

基于模拟的故障注入技术在高可靠处理器的应用

魏方巍, 刘海南, 黑 勇

(中国科学院 微电子研究所, 北京 100029)

摘 要: 高可靠处理器在设计过程中, 需要在不同阶段采用适当的故障注入技术, 对其可靠性进行验证和评估. 以 LEON3 高可靠处理器中的 TMR(Triple Module Redundancy)flip-flop 为例, 使用基于模拟的故障注入技术对高可靠处理器进行故障注入. 通过实验表明, 采用此类故障注入技术, 可以在设计前期对加固设计的可靠性进行快速验证, 缩短开发周期, 降低验证成本.

关键词: 高可靠性设计; LEON3; TMR; 故障注入

中图分类号: N945.17

文献标识码: A

文章编号: 1000-7180(2012)08-0070-04

The Application of Simulation-Based Fault Injection on High Reliable Processor

WEI Fang-wei, LIU Hai-nan, HEI Yong

(Institute of Microelectronics, Chinese Academy of Sciences, Beijing 100029, China)

Abstract: During the design process of high reliable processor, appropriate fault injection technologies should be adopted at different stages to analysis and evaluation the reliability of the processor. On the basis of LEON3 high reliable processor, the simulation-based fault injection has been used to inject faults into Triple Module Redundancy flip-flops. Experiments show that the simulation-based fault injection can be used in the early design stage of high reliable processor to shorten the development cycle and reduce the cost of validation.

Key words: high reliability design; LEON3; TMR; fault injection

1 引言

处理器的高可靠性设计通常会从器件工艺、电路、体系架构以及软件等不同级别进行加固. 处理器进行加固设计之后, 为了在较短时间内验证所加固的部分是否达到了可靠性方面的要求, 需使用特定的故障注入技术人为地注入故障, 之后观察并分析处理器的行为, 从而判断和评估加固设计对处理器可靠性的影响^[1].

通常来讲, 故障注入技术分为三大类: 物理注入、软件注入和基于模拟的故障注入^[2]. 在高可靠处理器的设计阶段, 基于模拟的故障注入技术是一种有效的可信性评估手段. 它在高可靠处理器的仿真

模型中注入故障, 通过进行仿真和分析, 可以在设计的早期阶段发现错误并进行修正, 从而缩短了开发周期, 降低了开发成本^[3]. 同时, 由于这类故障注入技术以仿真模型为基础, 在故障注入过程中可以提供更好的可观察性和可控制性.

2 基于模拟的故障注入技术

基于模拟的故障注入技术主要分为两类: 基于仿真命令的故障注入和基于模型修改的故障注入.

基于仿真命令的故障注入技术依靠仿真器提供的命令对 VHDL 代码中的信号和变量的值进行控制, 从而达到故障注入的目的. 信号控制的仿真命令序列如图 1 所示.

收稿日期: 2012-01-06; **修回日期:** 2012-02-05

基金项目: 国家重大科技专项(2009zx02306-003)“高可靠库单元与产品设计”

[1]Simulate Until	<fault injection time>
[2]Freeze Signal	<signal name><signal value>
[3]Simulate For	<fault duration>
[4]Unfreeze Signal	<signal name>
[5]Simulate For	<observation time>

图 1 信号控制的仿真命令序列

在信号控制的仿真命令序列中,故障的注入是通过改变信号的值来实现的. 首先将仿真停止在需要注入故障的时刻,然后强制改变信号的值为需要的值,设定故障注入的持续时间,之后去掉外部的强制赋值,继续仿真直至到达观察时间. 在实际注入过程中,可以根据注入的实际情况对上述序列进行适当调整,

变量控制根据变量是否和时间相关,又可以分为时间相关变量控制和时间无关变量控制. 时间相关变量的值较易控制,其故障注入与信号类似. 时间无关变量的值不易控制,需要更细粒度的控制序列才能改变其值^[1],如图 2 所示.

