

# ARINC 659 命令表配置工具设计与实现

曹朋朋, 田 泽, 许宏杰, 刘宁宁

(中国航空计算技术研究所, 陕西 西安 710068)

**摘 要:** ARINC 659 总线是主要用于机架内部各个在线可更换模块之间的通信, 在综合模块化航空电子系统中有着广泛的应用。采用表驱动访问的 ARINC 659 总线系统开发中需要开发总线操作调度表, 一种基于 ARINC 659 规范的命令表自动生成与编译工具可解决 ARINC 659 产品使用中的命令表设计的问题, 加速产品的应用于推广。文中介绍了 ARINC 659 总线的工作的基本原理, 设计实现了基于 ARINC 659 总线命令表配置工具, 完成了命令表的生成、编译与仿真, 是一种比较好的 ARINC 659 总线命令表的解决方案。

**关键词:** ARINC 659; 配置工具; 背板总线

**中图分类号:** TP39

**文献标识码:** A

**文章编号:** 1673-629X(2012)07-0235-03

## Design and Implementation of ARINC 659 Config Tools

CAO Peng-peng, TIAN Ze, XU Hong-jie, LIU Ning-ning

(Aeronautical Computing Technique Research Institute, Xi'an 710068, China)

**Abstract:** ARINC 659 bus is mostly used in the communication of the LRM (Line Replaceable Module) in cabinets. It is widely used in the IMA (Integrated Modular Avionics). The ARINC 659 bus is table driven protocol accessing and the system based on ARINC 659 needs to plan the bus operation scheduler table. A ARINC 659 tool that can automatically create and compile the command table is strongly recommended; And it will overcome the troubles of the ARINC 659 command table and accelerate the application and popularization of 659 products. Firstly introduce the work theories of ARINC 659, and then design and implement a ARINC 659 config tool. This tool can automatically create, compile and emulate the ARINC 659 command tables. The practical use of the tool indicates that it is a preferably resolvent of the ARINC 659 command tables.

**Key words:** ARINC 659; config tools; backplane bus

## 0 引言

ARINC 659 总线是一种是由 HoneyWell 公司提出的在综合模块化航空电子 (IMA) 机架中, 在线可更换模块 (LRM) 之间的进行数字数据传送的标准<sup>[1]</sup>。它定义了用于机架内通信的通用背板数据总线的接口要求特性及通过背板连接器连接的 LRM 与背板的电气特性和期望行为, 具有较高吞吐量、严格的故障隔离、数据传输确定等特性, 是一个在时间 (总线传输时间) 上和空间 (存储空间) 上具有高容错性和鲁棒性的高完整性底板总线<sup>[2]</sup>。ARINC 659 采用串行总线的方式减少了硬件, 简化了全并发监控, 从而提高了可靠性。每个 LRM 中 BIU 之间的交叉比较使得所有四条总线都具有双自检能力。四条通路之间进一步的交叉检验

提高了数据的可用性。

近年来, ARINC 659 广泛应用于航空电子综合化系统中, 在波音 777 飞机的飞机信息管理系统 (AIMS) 中、Honeywell 公司的 Primus Epic 系统等商用航空领域已得到应用<sup>[3]</sup>。ARINC 659 采用表驱动比例访问 (TDPA) 的通信机制, 总线操作按照预定的时间命令表 (总线操作调度表) 进行, 以往 ARINC 659 系统的开发通常由系统设计中需根据 ARINC 659 总线协议规范, 手动利用帧描述语言编写命令表, 并需要开发一种编译工具, 对编写的命令表进行编译, 编写命令表的过程中需对各窗口的时间以及帧的活动序列做出详细的策划<sup>[4]</sup>, 这必然增加了系统开发的难度和周期。文中基于 ARINC 659 总线产品的推广和使用需要, 开发 ARINC 659 协议处理芯片的支持软件, 实现命令表的配置、生成与编译, 满足了基于 ARINC 659 系统的应用需求。

