

Delphi 平台下的 1553B 总线卡通信应用技术研究

张 荣, 吴付岗, 陈 练, 邓 婷, 周继昆, 王 蕾

(中国工程物理研究院总体工程研究所, 四川 绵阳 621900)

摘要: 介绍一种基于 Delphi 平台的 PCI-1553B 总线卡接口通信应用技术, 主要包括 Delphi 下 1553B 总线周期消息、非周期消息定义、消息传输故障重试、消息数据的中断采集、消息数据的动态更新以及总线消息的传输监控记录等编程技术。并基于 1553B 总线卡和 Delphi 平台设计了一套电控模拟器通信系统, 验证了 1553B 总线的 BC- $\rightarrow$ RT 通信和 RT- $\rightarrow$ BC 通信以及 BM 总线监控功能。实践表明, 利用 Delphi 开发 1553B 总线通信应用技术以及 Delphi 对数据库技术的完美支持, 与其他开发语言相比, 该技术可快速建立完善的 1553B 总线通信测试系统。

关键词: Delphi; 1553B 总线; 周期消息; 故障重试; 中断传输;

Application Technique Research on 1553B Bus Interface Card with DELPHI platform

Zhang Rong, Wu Fugang, Chen Lian, Deng Ting, Zhou Jikun, Wang Lei

(Institute of Systems Engineering, CAEP, Mianyang 621900, China)

Abstract: Interface communication technique for a PCI-1553b bus card based on Delphi platform is introduced. It includes the programming techniques of definition of periodic-message and non-synchronous message, message transmission fault-retry, message data interrupt sampling, message data updating and message transmission monitoring of bus. And with designing an electrical control simulator system based on 1553b bus card and Delphi platform, BC- $\rightarrow$ RT, RT- $\rightarrow$ BC and BM bus monitoring function of 1553b bus are verified. Application has showed that, with using Delphi developing 1553b bus technique and with its approving of database, comparing to other language, this technique can build communication test system based on 1553b bus quickly.

Keywords: Delphi; 1553B-bus; periodic message; fault-retry; interrupt-transmitting

0 引言

1553B 总线即“飞机内部分式制指令/响应多路传输数据总线^[1]”, 广泛应用于飞机航空平台、空间航天器平台内各分布式计算机间的实时通信。该总线通信速率为 1 Mbps 甚至更高^[2], 总线传输的误码率达到 10^{-7} 以上, 并具备 A/B 双余度总线的热备切换传输功能, 是一种高可靠性传输总线。常用 1553B 总线通信的消息主要有 BC- $\rightarrow$ RT、RT- $\rightarrow$ BC、RT- $\rightarrow$ RT 以及方式 mode 几类, 所有消息的传输均由 BC 控制器发起, 各 RT 终端只能被动响应^[3]。

本文采用美国 Conder 公司^[4]生产的 PCI-1553B 总线卡设计某电控模拟器通信系统, 该 PCI-1553B 总线卡可在同一台电脑上模拟出 BC、RT 以及总线监控器 BM, 其功能复杂, 提供基于 VC++ 平台的二次应用开发例程。由于通常 1553B 总线通信系统涉及 ICD^[5] 文件操作, ICD 文件负责解析 1553B 总

线消息中数据和命令的物理意义, 多采用数据库构建, 但 VC++ 对数据库的编程能力相对较弱, Delphi 平台的数据库编程能力非常强大, 因此, 研究 Delphi 平台下 1553B 总线通信具有重要意义。

Delphi 平台下实现 1553B 总线的通信应用编程, 主要包括: 周期/非周期消息定义, 消息传输故障重试方法, BC- $\rightarrow$ RT、RT- $\rightarrow$ BC 消息传输中断响应处理, BC- $\rightarrow$ RT、RT- $\rightarrow$ BC 消息的数据动态更新以及总线监控器 BM 对总线活动数据提取方法等。