[1]Simulate Until	<fault injection time>
[2]SetSrcBreakOnLine	<LineNo><Library><Entity> <Library><Architecture>
[3]Repeat Step[2]at Different Lines	
[4]Simulate For	<time window>
[5]If (AtSrcBreak)	{
[6]Assign Variable	<var name><var value>
[7]DelAllSrcBreaks	
[8]Simulate For	<observation time>
	}

图 2 时间无关变量控制的仿真命令序列

在时间无关变量控制的仿真命令序列中,首先将仿真运行至需要注入故障的时刻,根据需改变的目标变量设置断点(可以设置多个断点),然后继续运行仿真,直至运行至断点处,此时改变变量的值,继续运行仿真至观察时刻,故障注入完成.

基于模型修改的故障注入技术通过修改 VHDL 代码来实现故障注入,这类故障注入技术有两种实现途径:一种是破坏技术,另一种是变异技术^[4]. 破坏技术是在 VHDL 模型中加入专门的故障注入模块,故障注入模块通过改变模块之间连接信号的值,或者通过改变其时序特征来实现故障注入. 变异技术是用一种新组成部分的描述替代原组成部分的描述. 当新组成部分被激活时,它会模拟所替代的原组成部分发生故障下的行为.

3 LEON3 高可靠处理器

LEON3 处理器核是一款基于 SPARC V8 结构

的 32 位处理器模型,其内部包含 7 级整数流水线,支持硬件乘/除法以及分立的指令与数据 Cache. LEON3 处理器核具有高可配置性,设计者可以通过配置优化处理器的性能、功耗、面积以及成本. LEON3 通过高速的 AMBA AHB 总线连接高速器件和接口,片外的低速接口通过 AMBA APB 总线连接,如 PS/2、定时器、VGA 等^[5],如图 3 所示.

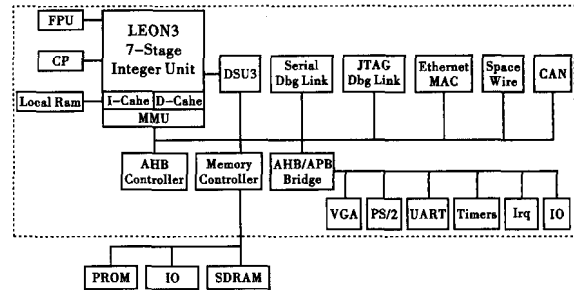


图 3 LEON3 处理器的典型系统

LEON3 高可靠处理器(以下简称 LEON3HR)是在 LEON3 处理器核的基础上,通过 TMR registers、奇偶校验、片上 EDAC(Error Detection and Correction)、流水线重启以及强迫 cache 丢失等处理器容错技术来实现对 SEU(Single Event Upset)的容错^[6].

本文以 LEON3HR 流水线中加固的 flip-flop 为例,进行模拟故障注入并验证加固效果. LEON3 处理器的流水线中大约有 800 个 flip-flop, LEON3HR 使用 TMR 技术对这 800 多个 flip-flop 进行加固. TMR 是最常用的冗余技术之一,它对原始电路进行复制,将原始电路和另外两路复制电路的信号通过表决器,以少数服从多数的原则输出最终结果. 按照此技术对 flip-flop 进行加固后,原来单一的 flip-flop 变为图 4 所示的结构.

