

分类号\_\_\_\_\_

学校代码 10487

学号 M202072959

密级\_\_\_\_\_

华中科技大学

# 硕士学位论文

(学术型☐ 专业型☒)

## 针对 Kintex-7 系列 FPGA 的单粒子翻转定向故障注入方法及其仿真验证

学位申请人：叶俊杰

学 科 专 业：电子信息

指 导 教 师：颜露新 教授

答 辩 日 期：2022 年 06 月 14 日

**A Thesis Submitted in Partial Fulfillment of  
the Requirements  
for the Master Degree in Engineering**

**Single Event Upset Directional Injection Method for  
Kintex-7 FPGA and its Simulation Verification**

**Candidate : Ye Junjie**  
**Major : Electronic information**  
**Supervisor : Prof. Yan Luxin**

**Huazhong University of Science and Technology**  
**Wuhan 430074, P. R. China**  
**June, 2022**

## 摘要

SRAM 型 FPGA 在卫星遥感、深空探测等空间领域应用广泛,受空间复杂环境中的高能粒子辐射影响,SRAM 型 FPGA 易发生单粒子翻转效应 SEU(Single Event Upset),造成 FPGA 中配置 RAM 存储单元发生位翻转,导致系统功能紊乱。动态刷新和三模冗余等抗单粒子翻转技术受到重点关注,单粒子翻转效应定量定向注入是验证抗单粒子技术的关键。现有的地面高能辐射测试、卫星搭载验证等手段成本高,难以实现对指定区域的单粒子翻转注入。为此,本文研究了一种针对 Xilinx Kintex-7 系列 SRAM 型 FPGA 单粒子翻转定向注入方法,将故障注入 FPGA 中的配置位来模拟真实空间环境下发生的 SEU 现象。具体内容如下:

首先,对于 SRAM 型 FPGA,配置 RAM 存储单元决定了用户电路的逻辑功能,针对 Kintex-7 系列 FPGA 器件配置存储器中帧结构未知的问题,为了提取出与配置 RAM 资源对应的配置帧,对配置存储器中的帧物理结构开展研究,提出了解析配置帧结构的方法。通过内部配置接口 ICAP(Internal Configuration Access Port)获取所有的帧地址数据后,按照不同的帧地址类型进一步解析,提取出其中与配置 RAM 资源对应的配置帧。

最后,针对 Kintex-7 系列 FPGA 器件配置 RAM 中配置位数量众多并且非关键位占比过大导致难以实现指定功能模块单粒子翻转定向注入的问题,提出一种基于基本位技术的自动配置内存故障注入的方法。1)从配置比特流文件中提取与指定功能模块相对应的基本位数据文件 EBD(Essential Bits Data);2)将其转换为故障注入地址,通过故障注入器指定特定位置发生位翻转,实现单粒子翻转定向注入。在基于 Kintex-7 FPGA 的某星载图像压缩平台中的帧头帧尾检测模块、帧间压缩模块、组帧模块三个用户电路进行验证,实验结果表明,定向故障注入方法能对指定功能模块完成故障注入,并且有效故障注入率由随机故障注入方法的 10.6%提升至 87.5%。

**关键词:** 单粒子翻转; Kintex-7 FPGA; 故障注入; 配置存储器

## Abstract

SRAM-based FPGAs are widely used in satellite remote sensing, deep space exploration and other space fields. Affected by high-energy particle radiation in complex space environment, SRAM-based FPGAs are prone to SEU(Single Event Upset), which causes bit flip in the configuration RAM memory cells in FPGAs, resulting in system dysfunction. Anti single event upset technology, such as dynamic refresh and triple modular redundancy, has attracted much attention. Quantitative directional injection of single event upset is the key to verify anti single event upset technology. The existing means of ground-based high-energy radiation testing and satellite-mounted verification are costly and make it difficult to achieve single event upset injection into designated areas. Therefore, this paper studies a single event upset directional injection method for Kintex-7 SRAM-based FPGA, where faults are injected into the configuration bits in the FPGA to simulate the SEU phenomenon in a real space environment. The specific contents are as follows:

Firstly, for SRAM-based FPGAs, the configuration RAM memory cell determines the logic function of the user circuit. To tackle the issue of unknown frame structure in the configuration memory, the frame structure of the configuration memory is studied and a method for analysing the configuration frame structure is proposed. After obtaining all the frame address data through the ICAP (Internal Configuration Access Port), the frames are further parsed according to the different frame address types and the configuration frames corresponding to the configuration RAM resources for Kintex-7 series FPGA devices are finally extracted.

