

## 1553B 总线即插即用解决方案\*

吕良庆<sup>1,2</sup>, 安军社<sup>1</sup>

(1. 中国科学院 国家空间科学中心 复杂航天系统电子信息技术重点实验室, 北京 100190;

2. 中国科学院大学, 北京 100049)

**摘要:**在传统 1553B 总线通信协议的基础上,通过增加初始握手过程和远置终端电子数据单的设计,形成具有即插即用特征的通信协议。基于这个协议,给出 1553B 总线控制器的即插即用设计方案,包括远置终端加入的总线表构建算法、管理数据结构和总线表动态调度的运行控制方式。总线表构建算法采用基于电子数据单的方法,根据远置终端的通信需求进行分析、计算、转换得到新的总线表。数据结构的思路是将总线表内容的分析提前到初始化时完成,以提高总线运行时的管控和数据交换的效率,以及方法的统一。总线表调度采用主总线表加插入消息的方式。对以上综合解决方案进行时间和适用场合的效果评估,结果表明该方案形成的总线周期、远置终端接入时间长度等指标达到了可以接受的程度,既保持了总线周期的稳定,又提供了即插即用所需要的控制灵活性。具有即插即用特征的 1553B 总线在 CCSDS - SOIS 架构中作为底层子网之一,为上层即插即用系统的构建提供了支撑。

**关键词:**1553B 总线;即插即用;航天器接口业务;电子数据单;空间数据系统

**中图分类号:**TP393.1 V443 **文献标志码:**A **文章编号:**1001-2486(2019)02-082-09

## Plug and play resolution on 1553B data bus

LYU Liangqing<sup>1,2</sup>, AN Junshe<sup>1</sup>

(1. Key Laboratory of Electronics and Information Technology for Space Systems, National Space Science Center, Chinese Academy of Sciences, Beijing 100190, China;

2. University of Chinese Academy of Sciences, Beijing 100049, China)

**Abstract:** Based on the traditional 1553B data bus protocol, a new protocol with plug and play features was designed by adding remote terminal EDS (electronic data sheet) and procedure of transferring EDS. According to this protocol, the plug and play resolution of 1553B bus controller was designed, including constructing a bus list algorithm with the remote terminals plug in, management data structure, and dynamic operation and control bus list method. The constructing bus list algorithm uses remote terminal EDS to analyze, calculate and transform an old bus list to a new one, to satisfy remote terminals' communication requirements. A management data structure is used for analyzing and storing bus list information during initialization phase, to increase efficiency of operation control and data exchanging. Dynamic operation and control bus list method adopts main bus list with plug in messages method. The timeliness and use cases of the resolution were evaluated. The results show that its time of bus cycle and time of remote terminal plugged into the system can be accepted. The resolution can remain stable in the communication process, and provide flexibility for plug and play needs. In CCSDS - SOIS architecture, 1553B data bus is one of the subnetworks on the bottom layer. With plug and play features, it can be extended to support constructing plug and play system in higher layer.

**Keywords:** 1553B data bus; plug and play; spacecraft onboard interface services; electronic data sheet; space data system

MIL - STD - 1553B 总线在国内外航天工程领域已经得到了广泛的应用,适用于低速、高可靠的数据传输,通常用于系统管理和控制。为此各航天组织也依之制定了各自的 1553B 通信协议,例如欧洲空间局(European Space Agency, ESA)的 ECSS - E - ST - 50 - 13C<sup>[1]</sup> 和我国的 GJB 289A - 97<sup>[2]</sup>,以及为不同通信形式而改造的

1553B 总线协议<sup>[3-5]</sup>。这些通信协议及其应用情况普遍有一个特点,就是参与通信的远程终端(Remote Terminal, RT)和传输的消息是事先经过协商确定好的,采用固定消息时序分配的方式通信。这种做法能够满足不同项目任务广泛的用户应用需求,但是对于开发方和系统构建过程而言,其协调、设计、验证工作是比较烦琐的。

\* 收稿日期:2017-10-30

基金项目:空间科学先导专项总体基金资助项目(XDA04080000)

作者简介:吕良庆(1969—),男,北京人,博士研究生,E-mail:lvliangqing@nssc.ac.cn;

安军社(通信作者),男,研究员,博士,博士生导师,E-mail:anjunshe@nssc.ac.cn

空间数据系统咨询委员会 (Consultative Committee for Space Data System, CCSDS) 的航天器接口业务 (Spacecraft Onboard Interface Services, SOIS)<sup>[6-9]</sup> 采用了即插即用的思想,系统架构底层可以接入多个异构子网,1553B 总线只是其中之一。架构上层的应用和业务过程对这些子网的访问是透明的,因此设备级的即插即用过程首先要子网内部来解决。

### 1 即插即用 1553B 总线需要的设计基础

1553B 总线通信涉及总线控制器 (Bus Controller, BC) 的应用部分 BCA 和服务部分 BCS、RT 的服务部分 RTS 和应用部分 RTA 共 4 个角色,它们之间的通信关系见图 1。

在 ECSS - E - ST - 50 - 13C 中归结了 5 项 1553B 总线服务,即时间服务、通信同步服务、分发和获取 - 置数和取数服务、数据块传输服务和终端管理服务,以原语方式说明了各项服务的交换信息和时序安排。同时给出了子地址编排、方式码消息使用建议等。

BCS 和 RTS 的实体中包括 1553B 总线协议芯片作为高级通信引擎 (Advanced Communication Engine, ACE),典型的如 DDC61580 系列<sup>[10]</sup>。ACE 实现了链路层协议,并且配有相应的配置寄存器和消息数据缓存区。

