ISSN 1004-9037,CODEN SCYCE4
Journal of Data Acquisition and Processing Vol. 36,No. 4,Jul. 2021,pp. 822－830
DOI:10. 16337/j. 1004-9037. 2021. 04. 020

http://sjcj. nuaa. edu. cn
E-mail:sjcj@nuaa. edu. cn
Tel/Fax:＋86-025-84892742

基于FPGA硬件的单粒子翻转模拟技术

施聿哲，陈　鑫，陈　凯，白雨鑫，张　颖

（南京航空航天大学电子信息工程学院，南京 211106）

摘　要：由于航空航天活动越发复杂，深空通信和姿态控制等航空航天电子系统大量采用集成电路芯片以提高各方面性能。随着集成电路工艺节点的进一步缩小，电路受到单粒子效应而发生错误的概率越来越大。评估集成电路对单粒子翻转（Single event upset，SEU）的敏感性对航空航天的发展具有重要意义。电路规模的增加和系统功能集成度的提高给评估速度带来了严峻挑战。本文提出了一种能适用于超大规模集成电路（Very large scale integration，VLSI）的快速故障注入方法。该方法可通过脚本自动分析电路，并修改逻辑使电路具备故障注入功能。实验结果表明，该方法实现的故障注入速度可以达到纳秒级，可大幅缓解电路规模和评估时间之间的矛盾，从而满足VLSI的评估需求。

关键词：单粒子翻转；超大规模集成电路；故障注入

中图分类号：TN47　　**文献标志码：**A

FPGA Based Single Event Upset Simulation Technology

SHI Yuzhe，CHEN Xin，CHEN Kai，BAI Yuxin，ZHANG Ying

（College of Electronic and Information Engineering，Nanjing University of Aeronautics and Astronautics，Nanjing 211106，China）

Abstract：Due to the increasing complexity of aerospace exploration，integrated circuits are applied in many aerospace electronic systems such as deep space communication and attitude control. With the further shrinking of integrated circuit technology，the probability of errors in circuit due to single event effects has become higher. Evaluating the sensitivity of integrated circuits to single event upset（SEU）is of great significance to the development of aerospace. The continuous increase of circuit scale and the improvement of system function integration pose severe challenges to the speed of evaluation. For this reason，this paper proposes a fast fault injection method suitable for very large scale integration（VLSI）. This method can automatically analyze the circuit through scripts，and modify the logic to make the circuit available for fault injection. Experiment results show that the fault injection speed can reach nanosecond level，which can alleviate the contradiction between circuit scale and evaluation time. Consequently，it can meet the evaluation requirements of VLSI.

Key words: single event upset(SEU); very large scale integration (VLSI); fault injection

基金项目：国家自然科学基金(61106029，61701228)资助项目；航空科学基金(20180852005)资助项目；模拟集成电路重点实验室基金(61428020304)资助项目。
收稿日期：2020-10-22；**修订日期：**2021-03-11

引　言

随着制造工艺的发展,集成电路特征尺寸减小,改变电路工作状态所需要的能量阈值呈几何级下降趋势,使得集成电路更容易受到辐射照射的影响[1]。地球外层空间为强辐射环境[2],给航天设备中集成电路的正常运作带来了极大的挑战。辐照对集成电路可恢复性故障中最主要的影响为单粒子翻转(Single event upset, SEU),表现为存储单元中的比特位翻转。因此,评估集成电路对SEU的敏感性至关重要。

通常采用故障注入技术来模拟宇宙空间环境中SEU对集成电路的影响,其中最经典的方法是将电路暴露在模拟的空间辐射环境中进行故障注入[3-4],以获得与实际辐射环境下最为接近的实验数据。但是这种方法存在损害电路的风险,且测试成本十分昂贵,通常只有在电路设计的最终测试阶段才会采用。基于软件仿真的故障注入技术则是辐射测试的有效替代方法。它支持多种类型的故障模型,实现方式非常灵活。文献[5]提出了一种基于TCL脚本在网表级电路中注入故障的方法,但是软件仿真的时间开销非常大,并不适用于超大规模集成电路(Very large scale integration, VLSI)辐射性能的评估。

