

## 【自动化技术】

## 1553B 分层总线信息传输延时产生机理及解决措施

奚国权, 杨 芳, 邵志刚

(中国空空导弹研究院, 河南 洛阳 471009)

**摘要:**对 1553B 分层总线信息传输延时产生的原因进行了详细分析,并设计了一种软件实现方法,使传输延时时间控制在系统要求的范围内。

**关键词:**武器系统;信息延时;1553B 总线

**中图分类号:**TJ762

**文献标识码:**A

**文章编号:**1006-0707(2011)08-0092-02

某载机系统通过 1553B 总线把各种飞行参数信息传递给分系统,分系统把该飞行参数信息经过处理、再通过 1553B 总线转发给下属的子系统,这样就构成了两级分层 1553B 总线。

分系统在处理、转发飞行参数信息时相对于载机系统会出现一定的传输时间延时,如果该延时时间过大或不准确都将对飞行参数信息带来影响,无法反映当前载机、目标的状态<sup>[1-2]</sup>。因此,必须对这个延时时间产生的各方面原因进行分析并使之符合系统要求。

## 1 分层总线传输

### 1.1 系统描述

载机通过 1553B 总线和分系统互联,分系统经过处理任务信息,通过下一层 1553B 总线和子系统互联,系统互联关系见图 1。

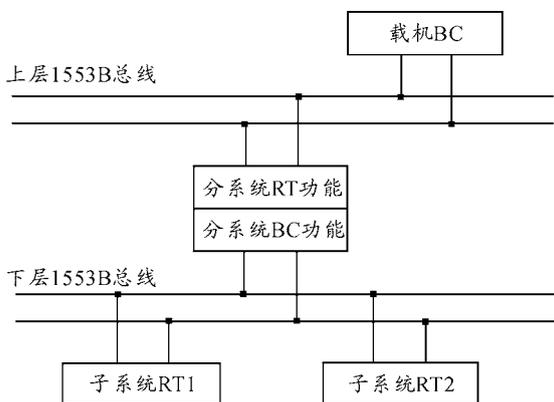


图1 系统互联关系

载机航电系统采集到的飞行参数信息(包含载机、目标的信息),通过总线控制器(BC)传递给分系统,此时分系统对载机而言作为远程终端(RT),分系统经过对信息进行处

理,同时作为下层总线的 BC,将飞行参数信息传递给各个子系统 RT。

### 1.2 信息延时时间定义

整个系统的飞行参数信息延时时间 TOLAU 定义为:从载机 BC 发出总线时,到分系统发送完总线所占用的时间。

一般载机系统根据飞行参数的精度规定了 TOLAU 不大于某个值  $\Delta T$ (例如 20 ms),如果超出了该值,总线上传输的飞行参数(载机、目标状态)可信度降低,精度无法保证,对于某些精确制导武器,可能造成无法正常工作。

## 2 信息传输延时分析

图 2 详细说明了分系统飞行参数信息处理时间的各个环节<sup>[3-4]</sup>。

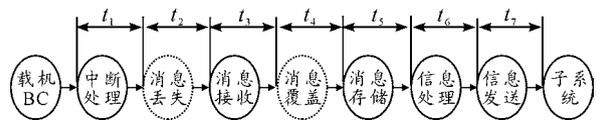


图2 飞行参数信息处理的各个环节

分系统的信息延时时间计算公式为

$$TOLAU = t_1 + t_2 + t_3 + t_4 + t_5 + t_6 + t_7$$

式中: $t_1$ 为 RT 消息中断处理的时间; $t_2$ 为 1553B 飞行任务消息丢失信息延时时间; $t_3$ 为 1553B 飞行任务消息接收处理的时间; $t_4$ 为 1553B 飞行任务消息覆盖信息延时时间; $t_5$ 为 1553B 飞行任务消息数据存储处理时间; $t_6$ 为 1553B 总线格式飞行参数信息处理时间; $t_7$ 为飞行参数信息发送给子系统的时间。

### 2.1 远程终端 RT 中断处理时间 $t_1$

在  $t_1$  时间环节中,分系统 1553B 接口芯片在接收到一个消息的命令字后,由芯片电路自动完成该消息的处理,最后产生一个外部中断信号。这个过程由软件和硬件配合完成,所需时间较短。CPU 响应处理外部中断的方式有 2 种:中断和查询。中断方式延时较少,而查询方式延时较长。

## 2.2 1553B 消息丢失延时时间 $t_2$

当 1553B 总线通讯频繁,中断服务程序执行的时间较长,或者消息之间间隔较小,都有可能发生消息累积、消息未处理的问题,造成信息丢失延时时间  $t_2$  的产生。

## 2.3 消息接收时间 $t_3$

分系统完成飞行参数信息的数据接收时间。获得当前消息的块状态字和消息块地址,对消息的块状态字进行故障判别,若处理器每个指令周期执行时间为 50 ns,则  $t_3$  为微秒级别。

## 2.4 消息覆盖延时时间 $t_4$

当 1553B 总线通讯频繁,中断服务程序执行的时间较长,或者消息之间间隔较小,都有可能发生消息累积、消息未处理的问题,造成信息丢失延时时间  $t_2$  的产生。

## 2.5 消息存贮时间 $t_5$

接收到飞行参数信息后,分系统处理器读取并存储数据的时间为  $t_5$ ,软件运行时间为微秒级别。

## 2.6 信息处理时间 $t_6$

完成飞行参数信息块处理、转换为子系统需要的飞行参数信息数据块。

由于飞行参数信息数据量多,处理全部为位运算方式,软件执行时间较长。以某分系统产品为例实测,  $44 \times 16 = 704$  位数据信息处理时间  $t_6$  为 3 ms 以内,且这个时间为固定值。

