

# 基于通用星载计算机 1553B 总线 IP 核的故障诊断及容错机制升级项目实施方案

西安电子科技大学 ICTT 实验室

2025 年 7 月 1 日

# Contents

<b>1</b>	<b>概述</b>	<b>2</b>
<b>2</b>	<b>引用文档</b>	<b>2</b>
<b>3</b>	<b>1553B 总线 IP 核及升级要求</b>	<b>2</b>
3.1	IP 核概述	2
3.2	IP 核的功能特性	2
3.3	升级要求	3
<b>4</b>	<b>升级策略</b>	<b>4</b>
4.1	升级流程	5
4.2	验证	5
<b>5</b>	<b>功能点及验证用例设计</b>	<b>6</b>
5.1	寄存器测试	6
5.2	RAM 测试	7
5.3	1553B 收发器拖尾测试	7
5.4	BC 功能点	7
5.5	RT 功能点	8
5.6	MT 功能点	8
<b>6</b>	<b>SOC 集成与验证</b>	<b>9</b>
<b>7</b>	<b>验证进度安排</b>	<b>9</b>
<b>8</b>	<b>质量保证措施</b>	<b>9</b>
<b>9</b>	<b>风险分析和解决措施</b>	<b>9</b>

## 1 概述

1553B 总线 IP 核是星载电子系统中重要的通信接口，负责实现控制器（BC）、远程终端（RT）和总线监控器（BM）之间的数据传输。采用 IP 核实现 1553B 控制器功能，不仅能够实现电子系统的小型化设计，而且对于降低功耗、降低成本同样具有重要的意义。与此同时，1553B IP 核的可靠性之间影响系统的可靠性，因此提高 1553B IP 核的可靠性具有重要意义。

