

# 一种基于双口 RAM 的环形数据缓存系统

李震

(东北大学 信息科学与工程学院, 辽宁 沈阳 110004)

**摘要:**在嵌入式系统的设计中,时常需要对大量数据进行缓冲处理。文中提出了一种基于双口 RAM 的环形数据缓存系统。这种设计最大限度地使用了双口 RAM 的存储空间,有效地解决了大量数据缓存和对缓存数据的复杂数据处理,大大提高了上位机的数据处理能力。这种设计同时提供了扩展系统总线带宽的方法,提高了上位机读取数据的速度,可以应用于需要总线扩展的场合。此方法解决了实际项目中的许多技术问题,具有良好的扩展性和可移植性。

**关键词:**双口 RAM; 环形数据缓存; 总线扩展

**中图分类号:** TP273+.5

**文献标识码:** A

**文章编号:** 1673-629X(2010)01-0201-04

## A Cycle Data Caching System Based on Dual-Port RAM

LI Zhen

(College of Information Science & Engineering, Northeastern University, Shenyang 110004, China)

**Abstract:** In embedded system, there are many cases that need mass data caching. Presents the design of cycle data caching system based on dual-port RAM. This design effectively solves mass data caching and complex data processing, therefore greatly enhances upper computer's ability of data processing. This design also provides a method of extending system's bandwidth, which increases the reading speed of upper computer, and this method can be applied in cases where bandwidth extending is needed. Based on this design, many technical problems were solved. This design has good expansibility and transfer ability.

**Key words:** dual-port RAM; cycle data caching; bandwidth extending

## 0 引言

在嵌入式系统的设计中,时常出现上位机和下位机之间的大量数据传输以及对传输数据的复杂处理。如果处理不当,则极易产生数据拥堵现象,这会对系统的实时处理能力产生不利的影响,在严重的情况下甚至导致系统不能正常工作。并行数据处理可以使得上位机和下位机之间的数据交换更加灵活快速,进而可以加快数据处理的速度<sup>[1]</sup>。在并行数据处理的思想下考虑设计合理的数据缓存系统,通过数据缓存提高上位机和下位机之间的数据传输速度,提高系统的实时性和可靠性。当上位机和下位机的时钟不同步时,如果采用常规的 SRAM,则会对硬件的设计提出很高的要求。一旦处理不当,则系统的稳定性得不到保障。双口 RAM 所提供的双端口异步读写能力则可以轻松解决时钟不同步的问题<sup>[2]</sup>。文中提出了一种基于双口

RAM 的环形数据缓存系统,成功地解决了上位机和下位机大量数据传输时所产生的数据拥堵问题,减轻了上位机数据处理的负担。这种设计方法同时提供了总线扩展的方法,可以将 8 位数据总线扩展到 16 位数据总线,其配置方法灵活可靠,适用于大量数据交换的总线扩展的场合。

## 1 双口 RAM 介绍

双口 RAM 是一种高速静态随机存储器。它具有两个独立的读写端口,两个端口公用一套存储单元,每个端口具有独立的控制线路(包括地址线、数据线和控制线),可以独立、异步地对存储器中任意位置上的数据进行独立的存取操作,并由空/满标识产生的逻辑进行读写地址比较来保证正确的读写操作<sup>[3]</sup>。

本设计中使用了美国 IDT 公司的 IDT7132。IDT7132 是 2k×8bit 的高速双口 SRAM。读写最高速度可以达到 25nm。最大功耗 325mW,在维持模式下的维持功耗仅为 1mW。它具有两套独立端口,提供了 BUSY 信号硬件仲裁机制。当两个端口对不同存储单元进行操作时,或者两个端口同时对同一存储单元进

收稿日期:2009-05-26;修回日期:2009-08-14

基金项目:东北大学创新项目

作者简介:李震(1986-),男,山东青岛人,硕士研究生,研究方向为智能控制系统;导师:周玮,副教授,研究方向为复杂工业过程建模、控制与优化、机器博弈等。

行读操作的时候,  $\overline{\text{BUSY}}_{\text{L}}$  和  $\overline{\text{BUSY}}_{\text{R}}$  信号都输出高电平。此时两个端口对双口 RAM 的操作和普通 RAM 完全相同。当两个端口同时对一个存储单元进行写操作时, 或者一个端口对某一存储单元进行读操作, 而另一个端口对同一个存储单元进行写操作的时候, 硬件仲裁机制就会发生作用。允许首先发出控制信号的那个端口进行读或者写操作, 同时屏蔽掉另一个端口的操作<sup>[4]</sup>, 并使被屏蔽掉的端口的  $\overline{\text{BUSY}}$  信号线输出低电平。

