

# 基于 ASAAC 标准的 BIT 设计

任国鹏, 柴小丽, 蒋琪明

(中国电子科技集团公司第三十二研究所, 上海 200233)

**摘 要:** 为满足航空电子系统的可测试性, 降低航空电子系统测试维护成本, 结合 ASAAC 标准设计一个机内自测试(BIT)系统。BIT 利用自身资源对系统进行故障检测或隔离, 采用总线层次化方法将 BIT 设计分为系统级、分系统级及模块级 3 层测试结构, 并给出模块级测试的软硬件设计方案。测试结果表明, 该系统能使测试与航电系统的健康管理和故障管理紧密结合, 在满足测试覆盖率等技术指标的同时, 提高航空电子系统可靠性与可测试性。

**关键词:** ASAAC 标准; BIT 设计; 航空电子系统; 故障管理; 可测试性; 可靠性

## Design of BIT Based on ASAAC Standard

REN Guo-peng, CAI Xiao-li, JIANG Qi-ming

(The 32nd Research Institute of China Electronics Technology Group Corporation, Shanghai 200233, China)

**【Abstract】**To meet the avionics electronic system testability and reduce avionics electronic system test and maintenance costs, this paper designs a Built-in Test(BIT) system referring to ASAAC standard. BIT is defined as a capability provided by electronic system itself for the fault detection or isolation. As a Hierarchy design, it divides the BIT system into three levels, system level, subsystem level and module level, and makes a design in details containing software and hardware design for module level test. Test results show that the system can make the test combining with the health management and fault management of avionics electronic systems more closely. It meets technical indicators and greatly improves avionics electronic system reliability and testability.

**【Key words】** ASAAC standard; BIT design; avionics electric system; fault management; testability; reliability

DOI: 10.3969/j.issn.1000-3428.2012.12.068

### 1 概述

随着机载计算机技术不断发展, 航空电子系统设计的复杂度大大提高, 系统高额的测试和维护费用已经成为航空电子系统设计中不可忽略的问题。为降低系统后期的测试与维护成本, 在航空电子系统内加入可测试性设计已经成为航电系统测试与维护的主流解决方案, 而机内自检测(Build in Test, BIT)技术以其故障检测快捷简便、定位隔离准确等特点逐渐为设计人员所重视。

BIT 是指利用系统自身资源进行自检测试, 监控系统的运行状态<sup>[1]</sup>(可用、不可用), 继而实现故障诊断和隔离等功能的一种测试方法。采用 BIT 设计能够大大提高复杂的电子系统故障诊断、定位和恢复的能力。由于新一代航空电子系统普遍采用综合模块化的设计理念, 在航电系统中使用 BIT 测试, 就要求设计人员在系统设计初期充分考虑综合模块化的系统结构以及测试要求, 根据测试要求结合系统自身可利用资源, 附加部分的测试电路实现 BIT 测试功能。

ASAAC 标准是欧洲联合标准航空电子委员会定义并验证了的一套开放型、综合化、模块化的先进航空电子体系结构标准<sup>[2]</sup>。作为综合模块化航空电子结构设计的主要标准之一, ASAAC 标准从软件结构、机械结构、网络功能和通信功能等方面对航电系统进行了规定。参照 ASAAC 标准进行航电系统 BIT 设计, 考量航电系统测试要求, 可以大大提高 BIT 测试的通用性以及可拓展性, 提高航电系统的可靠性, 降低航电系统的测试和维护成本。

本文参照 ASAAC 标准, 以通用处理模块为目标, 进行 BIT(机内自测试)设计, 主要包括 BIT 硬件设计、软件结构设计以及测试信息传输接口设计。

### 2 ASAAC 标准与 BIT 总体设计

ASAAC 标准根据模块功能的不同<sup>[3]</sup>, 将航电系统的模块分为数据处理模块、图形处理模块、大容量存储模块、电源转换模块、网络支持模块等, 并对模块上的软件架构、硬件设计进行统一规范, 航空电子系统设计时将各模块按照各自功能有机组成分系统, 再将各分系统综合成一个电子系统, 从而完成电子系统综合模块化的整体设计。ASAAC 模块软件体系结构如图 1 所示。

