

ARINC659总线多余度容错系统同步技术

李育*, 安刚, 李欣

中航工业第一飞机设计研究院, 陕西 西安 710089

摘要:介绍了 ARINC659 总线结构、窗口活动、同步方式, 分析了初始同步和重同步过程中时钟漂移和异步度产生的原因。以 ARINC659 背板总线为核心, 设计了满足高安全性和高可靠性需求的四余度综合化、模块化的飞机机载计算机系统, 分析了系统架构、总线桥接方式、工作原理。针对余度管理的同步核心功能, 设计了系统应用级的任务同步算法和总线通信级的消息同步算法。该同步技术已经应用于型号研制中。

关键词: ARINC659 总线; 计算机同步; 余度管理; 容错计算机

中图分类号: TP366 文献标识码: A 文章编号: 1007-5453 (2016) 12-0028-06

为了提高机载系统的可靠性, 在元器件基本可靠性指标一定的前提下, 核心部件如传感器、控制器、作动器、计算机等通常采用冗余技术, 即在一个系统中同时存在多个单元执行相同的任务, 优势在于可以有效隔离故障单元, 并将任务切换到正常单元继续执行。但是, 这样的设计在保证可靠性的同时带来了多余度容错系统的管理问题: 系统将面临如何确保各个计算单元之间的协调工作。

多余度容错系统的表决、故障隔离、降级等控制过程中, 同步是余度管理的基础和关键技术, 直接影响到容错系统的功能及运行的正确性。它保证了各冗余模块间保持步调一致地工作, 在某一时刻同时完成同一任务的某个基本动作。这样, 容错系统的表决过程才有意义, 才能保证系统中冗余模块的多数一致的原则, 屏蔽系统中的少数故障, 准确地切换到正常模块, 实现结构上的重构, 使容错系统能够持续、正确、可靠地运行。

ARINC659 总线源于霍尼韦尔公司的 SAFEBus 总线。1993 年, SAFEBus 总线被美国航空电子工程师协会(AEEC) 采纳并成为标准, 颁布为 ARINC659 背板总线规范。ARINC659 背板总线是基于时间触发架构的双-双余度配置的容错串行总线, 支持鲁棒的时间分区和空间分区, 是综合化、模块化航空电子系统的关键技术^[1]。该总线技术已经应用于波音 B777 的飞机信息管理系统 (Airplane

Information Management System, AIMS)。国内也有机型应用了该总线, 作为高安全性和高可靠性多余度容错计算机内部功能模块通信的板级背板总线和计算机间的通信总线, 本文对其多余度容错系统同步技术进行研究。

1 多余度容错计算机系统

1.1 总线结构

每个总线接口单元 (Bus Interface Unit, BIU) 均连接至 2 条总线。每个现场可替换模块 (LRM) 拥有 2 个总线接口单元 BIU_x 和 BIU_y, 分别通过 x 和 y 总线进行传输, 每条总线由 LRM 中独立的收发器进行驱动, 以防止单个故障对一条以上总线产生不利影响^[2]。

BIU 之间进行交叉校验, 4 条总线之间也进行交叉校验, 使得 LRM 拥有双自检能力。串行传输线的使用减少了硬件数量, 简化了全并发监控, 提高了总线基本可靠性。4 条总线之间的交叉校验提高了数据可靠性和可用性。总线体系结构、接口及传输线构成与连接关系如图 1 所示。

1.2 总线活动

计算机运行期间, 首先要保证 ARINC659 总线上的各个底板之间的数据同步, 即实现 ARINC659 总线同步, 保证数据的正确传输。ARINC 659 总线活动由交替的消息和间隙组成的窗口构成, 如图 2 所示, 每一个窗口占据相关 LRM 表命令规定的固定时间段^[2]。

收稿日期: 2016-09-13; 退修日期: 2016-10-27; 录用日期: 2016-11-04

* 通讯作者. Tel.: 029-86832774 E-mail: kingliyu@sina.com

引用格式: LI Yu, AN Gang, LI Xin. ARINC659 bus redundancy fault-tolerant system sync technology [J]. Aeronautical Science & Technology, 2016, 27 (12): 28-33. 李育, 安刚, 李欣. ARINC659 总线多余度容错系统同步技术 [J]. 航空科学技术, 2016, 27 (12): 28-33.