该型 PCI-1553B 总线卡提供用于二次开发的动态链接库 busapi32.dll^[6], Delphi 平台下的编程基于该 dll 文件, 使用 busapi.pas 头文件引用 dll 中的函数或向其函数传递参数。Delphi 下 dll 的调用参考文献 [7], 此处不赘述。

1 消息定义

1.1 消息分类

1553B 总线消息分周期消息和非周期消息两类。周期消息按照固定周期在总线上传输, 非周期消息则是由外部异步事件驱动, 在周期消息的传输过程中动态插入的单次消息。多条消息按照一定时间间隔排列构成一个消息次帧, 多个消息次帧构

收稿日期: 2014-05-10; 修回日期: 2014-06-10。

基金项目: 中物院总体所创新与发展基金(11CXJ-06)。

作者简介: 张 荣(1979-), 男, 四川资阳人, 工学硕士, 高级工程师, 主要从事环境动态测试技术、计算机软硬件开发技术方向的研究。

成一个消息主帧。总线以消息主帧方式进行某次具体数据传输。1553B 总线消息类型主要分为 BC→RT、RT→BC、RT→RT 以及 mode 码消息等。

1.2 BC 类消息结构体定义

在 Delphi 中, PCI-1553B 卡的各类消息用结构体表示。BC 类消息结构体定义为: api_bc_mbuf, 结构体主要成员及其含义如表 1 所示。

表 1 BC 类消息结构体主要成员及含义表

结构体成员	数据类型	成员含义
messno	word	消息标识号
control	word	消息控制字
messnext	word	下一条传输的消息号
command1, command2	word, word	消息命令字 (RT-RT 消息有两个命令字)
status1, status2	word, word	消息状态字 (RT-RT 消息有两个状态字)
data	array[0..1,0..count] of word	消息携带的数据, 存储在 A/B 两缓冲区中
gaptime	word	消息间时间间隔

表中 comand 控制字由 RT 地址 rtaddr、RT 子地址 subaddr、消息收发 (t/r) 标志以及消息传输的数据个数 num 确定, 根据 1553B 总线消息格式定义, command=2 048 * rtaddr + (t/r) * 1 024 + subaddr * 32 + num; 控制字 control 有多个参数可选: 标准 BC 消息 bc_control_message、消息重试 bc_message_retry 以及消息发送完是否允许中断 bc_message_interrupt 等; 对于周期消息, 用 bc_control_mframe_beg 以及 bc_control_mframe_end 标志分别表示某帧消息中的第一条消息和最后一条消息; data 用二维数组表示, 数据可配置在 A/B 两个缓冲区内, 每个缓冲区最多配置 32 个字; gaptime: 采用 μs 单位定义的该条消息与下条消息之间的传输时间间隔。

在 Delphi 中一条 BC 消息配置代码如下:
var bc_mess;api_bc_mbuf; //声明 bc 消息变量;
bc_mess.messno:=messno; //定义消息号;
bc_mess.control:=bc_control_message; //消息控制字;
bc_mess.messnext:=next; //定义下条待执行消息号;
bc_mess.command1=2048 * rtaddr+(t/r) * 1024+num+subaddr * 32; // 根据 rt 地址、子地址、收发标志以及数据传输个数 num 确定命令字;

for i;0 to num-1 do
bc_mess.data[0,i]:=busdata; //该消息传输的数据字;
bc_mess.gaptime:=50; //消息间隔为 50us;

消息配置完成后, 调 busapi32.dll 中写消息 api 函数 bustools_bc_messagewrite (card, messno, @bc_mess) 将配置的消息写到 PCI-1553B 板卡存储器中。

1.3 周期消息配置

配置周期消息时需确定消息帧的执行周期, 周期 timer 以 μs 为单位, 在 BC 初始化时利用 api 函数 bustools_bc_Init (card, timer) 设定到 pci-1553b 卡硬件中。但是周期消息配置时必须在 BC 消息结构体控制字中设置帧消息中的头消息标志和尾消息标志, 头/尾消息标志设置代码及含义如下:

```
//周期帧中第一条消息设置  
bc_mess.messno:=messno; //定义帧头消息号  
bc_mess.control:=bc+mess.control  
or bc_control_mframe_beg; //头消息标志  
bc_mess.messnext:=messno; //周期帧中下一条消息  
//周期帧中最后一条消息设置  
bc_mess.messno:=messno; //定义帧尾消息号  
bc_mess.control:=bc_mess.control  
or bc_control_mframe_end; //尾消息标志  
bc_mess.messnext:=messno; //消息号设置为头消息
```