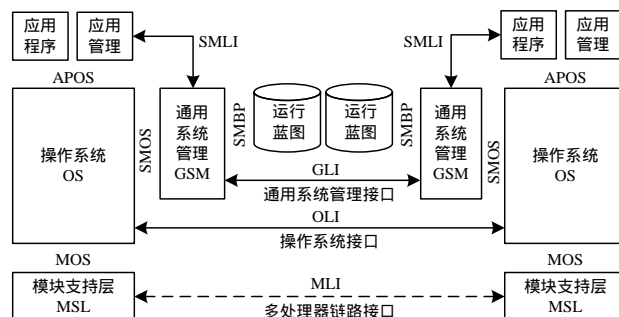


图 1 ASAAC 模块软件体系结构

ASAAC 模块软件结构从整体上主要分为模块支持层、操作系统层和应用层。

(1) 模块支持层: 负责与模块硬件直接通信, 封装了基础硬件的细节并提供对底层资源通用的独立访问。模块间建立

**作者简介:** 任国鹏(1988 - ), 男, 硕士研究生, 主研方向: 嵌入式计算技术; 柴小丽, 研究员; 蒋琪明, 高级工程师

**收稿日期:** 2011-11-01 **E-mail:** dragon6@mail.ustc.edu.cn

多处理器链路(MLI)接口通信,通过 MOS 接口使得操作系统与硬件接口无关。

(2)操作系统层:包括操作系统(OS)和通用系统管理(GSM)两部分,操作系统提供基本的操作系统服务,包括进程管理、存储管理等,并通过 SMOS 接口为通用系统管理提供服务;通用系统管理(GSM)主要包括健康管理、故障管理、配置管理、安全管理等,通过 SMBP 接口访问系统蓝图实现对系统的配置和管理。GSM 之间通过 GLI 系统管理逻辑接口进行上下级模块间的 GSM 通信,实现系统对下级模块的控制,接受上级模块的管理。

(3)应用层:包括应用和应用管理两部分,应用为航电系统应用程序实现,包括雷达检测等功能;应用管理是对这些应用程序进行管理。

结合 ASAAC 标准,BIT 测试可归结至 GSM 通用系统管理中,系统通过 GSM 调用自身或下级模块上的模块支持层 BIT 功能,为系统的健康管理和故障管理提供依据。

参照 ASAAC 标准<sup>[4]</sup>,航电系统测试采用总线层次化体系进行设计,系统嵌入式测试分为 3 个层次如图 2 所示,分别是模块级、分系统级和系统级测试。模块级嵌入式测试主要进行各类测试信息采集、上传测试信息、预留测试维护端口,并可根据用户需求进行简单的故障处理,如故障告警、故障隔离等。分系统级嵌入式测试接收来自模块级的嵌入式测试信息并进行故障诊断推理,根据推理结果对功能模块进行故障处理,将处理结果上传至系统级;系统级嵌入式测试接收来自分系统的测试信息,并在核心处理机中进行诊断推理与预测,显示故障和异常告警信息。

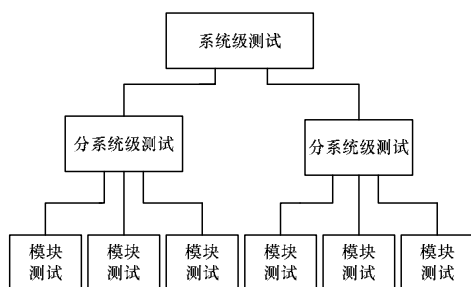


图2 ASAAC 测试结构

整个航空电子系统由各模块有机组成,由部分功能模块紧密地组合成分系统,分系统由一个处理模块统一管理;再由分系统构成整个系统,由综合处理模块统一管理。在每个模块上都有着相同的 ASAAC 软件结构,但负责通用系统管理的 GSM 部分略有不同,普通模块上 GSM 只负责本模块上的 BIT 测试;分系统中处理模块上 GSM 作为上级 GSM,在完成自身 BIT 测试的同时,通过控制其他模块 BIT 测试并接受测试结果来实现分系统的健康管理和故障处理;系统级的综合处理模块 GSM 是最高级的 GSM,完成自身 BIT 测试同时,负责控制处理分系统级 BIT 测试,对各分系统进行健康管理和故障处理。

