软硬件协同设计的 SEU 故障注入技术研究

王 晶[1],荣金叶[2],周继芹[3],于 航[3],申 娇[3],张伟功[1,3]
(1. 首都师范大学信息工程学院,北京 100048;2. 北京微电子技术研究所,北京 100076;
3. 首都师范大学电子系统可靠性技术北京市重点实验室,北京 100048)

摘 要: 针对现有容错计算机故障注入方法缺乏对空间环境中频发的单粒子故障模型的支持,本文提出了一种利用背板技术的软硬件协同仿真与故障注入技术,分别针对寄存器部件和存储器部件的特性,设计了多位错误的单粒子故障模型,在寄存器传输级实现了通过软件生成故障并注入到硬件设计中的软硬件协同故障注入方案,避免了在硬件设计中修改代码生成故障破坏系统完整性的问题. 基于 Leon2 内核的故障注入实验表明,本文设计的平台为处理器容错设计提供了一个自动化、非侵入、低开销的故障注入和可靠性评估方案.

关键词: 容错; 故障注入; 软硬件协同; 单粒子翻转; 微处理器; 寄存器传输级
中图分类号: TP302.8 **文献标识码:** A **文章编号:** 0372-2112 (2018)10-2534-05
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2018.10.030

The Research on Software-Hardware Co-designed SEU Fault-Injection Technology

WANG Jing[1], RONG Jin-ye[2], ZHOU Ji-qin[3], YU Hang[3], SHEN Jiao[3], ZHANG Wei-gong[1,3]
(1. *College of Information Engineering, Capital Normal University, Beijing* 100048, *China*;
2. *Beijing Microelectronics Technology Institute, Beijing* 100076, *China*;
3. *Beijing Key Laboratory of Electronic System Reliability Technology, Capital Normal University, Beijing* 100048, *China*)

Abstract: The existing real-world or simulated fault injection methods cannot meet the requirements of reliability verification of nanoscale microprocessors for space applications, since they may introduce problems such as high cost, poor flexibility, poor observability, and low accuracy. This paper proposes a hardware/software cooperated fault injection scheme based on backplane, the time and positions of fault are generated in software, and injected into hardware design at register transfer level. Further, a multi-bit fault model focuses on radiation-induced soft error is proposed for register and memory. Experimental results show that the proposed software and hardware co-designed fault injection platform provides a high automation, randomicity and non-intrusion reliability evaluation method for fault-tolerant processor design.

Key words: fault-tolerant; fault injection; software-hardware co-design; single event upset; microprocessor; RTL

1 引言

随着集成电路器件特征尺寸的不断减小,空间环境中 SEU 会更加频繁,所引发的故障不仅导致单位错误,还会引发大量多位翻转错误[1]. 可靠性评估作为高可靠容错处理器设计过程中必不可少的重要一环[2],能够在系统设计早期对容错方案进行验证,尽可能的提高容错能力. 基于仿真的故障注入方法通过人工激活向目标系统引入故障,有效地缩短了故障的潜伏期,加速了系统失效过程. 然而现有仿真故障注入技术缺乏空间 SEU 故障的模型,并且在硬件设计中增加故障注入代码[3~5],破坏了系统的完整性和测试的可信性[6~9]. 因此如何实现自动化、非侵入的模拟故障注入是容错设计领域关心的热点问题.

针对上述问题,本文提出一种基于仿真背板的故障注入技术,借助硬件描述语言外部接口实现底层硬件信号和上层软件之间通信数据传输和调度. 提出面向单粒子事件的多位错误故障模型,设计基于影响系

收稿日期:2016-12-20;修回日期:2017-08-14;责任编辑:覃怀银
基金项目:国防 973 项目;国家自然科学基金(No. 61772350, No. 61472260, No. 61741211);北京市高水平教师队伍建设计划(No. CIT&TCD201704082, No. CIT&TCD20170322);体系结构国家重点实验室开放课题(No. CARCH201607);北京市科技新星计划(No. XX2018081);深圳市科技计划项目(No. JCYJ20150529164656096, No. JCYJ20170302153955969)

数的随机故障生成算法. 实现了一种非侵入式的故障注入平台,设计了自动化的故障注入信号获取和基于信号池的自动化目标信号选取方法. 本文所实现的方案在实际容错处理器设计中进行了验证和测试,并借助所实现的平台对容错流水线设计和 Cache 设计方案进行了调试和验证,为容错处理器的设计、验证和优化提供了行之有效的方案.

