

# 基于 MicroBlaze 的测控终端数据处理模块的设计与实现

迟东明, 陈炜炜, 臧增辉

(上海航天电子有限公司, 上海 201800)

**摘要:** 为了解决实时遥测数据挑路及浮点运算在 FPGA 上实现难度较大的问题, 设计了一种基于 MicroBlaze 的测控终端数据处理模块; 测控终端数据处理模块利用 FPGA 嵌入式处理器实现了复杂遥测数据的挑路及浮点运算, 同时利用 FPGA 的动态部分重配置技术实现了遥测数据挑路的在线配置功能; 测试表明该模块可以实现对 1553B 等具有复杂数据结构的遥测数据进行处理, 并能够实现 20 帧/秒以上的处理速度; 测控终端数据处理模块可以实现很多以往需要 ARM 或 DSP 才能够实现的复杂遥测数据处理功能; 在不增加系统硬件成本的情况下, 有效提高了系统的集成度和设计灵活性。

**关键词:** FPGA; MicroBlaze; 遥测数据挑路; 浮点运算; 动态部分重配置

## Design and Implementation of TT&C Terminal Data Processing Module Based on MicroBlaze

Chi Dongming, Chen Weiwei, Zang Zenghui

(Shanghai Aerospace Electronics Co., Ltd., Shanghai 201800, China)

**Abstract:** According to the difficulty in real-time telemetry data picking-up and floating-point calculation based on FPGA, a TT&C terminal data processing module is developed based on MicroBlaze. The module uses FPGA embedded processor to realize complex telemetry data picking-up and floating-point calculation, and uses dynamic partial reconfiguring technology to realize on-line configuration about telemetry data picking-up. The test result shows that this module can process complex telemetry data such as 1553B, and can achieve 20 frames per second processing speed. This module can realize a lot of complex telemetry data processing functions that need ARM or DSP in the past. The system integration and design flexibility are improved effectively without increasing hardware cost.

**Keywords:** FPGA; MicroBlaze; telemetry data picking-up; floating-point calculation; dynamic partial reconfiguration

## 0 引言

飞行器在飞行过程中往往会产生大量遥测数据, 地面测控站在接收、保存这些原始遥测数据的同时会有针对性的将其中某些重要遥测数据(如姿态信息、油量信息、位置信息、温度信息等)提取出来, 并进行实时处理, 显示在终端上。这些信息反映了飞行器当前的工作状态、飞行轨迹等重要信息, 对于操作者的分析、决策有重要的参考价值。为了将原始遥测数据中的指定遥测数据提取出来, 地面测控站需要对这些原始遥测数据进行挑路处理, 将实时采集到的原始遥测数据根据实际需求选取部分进行分析、计算并将处理结果发送给担负不同任务的终端, 由这些终端再做进一步处理。因此遥测数据挑路功能设计的好坏将直接影响地面测控站的性能。

一般情况下, 对于实时遥测数据的挑路处理往往采用 FPGA+ARM 或 FPGA+DSP 的方式, FPGA 主要负责接口转换、遥测帧搜索、组帧等操作, ARM 或 DSP 主要负责

复杂数据处理及系统控制。但这种实现方案的集成度不高, 不利于设备的小型化。因此, 本文提出了一种基于 MicroBlaze 的测控终端实时数据处理模块(以下简称挑路模块), 将遥测数据的挑路及遥测数据计算功能全部集成到 FPGA 中实现, 同时也提供了挑路功能的在线升级能力。这样设计不仅提高了系统的集成度及设计灵活度, 同时也降低了产品的硬件成本。

## 1 遥测数据挑路原理

该设计应用于某飞行器配套的地面测控终端设备上。地面测控终端实时接收来自飞行器测控应答机的遥测数据, 并对接收到的遥测数据进行挑路操作, 将有用数据挑出、计算, 最后将得到的数据重新打包发送给终端计算机。

飞行器发送的遥测数据是以帧为单位的脉冲编码调制(Pulse Code Modulation, PCM)码流<sup>[1-2]</sup>, 每一帧遥测数据分为不同的副帧, 各个副帧包含有来自不同传感器的飞行器状态参数, 比如气压、电压、油面、功率、图像、GPS 信息、1553B 数据帧等遥测信息。挑路模块的主要功能就是将某些重要的遥测数据从以上原始遥测数据中挑选出来并进行计算, 最后将计算结果重新打包发送给终端计算机<sup>[3]</sup>。

