

## 基于 JBits 的一种可重构数据处理系统可靠性研究

任小西<sup>1</sup> 李仁发<sup>1</sup> 金声震<sup>2</sup> 张克环<sup>1</sup> 吴强<sup>1</sup>

<sup>1</sup>(湖南大学计算机与通信学院 长沙 410082)

<sup>2</sup>(中国科学院国家天文台 北京 100012)

(happyrx@hnu.cn)

## Research on Reliability of a Reconfigurable Data Processing System Based on JBits

Ren Xiaoxi<sup>1</sup>, Li Renfa<sup>1</sup>, Jin Shengzhen<sup>2</sup>, Zhang Kehuan<sup>1</sup>, and Wu Qiang<sup>1</sup>

<sup>1</sup>(College of Computer and Communications, Hunan University, Changsha 410082)

<sup>2</sup>(National Astronomical Observatories, Chinese Academy of Sciences, Beijing 100012)

**Abstract** space solar telescope (SST) is a scientific satellite that employs FPGA (field programmable gate array) to preprocess the huge data gathered by its sensors. Since its high building and maintaining cost and poor working environment, it's a great challenge to ensure the reliability. An improved TMR (triple module redundancy) architecture is presented, in which the data arbiter can find the difference among its three inputs and send an error message to the main controller to launch fault scanning operation. A new method for the main controller is proposed to detect and remove hardware faults based on the configuration data of reconfigurable system. The test circuits and test stimulus are constructed by generating different patterns of configuration data. Since the reconfiguration process and the structure of configuration data are very complex, JBits is used to simplify the process of configuration data, which is originally written to facilitate reconfiguration system development. These measures are able to detect faults when they appear and remove faults by hardware reconfiguration, thus taking full advantage of TMR and reconfigurable features to improve reliability. And experiment results are given to show that minor routing resource fault can be repaired by using JBits and JRoutes. The availability of reconfigurable system with the proposed new architecture and fault processing method is modeled and analyzed using Markov process theory. Analysis results show that the reliability is improved greatly.

**Key words** reliability; TMR; JBits; fault-tolerant

**摘要** 空间太阳望远镜(SST)是一颗对太阳进行观测的科学卫星,它使用 FPGA 芯片对每天采集的大量数据进行预处理。高昂的建造费用和恶劣的工作环境,确保 SST 数据的高可靠性成为一项艰巨任务。改进了常规 TMR 结构,提出一种基于配置数据的可重构硬件故障检测和修复方法,使用 JBits 工具简化对配置数据的各种操作。此结构和方法能及时检测到故障,通过硬件重构消除故障,提高系统可靠性。采用 Markov 过程理论对系统可靠性进行分析,结果表明可靠性可得到显著提高。

**关键词** 可靠性;三模冗余;JBits;容错

中图法分类号 TP302.8

太阳作为距离我们最近的恒星,对它的研究具有重要理论价值和实用价值。这不仅能满足经济建设、科技发展和社会进步的广泛需求,更有助于进一

步增强综合国力。空间太阳望远镜(space solar telescope, SST)是一个太阳观测科学卫星。其主要作用是在外层空间直接采集数据并传回地球进行分

析和处理. SST 上带有多个观测装置,以便从多个不同角度同时对太阳进行观测,每天产生数据量约 1728GB. 由于 SST 和地球之间有效通信时间短、传输速率低,因此在卫星上使用 FPGA 芯片进行高速数据预处理和压缩,再传回地球.

空间环境恶劣、维护困难,因此保证系统高可靠性成为一大挑战. 目前提高可靠性的常用方法包括:特殊防护材料、系统冗余备份、多套系统同时运行、通过特殊表决机制获得最终结果等. 另外也有一些研究专门针对 FPGA 应用进行容错处理,提高系统可靠性. 这方面的研究主要分为基于硬件和基于软件两类. 第 1 类仍然沿着硬件冗余思路,在 FPGA 芯片内设置备用资源实现容错. 如果某处资源损坏,就用备用资源进行置换<sup>[1]</sup>. 这类方案的置换过程复杂、成本较高. 另一些研究则通过设计特殊硬件结构以简化资源置换过程. 文献[2]提出一种基于配置数据移动的方案. 通过对 SRAM 进行特殊设计,配置数据能在行间、列间以及 I/O 块之间移动. 出现故障时,失效资源的配置数据将根据特定规则被转移到邻近空闲资源上,使系统恢复正常. 然而这种方案比基于软件的实现方案复杂,且不适用于 COTS (commercial-off-the-shelf) FPGA 芯片. 第 2 类基于软硬件协同容错方法,首先对 FPGA 芯片进行检测,并将损坏情况存入数据库. 随后进行逻辑映射时,从数据库读取检测结果,保证损坏部分不被使用. 最后重新布局布线,使 FPGA 恢复正常运行<sup>[3]</sup>. 但这些方案在检测 FPGA 芯片时,要求系统停止工作,降低了系统性能和灵活性. 因此研究人员又提出一种在线检测和动态重构方案,能在系统正常工作时实现容错处理<sup>[4-5]</sup>. 例如, Xilinx 公司的主流 FPGA 芯片(如 Virtex II)支持部分和动态重构,即重构 FPGA 上部分资源而其他部分仍正常工作<sup>[6]</sup>. 文献[4]对此特点进行了探讨,它将 FPGA 资源动态划分为两个部分,一部分实现用户逻辑,另一部分则执行硬件检测操作. 如果被检测部分没发现错误,则系统执行一次新的划分,将刚通过检测的区