现在广泛采用基于FPGA的硬件模拟技术的故障注入方法。FPGA硬件模拟速度很快,能够满足VLSI电路的评估需求,此外FPGA还具有良好的可控性和可观察性。此类方法又分为重配置和旁路注入。重配置首先回读配置存储器的比特文件,然后修改部分配置内容以模拟单粒子翻转,再重新写回配置存储器[6-8]。该方法基于FPGA的内部专用端口实现,没有额外的资源开销。但是重配置功能高度依赖FPGA硬件的支持,比特文件配置FPGA所消耗的时间很容易成为故障注入速度的瓶颈。旁路注入最常用的方式是复用扫描链,其故障注入速度可达到微秒级[9-10],相比重配置快了多个数量级,而且易于实现,可通过脚本直接在原始电路中插入。但复用扫描链是将故障经由扫描链串行移位至目标寄存器,该过程会增加非常明显的时间开销。另一种方式是修改触发器结构[11-12]。文献[11]通过修改FPGA内建库中构成电路的基本单元使电路具备故障注入功能,该方案理论上可达到的故障注入速度最快。但是重建库是十分耗时的过程,需要将原始器件库中的触发器、RAM、逻辑门和乘法器等所有单元进行替换,而且器件库和测试平台是紧密联系的,测试平台的升级或者更替可能导致器件库无法使用,因此技术的迁移性较差;所有寄存器单元还需要添加独占式的故障输入信号加以控制,硬件资源占用率非常高。文献[12]通过对网表中所有的触发器添加额外的组合逻辑和端口来注入故障,故障注入可以即时响应,所以注入速度相比扫描链更快,但该方案修改后的电路中每个触发器需添加3个查找表,额外的资源开销也很大。

本文提出了一种基于旁路注入技术的快速故障注入方法。该方法可通过脚本自动化分析基于硬件描述语言编写的寄存器传输级(Register transfer level, RTL)超大规模集成电路设计,并修改电路结构使之能够从旁路注入故障,结合仿真结果证明脚本可适用于任意规模电路。同时巧用时分复用机制保证脚本修改后的电路的原始功能不受影响。脚本实现电路修改的过程简单,耗时少,修改后电路增加的硬件资源开销相对其他技术方案也较低,适用于VLSI的评估。实验结果证明了其故障注入速度可以达到纳秒级别,具有优越的故障注入速度,在实际应用中具有不错的前景。

1 旁路注入关键技术

旁路注入原理是在原有的设计基础上,通过添加额外的组合逻辑,使待测电路具备故障注入功能。

1.1 旁路注入故障的单触发器结构

原始触发器结构和旁路注入触发器结构分别如图1(a)和图1(b)所示。enable控制电路是否处于故障注入模式:enable处于有效状态时,电路处于故障注入模式,故障数据(Faulty data)会输出到触发器的输入端;否则,enable处于无效状态,电路处于正常工作模式,原始数据(Data)送给触发器采样。

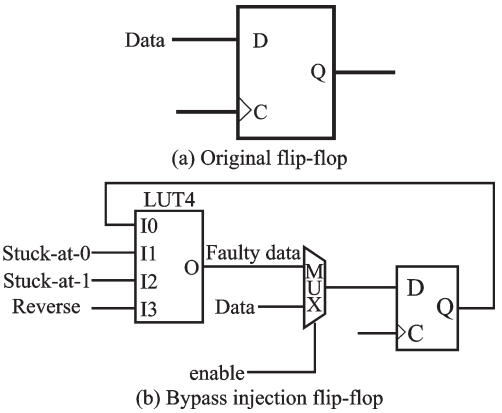


图1 单触发器结构

Fig.1 Structure of flip-flop

故障数据的类型由故障类型选通信号组决定,该信号组有3根信号线,分别是单比特翻转(Reverse)、单比特固定1(Stuck-at-1)和单比特固定0(Stuck-at-0),且在同一时间只有一根信号线有效。为了降低故障数据产生电路的资源开销,本文巧妙地利用了四输入查找表(LUT4)的内部结构,将正常数据和故障类型选通信号作为查找表的输入,并将对应故障类型数据作为输出,因此只需1个查找表即可完成故障数据的产生。

