AVR单片机Atmega128在FPGA配置中的运用

何永泰[1],李 莹[2]

(1.楚雄师范学院物理与电子信息科学系,楚雄 675000;
2.中国电子科技集团公司第四十七研究所,沈阳 110032)

摘 要:在嵌入式系统设计中,掌握MCU对FPGA的配置,对系统的设计是十分必要的。根据EPF10K10的配置时序和AVR单片机Atmega128的接口特点,详细介绍了Atmega128对EPF10K10配置的软硬件设计原理。

关键词:单片机;Atmega128;FPAG配置
中图分类号:TN876.3 **文献标识码**:A **文章编号**:1002-2279(2006)02-0005-02

Apply AVR Singlechip Atmega128 to Configure FPGA

HE Yong-tai[1],LI Ying[2]

(1. *Department of Physics and Electron Information Science, Chuxiong Normal University, Chuxiong* 675000, *China*;
2. *The 47th Research Institute of China Electronics Technology Group Corporation, Shenyang* 110032, *China*)

Abstract: In the design of embed system, grasping configure of MCU to FPGA is essential to design of system. Theory of configureing Atmega128 to EPF10K10 was introduced based on configuring timing of EPF10K10 and interface characteristic of singiechip Atmega128 in the paper.

Key words: Singlechip; Atmega128; FPGA

1 引 言

目前,在嵌入式系统设计中,微控制器MCU+可编程逻辑器件PLD的设计方式已经成为系统设计的一般方法。在系统设计中将MCU数据处理的优势与PLD超强的逻辑处理功能相结合,能够设计出大型的多层次的嵌入式系统。在系统设计中MCU主要采用8位和32位两类,其中,在8位MCU中,AVR单片机是运算速度最快,功能最强,使用最为方便的MCU,已经得到了广泛的推广使用。在PLD中,又分为CPLD和FPGA,其中,FPGA是基于SRAM LUT结构的可编程逻辑器件,由于其内部采用SRAM工艺,它的配置数据存储在SRAM中。由于SRAM的易失性,每次系统上电时,必须重新配置数据,即ICR(In-Circuit Reconfigurability),只有在数据配置正确的情况下系统才能正常工作。在线配置方式一般有两类:一是通过下载电缆由计算机直接对其进行配置,二是通过配置芯片对其进行配置。通过PC机对FPGA进行在系统重配置,虽然在调试时非常方便,但在应用现场是很不现实的。象Altera公司提供的配置芯片有一次可编程型和可擦除编程型两种:一次可编程型芯片只能写入一次,不适合开发阶段反复调试、修改及产品的升级;可擦除编程型价格昂贵,且容量有限,对容量较大的可编程逻辑器件,需要多片配置芯片组成菊花链形进行配置,增加系统设计的难度。为此,利用MCU对FPGA进行配置具有非常实用的价值,特别是AVR单片机ATmega128,由于它具有128K的Flash存储器,在无须外部存储器扩展的情况下,也能实现对FPGA的配置。下面将以ATmega128对EPF10K10的被动串行配置为例,介绍其配置的软硬件设计原理。

2 FPGA器件的配置原理和配置文件

EPF10K10象其它FPGA器件一样。它的配置方式可分为PS(被动串行)、PPS(被动并行同步)、PPA(被动并行异步)、PSA(被动串行异步)和JTAG(Joint Test Action Group)等五种方式。PS方式因电路简单,对配置时钟的要求相对较低,因而被广泛应用。AVR单片机对FPGA的配置也采用PS配置方式来实现ICR功能,在被动串行配置中与配置有关的引脚如表1所示,其配置时序图如图1所示。

在配置过程中,nCONFIG低电位使EPF10K10复位,在由低到高的跳变过程中启动配置;nSTATUS:双向漏极开路;命令状态下器件的状态输出。加电后,EPF10K10立即驱动该引脚到低电位,然后在100ms内释放掉它,nSTATUS必须经过1.0k电

作者简介:何永泰(1970-),男,云南南华人,硕士研究生,教师,主研方向:嵌入式系统的研究。
收稿日期:2004-10-10

阻上拉到 Vcc，如果配置中发生错误，EPF10K10 将其拉低；CONF_DONE：在配置期间，EPF10K10 将其驱动为低，所有配置数据无误差接收后，EPF10K10 将其置为三态，由于有上拉电阻，所以将变为高电平，表示配置成功；DCLK：为外部数据源提供时钟；DATA：数据输入端。当配置数据全部正确地移入目标芯片内部后，DCLK 必须提供几个周期的时钟（具体周期数与 DCLK 的频率有关），确保目标芯片被正确初始化，进入用户工作模式。

表 1 与配置有关的引脚

引 脚	方 向	说 明
DATA	输入	数据引脚
DCLK	输入	同步时钟
NCONFIG	输入	异步复位引脚
CONF_DONE	输出	配置状态
nSTATUS	输出	下载状态