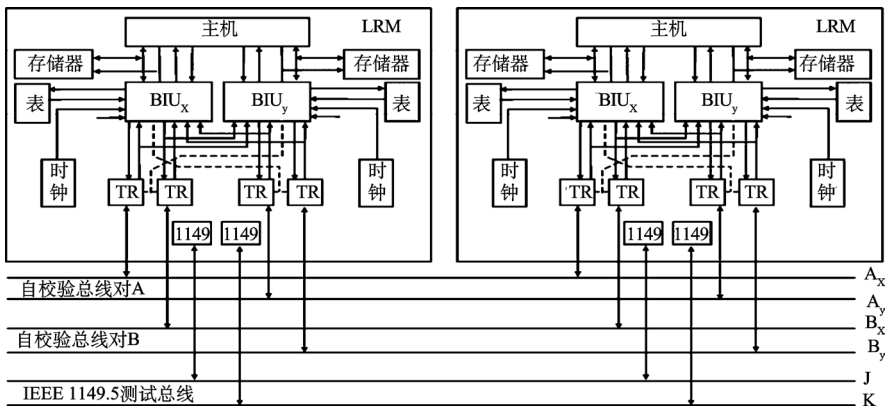


图 1 ARINC659 总线结构
Fig.1 ARINC659 bus architecture

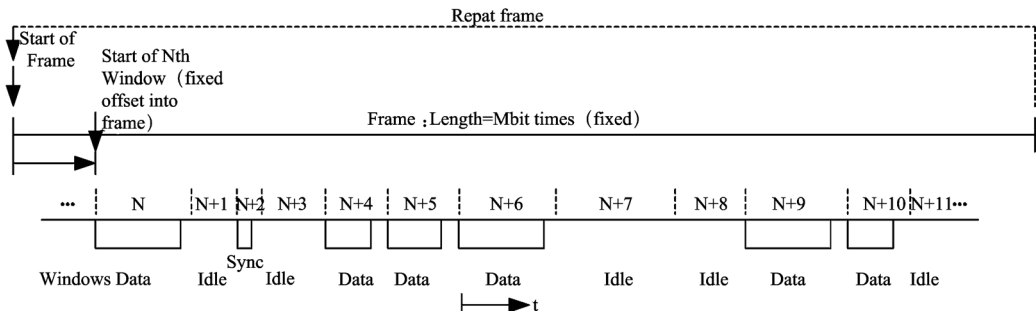


图 2 ARINC 659 总线活动
Fig.2 ARINC659 bus activity

窗口可以包含一个数据消息、同步信息或空闲。每一个窗口可以有一个唯一的发送器或是一组有限的后备发送器，它们被编程在表存储器中。每一个消息的源和目的地址包含在表存储器中而不是通过总线传送。系统设计 / 综合者可以组织多种类型的底板模块间的消息结构，支持模块—模块（点—点）通信、一个模块到一组模块（广播）通信，以及备选的一个模块到一组模块的通信。有两种类型的消息：基本消息和主 / 后备消息。基本消息用于单个源和单个目的或多个目的的情况。主 / 后备消息用于有多个备用源和单个目的或多个目的的情况，系统仲裁机制只允许主或后备源之一的发送器访问总线。一个后备源只有在主和其他优先级比它高的后备发送器，在预先确定的时间周期保持沉默时才能在总线上发送^[2]。

1.3 四余度计算机系统

ARINC659 总线具有的最大优势在于传输过程的时间确定性、双自检能力、故障检测和隔离能力，非常适合设计综合化和模块化的多余度容错计算机系统。以目前飞机设计中最高余度等级的四余度计算机系统为设计目标，介绍基于 ARINC659 总线技术的冗余计算机系统结构，如图 3 所示。