域合并到用户逻辑区,同时从用户逻辑区划分出一块新的区域作为待检测区. 这种方案功能很强,但动态重构过程十分复杂,需要非常了解 FPGA 的配置细节. Xilinx 提供的 JBits 工具集可简化这个开发过程<sup>[7]</sup>. JBits 核心是一组用 Java 语言实现的 API 函数. 调用这些函数,用户可对 FPGA 的配置比特流文件进行各种操作,从而不必掌握芯片上复杂的配置机制,仅通过编写 Java 程序就可访问和控制 FPGA 芯片上每一种资源,实现期望的电路结构和功能. JBits 工具还包含 XHWIF, JRoute 等部分. XHWIF (Xilinx hardware interface)负责完成与实际硬件之间的通信. 它分为硬件相关层和硬件无关层,前者运行在实际 FPGA 硬件电路板上,实现最基本的 FPGA 访问操作,后者通过调用前者来完成更复杂操作. JRoute 用于实现动态布线功能.

SST 预期寿命为 3 年,总可靠性要求为 0.90. 为了实现该设计目标,本文采取三模冗余结构与可修复硬件重构技术相结合的方法,避免常规三模冗余结构只容纳一个故障的不足,改进 TMR 数据仲裁器,能实时监测系统处理中的错误,及时启动故障检测和修复程序,避免故障积累;提出一种基于配置数据的可重构硬件故障检测方法,利用可重构硬件自身搭建测试电路,产生测试激励和保存测试结果;利用硬件可重构性对系统重新布局布线,避开故障资源,使系统恢复正常运行. 同时用 JBits 实现对配置数据的各种操作.

## 1 可重构数据处理系统结构及容错处理方法

SST 数据处理系统结构如图 1 所示. 该系统由数据采集、数据处理、数据存储、数据发送和主控制器 5 个部分组成. 首先,数据由传感器采集送入 FPGA 进行处理. 处理后的数据放入存储器. 当 SST 与地面控制中心通信时,从存储器中读取数据,传送给地面接收站. 为了保证数据处理的可靠性,

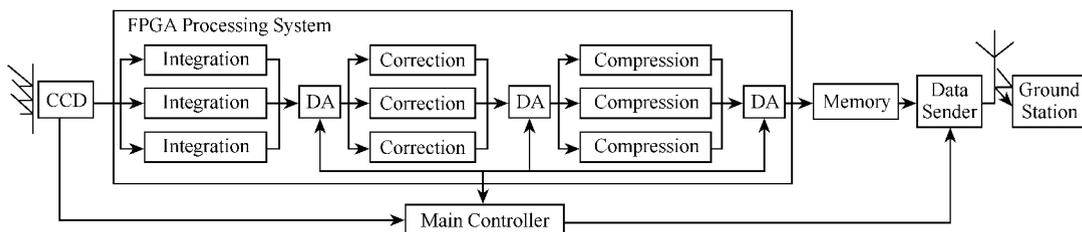


Fig. 1 System architecture with TMR and JBits.