为了方便描述, 完整的遥测数据帧的数据结构如图 1

收稿日期: 2019-08-07; 修回日期: 2019-08-29。

作者简介: 迟东明(1981-), 男, 吉林四平人, 高级工程师, 主要从事自动化测试方向的研究。

所示。

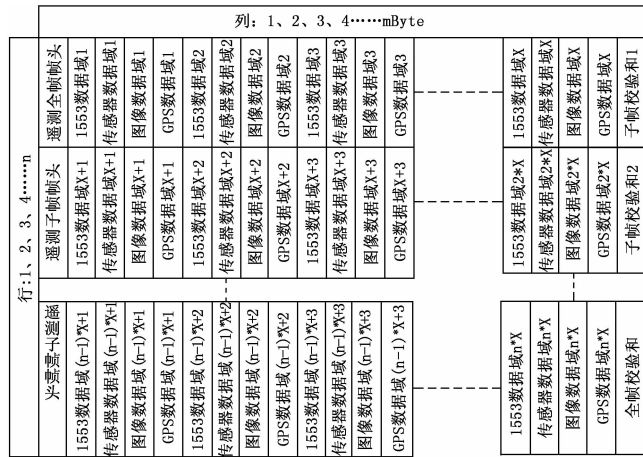


图1 完整遥测数据帧数据结构示意图

从整体看, 遥测数据帧是一个  $m$  列、 $n$  行组成的二维数据结构, 遥测数据帧的数据是按照逐行顺序传输的, 其中每一行为一个遥测子帧, 遥测子帧分为帧头、不同数据域的遥测数据及校验和。遥测子帧的长度为  $m$ Byte, 整个遥测全帧则由  $n$  个遥测子帧构成。遥测数据帧每一列为一个波道, 波道用于存放相同类型的数据。如果某些遥测数据的数据量较大, 该遥测数据就会分散存放在遥测子帧中若干个波道的数据域以及相邻多个遥测子帧的对应的波道数据域中。

以 1553B<sup>[4]</sup>数据挑路过程为例。1553B 数据帧的数据是离散存放在遥测数据帧中的 1553B 数据域中。因为 1553B 数据帧是由多个字节构成, 一个遥测子帧只能存放 1553B 数据帧的部分数据, 因此一个完整的 1553B 数据帧可能存放在相邻的多个遥测子帧中。而且由于 1553B 数据不是每一时刻都有, 因此有时候遥测数据帧并不包含 1553B 数据, 即使出现 1553B 数据, 其 1553B 数据帧的帧头在遥测数据帧中的相对位置也是不固定的, 且 1553B 数据帧的长度也并不固定, 这就给 1553B 数据帧的搜索带来了困难。

在处理 1553B 数据时就需要先进行挑路处理, 将各个遥测子帧中的 1553B 数据域中的数据提取出来, 存放在一个二维数据结构中, 如图 2 左所示。其中该表的列数  $x$  是遥测子帧中 1553B 数据域的个数, 行数为遥测子帧的个数。然后再从该二维数组中搜索 1553B 数据帧的帧头用以确定 1553B 数据包起始位置, 并将完整的 1553B 数据帧提取出来, 如图 2 右所示。最后将 1553B 有效数据提取出来并进行计算、打包、发送。

挑路处理模块有时也需要对某些遥测数据进行数据类型转换或浮点计算。涉及到的数据类型可能包含无符号整数、有符号整数、定点数、浮点数。比如, 传感器数据一般为偏移码或补码, 而传感器所测量的物理量一般为定点数或者浮点数。因此在某些情况下, 挑路模块在将挑路后的遥测数据重新打包发送给终端计算机前, 需要将某些遥测数据进行浮点运算。

