

# 基于 SRAM 型 FPGA 单粒子效应的 故障传播模型

吴 珊, 周学功, 王伶俐

(复旦大学专用集成电路与系统国家重点实验室, 上海 201203)

**摘 要:** SRAM 型 FPGA 在辐射环境中易受到单粒子翻转的影响, 造成电路功能失效. 本文基于图论和元胞自动机模型, 提出了一种针对 SRAM 型 FPGA 单粒子效应的电路故障传播模型. 本文将单粒子翻转分为单位翻转和多位翻转来研究, 因为多位翻转模型还涉及到了冲突处理的问题. 本文主要改进了耦合度的计算方式, 通过计算 FPGA 布局布线中的相关配置位, 从而使得仿真的电路故障传播模型更接近于实际电路码点翻转的结果, 与以往只计算 LUT 相关配置位的方法比较, 平均优化程度为 19.89%. 最后阐述了本模型在故障防御方面的一些应用, 如找出最易导致故障扩散的元胞.

**关键词:** 现场可编程门阵列; 单粒子翻转; 单位翻转; 多位翻转; 电路故障传播模型

**中图分类号:** TN47 **文献标识码:** A **文章编号:** 0372-2112 (2017)08-1976-09

**电子学报 URL:** <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2017.08.024

## SRAM-Based FPGA SEU Fault Propagation Model

WU Shan, ZHOU Xue-gong, WANG Ling-li

(State Key Laboratory of ASIC and System, Fudan University, Shanghai 201203, China)

**Abstract:** SRAM-based FPGA (Field Programmable Gate Array) could be affected by SEU (Single Event Upset) in radiation environment, causing circuit function failures. This paper proposes an SEU-induced fault propagation model for FPGAs based on the graph theory and the cellular automata. In addition, this paper divides SEU effect into two parts: SBU (Single-Bit Upset) and MBU (Multi-Bit Upset) because MBU has an extra problem about conflict management. The core part of this model is the computing method of the coupling degree which is based upon relative bits from FPGA placement and routing results to make the model more accurate. After validation between fault propagation model and fault injection experiment, this fault propagation model is 19.89% more relative to fault injection experiment than the fault propagation model only counting relative bits in LUT (Look Up Table). Finally, this paper analyses an application about this model to find cells most easily leading to fault diffusion.

**Key words:** field programmable gate array (FPGA); single event upset (SEU); single-bit upset (SBU); multi-bit upset (MBU); fault propagation model

## 1 引言

基于静态随机存储器(Static Random Access Memory, SRAM)的现场可编程门阵列(Field Programmable Gate Array, FPGA)芯片具有开发成本低、开发时间短、上市时间快等优点, 目前被广泛应用于医疗、通信、航空航天等领域. 而太空的辐射环境中存在着各种各样的高能粒子, 随着半导体制造工艺的进步、特征尺寸不断

缩小, SRAM 型 FPGA 产生单粒子翻转(Single Event Upset, SEU)效应越来越明显. 单粒子效应会造成 FPGA 可编程码点的翻转, 很可能影响到电路的功能; 但也可能只是翻转了无关码点, 对电路功能不造成直接影响. 电路故障如何传播主要取决于电路各子模块之间的耦合作用, 耦合的紧密程度直接导致了故障是否扩散传播. 因此, 针对 SRAM 型 FPGA 的特性, 通过故障注入模拟单粒子翻转效应, 并分析故障传播过程, 建立更为贴切

的故障传播模型至关重要。

关于单粒子效应的研究是近几年的热点。随着微电子技术的发展,集成电路的特征工艺尺寸得以大幅度地缩减,一位翻转 SBU (Single-Bit Upset)效应的概率逐渐降低,多位翻转 (Multi-Bit Upset, MBU)效应有着上升趋势<sup>[1-4]</sup>。当芯片受到辐射时,同时触发多个码点发生翻转称为 MBU。文献[5]通过辐照各种不同的重离子,统计不同位数 MBU 的占比。实验发现,翻转位数越高发生的概率越低。其中, SBU 占比超过 50%,这说明 SBU 现象还是占据了主导地位;2 位翻转占比约为 30%,因此 SEU 主要还是引起一位翻转及两位翻转;随着入射粒子的线性能量传输值 (Linear Energy Transfer, LET)增大,出现的多位翻转情况越多,然而当器件的翻转横截面 (Device Cross Section, DCS) 处于饱和状态之后,翻转的百分比就基本不再变化了。文献[5]中最多测试到了 8 位翻转。

在故障传播模型的研究方面,最经典的是基于图论的故障诊断模型。很多模型都是以图论为基础发展演化而来的。图论分析的主要方法是把电路的各子模块抽象为图的结点,把子模块之间的故障传播关系抽象为两结点的有向边。并且,基于故障传播构建的有向图往往是加权有向图,权重可以根据结点间的故障传播概率或是故障传播时间确定<sup>[6-10]</sup>。图论的方法一般只能用于解决单故障问题,系统复杂时难以转化为有向图来求解,但基本思想还是值得借鉴的。文献[11-15]提出了基于元胞自动机 (Cellular Automata, CA) 的电路故障传播模型。元胞自动机模型的应用非常广泛,几乎涉及社会和自然科学的各个领域,包括生物学、社会学、物理学、计算机科学等。在电路故障研究方面,主要是用来仿真故障覆盖率,从而针对扰动值、耦合度、传播概率等影响因子来提出降低故障覆盖率的解决方案。然而这是个较为通用的模型,需要加以改进才能体现出故障传播在不同电路的不同表现。文献[14]将基于元胞自动机的电路故障传播模型用在了 FPGA 上,然而最关键的参数——耦合度却没有很准确地体现。文献[16]采用基于 Petri 网的故障传播模型来分析多路径的故障传播问题,在大规模系统中找到所有故障可能传播的路径,从而提出提高系统可靠性的解决方案。Petri 网涉及到的传播关系非常复杂,往往求解困难。文献[17,18]研究的是基于复杂网络的电路故障传播模型,其中基于小世界网络模型最为经典。小世界网络是指介于规则网络和随机网络之间的网络,若用在故障传播的问题上,则是反应了故障的扩散程度,再结合蚁群算法就可以求得扩散能力最强的故障传播路径,为设计改进和故障防御提供了重要依据。小世界网络对系统的特征路径长度和聚类系数有要求,并非所有系统