图 1 基于 TMR 和 JBits 的 SST 数据处理系统结构

每个处理阶段都采用 TMR<sup>[8]</sup>方法实现,即每个阶段包含 3 个共享输入数据的子模块同时进行处理,三者处理结果送到一个数据仲裁器(DA)中,采用基于多数表决算法产生一个该处理阶段的最终结果.仅采用 TMR 结构并不能保证可重构系统的可靠性,这是由可重构系统硬件特性决定.一般的可重构系统硬件都采用 FPGA 芯片实现各种功能,FPGA 的功能是由其内部 SRAM 存储的配置数据决定<sup>[9]</sup>.由于 SRAM 中数据容易受外层空间环境影响,发生翻转,且翻转后的数据会被保存,直到被纠正或再次发生翻转,因此 FPGA 芯片的功能容易受环境影响而改变,使系统出现故障,且故障会累积.一般 TMR 结构只能容纳一个故障.当故障数超过一个时,数据仲裁器一般无法输出正确结果.因此对于可重构系统而言,须在 TMR 结构基础上添加额外方法和措施,尽早检测和消除故障.具体方法是:首先,及时检测系统处理过程中出现的错误.尽管一些错误会被 TMR 结构过滤而不影响系统正常运转,也将其检测出来、予以处理,避免故障积累.然后检测和定位故障.故障是硬件系统出现的异常现象,导致系统出错的根源.要消除错误,必须找到系统故障的位置.最后,根据前面故障检测和定位结果,采取措施排除故障,使系统恢复正常.

为了尽早发现系统处理过程中的错误和异常,本文改进了 TMR 数据仲裁器.改进后的仲裁器除输出表决结果之外,还检查 3 个输入数据是否一致.如三者存在差异,数据仲裁器将通知主控模块进行故障检测和定位.针对可重构系统的特点,本文设计了一种基于配置数据的硬件故障检测和定位方法.与一般系统故障检测方法不同,该方法采取迭代和逐渐扩展的自测试过程,首先对资源的可重构功能进行测试,然后利用一部分可重构资源构造测试

电路,检测另一部分.这个过程将逐渐扩展,直到整个可重构硬件资源都被检查一遍.整个检测任务是通过配置数据操作实现,无需利用外部特殊测试电路.图 2 给出了系统故障检测定位和修复流程.

SST 数据处理系统可能出现的故障主要有两类:一种是单粒子翻转事件导致的配置数据改变,另一种是 FPGA 芯片内部硬件资源损坏.当数据仲裁器发现输入数据不一致后,首先进行第 1 类故障检查,即配置数据是否发生改变.检查方法是:从 FPGA 中读出配置数据,与存储在 FLASH 中的标准配置数据比较.如果两者不一致,可判定出现了第 1 类故障.如果数据仲裁器在短时间内总是报错,且检测和修复了第 1 类故障之后仍报错,则需进行第 2 类故障检查,即硬件资源损坏.FPGA 硬件资源主要有:配置数据存储单元(SRAM)、查找表(LUT)、块存储器(BlockRAM)和布线资源.不同硬件资源有不同的故障检测和定位方法.存储配置数据 SRAM 资源的检测方法是:反复执行配置数据写入和读出操作,比较读出数据是否与写入数据一致(多次读写是为了排除偶然单粒子事件影响).如果总是不一致,说明 SRAM 出现了损坏.LUT 的检测方法是:首先向 FPGA 写入配置数据,将待检测 LUT 设置为特定逻辑功能(例如一个全加器、移位寄存器等),同时将它周围 LUT 和寄存器资源配置为一个能施加测试激励和表达测试结果的测试电路.随后,利用 FPGA 的 CAPTURE 功能,将测试结果保存到配置数据中.最后读回配置数据,从中提取测试结果进行分析,判断被测资源是否正常.BlockRAM 资源和布线资源的故障检测方法 with LUT 基本一致,只是测试电路的功能和结构不同.测试 BlockRAM 时,需执行 BlockRAM 的读写操作,而检测布线资源时,需更改配置数据,扫描与一个交换节点相连的所有网线.测试结果通过回读和分析配置数据得到.

基于上述检测方法,我们使用 JBits 构造测试电路,向 FPGA 写入和读出配置数据以及从配置数据中提取和分析测试结果.JBits 对配置过程和数据的细节进行了封装,用户只需调用相应方法就可完成各种复杂操作.由于配置数据与硬件资源之间存在一一对应关系,因此通过分析配置数据可知故障资源类型和物理位置.这些信息会被记录,为故障修复提供重要参考.故障修复也是通过操作配置数据实现.为了修复配置数据翻转导致的故障,需将正确的配置数据重写入 FPGA 芯片中.硬件故障的修复需根据损坏程度采取不同策略.当损坏较少(如少数 LUT 或部分布线资源出现故障)时,可在附近

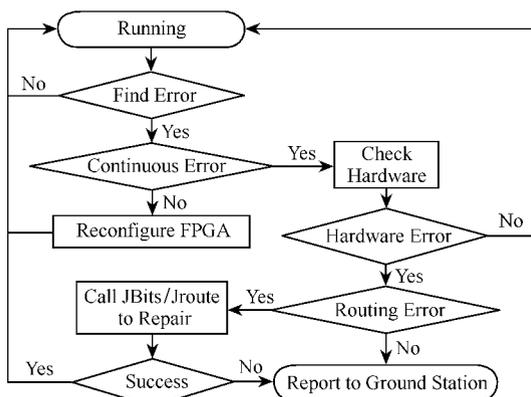


Fig. 2 System fault processing flow.