1.4 非周期消息配置

非周期消息是传输起始时间不定的单次消息, 非周期消息与周期消息的配置区别在于 BC 消息结构体中的 bc_mess.messno 应与 bc_mess.messnext 值相同即 bc_mess.messno:=bc_mess.messnext。

2 消息传输故障重试

1553B 总线消息传输故障时允许在双余总线上进行传输重试, 重试通常使用条件如表 2 所示。

表 2 1553B 总线传输消息故障重试条件表

重试条件标识	重试条件
BC_RETRY_ALTB	总线交换重试
BC_RETRY_NRSP	总线传输无响应
BC_RETRY_ME	消息发送错误
BC_RETRY_BUSY	总线传输忙

该型 PCI-1553B 卡允许选择消息重试执行的总线以及配置重试执行的次数, 利用数组 retry 设置重试时执行的总线, 利用参数 wretry 在板卡初始化时确定重试的条件标识。如当总线交换传输消息或总线消息传输无响应时, 设置最多七次故障重试的代码及其含义如下:

```
var retry:array[0..7] of word; //定义重试执行方法数组  
wretry:word; //定义消息传输重试标识  
retry[0]:=RETRY_ALTERNATE_BUS; //第一次重试交换总线;  
retry[1]:=RETRY_SAME_BUS; //第二次在当前总线上重试;  
retry[2]:=RETRY_ALTERNATE_BUS; //第三次重试再次交换总线;  
retry[3]:=RETRY_SAME_BUS; //第四次重试在原总线进行;  
retry[4]:=RETRY_ALTERNATE_BUS  
retry[5]:=RETRY_SAME_BUS;  
retry[6]:=RETRY_ALTERNATE_BUS;  
retry[7]:=RETRY_END; //结束重试  
status:=BusTools+BC_RetryInit(Card,@retry); //设置重试参数到硬件板卡  
wretry:=BC_RETRY_NRSP or BC+RETRY_ALTB; //定义重试条件标识  
status:=BusTools_BC_Init(Card_1553,0,0,wRetry,...); //初始化 BC
```

利用 api 函数 bustools_bc_retryinit () 以及 bustools_bc_init () 将消息重试的条件标识和重试的总线选择参数写入到板卡硬件中, 完成重试相关参数的设置。