## 2 应用背景

在笔者参与的项目大密度客流检测中, 下位机 C8051F040 板每隔 0.2ms 从红外传感器中采集原始数据, 数据格式为 12 字节, 经过数字滤波后的原始数据需要源源不断地传给上位机 S3C2410A, 上位机需要从原始数据中提取特征元素并构建数据库以进行下一步处理。由于原始数据中的特征元素耦合在一起, 在构建数据库的过程中时常需要对部分数据重复读取。由于需要处理的数据量较大, 如果采用直接的并口传输则会产生数据堵塞的情况, 而且使得上位机的软件实现异常复杂。为了解决这个难题, 笔者选择使用双口 RAM 构建环形数据缓存系统, 成功地解决了上位机数据库的建立, 同时解决了数据堵塞问题。与此同时, 将下位机的 8 位数据总线扩展成 16 位总线, 使上位机读取速度提高了一倍, 提高了上位机的数据处理速度。

文中提供的设计方法可以应用于如下场合:

(1) 写入和读出的数据量较大, 需要进行数据缓存的情况。

(2) 上位机下位机的时钟异步, 写入和读出需要异步同时进行。

(3) 数据处理复杂, 处理完毕的数据可以被覆盖。

(4) 系统需要更宽的总线, 需要转换数据带宽(如将 8 位总线转换成 16 位总线)。

## 3 几种数据缓存的方案

(1) 缓存空间固定分组的思想。

将 2k 存储空间分为两个数据区, 每个数据区各有 1k 空间。当下位机写第一个数据区时, 上位机读取第二数据区的内容; 当下位机写第二个数据区时, 上位机读取第一个数据区的内容。然后利用 I/O 信号线组成握手信号, 配合软件完成两个端口数据的读写<sup>[5]</sup>。

优点: 结构简单, 软件容易实现。

缺点: 2k 空间不能灵活运用, 缓存空间的使用效率低。对于内容上连续的缓存数据处理效率较低(连续的数据可能被分割存储在两个部分中, 会影响到上

位机的数据处理)。而且当上位机和下位机读写速度差别较大的时候, 容易出现数据拥堵的现象。

(2) 环形数据缓存的思想。

将 RAM 中的 2k 存储空间首尾相连形成环状, 并不将 2k 的存储区做固定的分块, 每个字节作为一个单元。下位机写指针保持在上位机读指针的前方, 下位机不断地写入新数据同时覆盖已经处理完的旧数据, 与此同时上位机不断地读出新数据。当上位机读指针或下位机写指针达到 2k 空间最低端的时候, 通过软件设置将指针调整到 2k 空间的开始位置。

初始化时下位机首先写入 1k 的数据, 然后上位机紧跟其后从开始位置读出数据。通过 IDT7132 硬件仲裁机制和上位机下位机的中断服务, 并结合软件系统的设计, 确保下位机写指针维持在上位机读指针的前面。

为了扩展总线带宽, 提高上位机读数据速度, 这里将两片 8 位 IDT7132 并联起来使用, 组成 16 位的数据总线。上位机可以一次读出一个 16 位的半字, 将读取速度提高为原来的两倍。

优点: 最大限度地使用了 2k 的存储空间, 满足系统对大量数据复杂处理的要求。对于内容上连续并需要连续处理的缓存数据尤为适用, 提高了上位机处理数据的效率, 并且解决了数据拥堵问题。同时将 8 位总线转换成为了 16 位总线, 提高了上位机读取数据的速度, 可以应用于需要扩展系统总线的场合。

## 4 环形数据缓存系统的构造

首先在不考虑总线带宽扩展的情况下, 说明使用 IDT7132 构建环形数据缓存系统的方法。然后结合实际应用介绍含有总线扩展功能的环形数据缓存系统的构建。

### 4.1 环形数据缓存系统的工作原理

这里规定双口 RAM 左端口连接下位机, 右端口连接上位机。下位机通过左端口不断地向双口 RAM 写入缓存数据, 上位机通过右端口不断地从双口 RAM 中读出缓存数据。