图 2 故障检测和恢复处理流程

寻找空闲资源予以替换。JBits 的 JRoute 程序能在指定的两个硬件资源间自动寻找一条新的布线路径,从而替换已经损坏的布线资源。当损坏严重时,需将故障检测和定位报告发送给地面控制中心,然后由控制中心人员根据故障类型和位置生成一个新的布局布线约束文件(UCF),禁止用户逻辑使用已损坏的硬件资源。最后根据新的约束文件重新布局布线,使系统绕过受损的硬件资源,恢复正常运行。

考虑 FPGA 芯片中布线资源面积一般超过 80% 以上,本文针对少量布线资源损坏情况,利用 JBits 工具进行布线故障修复实验。在 Virtex 系列

FPGA 芯片中,布线资源由信号线(wire)和交换节点(switch box)组成。各种交换节点和信号线之间的关系如图 3 所示。信号线分为短线(single line)和长线(hex line),短线连接两个直接相邻的可编程逻辑块(CLB),长线连接相隔 6 个单元的 CLB。交换节点包括:主交换节点(main switch box),短交换节点(single switch box)和长交换节点(hex switch box)。主交换节点直接与 CLB 的内部逻辑资源相连,起着外部信号线和内部逻辑资源的桥梁作用。短交换节点和长交换节点分别负责短信号线和长信号线的连接。

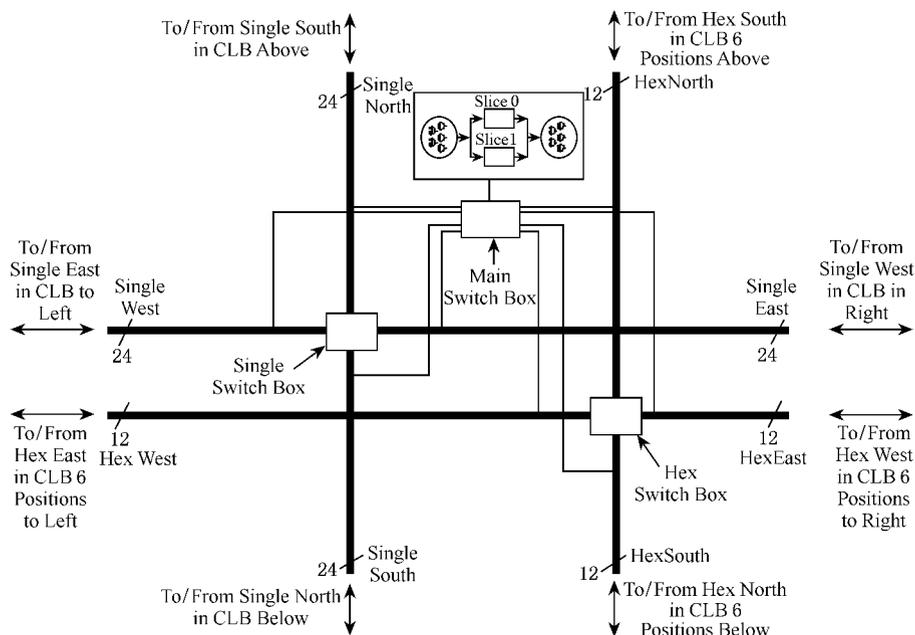
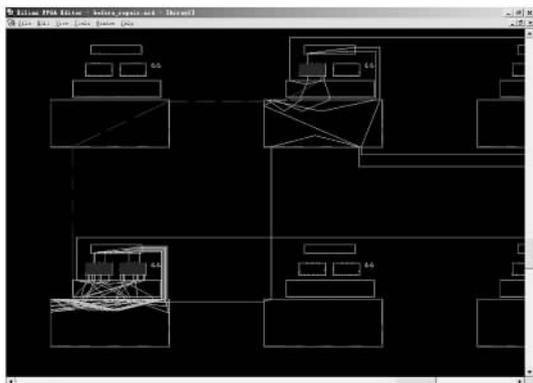


Fig. 3 Interconnection of switch boxes and routing wires.