从测试层次上划分,BIT 测试主要实现在模块级测试上,负责完成模块测试、通过传输接口将测试信息传送给分系统级处理以及接收来自分系统级的处理指令。其中,分系统级测试运行在分系统中的处理模块上,系统级测试运行在整个航空电子系统中的综合处理模块上。

从软件体系结构上划分,BIT 设计<sup>[5]</sup>主要是实现模块支持层的测试功能,与此同时为本地 GSM 提供 BIT 功能调用的 MOS 接口和与外地 GSM 通信的 MLI 接口,前者与通用

系统管理和操作系统间 SMOS 接口相对应,负责本地 GSM 与模块支持层测试通信,接收模块上测试指令、传送测试结果到 GSM;后者负责支持本地模块支持层与外地分系统处理模块上 GSM 通信,接收分系统测试指令并发送测试结果到分系统上。

本文设计模块级 BIT 根据 ASAAC 要求的总体结构如图 3 所示。

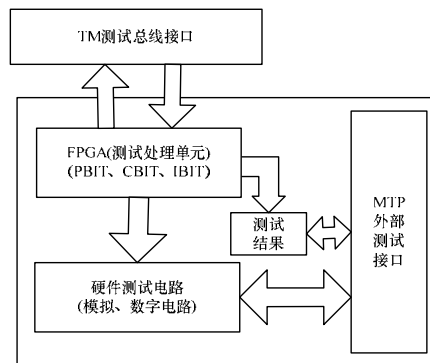


图3 BIT 总体测试结构

主要完成以下 6 点功能:

(1)上电自测试(PBIT)。作为自举过程的组成部分用于检查模块硬件的状态。在来自模块外部的任何控制被应用前,这些测试作为 MSL 的组成部分自动运行,在系统上电时对模块进行测试,将测试结果存储在模块中,并上传测试信息。

(2)周期自测试(CBIT)。正常操作期间模块健康的连续检查,允许运行特定类型的测试或全部测试,根据需求可以按一定周期对模块监控,并可以主动上传测试信息。

(3)初始自测试(IBIT)。作为故障管理过程的组成部分用于检查模块硬件的状态。为了在故障定位期间有帮助,IBIT 执行模块的全面测试。在模块运行中根据本地 GSM 或上级 GSM 的指令对模块进行测试,将测试结果存储在模块中,上传测试信息或由系统获取。

(4)测试总线接口(TM)。是连接系统内各模块 BIT 测试的物理接口与 ASAAC 软件结构中 MLI 层相对应。模块 BIT 测试后,通过 TM 与分系统处理模块和系统综合处理模块进行测试信息通信,包括获得测试指令、传输测试结果等功能。

(5)MTP 接口(测试与维护接口)。支持模块与外部测试连接,支持读取 BIT 测试结果、调用模块测试单元等功能。

(6)BIT 测试处理器。采用 FPGA+IP 核实现,根据从 TM 总线上接收的测试命令展开测试,通过驱动硬件测试电路上获得的模块上的测试信息,按照故障诊断的算法产生模块的测试结果。测试结果可以通过 MTP 接口提供给外部设备,也可以通过 TM 总线接口传送给上层分系统处理模块中。

### 3 BIT 系统的硬件设计

根据 ASAAC 标准和 BIT 测试的总体需求,BIT 系统硬件需要由以下 6 个部分组成:

(1)一个测试处理芯片。测试芯片由一块专用的 FPGA+IP 核实现,采用这样的设计方法有较高的灵活性,一些辅助的外围电路可以通过 FPGA 编程实现,可以节省开发时间,在设计定型后可以考虑改用单片机+功能芯片的方式替换方案。

(2)一个数字测试电路。数字测试电路主要负责模块上数字电路的测试,主要根据 IEEE1149.1 边界扫描协议<sup>[6]</sup>进行测试,通过 TDI、TDO 2 条 IO 线将模块上带有 JTAG 接口的芯片连接成一条扫描链,不具备边界扫描 JTAG 接口的设备可