2　基于仿真背板的软硬件协同故障注入平台

基于背板技术设计的系统框架如图 1 所示,其中调试控制软件负责编译和配置在目标处理器上运行的软件程序,可以通过图形界面或命令行提供输入输出,通过网络或串口同底层硬件通信,软件通过仿真背板将调试控制命令发送给硬件仿真环境中的目标处理器,同时也通过仿真背板接收硬件处理器返回的系统状态用于调试. 故障注入软件通过用户界面接收故障注入时间和位置等参数,通过仿真背板接收硬件发送来的硬件信号列表,通过一个先深搜索的循环遍历,完成所有信号的分层识别和记录,建立分层资源池用于在故障注入时直接定位注入信号,然后生成针对硬件信号的故障库,所生成的故障和仿真控制信息通过仿真背板发送给目标处理器,并通过仿真背板接收处理器注入后的状态. 容错处理器原型在硬件仿真器中执行,通过带有跟踪缓存的内部调试单元 DSU 接收软件命令并发送执行状态.

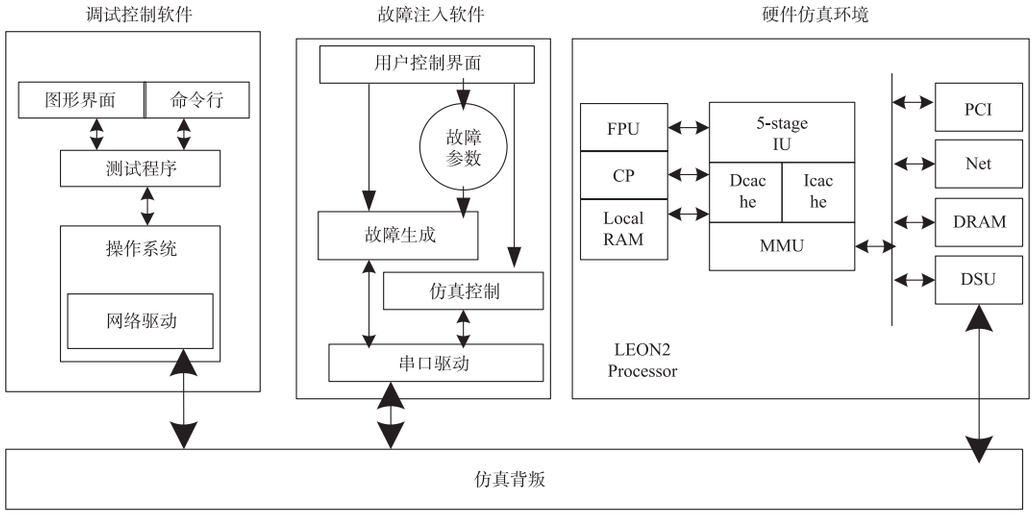


图1　系统平台框架图

仿真背板虚拟了网卡和串口等硬件设备,根据软硬件信息交互的需求将控制命令和故障注入信息发送给硬件 DSU,DSU 返回的信息则根据内容分发到不同的目的地,将调试器观察到的调试信息发送给调试软件,将信号状态等故障注入所需信息发送给故障注入软件,实现控制命令和返回结果等通信数据的传输和调度. 仿真背板通过 VHDL 和 C 语言的混合编程[10],借助于硬件描述语言的外部语言接口实现硬件仿真器和软件系统之间相互的关联、映射以及数据控制模块之间无缝的信息交互. 外部语言接口可以用 C 语言设计激励并在硬件仿真器中执行仿真验证任务. 硬件接口部分借助于硬件描述语言外部接口提供的信号监控和信号即时状态读取功能将硬件模块中关注的信号抽取出来,通过信号逻辑值强制赋值实现故障注入,注入后信号逻辑值立刻发生变化,不会影响电路中其他与该信号有逻辑关系的信号,模拟了单粒子事件发生时对电路的影响效果. 通过基于背板技术的协同仿真,完全脱离物理原型,使设计人员尽可能在设计早期发现并及时修改错误,从而降低开发成本. 