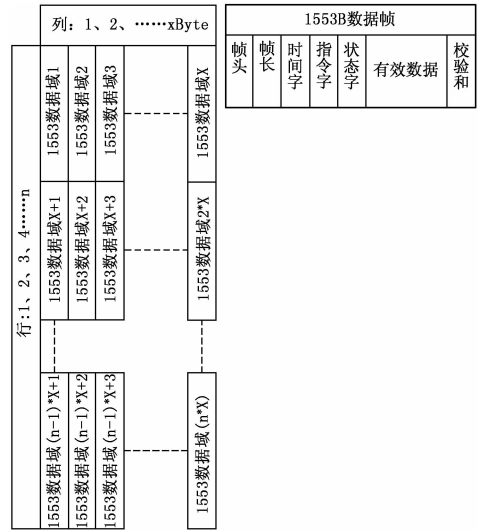


图2 挑路处理后的 1553B 数据帧结构

由以上分析可知, 使用 FPGA 实现遥测数据挑路及遥测数据处理存在以下的难点:

1) 遥测数据帧结构比较复杂, 且帧内某些遥测数据的位置及长度并不固定, 这就给遥测数据的挑路带来了困难。如果使用传统的 HDL 代码方式实现, 其实现难度及调试难度均较大。比如使用 FPGA 实现诸如 1553B 类型的遥测数据挑路, 其状态描述及控制将会变得非常复杂, 给实现及调试带来了较大困难。

2) 虽然使用 FPGA 可以进行浮点运算, 但实现难度较大, 且非常耗费 FPGA 逻辑资源。

针对以上问题, 本文提出了以 MicroBlaze<sup>[5-6]</sup>为核心的挑路模块设计方案, 用于解决 FPGA 在实现遥测数据挑路及浮点运算方面遇到的困难, 同时引入了动态部分重配置 (dynamic partial reconfiguration, 简称 DPR) 技术<sup>[7]</sup>使挑路模块具备在线重配置能力。

该挑路模块具备以下的特点:

1) 利用 FPGA 嵌入式系统实现遥测数据挑路及浮点运算, 无需外部 DSP 或 ARM 支持, 系统集成度高。利用嵌入式系统配置灵活、调试难度低的特点可以完美解决在挑路处理过程中遇到的数据挑路困难、浮点运算资源占用大等问题。通过嵌入式软件方式实现。这样不仅降低了实现难度, 也提高了开发效率。

2) 支持遥测数据处理功能的在线重配置。利用 DPR 技术, 可以针对不同的遥测数据在线下载相应的遥测帧处理模块, 从而提高了地面测控站的适应性。

## 2 基于 MicroBlaze 的挑路模块设计

### 2.1 挑路模块设计方案

该设计相关项目要求挑路模块应该具备以下的功能、性能指标:

1) 能够对最大为 96Byte (行) \* 96Byte (列) 的实时遥测数据帧进行挑路处理, 并能够对帧内指定遥测参数进

行浮点运算。

2) 能够将挑路后的结果重新打包组帧, 并发送给终端计算机。

3) 挑路模块能够处理 20 帧/秒以上的实时遥测数据帧。

4) 能够根据需要在线更改挑路策略。

为了实现以上功能, 该挑路模块的原理框图如图 3 所示。根据遥测数据处理方式的不同, 该模块主要分为两个子模块: 遥测数据预处理模块、遥测数据处理模块。

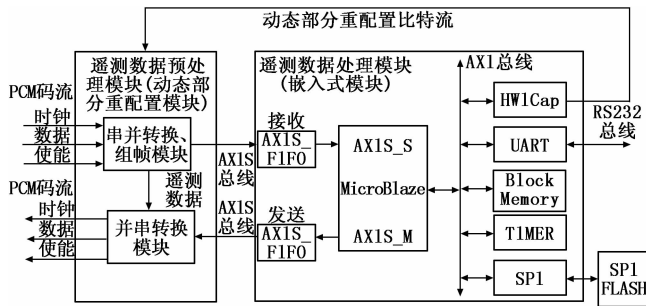


图 3 挑路模块原理框图

其中遥测数据预处理模块使用 HDL 代码实现, 具体功能包括:

5) 将接收到串行 PCM 码流转换为并行数据。

6) 完成原始遥测数据的预处理。对于并不复杂的遥测数据挑路, 遥测数据预处理模块会直接将挑路后的遥测数据组帧, 并发送给目标终端, 而对于复杂的遥测数据挑路或需要浮点运算的遥测数据, 遥测数据预处理模块会将预处理后的遥测数据发送给后端的遥测数据处理模块做进一步处理。

