一种面向多核的可重构容错方法

张绍林[1] 杨孟飞[1,2] 刘鸿瑾[1] 姜 宏[1] 王若川[1]
(北京控制工程研究所 北京 100190)[1] (中国空间技术研究院 北京 100094)[2]

摘 要 随着二代导航、载人航天、深空探测等空间应用对星载电子产品的低功耗和抗辐射容错能力提出更高的需求,传统多机冗余设计星载计算机面临着亟需进行设计升级换代。将可重构技术应用到多核片上系统的设计中,提出了一种基于动态可重构的容错体系结构,在硬件层提高系统的容错能力和扩展性对未来空间工程应用具有重要意义。首先介绍了多核片上系统和可重构技术的基本概念,简要分析了国际宇航可重构系统的研究案例。随后提出了一种基于动态可重构的容错体系结构,即通过基于系统降级的重构策略来实现系统级容错。在方案验证环节,采用LEON3作为处理单元,对容错模块功能进行了仿真验证。仿真结果表明,容错控制满足预期的设计需求。最后对后续工作做了简要规划,并对可重构容错方法设计进行了总结。

关键词 多核,片上系统,动态可重构,容错
中图法分类号 TP302.8 **文献标识码** A

Reconfigurable Tolerance Method for Multi-processor System

ZHANG Shao-lin[1] YANG Meng-fei[1,2] LIU Hong-jin[1] JIANG Hong[1] WANG Ruo-chuan[1]
(Beijing Institute of Control Engineering, Beijing 100190, China)[1]
(China Academy of Space Technology, Beijing 100094, China)[2]

Abstract Because of advancing space applications, such as Second Generation of Navigation System, Manned Spaceflight, Deep Space Missions, requirement on On-Board Computer(OBC) for higher performance has been arisen and normal OBCs are facing an upgrade of consume and rad-hard capability. Based on adaptive reconfiguration technology and System-on-Chip design method, this paper presented a reconfigurable tolerance method for MPSoC. High reliability and expansibility can be achieved at hardware level, which makes a great sense of space applications and military missions in future. After introducing the basic concepts of MPSoC and reconfigurable technique, several representative cases were presented as to show the development of reconfigurable processors. Then the basic infrastructure of our dynamic reconfigurable tolerance system was discussed. Moreover, tolerance strategy based on system degrading was presented in detail. A contrast analysis among single core, normal design and our method was provided. Besides, using leon3 to implement the process elements, simulation validation of tolerance module efficiency was given out and the test results show the control mechanism works well. Finally, the future related works were discussed and the proposed reconfiguration tolerance method was concluded.

Keywords Multi-processor, System on chip, Dynamic reconfiguration, Tolerance

1 引言

多核片上系统(MPSoC, Multi-Processor System on Chip)[1]作为ASIC设计方法学中的一项新技术,极大地提高了芯片的集成度、处理性能和开发效率[2],已越来越多地被应用到航空航天、工业控制等各行各业应用中。但是,多核片上系统因集成度较高,在空间辐射复杂环境中易受到高能粒子的撞击而发生故障,因此解决多核片上系统的空间辐射效应问题成为其在空间应用中所面临的亟需解决的问题。

近年来,随着可编程逻辑器件的大量应用,片上系统可重构技术逐渐成为一个热点[3]。尤其是基于SRAM型FPGA在线部分重配置技术的出现,更为片上系统可重构技术提供了新的思路。在线部分可重构允许用户可以在不中断其他系统功能模块正常工作的情况下,对可编程逻辑器件特定的区域进行实时重配置,实现容错纠错或功能扩展[4]。

早在20世纪60年代,美国加利福尼亚大学的Geraid Estrin就提出了可重构的概念,并提出了可重构计算机原型系统。几十年来,国际上对可重构技术的研究取得了丰富的