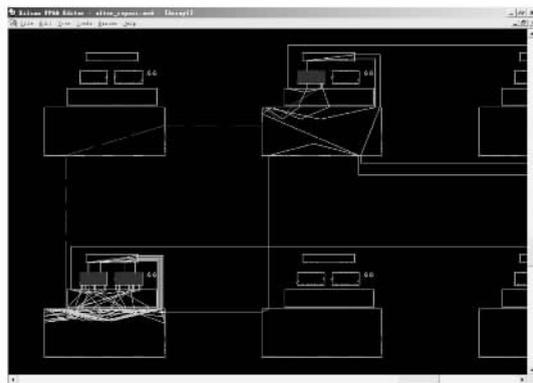
图 3 交换节点和信号线之间的连接图

图 4(a)(b)显示了 JBits/JRoute 修复布线故障的 FPGA 内部布线变化。图 4(a)中 4 个最大的矩形

是交换节点 4 对并排分布的小矩形(空心的和被填充的)表示 Slice。紧接 Slice 的上下两个长条形分



(a)



(b)

Fig. 4 Routings before and after repair. (a) Routing before repair and (b) Routing after repair.

图 4 JBits/JRoute 修复布线故障前后的布线图。(a) 修复前布线图 (b) 修复后布线图

别表示可编程逻辑块的输出和输入多路选择器. 其中图4(a)中虚线表示修复前一条出现故障的布线, 该线经过了图中上面两个交换节点. 为了修复该故障, 需在这两个交换节点之间找到新的路径. 修复时, 首先创建两个 PIN 对象(PIN 是 JBits 内部用来表示 FPGA 芯片硬件资源的一种数据结构), 分别对应故障布线资源的起点和终点. 然后利用 JBits/JRoute 中 ResourceFactory 对象将故障资源的属性设置为“已使用”, 随后调用 JRoute 在这两个节点之间找到了一条新布线资源来替换已经损坏的部分. 比较图4(b)与图4(a)中的虚线可以发现: 修复前后只是在出现故障的部分进行布线替换调整, 而其他无故障信号线(实线)仍保持了原来的布线结果.

## 2 基于 Markov 模型的可靠性分析

为了评估上节故障检测和修复方法的有效性, 我们分析和比较 3 种不同体系结构的系统可靠性. 这 3 种系统分别是: 单模系统、三模块冗余系统以及具有修复功能的三模块冗余系统. 系统中包含 3 个不同的处理操作. 由于系统中所有元器件都经过严格筛选, 失效率是常数, 因此每个处理模块(或者构成三模冗余的每个子模块)的可靠性都遵循指数变化规律, 即

$$r(t) = e^{-\lambda t}, \quad (1)$$

其中  $\lambda$  代表通用失效率<sup>[10]</sup>. 对于单模系统而言, 任何模块的失效都会导致整个系统失效, 因此系统的可靠性是 3 个处理阶段模块可靠性的乘积, 即

$$R_s = e^{-3\lambda t}. \quad (2)$$

在三模块冗余系统中, 系统正常工作仍然要求所有 3 个处理操作都正常完成, 但是由于每个处理阶段包括 3 个子模块, 因此每个处理阶段的可靠性不再是  $r$ . 每个处理阶段都使用 TMR 机制实现, 因此只要至少两个子模块正常就能保证对应的操作能正常完成, 相应的可靠性为:

$$R = 3e^{-2\lambda t} - 2e^{-3\lambda t}. \quad (3)$$

因此整个系统的可靠性为:

$$R_{\text{sys}} = (3e^{-2\lambda t} - 2e^{-3\lambda t})^3. \quad (4)$$

对于本文提出的 TMR 改进结构和故障处理方法, 整个系统的可靠性仍为 3 个处理阶段可靠性的乘积, 但由于系统可修复, 每个处理操作的可靠性计算变得相对复杂. 下面将使用 Markov 过程理论对

系统中模块状态的变化进行建模, 并在此基础上分析整体的可靠性. 状态迁移如图 5 所示:

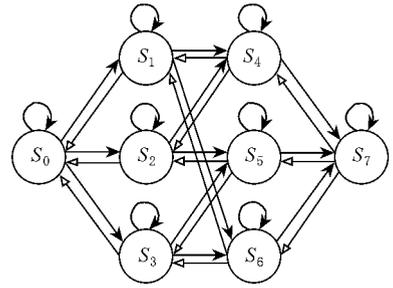


Fig. 5 The Markov state transition graph.