## 2 引用文档

《基于通用星载计算机 1553B 总线 IP 核的故障诊断及容错机制升级任务书》

## 3 1553B 总线 IP 核及升级要求

### 3.1 IP 核概述

1553B 总线 IP 核集成了 BC、RT、MT 三种控制器，同时寄存器、存储器的访问方式兼容 61580 芯片，其整体架构如图1所示：

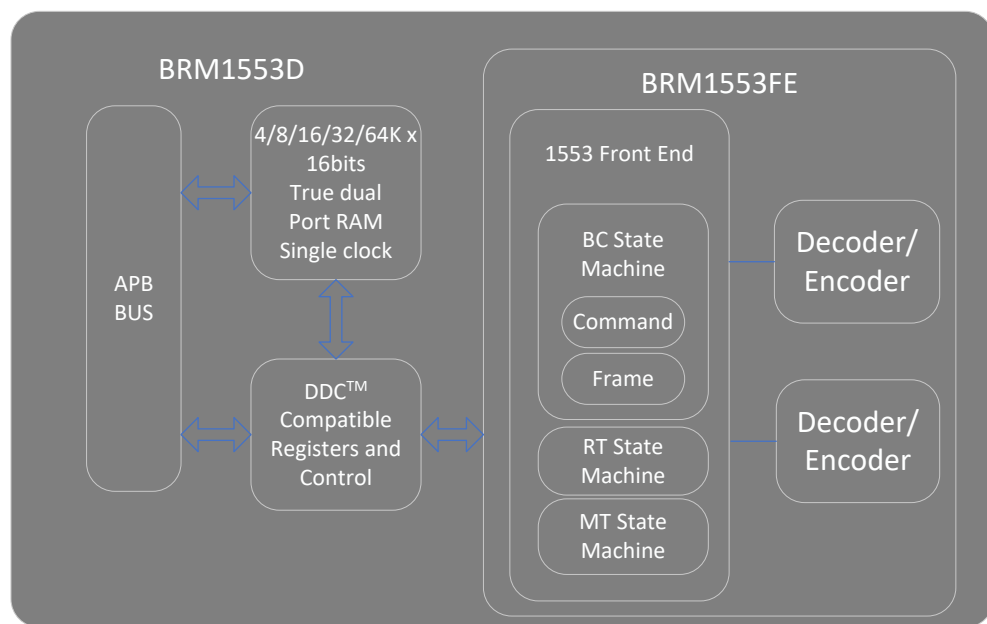


图 1: IP 核结构框图

### 3.2 IP 核的功能特性

1553B 总线 IP 核具备以下主要特性：

- 集成 BC、RT 和 MT 三种工作模式，支持通过软件灵活配置；
- 支持 1553B 总线协议规定的全部 10 种消息类型；
- 支持 A、B 双冗余通道，并可实现自主切换；
- 兼容 61580 芯片的寄存器访问方式及存储器管理机制；
- 提供标准双口 RAM，支持按用户需求裁剪容量；
- **BC（总线控制器）特性：**
  - 可配置的 A/B 工作区；
  - 多种可编程中断类型；
  - 自动重发功能，支持可编程的重发次数及通道选择；
  - 可编程的帧自动重复发送功能；
  - 可编程的消息间隔时间；
- **RT（远程终端）特性：**
  - 可配置的 A/B 工作区；
  - 多种可编程中断类型；
  - 可编程的 RT 地址；
  - 支持单缓冲、循环缓冲及双缓冲等多种存储器管理方案；
  - 可编程的非法命令表；
  - 可编程的 MODE 代码中断表；
  - 可编程的子地址忙表；
- **MT（总线监控器）特性：**
  - 支持字监视模式；
  - 支持可选的消息监视模式；

### 3.3 升级要求

针对本项目的升级任务，需重点开展以下工作：

- **已知缺陷修复：**
  - (1) 误触发中断：在特定工作场景下，存在重复触发中断的问题；
  - (2) 通道切换故障：在甲方研制的 RV 芯片上进行 1553B 总线通道切换测试时，无法按照 IP 核手册要求实现正常切换；

- (3) RAM 堆栈空间被改写：在某些场景下，1553B IP 核存储于 RAM 空间的状态信息及消息结果被异常覆盖；
  - (4) RT 中断状态无法清除：RT 工作于循环缓冲模式时，产生的中断状态无法被正常清除；
  - (5) 国产 1553B 收发器拖尾问题导致接收异常：使用国产收发器时，出现消息接收异常现象。
- 功能完善：
    - (1) MT 模式下，字监测功能需支持存储空间配置；
    - (2) 1553B IP 核的 RAM 空间由固定  $4K \times 16\text{bit}$ ，升级为  $4K \times 16$   $64K \times 16$  可配置；
    - (3) 开发专用的 1553B IP 核验证平台，提升验证效率和覆盖面。
  - 质量要求：
    - (1) 关键故障修复需达到预期效果，确保系统稳定可靠运行；
    - (2) 验证平台搭建需稳定可靠，能够支持多种故障场景的模拟与测试；
    - (3) 测试用例需充分覆盖全部功能点，确保高代码覆盖率；
    - (4) SOC 集成验证需确保 IP 核与甲方 SOC 原型系统的兼容性与稳定性。