到稿日期:2013-09-17 返修日期:2013-11-21 本文受国家十二五民用航天项目:基于SoC的航天器可重构控制系统一体化技术资助。
张绍林(1988—),男,硕士生,主要研究方向为SoC设计与验证、空间星载计算机容错技术等,E-mail:gtr496@126.com;**杨孟飞**(1962—),男,博士,研究员,博士生导师,主要研究方向为空间控制计算机系统和卫星控制系统;**刘鸿瑾**(1980—),男,博士,高级工程师,主要研究方向为空间应用SoC设计与验证、星载计算机容错技术等;**姜 宏**(1975—),男,博士生,主要研究方向为软硬件协同设计;**王若川**(1987—),男,硕士生,主要研究方向为可信计算。

成果。如德国慕尼黑工业大学 S. Ivars 等人基于动态冗余设计的理念提出了一种集成 3 个处理器核的可重构体系结构-ERViS[5]，在保证系统低耦合和高可靠的同时，实现了多种工作模式重构；NASA 戈达德航天中心(GSFC)研究基于动态重配置的可重构多核体系结构-SpaceCube[6-8]，采用动态刷新和在线部分重配置技术实现了四核处理器系统的容错加固，处理性能达到了 5000 MIPS，并已经成功应用在国际空间站的图像处理应用中；德国卡尔斯鲁厄大学 Lars Bauer 等人提出了一种用于处理器硬件加速的可重构设计方法[9]，采用一种称为专用指令(SI)的基本结构实现了层次化的可重构原型系统，实现了对用户透明的自动化可重构机制；日本 UEC 大学 Tomohiro Harada 等人提出了一种采用 Tierra 进化容错原理设计的进化容错方法[10]，亦即将生物进化机制应用到星载计算机硬件的容错机制中。

国内领域对可重构片上系统的研究起步较晚，多还处于理论研究和验证阶段。如航天 502 所的硬件进化重构容错研究，提出一种适用于空间应用领域进化容错的 FPGA 故障模型和故障检测方法[11]，并验证了进化重构容错的可行性。另外，文献[12]提出了一种基于二阶近似域划分的可重构容错片上系统，并验证了在商用 FPGA 上构建高可靠、低延时的复杂系统的可行性。

总之，可重构多核片上系统具有较强的容错能力和灵活性，既可以满足用户对高性能处理的需求，又可满足对系统的扩展性和适应性的要求。基于可重构设计思路，本文提出了一种面向多核处理器的可重构容错体系结构，即采用基于系统降级的容错策略来实现具备高可靠性的多核片上系统。

2 可重构容错系统体系结构

针对多处理器片上系统在空间环境中的可靠性问题，本文提出了一种针对多处理器的可重构体系结构，如图 1 所示。该体系结构采用基于冗余的动态可重构设计方案，主要包括处理单元、容错控制模块、I/O 仲裁模块、外部存储单元和接口等模块。其中对处理器以及容错控制单元、仲裁器、I/O 接口和 I/O 控制单元均进行冗余加固设计。

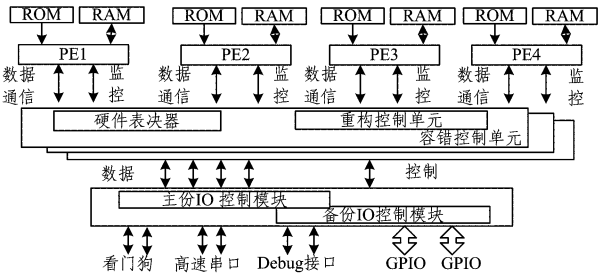

图 1 多处理器可重构体系结构图

该可重构体系结构有以下主要特点：

1)硬件动态冗余设计。该设计对容错控制和仲裁模块等关键模块进行了动态冗余设计，保证动态可重构的控制可靠性，使其能够适用于军事、宇航级工程应用。

2)基于系统降级的硬件级容错策略。本系统初始工作在拜占庭容错模式下，若某一时刻某处理器发生故障，则系统在控制单元的作用下降级为 TMR 容错工作模式，对故障进行修复(复位或重配置)后，重新回到拜占庭容错模式下继续工作。