1.2 任意触发器故障注入流程

VLSI电路中触发器数目众多,需要逐个标识,以实现对指定的任意触发器进行故障注入操作。任意触发器故障注入流程如图2所示。首先遍历电路所有模块中的触发器,然后依次分配唯一的ID编号。故障注入时,首先根据ID确定待注入故障的触发器,使能该触发器的enable,然后选择故障类型产生错误数据,在触发器采样时注入故障数据。其余未匹配的触发器仍旧正常工作,不受故障注入过程的影响。此外,通过该故障注入流程可以对任意电路进行任意位置的故障注入,不受电路规模限制。

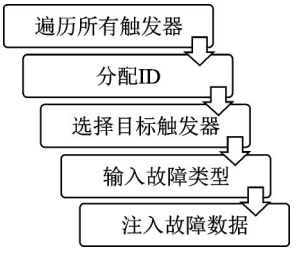


图2 任意触发器故障注入流程

Fig.2 Fault injection process of arbitrary flip-flop

1.3 时分复用机制

由于触发器只在边沿采样数据,对时序有很高的要求,这给故障注入带来挑战。此外,故障注入时也不能影响处于正常工作状态的触发器。为了确保能够稳定可靠的注入故障,本设计采用时分复用机制。将时间划分为故障注入时间clk_div1和正常运作时间clk_div2。在clk_div1下执行故障注入;在clk_div2下执行电路的原始逻辑。

图3所示的是时分复用机制下,单比特翻转故障的注入过程。t_1时刻ID匹配触发器,enable变为有效状态;t_2时刻故障类型输入为reverse,因此触发器的输出发生了翻转。此外,若clk为系统设定频率的两倍,还可保持电路原有的数据吞吐率。

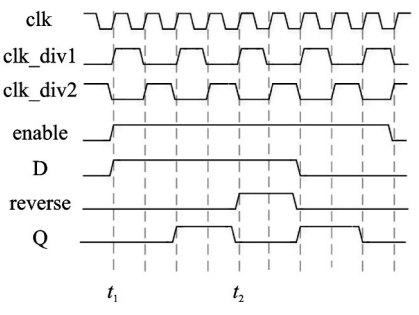


图3 时分复用机制

Fig.3 Time division multiplexing mechanism

2 自动化处理脚本设计

本章介绍如何通过脚本编程,对RTL级的设计代码进行再处理,实现旁路注入故障相关操作。所有工作均通过脚本自动化执行,电路分析时间开销小,且不受电路规模的限制。

2.1 提取触发器的脚本设计

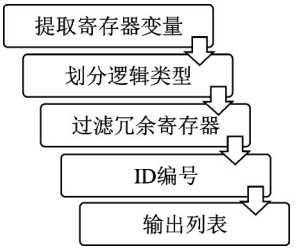


图4 提取触发器的脚本流程

Fig.4 Script flow of flip-flop extraction

提取触发器的脚本流程分析过程如图4所示。首先检索待测电路中所有的触发器,检索依据是所有触发器均需要基于关键字reg进行定义。检索后进一步文本处理可以得到触发器的名称和其对应的位宽。其次是过滤逻辑类型。SEU影响的是电路的时序逻辑,所以只需处理时序逻辑的语句块。检索依据是基于时钟边沿的关键字(posedge和negedge)。然后从触发器队列中移除非时序逻辑的触发器。最后还需要过滤冗余触发器,通常包括定义后却未使用的触发器。处理后的触发器列表再根据触发器位宽分配一定的ID范围,再加上控制时钟、所属模块等相关信息即可输出完整正确的触发器列表。

2.2 旁路注入逻辑的脚本设计

旁路注入逻辑的脚本伪代码如算法1所示。首先遍历所有的时序逻辑块,检索每个逻辑块中的触发器变量以及分配的ID范围,然后将表1中第1行到第6行的旁路注入相关逻辑插入到对应时序逻辑块中,将原始逻辑放在第7行else的分支中。最后,在输入输出端口声明处添加ID和故障类型选通信号作为输入端口。