3 消息数据的中断采集

1553B 总线通信的目的是采集消息中携带的数据, 目前采

集数据的方式主要有定时查询数据缓冲区法与中断处理法两类,但中断处理法比定时查询数据缓冲区法效率高,且每次获得的总线数据均是更新后的数据,是主要采用的取数方法。

3.1 BC 接收 RT 上传数据的中断采集

BC 接收 RT 上传数据的中断采集方法:在消息控制字中设定消息结束产生中断标志,总线发送完消息后即产生中断,BC 将总线消息传输的过程记录在中断队列 sIntFIFO 中,定义中断处理函数 BC_Int,在中断处理函数中,从中断队列头元素到尾元素轮流查找其中所需采集数据的消息定义即可,如查找 rtaddr、subaddr 等。但在执行中断处理函数前需先注册中断函数,再启动消息传输。涉及的 api 函数有:中断注册函数 bustools_registerfunction、中断处理函数 Bc_Int 及中断数据读取函数 busTools_bc_messageRead;涉及的中断变量类型有:api_int_fifo、api_bc_mbuf;例如 BC 向地址为 3 的 RT 的子地址 3 请求数据的代码及含义如下:

```
function BC_Int(cardnum:integer;sIntFIFO:pointer):integer;stdcall;
Var sIntFIFO_BC:api_int_fifo;tail:integer;//中断队列及队列指针定义
msg_buffer_read;API_BC_MBUF;//BC 消息数据存储缓冲区定义
begin
SintFIFO_BC:=sIntFIFO;
tail:=sIntFIFO_BC.tail_index;
while(tail<>sIntFIFO_BC.head_index)do//中断队列内有元素
begin
if((sIntFIFO_BC.fifo[tail].rtaddress=3)and(sIntFIFO_BC.fifo[tail].transrec=1)and(sIntFIFO_BC.fifo[tail].subaddress=3))//比对消息(RT 地址 3 中子地址 3,发送)传输过程是否已记录在队列中
then begin
status:=BusTools_BC_MessageRead(Card_1553,2,@msg_buffer_read);//读该消息携带的数据字
Edit1.Text:=InttoStr(msg_buffer_read.data[0][0]);end;//数据字显示
inc(tail);tail:=tail and sIntFIFO_BC.mask_index;
sIntFIFO_BC.tail_index:=tail;end;
BC_Int:=API_SUCCESS;end;
//中断函数注册
sIntFIFO.function:=BC_Int;//中断服务例程指针传递
BusTools_RegisterFunction(Card_1553,@sIntFIFO1,REGISTER_FUNCTION);
//BC 消息配置时控制字应允许中断
bc_mess.control:=bc_mess.control or bc_message_interrupt;
```

3.2 RT 接收 BC 下传数据的中断采集

BC 向 RT 终端的某子地址发送数据,RT 终端应用程序采用中断方式读取 RT 接收缓冲区的数据,需采用四步进行。

第一步,RT 终端接收子地址允许接收中断,定义 RT 消息缓冲区写结构体变量:

msg_buffer_write;api_rt_mbuf_write,设定中断标志参数 BT1553_INT_END_OF_MESS,调用 api 函数 bustools_rt_messageWrite 将中断参数写入板卡,例如允许 RT 地址为 5 的子地址 4 中断的伪代码:

```
msg_buffer_write.enable:=BT1553_INT_END_OF_MESS;//允许接收消息时中断
status:=BusTools_RT_MessageWrite(Card,5,4,1,0,
```

```
@msg_buffer_write);//配置 RT 的子地址 4 允许中断
```

第二步,定义 RT 接收中断处理函数 RT_Int,在中断函数中设置变量 rtIntFIFO:api_int_fifo,msg_rt_read;api_rt_mbuf_read,RT 接收数据产生的中断信息保存在中断队列 rtIntFIFO 中,再调用 bustools_rt_messageRead 函数在中断函数中采集 RT 接收缓冲区的数据,代码及含义如下:

```
function RT_Int(cardnum:integer;sIntFIFO:pointer):integer;stdcall;
var
RT_IntFIFO:api_int_fifo;
tail:integer;//
msg_rt_read;API_RT_MBUF_READ;//RT 的消息接收缓冲区读结构体
begin
while(tail<>RT_IntFIFO.head_index)do
begin
if((RT_IntFIFO.fifo[tail].rtaddress=5)and(RT_IntFIFO.fifo[tail].transrec=0)
and(RT_IntFIFO.fifo[tail].subaddress=4))//子地址 4 产生接收中断
then begin
status:=BusTools_RT_MessageRead(Card_1553,5,4,0{bc->RT},0,@msg_rt_read);
Edit1.Text:=IntToStr(msg_rt_read.mess_data[0]);
end
...//移动队列元素指针直到结束队列轮循
RT_Int:=API_SUCCESS;//中断处理函数返回
end;
```

第三步:调用 bustools_registerfunction 注册中断函数;

第四步:启动板卡为 RT 模式,此时 RT 端将处于监听中断模式,一旦 BC 向 RT 发送数据,RT 的读缓冲区数据中断函数将响应,完成 RT 接收数据的中断采集处理。