3)基于时钟同步的硬件表决设计。多处理器采用硬件表决器来实现处理器数据输出的数据表决，并与处理单元共享系统全局时钟，从而实现了整个系统的时钟级同步，相比广泛被采用的软件比对方案，该策略大大提高了系统的处理效率。

4)基于部分可重构的故障恢复机制。本设计基于在线部分重配置技术，设计并验证了针对 PE 模块的故障恢复方法，提高了系统的灵活性和可靠性。

5)多种可扩展工作模式设计。本多处理器系统可以通过对控制器的控制，使得系统工作在多种模式下：QMR 容错模式、TMR+Sleeping 模式、多核并行模式、单核工作模式等，满足用户不同功能需求和实时处理需求。

3 重构容错方案

容错控制模块的主要功能包括故障探测和故障修复两部分。故障探测主要通过对各个处理器的状态进行监视和对处理结果进行表决来完成；故障修复可以由系统软件实现，也可由硬件控制器来实现，修复方法包括处理器复位、系统复位、FPGA 部分重配置、FPGA 全局重配置等，具体由控制策略决定。

本方案设计中多个同构处理器核通过 IP 集成方式在同一芯片内部实现，4 个处理器核共享全局时钟。采用面向多核处理器的可重构结构来实现具备高可靠、高性能、低功耗的多处理器，论文设计的多处理器可重构体系结构的可靠性提高主要基于如下几个方面：

1)处理单元是本系统中设计最为庞大的模块，占用芯片面积也最大，在空间辐射环境中极易发生单粒子效应，因此采用针对处理单元的基于降级的重构容错策略来提高系统处理单元的可靠性。

2)设计基于在线部分重构的方式，在不影响其他处理单元正常工作的前提下，完成对处理单元的故障重构修复，进一步提高了处理单元的可靠性。

3)系统可靠性的提高要求控制模块必须足够可靠。本系统对容错控制模块(包括了硬件表决、输入控制模块等)进行了基于 TMR 和双模备份的加固设计。

4)未来在自动化可重构研究中，控制模块将在片外 FLASH 型或反熔丝型 FPGA 内实现，保证控制模块具备较高的容错能力。

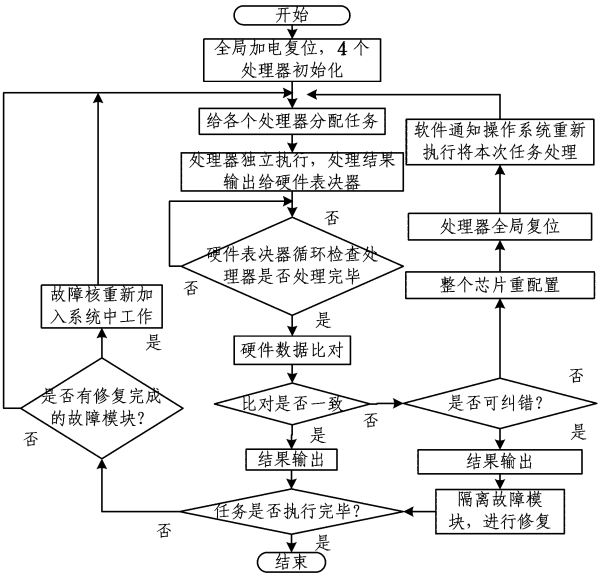

图 2 多处理器容错处理结构图

图 2 示出系统容错控制流程，4 个处理器接收到相同的外部处理数据输入，各自独立进行数据处理，由容错控制单元来进行数据的表决，由仲裁模块输出正确的处理结果数据。当某个处理器自检错误或其处理结果与其它处理器不一致时，在容错模块的控制下，对故障处理器核进行隔离、修复（复位或重构）以及同步处理，然后在操作系统的调度下重新开始新的处理过程。