根据其电路连接和结构采用边界扫描置入的方法,通过置入芯片将待测的设备加入扫描链路中,采用簇测试的方法完成测试。

(3)一个模拟测试电路。模拟测试电路主要负责模块上模拟电路的测试。对模拟电路进行设计首先要在电路可测性设计时人为插入模拟探测节点,按照预先设定好的规则定时获取状态信息,主要用来探测电路板上各探测点的温度、电流、电压等模拟量。通过使用不同的传感器、检波器或相关电路将待检测信号转换成一定的电流或电压信号,最终经过 A/D 转换器转换成数字信号并通过测试总线传递到测试控制器,由控制器来对这些模拟测试节点进行信息处理、融合以获取是否发生故障的信息。

(4)2 个存储单元。一个存储单元 rom,负责存储 PBIT、CBIT、IBIT 的测试程序;一个存储单元 Flash 负责存储测试结果。分开存储测试程序和测试结果,主要是为了满足航电

系统上电测试的时效需求,另外也方便测试程序更新。

(5)一个测试总线和协议转换芯片。TM 测试总线负责实现模块间通信的 MLI 接口,模块与分系统处理模块间测试指令和测试信息传递的主要接口。由于测试芯片 FPGA 通过 JTAG 链路获取测试信息,单独采用一套 JTAG 链路作为测试总线会增加系统内连接接口,使得总线维护更为困难,因此 BIT 设计时可以根据系统内总线特征通过协议转换芯片将测试总线与系统内传输总线复用,如系统采用 CAN 总线,测试单元可以通过一个 JTAG 到 CAN 的转换芯片实现总线复用。

(6)一个 MTP 接口。MTP 接口负责与外部主机或 ATE 通信,可以通过其向板内存储器进行读写操作,MTP 接口上的信号应该包括 JTAG 总线信号 TCLK、TMS、TDI、TDO、TRST\* 和电源、地信号。FPGA 与 MTP 接口的信号是 FPGA 本身的 JTAG 信号,还可以用来向 FPGA 烧录程序。