下位机写入数据时, 地址指针向上增长。当写入地址达到 2k 地址空间的最大值时, 写入指针通过软件调整到了 2k 地址的开始位置。写入新数据的同时, 覆盖处理过的旧数据。

上位机的读出数据的指针动作方式和下位机写入的方式完全相同。只是上位机的读指针位置必须保持在下位机写指针的后面。这样就形成了一个环形的写入和读出的方式, 最大效率地使用了 2k 的存储空间。如图 1 所示。

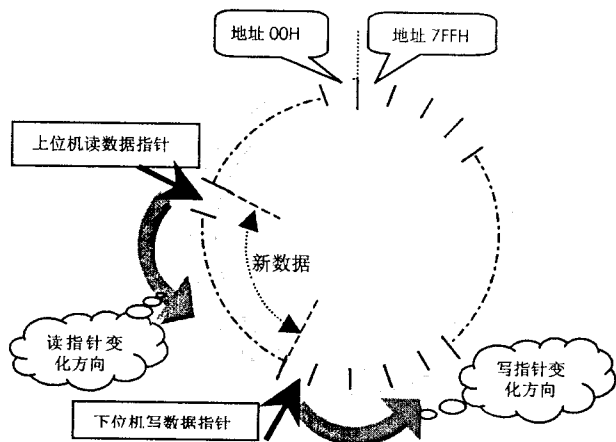


图1 环形数据缓存示意图

这种环形数据缓存系统只要保证写数据指针和读数据指针的相对位置就可以正常工作。这里存在两个问题需要解决:

①如何保证读数据指针不跃过写数据指针而进行非法读操作。

如果读数据指针越过了写数据指针,那么读出的数据就是以前处理过的无序数据。这些数据对于当前的数据处理而言就是错误的。而且如果读数据指针跃过写数据指针,那么会对后续的操作产生极大的影响,在接下来读出的2k字节的数据都将会是处理过的旧数据。这样会破坏上位机读出数据的连续性,使得上位机数据库的建立产生严重错误,并最终使得数据缓冲失效。因此必须避免这种情况的发生。

为了解决这一问题,需要将硬件设计和软件设计结合起来。详细的设计将在下文中介紹。

②如何保证写数据指针不跃过读数据指针而进行非法写操作。

如果写数据指针跃过了读数据指针,也就是说下位机写入的速度过快而上位机读出的速度无法跟随。那么最新写入的数据就会覆盖没有被及时处理的数据,造成大量数据丢失。这在缓存系统的设计中是不允许的。

解决这个问题最直接的方法就是提高上位机读出数据的速度,来使得读数据指针的移动速度大于写数据指针的移动速度。这里所提供的总线扩展方法将8位总线转换成16位总线。在写入速度不变的前提下,上位机读数据的速度相对于8位数据总线时提高了一倍,从而满足了系统的要求。

## 4.2 不含总线扩展的环形缓存系统

### 4.2.1 硬件设计

硬件设计时充分利用了IDT7132的硬件仲裁机制,并结合上位机和下位机的外中断,有效地解决了指针越限。采用1片IDT7132作为数据缓存,左侧端口

连接8位的下位机AT89C52,右侧端口连接8位的上位机C8051F040。左端口的 $\overline{BUSY}_L$ 分别连接上位机和下位机的 $\overline{INT1}$ 上,右端口的 $\overline{BUSY}_R$ 连接到上位机的 $\overline{INT0}$ 上。片选信号 $\overline{CE}_L$ 和 $\overline{CE}_R$ 分别接地, $\overline{OE}_L$ 连到下位机P3.7, $\overline{OE}_R$ 连到上位机P3.4。 $\overline{OE}$ 信号线的作用是在IDT7132闲置时,将芯片设置到低功耗状态。由于双口RAM的地址线是11位,采用3根P3口的I/O线作为高位地址线,从而达到访问2k地址空间的目的<sup>[6]</sup>。详细的连线方法如图2所示。