图 5 系统状态转移图

系统中每个处理阶段由 3 个子模块构成, 因而一共存在 8 种可能的状态, 分别记为  $S_0, S_1, S_2, S_3, S_4, S_5, S_6, S_7$ . 其中,  $S_0$  表示 3 个子模块都正常工作,  $S_1, S_2$  和  $S_3$  表示有一个模块失效,  $S_4, S_5$  和  $S_6$  表示有两个模块不能正常工作,  $S_7$  代表所有 3 个模块全失效. 图 5 中一共有 3 种状态迁移类型, 每种类型具有不同含义和发生概率. 第 1 种类型是从某个状态迁移到标签值增大的状态, 例如从  $S_0$  迁移到  $S_1$ . 它代表系统中有一个子模块出现故障. 由于每个子模块的工作可靠性服从指数分布, 因此一个子模块在  $t + \Delta t$  时刻失效的概率为:

$$P(t + \Delta t) = 1 - e^{-\lambda \Delta t}, \quad (5)$$

将它展开为泰勒级数可以得到:

$$P(t + \Delta t) = \lambda \Delta t - O(\lambda \Delta t), \quad (6)$$

其中,  $O(\lambda \Delta t)$  为  $\Delta t$  的高阶无穷小量, 当  $\Delta t \rightarrow 0$  时可以忽略, 于是式(6)简化为:

$$P(t + \Delta t) = \lambda \Delta t. \quad (7)$$

第 2 种迁移类型是从某状态迁移到标签值减小的状态, 例如从  $S_1$  迁移到  $S_0$ . 这代表系统中一个子模块被修复而重新开始工作. 为了分析这种迁移的发生概率, 这里定义两个变量:  $k$  和  $\mu$ .  $k$  是 FPGA 修复成功对系统可靠性的影响因子(表示 FPGA 在系统中所占比例);  $\mu$  为通用修复率, 用于表达成功修复 FPGA 芯片的概率, 定义为  $\mu = 1/MTTR$ , 其中  $MTTR$  为平均修复时间(mean time to repair). 由于故障类型基本相同, 出现故障后具有相同修复算法, 因此  $MTTR$  为常数, 从而  $\mu$  是常数, 因此修复成功率也服从指数规律, 即

$$r(t) = k \times (1 - e^{-\mu t}). \quad (8)$$

与第 1 类转移发生概率的过程类似, 可以得到第 2 类迁移类型发生的概率为:

$$P(t + \Delta t) = k\mu \Delta t. \quad (9)$$

第 3 种迁移类型是自身迁移的一个过程,例如从  $S_0$  到  $S_0$ , 表示维持自身状态不变. 根据概率互补定律, 1 减去前面两种类型的发生概率即为这种类型的发生概率, 例如  $S_0$  的第 3 种迁移概率就为:

$$P(t + \Delta t) = 1 - 3\lambda\Delta t. \quad (10)$$

根据图 5 和上述迁移类型及其发生概率, 可得如下的离散差分方程组:

$$\begin{aligned} p_0(t + \Delta t) &= (1 - 3\lambda\Delta t)p_0(t) + \\ & k\mu\Delta t(p_1(t) + p_2(t) + p_3(t)), \\ p_1(t + \Delta t) &= (1 - 2\lambda\Delta t - k\mu\Delta t)p_1(t) + \\ & k\mu\Delta t(p_4(t) + p_6(t)) + \lambda\Delta t p_0(t), \\ p_2(t + \Delta t) &= (1 - 2\lambda\Delta t - k\mu\Delta t)p_2(t) + \\ & k\mu\Delta t(p_4(t) + p_5(t)) + \lambda\Delta t p_0(t), \\ p_3(t + \Delta t) &= (1 - 2\lambda\Delta t - k\mu\Delta t)p_3(t) + \\ & k\mu\Delta t(p_5(t) + p_6(t)) + \lambda\Delta t p_0(t), \\ p_4(t + \Delta t) &= (1 - \lambda\Delta t - 2k\mu\Delta t)p_4(t) + \\ & \lambda\Delta t(p_1(t) + p_2(t)) + k\mu\Delta t p_7(t), \\ p_5(t + \Delta t) &= (1 - \lambda\Delta t - 2k\mu\Delta t)p_5(t) + \\ & \lambda\Delta t(p_2(t) + p_3(t)) + k\mu\Delta t p_7(t), \\ p_6(t + \Delta t) &= (1 - \lambda\Delta t - 2k\mu\Delta t)p_6(t) + \\ & \lambda\Delta t(p_3(t) + p_1(t)) + k\mu\Delta t p_7(t), \\ p_7(t + \Delta t) &= (1 - 3\lambda\Delta t)p_7(t) + \\ & \lambda\Delta t(p_4(t) + p_5(t) + p_6(t)). \end{aligned} \quad (11)$$

