

无消息间隔约束的 1553B 总线通信软件设计方法

朱剑辉¹, 郑 渊², 洪小骏¹, 王志国¹, 王城浩¹

(1. 上海航天电子技术研究所, 上海 201109; 2. 上海航天技术研究院, 上海 200235)

摘 要: 介绍 1553B 总线通信软件常用的消息处理方式, 提出一种 BC 软件采用无间隔的消息序列方式批量处理各类总线消息和通过矢量字握手采集 RT 数据的设计方法, 减少 BC 软件设计约束, 提高总线消息处理效率。针对消息间隔过短导致 RT 软件丢消息的风险, 采取中断轮询消息堆栈和循环缓冲接收数据相结合的处理方式, 实现了 RT 软件对接收任意间隔消息的正确处理。分析及测试结果表明: 改进后的 BC 软件设计方式无消息间隔约束, 提高了总线消息处理效率和设计通用性, 而且避免了 RT 软件丢消息隐患, 提升了总线通信的可靠性。

关键词: 处理效率; 消息序列; 矢量字; 消息堆栈; 循环缓冲

中图分类号: TP 336; TP 311 **文献标志码:** A

DOI: 10.19328/j.cnki.2096-8655.2023.04.007

Design Method of 1553B Bus Communication Software Without Message Interval Constraint

ZHU Jianhui¹, ZHENG Yuan², HONG Xiaojun¹, WANG Zhiguo¹, WANG Chenghao¹

(1. Shanghai Aerospace Electronic Technology Institute, Shanghai 201109, China;

2. Shanghai Academy of Spaceflight Technology, Shanghai 200235, China)

Abstract: The message processing methods commonly used in 1553B bus communication software are introduced, and a design method for bus controller (BC) software is proposed to process all kinds of bus messages in batches by message sequence without intervals and to collect remote terminal (RT) data through vector word handshake, with which the design constraints of BC software are reduced and the efficiency of bus message processing is improved. In view of the losing message risk of RT software due to too short message intervals, a method combining interrupt polling message stack and circle buffering received data is adopted to realize the correct processing of receiving messages at any interval by RT software. The analysis and test results show that the improved BC software design mode has no message interval constraint, improves the bus message processing efficiency and design versatility, avoids the hidden danger of RT software message loss, and improves the reliability of bus communication.

Key words: processing efficiency; message sequence; vector word; message stack; circular buffer

0 引言

MIL-STD-1553B 总线最初是由美国在 20 世纪 70 年代提出的航空电子系统联网标准, 具有高可靠性、抗干扰能力强、扩充和维护简便等特点, 因而被广泛应用于航空和航天等领域^[1-2]。1553B 总线的传输协议为命令/响应方式, 数据传输的最小单位为“消息”, 每条消息最多 32 个字 (16 bit 位宽), 所有的消息传输均由总线控制器 (Bus Controller, BC) 发出命令, 远程终端 (Remote Terminal, RT)

响应^[3-4]。

1553B 总线协议规定有 32 个 RT 子地址用于消息传输。BC 作为 1553B 总线的控制端, 控制所有子地址消息的传输, 包括方式代码传输、BC 与 RT 之间数据传输、RT 与 RT 之间的数据传输和 BC 向所有 RT 设备广播数据等^[3-6]。RT 作为总线响应端, 通过各子地址控制字配置成单消息、双缓存和循环缓冲 3 种数据收/发缓存模式, 以及相应的消息结束中断方式用于响应总线数据传输, 通过各子地址查

收稿日期: 2023-04-18; 修回日期: 2023-04-30

作者简介: 朱剑辉 (1989—), 男, 硕士, 高级工程师, 主要研究方向为航天星载计算机软件和下位机软件开发。

找表配置消息数据收/发缓存地址^[7]。

受限于下位机设备的处理能力,通常 BC 软件需要严格控制与各下位机 RT 软件通信的周期及消息间隔,避免因消息响应或处理不及时导致通信异常。主要包括以下 2 个方面:1) RT 软件在 1553B 总线中断时通过读取命令字寄存器获取当前子地址信息。若 BC 发送不同的子地址消息的间隔小于软件响应消息结束中断到读取命令字寄存器的最大延时,将导致消息命令字被覆盖异常^[8]。2) RT 软件在 1553B 总线中断时设置数据处理标志,主流程判断标志有效后再执行。若 BC 向 RT 发送同一个子地址数据间隔小于 RT 软件最大处理延时,将导致接收的数据被覆盖异常。同理,若 BC 从 RT 同一个子地址采集数据的间隔小于 RT 软件参数更新最大延时,将导致采集的参数半新半旧或者与上一帧重复。