BIT 系统硬件设计如图 4 所示。

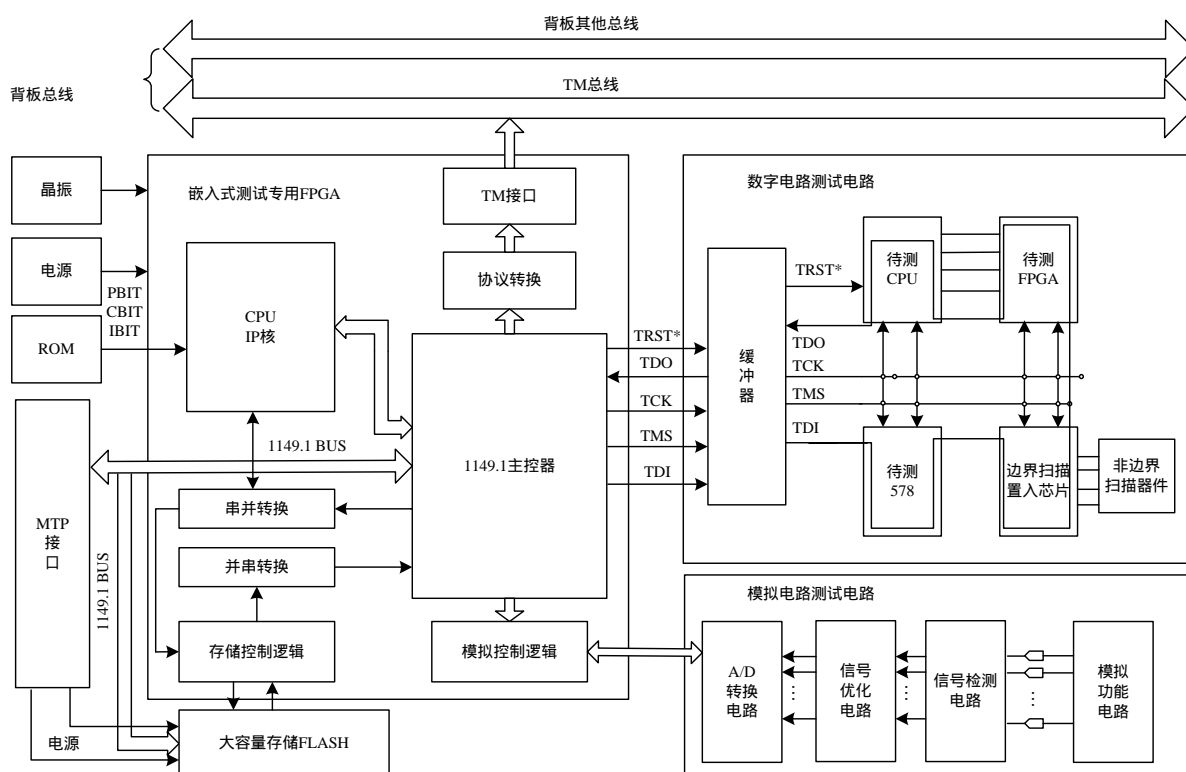


图 4 BIT 模块级硬件设计

BIT 硬件设计可以为 BIT 测试软件提供必要的测试资源,通过这些测试资源,BIT 系统可以方便地获得模块上各部分的工作状态,通过结合各自预订工作状态来检测模块故障,一个好的 BIT 硬件设计应尽最大限度提高测试的覆盖率,使得测试的结果具有较高的准确性和说服力。

#### 4 BIT 系统软件结构设计

BIT 软件结构可以从两方面着手进行设计,一方面为 BIT 的测试功能实现,另一方面为 BIT 接口实现。BIT 测试功能<sup>[7]</sup>实现体现在测试程序 PBIT、CBIT、IBIT 的设计上,根据测试需求,BIT 测试流程如图 5 所示。在系统上电启动后首先进行上电机内自测试 PBIT,其中 PBIT 测试应该包括上电电源检测、CPU 工作状态检测、模块内存检测、时钟测试、存储模块测试以及网络等接口测试。在完成 PBIT 测试后需给出测试结果。如果未通过 PBIT 测试,则需保存测试结果供维护人员查看。

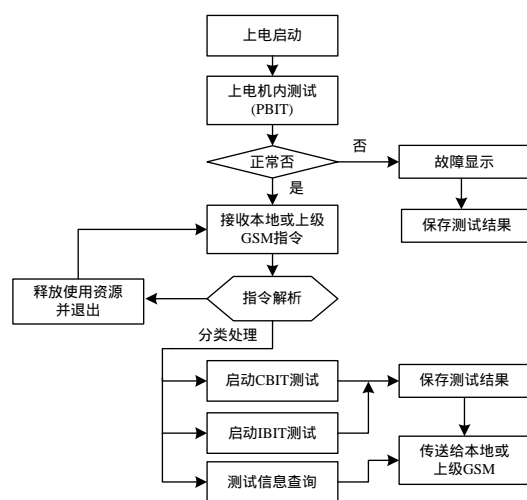


图 5 BIT 软件测试流程

在模块正常启动后, 可根据接收到的系统指令, 进行 CBIT、IBIT 测试或者对测试信息进行查询。CBIT 连续机内测试可以根据模块的运行状态, 利用模块空闲时间对模块进行检测, 应支持测试中断, 一旦模块处于工作状态, 退出 CBIT 测试保存测试状态, 下次进入 CBIT 时可选择从上次的测试中断点开始测试。测试完成后保存测试结果, 传送给本地或上级 GSM。

IBIT 初始自测试是根据接收到的系统指令执行模块的全面测试。这些测试可以经 MLI 链路在某个控制模块上的 GSM 控制下远程运行, 或者经一个 MOS/SMOS 调用(startIbit)从已经可用的本地 GSM 运行。测试完成后保存测试结果, 传送给本地或上级 GSM。由于 IBIT 可能具有破坏性, 导致测试完成后模块不可使用, 如内存全面检测后模块死机, 因此使用 IBIT 时应注意不影响系统安全。

执行 PBIT、CBIT、IBIT 后产生的测试结果可以根据系统指令进行调用。测试维护人员既可以通过模块自身 GSM 来调用测试结果, 也可以通过 MLI 由远程的分系统处理模块调用测试结果。在连接外部测试设备时, 测试维护人员还可以 MTP 接口调用 FLASH 中的测试结果。

BIT 接口实现主要依照 ASAAC 标准<sup>[8]</sup>设计, 按照 ASAAC 标准的要求为本地通用系统管理(GSM)访问模块支持层上的 BIT 功能提供 SMOS、MOS 接口, 为远程分系统处理模块上 GSM 与本地模块支持层通信提供 MLI 接口。