## 1 ARINC 659 配置工具工作原理

根据 ARINC 659 总线规范定义, 完整命令表的所

收稿日期: 2012-03-20; 修回日期: 2012-06-22

基金项目: 中国航空工业集团公司创新基金 (2010BD63111); 总装“十二五”预研项目 (51308010601)

作者简介: 曹朋朋 (1984-), 男, 陕西兴平人, 硕士, 研究方向为集成电路设计; 田 泽, 博士, 研究员, 中国航空工业集团首席技术专家, 研究方向为 SoC 设计、嵌入式系统设计、VLSI 设计。

有数据存放在 ARINC 659 芯片外部的 PROM 中,当系统上电后,ARINC 659 芯片的命令表自加载逻辑自动将外部 PROM 中的命令表搬到内部 SRAM 中<sup>[5]</sup>。命令表自加载模块自动产生地址从外部 FLASH 中读出数据,并写入内部 SRAM 中,不需要主机干预。命令表加载完成后,ARINC 659 芯片将开始初始化、预译码命令,并且按命令表执行命令<sup>[6]</sup>。

ARINC 659 命令表是通过帧语言写的程序。帧描述语言定义了 LRM 间总线通信的一个 ARINC 659 总线集。它不但描述了数据的运动,而且给出了数据传输时序<sup>[7]</sup>。定义了 ARINC 659 总线数据帧中的每个窗口操作,不仅定义了每一个窗口的长度,还定义了安排在给该窗口的时间内,哪一个 LRM 发送、接收和忽略总线。ARINC 659 总线的通信按此语言对应生成的命令表进行。ARINC 659 产品的使用需开发相应编译工具,对生成的命令表进行编译,生成符合 Intel MCS-86 Hex Object 文件格式的命令表文件。

ARINC 659 命令表配置工具基于应用需求,实现了一款便于系统调试与维护的 ARINC 659 支持软件,实现根据系统应用要求,对系统内各节点间的通信和节点各任务进行可视化配置和设置、生成总线命令表,并对命令表进行图形化仿真,为调试人员提供图形化显示命令表各窗口信息,从而迅速准确地建立系统综合的框架。

## 2 ARINC 659 配置工具系统功能设计

在 ARINC 659 命令表配置工具的设计上,主要完成 ARINC 659 总线活动的配置,利用可视化界面完成命令表的配置,窗口界面的配置输入需要提取命令表的机架、模块等关键参数,并对用户配置输入进行合法性检查;自动生成的命令表格式符合 ARINC 659 标准协议和相关技术标准;完成命令表项目的保存、打开与关闭,命令表文件的打开、编辑、保存功能,方便用户查看与分析命令表;命令表的仿真利用图形界面完成命令表帧活动窗口的时序显示,并支持图形窗口的放大,并提供友善的坐标显示,按时间关系用不同的颜色显示帧中不同类型的窗口。

根据其功能,可将 ARINC 659 配置工具分为三大模块,分别为命令表生成模块、命令表编译模块及命令表仿真模块。

## 3 ARINC 659 命令表配置工具实现

### 3.1 命令表生成模块

命令表生成模块主要实现命令表项目管理功能负责维护整个项目的命令表工程的编辑管理,模块定义及帧的管理,利用变成语言提供的窗体管理命令表项

目及帧具体信息<sup>[8]</sup>。本程序采用 VC++ 6.0 建立窗体程序,用户通过可视化界面维护整个命令表工程的相关信息,并根据所配置的机架信息,生成 .txt 格式的命令表。该部分为配置工具的基本部分,提供友善的用户界面,并在用户所做的每一步配置,根据 ARINC 659 规范做出合法性检查,对不符合的操作应给出提示。根据用户配置的信息,自动生成指定格式的命令表。命令表格指令格式是指令在源程序中的书写格式,指令以行为单位,基本格式为:

标号 指令助记符 模块编码 操作参数。

标号——某条指令所存单元的符号地址。

指令助记符——指明该语句将执行的操作。

模块编码——与该操作有关的 LRM 模块的编码。

操作参数——为进一步说明命令操作的参数。

单行注释——由分号开始,用来对指令和程序加以说明,便于阅读。注释可以单独一行,也可以放在指令结束之后<sup>[9]</sup>。

跨行注释——使用类 C 语言风格,以“/\*”开始,以“\*/”结束。

### 3.2 命令表编译模块

配置工具的编译功能,将 ARINC 659 总线命令表汇编语言翻译成目标代码,并提供对总线命令表的语法检查功能。

命令表编译模块对利用本工具生成的命令表或用户已有命令表进行编译,生成仿真 .mif 文件、布局布线文件 .code 及 Intel MCS-86 Hex Object .mcs 文件。整个编译过程主要包括标号扫描、语法检查、编译仿真 .mif 文件,之后根据 .mif 文件生成布局布线文件 .code 以及 Intel MCS-86 Hex Object .mcs 文件<sup>[10]</sup>。

编出输出的文件如下:

1).txt 源文件,自动生成的命令表文件或已存在的命令表文件。

2)编译输出文件:.src 源文件整理后的文件(转换为大写,删除注释和“;”,标点前后加空格)。

3).lst 编译过程记录文件。

4).coe 编译生成布局布线文件。

5).mif 编译生成仿真文件。

6)mcs Intel MCS-86 Hex Object 文件。

编译流程如图 1 所示。

命令表编译过程中,通过预处理及语法检查,判断命令表中是否存在错误,根据 ARINC 659 总线协议规范,确保命令表没有错误,在此基础上进行编译过程,首先生成二进制文件,将命令表错做命令转换为 mif 文件,命令操作码编码如表 1 所示。

编译生成仿真文件 mif 文件后,需生成是 PROM 和 Flash 都可识别的文件,此处,主要支持 PROM,需生

成 Intel MCS-86 Hex Object,该文件每行以“:”开头后跟若干个 16 进制位,格式如图 2 所示。

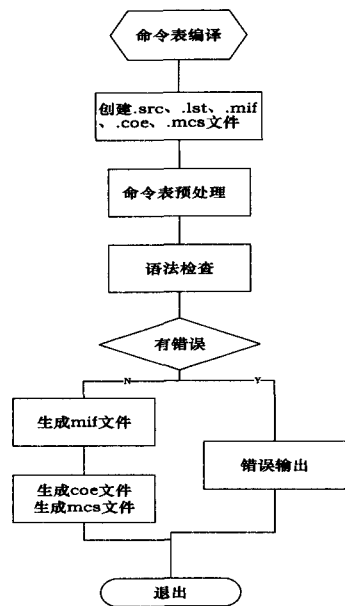


图 1 命令表编译流程

表 1 OPCODE 编码

指令编码	汇编助记符	长度	指令说明
0000	FREE	1	空闲
0001	BOW、TX、RX	4	数据命令,基本消息
0010	SSYNC	1	短同步
0011	ERU	2	转入同步
0100	FCU	2	帧切换
0101	JUMP	1	跳转
0110	CALL	1	调用
0111	RET	1	返回
1000	INT	2	中断
1001	BOW、TX、RX	4	数据命令,主后备消息
1010	-	-	-
1011	ERV	2	带版本校验的转入同步
1100	FCV	2	带版本校验的帧切换
1101	JUMPI	1	带固有空闲的跳转
1110	CALLI	1	带固有空闲的调用
1111	RETI	1	带固有空闲的返回

:	byte count (1 Byte)	Address (2Byte)	record type (1Byte)	Data (以 Byte 为单位)	Checksum (1 Byte)
---	---------------------------	--------------------	---------------------------	----------------------	----------------------

图 2 MSC 文件构成

Byte count 是以十六进制表示的,后面 data 域的 byte 数。

Address 即 Flash 的低 16 位地址,而 Flash 的完整的地址是 32 位的。

data 是数据域。

checksum 校验和,域所有的字节各自都按位取反,再加 1。

record type 表示后续数据块的类型。

每 4k 为一段加起始标记如“:020000040000FA”开头的 02 表示 data 域有两个字节,数据域的两个字节为段号,取值范围 0~3。

3.3 命令表仿真模块

总线命令表方针模块在命令表无语法错误的情况下,扫描命令表,分帧统计帧中窗口信息,根据窗口信息,分别计算出每帧数据的起始时间和窗口占用使用<sup>[11]</sup>,最终以波形的形式显示命令表帧中窗口的时序信息,以方便用户连接命令表活动的相关信息,将系统数据的传送流程形象的图形化显示出来。