算法1 旁路注入逻辑伪代码

(1) if ID matched then

// 触发器ID匹配

(2) if reverse then // 单比特翻转

(3) D ← ~D

//触发器的数值翻转

(4) else if Stuck-at-0 then //单比特固定0

(5) D ← 0

//触发器数值变为0

(6) else D ← 1

//触发器数值变为1

(7) else D ← D //原始逻辑

2.3 时分复用逻辑的脚本设计

单比特触发器加入旁路注入逻辑比较简单,可以直接通过if-else的分支语句实现。但对多比特触发器而言,通过分支语句对多比特触发器某一位注入故障会导致该触发器的其他位也无法正常工作。为此,本文巧妙地设计时分复用机制,在时间上错开故障注入过程和电路正常工作过程。时分复用逻辑伪代码如算法2所示。

算法 2 时分复用逻辑伪代码

(1) if (clk_div1 ==1)then //故障注入时间

(2) 故障注入

(3) else 执行正常逻辑 //正常运作时间

进一步设计电路来验证脚本功能的正确性,选择 1 位寄存器来实现产生方波信号的电路。将设计的 HDL 模型通过脚本处理,处理前后的逻辑如图 5 所示。

图 5(b)中的 err_data 代表故障数据,可通过组合逻辑实现。组合逻辑的表达式为

$$Y = X\overline{A}\,\overline{C} + \overline{X}\overline{A}\,\overline{B}C + \overline{X}\overline{A}B\overline{C}$$

式中:Y 表示 2.3 输出的故障数据;X 表示当前触发器的数值;A、B、C 分别代表 3 类故障(单比特固定 0,单比特固定 1 和单比特翻转)。逻辑功能和门级电路的设计相吻合,在 ID 匹配时对目标触发器注入故障,且在任意时刻只能有一种故障类型有效。

```
always@(posedge clk)
begin
    if(!rst)
    begin
        cnt <= 'd0;
    end
    else if(cnt == 'd1)
    begin
        cnt <= 'd0;
    end
    else
    begin
        cnt <= cnt + 1;
    end
end
```

(a) Original logic

```
always@(posedge clk)
begin
    if(clk_div1 == 1)
    begin
        if(id == 32'd1)
        begin
            cnt <= err_data;
        end
        else
        begin
            ;
        end
    end
    else
        if(!rst)
        begin
            cnt <= 'd0;
        end
        else if(cnt == 'd1)
        begin
            cnt <= 'd0;
        end
        else
        begin
            cnt <= cnt + 1;
        end
end
```

(b) Modified logic

图 5 脚本插入故障逻辑前后对比

Fig.5 Comparison of the original logic and the modified logic

3 实验结果

3.1 仿真验证

为验证脚本功能的正确性,本文选取经过脚本处理后的 4 比特循环计数器作为待测电路进行测试,从低位到高位依次分配 ID(1~4)。对 ID 为 1 的触发器 cnt[0] 在某一时刻注入一次故障,以单比特翻转 reverse 为例,仿真波形结果如图 6 所示。

从图 6 可见,当 ID 为 0 时,此时无触发器被选中,电路处于正常工作状态;当 ID 为 1 时,cnt[0]被选为故障注入对象,该触发器进入故障注入模式。随后在 clk 的上升沿,检测到故障类型输入 reverse 有效,此时对触发器注入一次故障,使得 cnt 的计数值从 7 跳变为了 6。进一步观察,可发现计数器值发生改变的原因是 cnt[0]的值从 1 变为了 0,而 cnt 的高三位仍保持正常状态。由此可验证脚本的自动化流程可以使电路具备正确的故障注入功能。

3.2 板级测试

为进一步验证本文提出的设计方案对于复杂电路系统的适用性,采用经典的 MIPS 32 位处理器的测试电路进行实测[13]。测试平台为 Xilinx 的 Zynq[14] xc7z035ffg676-2 SoC。整个测试系统的框架如图 7 所示。