针对上述限制,当 BC 向某个 RT 设备发送多条遥控指令时,需要 BC 软件设计指令缓存,控制周期逐条发送;当 BC 从各 RT 采集遥测参数时,BC 软件也要控制不同采集周期;BC 软件为了保证各 RT 的响应时间,所有消息设置统一的消息间隔,或增加软件延时,如每次只启动单个 RT 设备的一种消息传输,等待 RT 返回的数据或消息反馈后才继续准备后续消息。以上约束严重影响了 BC 软件的设计通用性和总线传输效率与可靠性^[9-10]。为此,文中设计了一种通用的 1553B 总线消息处理方式,消除 BC 软件传输消息间隔限制,减少 BC 软件的设计约束,并解决 RT 软件对任意消息间隔的适应性问题。

1 BC 软件设计

1.1 BC 软件常规设计方式

BC 软件根据消息传输类型、传输总线 A/B、RT 地址、RT 子地址、数据内容及长度等信息,构造消息命令字和控制字,按消息格式写入总线缓存^[6];然后在堆栈中构造 4 个字的消息描述符,写入消息间隔和消息缓存起始地址(块状态字和时间标签由芯片自动填写);接着写总线消息个数配置字和启动寄存器,启动消息传输;最后等待消息传输结束,读取状态字反馈及 RT 返回数据等信息。以 BC 向 RT 传输数据为例,消息数据存储格式如图 1 所示。

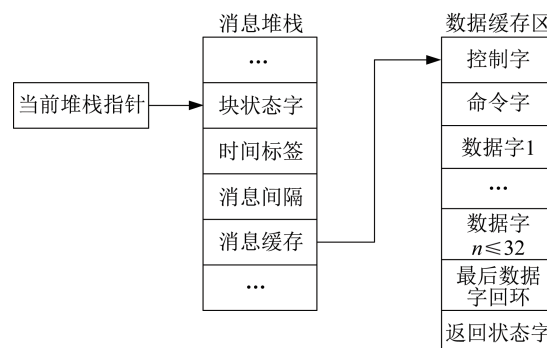


图 1 BC->RT 消息格式

Fig. 1 Diagram of the BC->RT message format

1.2 改进后的 BC 软件设计方式

根据 1553B 总线芯片 BC 模式的特性,内部缓存配置区存在 A/B 区冗余设计,可通过配置寄存器 1 的 bit 13 配置当前的工作区。当指定 A 区为当前工作区时,当前总线活动由 A 区的配置空间控制,包括设置消息堆栈和消息个数配置字等;与此同时 B 区作为非活动区,可以由软件随意操作而不影响当前总线芯片工作。因此,当 A 区在传输总线消息时,可以同时 B 区所指向的缓存空间准备后续消息;当 A 区消息传输结束时,通过配置寄存器 1 切换当前的活动区,B 区准备好的消息就可以直接启动数据传输。BC 可一次性启动最多 512 条消息的序列传输,消息个数配置字中写入相应的值并启动传输后,芯片就会依次从消息堆栈中往后解析各消息描述符及其所指向的缓存区中的消息格式,并启动相应的消息传输。

基于上述特性,在芯片内部缓存空间允许的情况下,根据总线应用协议,合理划分总线消息序列大小。BC 软件一次性启动当前消息序列传输,同时在非活动区准备后续总线消息序列;当前一个消息序列传输结束后,在处理相应的消息反馈数据前,切换总线活动区,启动后一个消息序列传输,如此周期循环。通过以上改进,不仅节约了软件等待数据传输时间,并且不设置消息传输间隔,充分发挥总线芯片的数据传输效率。

BC 软件向各下位机发送总线消息时,不控制相同子地址数据的传输周期,例如当某个需要发送多条遥控指令时,不做数据缓存,一次性批量发送,由各下位机 RT 软件自行缓存处理。BC 软件从各下位机采集数据时,采用下位机举手的方式握手:RT 准备好数据后举手,BC 软件判断下位机举手时