当  $\Delta t \rightarrow 0$ , 则可得连续微分方程组:

$$\begin{aligned} \frac{d}{dt}p_0(t) &= -3\lambda p_0(t) + k\mu(p_1(t) + \\ & p_2(t) + p_3(t)), \\ \frac{d}{dt}p_1(t) &= (-2\lambda - k\mu)p_1(t) + k\mu(p_4(t) + \\ & p_6(t)) + \lambda p_0(t), \\ \frac{d}{dt}p_2(t) &= (-2\lambda - k\mu)p_2(t) + k\mu(p_4(t) + \\ & p_5(t)) + \lambda p_0(t), \\ \frac{d}{dt}p_3(t) &= (-2\lambda - k\mu)p_3(t) + k\mu(p_5(t) + \\ & p_6(t)) + \lambda p_0(t), \\ \frac{d}{dt}p_4(t) &= (-\lambda - 2k\mu)p_4(t) + \lambda(p_1(t) + \\ & p_2(t)) + k\mu p_7(t), \\ \frac{d}{dt}p_5(t) &= (-\lambda - 2k\mu)p_5(t) + \lambda(p_2(t) + \\ & p_3(t)) + k\mu p_7(t), \end{aligned}$$

$$\frac{d}{dt}p_6(t) = (-\lambda - 2k\mu)p_6(t) + \lambda(p_3(t) + p_1(t)) + k\mu p_7(t),$$

$$\frac{d}{dt}p_7(t) = -3\lambda p_7(t) + \lambda(p_4(t) + p_5(t) + p_6(t)). \quad (12)$$

求解方程组, 可得到系统在某个特定状态的概率. 由于在状态  $S_0, S_1, S_2, S_3$  下系统才能正常工作 (即至少有两个模块正常工作), 因此系统在这 4 个状态上的概率和就是系统可靠工作的概率, 即可靠性. 经过求解可得最终表达形式为:

$$\begin{aligned} R &= (3\lambda^3 e^{-\lambda(k\mu+\lambda)t} - 2\lambda^3 e^{-\lambda(k\mu+\lambda)t} + \\ & 6\lambda^2 e^{-(k\mu+\lambda)t} k\mu - 3e^{-\lambda(k\mu+\lambda)t} \lambda^2 k\mu + 3\lambda k^2 \mu^2 + \\ & \mu^3 k^3)(\lambda^3 + 3\lambda k^2 \mu^2 + \mu^3 k^3 + 3\lambda^2 k\mu). \end{aligned} \quad (13)$$

根据以上公式, 可算出不同修复率下, 可修复系统和普通 TMR 系统在 3 年寿命期内的可靠性, 如表 1 所示 (设  $t = 30000\text{h}$ ).

Table 1 Reliability Comparison under Different Parameter Values

表 1 不同参数条件下可靠性比较

$\lambda$	$k \times \mu$	$R_{\text{JBits}}$	$R_{\text{TMR}}$
	0	0.833	
	$10^{-5}$	0.870	
$10^{-5}$	$5 \times 10^{-5}$	0.947	0.833
	$10^{-4}$	0.978	
	$3 \times 10^{-4}$	0.996	

当  $k \times \mu = 0$  时, 表示系统不具备修复能力, 此时两类系统可靠性相同. 当  $k \times \mu > 0$  时, 普通 TMR 系统可靠性  $R_{\text{TMR}}$  维持在 0.833 不变, 而可修复的可重构系统可靠性  $R_{\text{JBits}}$  随  $k \times \mu$  增加得到明显提高.

### 3 小 结

为了保证 SST 数据处理系统的可靠性, 本文提出了一种基于配置数据的故障检测和修复方法, 并使用 JBits 工具完成对配置数据的操作. 该方法可检测 FPGA 芯片出现的各种故障, 甚至能在芯片某些部分遭受损坏情况下检测到损坏区域, 通过重布局布线避开此区域使系统恢复正常. 采用 Markov 模型分析了系统可靠性, 结果显示该方法能有效提高系统的可靠性. 目前我们已经完成了设计, 并对其可行性进行了初步验证. 下一步将在 SST 工程样机开发环境下做进一步验证.