模块支持层 BIT 应为系统提供与硬件无关的 MOS 接口, 该接口包括启动测试和获取测试结果两方面且与 SMOS 接口一一对应, 主要包括函数:

#### (1) 上电自测试 PBIT 函数

Int startPbit(bitstatus pbitstatus);

startPbit 是模块上电时必须进行的上电自测试, 上电完成后将测试结果存在 PbitResult 中, 如果测试函数正确运行返回 SUCCESS, 否则返回 ERROR。如 PBIT 测试通过则将 pbitstatus 标识为 OK, 否则将 pbitstatus 标识为 FAILED, 并在启动界面上输出测试到的故障信息供测试人员查看。

#### (2) 启动 CBIT 连续自测试函数

int startCbit(unsigned int test\_code, CbitModeType mode, CbitTestStatus cbit\_test\_status);

startCbit 函数负责启动 CBIT 测试, 根据输入参数 test\_code 和 mode 对模块进行测试, 函数运行后返回 cbit\_test\_status, 并保存测试结果。CBIT 是模块运行时主要的测试方法, 它可以根据系统需要周期性对模块进行监控, 在故障发生时将故障信息汇报给 GSM, 或者有 GSM 周期调用 CBIT 测试结果获取函数取得故障信息。

参数 mode 配置 CBIT 执行 2 种不同测试方式: COMPLETE(完整的)模式和 PARTITIONED(部分的)模式。完整模式下 CBIT 在一次调用中完成测试, 操作系统 OS 可以根据系统运行情况抢占 startCbit, 抢占后从断点继续 CBIT 测试; 部分模式下 CBIT 只完成 test\_code 指定的部分模块测试, 操作系统 OS 不应抢占这个资源。

cbit\_test\_status 是 CBIT 运行结束后返回的状态, 用于标识 startCbit 函数运行状态, PASSED 代表最近一次 CBIT 测试调用完成并通过; ONGOING 只在部分测试模式下有效, 直到所有测试完成后改变成 PASSED 或其中有一次测试失败变成 FAILED; FAILED 标识最后一次的 startCbit 测试失败。

test\_code 是 CBIT 选择的测试项代号, 0 代表进行所有测试项测试, mode 自动变成 COMPLETE(完整的)模式; 其他

代号分别代表 CPU、内存、FLASH、网络等测试项。

如果 startCbit 被正确调用, 则返回值为 SUCCESS, 否则返回 ERROR。

#### (3) 启动 IBIT 初始自测试函数

Int startIbit(bitStatus ibitstatus)

startIbit 函数负责启动 IBIT 测试, 测试完成后将测试结果存储在 IbitResult 中, 如果 IBIT 正确运行则返回 SUCCESS, 否则返回 ERROR。如 IBIT 测试通过则将 ibitstatus 标识为 OK, 否则标识为 FAiLED。

#### (4) BIT 测试结果获取函数

int getPbitResult(bitResult pbit\_result);

int getCbitResult(bitResult cbit\_result);

int getIbitResult(bitResult ibit\_result);

BIT 测试结果获取函数结构相似, 只是获得的测试结果存放的位置不同, 其中, getPbitResult 函数负责获取上电启动的 PBIT 的结果; getCbitResult 函数负责获取当前 CBIT 的测试结果; getIbitResult 函数负责获取最近启动的 IBIT 的结果。若以上函数能获得请求的 PBIT、CBIT、IBIT 结果(注意: 结果是 OK 还是 FAILED 并无关系), 则它应返回值 MOS\_BIT\_CALL\_OK。若本函数不能获得 PBIT、CBIT、IBIT 结果或使用 PBIT、CBIT、IBIT 测试结果服务之前来发动获取 BIT 测试结果, 则它应返回值 MOS\_BIT\_CALL\_FAILED。

bitResult 结构定义为:

```
Struct bitResult{
    BitFinalResult    bit_final_result;
    bitDetailedResult bit_detailed_result;
};
```