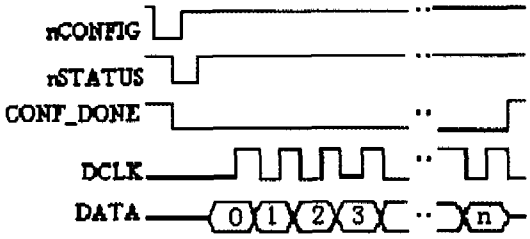

图 1 被动配置时序图

MAX + PLUS II 开发工具可以生成多种配置或编译文件，用于不同配置方法的配置系统，而对于不同系列的目标器件配置数据的大小也不同，如表 2 所示。配置文件的大小一般由 .rbf 文件决定。.rbf 文件即二进制文件，该文件包括所有的配置数据，一个字节的 .rbf 文件有 8 位配置数据，每一字节在配置时最低位最先被装载。单片机 ATmega128 可以读取这个二进制文件，并把它装载到目标器件中。Altera 提供的软件工具不自动生成 .rbf 文件，须按照下面的步骤生成：①在 MAX + PLUS II 编译状态，选择文件菜单的变换 SRAM 目标文件命令；②在变换 SRAM 目标文件对话框，指定要转换的文件并且选择输出文件格式为 .rbf，然后确定。

3 配置的硬件电路设计

用单片机配置 FPGA，可以使用普通输入输出口或串行口。使用普通 I/O 口（如 P1 口），向 PLD 发送 1Bit 数据至少需要 4 个指令周期。一个指令给 DATA 赋值，两个指令产生 DCLK 时钟，一个指令移位取数据。如果晶振为 fosc，一个指令周期为 12/fosc，因此它的下载速率为 fosc/48。然而如果采用串行口方式 0，其下载速率提高为 fosc/12。考虑到 PLD 配置文件数据比较大，通常都在数十千字节以上（其配置文件大小如表 2）。为了加快配置速度，在系统中采用单片机 ATmega128 的串行口实现对 EPF10K10 的配置，配置的硬件电路如图 2 所示。在图 2 中，单片机 ATmega128 是 AVR 单片机中功能最强的一种，在相同的晶振下，其运行速度是 MCS51 内核单片机的 10 多倍。它可以通过同步串行 SPI 口或边界扫描 JTAG 口实现对 128K Flash 存储器的编程下载，因此，具有现场编程功能。由于它有 128K 字节的 Flash 存储器，因此，它可以对表 2 中除 EPF10K70 外的各种器件实现配置。在系统中，配置程序、系统的用户应用程序以及配置文件在内的 .rbf 文件，经过组合后，形成一个新的 .hex 文件，它通过 SPI 接口下载到 ATmega128 的 Flash 程序存储器中。系统上电后，在配置程序的控制下，包括配置文件在内的 .rbf 文件中的数据通过串行口对 EPF10K10 实现自动配置。

表 2 部分 FPGA 的配置文件大小

器件	配置数据大小（Bits）	配置文件大小（Kbytes）
EPF10K10	118,000	15
EPF10K20	231,000	29
EPF10K30	376,000	46
EPF10K40	498,000	61
EPF10K70	892,000	109

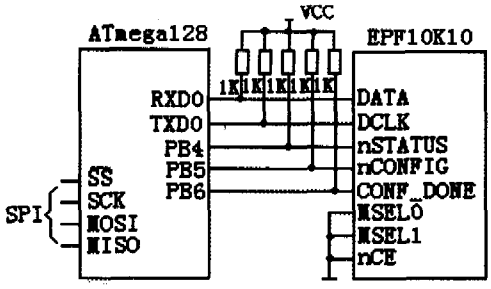

图 2 配置的电路原理图

4 配置的软件设计

在软件设计时，结合硬件设计原理，使用了串行口移位寄存器输入输出方式，即 UART0 口 PE0（RXD0）、PE1（TXD0）工作在通信方式 0 模式。本系统只需用到输出方式，串行数据通过 RXD0 引脚输出，而在 TXD0 引脚输出移位时钟。当一字节数据写入串行数据缓冲器 SBUF 时，就开始发送。在此期间，发送控制器送出移位信号，使发送移位寄存器的内容右移一位，直至最高位（D7 位）数字移出后，停止发送数据和移位时钟脉冲。配置程序的设计中，主要包括两部分即主程序部分和串行发送完成中断程序部分。其中，主程序主要完成对系统的初始化，包括设置各配置端口的初始状态、开中断及启动串行发送等；串行发送中断程序主要完成对 CONF_DONE 信号即 PB6 的检测和数据发送，程序流程图如图 3、4 所示。 （下转第 10 页）

经完成等等。由于 EDMA 所有的通道共享一个中断,所以 ISR 中首先需要判断是哪个通道导致中断服务程序的运行,然后修改对应的传输任务队列计数值,最后调用 callback 函数通知适配器传输已经完成。