首先由 PC 配置故障参数给 Zynq 的 PS 端,参数主要包括 ID 和故障类型。然后 PS 端根据参数产生故障向量并下发给 PL 端的故障注入控制器,其中故障向量不仅包含了 PC 配置的基本参数,还包含了执行次数和故障注入时间等信息。实验时参考设计和待测设计同时执行一个运算程序,故障注入控制器在执行过程中对待测设计注入故障,随后由故障监控器比较二者的运算结果,如果结果不一致则判

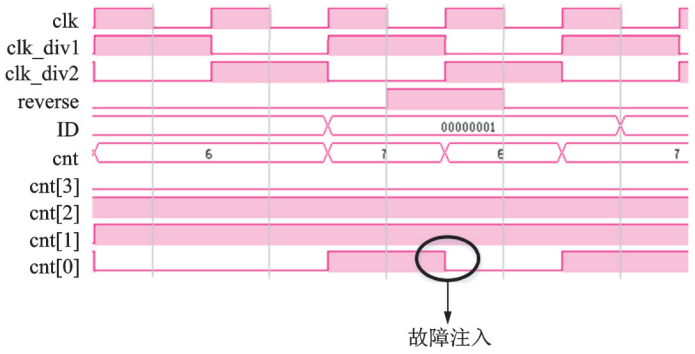


图6 故障注入仿真

Fig.6 Fault injection simulation

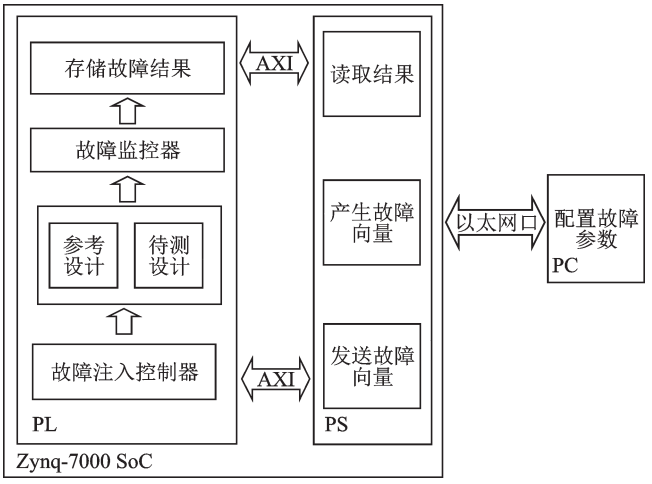


图7 系统框架

Fig.7 System framework

定错误发生,并将结果保存到存储器,最后由PS读取故障结果数据。

实验从资源开销和速度两方面来评估性能。在资源开销方面,处理器主要分为取指、译码、执行、存储和写回5个模块,故只需对这5个模块的原始资源和修改后的资源开销进行对比,具体如表1所示。

可见修改后电路总共消耗的LUT增加了69.8%,触发器的开销几乎没有增加。写回模块不含有触发器,脚本处理前后没有变化。而文献[9]使用扫描链处理类似规模电路(触发器数目为418,LUT为2 322)时,触发器的开销增加130.1%~360.8%,LUT增加的开销为42.8%~237.9%;文献[10]的扫描链在处理电路(触发器数目为530,LUT为1 543)时,触发器和LUT增加的开销分别为33.9%和29.1%;文献[11]采用修改内置库的方式处理电路(触发器数目为3 991,LUT为3 482)增加了1.5倍的LUT资源;文献[12]使用旁路注入处理小规模电路(触发器数目为119,LUT为362)增加了143.1%的LUT。比较后可发现本文方案的资源开销相比大多数方案较小,对测试平台的要求较低。

考虑到工艺的发展以及电路规模的增大,以一定的面积开销来换取快速故障注入的实现是可以接

表1 原始逻辑和修改后的逻辑资源开销对比

Table 1 Comparison of resource cost between the original and modified logics

模块	Flip-flop(触发器)			LUTs(查找表)		
	修改前	修改后	增加开销/%	修改前	修改后	增加开销/%
处理器核	1 712	1 743	1.8	2 767	4 698	69.8
取指模块	100	102	2	989	1 161	17.4
译码模块	1 194	1 202	0.67	666	1 376	106.6
执行模块	206	209	1.5	510	909	78.2
存储模块	212	230	8.5	586	1 236	110.9
写回模块	0	0	0	16	16	0