7) 将重新组帧后的遥测数据帧转换为 AXI Stream 总线数据流发送给遥测数据处理模块。

8) 接收从 MicroBlaze 通过 AXI Stream 总线发送过来的重新组帧后的遥测数据, 并将该数据转换为串行 PCM 码流发送给主控计算机。

同时, 该模块使用了 Xilinx 的 DPR 技术, 利用该技术可以实现遥测数据预处理模块的在线重配置功能, 可以在不影响 FPGA 其他功能模块正常运行的情况下在线配置遥测数据预处理模块, 从而实现不同的遥测数据处理功能, 这样可以有效提高遥测数据处理的灵活性。为了实现遥测数据预处理模块的 DPR 功能, 在设计前期需将 FPGA 指定区域划分给遥测数据预处理模块作为 DPR 区域, 同时在嵌入式系统中集成硬件内部配置访问接口<sup>[6]</sup> (hardware internal configuration access port, 简称 HWICAP), 以便实现 MicroBlaze 对 DPR 区域的配置。

遥测数据处理模块的具体功能包括:

1) 对遥测数据预处理模块通过 AXI Stream 总线发送过来的遥测数据做进一步处理, 如对位置及长度不固定的遥测数据进行挑路。

2) 对需要浮点运算处理的遥测数据进行浮点计算。

3) 将挑路后或者浮点运算后的遥测数据重新组帧并通过 AXI Stream 总线发送给遥测数据预处理模块。

遥测数据处理模块使用了以 MicroBlaze 为核心的 FPGA 嵌入式系统进行复杂挑路操作及浮点运算。MicroBlaze 是一款 32 位精简指令集软处理器核。该处理器在 150 MHz 时钟下可提供 125D—MIPS 的性能, 非常适合于价格敏感的网络、电信、数据通信和消费市场等领域的嵌入式应用。该嵌入式系统的本地总线主要采用了 AXI (Advanced extensible Interface) 总线<sup>[9]</sup>。AXI 是 ARM 公司提出的一种高性能、高带宽、低延迟的片内总线, 遥测数据处理模块中的各个模块均是通过 AXI 总线与处理器进行连接的。MicroBlaze 提供了若干的 AXI Stream 接口与外部数据源连接, 外部遥测数据可以通过该接口直接被 MicroBlaze 读取并处理, 为遥测数据的高效处理提供了条件。为了解决跨时钟域的问题, 该设计使用了一个双端口异步 FIFO (AXIS\_FIFO<sup>[8]</sup>) 连接遥测数据源与 MicroBlaze。遥测数据通过 AXIS\_FIFO 被 MicroBlaze 直接读取并进行处理。

同时, 遥测数据处理模块的嵌入式系统也包含有一系列其它 IP 核以保证系统的正常运行。如 SPI 模块用于 MicroBlaze 访问外部配置 Flash, 以便保存重要参数; Timer 模块用于遥测数据接收超时处理; Block Memory 用于存储 MicroBlaze 本地运行程序; UART 模块用于调试数据输出以及 DPR 数据传输; HWICAP 模块用于 DPR 功能。

## 2.2 挑路模块嵌入式软件设计

遥测数据处理模块中嵌入式软件的遥测数据挑路处理流程如图 4 所示。

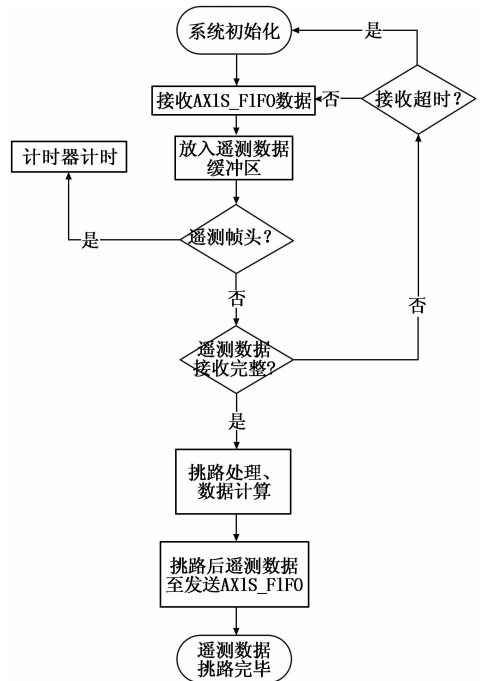


图 4 遥测数据挑路软件处理流程

大致步骤如下:

1) 挑路模块的初始化。包括接收/发送缓冲区、定时

器、关键变量的初始化。

2) MicroBlaze 不断访问接收 AXIS\_FIFO, 将其中的有效数据存入遥测数据缓冲区进行缓存, 如果接收到的数据为遥测数据的帧头, 则 MicroBlaze 启动计时器进行计时, 并继续接收数据。

3) 在接收到遥测数据帧头后, 如果在计时器计时周期内未接收到完整遥测数据, 则 MicroBlaze 初始化挑路模块, 重新接收数据; 如果在计时器计时周期内接收到了完整的遥测数据, 则进行下一步处理。

4) 在接收到完整的遥测数据后, MicroBlaze 会对存储在遥测数据缓冲区的遥测数据进行挑路处理, 同时对需要浮点运算的遥测数据进行浮点运算, 最后将处理后的遥测数据重新组帧。

5) MicroBlaze 将重新组帧后的遥测数据发送给发送 AXIS\_FIFO, 完成本次的挑路处理。

### 2.3 挑路模块浮点运算设计

对于遥测数据的浮点运算, 假设原始遥测数据  $X$  为 1 Byte 的无符号整形数, 需要经过运算并转换为单精度数据  $Y$  进行输出,  $Y$  数据位宽为 4 Byte。  $Y$  与  $X$  关系如下:  $Y = X * 5/255$ 。

如果使用 HDL 代码实现, 则需要将  $X$  先转换为浮点数, 然后使用浮点运算 IP 进行浮点运算。以上几种操作的 FPGA 资源占用情况如表 1 所示 (以 Xilinx 的 Artix7 系列 FPGA 作为硬件平台)。

表 1 HDL 实现浮点运算资源占用情况

运算	LUTs	FFs	DSP48	BRAM
浮点除法	763	1352	0	0
浮点乘法	92	166	2	0
定点转浮点	157	228	2	0

需要注意的是, 这仅仅是基于以上简单的浮点运算为例的情况, 而对于一些更为复杂的浮点运算, 其 FPGA 资源占用将大大增加。可以看出, 在 FPGA 上使用 HDL 的方式实现浮点运算, 过大的逻辑资源占用情况很难解决, 因此需要使用其他方式实现。

而 MicroBlaze 不仅支持 IEEE754 标准的浮点数据运算, 同时也支持浮点运算单元 (Floating Point Unit, FPU), 可以在很大程度上缩短浮点运算时间, 这就为 FPGA 的浮点数据处理提供了更加灵活的实现方式。

### 2.4 遥测数据预处理模块 DPR 功能设计

为了实现遥测数据预处理模块的 DPR 功能, 需要按照如图 5 所示的流程进行设计。

具体步骤如下:

1) 在 FPGA 前期设计阶段, 为遥测数据预处理模块划分出一块独立的区域用于存放模块的逻辑资源, 且该区域所包含的 FPGA 逻辑资源应该足够挑路模块使用。

2) 遥测数据预处理模块的 DPR 区域划分完成后, 需要对相应的 FPGA 工程进行编译、综合, 最后生成 DPR 比

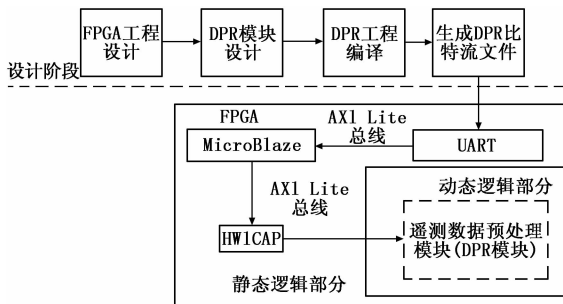


图 5 遥测数据预处理模块 DPR 功能实现流程

特流文件。在此步骤可以根据不同需求, 设计实现不同功能的挑路 DPR 模块, 以备后续阶段使用。

3) 该设计中使用了 RS232 串口作为 DPR 数据的输入接口。在工作过程中 MicroBlaze 会不断访问串口, 一旦接收到 DPR 比特流数据, MicroBlaze 会将这些 DPR 数据写入 HWICAP 模块, 直到 DPR 比特流文件传输完成。