除了上面详细列出的函数,DLIO 适配器与 LIO 控制器结构中还定义了诸如表 1 中所列出的打开设备、关闭设备、控制命令等等函数,这些函数与上述的传输函数一起构成了驱动程序的整体。由于篇幅关系,这里不一一阐述。

3.4 数据通路

如图 3 所示,SIO API 函数以及适配器、控制器函数结合使用,就完成了驱动程序所要求的数据传输工作。

值得注意的是,DSP 与外设之间无论是数据输入还是数据输出,都要调用上述的 ISSUE/RECLAIM 全过程。向外设发送数据的时候,我们将数据帧发送给外设,而从外设获得空的数据帧;从外设接收数据的时候,我们将空的帧发送给外设,而外设返回满的数据帧。这个过程,无论是使用标准传输模式还是 ISSUE/RECLAIM 模式都是要进行的。

4 总 结

本文中所设计的是一种基于 DLIO - LIO Controller 双层结构的驱动程序,在 CCS2.2 开发环境中调试通过并在 C6711 平台中稳定运行。实验比较证明,这种驱动程序结构具备了 DSP 系统实时 IO 所要求的可靠性和灵活性,并且能有效的节省程序代码所占用的存储器空间。

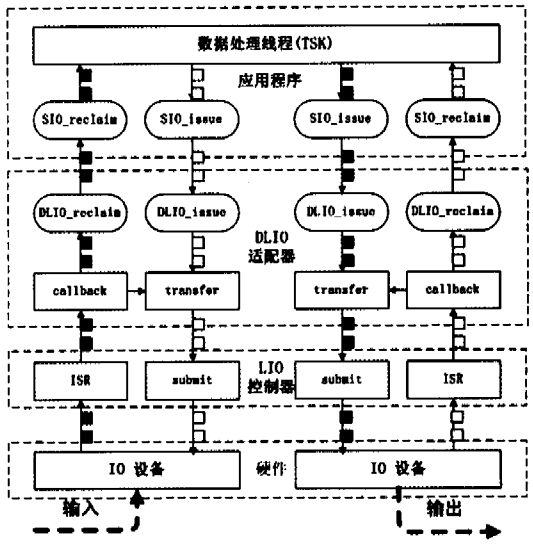

图 3 DLIO 数据流图

从 CCS2.2 开始 DSP/BIOS 还提供 LIO - IOM 结构的驱动模型,相对本文中采用的这种结构,IOM 驱动除了提供了基于帧的传输,也能进行基于数据单元的传输,是将来驱动开发的一个趋势。

参考文献:

[1] Texas Instruments Inc. TMS320 DSP/BIOS User's Guide [M]. 2002.

[2] Texas Instruments Inc. Writing DSP/BIOS Device Drivers for Block I/O[M]. 2003.

[3] Texas Instruments Inc. DSP/BIOS Driver Developer's Guide[M]. 2002.

[4] Texas Instruments Inc. TMS320C6000 Chip Support Library API User's Guide[M]. 2002.

(上接第 6 页)

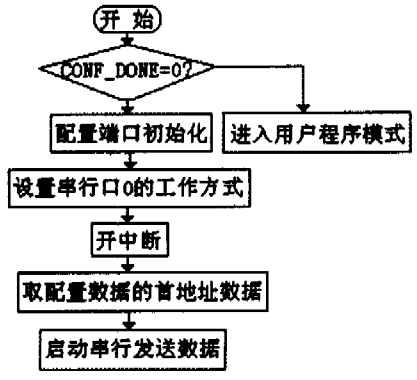

图 3 配置主程序的流程图

5 结 束 语

在利用单片机对 FPGA 进行配置时,除了利用单片机的 I/O 端口准确地模拟各配置端的时序外,单片机的配置程序与用户应用程序之间,一定要注意在上电时执行配置程序,复位时执行用户应用程序的关系。在用配置数据文件.rbf 与配置程序和用户应用程序的.hex 文件,组合成新的.hex 文件时,要注意配置文件在程序存储器中的地址,保证与配置程序中访问的地址一致。

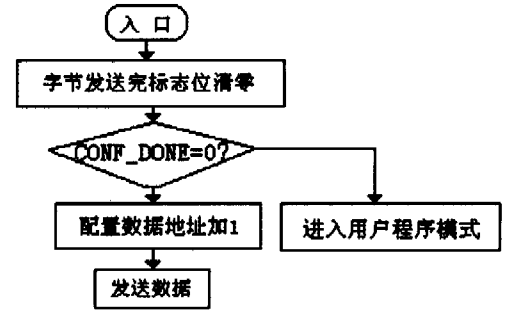

图 4 中断程序的流程图

参考文献:

[1] Atmega128 datasheet[DB/OL]. www.sl.com

[2] http://www.pld.com

[3] 褚振勇,翁木云编著. FPGA 设计及应用[M]. 西安:西安电子科技大学出版社,2002.