都能转化为小世界网络。

为了测试 SEU 效应,多采用辐照实验的方法。但真实辐照实验成本高、并且很有可能会损坏待测器件,因此国内外普遍采用模拟故障注入的方法来测试 SEU 效应。通过配置故障注入后的部分位流的方法<sup>[19]</sup>来刷新电路功能,从而大大提高测试效率,进行快速错误注入。文献[20]根据不同种类可编程点具有不同单粒子翻转敏感性,建立了基于权重的故障注入模型,并通过实验数据得出,基于权重的故障注入模型下可编程码点的翻转比率与辐射实验结果相吻合。因此本文在基于权重的故障注入模型下进行模拟单粒子翻转效应的故障注入实验。

本文的模型是以 SRAM 型 FPGA 发生的 SEU 效应为研究对象,综合图论和元胞自动机模型进行改进,使模型适用于不同的电路设计。借鉴元胞自动机模型中元胞的概念,本文的模型主要改进了演化规则,比如不同电路的元胞排布不一样、不同电路的元胞连接关系不一样、不同电路非空元胞的耦合关系不一样等。FPGA 是具有一定规模的 Tile 阵列,Tile 就可以看作是元胞。FPGA 各 Tile 的连接均靠布线的开关盒 (General Switch Block, GSB)<sup>[21]</sup>来实现,因此与耦合度最相关的是布线信息,这也就成为了计算耦合度的重点来源。本文的研究是基于实验室自主研发的 FDP3P7 可编程 FPGA 芯片<sup>[22]</sup>,可通过解析布局布线后的网表文件来提取关键信息。模型仿真之后再用实验室自主研发的 FPGA 故障注入平台进行测试,在不同电路中进行验证,从而得出了更贴近 SRAM 型 FPGA 实际电路的故障传播模型。

## 2 故障传播模型的建立

设元胞空间的大小为  $M \times N$ 。非空元胞赋值为 1,空元胞赋值为 0。某元胞的位置表示为:

$$c = (i, j), i = 1, 2, \dots, M, j = 1, 2, \dots, N$$

借鉴元胞自动机模型,我们把相邻元胞的连接关系称为邻居关系。如图 1 所示,图 1(a)为元胞自动机模型中的 Von Neumann 邻居模型 (或称为 5 邻居模型)<sup>[13]</sup>,该模型中,中心元胞的邻居即为上下左右四个元胞。图 1(b)为本文的改进模型,设定相连的元胞互为邻居,突破了传统二维空间的限制。由于 FPGA 布线很

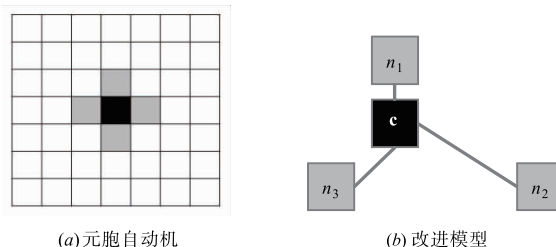


图1 元胞自动机模型与改进模型的元胞连接关系对比

多时候会跨越多个 Tile, 因此邻居关系更为复杂. 如图 1(b), 中心元胞为  $c$ , 设元胞  $c$  的邻居分别为元胞  $n_1, n_2, \dots, n_q$ , 其中  $q$  为元胞  $c$  的邻居总数, 设  $1 \leq k \leq q$ .

故障传播模型的演化规则如下:

$$G_{t+1}(c) = \frac{x_{t+1}(n_1)G_t(n_1) + x_{t+1}(n_2)G_t(n_2) + \dots + x_{t+1}(n_q)G_t(n_q) + G_t(i,j)}{x_{t+1}(n_1) + x_{t+1}(n_2) + \dots + x_{t+1}(n_q) + 1} \quad (2)$$

$$G_{t+1}(c) = \frac{x_{t+1}(n_1)G_t(n_1) + x_{t+1}(n_2)G_t(n_2) + \dots + x_{t+1}(n_q)G_t(n_q) + G_t(i,j)}{x_{t+1}(n_1) + x_{t+1}(n_2) + \dots + x_{t+1}(n_q) + 1} + R \quad (3)$$

$$x_{t+1}(n_k) = \begin{cases} 1, & (E_{cnk} > e) \cap (O_t(n_k) = 1) \\ 0, & (E_{cnk} \leq e) \cup (O_t(n_k) \neq 1) \end{cases} \quad (4)$$

式中:

(1)  $O_{t+1}(c)$  表示  $t+1$  时刻元胞  $c$  的输出状态, 输出状态有 1、0、-1 三种. 状态为 1 时表示元胞正常运行; 状态为 -1 时表示元胞发生故障; 状态为 0 表示该元胞为空, 即该电路设计没有用到这个元胞.

(2)  $G_{t+1}(c)$  表示  $t+1$  时刻元胞  $c$  的故障程度值, 仿真初始化时需设置, 其值越大, 下一时刻对该元胞及其邻居元胞的影响就越大. 故障程度阈值为  $\delta$ , 若故障程度大于等于该阈值则元胞输出状态为 -1.

(3)  $R$  表示  $t$  时刻外部加入的故障扰动, 仿真时需设置, 其值越大, 则表示注入的故障扰动越大, 下一时刻对该元胞及其邻居元胞的影响越大. 一般设置  $R > 1$ , 这样才能保证起始元胞发生故障.

(4)  $x_{t+1}(n_k)$  表示  $t+1$  时刻元胞  $c$  与邻居  $n_k$  的关系. 若  $t$  时刻元胞  $c$  处于正常运行状态 (即输出状态为 1)、并且元胞  $c$  与元胞  $k$  的耦合度  $E_{cnk}$  大于阈值  $e$ , 则对应的  $x_{t+1}(n_k)$  取值为 1, 否则取值为 0. 耦合度值在仿真初始化时需设置, 只有大于阈值时, 才可能对邻居元胞有影响.