仿真背板上软件部分对应硬件描述语言模块中的敏感信号例化,当逻辑电路触发该模块的敏感信号时,仿真器就会调用对应动态链接库中的程序,将硬件描述语言中定义的信号无缝映射到高级语言环境下,此时对这个信号逻辑值的任何处理都可以基于软件的算法来实现,高级语言无论在流程控制还是函数调用方面都比硬件语言和脚本语言要容易,不再受到硬件语言以及模拟器仿真命令的局限性,因而能够支持更加复杂的故障模型和算法,并且高级语言具有良好的可移植性,便于更多功能的扩展. 而软件程序下载到芯片中时不会被综合,因此不增加额外逻辑,也避免了对硬件模块的侵入和影响,既保护了目标模型的完整性,又使得测试结果更具有真实性.

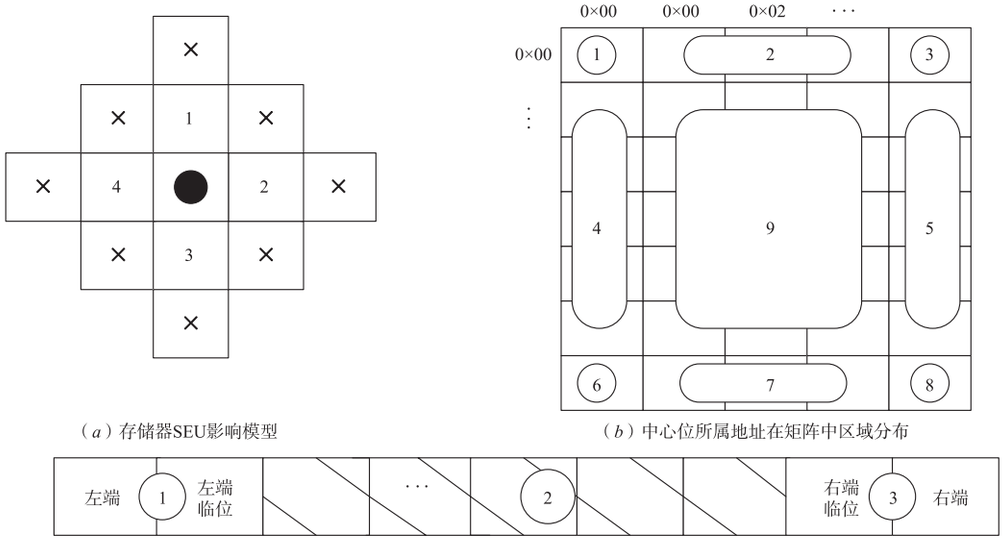
2.1　寄存器故障模型

由于寄存器和存储器的结构差别较大,因此所对

应的故障模型也有所不同. 对于寄存器模型,本文以 8 位寄存器和最多 4 位的多位翻转为例设计. 假设高能粒子在寄存器 0 ~ 8 个数据位间随机打中一位,将此位设为中心位. 中心位分为端口位和非端口位. 中心在端口位的情况只会在一个方向上产生多个同时翻转的数据位,因此只需要在 0 ~ 3 范围内产生一个随机数 neighbor 代表与相邻 neighbor 位同时翻转. 非端口位由于会在两个方向上出现同时翻转的数据位,因此需要进行 2 次随机决定翻转位的位置. 第一次随机根据在 0 ~3 之间取值的首次影响系数选择,出现 0 代表只本位翻转,1 代表传递方向向左,2 代表传递方向向右,3 代表双向传播;第二次随机根据取值在 0 ~ 2 之间的递进系数选择,当首次影响系数为 1、2 时,递进系数为 0 代表无影响,1 代表与传递方向相同的相邻第 2 位参与翻转,2 代表第 3 位也一同参与翻转;当首次影响系数为 3 时,递进系数为 0 代表无影响,1 代表与左侧相邻的第 2 位参与翻转,2 代表右侧相邻第二位翻转. 当首次影响系数为 0 时递进系数无效. 最后每当有相邻位需要翻转时,都必须检查该相邻位是否超过寄存器数据位数.