本多核体系结构能够实现基于系统降级的可重构容错，重构容错过程状态转移图如图 3 所示。

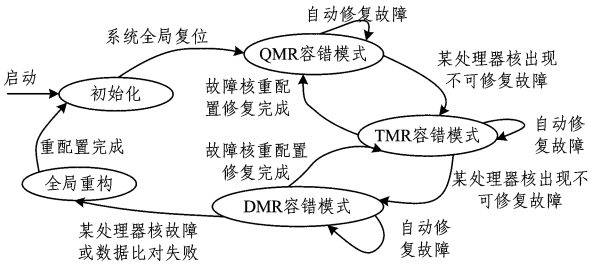

图 3　重构容错状态转移图

系统初始状态工作在拜占庭容错模式（QMR，Quad Modular Redundancy）下，在某一时刻某处理器发生故障时，系统将该故障处理器核隔离，系统降级为三模冗余工作模式（TMR，Triple Modular Redundancy），待该故障核修复完成后恢复到上一级四核容错模式工作。同理当系统工作在 TMR 工作模式，若有处理器故障，则系统降级为双机工作模式（DMR，Dual Modular Redundancy）继续工作。

系统在 DMR 模式下数据比对不一致，则系统无法辨别正确的处理器结果，系统进入一种全局错误状态，需要对整个 FPGA 进行修复（复位或重配置），重新回到初始拜占庭容错模式下继续工作。

该容错方案可以保证系统持续工作在高可靠性状态，假设处理器核出现故障的概率为 P_c，不考虑控制模块和 IO 等模块的故障情况，该多核处理器单元的可靠性概率：

$$P=\prod_{i=2}^{4}C_4^iP_C^i(1-P_C)^{4-i} \tag{1}$$

对于传统的 TMR 容错结构设计的处理器单元，其可靠性概率：

$$P=\prod_{i=2}^{3}C_3^iP_C^i(1-P_C)^{3-i} \tag{2}$$

对比四模容错（QMR）方案、传统三模容错（TMR）结构容错方案以及双模容错（DMR）结构在不同故障率（P_c）下系统的可靠概率，如图 4 所示，从图中可以直观地看到本方案在单核故障率低于 0.7 的范围内比 TMR 结构和单核结构具有明显的优势。系统初始状态下工作在可靠性最高的 QMR 状态，当某处理模块发生故障后，系统降级为 TMR 模式。如果在故障核修复期间又有 PE 出现故障，则系统继续降级，在 DMR 模式下工作。

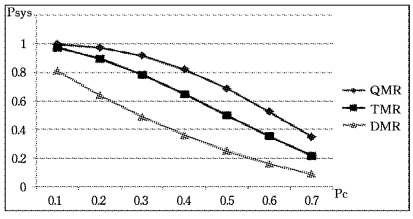

图 4　3 种结构的系统可靠性概率对比

4　PE 模块部分可重构恢复设计

在线部分可重构方法的设计流程分为模块划分、静态设计与验证、生成在线部分重配置流文件 3 部分。下面采用部分可重构方法对 PE 进行故障恢复设计。首先选定 PE 模块作为部分可重构设计的目标单元，并明确其与系统其他模块的数据通信接口。如图 5 所示，我们将系统划分为静态模块和动态模块两部分。静态模块包括了全局时钟、硬件表决器、输入控制模块和可重构容错控制模块等，这一部分在可重构过程中保持不变，动态模块主要包括 4 个 PE 模块，每个 PE 模块是一个能够支持动态部分重配置的单元模块。

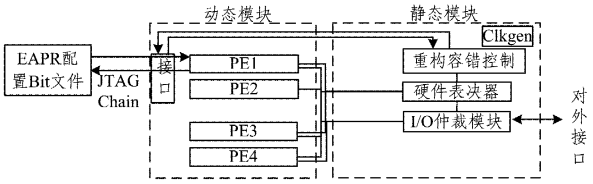

图 5　PE 部分可重构模块划分示意图

在进行完模块划分以及静态模块和动态模块的仿真验证后，建立部分可重配置工程，加载静态模块和动态模块对应生成的网表文件，在指定动态模块的区域约束后生成对应不同动态模块内部设计的部分可重配置流文件，存放在系统外部存储器中，在系统内 PE 模块发生故障后，对其进行实时在线部分重构，实现故障的修复。

