设计、信号检测与处理。



扫码下载电子文档

(上接第 130 页)

- [3] 弓雷.ARM 嵌入式 Linux 系统开发详解[M].第 2 版.北京:清华大学出版社,2014.
- [4] 杨宗德.Linux 高级程序设计[M].第 3 版.北京:人民邮电出版社,2012.
- [5] 付扬.基于 Nios 软核的音频播放系统设计与实现[J].现代电子技术,2020,43(10):59-62.
- [6] 付志利.基于音频编解码器 TLV320AIC3106 与 FPGA/CPLD 的数字化语音处理系统[J].电子质量,2021(3):28-33.
- [7] 陈子为.基于 zynq 的视频实时拼接系统[J].电子技术应用,2021,47(2):67-71.

- [8] 张艳辉.Vivado HLS 嵌入式实时图像处理系统的构建与实现[J].电子技术应用,2016,42(9):115-117.

(收稿日期:2022-02-22)

作者简介:

张来洪(1989-),男,硕士,工程师,主要研究方向:嵌入式技术、电台主控及波形。

吴道远(1992-),男,硕士,工程师,主要研究方向:嵌入式技术、电台主控及波形。

卢继川(1987-),男,硕士,工程师,主要研究方向:嵌入式技术、电台主控及波形。



扫码下载电子文档

版权声明

经作者授权，本论文版权和信息网络传播权归属于《电子技术应用》杂志，凡未经本刊书面同意任何机构、组织和个人不得擅自复印、汇编、翻译和进行信息网络传播。未经本刊书面同意，禁止一切互联网论文资源平台非法上传、收录本论文。

截至目前，本论文已经授权被中国期刊全文数据库（CNKI）、万方数据知识服务平台、中文科技期刊数据库（维普网）、DOAJ、美国《乌利希期刊指南》、JST 日本科技技术振兴机构数据库等数据库全文收录。

对于违反上述禁止行为并违法使用本论文的机构、组织和个人，本刊将采取一切必要法律行动来维护正当权益。

特此声明！

《电子技术应用》编辑部

中国电子信息产业集团有限公司第六研究所