1553B 总线通信通过编排和运行总线表实现。为适应不同消息的传输需求和消息时间间隔 ( $T_{ITV}$ ) 要求,总线表编排采用消息服务请求方式,划分为多个消息小帧,每个小帧以查询服务请求的矢量字消息开头,以同步消息 (表示为 Syn) 结尾,中间为需传输的消息。每个小帧周期 ( $T_{FRM}$ ) 相等。如果某个消息不需要传输,则该消息不上线,但是时间仍然占用,以保证  $T_{FRM}$  的稳定。这样组织的总线表称为主总线表,其中  $T_{FRM}$  表现为总线周期 ( $T_{BUS}$ ),即二者在稳定运行期间是相等的,只在总线表重构过程中有所区别。

在主总线表运行期间,BC 可通过切换总线表的方式,改变消息序列的运行。这种运行控制方式称为插入消息方式,可作如下约定:

- 1) 每次插入的消息只能运行一次。
- 2) 插入消息运行完毕后,如果有新的插入消息请求,可继续插入消息运行,如果没有则返回主总线表断点处继续运行。

3) 插入消息采用帧结束 (End Of Frame, EOF) 处理方式,先停止总线消息运行,对消息帧中的所有消息进行一次性处理,切换总线表,再重新启动运行。因此插入消息序列中的消息时间 ( $T_{MSG}$ ) 可按照最小时间设置,不必留有余量。

上述各种时间概念和关系示意图 2。

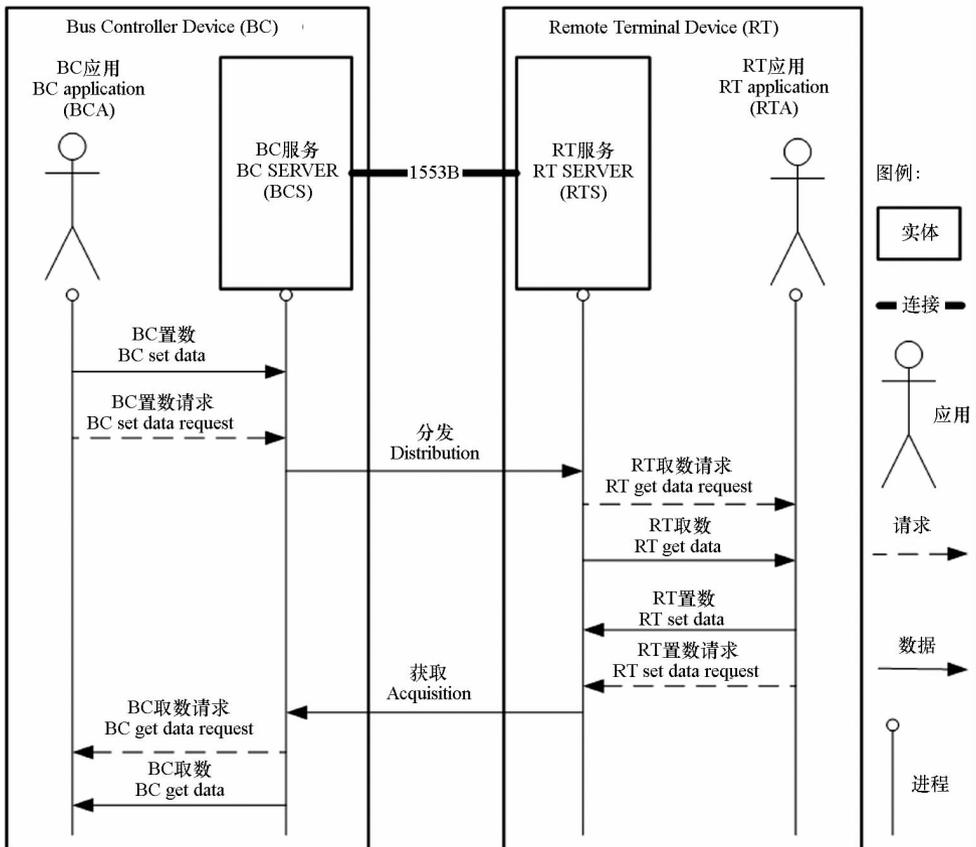


图 1 1553B 总线通信顺序

Fig. 1 1553B bus communication sequence diagram

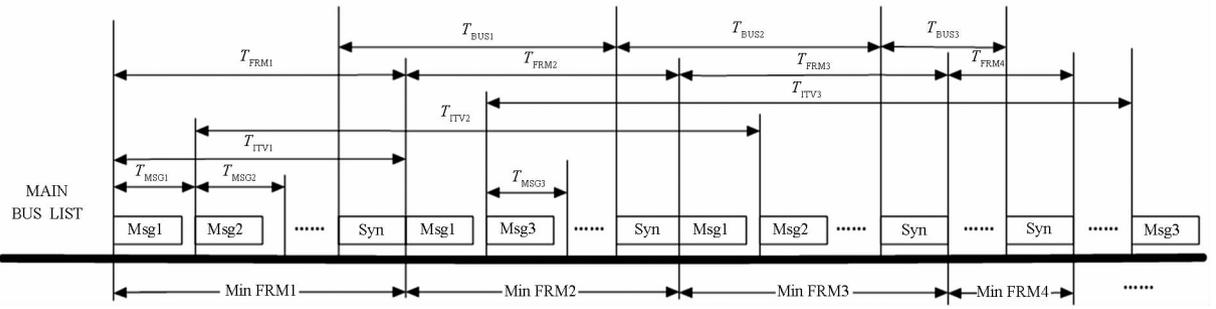


图 2  $T_{MSG}$ 、 $T_{ITV}$ 、 $T_{FRM}$ 、 $T_{BUS}$  概念示意  
Fig. 2 Concepts of  $T_{MSG}$ 、 $T_{ITV}$ 、 $T_{FRM}$ 、 $T_{BUS}$

主总线表加插入消息方式的总线运行控制示意图见图 3。

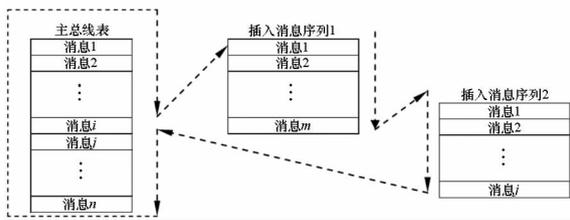


图 3 主总线表和插入消息运行的转换关系  
Fig. 3 Switch relationship between main bus list and insert message list