### 3 故障传播模型中耦合度的计算

本文建立模型的核心就是耦合度的计算问题. FPGA 中占比最大的资源是布线, 因此考虑好布线因素的影响最为重要. 文献[14]中是用相关配置位数来计算耦合度的, 主要是考虑 LUT (Look Up Table) 的个数, 相对于通过真实布局布线来计算耦合度而言确实简便很多, 然而只考虑 LUT 明显不够全面, 因此最好能找到方法利用布局布线信息统计出该电路设计中用到的所有相关配置位.

本文是通过解析布局布线后的网表文件来逐步计算出耦合度. 解析布局布线后的网表文件时, 因为布线网表包含了布局信息, 因此实际操作中只需要解析布线网表就可以了. 由于网表文件中只会出现关于该电路设计的信息, 因此解析出来的结果都是电路相关的, 如连接关系、相关配置位等. 本文模型中耦合度的计算流程如图 2 所示.

在此以 16 位乘法器的网表解析为例来说明耦合

$$O_{t+1}(c) = \begin{cases} 1, & 0 \leq G_{t+1}(c) < \delta \\ 0, & \\ -1, & G_{t+1}(c) \geq \delta \end{cases} \quad (1)$$

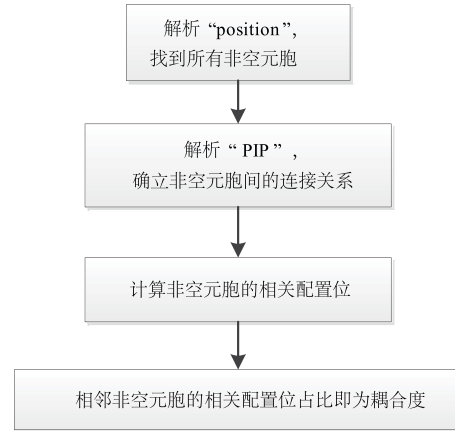


图2 故障传播模型中耦合度的计算流程

度的具体计算过程, 图 3 为布线后网表文件的截图, 网表文件是通过实验室自主研发的 FDP3P7 芯片系列开发软件 FDE 生成<sup>[23]</sup>.

“position”的信息从“net”模块中提取, 如图 3(a) 所示, 从而得到了所有非空元胞的坐标.

“PIP”的信息表示出了连接关系. 依据“PIP”的描述, 元胞依次相连. 比如图 3(a) 中就是 Tile(12,22) 连接到 Tile(12,16), Tile(12,16) 连接到 Tile(19,16), 以此类推. 同时也要注意标记连接顺序, 这决定了故障从哪个元胞出发、传播到哪个元胞.

相关配置位的信息来源于四个部分: slice 内部互连、LUT、IO、slice 外部互连. slice 内部互连及 LUT 的相关配置位信息来源于“slice”模块, 如图 3(b) 所示. 比如 <config name = "BXMUX" value = "#OFF"/> 这一行对应的就是一个 SRAM: BXMUX, 算作 1 个配置位; 比如 <config name = "F" value = "#LUT;D=(A2\*A1)"/> 和 <config name = "G" value = "#LUT;D=(A2\*A1)"/>, F 和 G 表示的都是 LUT, LUT 的配置位为 16. IO 的相关配置位信息来源于“iob”模块, 与“slice”结构类似. slice 外部互连的相关配置位信息来源于“net”模块. 如图 3(a), 从“PIP”信息中可以得到某坐标 Tile 中用了几个 PIP, 1 个 PIP 为一个配置位.

接下来, 计算耦合度. 如图 1(b), 若下一时刻  $t+1$ , 元胞  $c$  要将自身的故障传给一个邻居, 则会从与其相连的元胞中随机选择一个来进行故障传播. 设函数 bit()

```
<net name="net_Lut-U386_0_0">
  <portRef name="X" instanceRef="iSlice_393_"/>
  <portRef name="G3" instanceRef="iSlice_238_"/>
  <portRef name="G1" instanceRef="iSlice_242_"/>
  <pip from="S0_X" to="OUT5" position="12,22" dir="-&gt;"/>
  <pip from="OUT5" to="H6W2" position="12,22" dir="-&gt;"/>
  <pip from="H6E2" to="V6S3" position="12,16" dir="-&gt;"/>
  <pip from="V6N3" to="V6S3" position="19,16" dir="-&gt;"/>
  <pip from="V6S3" to="N19" position="19,16" dir="-&gt;"/>
  <pip from="S19" to="S_P19" position="18,16" dir="-&gt;"/>
  <pip from="S_P19" to="S1_G_B3" position="18,16" dir="-&gt;"/>
  <pip from="S0_X" to="OUT0" position="12,22" dir="-&gt;"/>
  <pip from="OUT0" to="V6N0" position="12,22" dir="-&gt;"/>
  <pip from="V6M0" to="W3" position="9,22" dir="-&gt;"/>
  <pip from="E3" to="W3" position="9,21" dir="-&gt;"/>
  <pip from="E3" to="S1" position="9,20" dir="-&gt;"/>
  <pip from="N1" to="N_P1" position="10,20" dir="-&gt;"/>
  <pip from="N_P1" to="S1_G_B1" position="10,20" dir="-&gt;"/>
</net>

(a) 某线网的相关配置位信息

<instance name="iSlice_323_" moduleRef="slice" libraryRef="template_work_lib">
  <property name="position" type="point" value="14,35,0"/>
  <config name="BXMUX" value="#OFF"/>
  <config name="BYMUX" value="#OFF"/>
  <config name="CEMUX" value="#OFF"/>
  <config name="CKINV" value="#OFF"/>
  <config name="COUTUSED" value="#OFF"/>
  <config name="CY0F" value="#OFF"/>
  <config name="CY0G" value="#OFF"/>
  <config name="CYINIT" value="#OFF"/>
  <config name="CYSELF" value="#OFF"/>
  <config name="CYSELG" value="#OFF"/>
  <config name="DXMUX" value="#OFF"/>
  <config name="DYMUX" value="#OFF"/>
  <config name="F" value="#LUT:D=(A2*A1)"/>
  <config name="F5USED" value="#OFF"/>
  <config name="FFX" value="#OFF"/>
  <config name="FFY" value="#OFF"/>
  <config name="FXMUX" value="F"/>
  <config name="G" value="#LUT:D=(A2*A1)"/>
  <config name="GYMUX" value="G"/>
  <config name="INITX" value="#OFF"/>
  <config name="INITY" value="#OFF"/>
  <config name="RAMCONFIG" value="#OFF"/>
  <config name="REVUSED" value="#OFF"/>
  <config name="SRFFMUX" value="#OFF"/>
  <config name="SRMUX" value="#OFF"/>
  <config name="SYNC_ATTR" value="#OFF"/>
  <config name="XBUSED" value="#OFF"/>
  <config name="XUSED" value="0"/>
  <config name="YBMUX" value="#OFF"/>
  <config name="YUSED" value="0"/>
</instance>
```