才采集相应的数据,数据采集周期由 RT 软件举手的周期来控制。通过以上改进,消除了 BC 软件与各下位机 RT 软件通信周期的约束。

2 RT 软件设计

2.1 RT 软件常规设计方式

作为总线响应端,RT 软件在响应总线消息传输前,首先需根据总线应用协议配置好各 RT 子地址控制字和查找表。在收到消息中断后,从消息命令字寄存器读取当前消息的命令字,获取 RT 子地址和数据长度等信息,设置相关数据处理标志。最后再在软件主流程判断相关数据处理标志有效后,按照总线应用协议要求进行处理,如果是接收消息结束中断,则读取接收数据并处理;如果是发送消息结束中断,则接着准备填写下一帧数据等待 BC 获取。RT 软件中断方式处理流程如图 2 所示。

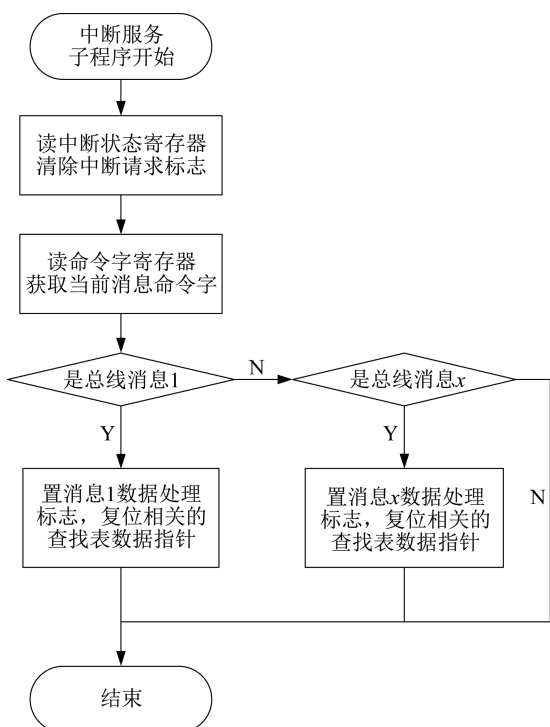


图 2 RT 软件中断方式处理流程

Fig. 2 Processing flow diagram of the interrupt mode by the RT software

2.2 中断遍历消息堆栈方式

根据 1553B 总线芯片 RT 模式的特性,在缓存中有 256 字的消息堆栈,依次存储接收的 4 个字的

总线消息描述符^[7-8]。因此,只要 RT 软件在两次读取消息堆栈之间收到的消息不超过堆栈大小,那么就可以通过遍历堆栈的方式将接收到的消息命令字逐个读出而不被覆盖。改进后的 RT 软件中断处理流程如图 3 所示。