2.2 存储器 SEU 故障模型

存储器作为大量规整的记忆单元,电荷共享现象更加明显,是单粒子多位翻转的高发地区[11]. 与多位寄存器一维模型不同,存储器需要建立二维模型. 在一个单元被粒子打翻后,其上下左右单元均可能受到影响,考虑多位翻转的故障模型如图 2(*a*)所示,中间黑色的点为入射中心位,编号为 1、2、3、4 的 4 个点称为首次影响点,其余的点称为递进影响点.



(*a*) 存储器SEU影响模型 (*b*) 中心位所属地址在矩阵中区域分布

(*c*) 中心位在字中的区域分布

图2 存储器故障模型

当故障发生在矩阵边界附近需要对模型进行适当的剪裁,模型的裁剪主要由粒子入射的中心位所在字中的位置以及该字在矩阵中的位置共同决定. 中心位所在字位置控制横纵两个方向故障位置的选择,因此将存储器模型分为 9 个区域,如图 2(*b*)所示. 字内的数据分为 3 部分,如图 2(*c*)所示,以两边端口向内侧相邻的一位为界. 当中心位落在阴影部分处,则横向翻转的数据位只在该字内;如果中心位落在阴影两侧的部分,则横向翻转的数据位可能分别影响到各自相邻的字. 在具体对模型进行裁剪、判断入射粒子可能影响相邻区域的范围时,需要结合上述两个因素共同分析. 第一步,先要判断发生 SEU 的中心位所在字的区域,将中心位置与两个临界值做比较. 第二步,判读该中心位所属字在矩阵的位置. 这一步需要结合存储矩阵中地址间的排列结构. 第三步,结合第一步与第二步取其交集,确定具体 SEU 的影响模型. 在建立完 SEU 影响区域模型后,就要确定具体的故障位置,方法则采用多位寄存器的算法. 首先随机产生中心位纵方向被影响的地址及数据位,然后再以它们各自为中心位结合影响区域模型随机产生横向地址及数据位,并记录下来.

2.3 软硬件协同故障注入机制

故障模型根据测试人员所配置的注入参数,通过故障生成算法产生一系列的随机故障序列,在处理器仿真时根据所生成的故障序列进行故障注入. 系统中哪些信号以及其逻辑值发生怎样的改变,要符合测试人员设计此次故障试验的目的,能够反应出人们所希望发生的某类故障对系统的影响效果. 因此故障序列的设计直接决定了能否有效的模拟预期故障.

故障模型中每个故障的特征通常由故障注入时间、故障类型、故障持续时间,故障位置和错误位数等参数共同表示,每个参数的变化都代表着一组新的故障序列,其模式可用如下所示的 5 元式表示:$F_{\text{Mode}} = < T,L,M,D,N >$. 其中 T 表示故障注入时间,决定系统在何时引入故障,它可以是试验过程中的任意时刻. 对于故障注入时间的选取,测试者可以规定在一个时间范围内随机取值,也可以遵循某种分布有规律的产生. L 表示故障发生位置,故障位置同故障类型相关,例如 SEU 故障通常发生在具有记忆功能的单元中,如寄存器和存储器中. M 表示故障类型,可以是固定 0/1、位翻转等. D 代表故障延时时间,也称为故障持续时间,通过设定延时并在持续一段时间后将故障效果撤销,表示故障影响的时间范围. N 表示错误位数,针对 SEU 故障设定故障位数为 1 ~4 位.