4) 一旦 DPR 比特流文件传输完成, HWICAP 会自动复位 DPR 模块, 同时 MicroBlaze 会根据 DPR 模块的特征字选取预存的挑路程序进行匹配。

## 3 实验结果与分析

实际测试中, 该设计的 FPGA 采用了 Xilinx 公司的 XC7A200T, FPGA 开发环境使用 VIVADO, 而嵌入式软件开发平台使用了 Xilinx SDK<sup>[10]</sup>。MicroBlaze 工作时钟为 100MHz, 且配备有硬件 FPU 和专用乘法器, PCM 码流时钟为 40 MHz。

该设计的 FPGA 资源占用情况如表 2 所示, 其中包含了完整的挑路模块。可以看出, 该设计的 FPGA 资源占用率是非常低的。

表 2 挑路模块资源占用情况

Utilization	Post-Synthesis   Post-Implementation		
	Graph	Table	
Resource	Utilization	Available	Utilizatio...
LUT	4129	133800	3.09
LUTRAM	181	46200	0.39
FF	4846	267600	1.81
BRAM	5.50	365	1.51
DSP	5	740	0.68
IO	80	400	20.00
BUFG	6	32	18.75
MMCM	1	10	10.00

表 2 中使用的 5 个 DSP Slice 是用于实现 MicroBlaze 的 FPU 及乘法运算单元。同时, MicroBlaze 使用了 2 块 BRAM 作为程序存储器, 容量为 8 kByte。当需要更大容量的程序存储器时, 也可以使用更多的 BRAM 作为程序存储器。可以看出, 由于该设计中大部分的遥测数据挑路处理是由嵌入式软件实现的, 因此如果更需要实现更复杂的遥测数据挑路处理, 也仅仅需要增加 BRAM, 为嵌入式程序提

供更大的存储空间,而基本不会占用其他的 FPGA 资源。

与表 1 中 HDL 实现的浮点运算所占用 FPGA 资源进行对照,使用嵌入式系统进行浮点运算在 FPGA 资源占用率上具有更大的优势。

经过硬件测试,采用 MicroBlaze 的挑路模块可以完成实时遥测数据帧的挑路处理,且该模块最快可以处理每秒 20 帧的实时遥测数据帧,能够满足设计需求。同时,由于复杂的遥测数据挑路处理及浮点运算均是在嵌入式软件中完成的,其调试效率要远高于基于 HDL 的硬件调试,因此具有较高的设计开发效率。同时该设计也支持通过串口在线下载遥测数据预处理模块的 DPR 比特流,实现不同挑路功能的动态切换。

由于该设计中使用的嵌入式处理器是软核处理器,其主频较低,和当前主流 ARM 或 DSP 相比处理能力有限。因此如果对遥测数据的处理速度有更高的要求,可以考虑采用诸如 ZYNQ 等内部集成高性能 ARM 核的 FPGA,使挑路模块具备更强的遥测数据处理能力。

#### 4 结束语

本文提出的基于 MicroBlaze 的测控终端数据处理模块,能够在不增加硬件成本的条件下,对复杂的遥测数据进行挑路及各种浮点运算处理。相对于 FPGA + ARM (DSP) 的挑路实现方式,这种实现方式具有硬件成本低、集成度

(上接第 221 页)

#### 4 总结

针对射频识别标签防碰撞算法的研究成为了物联网技术的一个重要的研究方向。在射频识别标签防碰撞算法研究领域,虽然专家们有着很多的研究工作,但是如何使得算法具有低复杂度、效率高仍然具有很大的挑战难度。本文提出一种基于功率控制的分组算法,以达到提高防碰撞算法性能的目标。另一方面,由于功率控制的使用,使系统中的阅读器可以利用不同的发射功率来实现与不同距离的标签进行通信,所以该种方案可以在提高了系统性能的同时实现节能。