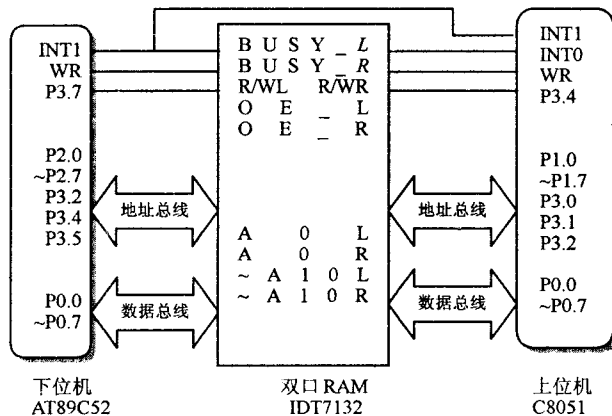


图2 不含总线扩展的环形缓存系统

当上位机读数据指针将要越过下位机写数据指针时,会发生如下两种情况:

(1) 读数据和写数据的指针分别执行到相同位置,下位机写入信号稍快于上位机读出信号。下位机写入无影响,上位机此次读出的数据无效。

(2) 读数据和写数据的指针分别执行到相同位置,下位机写入信号稍慢于上位机读出信号。此时上位机写入无效,上位机读取的数据也无效。

当第(1)种情况发生时, $\overline{BUSY}_R$ 输出低电平有效信号,这会激活上位机 $\overline{INT0}$ 中断,通知上位机当前读取的位置为下位机写入的位置,此次读入的数据无效,读指针减1,并需要加入延时,等待下位机写入更多的数据。这些相应的操作在上位机的 $\overline{INT0}$ 中断服务程序中得以实现。

当第(2)种情况发生时, $\overline{BUSY}_L$ 输出低电平有效信号,同时激活下位机的 $\overline{INT1}$ 和 $\overline{INT1}$ 上位机的,通知下位机此次写入的数据无效,需要重新写入数据。这些操作在下位机的 $\overline{INT1}$ 中断服务中得以实现。与此同时,通知上位机此次读数据操作和下位机写操作冲突,读出的数据无效,需要加入延时,等待下位机写入更多的数据。这些操作在上位机的 $\overline{INT1}$ 中断服务程序中得以实现。这里需要注意,下位机需要自己的寄存器中分出少量空间来进行数据缓冲。这样做的目的就是在写入双口RAM当前数据的同时,保留上一次写

入的数据,以应对第 2 种情况发生时造成的写入数据丢失。

#### 4.2.2 软件设计

如果只用以上所说的中断服务软件设置,而不加入任何其它的软件处理,那么系统可以正常的运行,但是这种情况上位机和下位机中断服务的次数较多,会消耗一部分系统性能。为了尽可能地提高系统的性能,减少进入中断的次数,除了在硬件部分介绍的硬件中断服务以外,还需加入相应的软件处理。具体做法如下:

下位机写入数据时,预先在下一个地址空间上写入一个警告标识(如:01010101),应该保证这个警告标识在现场采集的数据中是不可能发生的。上位机读出数据时首先判断读出的数据是否是警告标识,如果是则立即停止读取新数据,此次读出的数据无效,读数据指针减 1,并调用延时程序等待下位机写入新数据。

这样做的好处就是:可以较大地减少硬件冲突的次数,减少单片机中断响应的次数,提高上位机和下位机的处理速度。

#### 4.3 含有总线扩展的环形数据存储系统

在实际应用中,下位机需要不断地采集红外线扫描传感器输出的原始数据,采样间隔约为 0.2ms。这些数据经过简单的数字滤波后输入到双口 RAM 左侧端口。上位机需要将这些数据从右侧端口读出,并从读出的这些原始数据中分离出有用的特征值构建数据库,从而为神经网络提供数据。由于上位机数据处理的要求,上位机需要对部分缓存数据重复读取。

由于读取的数据量较大,为了提高读取速度,采用两片 8 位的 IDT7132 并联使用,将系统总线扩展为 16 位。下位机写入数据的时候,通过控制两个芯片左侧的片选端口来分别写入 8 位数据。上位机 ARM 读出数据的时候,通过 16 位数据总线一次读出 16 位数据,从而将读取数据的速度提高了 1 倍。进而提高了上位机系统对数据的处理速度。

硬件连线如图 3 所示。

下位机的 P1.7 连接反相器,将 P1.7 的输出信号和取反信号分别连接到两片 IDT7132 的片选  $\overline{CE}_L$  上<sup>[7]</sup>,以达到对两片 IDT7132 轮流选择并分别写入的目的。两片 IDT7132 左侧的两个  $\overline{BUSY}_L$  信号线分别连接到下位机的两个外中断  $\overline{INT0}$  和  $\overline{INT1}$  上,并同时并联到 S3C2410A 的两个中断 GP2 和 GP3<sup>[8]</sup>。这样做保证了下位机对两片 IDT7132 进行分别写入时会有