受的。

在速度性能方面,实验设定为每个时钟周期都对选定的某个触发器注入故障,可计算系统的最快故障注入速度。测试覆盖全部1 712个触发器,参考设计工作频率为100 MHz,待测设计由于时分复用需将频率设置为200 MHz。测试结果如表2所示,其中模拟时间包括电路执行程序,测试向量的配置和下发以及数据分析的全部开销。虽然PS平均每次生成和发送测试向量产生了5.96 μs左右的开销,但故障注入速度依然达到6.8 ns/故障,非常接近系统的工作频率。其中生成每个测试向量所需时间为定值,如果待测电路的执行时间越长,系统等待新测试向量的开销就更小,故障注入速度也会越快。

表2 故障注入测试结果

Table 2 Fault injection test results

参数	数值
测试向量个数	17 120
执行时间/cycle	5 000
故障注入个数	85 600 000
故障次数	68 163 280
故障率/%	79.63
模拟时间/s	12
向量生成/μs	5.96
故障注入速度/(ns·故障$^{-1}$)	6.8

进一步比较本文所提及的其他故障注入技术,对速度进行对比,结果如表3所示。

表3 速度性能对比

Table 3 Speed performance comparison

方案	注入方式	速度	时钟频率/ MHz	速度/时钟周期
文献[9]	扫描链	8.25 μs/故障	25	206
文献[10]	扫描链	3.85 ms/故障	100	385 000
文献[11]	修改内置库	1 000 μs/故障	10	10 000
文献[12]	旁路注入	10 μs/故障	100	1 000
本文方案	旁路注入	6.8 ns/故障	200	1.36

由于各类方案的待测电路规模、执行时间和时钟频率各不相同,表3中新增了故障注入速度(归一化为ns)和时钟频率(ns)的比值以便于直观比较,比值越小说明故障注入速度和电路的工作频率越接近,相对更快。文献[9,10]虽然注入方式一致,但文献[9]优化了故障注入机制,减少了主机和测试平台的交互,因此具备速度优势;文献[10]则通过串口传输故障激励,造成时间开销太大。基于扫描链的

方式相对于本文方案在速度性能上仍旧存在明显差距。文献[11]的故障注入方式虽然较快,但测试平台中包含两块FPGA,其中一块用于和主机的数据通信和故障检测,另一块用于实现故障电路,二者通过接口相连,导致整体数据链路过长,造成速度不理想。同理,文献[12]虽然采取和本文同样的实现方式,但故障激励的传输开销太大,导致速度偏慢。本文通过网口传输故障配置参数,PS自动生成故障激励,减少了和主机的通信频率,并通过AXI总线传递给PL,显著缩短了故障向量的传输开销,降低了系统的等待时间。数据结果表明,本文技术方案的故障注入速度相比现有的方案高了2~3个数量级。

4 结束语

本文提出了一种适用于VLSI电路的快速故障注入方法,通过脚本自动完成电路分析和旁路注入故障逻辑的修改。最后选取复杂的微处理器设计验证性能,实验结果表明,本方法故障注入速度可以达到10^8个故障/s,可满足超大规模集成电路的单粒子效应故障注入的模拟需求。

参考文献:

[1] DIXIT A, WOOD A. The impact of new technology on soft error rates[C]//Proceedings of 2011 International Reliability Physics Symposium. [S.l.]: IEEE, 2011. DOI: 10.11 09/IRPS. 2011.5784522.

[2] 代海林, 袁伟峰, 贺云,等. 基于增量式PID算法和逆变调功的外层空间温度环境模拟系统设计[J]. 数据采集与处理, 2019, 34(4): 744-752.
DAI Hailin, YUAN Weifeng, HE Yun, et al. Design of temperature environment simulation system for outer space based on PID control method and inverter power adjusting technology[J]. Journal of Data Acquisition and Processing, 2019, 34(4): 744-752.

[3] 张敏, 孟令军. 关于中子辐射的单粒子翻转效应测试与加固研究[J]. 电子器件, 2019, 42(6): 1365-1370.
ZHANG Min, MENG Lingjun. Research on testing and reinforcement of single event upset effect of neutron radiation[J]. Electronic Device, 2019, 42(6): 1365-1370.