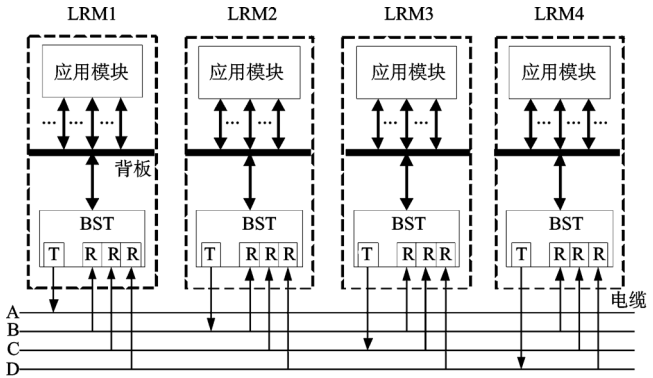


图 3 四余度计算机系统
Fig.3 Four-redundancy computer system

每个 LRM 为一个独立控制通道。通道以 ARINC659 背板总线为核心，内部由应用模块、ARINC659 背板和总线桥接传输（BST）模块组成。各功能模块均为智能节点，含有处理器或微控制器，对功能板上的信号进行调理和监控，以资源的方式挂接总线。每一个功能模块都能够进行信息交换、资源共享和资源重构，最终达到四余度系统能够以功能模块为最小单元进行资源动态重构和系统容错。

BST 板负责各余度计算机间的 ARINC659 总线数据的

交换与共享,将背板模式转换为电缆模式转发到其他 LRM,是总线信号的延续和背板信号在计算机级的转换和传输,实现冗余通道间的交叉通信数据链路 (CCDL)。

ARINC659 总线规范了物理层和数据链路层,其数据通信同步主要工作于数据链路层。在计算机启动后,由 BST 负责对 CCDL 进行初始化同步,实现各通道间的数据同步。然后,各个 BST 发起对通道内 ARINC659 总线的初始化同步请求,实现机内功能模块间的数据同步。任务周期内,由 BST 负责维持各通道之间总线同步和机内同步。

2 同步方法

2.1 同步方式

对于系统 c ,若理想时钟的时间为 T ,并设该节点的时钟时间为 $c_c(T)$ 。对于另外 2 个系统 p 、 q ,若满足 $|c_p(T) - c_q(T)| \leq \delta_{\max}$ (其中 $\delta_{\max}$ 为最大时钟异步度),则称系统 p 和 q 同步。根据同步周期的不同,同步技术通常可分为:时钟级同步、松散同步及任务级同步。

时钟级同步,是指分布式系统采用相同的时钟源驱动各节点,使各节点在每个时钟周期进行重同步,所以系统中各节点按照相同时序同步运行,如图 4 所示。时钟级同步主要基于硬件方式实现,优点是同步严格,无额外的时间开销,但是对时钟的要求很高,往往需要采用容错时钟^[3]。

松散同步,采用的是一种基于时间片的同步方式,如图 5 所示。每个时间片由多个时钟周期组成,在每个时间片内,它允许系统中的各个节点存在异步度,只要异步度在允许的范围内,则可认为系统中的各个节点同步,在下一个时间片内进行重同步。该种方式通常采用具有逻辑时钟的同步软件实现。松散同步的主要目的是周期性地将各个节点的时钟对齐,这样可减小或者消除节点之间的异步度,减少

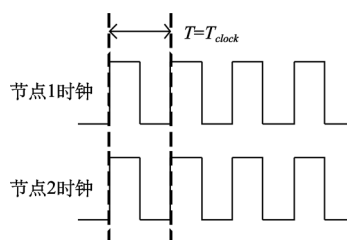


图 4 时钟级同步
Fig.4 Clock sync

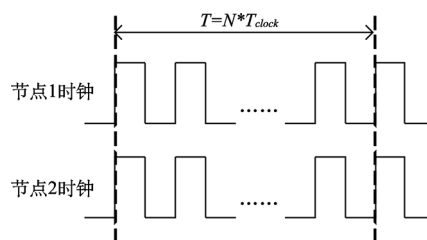


图 5 松散同步
Fig.5 Loose sync

时钟的误差累积,从而为各节点完成相同的任务建立相同的条件,抑制模块之间由于异步而造成的故障蔓延。松散同步允许同步有一定的偏差,能消除某些瞬时故障的影响,但时间的开销较大。