独立的中断服务与之对应,写入过程中如果出现冲突,则可针对出现冲突的那一片 IDT7132 进行处理。分别并联到了 S3C2410A 的两个中断 GP2 和 GP3,和没有总线扩展时上位机  $\overline{INT1}$  中断的作用相同。IDT7132 右侧的两个  $\overline{BUSY}_R$  连接到 S3C2410A 的中断 GP4 上,GP4 中断处理过程和没有总线扩展时上位机  $\overline{INT0}$  中断的作用相同。

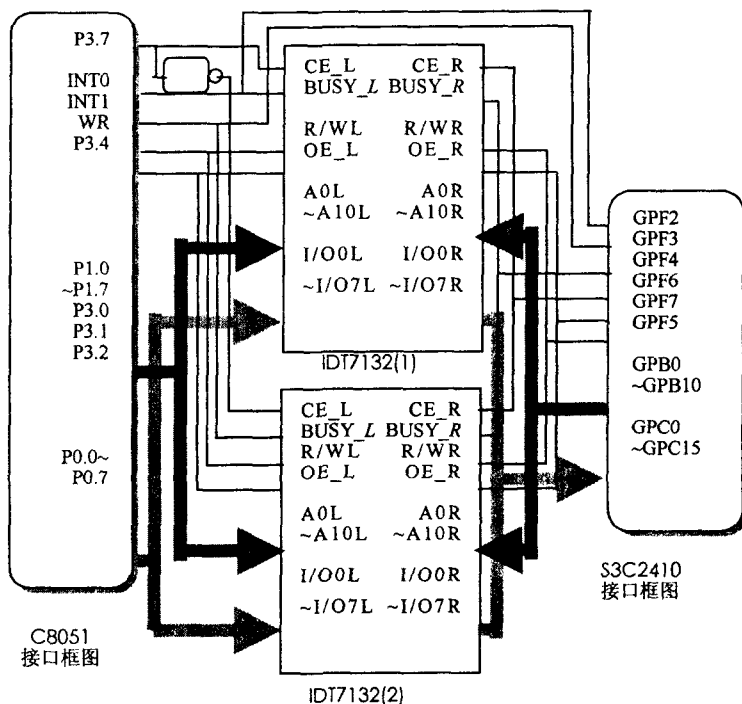


图 3 含有总线扩展的环形缓存系统

在下位机的中断服务程序中需要做如下设置,由于上位机和下位机都增加了一个外部中断,因此需要配置好增加的中断服务。下位机增加了  $\overline{INT1}$  所对应的外部中断。这个外部中断服务程序的配置方法和无总线扩展时下位机  $\overline{INT0}$  的中断服务程序配置方法大致相同,唯一不同的地方就是区分出了是哪一片 IDT7132 出了问题,并对其修正。上位机的 GP2 和 GP3 的中断服务目的相同,都是通知上位机当前的读出操作出现异常,抛弃刚读入的 16 位数据,加入延时后再进行读取。这两个中断服务程序的配置方法和无总线扩展时候  $\overline{INT1}$  的中断服务程序的配置方法相同。

在总线扩展的场合,采用下位机预先写入警告标志的方法来减少系统中断的次数的方法依然适用。相应的软件处理和无总线扩展时介绍的方法相同。

## 5 结束语

这种基于双口 RAM 的环形数据缓存系统设计可以有效地解决上位机和下位机之间的复杂数据处理问

(下转第 208 页)

模块为备 1 模块,4 号模块为备 2 模块,5 号模块为备 3 模块,发送地址均为 DPRAM 起始地址;1 模块为接收模块,接收地址为 0x1000)和一个中断号为 16(1 号模块发)的中断命令组成。与之对应的 HDL 测试程序主体如下所示:

```
begin
...
for (i=0;i<5;i=i+1)
begin
mem_write(32'h91300004+i,256,32'h0000000A);
mem_write(32'h91400004+i,256,32'h0000000A);
end
reg_write(32'h91100000,32'h80000000);
reg_write(32'h91200000,32'h80000000);
...
for (i=0;i<5;i=i+1)
begin
@(uint_bfm.BIUx_int_out);
mem_read(32'h91100004,256);
mem_read(32'h91200004,256);
end
...
end
```