Finally, to address the problem of the large number of configuration bits and the large proportion of non-critical bits in the configuration RAM of Kintex-7 series FPGA devices, which makes it difficult to achieve single event upset directional injection of specified functional modules, a method of automatic configuration memory fault injection based on the essential bits technology is proposed. 1) Extract the essential bit data file corresponding to the specified function module from the configuration bitstream file; 2) Convert them into fault injection addresses. Then a bit flip occurs at a specific location designated by the fault injector to achieve single event upset injection. To verify the proposed method in this paper, a series of experiments are carried out in a spaceborne satellite image processing platform

## 硕士学位论文

---

based on Kintex-7 FPGA. Specifically, the tail frame head frame detection module, the interframe compression module, and framing module are tested. The experimental results show that the directional fault injection method can complete the fault injection on the specified function module, fault injection rate increased from 10.6% for the random fault injection method to 87.5% for the directed fault injection method. 87.5%.

**Key words:** SEU; Kintex-7 FPGA; Fault injection; Configuration memory

# 目 录

1 绪论.....	1
1.1 研究背景与意义.....	1
1.2 国内外研究现状.....	2
1.3 本文主要内容.....	7
2 Kintex-7 FPGA 单粒子效应机理分析 .....	9
2.1 Kintex-7 FPGA 基本结构及工作原理 .....	9
2.2 Kintex-7 FPGA 单粒子效应 .....	17
2.3 单粒子翻转对 Kintex-7 FPGA 的故障解析 .....	21
2.4 单粒子翻转故障定向注入技术难点分析 .....	25
2.5 本章小结.....	27
3 基于 ICAP 回读技术的配置帧结构解析方法 .....	28
3.1 Kintex-7 FPGA 配置存储空间分析 .....	28
3.2 基于 ICAP 回读技术的帧结构解析方案 .....	31
3.3 ICAP 读写控制模块设计 .....	33
3.4 帧地址解析模块设计.....	42
3.5 本章小结.....	47
4 基于基本位的 FPGA 单粒子翻转定向注入技术 .....	48
4.1 Kintex-7 FPGA 用户电路基本位提取 .....	48
4.2 自动配置内存故障注入.....	53
4.3 定向故障注入测试及结果分析 .....	58
4.4 定向故障注入技术局限性分析 .....	65
4.5 本章小结.....	66
5 总结与展望.....	68
5.1 本文工作总结.....	68
5.2 未来工作展望.....	69
参考文献.....	70

## 1 绪论

### 1.1 研究背景与意义

FPGA 自诞生之日起, 凭借其高逻辑密度、可重复配置以及高性能等特点, 逐渐成为集成电路应用市场上的主流产品, 其低功耗、高密度的优点, 使其得到了航天航空电子系统的广泛应用, 特别是涉及小型卫星的空间任务, FPGA 常被用于缩减航天系统的总成本<sup>[1]</sup>。另外, FPGA 还具备重新配置的功能, 当卫星已被成功部署后, 根据用户的具体需求, 通过地面发送新的配置文件, 动态修改设备的程序实现功能的更新<sup>[2]</sup>。例如, 基于 SRAM 型的 FPGA 芯片价格低、功耗少、处理速度快, 并且支持反复重新配置, 凭借这些优势 SRAM 型 FPGA 很快占据上述低成本空间应用领域的主要份额<sup>[3]</sup>。FPGA 芯片实现逻辑电路功能的原理是: 通过将用户电路的特定位流文件烧写进配置存储区域中来实现逻辑功能。然而, 由于 SRAM 型 FPGA 自身的特殊结构, 以及恶劣的空间环境, 使得其对空间、临近空间至航空高度环境中的辐射效应异常敏感, 持续暴露在高能粒子环境下, 极易发生单粒子效应 (SEE), 包括单粒子翻转 (SEU)、单粒子功能中断 (SEFI)、单粒子门锁 (SEL) 等, 引起的单粒子效应类型主要取决于高能粒子影响的器件部位<sup>[4-6]</sup>。单粒子效应将会改变 FPGA 存储单元中的内容, 发生计算结构失效、程序执行序列紊乱等故障, 进而导致 FPGA 的功能失效, 甚至损坏器件, 严重影响系统的安全与稳定<sup>[7,8]</sup>。因此, 对应用于航天航空电子系统的 SRAM 型 FPGA 必须进行加固来提高系统的可靠性。