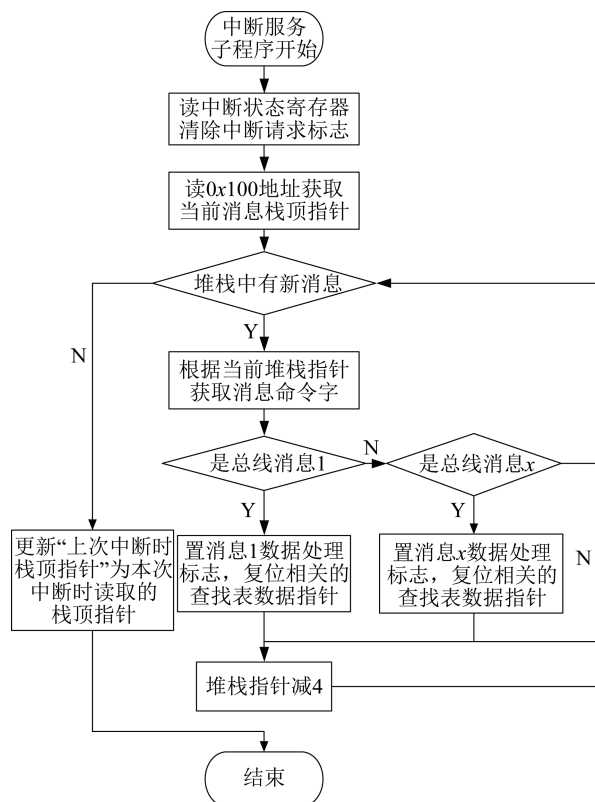


图 3 改进后的 RT 软件中断方式处理流程

Fig. 3 Processing flow diagram of the interrupt mode by the improved RT software

2.3 循环缓冲方式接收单消息数据

根据 RT 子地址配置成单消息时查找表指针会自主复位而循环缓冲模式自动向后移动并回卷的特性,为了避免消息数据处理不及时被覆盖,改进后的 RT 软件将接收子地址由原单消息接收和消息结束中断模式,改成循环缓冲接收和消息结束中断模式。软件随着接收消息查找表指针的变化记录该子地址在循环缓冲中的“当前接收数据首地址”;当接收该消息产生中断时,中断服务子程序设置数据处理标志,主流程判断标志有效后,从“当前接收数据首地址”开始处理数据,根据相关特征判断当前消息是否有效,若有效,则读取相应的数据并处理,然后将缓存区中的消息特征设

置为无效,并继续向后读取数据,直到读到当前消息为无效为止,并更新“当前接收数据首地址”为下一个地址。为了增强软件可靠性,每次数据处理前先判断“当前接收数据首地址”是否合法,若

非法则重新配置子地址查找表指向循环缓冲首地址。以接收 32 个字的消息(第一个字用作有效标志)配置成 128 字循环缓冲为例,软件接收消息数据时缓冲区状态变如图 4 所示。

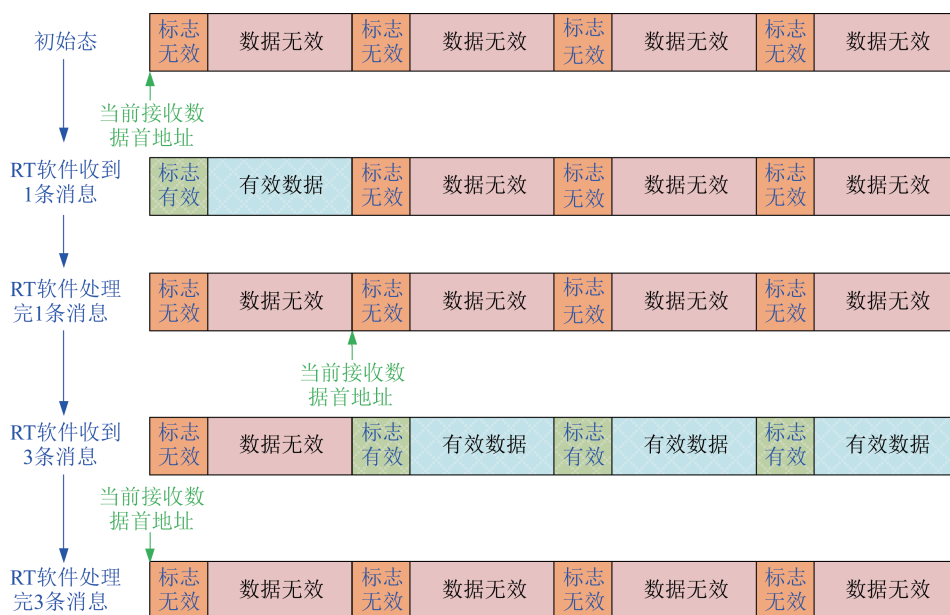


图 4 接收消息循环缓冲区状态变化

Fig. 4 State change diagram of the receiving messages in the circular buffer

2.4 矢量字握手

受限于下位机软件的处理能力或参数状态更新下行周期的约束,BC 软件并不能随意的采集 RT 软件的数据。为了消除该限制,BC 向 RT 软件采集参数时,采用方式代码“发送矢量字”握手,矢量字每个位可以分别表示不同类型的数据举手状态。RT 准备好数据后按协议设置相应的矢量字位,BC 软件每个周期通过“发送矢量字”方式代码获取下位机的举手状态,当判断下位机举手后从相应的子地址采集数据,RT 判断 BC 将数据读完后,清除矢量字位,并开始下个周期参数采集更新。BC 与 RT 通过矢量字方式握手时序如图 5 所示。

3 测试结果及分析

对改进前后的软件在某型号综合电子计算机和下位机平台开展对比测试。首先综合电子计算机 BC 软件采用无消息间隔的消息序列方式与改进前的 RT 软件周期通信,周期采集 10 000 条遥测消息(采集周期 500 ms,RT 软件最大主流程周期