4 结束语

文中结合 ARINC 59 芯片的使用和推广<sup>[12]</sup>,开发了命令表配置工具,利用 ARINC 659 背板总线配置工具软件,用户可以简单高效地助用户快速的根据系统应用需求创建命令表,使得用户脱离了 ARINC 659 命令表的 txt 文件的编写过程,在应用过程中,用户不需要对 ARINC 659 总线命令表的语法结构做深入理解便可快速构建 ARINC 659 总线活动的序列,加速了基于 ARINC 659 相关产品的的开发与应用,利用命令表仿真功能可视化模拟命令表运行过程中各窗口的数据传送关系,为系统用户提供了良好的人机开发界面,具有广泛的应用前景,方便了 ARINC 659 芯片的使用与推广,对具有同类需求的产品使用和推广具有借鉴意义<sup>[13]</sup>。

参考文献:

[1] Airlines Electronic Engineering Committee. ARINC 659 Specification 659 Backplane Data Bus[S]. Annapolis, Maryland, USA:Aeronautical Radio, Inc,1993.

[2] 郭 亮,李 玲,田 泽,等. ARINC659 总线接口芯片的 FPGA 原型验证[J]. 计算机技术与发展,2009,19(12): 240-247.

[3] 张喜民,魏 婷. ARINC659 背板数据总线应用研究[J]. 航空计算技术,2011,41(5):105-109.

[4] Zhang Wei. Stability analysis of networked control systems [D]. Ohio,US:Case Western Reserve University,2001.

[5] 强健新,田 泽,淮治华. 基于 ARINC 659 的 FPGA 原型验证平台的构建与实现[J]. 计算机工程与设计,2010,31(12):2726-2729.

[6] 魏 婷,张喜民. ARINC 659 背板总线调试方法研究[J]. 测控技术,2008,27(S0):84-86.

[7] 徐文辉. ARINC659 总线简介[J]. 航空电子技术,1999,30(2):22-27.

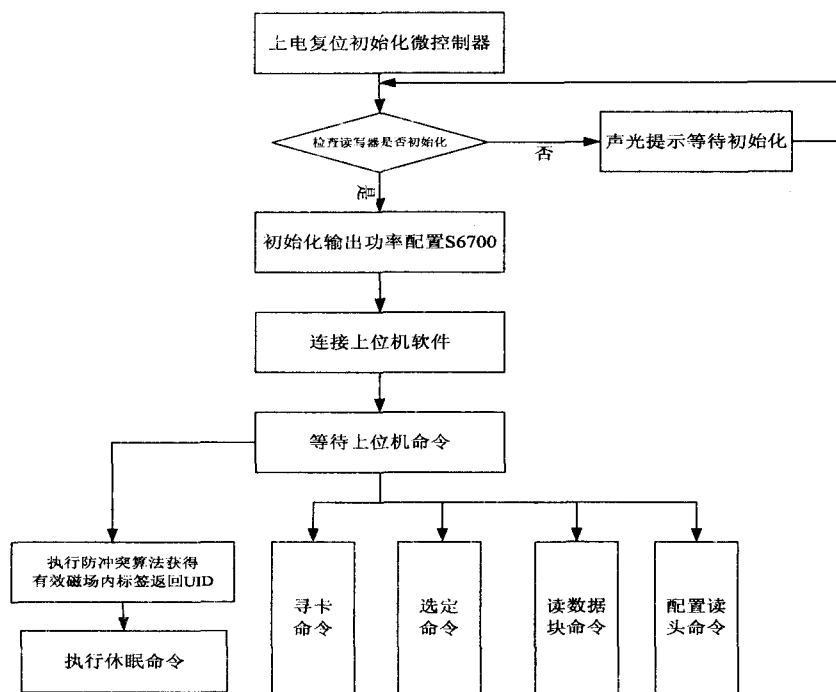


图5 读写器总的工作流程

亮;