[4] DU B, STERPONE L, AZIMI S, et al. Ultrahigh energy heavy ion test beam on Xilinx Kintex-7 SRAM-based FPGA[J]. IEEE Transactions on Nuclear Science, 2019, 66(7): 1813-1819.

[5] NIDHIN T S, BHATTACHARYYA A, BEHERA R P, et al. Verification of fault tolerant techniques in finite state machines using simulation based fault injection targeted at FPGAs for SEU mitigation[C]//Proceedings of 2017 4th International Conference on Electronics and Communication Systems (ICECS). [S.l.]: IEEE, 2017: 153-157.

[6] 潘雄, 邓威, 苑政国,等. SRAM型FPGA单粒子随机故障注入模拟与评估[J]. 微电子学计算机, 2018, 35(7): 23-27.
PAN Xiong, DENG Wei, YUAN Zhengguo, et al. SRAM FPGA single event random fault injection simulation and evaluation [J]. Microelectronic Computer, 2018, 35(7): 23-27.

[7] 朱启, 赖晓玲, 巨艇,等. 基于SRAM型FPGA可重构技术的故障注入系统设计[J]. 空间电子技术, 2017(5): 22-26.
ZHU Qi, LAI Xiaoling, JU Ting, et al. Design of fault injection system based on reconfigurable technology of SRAM FPGA [J]. Space Electronic Technology, 2017(5): 22-26.

[8] 芦浩. 基于动态重配置的航空总线单粒子翻转效应测试系统研究[D]. 天津:中国民航大学, 2019.
LU Hao. Research on aero bus single event upset effect test system based on dynamic reconfiguration[D]. Tianjin: Civil Aviation University of China, 2019.

[9] LOPEZ-ONGIL C, GARCIA-VALDERAS M, PORTELA-GARCIA M, et al. Autonomous fault emulation: A new FPGA-based acceleration system for hardness evaluation[J]. IEEE Transactions on Nuclear Science, 2007, 54(1): 252-261.

[10] 徐松. 基于FPGA的集成电路故障注入攻击硬件模拟方法研究[D]. 天津:天津大学, 2016.

XU Song. Research on hardware simulation method of IC fault injection attack based on FPGA[D]. Tianjin: Tianjin University, 2016.

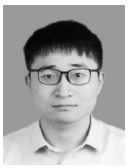
[11] MANSOUR W, VELAZCO R. An automated SEU fault-injection method and tool for HDL-based designs[J]. IEEE Transactions on Nuclear Science, 2013, 60(4): 2728-2733.

[12] SERRANO F, CLEMENTE J A, MECHA H. A methodology to emulate single event upsets in flip-flops using FPGAs through partial reconfiguration and instrumentation[J]. IEEE Transactions on Nuclear Science, 2015, 62(4): 1617-1624.

[13] JIFANG Jin. Classic 5-stage pipeline MIPS[EB/OL]. (2017-02-28). https://opencores.org/projects/mips32.

[14] 吴志忠, 邓敏, 李毅航, 等.4K视频流异构多核的多路分屏方法[J]. 数据采集与处理, 2020, 35(5): 1001-1010.
WU Zhizhong, DENG Min, LI Yihang, et al. Heterogeneous multi-core method of multi-channel split screen transmission for 4K video stream[J]. Journal of Data Acquisition and Processing, 2020, 35(5): 1001-1010.

作者简介:



施聿哲(1996-),**通信作者**,男,硕士,研究方向:数字集成电路设计,E-mail: yuzheshi @nuaa.edu.cn。

陈鑫(1982-),男,副教授,博士,研究方向:数字集成电路设计,E-mail: xin_chen@ nuaa.edu.cn。

陈凯(1997-),男,硕士研究生,研究方向:集成电路数字设计。

白雨鑫(1999-),女,硕士研究生,研究方向:集成电路数字设计。

张颖(1977-),女,讲师,博士,研究方向:集成电路设计与测试硬件安全。

(编辑:王静)