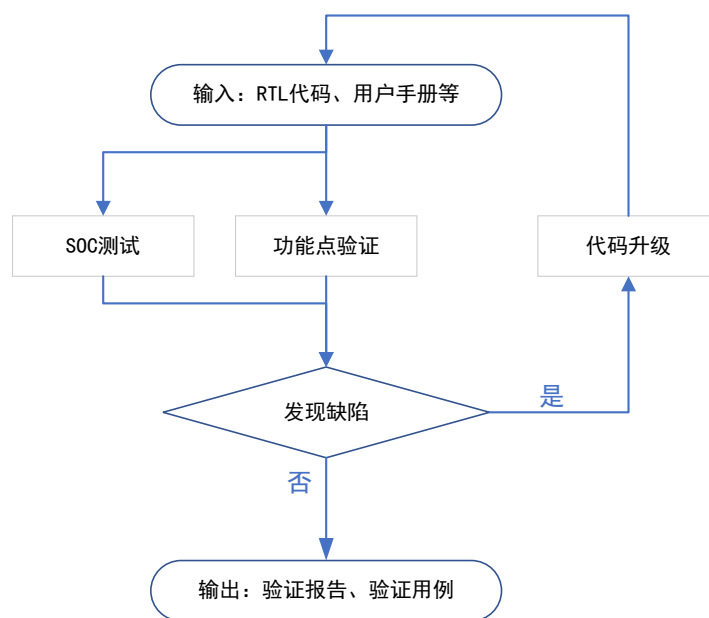


图 2: 升级工作流程图

## 4 升级策略

## 4.1 升级流程

本项目的输入包括任务书、IP 核数据手册等相关文档，以及存放于甲方服务器上的 RTL 代码。整体升级流程分为三步，如图2所示：

- (1) **功能点梳理与验证平台开发：**梳理 1553B IP 核的全部功能点，开发专用的验证平台，为后续测试和验证工作提供基础环境。
- (2) **测试用例开发与功能验证：**针对各功能点，尤其是与已知缺陷相关的部分，设计并实现测试用例，系统性验证功能正确性。采用故障注入技术，重点测试 IP 核的故障诊断与容错能力。
- (3) **缺陷处理与升级复测：**对发现的缺陷进行详细记录和分析，制定合理的升级方案，完成升级后进行复测，确保问题彻底解决。整个过程与甲方保持数据同步，确保信息一致。

## 4.2 验证

针对 1553B 总线的网络拓扑接口，本项目设计的验证平台如图所示。1553B 总线是一种主从式半双工通信总线，网络中包含 1 个总线控制器（BC）、最多 31 个远程终端（RT）和 1 个总线监控器（MT），并采用双冗余总线结构。验证平台说明如图3所示：

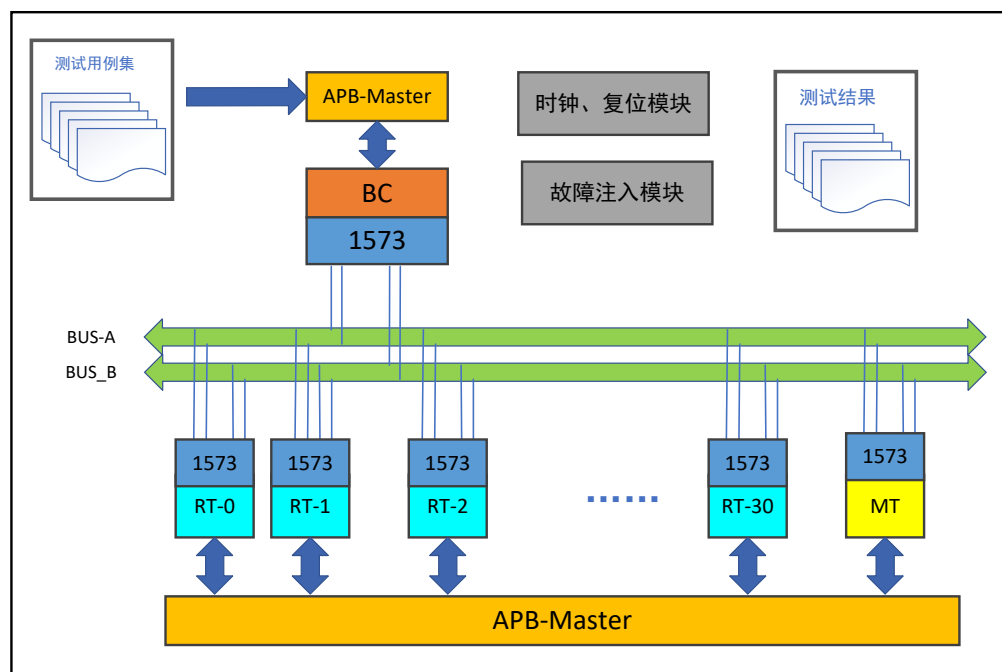


图 3: 验证平台结构示意图

- (1) BC、RT0-30、MT 均为 1553B IP 核实例，分别配置为不同的功能模式。其中 RT0 RT30 的编号对应 1553B 协议规定的 31 个 RT 地址。
- (2) APB MASTER 为 APB 总线主设备，直接连接各 IP 核，负责初始化、消息配置及结果读取。

- (3) 1553B 收发器用于实现总线信号的三态驱动和信号畸变等功能，模拟真实物理总线环境。
- (4) 测试用例集针对 1553B IP 核各功能点设计，所有消息均由 BC 发起，APB master 负责读取和配置，消息在总线上传输。
- (5) 各模式（BC、RT、MT）会将测试结果写入 log 文件，便于按 RT 地址检索和分析。