5　测试验证

5.1　PE 最小系统结构设计

本多核体系结构中 PE 处理单元采用 Gaisler 公司的 LEON3 处理器[9]，主要包括如下组成部分：LEON3 处理器核、AMBA 总线及控制器、DSU 调试接口、存储控制器、低速 UART、定时器、I/O 接口等，如图 6 所示。

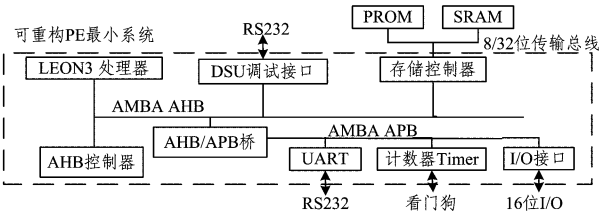

图 6　PE 最小系统体系结构设计

LEON3 是 Gailser 推出的一款基于 SPARC V8 架构的 32 位微处理器核，采用哈弗结构的七级流水线设计，主要针对嵌入式应用，具备高性能、低复杂度和低功耗的特点。

5.2　门级仿真验证

为了验证重构容错控制模块的功能，本文采用 QuestaSim 仿真环境对其进行仿真验证，图 7、图 8 所示为对容错控制模块的门级仿真验证波形图，图 7 中模拟了系统出现一个 PE 输出结果错误和四核出现拜占庭错误的情形，在单核故障下系统能够识别故障模块，其状态置为故障状态（不恒为 1）。当系统无法识别正确的结果时，则认为系统出现了故障，从而进入一种全局故障状态，需要进行全局复位修复后重新开始工作。

图 8 所示为门级仿真的时延分析，从图中可以看到容错模块的门级处理时延约为 9ns。

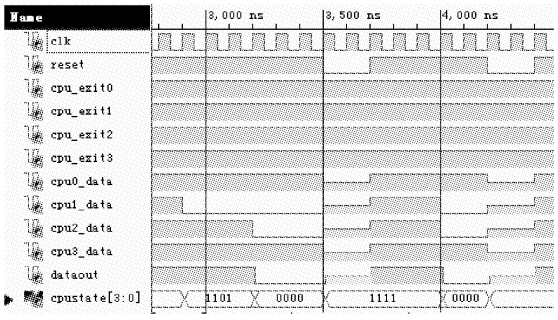

图 7　布局布线后容错控制仿真图

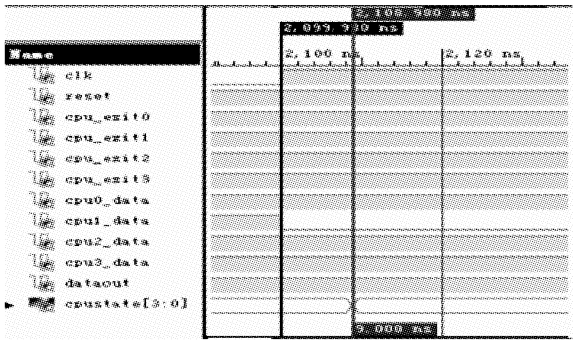

图 8　后仿时延分析图

5.3　硬件平台验证

为了对本系统设计进行实际硬件环境下的验证，我们搭建了基于 JTAG 的可重构验证平台。图 9 所示为本验证所采用的多处理器可重构验证平台，该平台采用 JTAG 连接线将计算机终端与可重配置 FPGA 相连接，通过 Xilinx 提供的 ChipScope 测试工具的交互式通信功能来对本论文设计的多核可重构系统功能进行验证。