工程项目中的一般做法是 BC 方还需要与各 RT 方协调,进行 RT 地址和子地址(消息类型)的分配,定义消息长度和数据格式,确定消息传输时间和时序,形成项目级的 1553B 总线通信协议和接口控制文件。

## 2 即插即用 1553B 总线通信协议

RT 方的设计是固定而简单的,不能因为增加了即插即用的概念而造成 RT 方的设计复杂,而 BC 方应对 RT 方设计的多样性和针对性具有自适应能力,这是即插即用系统应该具有的特征。但是还是需要 RT 方至少将其传输需求告知 BC 方,BC 才能进行通信管控。为此需要增加 RT 接入时的握手过程,以传递 RT 的电子数据单(Electronic Data Sheets,EDS),供 BC 动态构建总线表。

### 2.1 RT 子地址的使用约定

为了建立初步的握手关系,需要对子地址的分配进行一些固定的约定。RT 的子地址在 ECSS - E - ST - 50 - 13C 基础上约定见表 1。

表 1 中 RT 可自行配置的子地址允许继承使用已有的配置,并事先与 BC 方约定。如果不约定,则 BC 根据 RT 的 EDS 动态编排总线表,以主动适应不同 RT 的不同传输需求实现即插即用的特征。

表 1 RT 子地址使用约定

Tab. 1 RT subaddress allocation

子地址	RT 发送	RT 接收	说明
0	不用	不用	
1 ~ 25	获取数据块	分发数据块	与 ECSS - E - ST - 50 - 13C 一致,RT 可自行设置使用
26	EDS 发送块	数据注入	EDS 内容根据需要定义
27 ~ 30	.....	.....	与 ECSS - E - ST - 50 - 13C 一致,但与本文无关
31	矢量字消息	不带数据字的同步(广播)	用于服务请求和标识小帧结构

矢量字消息用于标识 RT 方的消息传输服务请求,消息中数据字 16 bit 中的某 1 位置 1 表示有某种消息需要传输,BCS 将据此安排相应消息传输。该数据字定义如下:

- 1) D0 ~ D14: RT 可个性化定义对应所需消息。
- 2) D15: 固定用于请求发送 EDS 消息。

### 2.2 RT EDS 的设计

RT 的 EDS 内容分为 2 层:1553B 总线通信层 EDS(简称 RT EDS)和应用 EDS。

RT EDS 目的是描述 RT 方所需的传输消息、子地址定义以及矢量字服务请求关系<sup>[11]</sup>,见表 2。

表 2 1553B 通信层 EDS 格式<sup>[12]</sup>

Tab. 2 1553B communication EDS format<sup>[12]</sup>

矢量字定义			子地址消息定义				
矢量字标识	矢量字 mask	数据域	子地址标识	T子地址 mask	T子地址定义	R子地址 mask	R子地址定义
16	16	8 × 16	16	32	64 × 16	32	64 × 16

1) 矢量字定义通过矢量字 mask 和数据域建立 16 bit 请求位与子地址的对应关系。

2) T/R 子地址 mask 定义了哪些子地址被使用或未使用。

3) T/R 子地址定义规定了被使用的子地址的消息类型、数据字个数和消息间隔  $T_{ITV}$  要求。

4) 消息间隔  $T_{ITV}$  要求: 16 bit, 单位 ms。

- 全 1 表示 65.535 s, 全 0 表示没有周期要求, 属于根据条件、请求、需要而传输的消息。

- 划分时间 ( $T_{DIV}$ ) 的取值用于判定消息是直接安排在主总线表中 ( $T_{ITV} < T_{DIV}$ ) 还是采用插入方式 ( $T_{ITV} \geq T_{DIV}$ )。

应用 EDS 用于描述系统上层应用, 是 RT 对本设备的功能、性能、接口、工作方式的 EDS 描述, 传递给 BCA 以供构建系统上层的服务关系。这部分是遵从可扩展标识语言 (eXtensive Markup Language, XML) 语法语义规定<sup>[13-14]</sup> 描述的对象, 经过翻译后的解释数据。

EDS 传输格式采用 CCSDS 空间包格式<sup>[15]</sup>。包中的应用过程标识符 (Application Process Identifier, APID) 指明了内容对应到 BC 或 RT 端内部的不同层次的应用过程, 内容的区分通过包副导头定义。

### 2.3 即插即用的建立过程协议

RT 接入步骤如下:

**Step 1:** 以 30 个 RT 矢量字消息组成查询帧, 周期轮询 RT 的服务请求, 直到发现有 RT 接入 (消息不超时)。

**Step 2:** BCS 对查询帧只进行矢量字中 EDS 请求位的判读。

**Step 3:** 当 1 个 RT 接入后, 首先向 BC 提出发送 EDS 的服务请求。

**Step 4:** BCS 响应 RT EDS 服务请求, 以插入消息的方式将 RT EDS 安排传输并接收。

**Step 5:** BC 根据 RT EDS 构建总线表。

**Step 6:** 总线表建立并启动运行后, 调整插入查询帧运行的时间间隔 ( $T_{LOP}$ ), 并按照  $T_{LOP}$  插入查询帧查询剩余 RT。

**Step 7:** 重复 Step 3 ~ 6 步, 实现 RT 逐个加入的过程。

**Step 8:** 当所有 (30 个) RT 加入总线表运行后, 不再插入查询帧。

**Step 9:** 每当有 1 个新的 RT 接入时, BC 还需要解决新 RT 通信数据的收发控制, 通过 BCA 和 BCS 之间的标准数据结构和统一服务机制来解决。

**Step 10:** 在某个 RT 总线通信建立后, 如果

RT 有应用 EDS 需要传递, 可继续通过 EDS 请求向 BC 传输。

RT 撤出步骤如下:

**Step 1:** 当某个 RT 连续 1 min 出现超时响应, 且通过切换总线不能恢复通信 (再加 1 min), 则可判定此 RT 下线。

**Step 2:** 判定 RT 下线后, BC 对总线表进行重构, 删除下线 RT 的所有消息, 在查询帧中增加对下线 RT 的矢量字消息, 并调整  $T_{LOP}$ 。

**Step 3:** 该 RT 退出引发的应用功能上的影响和反应由 BCA 进行处理和系统重构。

## 3 即插即用 1553B 总线通信设计

### 3.1 标准数据结构设计

总线管理设计需要的标准数据结构是为了在 BCA、BCS、ACE 之间建立起统一的控制服务和数据传递服务。

标准数据结构按照位置分为 ACE 内存和 CPU 内存两部分。ACE 内存部分主要包括总线表 (BUS LIST) 和相关的配置寄存器<sup>[3]</sup>, 是 ACE 定义好的数据结构, 只能按照芯片使用手册的要求进行配置和设计。

CPU 部分又分为两部分:

一部分处于 BCS 与 ACE 之间, 主要包括消息表 (MSGTAB)、消息号表 (MSGNUMTAB)、消息块表 (MSGBLKTAB)。

MSGTAB 通过消息号对主总线表和插入消息表中所有消息建立索引, 目的是把消息内容的分析过程提前到初始化时进行, 在运行过程中只对该表进行查询, 就能获得诸如消息类型、数据区地址等信息, 避免对总线运行产生干扰。消息类型还可以起到与应用建立联系的作用。

MSGNUMTAB 是针对主总线表中的每个小帧建立的消息号索引, 目的是供 BCS 查询该表, 以确定某个小帧中是否有需要的消息存在, 实现主总线表中消息的上线和下线操作。

MSGBLKTAB 建立了消息块的地址索引, 目的是供切换冗余总线使用, 而不干扰总线运行。

另一部分处于 BCS 与 BCA 之间, 主要包括插入消息请求表 (INSMMSGFLAG)、数据区及其控制标志。为适应即插即用的随机性, 数据区及其标志可采用请求队列加缓存的统一方式组织。

### 3.2 BC 总线管理设计

总线管理设计分为总线初始化、总线控制和总线应用。

总线初始化包括总线表构建和对 ACE 总线表、寄存器的设置,以及根据总线表分析生成上述数据结构。在即插即用背景下,随着 RT 的增减,总线表会动态构建,每次都要重复上述过程。

总线运行控制方式包括对主总线表和插入消息表操作两种方式。主总线表的运行控制采用消息结束(End Of Message, EOM)处理(包括中断和查询),安排消息上线和消息下线。插入消息表运行控制采用 EOF 处理,并进行总线表切换。

总线应用是指 BCA 与 BCS 之间在取数和置数以及控制方面的交互问题。

BC 管理的基础功能模块共有 13 个,见表 3。

表 3 1553B 基础功能模块

Tab. 3 1553B basic function module

序号	名称	使用者	作用
1	CONSTRUCTBUS	BCS	根据 RT EDS 进行总线表的构建
2	INITBC	BCS	根据 CONSTRUCTBUS 构建的总线表在运行前进行分析,形成 MSGTAB、MSGBLKTAB 和 MSGNUMTAB,完成对 ACE 的寄存器的设置
3	ARRANGEMSG	BCS	根据 MSGNUMTAB,安排主总线中的消息上线
4	SKIPMSG	BCS	将主总线上的消息撤下总线,与 ARRANGEMSG 配套使用
5	WRITEBC	BCS	将发送数据从发送缓冲区置入 ACE 消息块中,即置数过程
6	READBC	BCS	从 ACE 中读取接收数据,放入接收数据缓冲区,即取数过程
7	SETINSFLAG	BCA, BCS	向 BCS 提出插入消息请求,包括需要插入消息的起点和消息个数
8	INSMMSG	BCS	根据插入消息请求信息,进行总线表停止、切换和重新启动的操作
9	RUNBC	BCS	启动总线表运行
10	STOPBC	BCS	停止总线表运行
11	SWITCHBUS	BCS	根据 MSGBLKTAB 切换总线
12	READDATA	BCA	从接收数据缓冲区中读取数据
13	WRITEDATA	BCA	向发送数据缓冲区中放入数据

### 3.3 总线表的构建算法

#### 3.3.1 总线表构建的前提约定

总线表的构建算法就是根据 RT EDS 生成新的总线表。即插即用的总线通信使用的主总线表可以只有固定的 30 个 RT 矢量字消息加 1 条同步消息,而所有其他有效消息均采用插入消息的方式。这种方式能保证所有 RT 的所有消息得到传输,但消息的周期性难以保证。因此有必要采用主总线表加插入消息的方式来平衡这一问题,而主总线表的构建算法就成为主要内容。为保证构建的总线表时间特性,需要对各种消息  $T_{MSG}$  进行约定,约定的原则如下:

1) 1553B 总线码速率为 1 Mbit/s, 1 个数据字在总线传输时为 20 bit (3 bit 同步 + 16 bit 数据 + 1 bit 校验), 1 bit 传输需 1  $\mu$ s, 加上状态字响应时间 4 ~ 12  $\mu$ s<sup>[16]</sup>, 因此最大 32 字的消息可以计算为: 指令字 20  $\mu$ s + 数据字 (32  $\times$  20  $\mu$ s) + 响应时间 (4 ~ 12  $\mu$ s) + 状态字 20  $\mu$ s + 间隙 (> 4  $\mu$ s)。为方便使用, 32 字消息的最小  $T_{MSG}$  取值 800  $\mu$ s。