任务同步,属于松散同步的一种,时间的基本单位为任务周期,每个任务周期包括任务执行时间和任务空闲时间。多个任务周期组成重同步周期,各个节点在每个任务周期重复执行任务,以其中某个节点的时钟或者所有节点综合出来的时钟作为基准时钟,若有节点的时钟异步度大于最大异步度,则以基准时钟为标准对所有节点的逻辑时钟进行校正,使系统中的各个节点重新回到同步状态,如图 6 所示。优点是各个节点的时钟可以独立运行,且实现简单易行。

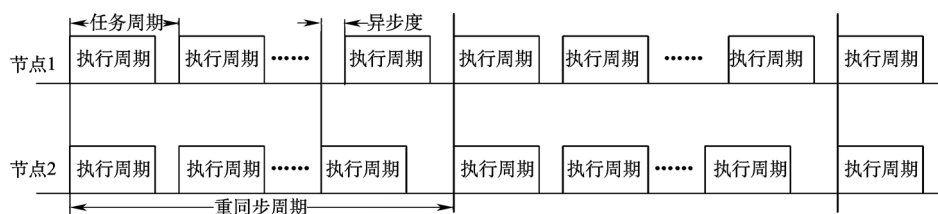


图 6 任务同步
Fig.6 Task sync

2.2 应用级任务同步设计

应用级的任务同步指冗余计算机根据所服务系统的具体应用需求进行的任务级同步过程,包括初始化同步和重同步。

初始化同步如图 7 所示:LRM1, LRM2, LRM3, LRM4 开始任务周期的时刻分别为 t_a, t_b, t_c, t_d , 其中 $t_a > t_b > t_c > t_d$, 几个时刻均不重叠。 T_s 表示初始同步所需要的时间, T_c 表示周期任务所需要的时间。在第一个周期内,LRM1 未收到其他

3 个 LRM 的时间信息,按照预定的延时时间 T_d 运行,其他 LRM 虽能够收到 A 的时间信息,但时间偏移量的计算还没完成,故同样照预定的延时时间 T_d 运行。在第二个周期内,LRM1 的时间作为整个网络的基准时间,则其仍按延时时间 T_d 运行,所以 $T_{da}=T_d$,而其他 3 个 LRM 按照计算出的延时时间运行,分别为 T_{db} , T_{dc} , T_{dd} 。

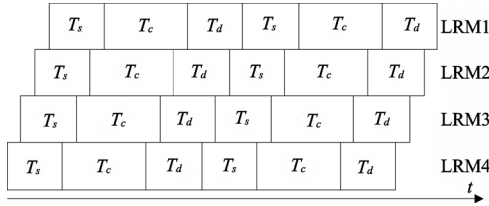


图7 任务周期起始情况
Fig.7 Task period start

若 LRM1, LRM2 在第 $M+1$ 个周期时实现同步,则公式

(1) 成立:

$$\begin{aligned} T_s + T_d + T_c + M * (T_s + T_c + T_{da}) + (t_a - t_b) \\ = T_s + T_d + T_c + M * (T_s + T_c + T_{db}) \end{aligned} \quad (1)$$

同样,若 LRM1, LRM3 在第 $N+1$ 个周期时实现同步,则公式 (2) 成立:

$$\begin{aligned} T_s + T_d + T_c + N * (T_s + T_c + T_{da}) + (t_a - t_c) \\ = T_s + T_d + T_c + N * (T_s + T_c + T_{dc}) \end{aligned} \quad (2)$$