验证平台在 Linux 环境下搭建，采用 VCS201809 作为仿真工具。平台目录结构如下：DUT 位于 rtl 目录，仿真模型在 tb 目录，脚本存放于 vcs 目录，验证激励放在 testcase 目录。通过 vcs 目录下的 Makefile 运行仿真，最终生成波形文件 my.vpd 和 log 信息文件。

## 5 功能点及验证用例设计

依据 IP 核的实际工作场景，该 IP 核中有 1553B 编解码器的时钟，固定为 16MHz，APB 时钟，与 16Mhz 时钟为异步时钟，并且甲方测试发现，apb 时钟在不同频率下，该 IP 核表现出了不同的结果，因此本次测试用的 apb 时钟分别设置为 16MHz，100Mhz 及 500Mhz。

### 5.1 寄存器测试

1553B IP 核的寄存器列表及属性如表1所示，使用 APB MASTER 进行读写测试。

地址	寄存器名称	访问方式
0	中断屏蔽寄存器	RD/WR
1	配置寄存器 1	RD/WR
2	配置寄存器 2	RD/WR
3	启动/复位寄存器	WR
	命令堆栈指针寄存器	RD
4	BC 控制字/RT 子地址控制字寄存器	RD
5	时间戳寄存器	RD
6	中断状态寄存器	RD
7	配置寄存器 3	RD/WR
8	配置寄存器 4	RD/WR
9	配置寄存器 5	RD/WR
A	数据堆栈寄存器	RD
B	BC 帧时间保持寄存器	RD
C	BC 下一条消息时间保持寄存器	RD
D	RT 上一个命令字寄存器	RD
E	RT 状态字寄存器	RD
F	RT BIT 字寄存器	RD

表 1: 1553B IP 核寄存器列表及访问方式

## 5.2 RAM 测试

1553B IP 核有私有 4K\*16bit 的 RAM，进行读写测试。

## 5.3 1553B 收发器拖尾测试

故障注入，模拟 1553B 收发器拖尾的现象，验证引发的故障。

## 5.4 BC 功能点

序号	测试内容
1	针对 1553B 协议规定的 10 种消息类型，分别构造相应的消息，在 A、B 两条总线上进行发送测试，验证消息类型支持及通道切换功能。
2	测试堆栈区 A 的消息发送功能，覆盖 1 条消息至 64 条消息的不同场景，验证堆栈区 A 的消息缓存与发送能力。
3	测试堆栈区 B 的消息发送功能，覆盖 1 条消息至 64 条消息的不同场景，验证堆栈区 B 的消息缓存与发送能力。
4	验证 BC 模式下消息的发送与存储功能，确保消息能够正确下发并在 RAM 中正确存储。
5	验证 BC 模式下消息发送的 668us 超时功能，确保超时机制能够按协议要求正确响应。
6	配置不同的消息间隔时间，测试 BC 的消息间隔功能，验证消息发送间隔的可编程性和准确性。
7	BC 中断测试：分别测试 EOM（消息结束）、EOF（帧结束）、特定消息的中断等，验证中断响应的正确性。
8	消息重试功能测试：通过故障注入，发送消息后重试 1 次，更换总线，重试 1 次后成功，验证重试及总线切换机制。
9	消息重试功能测试：通过故障注入，发送消息后重试 1 次，更换总线，重试 1 次后不成功，验证失败处理机制。
10	消息重试功能测试：通过故障注入，发送消息后重试 1 次，不更换总线，重试 1 次后成功，验证同一总线重试机制。
11	消息重试功能测试：通过故障注入，发送消息后重试 1 次，更换总线，重试 1 次后不成功，验证多次失败处理。
12	消息重试功能测试：通过故障注入，发送消息后重试 1 次，更换总线，重试 2 次后成功，验证多次重试及切换机制。