2) 同样的消息在主总线表中中和插入消息表中的  $T_{MSG}$  是不同的, 主要是因为主总线表中的消息采用 EOM 方式处理, 需要为 BCS 的响应和处理留出时间余量, 而插入消息表采用 EOF 方式处理, 处理时总线表处于停止状态, 因此消息  $T_{MSG}$  可以不留余量。

根据这 2 条原则, 对各种消息  $T_{MSG}$  作如下约定:

1) RT 矢量字消息在主总线表中每条 3 ms, 在查询帧中 100  $\mu$ s。

2) 同步消息 100  $\mu$ s。

3) 主总线表中的  $T_{MSG}$ : 带 1 个数据字的消息为 1380  $\mu$ s, 每增加 1 个数据字, 时间增加 20  $\mu$ s, 32 个数据字的消息为 2 ms。

4) 插入消息表中的  $T_{MSG}$ : 带 1 个数据字的消息为 180  $\mu$ s, 每增加 1 个数据字, 时间增加 20  $\mu$ s, 32 个数据字的消息为 800  $\mu$ s。

#### 3.3.2 初始主总线表的建立

收到第 1 个 RT EDS 后, 主总线表的构建步骤如下:

**Step 1:** 找出 RT EDS 中期望的非 0、小于  $T_{DIV}$  的最大  $T_{ITV}$ , 作为初始  $T_{BUS}$ 。

**Step 2:** 主总线表先安排 1 条 RT 矢量字消息 + 1 条同步消息。

**Step 3:** 将 RT EDS 中的消息按顺序排列在主总线表 RT 矢量字和同步消息之间。

**Step 4:** RT EDS 中  $T_{ITV}$  为 0 或超长 ( $\geq T_{DIV}$ ) 的消息按照插入消息安排,不编入主总线表中。

**Step 5:** 全部安排完毕后,累加主总线表所有消息时间  $T_{MSG}$  作为  $T_{FRM}$ ,若  $T_{FRM} \geq$  初始  $T_{BUS}$ ,则设置完成,并以  $T_{FRM}$  为  $T_{BUS}$ ;若  $T_{FRM} <$  初始  $T_{BUS}$ ,则将同步消息前的最后 1 条消息的  $T_{MSG}$  加长,使  $T_{FRM} =$  初始  $T_{BUS}$ 。

**Step 6:** 将生成的总线表作为 1 个小帧,复制为 8 个。

### 3.3.3 第 $n$ 个 RT 加入主总线表

收到第  $n$  个 RT 的 RT EDS 后操作步骤:

**Step 1:** 记录当前  $T_{FRM}$ ,将 8 个小帧中同步消息前最后 1 条消息的  $T_{MSG}$  恢复缺省值。

**Step 2:** 在当前 8 个小帧中加入该 RT 矢量字消息, $T_{FRM}$  加上 3 ms。

**Step 3:** 根据该 RT EDS 将其 RT 消息加入 8 个小帧。加入消息时需要考虑  $T_{ITV}$  和位置问题。

1) 加入的消息  $T_{ITV}$  是当前  $T_{FRM}$  的  $n$  倍。

- $n < 2$ : 将其加入 8 个小帧,每个小帧 1 个。

- $2 \leq n < 3$ : 将其加入 4 个小帧,从  $T_{FRM}$  最短的小帧开始,每间隔 1 个小帧 1 个。

- $3 \leq n < 5$ : 将其加入 2 个小帧,从  $T_{FRM}$  最短的小帧开始,每间隔 3 个小帧 1 个。

- $5 \leq n \leq 8$ : 将其加入 1 个  $T_{FRM}$  最短的小帧。

- $n > 8$ , 但  $T_{ITV} < T_{DIV}$ : 暂时将其加入插入消息表中。

- 消息  $T_{ITV} \geq T_{DIV}$ : 将其作为插入消息,不编入主总线表。

2) 加入时的位置按照同类消息放在一起,同类消息按 RT 加入顺序排列。如果是新类型消息,则加在所有已有消息之后、同步消息之前。

3) 1 条消息加入完毕后,将该消息的  $T_{MSG}$  加入  $T_{FRM}$ ,作为新的  $T_{FRM}$ 。

**Step 4:** 继续加入该 RT 的下一条消息,重复 Step 3,直到 RT EDS 要求的所有消息加入完毕。

**Step 5:** 该 RT 的所有消息加入完毕后,检查在 Step 3 中暂时作为插入消息的  $T_{ITV}$  与刚生成的  $T_{FRM}$  的倍数关系  $n$ ,如果  $n \leq 8$ ,则将该消息从插入消息表中删除,按照 Step 3 将其加入主总线表小帧中。

**Step 6:** 求出所有小帧的新  $T_{FRM}$ ,取其中最大的  $T_{FRM}$ 。若 RT EDS 中的消息最大  $T_{ITV} >$  最大  $T_{FRM}$ ,加长该小帧中同步消息前的最后 1 条消息的  $T_{MSG}$ ,使  $T_{FRM} =$  该 RT 消息的最大  $T_{ITV}$ 。

**Step 7:** 若该 RT EDS 中的消息最大  $T_{ITV} \leq$  最大  $T_{FRM}$ ,将其余小帧的  $T_{FRM}$  均调整为与最大  $T_{FRM}$

相同(即所有  $T_{FRM}$  相等),方法同上。

**Step 8:** 若新  $T_{FRM} <$  调整前的  $T_{FRM}$ ,则将每个小帧的  $T_{FRM}$  继续加长到调整前  $T_{FRM}$ 。

### 3.3.4 有 RT 撤出

在运行过程中,有 RT 撤出时操作步骤:

**Step 1:** 将撤出 RT 的消息从总线表中删除。

**Step 2:** 删除后,重新计算各  $T_{FRM}$ ,取其大者作为  $T_{BUS}$ 。

**Step 3:** 在剩余所有的 RT EDS 中找到消息最大  $T_{ITV}$ ,若消息最大  $T_{ITV} \leq$  新  $T_{BUS}$ ,则调整完成,否则所有  $T_{FRM}$  均延长(方法同上),作为新的  $T_{BUS}$ 。