同样,若 LRM1, LRM4 在第 $P+1$ 个周期时实现同步,则公式 (3) 成立:

$$\begin{aligned} T_s + T_d + T_c + P * (T_s + T_c + T_{da}) + (t_a - t_d) \\ = T_s + T_d + T_c + P * (T_s + T_c + T_{dd}) \end{aligned} \quad (3)$$

因此,LRM2, LRM3, LRM4 分别经过 $M = (t_a - t_b) / (T_{dc} - T_d)$, $N = (t_a - t_c) / (T_{dc} - T_d)$, $P = (t_a - t_d) / (T_{dd} - T_d)$ 个任务周期后,均与 LRM1 同步,4 个 LRM 的延时时间也均为 T_d ,最终实现初始化同步。

重同步:初始化同步完成后,由于各个 LRM 硬件时钟源时钟漂移问题的存在,任务周期也存在细微的差异,随着时间的推移,时钟漂移差异不断累积,表现出较大的异步度。若异步度超过所允许的最大范围,则认为 4 个 LRM 不在同步状态,需要进行重同步操作。如图 8 所示,一段时间后,LRM2, LRM3, LRM4 与 LRM1 相比,分别出现了 δ_{ab} , δ_{ac} , δ_{ad} 的异步度,其中 $|\delta_{ad}|$ 最大,随着时间的推移, δ_{ad} 必然先达到所允许的最大异步度 δ_{max} ,此时,需要减小中央处理单元 D 的延时,按照新计算出来的延时 T_{dd} 运行,使得 A 和 D 实现重同步,B, C 的重同步过程也和上述过程

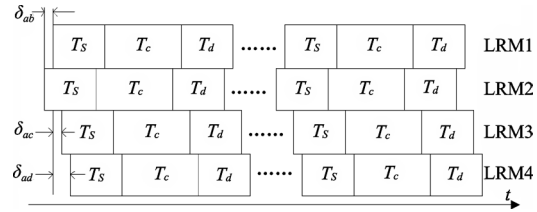


图8 任务周期重同步
Fig.8 Task period resynchronization

类似。

图 3 所示冗余度容错计算机系统的 4 台 LRM,分别用 W, X, Y, Z 表示。系统动态运行过程中,每一台 LRM 自己的身份是 W,其余 LRM 的身份是 X, Y, Z。这样 4 台 LRM 与 W, X, Y, Z 之间的对应关系如下:

LRM1: W-LRM1, X-LRM2, Y-LRM3, Z-LRM4;
LRM2: W-LRM2, X-LRM4, Y-LRM3, Z-LRM1;
LRM3: W-LRM3, X-LRM1, Y-LRM2, Z-LRM4;
LRM4: W-LRM4, X-LRM1, Y-LRM2, Z-LRM3。

在以上对应关系下,同步流程如图 9 所示。

2.3 总线通信级同步设计

表存储器可以存储长度不同的帧命令序列,帧之间的切换由帧切换命令实现。有两种不同类型的帧:版本帧和非版本帧。在一个版本帧中,总线上活动的所有 BIU 应该具有相同的表版本号。版本帧切换机制保证了所有同步的 BIU 根据接收到的帧切换消息中的版本数据校验自己的版本号,如果表版本号不一致,它就同总线失去同步。

在一个非版本帧中,表的版本被忽略,只要一个 LRM 中的 BIU 能够与总线同步,它们就能够参与非版本帧。另外,非版本帧中的所有消息应由最大时间间隙(9 位时间)分割,对于主/从消息必须用最大步进时间间隙(10 位时间)。

当 BIU 处于非同步状态,它将试图与活动的 BIU 恢复帧同步。初始化同步限定等待时间是指 BIU 在决定发送初始化同步脉冲之前,搜索同步消息(短同步消息、长同步消息或者初始化同步消息)所等待的时间长度。如果已经过了初始化同步限定等待时间,BIU 仍未看到一个同步脉冲,BIU 就会传输一个初始化同步脉冲。如果 BIU 收到了一个初始化同步消息(由其他模块或自己发送的),它将立即执行一个固有命令去接收一个非版本进入同步消息,消息中同步码为“0”,全分辨率时间值为“0”。帧级同步过程如图 10 所示。