## 2.7 消息存贮时间 $t_7$

信息发送的时间  $t_7$  的计算公式为

$$t_7 = n \times m \times t_s。$$

其中:  $n$  为向子系统发送的总线数据个数;  $m$  为每一个总线数据的位数;  $t_s$  为发送一位数据占用时间(位时)。

每个系统的总线数据格式是固定不变的,包括个数和位置意义,1553B 总线上传输的位数也是固定的,1553B 总线数据波特率为 1 M,  $t_s = 1 \mu s$ ,故  $t_7$  为固定值。

## 2.8 分析结论

由上述分析可看出:信息延时时间 TOLAU 的 7 个环节中,  $t_3$  和  $t_5$  对信息延时时间 TOLAU (20 ms) 的影响可忽略。  $t_6$  和  $t_7$  是运行环境系统的固有特性,是难以进行优化设计的。因此 TOLAU 为

$$TOLAU = t_1 + t_2 + t_3 + t_4 + t_5 + t_6 + t_7 \approx$$

$$t_1 + t_2 + t_4 + t_6 + t_7 = t_1 + t_2 + t_4 + T(\text{固定})$$

影响 TOLAU 仍有 3 个时间环节  $t_1$ 、 $t_2$ 、 $t_4$  是需要设计解决的问题,任何一个处理不当都会造成 TOLAU 的超标。

# 3 设计方法

## 3.1 RT 消息中断处理 ( $t_1$ )

RT 消息处理可采用中断方式或查询方式。

如果采用主程序查询方式,受主程序大周期循环处理的影响,无法保证飞行任务信息的实时处理,如:主程序中 I/O 滤波程序运行时间约为 9 ms,即  $t_{1,max} = 9 \text{ ms}$ 。

如果采用中断方式,  $t_1$  非常小,但由于需要处理数据,运行时间较长(达到毫秒级别),会对主程序和其他中断程序产生影响,且 1553B 消息会产生累积,发生丢失消息故障。

比较 2 种处理方式,为了保证 TOLAU 的要求,采用中断

方式处理。主程序控制逻辑时序的延时效应,通过合理的时间参数容差(最少几十毫秒)来满足。将 RT 中断确定为高优先级,在信息处理完成后启动对子系统总线发送。采用这种设计方法,  $t_1$  将会是微秒级的,对 TOLAU 可忽略不计。

## 3.2 1553B 消息堆栈处理 ( $t_2$ )

分系统 RT 接收到消息后,会在堆栈里存储信息,大多数情况下,中断服务程序中只会处理一个消息。处理完毕后,需要将堆栈指针清零指向 0x0 地址单元(否则会发生堆栈溢出问题)。但当正在处理当前消息时,RT 又接收了消息,等待处理。而堆栈指针已清零指向 0x0 地址单元,这样就丢失后面的消息。

对于堆栈描述可这样设计,在堆栈空间的某一个地址设置为堆栈的上限,如 0x40 (10 个消息)。进入中断处理程序后,首先读取堆栈指针地址,若大于 0x40 时,堆栈指针清零;若不大于 0x40 时,则堆栈指针不清零。即每次进入中断,会处理所有未处理的消息(包括上次或以前累积的),处理流程如图 3 所示。采用这种设计方法,就能够保证不会丢失任何一个消息,1553B 消息丢失延时时间  $t_2$  为零。

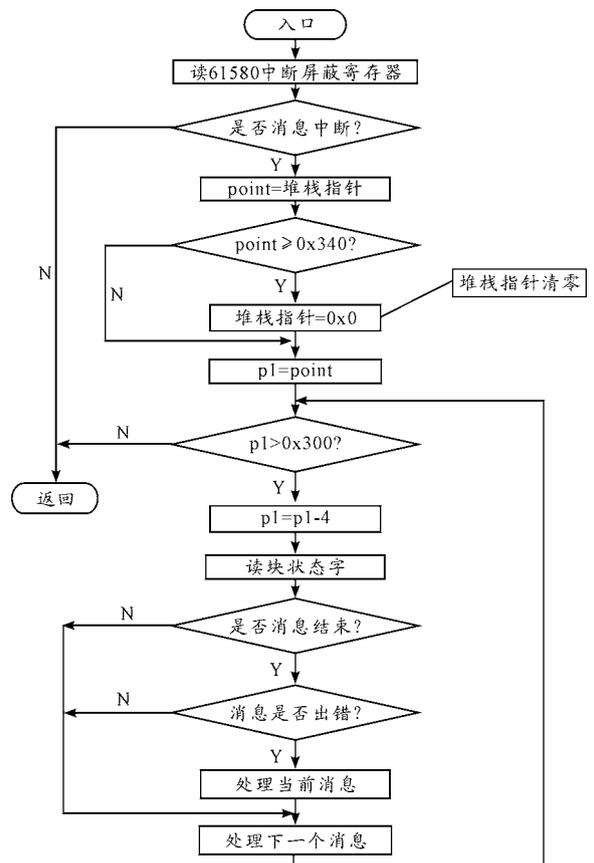


图 3 1553B 消息堆栈处理流程

## 3.3 飞行任务消息覆盖问题 ( $t_4$ )

分系统接收到飞行参数信息后,开始处理、转换为子系统格式数据。如果在信息处理过程中,又接收到了 1553B 格式的飞行任务消息数据,将会把当前未处理信息数据覆盖为下一个周期的数据,造成飞行任务消息错误, TOLAU 减小。

解决的方法是:将接收到的飞行参数信息备份存储为另一数组,并设置数据刷新标志。信息处理(下转第 116 页)