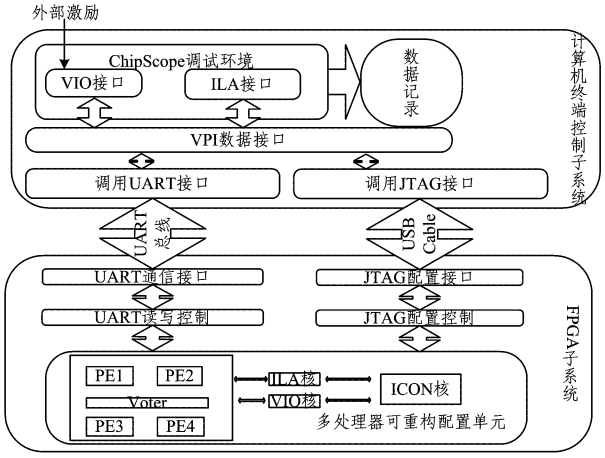

图 9　多处理器可重构验证平台示意图

如图 10 所示为基于本多处理器可重构硬件平台，该板集成了一块 Virtex 系列的 XC4VLX160 FPGA 芯片，并配备了 4 个 422 异步通信串口用于芯片与 PC 终端的通信和调试。我们通过 JTAG 下载链路将多处理器可重构系统生成的 BIT 文件加载到 FPGA 中，并通过边界扫描端口进行部分重配置测试。如图 10－图 13 所示为在硬件环境下对系统 PE 模块进行故障注入，从 ChipScope 逻辑分析仪中抓取到的芯片内部信号波形。从图中可以看到，系统从 QMR 模式降级到 TMR 模式，再降级到 DMR 模式的过程中处理模块输出和各个处理模块状态的变化。

在进行完系统降级的容错验证后，需要采用部分可重构方法，对发生故障的系统进行修复，如对于 TMR 结构下的故障模块，采用在线部分重配置进行修复，修复完成后在控制模块和操作系统的作用下重新加入到系统任务处理中。