VerilogHDL 测试程序主要的过程为:在激活 5 个 LRM 模块之前,先写 256 个数据分别到 LRM2、LRM3、LRM4 和 LRM5 的 BIUx 和 BIUy 的 DPRAM 存储器模型中并分别写发送标志位,之后激活个模型;分别等 5 个中断,每个中断之后判断 LRM1 的接收数据是否分别为 2-5 模块所发出的数据。

该验证平台在应用中具有以下优点:建立了一种统一的可以配置的系统测试环境,提高了验证代码的

可重用性;可以在门级仿真和后仿真乃至不同层次的混合仿真时复用,只需要把被测模块的 RTL 代码换成门级网表或加上后仿真 SDF 文件就可以了;BFM 综合使用了直接测试、随机约束、基于断言的属性检查等多种验证方法,使得验证更直观,效率更高。

## 5 结束语

文中介绍了 ARINC659 总线接口芯片功能验证的方法、验证平台的实现以及验证测试的实施过程,提出了一种验证平台的实现方法,提高了验证的效率,缩短了验证的时间,对于类似设计的验证均有参考价值。

### 参考文献:

- [1] ARINC Specification 659 Backplane Data Bus[M]. [s.l.]:the Airlines Electronic Engineering Committee,1993.
- [2] 左航,金玉丰.一种基于 Vera 的集成电路建模验证方法[J].计算机技术与发展,2007,17(1):94-97.
- [3] 郭蒙,田泽,蔡叶芳,等.1553B 总线接口 SoC 验证平台的实现[J].航空计算技术,2008(6):99-104.
- [4] 李鸿儒,朱志强.数字集成电路验证中的模型使用[J].科学技术与工程,2008(2):45-51.
- [5] Rashinkar P, Paterson P, Singh L. System on a Chip Verification - Methodology and Techniques[M]. New York: Kluwer Academic Publisher,2002:107-115.
- [6] Rashinkar P, Paterson P. System on a Chip Verification: Methodology and Techniques[M]. [s.l.]: Kluwer Academic Publishers,2001.
- [7] 吴英攀,于立新.基于层次化验证平台的存储器控制器功能验证[J].微电子学与计算机,2009(2):69-73.
- [8] 王世好,王歆民,刘明业.嵌入式系统软硬件协同验证中软件验证方法[J].计算机研究与发展,2005,42:78-84.

(上接第 204 页)

题。通过这种缓存技术,可以最大效率地使用缓存空间,增加上位机对缓存数据处理的灵活度,简化上位机软件设计的难度,扩展总线的宽度进而提高读取速率,具有很好的推广意义。

### 参考文献:

- [1] Li Xiaotu, Sun Jizhou, Nie Weifang, et al. Design of dual port RAM for parallel volume rendering system [C]//IEEE CCECE 2004, Electrical and Computer Engineering. [s.l.]: [s.n.],2004:177-180.
- [2] 刘黎黎,马文丽,姚文娟,等.基于 ARM 和 DSP 的嵌入式 ICE 芯片扫描分析系统[J].计算机技术与发展,2008,18(8):195-198.
- [3] 郝晓莉,刘洪波,沈绪榜.异步 FIFO 中存储单元的分析设

计[J].计算机技术与发展,2007,17(3):40-41.

- [4] Zhang Jianwu, Xu Xiaorong, Li Bo. Design of Data Acquisition System on Embedded Linux with Dual Port Asynchronous RAM [C]//IEEE ICIEA 2006, Industrial Electronics and Applications. [s.l.]: [s.n.],2006:1-4.
- [5] 柳欣,尹志强,穆国宝,等.双口 RAM 非竞争模式下的并行数据通讯研究[J].合肥工业大学学报,2006,29(2):234-237.
- [6] 范立南.微型计算机原理及应用[M].沈阳:东北大学出版社,2004.
- [7] 潘新民,王燕芳.微型计算机控制技术实用教程[M].北京:电子工业出版社,2007.
- [8] 范书瑞,赵燕飞,高铁成. ARM 处理器与 C 语言开发应用[M].北京:北京航空航天大学出版社,2008.