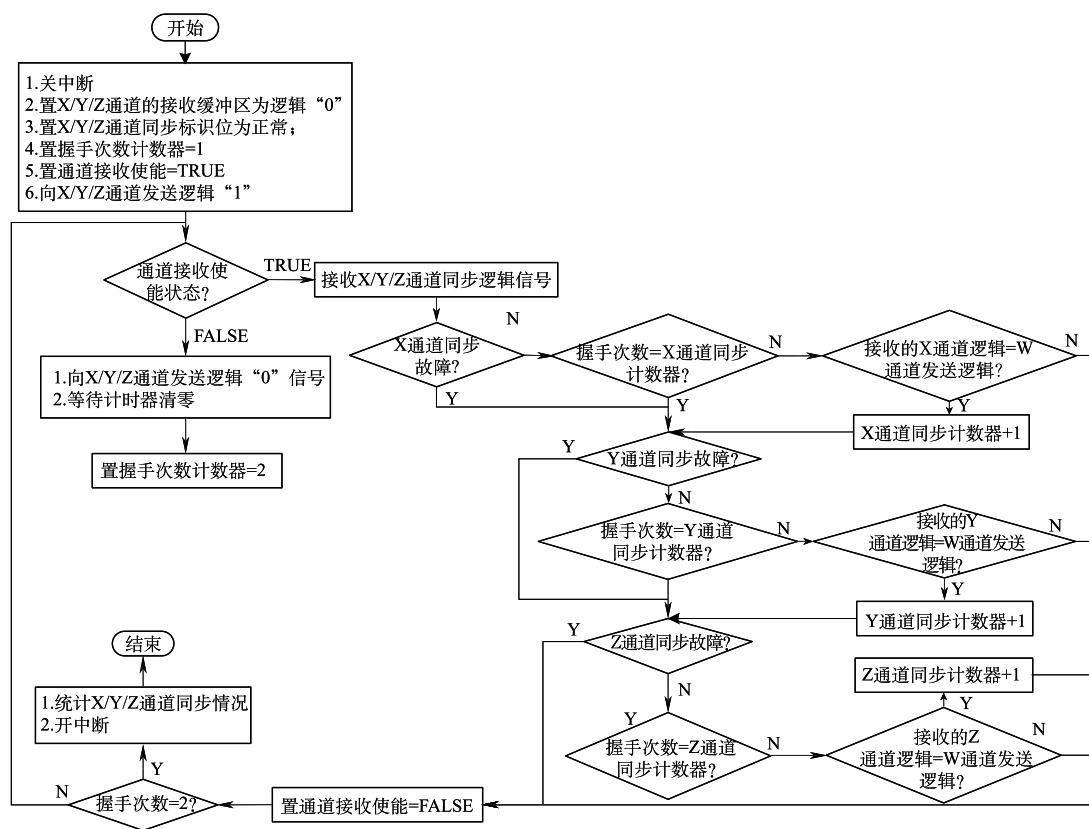


图9 同步流程
Fig.9 Sync process

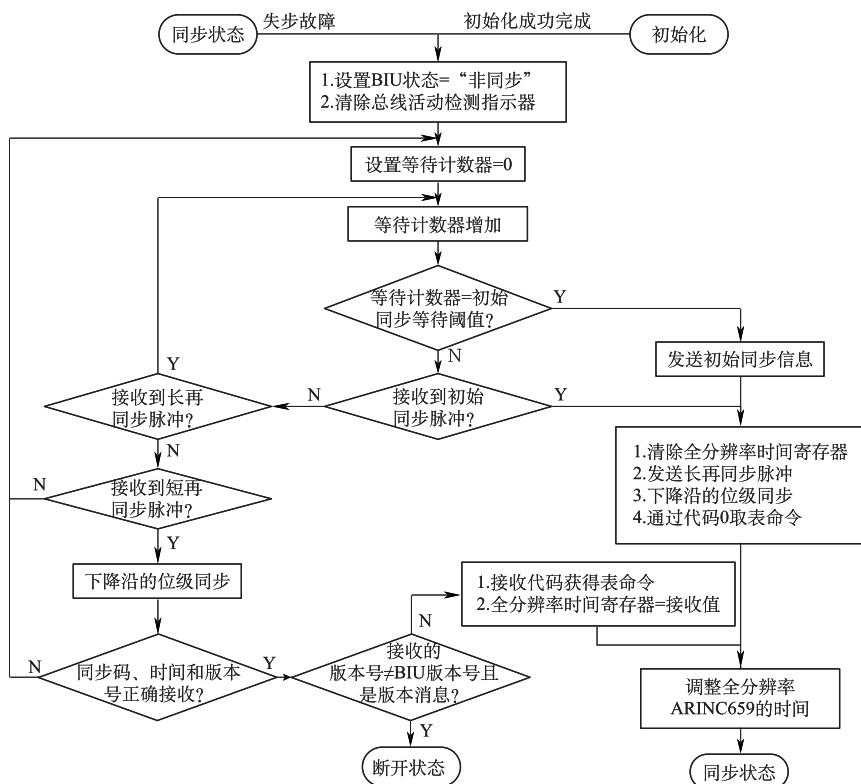


图10 帧级同步过程的流程图
Fig.10 Flow chart of frame level sync process

3 结束语

ARINC659总线具有数据吞吐量大、编码效率高、传输时间确定、故障隔离严格、容错性好等优点,适用于高安全性、高可靠性和高综合度要求的飞机机载计算机、控制器等的通信系统,既能设计成为计算机内背板总线,也能满足计算机间通信需求。以ARINC659总线为核心设计四余度容错计算机系统,同步是挂接在ARINC659背板总线上的计算机内功能模块和计算机间数据传输、余度管理的基础。通过对总线特点、四余度计算机结构特点、同步传输方式、时钟漂移、异步度产生原因的分析,设计应用任务和总线通信两级同步算法,确保了数据从应用层到数据链路层的同步传输。

AST

参考文献

[1] 霍曼. 综合航空电子技术发展展望 [J]. 航空电子技术, 2003

(3): 12-17.

HUO Man. Prospects for the development of integrated avionics technology[J]. Avionics technology, 2003 (3): 12-17. (in chinese)

[2] Airlines Electronic Engineering Committee. Arinc Specification 659 Backplane Data Bus [Z]. Aeronautical Radio Inc, 1993.

[3] 支超有. 机载数据总线技术及其应用 [M]. 北京: 国防工业出版社, 2009.

ZHI Chaoyou. Airborne data bus technology and its application [M]. Beijing: National Defense Industry Press, 2009. (in chinese)

作者简介

李育(1978—)男,硕士,高级工程师。主要研究方向:系统集成和飞行仿真。

Tel: 029-86832774

E-mail: kingliyu@sina.com

ARINC659 Bus Redundancy Fault-tolerant System Sync Technology

LI Yu*, AN Gang, LI Xin

AVIC The First Aircraft Institute, Xi'an 710089, China

Abstract: This paper introduced ARINC659 bus structure, the window activity and sync mode. And the cause of clock drift and asynchronous degree in the process of initial sync and resynchronization were analyzed. With ARINC659 backplane bus as the core, designed to meet the requirements of high security and high reliability of four-redundant integrated, modular aircraft airborne computer system, and analyzed the system architecture, bus bridge mode, working principle. In view of the synchronization core function of redundancy management, the system application level task sync algorithm and bus communication level message sync algorithm were designed. The sync technology has been applied in the development of aircraft design project.

Key Words: ARINC659 bus; computer sync; redundancy management; fault-tolerant computer

Received: 2016-09-13; Revised: 2016-10-27; Accepted: 2016-11-04

*Corresponding author. Tel.: 029-86832774 E-mail: kingliyu@sina.com