(b) 某slice的相关配置位信息

图3 16位乘法器布线后网表文件包含的相关配置位信息

用来计算某 Tile 一共用到的相关配置位数,  $E_{cnk}$  表示元胞  $c$  与元胞  $n_k$  的耦合度, 则

$$E_{cnk} = \frac{\text{bit}(n_k)}{\text{bit}(n_k) + \text{bit}(c)} \times \omega \quad (5)$$

其中  $\omega$  为仿真时调节耦合度大小的系数,用来提高仿真效率.耦合度的含义实际上就是指  $c$  的故障会传递多少给它的邻居.  $G_c(c)$  表示  $t$  时刻元胞  $c$  的故障程度值,那么下一时刻元胞  $n_i$  的故障程度值

$$G_{t+1}(n_k) = G_t(c) \times E_{cnk} \quad (6)$$

#### 4 故障传播模型的仿真流程

### 4.1 一位翻转的仿真流程

先来阐述一位翻转的故障传播过程,模型仿真流

程如图 4 所示. 其中, Count 用来统计第几次注入故障时,故障能够传播到 IO.

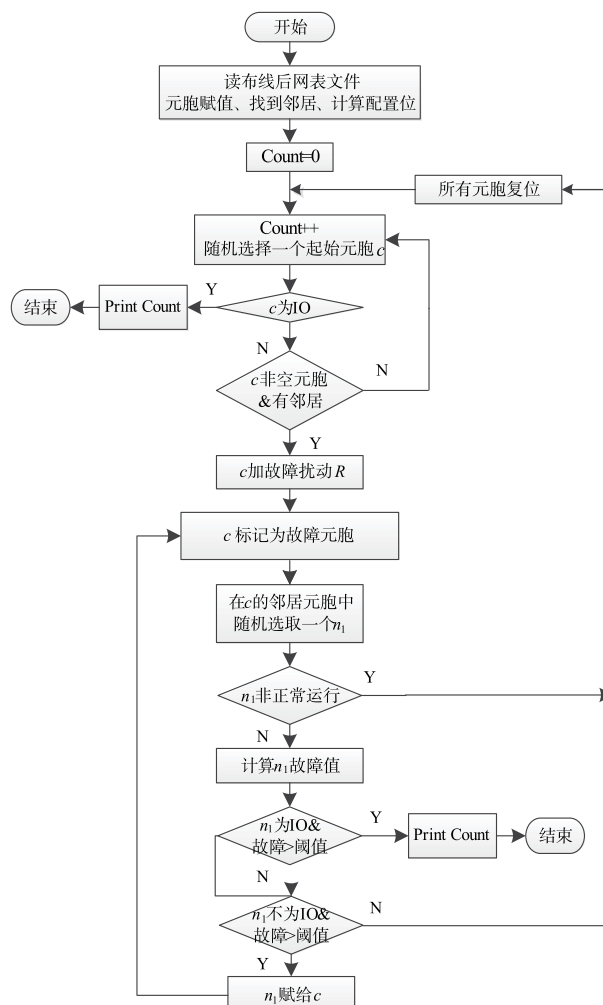


图4 一位翻转仿真流程图

**初始化** 用 python 解析 xml 文件(布局布线后的网表文件), 与该电路设计有关的元胞标记为 1、其余为 0。接下来故障传播涉及到的元胞都只是非空元胞。解析 PIP 信息得到非空元胞之间的连接关系, 通过统计布局加布线的相关配置位数来计算两个相邻非空元胞的耦合度, Count = 0。

**选择起始点** 初始化之后  $\text{Count}++$ , 随机选择起始元胞, 若该元胞非空、并且不是 IO (认为故障传播路径搜索到 IO 即结束)、并且有邻居元胞 (即故障有机会传播下去), 此时则为选定了起始点, 否则重新生成随机点。选择好起始元胞后, 该元胞的故障值加上扰动  $R$ 。在之后的故障传播路径搜索中, 若不满足形成路径的条件, 则所有元胞的值复位、返回来重新换一个起始点。

**选择邻居** 有了第一个有故障的元胞  $c$  之后,把  $c$  标记为故障元胞,即元胞值由 0 变为  $-1$ ,  $c$  的相关配置



位数清零,与 $c$ 为邻的所有元胞的邻居总个数减1,元胞值为-1后即当做空元胞处理,不再加入到任何故障传播路径当中。接下来在 $c$ 点的邻居中随机选择一个点 $n_1$ ,如果 $n_1$ 的元胞值不是1,或者元胞值为1但邻居为0,则所有元胞复位重新选择起始点。