4 消息数据的动态更新

4.1 BC 消息的数据动态更新

在 BC→RT 通信时,BC 板卡在初始化时即将各条消息配置在板卡硬件上,要更新 BC 消息的数据区,对于周期消息,在消息发送过程中,根据消息号 messno 索引消息的数据区,再次调用 api 函数 bustools_bc_messageWrite 对数据区域进行更改即可。更改数据后,在下一个 1553B 总线消息传输周期传输的数据即为更改后的数据。但周期消息应保证更新数据的时间小于 BC 消息的传输周期,确保 BC 消息在发送时消息中的数据已准备好,否则会引起 RT 端接收到错误数据的情况。

4.2 RT 上传数据的动态更新

RT 上传数据的更新是通过向 RT 发送缓冲区写入 RT→BC 消息数据来实现,更新数据必须确保 RT→BC 消息发送前准备好 RT 中的数据,否则消息传输时在 BC 端将采集到错误的数据。定义 RT 数据区写结构体 rt_write;api_rt_mbuf_write,由 rt_write.messdata[] 索引缓冲区的数据,调用 api 函数 bustools_rt_messageWrite 将 rt_write.messdata[] 的数据写入 RT 发送缓冲区完成 RT 端数据的动态更新,伪代码如下:

```
rt_write;API_RT_MBUF_WRITE;//定义 RT 缓冲区写结构体
```

```
for i:=0 to 31 do
rt_write.mess_data[i]:=i;//根据 RT→BC 消息更新 RT 数据
status:=BusTools_RT_MessageWrite(Card_1553,RTAdress,5,2,
1,@rt_write);//将更新的数据写入板卡硬件中
```

5 BM 总线监控功能的实现

将板卡初始化为 BM 模式，该卡可监视 1553B 总线上的所有消息传输活动，在总线传输活动的事后分析中具有重要意义。

BM 总线监控数据结构由 api_bm_mbuf 数组表示。api_bm_mbuf 结构体主要成员定义如表 3 所示。

表 3 api_bm_mbuf 结构体主要成员及含义表

结构体成员	数据类型	成员含义
messno	word	消息标识号
time	bt1553_time	消息传输时间码
status_cl	word	消息类型区分符 =1;bc→rt; =2;rt→bc; =3;rt→rt; =4;mode;
commandl	word	消息命令字
response1	bt1553_bmresponse	消息响应时间(us)
status_sl	word	消息状态字
value	array[0..31]of word	消息携带的数据

根据表 3 定义 bm_read 总线监视记录的数据结构，调用 api 函数 bustools_bm_messagereadblock 将监视的总线活动信息记录在 bm_read 数组中，对数组的元素逐一解析得到 1553B 总线的活动情况。具体解析时通过 command 命令字的移位和与运算，可解析出消息号 messno 对应的 rtaddr、sub-addr、收发标志 (t/r) 以及该消息携带的数据个数 num 等。