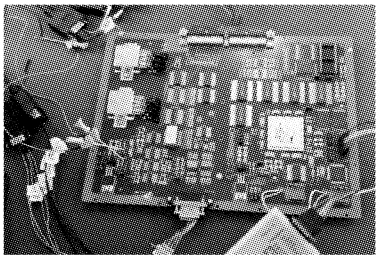

图 10　系统方案硬件平台

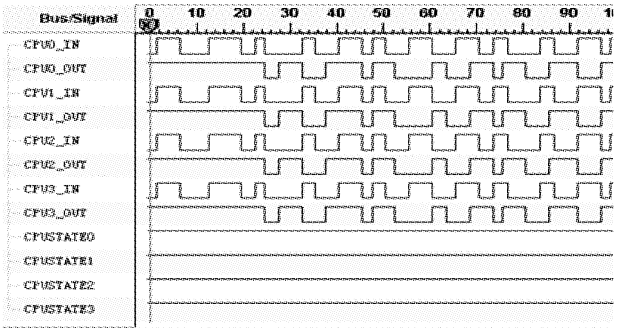

图 11　QMR 模式下系统内部信号波形图

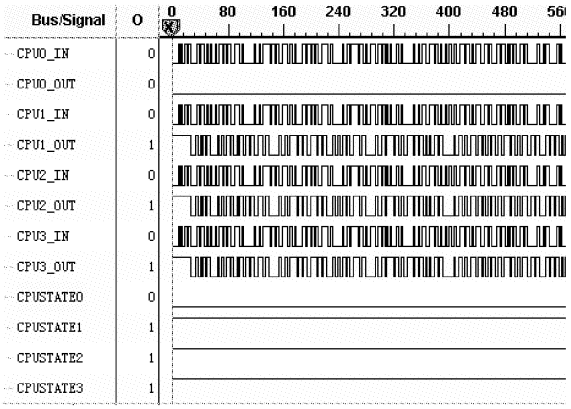

图 12　TMR 工作模式内部信号波形图

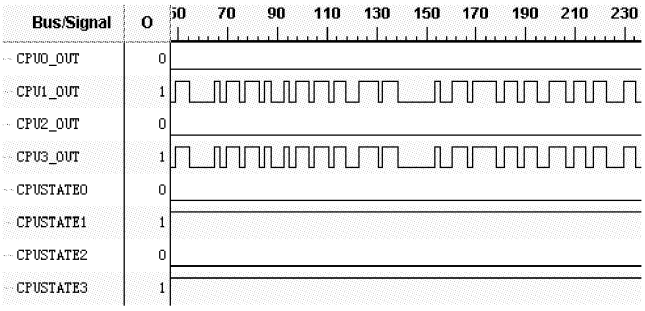

图 13　DMR 工作模式内部信号波形图

最后，将本文设计方案与 NASA 戈达德航天中心的四核可重构体系结构计算机 SpaceCube 做简单比较，如表 1 所列。从表中可以看到，本文设计的可重构系统支持对处理器核的重构，且采用片内 IP 软核集成方式实现了多个处理器，可以充分利用片上总线传递实现核间高速通信。由于本系统设计采用的频率较低，因此系统功耗也较低。由于采用对处理单元的重配置，导致了重配置时间较长，因此如何缩短重配置时间将是今后研究的重要方向。

表 1　两种设计方案对比分析表

对比项	本论文可重构设计方案	SpaceCube 设计方案
处理器核	LEON3	PowerPC405
处理器核是否可重构	是	否
处理器设计是否可裁剪	是	否
布局方式	单个芯片内集成	两个 FPGA 分离实现
互连方式	片内高速总线	片外总线
时钟频率	50 MHz	300MHz～450MHz
功耗	4.576W	＞10W
重配置时间	较长(毫秒级)	较短

结束语　当前星载电子设备亟需产品功能和性能的更新换代，本文提出了一种面向多处理器的可重构体系结构。本文将片上系统设计方法应用于多处理器系统设计中，实现了星载计算机系统的高度集成、小型化和低功耗设计。此外，本文在片上系统设计中引入基于系统降级的重构设计，并设计了基于硬件比对的容错控制策略，在硬件级提高系统可靠性和灵活性。动态部分重配置技术作为可编程器件的一种新技术，在本文设计中用于实现对故障模块的在线动态修复和实现系统功能的扩展，在进一步提高系统可靠性的同时，为系统设计带来了灵活性。

需要指出的是，本文研究依然存在一些有待改进和提高之处，例如对于复杂模块的可重构设计，缺少良好的指导方法；当系统时钟频率较高时，要求用户参与到布局过程中，增加了系统设计难度等。

参 考 文 献

[1] Wolf W, et al. Multiprocessor System-on-Chip (MPSoC) Technology[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2008, 27(10): 1701-1713

[2] Steven D, Guertin M. System on a Chip Devices—FY10[R]. JPL Publication, Dec. 2010: 10-20

[3] 张少林，杨孟飞，刘鸿瑾. 空间应用 SoC 研究现状[J]. 航天标准化，2012，149(3)：14-20

[4] Rafal G. Exploratory Study about the Use of New Reconfigurable FPGAs in Space[C]// San Diego, CA, 2011 NASA/ESA Conference on Adaptive Hardware and Systems (AHS). Piscataway: IEEE Operations Center, 2011: 220-226

[5] Ivars S, et al. Flexible High-Performance PPC On-Board Computer Archticture Based On Silicon-On-Insulator Technology[C]// Ivars S, Glauert W, Frickel J, et al. Flexible high-performance ppc on-board computer architecture based on silicon-on-insulator technology. 58th International Astronautical Congress. Hyderabad, India, 2007

[6] Tom F. Advanced Hybrid On-Board Science Data Processor-SpaceCube 2.0[C]// Arlington, NASA/GSFC, ESTO Earth Science Technology Forum. June 2010

[7] John S. Space Cube to Debut in 2007[J]. Goddard Tech Trends, 2006, 4(2): 2-3

[8] Dan E, et al. SpaceCube On-Board Science Data Processor[C]// Albuquerque, New Mexico, Military/Aerospace Programmable Logic Devices-2010. Nov. 2010