**故障传播** 选定了邻居 $n_1$ 之后,计算出 $n_1$ 的故障程度值,如果元胞 $n_1$ 为IO并且故障程度值大于阈值,说明故障已经传播到IO了,该路径成立,打印出Count值,程序结束;如果元胞 $n_1$ 不为IO并且故障程度值大于阈值,说明故障从原来的元胞 $c$ 已经传播到了元胞 $n_1$ ,把 $n_1$ 的属性赋给 $c$ ,回到标记 $c$ 为故障元胞的位置,继续循环;剩下的情况就是元胞 $n_1$ 的故障程度值小于阈值,说明故障没有传播出去,那么所有元胞复位重新选择起始点。

## 4.2 多位翻转的仿真流程

多位翻转是在一位翻转的基础上加一层循环,如图5所示。翻转位数设为 $F$ ,如果翻转位数大于1,那么先选择一个符合故障传播条件的起始点 $C_1$ ,在以 $C_1$ 为顶点的矩形框内随机选择 $F-1$ 个点,接下来就分别进行每一条路径的故障传播路径搜索,流程同图4。若有一条路径成立,则结束,返回Count;若 $F$ 条路径都不成立,则重新选择起始点。其中,同时搜索 $F$ 条路径时,故障元胞累加,因此故障传播效率会更高。

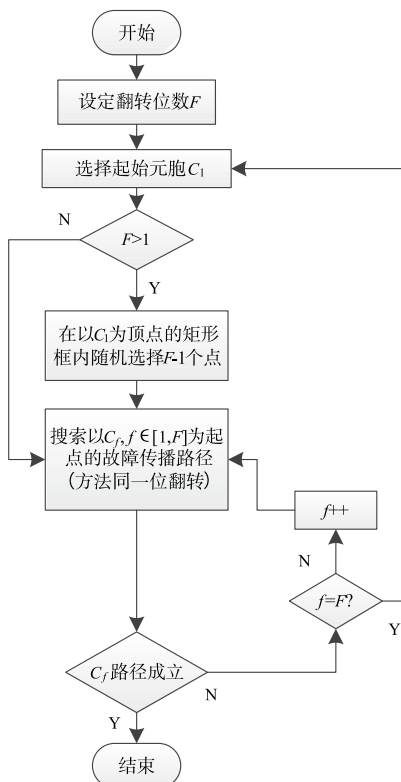


图5 多位翻转仿真流程图

此外,在处理多路径传播的问题中,涉及到冲突处理的问题。因为本文模型中,元胞间的故障传播允许出现多对一的情况。针对冲突问题,解决的原则可以概括为先到先得。冲突处理流程如图6所示。此时为了简化流程来进行说明,假设故障均能传播到下一个元胞、故障程度值均大于阈值。

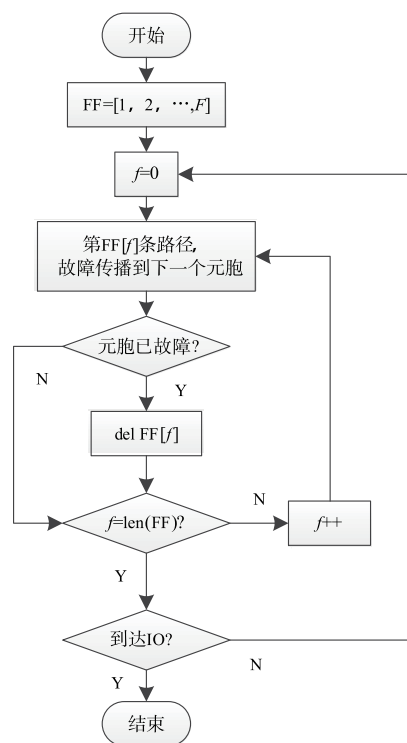


图6 冲突处理流程图

以三位翻转为例进行具体说明。如图7所示,假设 $t=0$ 时,有 $a_0$ 、 $b_0$ 、 $c_0$ 三位翻转; $t=1$ 时,故障开始向相邻的一个元胞传播,此时出现了 $a_0$ 和 $b_0$ 找到的邻居元胞为同一元胞的情况,这时假定以 $a$ 、 $b$ 、 $c$ 为序来进行处理,也就是说先是 $a_0$ 选择了邻居 $a_1$ ,并把故障传播给了 $a_1$ ,所以接下来 $b_0$ 找到邻居 $b_1$ 的时候,检测到 $b_1$ 已经是故障元胞了,也就是说 $b$ 的传播路径无效了; $t=2$ 时, $a$ 和 $c$ 依然正常传播故障, $b$ 路径不再考虑; $t=3$ 时,出现了 $a_2$ 的邻居找到了 $c_2$ 的情况,此时 $c_2$ 已经成为故障元胞了,那么 $a$ 的传播路径也无效了, $c$ 受影响继续传播; $t=4$ 时,就只剩 $c$ 一条路径正常传播了。到最后,若 $c$ 故障传播到了IO,则 $c$ 路径成立;若 $c$ 故障没能传播成功,则Count加1,重新选择起始三个元胞。