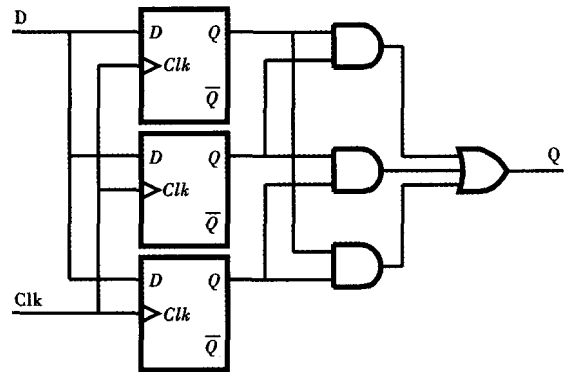


图 4 使用 TMR 加固后的 flip-flop 结构

如果其中的一个 flip-flop 发生翻转,由于 TMR 采用的是多数判决原则,在一个时钟周期内错误会

被自动清除,并不会影响 flip-flop 数据的正确性,但是加固后资源消耗会有较明显的增加. 根据 TMR flip-flop 的结构,对 800 个 flip-flop 进行 TMR 加固,大约会增加 1 600 个寄存器. 若选用 Stratix II 系列的器件,使用 Quartus II 进行综合,加固前后的资源消耗分别如表 1 和表 2 所示.

表 1 flip-flop 加固前的资源消耗

	占用数	总数	占用比例
组合算术逻辑运算单元	3 973	12 480	32%
专用逻辑寄存器	2 316	12 480	19%
针脚	311	367	85%
存储器	81 408	419 328	19%

表 2 flip-flop 加固后的资源消耗

	占用数	总数	占用比例
组合算术逻辑运算单元	5 127	12 480	41%
专用逻辑寄存器	4 011	12 480	32%
针脚	311	367	85%
存储器	81 408	419 328	19%

从表 1 和表 2 中可以看到,加固后组合算术逻辑运算单元和专用逻辑寄存器的数目均有明显增加. 所占用的组合算术逻辑运算单元从 3 973 个增加到 5 127 个,占用比例增加了 9%. 寄存器总数从 2 316 个增加到 4 011 个,实际增加了 1 695 个.

4 故障注入

本实验以 LEON3HR 流水线中加固的 flip-flop 为例,以 Modelsim SE 仿真器为平台,分别对 LEON3HR 的 TMR flip-flop 进行基于仿真命令的故障注入和基于模型修改的故障注入,并对 TMR flip-flop 的加固效果进行验证.

4.1 使用仿真命令注入故障

基于仿真命令的故障注入涉及到的信号和变量的注入. 实验中模拟瞬态故障效应进行故障注入,实际执行的故障注入分为两部分:信号控制和时间相关变量控制.

按照第 2 部分中给出的信号控制仿真命令序列,对 LEON3HR 流水线中 Register Access 阶段的 TMR flip-flop 注入故障. 该阶段的 TMR flip-flop 涉及到四个信号 rtmr0rd、rtmr1rd、rtmr2rd 和

routrd. 其中 rtmr0rd、rtmr1rd 和 rtmr2rd 是 TMR flip-flop 中三路触发器对应的信号,routrd 是三路触发器信号经过判决之后的输出. 具体的故障注入步骤如下:首先将整个工程的仿真运行至需要注入故障的时刻 4 300 ns 处,之后对信号 rtmr0rd 注入故障值 10001000,确定故障注入的持续时间为 40 ns,然后移除仿真器的强制赋值,继续运行仿真至观察时刻,如图 5 所示.

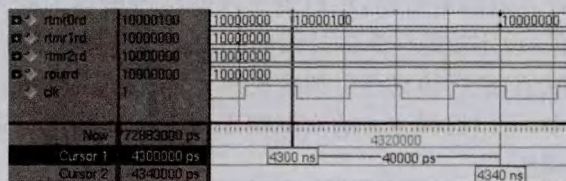


图 5 TMR flip-flop 中信号的瞬态故障注入

从图 5 可以看出,虽然在仿真的时间区间 4 300~4 340 ns 内对信号 rtmr0rd 进行了故障注入,强制将其值 10000000 改变为 10001000. 但是由于 TMR 的故障屏蔽效应,在时间区间 4 300~4 340 ns 内的最终输出 routrd 并没有发生错误,一直保持为正确值 10000000. 通过对 TMR flip-flop 中的信号进行故障注入,可以验证当 TMR flip-flop 中的任一 flip-flop 发生翻转时,TMR flip-flop 的输出并不受影响,达到了加固的目的.

以上是对信号进行控制实现故障注入,对于 TMR flip-flop,也可以通过控制变量的值来达到相同的故障注入效果,如图 6 所示.