单粒子故障通常是瞬时故障,故障类型表现为位翻转,因此持续时间和类型参数在面向单粒子的故障注入模型中可以设为固定值,而故障模型可以化简为(时间,位置,错误位数)的三元组. 故障影响的位数在 1 ~4 之间随机选取,其中 1 位故障比例高于多位故障. 而故障注入时间方面,我们规定每次注入只模拟一个高能粒子轰击的情况,即只发生一次 SEU 效应,于是注入次数 MAX 就会产生 MAX 个注入时间. 通过配置故障注入的时间范围参数(T_s,T_e)以及故障总次数 MAX,利用随机函数即可生成故障时间序列 $T(\text{MAX},T_s,T_e) = \{t_1,t_2,t_3\cdots t_{\max}\}, t_i \in (T_s,T_e)$,所生成的时间序列按从小到大排序.

在硬件模块中加入一个时间脉冲发生器. 时间脉冲发生器以纳秒为单位触发故障注入模块. 注入模块被调起后获取此时系统仿真时间,检测是否满足故障库中元素的故障注入时间. 元素在故障库中按照时间顺序从小到大排列,每次只检测指针所指的元素,如果系统仿真的时间恰好匹配指针所指元素的故障注入时间,则开始实施故障注入,完成注入后指针下移.

3　实验评测

采用本文所设计和实现的容错平台,验证一款面向航天应用的自修复双冗余流水线的容错方案[12]. 以最多 4 位翻转的 SEU 模型为例,分析了分别向处理器中的 Cache 和自修复双冗余流水线中的级间寄存器进行 MBU(Multiple Bit Upset)故障注入的效果. 每个测试程序的仿真运行时间约为一万纳秒,因此将其平均分为 10 段,每次随机从信号故障池中挑选一个寄存器在每个时间段中进行故障注入. 工作负载采用控制类程序冒泡排序,大量 Cache 访问的矩阵乘法,频繁使用 ALU 的加法计数,以及 Codelink 基准测试程序. 用 C 语言编写程序源代码,由针对 SPARC V8 体系结构的编译器 SPE-C 进行编译. 表 1 首先统计向流水线注入故障所引发的单位和多位错误的数量,可以看到所有的故障注入都在处理器中引发了错误.

表 1　目标流水线发生错误数量统计

	故障错误位数分布			
注入总次数	1 位错	2 位错	3 位错	4 位错
10000	7218	1318	1331	133
5000	3664	639	637	60
1000	713	121	157	9

对 Cache 的数据存储器和标记存储器分别注入故障,统计 1 位到 4 位错误发生的次数,结果如表 2 所示,其中由于存储器的故障模型每次故障最多影响 9 位,因此注入故障影响的字数大于注入次数. 此外,流水线中部件分散,且存在大量单比特的信号,因此错误位数的分布也保持了大部分为单比特错误,同时发生一定比例多位错误的特性. 而 Cache 中存储密集,因此多比特错误比例大于单比特错误.

表 2　目标 Cache 发生错误数量统计

	Cache 数据存储器			
	1 位错	2 位错	3 位错	4 位错
冒泡排序	1179	2070	2849	594
逻辑运算	82	129	66	44
矩阵乘法	2259	72	2416	1843
CoreMark 程序	13	120	58	119
	Cache 标记存储			
	1 位错	2 位错	3 位错	4 位错
冒泡排序	2	3286	0	177
逻辑运算	4	132	40	72
矩阵乘法	5	1476	5698	1
CoreMark 程序	4	35	29	3

分析故障注入平台向运行着不同负载的容错处理器进行故障注入后引发的容错效果. 虽然四种测试用例由于各类指令所占的比例不同而对单粒子效应的敏感程度有差别,但哑故障的比例都在 65% 左右,只有 35% 左右的故障因为影响程序的执行而被检测出来进行流水线回退. 容错方案使得 65% 左右的哑故障被忽略,30% 左右的故障被自校验定位,启用流水线快速恢复机制,剩余 5% 左右的故障无法根据自校验进行定位,将会启用流水线整体恢复机制. 实验结果证明故障注入平台能够有效的触发处理器中的故障,并且引发不同容错方案的执行,能够达到早期调试和验证处理器及其容错方案的目标.