## 5 仿真与模拟测试结果

### 5.1 基于FDP3P7的故障注入模拟平台

故障注入测试是基于实验室自主研发的FDP3P7芯片及其模拟故障注入平台,如图8所示。由USB口将

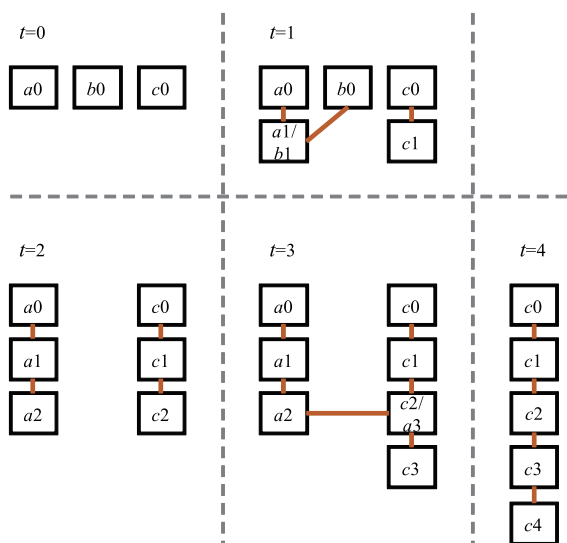


图7 冲突处理示例

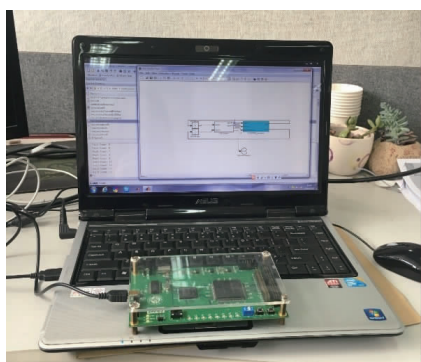


图8 模拟故障注入平台

位流下载到板子上,模拟 SEU 时,每次下载位流仅改变部分位流的配置,有效缩短了配置时间,从而实现快速模拟故障注入,并能快速检测出电路敏感点. 这种方法不但能更准确地模拟 FPGA 的抗辐射性能,而且大大提高了故障注入实验效率.

对于某个电路设计,首先用 Synopsys 公司的 Design Compiler 中的 DFT Compiler 对用户电路(.v)进行 DFT 综合,即插入扫描链. 接下来用 Synopsys 公司的 Tetramax 工具,对 Design Compiler 生成的电路产生相应的测试向量,用于测试电路功能. 通过自主研发的 FDE 软件,得到电路设计的位流文件. 使用 Matlab Simulink 搭建故障注入系统,如图 9 所示,其中 S-Function 是 Simulink 提供给用户实现自定义的模块,包含了:(1)测试激励产生;(2)待测电路响应分析这两个功能. 而 VLFDP3P7\_template 则作为一个抽象的待测电路存在于本系统中. 另外,Unit Delay 模块是为了使该系统不至于成为一个无延时的环路而使得 Simulink 无法对该系统求解仿真. 另外,由于本实验的终止往往是由程序逻辑决定的,也就是说,仿真的时间实际上未知,因此

需要用 STOP 模块来控制仿真的结束与否. 对于模拟真实辐射环境来进行故障注入,我们所观察的结果是打翻多少次码点之后,测到电路结果出错,这与本文的故障传播模型中所观察的 Count 值含义一致. 模拟 MBU 也是采用随机选取码点进行翻转的方法,当翻转位数大于 1 时,也是采用类似仿真模型中的划定矩形框的方法来选择要打翻的码点.

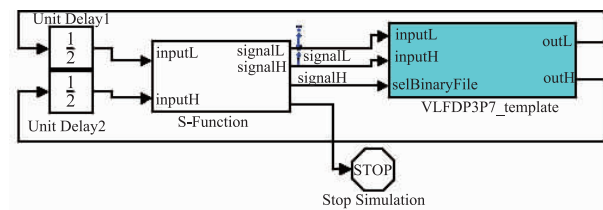


图9 Matlab Simulink 搭建的故障注入系统

## 5.2 改进模型的模拟测试结果

对于 FDP3P7 芯片,元胞即为 Tile,则元胞空间大小为  $35 \times 55$ . 初始时所有元胞均无故障即  $Gt = 0(i, j) = 0$ ,故障程度阈值设为  $\delta = 1$ ,外部加入的故障扰动  $R = 100$ ,耦合度系数  $\omega = 1$ ,耦合度阈值  $e = 0.1$ . 为了提高实验效率, $R$  或者  $\omega$  的值可以设大些, $\delta$  或者  $e$  可以设小一些.

由文献[5]可知最多能测到 8 位翻转,因此本模型对于 SEU 效应的测试数据为从 1 位翻转到 8 位翻转. 如表 1 所示,我们比较了实际注入、对比模型、本模型三种模型在不同的电路设计中的测试结果. 比较的参数为 Count 值,即注入多少次后故障成功传播到 IO. 实际注入是在打翻码点之后,用测试向量来测试电路功能,若功能无误则恢复之前被翻转的码点、继续打翻下一批码点,若功能有误则程序结束;对比模型只计算 LUT 的相关配置位<sup>[14]</sup>,故障传播算法跟本模型一致,从而可以比较出耦合度计算精确之后对仿真结果的改善. 我们用相关系数来衡量模型效果. 比如 1 位翻转时,先统计出所有 Count 值(取 100 次测试的平均值),然后用 CORREL 函数来计算相关系数,比如对比模型的相关系数指的是采用对比模型所有例子的 Count 值连成的曲线与实际注入测试的 Count 值的曲线的相关系数,相关系数越高,则表示两条曲线越相似,即该模型效果越好. 从表 1 可见,通过多个例子、多位翻转的比较,基于耦合度计算方式的优化,本模型对比模型平均优化了 19.89%.

由上述内容可见本模型可以更为有效地仿真 SEU 引起的故障传播. 通过应用该模型,还可以计算出相关配置位数最高的元胞或者是邻居数最高的元胞,从而得到最易导致故障扩散的电路模块,有助于故障预防.

设函数 RelationBit() 用来计算某元胞  $c$  的总相关配置位数,则:

表 1 三种方法测试结果对比

MBU 位数	对比测试	add_bx	add_jl	add4_2	adder16	adder4	adder8	alu	alutask	multi	parity	resource1	相关 系数	优化 百分比
	IOB 个数	28	28	16	52	16	28	29	17	66	12	24		
	SLICE 个数	98	102	50	90	50	46	394	202	3364	42	136		
	PIP 个数	296	307	137	378	138	177	974	460	7404	92	335		
1bit	实际注入	43.89	46.38	69.94	57.42	62.70	56.67	17.55	55.05	59.58	161.21	38.00		
	对比模型	58.93	58.50	90.69	24.60	79.90	50.00	39.61	87.13	26.56	128.35	58.00	0.7351	
	本模型	22.64	21.85	41.89	19.95	42.22	28.26	17.41	34.76	7.86	69.72	27.33	0.8309	13.03%
2bit	实际注入	20.28	23.83	53.83	28.55	41.76	49.83	16.17	25.40	30.05	72.20	24.20		
	对比模型	39.05	35.20	85.57	25.98	78.67	45.57	27.75	82.97	14.37	123.88	37.41	0.7656	
	本模型	19.41	18.24	39.21	17.90	39.25	26.18	14.74	27.35	2.21	60.51	20.77	0.8305	8.48%
3bit	实际注入	19.35	10.14	32.83	24.25	29.07	25.50	14.89	17.33	17.26	42.67	23.67		
	对比模型	33.67	36.36	74.77	19.84	77.95	41.66	25.95	80.15	7.84	90.29	37.78	0.6447	
	本模型	18.85	16.11	40.16	13.38	35.56	23.96	13.59	22.16	2.48	58.43	19.69	0.8714	35.16%
4bit	实际注入	15.52	9.85	28.75	21.60	27.20	22.79	12.03	15.83	14.91	33.40	24.09		
	对比模型	30.45	32.61	73.64	20.81	71.61	37.82	19.80	81.29	7.75	89.27	32.68	0.6399	
	本模型	16.45	15.42	39.88	12.63	34.53	23.49	13.22	21.06	3.01	50.68	20.20	0.8428	31.71%
5bit	实际注入	16.67	5.80	27.83	17.15	28.33	21.50	7.58	15.75	12.25	32.25	23.75		
	对比模型	25.49	31.25	75.23	18.08	73.44	38.38	16.47	73.71	6.49	91.97	28.90	0.7251	
	本模型	17.65	16.01	35.22	12.71	33.49	22.85	10.59	20.89	2.50	47.33	19.69	0.8468	16.78%
6bit	实际注入	16.50	2.67	30.14	12.11	24.68	19.67	7.49	16.00	12.90	32.99	17.25		
	对比模型	23.30	29.41	69.67	15.35	60.71	38.12	16.94	37.68	5.30	55.45	21.04	0.7802	
	本模型	15.42	12.75	31.30	11.86	30.05	21.92	9.23	19.25	1.88	44.45	20.82	0.8730	11.89%
7bit	实际注入	12.75	3.25	24.25	13.25	21.33	16.89	6.01	13.67	11.18	33.50	15.80		
	对比模型	22.99	29.50	64.36	15.49	54.19	36.19	16.92	36.84	5.20	56.00	22.32	0.7483	
	本模型	15.29	12.82	28.48	12.61	28.57	21.24	8.54	20.38	2.25	39.25	20.24	0.8843	18.18%
8bit	实际注入	12.50	3.75	19.00	9.67	15.67	15.59	5.25	12.53	9.38	31.75	14.25		
	对比模型	24.82	28.88	51.74	15.52	59.95	37.97	15.16	33.25	5.60	57.16	22.50	0.7215	
	本模型	15.18	8.75	29.16	10.98	29.32	19.03	8.81	19.65	1.74	40.30	15.61	0.8936	23.85%
平均优化百分比														19.89%

$$\text{RelationBit}(c) = \text{bit}(n_1) + \text{bit}(n_2) + \cdots + \text{bit}(n_q) \quad (7)$$

元胞的总邻居数可通过解析布线后网表文件统计得出。

耦合度的含义是邻居元胞间相关配置位数的占比,因此相关配置位数越高、邻居数越多,则耦合程度越高,越易引起故障扩散。

我们以 16 位乘法器的一位翻转为例,求得所有元胞中,相关配置位数、邻居数排名前五的元胞信息,见表 2。

这些相关配置位数或邻居数较高的元胞可以认为是最易引起故障传播扩散的元胞,从而在这些元胞可

能引发的故障传播路径上进行防御。

表 2 相关位数和邻居数排名前五的元胞信息

	相关配置位数		邻居数	
	元胞坐标	总数	元胞坐标	总数
Top1	0,0	4303	0,0	66
Top2	20,17	3710	10,20	47
Top3	20,16	3620	13,34	44
Top4	19,16	3407	20,15	44
Top5	10,20	3392	11,18	43

## 6 总结

本文针对 SRAM 型 FPGA 的 SEU 效应,研究故障传播模型.本模型以图论和元胞自动机的故障模型为基本思想,针对 SRAM 型 FPGA 的特性加以改进.以实验室自主研发的 FDP3P7 芯片为测试对象,通过解析布局布线网表,不仅得到了更为准确的元胞状态及元胞连接关系,而且可以更加精确地计算出每个 Tile 在电路设计中用到的相关配置位数,使得耦合度的计算结果更贴近实际.测试结果表明,本模型提高了耦合度的精确度,对故障传播模型的改善较为显著,通过将仿真数据与测试数据进行比较,得到相关系数,本模型对比模型平均优化了 19.89%.最后,应用该模型,可求出某电路设计中相关配置位数最高的模块、邻居最多的模块,即某些最易造成故障扩散的模块,从而可以对这些模块进行重点 SEU 防范.