表 2: BC 功能点测试项汇总



## 5.5 RT 功能点

序号	测试内容
1	RT 中断测试, EOM 中断
2	RT 中断测试, SA EOM 中断
3	RT 中断测试, 循环缓冲卷回中断, 设置缓冲区 256、512、1024、2048 大小
4	RT 响应时间测试
5	RT 响应超时测试-故障注入
6	RT 发送消息超时测试-故障注入
7	RT 接收缓冲区配置不同地址的测试
8	RT 发送缓冲区配置不同地址的测试
9	RT 广播缓冲区配置不同地址的测试
10	RT 堆栈 A, 消息索引功能, 测试
11	RT 堆栈 B, 消息索引功能, 测试
12	RT 堆栈 A, 卷回测试
13	RT 堆栈 B, 卷回测试
14	RT 消息存储功能
15	RT 消息循环缓冲模式-发送消息测试, 设置缓冲区 256、512、1024、2048 大小
16	RT 消息循环缓冲模式-接收消息测试, 设置缓冲区 256、512、1024、2048 大小
17	RT 消息双缓冲模式-发送消息测试
18	RT 消息双缓冲模式-接收消息测试
19	RT 消息非法化测试
20	RT 消息子地址忙测试

表 3: RT 功能点测试项汇总

## 5.6 MT 功能点

序号	测试内容
1	MT 的中断测试
2	MT 的消息存储测试
3	MT 的字监视功能测试
4	MT 的消息监视功能测试

表 4: MT 功能点测试项汇总

## 6 SOC 集成与验证

在验证过程中，将 IP 核集成到甲方的 SOC 中，并进行板级测试。具体流程包括：首先将经过仿真验证的 1553B IP 核集成到 SOC 设计中，完成 RTL 级的集成和接口适配。随后，进行 SOC 的综合、布局布线和流片，获得实际的芯片样品。将芯片焊接到开发板或测试板上，搭建完整的硬件测试环境。通过板级测试，验证 IP 核在 SOC 系统中的功能、性能和稳定性，包括与其他模块的协同工作、接口时序、数据传输正确性等。同时，结合自动化测试脚本，对各项功能点和异常场景进行全面测试，确保 IP 核在实际应用环境下的可靠性和兼容性。测试过程中发现的问题将及时反馈并修正，直至满足项目要求。

## 7 验证进度安排

序号	任务项	完成时间	交付物
1	故障诊断及升级	合同签订后 6 周内	设计报告、RTL 代码
2	仿真验证阶段	合同签订后 15 周内	测试报告、验证平台及测试用例
3	验收阶段	合同签订后 16 周内	研制总结报告

表 5: 研制进度安排

## 8 质量保证措施

针对本项目的验证工作，拟采取以下质量保证措施：

### (1) 基于覆盖率驱动的验证用例设计

采用覆盖率驱动的验证方法，能够量化验证进度，确保验证的全面性和科学性。本项目设定的覆盖率目标为：功能覆盖率达到 100%，代码测试覆盖率达到 90%（包括分支覆盖率、状态机覆盖率、行覆盖率），对于未覆盖的语句需进行说明。实现功能点 100% 覆盖的前提是功能点明确定义，第 5 章已汇总本设计的全部功能点。在验证实施过程中，将结合代码覆盖率、总线协议符合性等信息，与设计人员协作，持续完善和补充验证点列表。

### (2) 加强阶段性评审

通过阶段性评审，完善验证场景，补充功能点和验证用例。项目设置至少三个评审节点：1) 实施方案评审；2) RTL 升级后评审；3) 仿真验证结果评审。根据每次评审结果，动态补充和完善验证用例，确保验证工作的系统性和完整性。

## 9 风险分析和解决措施

本项目验证工作主要存在以下风险及应对措施：

### (1) 验证点遗漏风险

**风险描述：**功能点主要依据设计中各模块的主要功能特性进行定义，实际操作中可能存在部分验证点遗漏的情况。

**解决措施：**(1) 采用“功能点 + 代码覆盖率”相结合的方式，确保验证的全面性。如果功能点均已覆盖但代码覆盖率仍不高，需警惕功能点列举是否存在遗漏；(2) 通过阶段性评审或与设计人员协作，持续完善和补充验证点列表，降低遗漏风险。

### (2) 验证进度失控风险

**风险描述：**验证进度失控的主要原因包括：对验证结果缺乏信心，导致不断补充和修改验证用例，影响整体进度。

**解决措施：**与设计人员共同梳理和讨论典型工作场景，科学合理地构建测试用例，避免盲目开发。结合功能覆盖率和代码覆盖率等量化指标，客观评价验证用例的质量和有效性，确保验证工作有序推进。