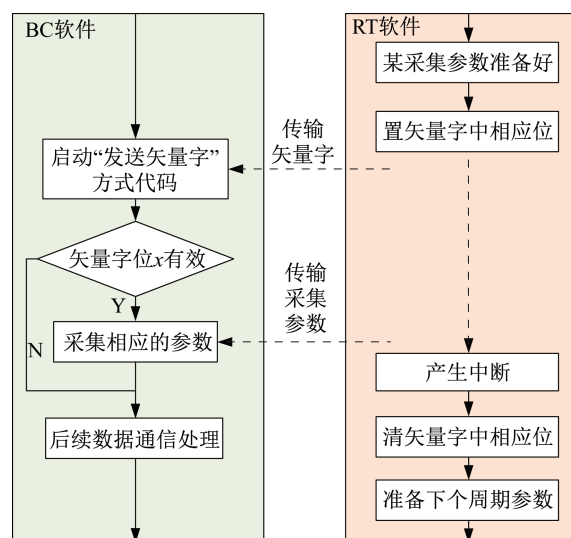


图 5 BC 与 RT 矢量字握手时序

Fig. 5 Diagram of the word handshake timing of the BC and RT vectors

约 600 ms),通过地面测试软件随机注入 1 000 条遥控指令(一次注入 1 条或多条),长度均 32 个字。测试结束后,通过遥测参数查看 RT 软件接收指令计数为 815,即出现了丢指令异常;遥测参数出现

112 次帧计数不连续异常、23 次累加和校验错误,表明遥测参数出现了遥测帧重复和半新半旧异常。改成采用无消息间隔的消息序列、矢量握手方式和改进后的 RT 软件通信,重复上述测试过

程,未发现丢指令异常,也未发现遥测帧计数和校验异常。测试结果表明,文中提出的设计方式达到预期效果。改进前后的软件测试结果对比见表 1。

表 1 改进前后的软件测试结果对比

Tab. 1 Comparison of the software test results before and after improvement

测试软件情况	遥测帧计数不连续计数	遥测帧校验错误计数	遥控指令丢指令计数
无消息间隔的 BC 软件与常规 RT 软件测试	112	23	185
无消息间隔的 BC 软件与改进后的 RT 软件测试	0	0	0

4 结束语

文中通过对常用的 1553B 总线通信软件消息处理方式进行分析,引发出对提升软件通信效率和设计通用性的思考,提出相应的改进措施,分析并解决了由此引入的风险,最终形成了一种高效、通用且可靠的设计方式。文中的分析及设计思路对复杂系统 1553B 总线通信软件的设计和异常问题排查均有一定的参考借鉴意义,对其他类似串行总线通信同样具有启发意义。

参考文献

[1] 凌志华,谭立英,陶坤.基于 BU-65170 的 1553B 终端设计[J].电子设计,2008,24(1):273-274.
[2] 马捷中,陆艳洪,翟正军.1553B 总线控制器和远程终端软件设计[J].测控技术,2003,22(2):38-40.
[3] 安然,罗小成,冯剑锋,等.基于 1553B 的卫星通用测试

平台的设计与实现[J].现代防御技术,2020,48(2):81-88.
[4] 张娟娟,李瑞婷.基于 Linux 系统的 1553B 驱动设计[J].数码设计,2017,6(7):11-13.
[5] 赵爽,刘占渠,苗克坚.通用 1553B 总线消息解析软件的设计与实现[J].电子测量技术,2010,33(1):78-81.
[6] 张玉.1553B 总线远程终端建模[D].哈尔滨:哈尔滨工业大学,2013.
[7] 杨芳.基于 BU61580 的 1553B 总线远程终端软件设计[J].黑龙江科技信息,2009,(30):39-40.
[8] 郭坚,刘欣,毛亮.1553B 总线终端消息覆盖的原因及解决方案[J].航天器工程,2011,20(5):88-92.
[9] 桂亮,罗洁,陈晓,等.运载火箭 1553B 总线拓扑反射计算的应用研究[J].上海航天,2016,33(1):102-106.
[10] 杨敏,何志壮,谈寅,等.嫦娥五号轨道器 1553B 总线网络设计[J].上海航天,2022,39(1):33-39.