4 结论

本文针对空间辐照环境下的 SEU 故障,提出了基于仿真背板技术的软硬件协同故障注入技术. 针对流水线中不同结构部件特性分别建立 SEU 故障模型和故障生成算法,并通过硬件虚拟化实现了软硬件协同控制,通过高级语言和硬件描述语言的混合编程实现了通信信息的传输和调度. 在实际容错处理器原型中容错方案的测试结果证明,本文所提出方法不仅简单实用、效果直接、易于建模而且还具有自动化程度高、移植性强、不破坏系统目标完整性等特点,为处理器的仿真验证与可靠性评估工作提供了重要支持.

参考文献

[1] Touloupis E, Flint J A, Chouliaras V A. Study of the effects of SEU-induced faults on a pipeline protected microprocessor[J]. IEEE Transactions on Computers, 2007, 56(12): 1585 – 1596.

[2] Tang D, Iyer R K. Dependability measurement and modeling of a multicomputer system[J]. IEEE Transactions on Computers, 1993, 42(1): 62 – 75.

[3] Barton J H, Czeck E W, Segall Z Z, et al. Fault injection experiments using FIAT[J]. IEEE Transactions on Computers, 1990, 39(4): 575 – 582.

[4] Seungjae Han and K. G. Shin. Experimental evaluation of failure-detection schemes in real-time communication networks[A]. Proceedings 27th International Symposium on Fault Tolerant Computing[C]. Los Alamitos: IEEE Press, 1997. 122 – 131.

[5] Kanawati G A, Kanawati N A, Abraham J A. FERRARI: a flexible software-based fault and error injection system[J]. IEEE Transactions on Computers, 1995, 44(2): 248 – 260.

[6] Arlat J, Boué J, Crouzet Y, et al. Fault Injection Techniques and Tools for Embedded Systems Reliability Evaluation [M]. Boston: Springer, 2003. 177 – 193.

[7] Gil D, Baraza J C, Gracia J, et al. VHDL Simulation-based Fault Injection Techniques[M]. Boston: Springer, 2003. 159 – 176.

[8] Zarandi H R, Miremadi S G, Ejlali A. Ejlali. Dependability analysis using a fault jnjection tool based on synthesizability of HDL models[A]. Proceedings 18th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems[C]. Washington, DC, USA: IEEE Computer Society, 2003. 485 – 492.

[9] Baraza J C, Gracia J, Blanc S, et al. Enhancement of fault injection techniques based on the modification of VHDL code[J]. IEEE Transactions on Very Large Scale Integration Systems, 2008, 16(6): 693 – 706.

[10] Czeck E W, Siewiorek D P. Effects of transient gate-level faults on program behavior[A]. Digest of Papers Fault-Tolerant Computing: 20th International Symposium[C]. New York: IEEE, 1990. 236 – 243.

[11] Kim S, Somani A K. Soft error sensitivity characterization for microprocessor dependability enhancement strategy [A]. Conference on Dependable Systems and Networks [C]. New Zealand: Digital Press, 2002. 416 – 425.

[12] Wang Jing, Yang Xing, Zhao Yuanfu, et al. Multi-bits error detection and fast recovery in RICS cores[J]. Journal of Semiconductors, 2015, 36(11): 75 – 79.

作者简介

王 晶 女,1982 年生于黑龙江哈尔滨,首都师范大学副教授. 研究兴趣包括计算机系统结构、容错计算、高能效计算.
E-mail: jwang@cnu.eud.cn

荣金叶 女,1983 年生于河北邯郸,硕士,工程师. 研究方向为嵌入式系统设计.
E-mail: yezi_0306@163.com

张伟功(通信作者) 男,1967 年生于山西临猗,首都师范大学信息工程学院研究员、博士生导师. 研究方向为高可靠嵌入式计算机体系结构与应用技术.
E-mail: zwg771@cnu.edu.cn