### 参考文献

- [1] W Wu, N Seifert. MBU-Calc: A compact model for Multi-Bit Upset (MBU) SER estimation [A]. IEEE International on Reliability Physics Symposium (IRPS) [C]. Monterey, CA, 2015. SE. 2. 1 – SE. 2. 6.
- [2] C Frenkel, J D Legat, D Bol. A Partial Reconfiguration-based scheme to mitigate Multiple-Bit Upsets for FPGAs in low-cost space applications [A]. 10th International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC) [C]. Bremen, 2015. 1 – 7.
- [3] M Ebrahimi, M B Tahoori. Stepped parity: A low-cost multiple bit upset detection technique [A]. IEEE International on Test Conference (ITC) [C]. Anaheim, CA, 2015. 1 – 8.
- [4] W Wei, K Namba, Y B Kim, F Lombardi. A novel scheme for tolerating Single Event/Multiple Bit Upsets (SE/MBU) in non-volatile memories [J]. IEEE Transactions on Computers, 2016, 65(3): 781 – 790.
- [5] 郭红霞, 罗尹虹, 姚志斌, 等. 亚微米特征工艺尺寸静态随机存储器单粒子效应实验研究 [J]. 原子能科学技术, 2010, 44(12): 1498 – 1504.  
Guo Hongxia, Luo Yinhong, Yao Zhibin et al. Experimental research of SEU and SEL in high density SRAMs with sub-micron feature sizes [J]. Atomic Energy Science and Technology, 2010, 44(12): 1498 – 1504. (in Chinese)
- [6] 吴军强, 梁军. 基于图论的故障诊断技术及其发展 [J]. 机电工程, 2003, 5: 188 – 190.  
Wu Junqiang, Liang Jun. Technology and development of fault diagnosis based on graph theory [J]. Mechanical & Electrical Engineering Magazine, 2003, 5: 188 – 190. (in Chinese)
- [7] Yuan, Haibin, J Li, Q Wang. Research on directed fault propagation graph model and optimization [A]. 9th International Conference on Electronic Measurement & Instruments [C]. Beijing, 2009. 241 – 245.
- [8] Yuan Haibin. Development of simulation model based on directed fault propagation graph [A]. International Conference on Computer Application and System Modeling (IC-CASM) [C]. Taiyuan, 2010. 686 – 690.
- [9] F Yang, D Xiao, S L Shah. Signed directed graph-based hierarchical modelling and fault propagation analysis for large-scale systems [J]. IET Control Theory & Applications, 2013, 7(4): 537 – 550.
- [10] Yan Xiaobo, Wang Yichen, Li Jianxing. Signal-component fault propagation model for embedded software [A]. First International Conference on Reliability Systems Engineering (ICRSE) [C]. Beijing, 2015. 1 – 8.
- [11] M Halbach, R Hoffmann. Implementing cellular automata in FPGA logic [A]. 18th International on Parallel and Distributed Processing Symposium [C]. Santa Fe, New Mexico, 2004. 258 – 262.
- [12] S Murtaza, A G Hoekstra, P M A Sloot. Performance modeling of 2D cellular automata on FPGA [A]. International Conference on Field Programmable Logic and Applications (FPL) [C]. Amsterdam, 2007. 74 – 78.
- [13] 吴继梅. 基于元胞自动机的电路故障传播建模与应用研究 [D]. 上海: 东华大学, 2008.  
Wu Jimei. Modeling and application of circuit fault propagation based on cellular automata [D]. Shanghai: Donghua University, 2008. (in Chinese)
- [14] 何伟. SRAM 型 FPGA 单粒子故障传播特性与测试方法研究 [D]. 长沙: 国防科学技术大学, 2011.  
He Wei. Characteristic and testing methodology of SEE soft error propagation in SRAM-based FPGA [D]. Changsha: National University of Defense Technology, 2011. (in Chinese)
- [15] Yan Xiaobo, Wang Yichen, Zhu Anzhi, Wang Yikun. A fault propagation model for embedded software [A]. IEEE International Conference on Software Quality, Reliability and Security-Companion (QRS-C) [C]. Vancouver, BC: IEEE, 2015. 72 – 79.
- [16] Y Wang, Hao Shi, Limin Jia, Shuai Lin, Lei Guo. Multi paths fault propagation model for network modeled system [A]. 11th World Congress on Intelligent Control and Automation (WCICA) [C]. Shenyang, 2014. 5915 – 5920.
- [17] 李果, 高建民, 高智勇, 姜洪权. 基于小世界网络的复杂系统故障传播模型 [J]. 西安交通大学学报, 2007, 3: 334 – 338.  
Li Guo, Gao Jianmin, Gao Zhiyong, Jiang Hongquan. Fail-



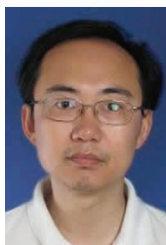
- ure propagation model of complex system based on small world net [J]. Journal of Xi'an Jiaotong University, 2007, 3: 334–338. (in Chinese)
- [18] Jianmin Gao, Guo Li, Zhiyong Gao. Fault propagation analysis for complex system based on small-world network model [A]. RAMS on Reliability and Maintainability Symposium [C]. Las Vegas, NV, 2008. 359–364.
- [19] M M Ibrahim, K Asami, M Cho. Evaluation of SRAM based FPGA performance by simulating SEU through fault injection [A]. 6th International Conference on Recent Advances in Space Technologies (RAST) [C]. Istanbul, 2013. 649–654.
- [20] 刘智斌, 王伶俐, 周学功, 等. 基于动态局部重配置的 FPGA 抗辐射模拟 [J]. 计算机工程, 2010, 36(14): 218–220, 226.
- Liu Zhibin, Wang Lingli, Zhou Xuegong et al. Radioresistance emulation of FPGA based on dynamic partial reconfiguration [J]. Computer Engineering, 2010, 36(14): 218–220, 226. (in Chinese)
- [21] K Ma, L Wang, X Zhou, S X D Tan, J Tong. General switch box modeling and optimization for FPGA routing architectures [A]. International Conference on Field-Programmable Technology (FPT) [C]. Beijing, 2010. 320–323.
- [22] 江峻. FDP FPGA 芯片可编程逻辑单元建模与故障测试 [D]. 上海: 复旦大学, 2010.

- [23] 复旦大学. FPGA/SOPC 后端编译软件 [简称: FDE 软件] V1.0 [CP/CD]. 著作权登记号: 2013SR067117.

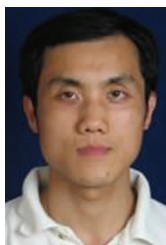
#### 作者简介



吴珊 女, 硕士研究生, 主要研究方向为 FPGA 布线、位流、SEU 故障传播.  
E-mail: yvette53@163.com



周学功 (通信作者) 男, 助理研究员. 2007 年毕业于复旦大学计算机与信息技术系, 获理学博士学位. 主要研究方向为 EDA 算法、可重构嵌入式系统的体系结构与应用开发方法学等.  
E-mail: zhouxg@fudan.edu.cn



王伶俐 男, 教授, 主要研究方向为集成电路与 EDA 算法设计、可重构计算与量子计算.  
E-mail: llwang@fudan.edu.cn