其中, BitFinalResult 为枚举型变量, enum BitFinol Result {BIT\_FINAL\_RESULT\_OK, BIT\_FINAL\_RESULT\_FAIL}; bitDetailedResult 为字符数组 struct bitDetailedResult{char bit\_result[MAX\_BIT\_DETAILED\_RESULT];} bit\_result 数组根据 BIT 测试的测试项目多少定义数组大小。

其次模块支持层还应为外地分系统处理模块 GSM 或系统综合处理模块 GSM 调用提供对应的 MLI 接口, 如对测试总线 CAN 总线的支持。远程 GSM 通过调用 CAN 接口将测试指令传输给本地模块并取回测试信息。MLI 接口上主要包括以下函数:

#### (1) 测试命令发送函数

Int sendBitTest (tId id, tMode mode, unsigned int test\_code,

CbitModeTye cbitmode)

sendBitTest 函数负责分系统处理模块或系统综合处理模块上 GSM 传送测试指令到待测模块, id 标识待测模块的 id 号, mode 指示待测模块进行哪种模式的测试, CBIT\_COMPLETE 表示待测模块执行 CBIT 全面测试; CBIT\_TESTCODE 表示待测模块执行 CBIT 指定测试项测试, 由 test\_code 和 cbitmode 标识待测试的指定项; IBIT\_TEST 标识待测模块执行 IBIT 测试。函数运行正确后返回 SUCCESS, 否则返回 ERROR。

#### (2) 测试信息获取函数

Int receiveBitResult(tID id, bitResult\* rbitResult)

receiveBitResult 函数负责分系统处理模块或系统综合处理模块上 GSM 获取下级测试模块的测试结果, id 标识待收集测试结果模块的 id 号, rbitResult 指定为存储测试结果的变量。函数正确运行后返回 SUCCESS, 否则返回 ERROR。

操作系统层将 MOS 层的 BIT 接口按照对应关系, 为 GSM

(下转第 235 页)

图8分别产生240 MB/s、320 MB/s、360 MB/s和400 MB/s的测试数据, 设测试数据产生速度为 $V_1$  MB/s, 2个SATA通道的DMA传输速度分别为 $V_a$  MB/s和 $V_b$  MB/s, 图8(a)和图8(b)分别表示产生240 MB/s时, 各SATA存储通道的瞬时DMA存储速度统计, 可以看出2个存储通道的DMA速度稳定在123 MB/s, 匹配于测试数据的产生速度。图8(c)和图8(d)、图8(e)和图8(f)分别表示产生320 MB/s和360 MB/s测试数据时, 各存储通道的瞬时DMA速度统计。可以看到, 随着测试数据产生速度提升, 2个SATA存储通道的瞬时DMA存储速度也相应提高, 且始终匹配于测试数据的产生速度, 传输速度稳定。图8(g)和图8(h)表示产生400 MB/s测试数据时存储通道的瞬时DMA速度统计, 可以看出此时单个存储通道的存储速度最高达到了200 MB/s, 但是速度稳定度不高, 存储过程中频繁出现速度陡降, 存储速度由稳定时的200 MB/s降至80 MB/s。此时将不再满足嵌入式实时存储系统的需要。

实验表明, 单个SATA通道可以实现180 MB/s的稳定的传输速度。Xilinx V5-FX70T芯片有11 200个Slice资源, 148个36 KB BlockRAM资源, 而单个SATA通道仅占用1 285个Slice资源和24个36 KB lockRAM资源, 根据资源占用率计算, 在Xilinx V5-FX70T芯片中, 至少可以例化实现4个SATA通道。当多个通道并行工作的情况下, 通过一片FPGA就可以实现700 MB/s以上的传输速度。目前, 市面上已经出现容量大于512 GB的SATA固态电子盘, 4个通道即可实现高达2 TB的存储容量, 固态电子盘抗振动性能优异, 可以应用于车载机载等环境中。据此构建的存储系统可以显著提高存储速度和存储容量, 应用广泛, 具有很强的应用价值和现实意义。

## 5 结束语

在传统的嵌入式存储系统中, PCI存储系统简单通用, 但是存储带宽较低。FLASH存储系统具有较高的存储带宽, 但是牺牲通用性与扩展性。本文提出了一种基于SATA的嵌入式直接存储系统的设计方法, 该方法在FPGA中建立了