#### 参考文献:

- [1] 李泽兰,何怡刚,刘拓晟. 基于广义地址码的 RFID 防碰撞算法 [J]. 计算机测量与控制, 2010, 18 (5): 1114 - 1117.
- [2] Zhu Lei, Yum T S P. Optimal Framed Aloha Based Anti-Collision Algorithms for RFID Systems [J]. IEEE Transactions on Communications, 2011, 58 (12): 3583 - 3592.
- [3] Choi Jinchul, Lee C. An MILP-based cross-layer optimization for a multi-reader arbitration in the UHF RFID system [J]. Sensors, 2011, 11 (3): 2347 - 2368.
- [4] La Porta T F, Maselli G, Petrioli C. Anticollision Protocols for Single-Reader RFID Systems; Temporal Analysis and Optimization [J]. IEEE Transactions on Mobile Computing, 2011, 10 (2): 267 - 279.

高、调试方便、开发周期短等特点,非常适合于应用在价格敏感或集成度要求高的测控终端设备上,具有很好的推广和借鉴作用<sup>[9-11]</sup>。

#### 参考文献:

- [1] 严平. FPGA 的实时 PCM 遥测数据传输系统设计 [J]. 单片机与嵌入式系统应用, 2018, 1.
- [2] 陈建行,等. 某机载遥测原始数据处理软件设计与实现 [J]. 微处理机, 2018, 6.
- [3] 刘捷,等. 遥测数据实时挑路处理的设计与实现 [J]. 海上靶场学术, 2014 (12).
- [4] DDC. MIL-STD-1553 Designer's Guide [Z]. 1998 (6).
- [5] Xilinx. Vivado Design Suit User Guide - Embedded Processor Hardware Design [Z]. 2018 (1).
- [6] Xilinx. MicroBlaze Processor Reference Guide [Z]. 2018 (1).
- [7] Xilinx. Vivado Design Suit User Guide - Partial Reconfiguration [Z]. 2018 (1).
- [8] Xilinx. AXI HWICAP Product Guide [Z]. 2018 (1).
- [9] Xilinx. Vivado Design Suit - AXI4 - Stream FIFO [Z]. 2018, 4 (1).
- [10] Xilinx. Vivado Design Suit - AXI Reference Guide [Z]. 2017 (4).
- [11] Xilinx. Embedded System Tools Reference Manual [Z]. 2018 (1).
- [5] Li Zhonghua, Li J, He C. Artificial immune network-based anti-collision algorithm for dense RFID readers [J]. Expert Systems with Applications, 2014, 41 (10): 4798 - 4810.
- [6] Jing Shikai, et al. Optimal layout and deployment for RFID system using a novel hybrid artificial bee colony optimizer based on bee life-cycle model [J]. Soft Computing, 2017, 21 (14): 4055 - 4083.
- [7] Shahbazi Saeed, Karunasekera S, Harwood A. Improving performance in delay/disruption tolerant networks through passive relay points [J]. Wireless Networks, 2012, 18 (1): 9 - 31.
- [8] Pan Qihe, Narayanan R M. Design of a Covert RFID Tag Network for Target Discovery and Target Information Routing [J]. Sensors, 2011, 11 (10): 9242 - 9259.
- [9] Mohammed Ahmed, Wang Q. Multi-criteria optimization for a cost-effective design of an RFID-based meat supply chain [J]. British Food Journal, 2017, 119 (3): 676 - 689.
- [10] Mo, et al. RFID infrastructure design; A case study of two Australian RFID projects [J]. IEEE Internet Computing, 2009, 13 (1): 14 - 21.
- [11] Zhu Yuye, et al. Shipborne anti-collision sounding system optimization based on ACT algorithm and Internet of things [J]. Journal of Intelligent & Fuzzy Systems, 2018, 35 (4): 4173 - 4182.
- [12] Su Jian, Wen G, Hong D. A New RFID Anti-collision Algorithm Based on the Q-Ary Search Scheme [J]. Chinese Journal of Electronics, 2015, 24 (4): 679 - 683.