**Step 4:** 查询帧增加 RT 矢量字消息。

**Step 5:** 调整  $T_{LOP}$ ,并按照  $T_{LOP}$  插入查询帧。

**Step 6:** RT 的撤出次序与 RT 号的大小无关。

## 3.4 总线表构建适应性评估

总线表构建效果评估包括时间效果和适用场景两方面。

### 3.4.1 时间效果的评估

查询握手协议过程的时间主要是指  $T_{FRM}$  对  $T_{LOP}$  的影响,以及查询帧的消息时间。影响这一关系的主要有 2 个值,即  $T_{DIV}$  和  $T_{LOP}$ 。 $T_{DIV}$  的取值由主总线表中 8 个小帧周期之和最大值确定。 $T_{LOP}$  需要考察主总线表和插入消息之间的时间关系。为分析确定这 2 个取值,作如下约定:

1) 主总线表最大 512 条消息;

2) 主总线表平均分为 8 个小帧,每个小帧中最多 64 条消息;

3) 单消息按照 32 字来计算。

以小帧为单位,根据上述前提进行时间效果分析,计算公式如下:

$$T_{BUS} = T_{FRM} = \text{矢量字消息数} \times 3 + \text{消息数} \times 2 + 0.1$$

在稳定运行期间,没有插入消息,小帧满 64 条消息(矢量字消息数最多 30 个,消息数为 33 个)时, $T_{FRM}$  最大,达到  $30 \times 3 + 33 \times 2 + 0.1 = 156.1$  ms,8 个小帧时间为  $156.1 \times 8 = 1248.8$  ms。因此  $T_{DIV}$  取值 1.25 s 比较合适。

由于  $T_{FRM}$  的计算公式是线性的,因此其最大值随着 RT 个数的增加而线性增加。实际运行情况比这复杂,但都包络在最大值范围内,因此  $T_{FRM}$  效果是可以接受的。

$T_{LOP}$  与插入查询帧有关,还需要考虑插入消息时间的影响。此时至少有 1 条是插入矢量字消息,因此时间最长的插入消息时间是其余 511 条消息的总和。按每条消息 800  $\mu$ s 计算,511 条消息时间总和为 408.8 ms。在主总线表和插入消

息全部运行的情况下,  $T_{FRM}$  最大为  $29 \times 3 + 34 \times 2 + 0.1 = 155.1$  ms, 最大总线周期  $T_{BUS} = 155.1 + 408.8 = 563.9$  ms。在主总线表只有 1 个 RT 矢量字且没有查询到消息请求时,  $T_{BUS}$  最小为 3.1 ms。因此插入消息的运行方式加上 BCS 干预和处理的时间开销会导致  $T_{BUS}$  产生波动。

综上, 在具体程序设计时, 有两种方法确定  $T_{LOP}$  取值: 一是  $T_{LOP} = T_{FRM} \times n$ 。  $n$  太小会因为主总线频繁插入查询帧和插入消息对  $T_{BUS}$  的稳定不利, 太大又会影响对 RT 接入的时间延迟效果。二是为简化设计而取固定值。两种方法可以结合使用, 即根据不同条件(如总线流量的平均统计情况对  $T_{FRM}$  的影响)在每次总线表重构时动态调整  $n$  值, 而以 5 s 作为最大值保底。

BC 总线表重构过程是在 CPU 内存中进行的, 不影响 ACE 中总线表的运行。由于 RT 个数不同, 消息需求不同, 重构过程所需要的时间是不确定的, 与 BC 方 CPU 处理能力有关。

总线表重构后加载到 ACE 中的过程时间一般为毫秒级。

由此可知, 即插即用的 RT 识别周期和加载时间在条件确定时不会有大的差别, 而总线表重构时间是影响 RT 识别时间长度的主要因素, 在 CPU 处理能力确定的情况下, 主要看重构算法的优化。

### 3.4.2 适用场景评估

主总线表重构面临的适用场景有两种:

1) BC 第一个加电运行, 然后从无到有, 所有 RT 一一接入的过程。这种情况与地面测试、在轨飞行过程中的设备正常加电的过程一致, 也是本协议的出发点。

2) 有 RT 先加电运行, 当 BC 加电运行时, 一次可以查询到多个 RT 同时接入。这种情况与系统复位(掉电)后, 所有设备又同时上电工作的情况一致。

第 2 种情况下, BC 处理与第 1 种情况在本质上相同, 只是在构造完 1 个 RT 的总线表后, 不是直接运行, 而是看是否还有第 2 个 RT 接入, 即多 RT 接入过程一次性完成。为此总线表构建过程将重复多次, 看似冗余, 但可以适应任意 RT 数量和顺序的接入。

## 4 即插即用 1553B 的系统层次架构

SOIS 架构中, 通过设备发现服务 (Device Discovery Service, DDS<sup>[17]</sup>)、设备枚举服务 (Device Enumeration Service, DES<sup>[18]</sup>)、设备虚拟服务 (Device Virtualization Service, DVS<sup>[19]</sup>) 和设备访问服务

(Device Access Service, DAS<sup>[20]</sup>) 可以构建上层系统的设备即插即用, 本文设计的 1553B 总线管理基础功能模块对它们的支持和衔接关系见图 4<sup>[21]</sup>。

图 4 中上 3 层属于 SOIS 的相关业务架构。在收到 RT EDS 后, 从 CONSTRUCTBUS 开始进行总线表构建, 向下初始化总线 INITBC, 向上将 RT EDS 内容(设备虚拟访问信息和应用 EDS)报告给 DDS。DDS 依之建立上层逻辑地址与下层物理地址之间的映射关系, 将应用 EDS 上传给 DES, 纳入系统的管理信息数据库 (Management Information Base, MIB) 中, 提供给应用层使用。