6 PCI-1553B API 函数调用顺序

6.1 BC 模式

板卡初始化为 BC 模式时，实现消息传输的 api 函数调用顺序及功能如图 1 所示。

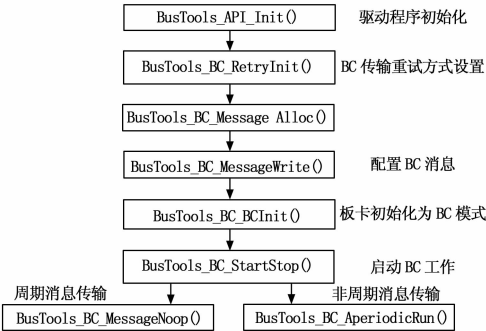


图 1 BC 模式下板卡 API 函数调用顺序图

6.2 RT 模式

板卡初始化为 RT 模式时，其 api 函数调用顺序及功能如图 2 所示。

6.3 BM 模式

板卡初始化为 BM 模式时，其 api 函数调用顺序及功能如

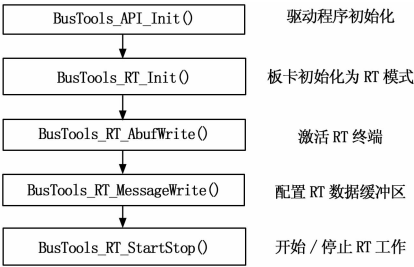


图 2 RT 模式下板卡 API 函数调用顺序图

图 3 所示。

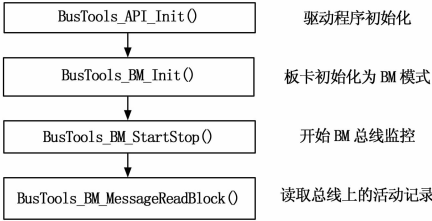


图 3 BM 模式下板卡 API 函数调用顺序图

7 应用案例

在电控模拟器研制项目中，电控模拟器由上位机和下位机组成。上位机与下位机通信采用 1553B 总线。分别在上位机和下位机中安装 Conder 公司的 PCI-1553B 总线卡，其中上位机设置为 BC 端，下位机作为 RT 端，上位机通过 BC→RT 消息向下位机发送电控命令，下位机接收到电控命令后控制相关的电磁继电器动作，并将下位机中的电源电压、电流检测结果周期性的通过 RT→BC 消息发送到上位机中并显示出来。上位机/下位机应用编程均采用 Delphi 平台，BC→RT 以及 RT→BC 通信数据的采集均采用中断方式。

下位机 RT 的地址设置为 4，接收电控命令的子地址设置为 5，电控命令属于 BC→RT 异步单次消息传输，消息的数据个数为 5，其数据格式如下：

0x01	K1	K2	K3	K4	K5
------	----	----	----	----	----

其中，第一个标识 0x01 代表有效的电控命令，K1~K5 代表 4 个继电器开关控制量，值为 0/1 分别关闭/接通继电器。

下位机 RT 上传 5 路电压信号，存放在 RT 子地址 4 中，上传的 5 路电流信号，存放在 RT 子地址 3 中，利用两条 BC→RT 的周期消息按照 512ms 周期向 BC 上传数据。电压与电流的数据格式如下：

0x02	V1	V2	V3	V4	V5
------	----	----	----	----	----

0x03	I1	I2	I3	I4	I5
------	----	----	----	----	----

电控模拟器上位机的 1553B 总线通信界面如图 4 所示。采用本文介绍的 Delphi 下 1553B 总线编程方法，实现了电控模拟器的上下位机有效通信。



图 4 电控模拟器上位机 1553B 总线通信控制界面

8 结论

本文介绍了 Delphi 平台下 Conder 公司的 PCI-1553B 总线卡通信应用编程技术，该技术可将 Delphi 强大的数据库编程和界面编程技术很好的与 1553B 总线通信编程技术结合起来，从而加快项目的研制进度。其次，本文介绍的编程方法包含构建一个基本的 1553B 总线通信系统的各关键部分代码，

并依据该技术实现了电控模拟器的通信部分设计，对应用该板卡的相关开发人员提供一定借鉴。

参考文献：

[1] 张世兵，苗克坚. 基于 1553B 总线的 BU-61580 芯片测试系统的设计与实现 [J]. 电子设计工程, 2013, 21 (3): 85-89.
[2] 杨 兵，于宗光，魏敬和. 一种 10MB/s 1553B 总线接口的设计 [J]. 微电子学, 2012, 42 (6): 782-786.
[3] 郑先成，王二虎，张晓斌. VxWorks 下 1553B 总线在飞机配电系统中的软件开发 [J], 计算机测量与控制, 2008, 16 (6): 874-876.
[4] 刘 瑛，刘一俊，邵玉梅. 1553B 总线通用仿真测试系系统设计与研究 [J]. 微计算机信息, 2009, 25 (3): 23236-238.
[5] 夏庆梅，徐亚军，熊华刚. 航空电子接口控制文件 数据库管理 [J]. 航空计算技术, 2001, 31 (3): 39-40.
[6] Conder 公司 PCI-1553B 卡库 busapi32net.vb 下载 [EB/OL]. http://ev.verysource.com/code/1188802_1_1/ busapi32net.vb.html.
[7] 何员子，李九英，等. DELPHI 环境下动态链接库的开 开发与应用 [J]. 电脑开发与应用, 2007, 20 (3): 4-7.