## 参 考 文 献

- [1] G H Chapman, B Dufort. Using laser defect avoidance to build large-area FPGAs [J]. IEEE Test & Design on Computers, 1998, 15(4): 75-81
- [2] A Doumar, S Kaneko, H Ito. Defect and fault tolerance FPGAs by shifting the conguration data [C]. Int'l Symp on Defect and Fault Tolerance in VLSI Systems, Albuquerque, NM, 1999
- [3] F Hanchek, S Dutt. Methodologies for tolerating cell and interconnect faults in FPGAs [J]. IEEE Trans on Computers, 1998, 47(1): 15-33
- [4] M Abramovici, C Stroud, S Wijesuriya, et al. Using roving STARs for on-line testing and diagnosis of FPGAs in fault-tolerant applications [J]. IEEE Int'l Test Conf, Atlantic City, NJ, 1999
- [5] J Emmert, C Stroud, B Skaggs, et al. Dynamic fault tolerance in FPGAs via partial reconfiguration [C]. IEEE Symp on Field-Programmable Custom Computing Machines (FCCM 2000), Napa Valley, CA, 2000
- [6] Xilinx Inc. Virtex-II Platform FPGA User Guide [OL]. <http://direct.xilinx.com/bvdocs/userguides/ug002.pdf>, 2006-11-19
- [7] Xilinx Inc. JBits Software and Documentation [OL]. <http://www.xilinx.com/labs/downloads/jbits/jbits3.jar>, 2006-11-19
- [8] Xilinx Inc. XAPP197: Triple Module Redundancy Design Techniques [OL]. <http://www.xilinx.com/bvdocs/appnotes/xapp197.pdf>, 2006-11-19
- [9] Li Renfa, Zhou zude, et al. Hardware for reconfigurable computing [J]. Journal of Computer Research and Development, 2003, 40(3): 500-506 (in Chinese)  
(李仁发, 周祖德, 等. 可重构计算的硬件结构 [J]. 计算机研究与发展, 2003, 40(3): 500-506)
- [10] Gu Ying. Reliability Engineering Mathematics [M]. Beijing: Publishing House of Electronic Industry, 2004 (in Chinese)  
(顾瑛. 可靠性工程数学 [M]. 北京: 电子工业出版社, 2004)



**Ren Xiaoxi**, born in 1978. Ph. D. candidate in computer engineering. Her current research interests are reconfigurable computing and design methodology of embedded systems.

任小西, 1978年生, 博士研究生, 主要研究方向为可重构计算、嵌入式系统设计方法学。



**Li Renfa**, born in 1957. Professor and Ph. D. supervisor of Hunan University. His current research interests include reconfigurable computing, embedded computing systems, distributed computing and networks.

李仁发, 1957年生, 教授, 博士生导师, 主要研究方向为可重构计算、嵌入式计算系统、分布式计算与网络。



**Jin Shengzhen**, born in 1944. Research professor and Ph. D. supervisor of the National Astronomical Observatories, Chinese Academy of Sciences. His main research interests include astronomical instrument and technology in space.

金声震, 1944年生, 研究员, 博士生导师, 主要研究方向为空间天文技术方法。



**Zhang Kehuan**, born in 1978. Master. His main research interests include system-on-chip and its architecture.

张克环, 1978年生, 硕士, 主要研究方向为SoC及其体系结构。



**Wu Qiang**, born in 1974. received his Ph. D. degree in computer engineering from Tsinghua University in 2004. His main research interests include reconfigurable computing and electronic design automation.

吴强, 1974年生, 博士, 主要研究方向为可重构计算、电子设计自动化。

## Research Background

This paper is supported by the National High-Tech Research and Development Plan of China (863-2.5.1.25). The aim of space solar telescope (SST) project is to send a solar telescope into earth orbit and observe the sun in wide spectrum continuously. The project covers several different research fields, including both theoretic and engineering issues. One of the key challenges is the need to process massive data quickly and reliably. This paper proposes a solution to the challenge by adopting the reconfigurable computing technology. The reconfigurable computing technology is based on the FPGA (field programmable gate array), which can reorganize its hardware resources to implement specific function or use unoccupied resources to make backups of the working parts. The former feature is used to achieve a performance comparable to ASIC (application specific integrated circuit), which is capable of processing massive data quickly. The latter feature is leveraged to enable the on-board computer system of SST to resume its normal operation by re-implementing its function on remaining good resources when current working parts are damaged by cosmic rays or particles. Applying reconfiguration and FPGA technology in space is the trend for future space on-board computer technology.