在系统运行过程中, DVS 根据应用层的使用要求和虚拟设备地址, 与 DES 提供的物理地址进行匹配, 然后交由 DAS 执行对设备的访问。

DAS 访问设备时可以使用存储访问服务 (Memory Access Service, MAS<sup>[22]</sup>) 和包服务 (Package Service, PS<sup>[23]</sup>) 两种方式。这两种服务都需要 DDS 提供下层 1553B 总线的具体访问方式(地址和数据), 使用取数 (READDATA 和 READBC) 和置数 (WRITEDATA 和 WRITEBC) 服务实现与 BCS 之间的数据传递。

DAS 通过 SETINSFLAG 和 ARRANGEMSG 向 BCS 提出消息请求。BCS 使用 ARRANGEMSG 服务和 SKIPMSG 服务安排主总线表消息上线和下线, 使用插入消息 INSMMSG 服务切换总线表, 使用 STOPBC 和 RUNBC 服务来停止和启动总线的运行。

图 4 的系统架构只有 1553B 总线一个子网。如果还有其他子网, 则系统需增加汇聚层协议, 而图 4 中 BCS 的数据发送和接收缓冲区, READDATA、WRITEDATA 和 SETINSFLAG 服务都将划归汇聚层协议。

## 5 结论

本文的 1553B 总线即插即用解决方案试图从由传统的接入设备通过标准化接口适应系统的设计思路, 转变为由系统对异构和未知设备进行自动识别和配置, 并构建形成新的能力的思路。因此本方案在兼容已有的 1553B 总线通信协议(主要是 ECSS - E - ST - 50 - 13C)基础上, 只增加了初始握手部分的协议内容, 后续的正常通信过程和内容由通信双方通过 EDS 交流的方式, 建立起适应性的通信关系。从效果评估看, 时间特性可接受, 能够适应实际使用场景, 而且即插即用对 RT 方编程的改变仅在于要求其能够一次性编辑形成自身的 EDS, 并按照握手协议将 EDS 传递给 BC, 其余可保持原有设计不变。

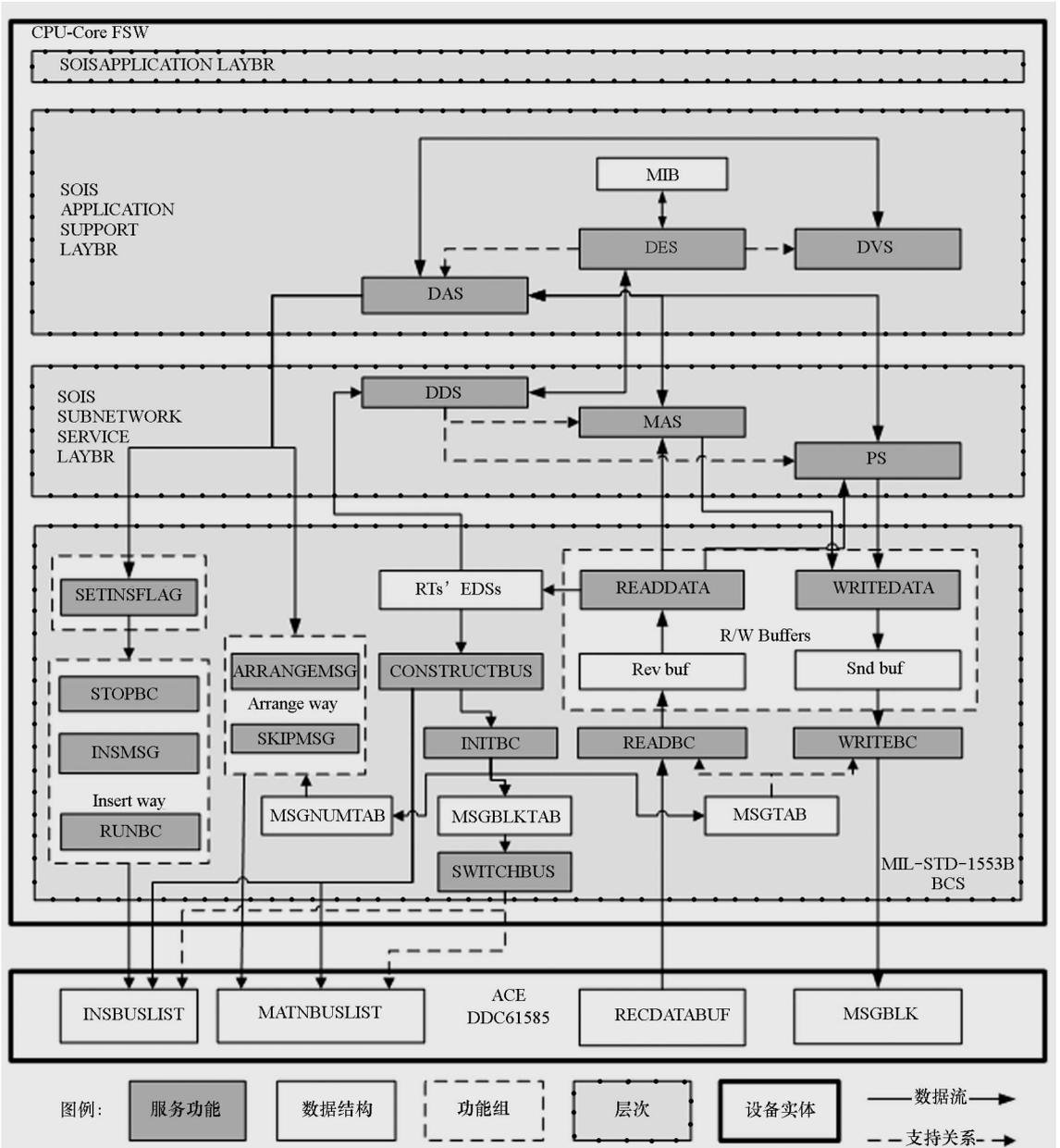


图 4 1553B 总线 BC 与 SOIS 的服务衔接层次关系<sup>[21]</sup>

Fig. 4 Services of 1553B bus controller in SOIS architecture<sup>[21]</sup>

### 参考文献 (References)

[1] Drabbe J. Interface and communication protocol for MIL-STD-1553B data bus onboard spacecraft: ECSS-E-ST-50-13C [S]. European Cooperation for Space Standardization, 2008.