(上接第 2911 页)
器，以及测试完一套专用适配器中的产品后是否自动测试下一套专用适配器中的产品（若存在）。如不指定则默认为测试第一套专用适配器中的产品。而且程序可以自动显示正在测试的专用适配器中的产品序号。专用适配器切换控制部分程序流程设计如图 4 所示。

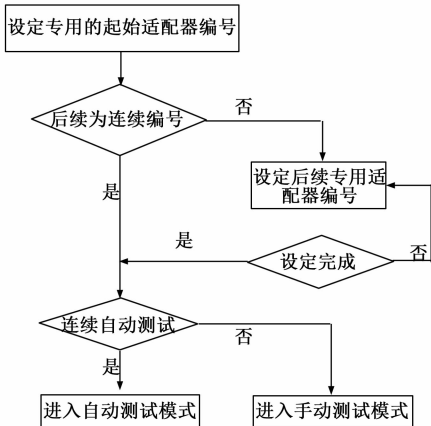


图 2 专用适配器切换控制部分流程图

3 实验与分析

3.1 自检实验

所谓自检实验就是利用 BMK 接收卡模拟被测飞行控制组件产品发送的 BMK 遥测数据，通过短接电缆连接至 BMK 遥测数据选择器的各个输入通道，而后再对 BMK 遥测数据选择器输出通道的数据进行接收采集，最后与发送的数据进行对比，以检验 BMK 遥测数据选择器各个通道的正确性。自检实验中对 BMK 遥测数据选择器的每个通道均进行了测试，每个通道选通后均进行 100 次递增数据、递减数据、同一数据、随

机数据 BMK 遥测数据帧收发实验。实验结果表明 5 个通道自检均正确。

3.2 拷机实验

拷机实验是对飞行控制组件测试系统和多通道 BMK 遥测数据选择器的综合测试。拷机实验连续进行 72 小时，保持设备不断电，同时每隔 4 小时对 5 套被测飞行控制组件产品重复进行 10 次综合性能测试。72 小时共对被测产品进行 190 次综合性能测试，测试结果均为合格。

4 结论

研制的多通道 BMK 遥测数据选择器结构精简、功能灵活、便于集成应用，可以有效缩短测试时间，提高测试效率，在飞行控制组件相关测试设备中有着广泛的应用前景。同时，自检实验和拷机实验表明研制的多通道 BMK 遥测数据选择器工作稳定、可靠。

参考文献：

[1] 刘莹莹. 某导弹飞控系统数字遥测信息接收装置的设计 [J]. 航空兵器, 2007, (1): 57-61.
[2] 任子龙，林青松. 基于 FPGA 的 BMK 遥测数据采集的实现 [J]. 航空兵器, 2010, (4): 40-42.
[3] 潘建国，等. 基于 FPGA 的高速大容量数据采集系统设计 [J]. 电子测量技术, 2008, 31 (9): 166-169.
[4] 周 健，等. CPLD 在惯性测量单元数据采集系统中的应用 [J]. 电子测量技术, 2008, 31 (11): 80-82.
[5] 单立超，等. 基于 FPGA 的混合遥测数据复接技术的研究 [J]. 嵌入式技术, 2012, 38 (10): 12-18.
[6] 宋学瑞，蔡子裕，段青青. 基于 FPGA 的数字复接系统的设计与实现 [J]. 计算机测量与控制, 2008, 16 (8): 1174-1176.
[7] 钱 渊，郭 颖. 基于现场可编程门阵列技术的遥测数据处理接口设计方法研究 [J]. 探测与控制学报, 2006, 38 (3): 60-63.
[8] S Palnitkar. Verilog HDL A Guide to Digital Design and Synthesis [M]. 2th Edition. SunSoft Press A Prentice Hall Title, 2003.