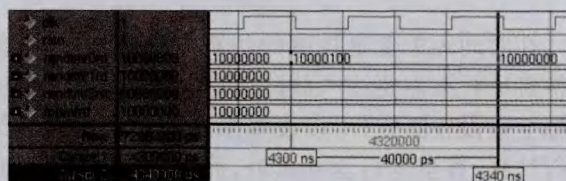


图 6 TMR flip-flop 中时间相关变量的瞬态故障注入

其中 randtmr0rd、randtmr1rd、randtmr2rd 是三路 flip-flop 信号分别相与后的变量输出,rortmrrd 是上述三个变量多数表决后的输出变量. randtmr0rd、randtmr1rd、randtmr2rd 和 rortmrrd 都是时间相关的变量.

4.2 修改模型注入故障

通过修改 VHDL 模型注入故障的形式比较多,但是总体上可以按照第 2 部分的介绍分为两类:破坏技术和变异技术^[7]. 本实验采用变异技术,将 TMR flip-flop 中的一条支路用新的描述来替代,另外两条支路保持原描述不变.

对于 TMR flip-flop,新描述在原描述的基础上引入一个选择信号 sel,当选择信号 sel 未被激活时,新描述与原描述等价.当选择信号 sel 被激活时,原寄存器值的取反后送入判决器.只要注入时间控制得当,其效果等同于注入了单粒子翻转故障.若采用 CASE 语句来描述变异的支路,其关键部分的伪代码如下所示:

```
CASE sel IS
  WHEN 00=>正常赋值电路描述;
  WHEN 01=>寄存器值取反后的赋值电路描述;
  WHEN 10=>其他故障模式下的赋值电路描述;
  ...
END CASE
```

实验中仍对寄存器 rtmr0rd^[3]注入故障,将该支路的 sel 激活,持续时间为 40 ns.在该段时间内,rtmr0rd 由原来的 0 变为了 1.但是由于 TMR 的故障屏蔽效应,判决后的输出并未受 rtmr0rd 值变化的影响,说明 TMR flip-flop 达到了加固的目的.

5 结束语

文中以高可靠处理器 LEON3HR 流水线中的 TMR flip-flop 为模型,在设计前期分别使用仿真命令控制和模型修改技术对 Register Access 阶段的 TMR flip-flop 进行故障注入实验,并验证了其加固效果.通过实验可以看到,使用基于模拟的故障注入技术可以在高可靠处理器的设计前期对其加固环节进行快速有效的验证,缩短了开发周期,节省了验证成本.

参考文献:

[1] 段青亚,黄士坦,辛明瑞.空间单粒子故障容错设计的

验证技术研究[J].微电子学与计算机,2007,24(11):38-41.

- [2] Gracia J C J, Gil D. Improvement of fault injection techniques based on VHDL code modification [C]// High-Level Design Validation and Test Workshop. USA: Washington, 2005:19-26.
- [3] Jenn E, Arlat J, Rimén M, et al. Fault injection into VHDL models: the MEFISTO tool [C]// Proceedings 24th International Symposium on Fault-Tolerant Computing (FTCS-24). Austin, TX, USA, 1994: 356-363.
- [4] 曾宪炼,马捷中,任向隆,等.基于 VHDL 的故障注入技术[J].计算机工程,2010,36(11):244-246.
- [5] Jiri Gaisler. GRLIB IP core user's manual [EB/OL]. <http://www.gaisler.com>,2007.
- [6] Jiri Gaisler. A portable and fault-tolerant microprocessor based on the SPARC V8 Architecture [C]// IEEE Proceedings of the International Conference on Dependable Systems and Networks(DSN'02). USA: Bethesda, 2002.
- [7] 赵广燕,孙宇锋,康锐,等.电路故障仿真中的故障建模、注入及判定方法研究[J].微电子学与计算机,2007,24(1):143-146.

作者简介:



魏方巍 男,(1986—),硕士研究生.研究方向为高可靠性数字电路设计.