[9] Bauer L, et al. Concepts, Architectures, and Run-time Systems for Efficient and Adaptive Reconfigurable Processors[C]// AHS-2011. June 2011

[10] Harada T, et al. Evolving Complex Programs in Tierra-based On-Board Computer on UNITEC-1[C]//IAC-10. 2010

[11] 龚健. 基于在线进化的硬件容错方法研究[D]. 北京：中国空间技术研究院，2010

[12] 尚利宏，周密，胡瑜. 一种基于二阶近似域划分的可重构容错片上系统[C]//第六届中国测试学术会议. 2010：251-257

[13] Jiri G. A Portable and Fault-Tolerant Microprocessor Based on the SPARC V8 Architecture[C]// Proceedings. International Conference on dependable Systems and Networks. IEEE, Washington D C, 2002: 409-415

(上接第 54 页)

最大距离的测试码的最终目标，同时又降低了测试生成算法的复杂度，可以十分方便地实现测试码的生成算法，从而获得准完全最大距离测试序列的实用算法。大量的实验数据已经证明，本文提出的算法所生成的测试序列 QPMDTS 确实更符合和接近最大总距离测试序列的目标，大大地提高了传统随机测试法的测试效率，减少了测试序列长度，降低了测试成本。同时，大量实验也验证了本文所提算法在实践中的有效性和可行性。

参 考 文 献

[1] Bardell P H, McAnney W H, Savir J. Built-in test for VLSI pseudorandom Techniques [M]. NY, USA: John Wiley & Sons, 1987

[2] Dasgupta P, Chattopadhyay S, Chaudhuri P P, et al. Cellular automata-based recursive pseudoexhaustive test pattern generator [J]. IEEE Transactions on Computers, 2001, 50(2): 177-185

[3] Chaudhuri P P, Chowdhury D R, Nandi S, et al. Additive cellular automata: theory and applications[M]. Los Alamitos, Calif., USA: IEEE Computer Society Press, 1997

[4] Wagner K D, Chin C K, McCluskey E J. Pseudorandom testing [J]. IEEE Transactions on Computers, 1987, C-36(3): 332-343

[5] Chin C K, McCluskey E J. Test length for pseudorandom testing [J]. IEEE Transactions on Computers, 1987, C-36(2): 252-256

[6] Dan L, Jone W B. Pseudorandom test-length analysis using differential solutions [J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 1996, 15(7): 815-825

[7] Seth S C, Agrawal V D, Farhat H. A statistical theory of digital circuit testability [J]. IEEE Transactions on Computers, 1990, 39(4): 582-586

[8] Majumdar A, Vrudhula S B K. Fault coverage and test length estimation for random pattern testing [J]. IEEE Transactions on Computers, 1995, 44(2): 234-247

[9] Pradhan D K, Chatterjee M. GLFSR-a new test pattern generator for built-in-self-test [J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 1999, 18(2): 238-247

[10] Pradhan D K, Kagaris D, Gambhir R. A Hamming distance based test pattern generator with improved fault coverage[C]// Proceedings of the IOLTS. French Riviera, France, 2005: 221-226

[11] Xu Shi-yi, Chen Jian-wen. Maximum distance testing[C]// Proceedings of IEEE the 11th Asian Test Symposium (ATS' 2002). Guam, USA, 2002: 15-20

[12] Xu Shi-yi. Orderly random testing for both hardware and software [C]// Proceedings of 14th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC' 2008). Taipei, Taiwan, 2008: 160-167

[13] Xu Shi-yi. A quasi best random testing[C]// Proceeding of IEEE the 19th. Asian Test Symposium (ATS' 2010). Shanghai, China, 2010: 21-26

[14] Hyung K L, Dong S H. HOPE: an efficient parallel fault simulator for synchronous sequential circuits [J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 1996, 15(9): 1048-1058