SATA存储通道, 包括完整实现SATA物理层和链路层协议; 根据嵌入式存储需求组合SATA传输层和应用层协议; 建立DMA控制模块管理数据流的存储。实际应用结果表明, 单个SATA存储通道可以稳定地达到180 MB/s的存储带宽, 而通过扩展SATA存储通道即可实现存储系统在容量与带宽上的扩展。

## 参考文献

- [1] 张俊杰, 章凤麟, 叶家俊. 高速数据采集系统设计[J]. 计算机工程, 2009, 35(1): 207-209.
- [2] 陈庚锋, 吴顺君, 王翠平. 基于PCI总线的高速实时数据采集系统[J]. 电子设计应用, 2004, (3): 51-53.
- [3] 王超, 刘伟. 基于文件系统的高速数据记录系统[J]. 北京理工大学学报, 2010, 30(5): 543-547.
- [4] Kim J, Kim J M, Noh S H, et al. A Space-efficient Flash Translation Layer for Compact Flash Systems[J]. IEEE Transactions on Consumer Electronics, 2002, 48(2): 366-375.
- [5] 陈晓东, 苏宛新, 王化龙. 嵌入式SATA存储系统的研究[J]. 电子技术应用, 2008, 34(4): 28-30.
- [6] Serial ATA International Organization. Serial ATA Specification Rev. 2.5 Gold[Z]. 2005.
- [7] Wu Wei, Su Haibing. A High Performance Serial ATA Host Controller[C]//Proc. of 2008 IEEE International Conference on Computer Science and Software Engineering. [S. l.]: IEEE Press, 2008: 18-23.
- [8] Xilinx, Inc.. Virtex-5 RocketIO GTP Transceiver User Guide[Z]. 2006.
- [9] Xilinx, Inc.. Virtex-5 FPGA Serial ATA Generation 2 Protocol Standard[Z]. 2007.
- [10] American National Standards Institute, Inc. AT Attachment with Packet Interface-6(ATA/ATAPI-6)[Z]. 2002.

编辑 索书志

(上接第231页)

提供相应的SMOS接口, 供本地GSM调用StartCBit()等测试函数接口, 以及getPBitResult()等测试结果获取接口。GSM通过上述函数接口调用本地模块或下级模块的BIT测试功能, 然后依据获取测试结果进行分析处理, 从而实现健康管理 and 故障处理的功能。

## 5 结束语

本文采用基于ASAAC标准BIT设计方法, 可以将BIT测试与航空电子系统的健康管理和故障管理紧密结合, 在满足BIT测试覆盖率等技术指标的同时, 提高航空电子系统可靠性、可测试性, 可以有效降低航电系统测试与维护的成本。设计的BIT测试方法虽然提高了系统可测试性, 但同时也在硬件设计上增加了冗余的测试电路, 如何保证引入冗余测试电路的同时保证系统的可靠性, 成为设计需要考虑的另一个重要问题。本文下一步将结合实践完善本文的测试性设计, 研究如何在保证电子系统可靠性的同时提高系统可测试性。

## 参考文献

- [1] 王运盛, 陈颖. ASAAC航空电子体系结构标准分析[J]. 电讯

技术, 2007, 47(5): 159-162.

- [2] 郑朝辉, 陈新中, 张晓先. 综合模块化航空电子系统的可靠性设计[J]. 计算机工程, 2009, 35(23): 272-273.
- [3] 易建平, 韩庆飞. 飞机综合模块化航电系统总体设计研究[J]. 科学技术与工程, 2010, 10(19): 4709-4714.
- [4] 联合标准化航电系统架构协会. ASAAC Standard: 00-78-2005 Proposed Standards for Architecture[S]. 2005.
- [5] 白恩杰. 一个计算机系统的BIT设计[J]. 电光与控制, 1993, (2): 42-45.
- [6] 崔凯, 孙允明. 嵌入式机载计算机系统的BIT设计与LRU测试[J]. 现代计算机, 1999, (1): 10-12.
- [7] IEEE Computer Society. IEEE Std: 1149.4-1999 IEEE Standard for a Mixed\_Signal Test Bus[S]. 1999.
- [8] 联合标准化航电系统架构协会. ASAAC Standard: Def Stan 00-74-2005 Proposed Standards for Software[S]. 2005.

编辑 索书志