[2] 数字式时分制指令/响应型多路传输数据总线: GJB 289A-97[S]. 北京: 国防科学技术工业委员会, 1997. Digital time division command/response multiplex databus: GJB 289A-97 [S]. Beijing: State Administration of Science, Technology and Industry for National Defence PRC, 1997. (in Chinese)

[3] 周莉, 安军社, 谢义方, 等. 星载高速无线数据网络协议的设计[J]. 国防科技大学学报, 2014, 36(5): 1-7. ZHOU Li, AN Junshe, XIE Yifang, et al. Design of the network protocol of a wireless spacecraft high speed data network [J]. Journal of National University of Defense

Technology, 2014, 36(5): 1-7. (in Chinese)

[4] 方青文, 安军社, 周莉, 等. 面向航天应用的 SoC 设计[J]. 微电子学与计算机, 2013, 30(5): 10-13. FANG Qingwen, AN Junshe, ZHOU Li, et al. The design of SoC oriented space application [J]. Microelectronics & Computer, 2013, 30(5): 10-13. (in Chinese)

[5] 方青文. 航天专用数据采集和状态控制 ASIC 设计与验证[D]. 北京: 中国科学院大学, 2013. FANG Qingwen. Design and verification of data acquisition and state control ASIC oriented space application [D]. Beijing: University of Chinese Academy of Sciences, 2013. (in Chinese)

[6] 谭维炽, 顾莹琦. 空间数据系统[M]. 北京: 中国科技出版社, 2008: 6-12. TAN Weizhi, GU Yingqi. Space data system [M]. Beijing: China Science and Technology Press, 2008: 6-12. (in Chinese)

[7] 张庆君, 郭坚, 董光亮. 空间数据系统[M]. 2 版. 北京:

- 中国科技出版社, 2016: 7.
- ZHANG Qingjun, GUO Jian, DONG Guangliang. Space data system[M]. 2nd ed. Beijing: China Science and Technology Press, 2016: 7. (in Chinese)
- [8] 何熊文, 朱剑冰, 程博文. 星载标准接口业务在航天器中的应用方法[J]. 航天器工程, 2015, 24(6): 52-58.  
HE Xiongwen, ZHU Jianbing, CHENG Bowen. Application method of spacecraft onboard interface service in spacecraft[J]. Spacecraft Engineering, 2015, 24(6): 52-58. (in Chinese)
- [9] Spacecraft onboard interface services: CCSDS 850.0-G-2[S]. USA: Consultative Committee for Space Data Systems, 2013.
- [10] MIL-STD-1553A/B notice 2 RT and BC/RT/MT, advanced communication engine (ACE): BU-65170/61580; BU-61585 [R]. USA: Data Device Corporation, 1992.
- [11] 吕帅. 1553B 总线远置终端即插即用研究与设计[D]. 北京: 中国科学院大学, 2018.  
LYU Shuai. Research and design of 1553B bus remote terminal plug-and-play [D]. Beijing: University of Chinese Academy of Sciences, 2018. (in Chinese)
- [12] 吕帅, 吕良庆. 一种 1553B 总线远程终端的即插即用的设计[J]. 计算机测量与控制, 2018, 26(6): 184-188.  
LYU Shuai, LYU Liangqing. One plug and play design of 1553B bus remote terminal [J]. Computer Measurement & Control, 2018, 26(6): 184-188. (in Chinese)
- [13] Spacecraft onboard interface services—XML specification for electronic data sheets; CCSDS. 876.0-R-2 [S]. USA: Consultative Committee for Space Data Systems, 2016.
- [14] Spacecraft onboard interface services—specification for dictionary of terms for electronic data sheets; CCSDS. 876.1-R-2 [S]. USA: Consultative Committee for Space Data Systems, 2016.
- [15] Space packet protocol; CCSDS. 133.0-B-1 [S]. USA: Consultative Committee for Space Data Systems, 2003.
- [16] 罗志强. 航空电子综合化系统[M]. 北京: 北京航空航天大学出版社, 1990: 32-44.  
LUO Zhiqiang. Integrated avionics system [M]. Beijing: Beihang University Press, 1990: 32-44. (in Chinese)
- [17] Spacecraft onboard interface services—subnetwork device discovery service; CCSDS. 854.0-M-1 [S]. USA: Consultative Committee for Space Data Systems, 2009.
- [18] Spacecraft onboard interface services—device enumeration service; CCSDS. 871.3-M-1-S [S]. USA: Consultative Committee for Space Data Systems, 2014.
- [19] Spacecraft onboard interface services—device virtualization service; CCSDS. 871.2-M-1-S [S]. USA: Consultative Committee for Space Data Systems, 2014.
- [20] Spacecraft onboard interface services—device access service; CCSDS. 871.0-M-1-S [S]. USA: Consultative Committee for Space Data Systems, 2013.
- [21] 吕良庆. 航天器智能软件体系架构设计与应用研究[D]. 北京: 中国科学院大学, 2018.  
LYU Liangqing. Design and application study of intelligent flight software architecture on spacecraft [D]. Beijing: University of Chinese Academy of Sciences, 2018. (in Chinese)
- [22] Spacecraft onboard interface services—subnetwork memory access service; CCSDS. 852.0-M-1 [S]. USA: Consultative Committee for Space Data Systems, 2009.
- [23] Spacecraft onboard interface services—subnetwork packet service; CCSDS. 851.0-M-1 [S]. USA: Consultative Committee for Space Data Systems, 2009.