(4) 打开测试软件进行测试。

· 测试内容:

(1) 指令模式测试。

(2) 自动寻卡模式测试。

指令模式下的测试目的是检验读写器功能的完整性,测试中使用的标签是 TI 的 Tagit,测试结果表明读写器能很好地执行自定义读写器命令及标签支持的 ISO/IEC15693 操作命令。

自动寻卡模式下的测试目的是检测读写器的性能指标,如读写的距离、抗干扰能力、读写的速度等。测试结果表明读写器在单天线的正常识别距离可以达到 70cm,以此推断双天线的工作距离可达到 1.3m 以上,达到了某些场合的商用指标。

## 5 结束语

RFID 技术是当今的研究热点,而且 RFID 技术的

应用也越来越广泛,应用范围不仅涉及到生产、管理等领域,而且涉及到了人们日常生活的方方面面。这样一个背景下,文中设计开发了一款基于 ISO/IEC 15693 标准的高频远距离 RFID 读写器,本读写器目前已经达到了应用推广阶段。

## 参考文献:

- [1] 梁家海,陈海. RFID 技术在仓储管理中的应用[J]. 微计算机信息, 2008,24(20):247-249.
- [2] Finkenzeller K. 射频识别(RFID)技术[M]. 第2版. 陈大才译. 北京:电子工业出版社,2001.
- [3] 刘东生,邹雪城,杨秋平. 高频 RFID 读写器射频模拟前端的实现[J]. 半导体技术,2006(9):669-672.
- [4] 严光文,张其善. 射频识别卡读写模块的设计[J]. 北京航空航天大学学报,2006,29(2):178-180.
- [5] 陈邦媛. 射频通信电路[M]. 北京:科学出版社,2002.
- [6] ISO/IEC identification cards-contactless integrated circuit(s) cards-proximity cards-pards-part3: initialization and anticollision[S]. 2001.
- [7] 祝建平,王广志,丁辉. 射频识别技术的竞时自动计时系统设计和实现[J]. 传感技术学报,2006,19(1):112-115.
- [8] 刘舒祺. 基于 TIS6700 系列芯片的 RFID 阅读器设计[J]. 世界电子元器件,2008(8):51-53.
- [9] ISO/IEC identification cards-contactless integrated circuit(s) cards-vicinity cards-part1: physical characteristics,15693-1[S]. 2000.
- [10] 鲁公羽,陈雄,倪斌. 射频识别系统中读写器开发与研究[J]. 计算机工程与应用,2006(7):89-91.
- [11] Lee J G, Hwang J S. Software architecture for a multi-protocol RFID reader on mobile devices [C]//Second International Conference on Embedded Software and System. [s. l.]: [s. n.], 2005:16-18.

(上接第 237 页)

- [8] 隋颖,于秀山,杨豹. GUI 软件测试文档辅助工具的设计与实现[J]. 计算机技术与发展,2009,19(12):146-149.
- [9] 刘慕涛,张磊,王艳,等. 基于 XML 的 API 自动化测试工具设计与实现[J]. 计算机工程,2007,33(13):96-98.
- [10] 黄晖,王泉. 航空软件配置管理系统设计和关键技术研究[J]. 航空计算技术,2010,40(4):69-71.
- [11] Rushby J. CSL Technical Report: A Comparison of Bus Archi-

tectures for Safety Critical Embedded Systems[R]. California: SRI International, Menlo Park, 2003.

- [12] 邵刚,田泽,韩炜. ARINC 659 总线收发器的设计与优化[J]. 计算机技术与发展,2011,21(8):39-45.
- [13] Black R, Fletcher M. Next Generation Space Avionics: A Highly Reliable Layered System Implementation[J]. IEEE AES Systems Magazine, 2005, 20(12):9-14.