基于 SRAM 型 FPGA 结构主要由静态存储单元构成, 通过静态存储单元对查找表、可编程输入输出资源、BRAM 资源完成的具体操作进行定义, 从而实现对 FPGA 的重新配置。经过几十年的发展, SRAM 型 FPGA 的特征尺寸不断减小, 其资源利用率与性能也随之显著提高, 与此同时, 单个芯片上集成的晶体管数量也在与日俱增, 庞大的晶体管数量与过低的设备电压导致集成电路器件单粒子效应发生概率呈指数级增长<sup>[9]</sup>。伴随着特征尺寸的减小, SRAM 型 FPGA 普遍采用

最前沿的制造工艺来生产，从而才能满足高性能和低功耗的需求。对于本文研究的 Kintex-7 系列 FPGA，采用 28nm 制造工艺，内核电压为 1.0V。在集成度如此高的小型器件结构中，高能辐射粒子导致的单粒子效应已不仅仅局限于单个比特位的翻转(Single Bit Upset, SBU)，排列紧密的多个存储单元都有可能受到影响，引起多位翻转(Multiple Bit Upset, MBU)<sup>[10-12]</sup>。因此，持续暴露在恶劣环境中的高集成度 SRAM 型 FPGA 引发的单粒子翻转效应已经到了不可忽视的地步。

基于 SRAM 型 FPGA 芯片实现的完整用户电路具备多种多样的功能，由于各部分功能使用的逻辑资源在类型与数量上存在明显差异，因此，由高能粒子辐射引起的单粒子翻转效应概率也不同，如果全局性地对整个电路系统进行抗辐射加固，将会给原本就资源紧张的星载 FPGA 带来严重的负担，所以根据系统中各部分功能的重要等级，针对性地对优先级更高的功能进行加固十分有必要。本文旨在研究一种针对 Kintex-7 系列 FPGA 的定向故障注入方法，其意义在于：

- (1) 通过对 FPGA 配置存储器中的配置帧进行结构解析可以推动比特流文件到电路功能的映射解析工作；
- (2) 模拟星载 FPGA 在真实空间环境中由高能粒子辐射引起的单粒子翻转效应进而评估用户电路的故障率与敏感性，针对重要等级更高以及单粒子翻转敏感度更高的模块进行抗辐射加固。

## 1.2 国内外研究现状

随着航天事业的高速发展，SRAM 型 FPGA 在航天航空领域得到越来越广泛的应用，作为宇航级器件，由于单粒子效应对其产生的重要影响，目前国内外包括美国国家航天局实验室、罗斯阿拉莫国家实验室等重要科研机构都对单粒子效应故障注入技术以及 SRAM 型 FPGA 对单粒子效应的敏感性开展了深入且广泛的研究。

### 1.2.1 单粒子翻转效应验证技术研究现状

空间领域中的电子设备由于处于不同的轨道中，因此受到不同程度的空间辐射影响，但都可以归结为两类：单粒子效应与总剂量效应(Total Ionizing



Dose,TID)<sup>[13-15]</sup>。总剂量效应是一种长时间积淀的累积效应，而单粒子效应是由单个高能粒子辐射引发的短期效应，从时间角度出发，SRAM 型 FPGA 在应用时更加关注短时间内引起的辐射影响，因此空间辐射效应的主要研究对象为单粒子效应。单粒子效应按照定义与易发生的位置，具体可以划分为单粒子翻转(Single Event Upset, SEU)、单粒子瞬态(Single Event Transient, SET)、单粒子闩锁(Single Event Latchup, SEL)等等。其中单粒子翻转是引发 SRAM 型 FPGA 故障的主要原因，因此本文主要对单粒子效应中的单粒子翻转展开研究。目前国内外研究机构对 FPGA 单粒子翻转效应评估的方法常用的有三种：地面辐射测试<sup>[16,17]</sup>、卫星搭载<sup>[18,19]</sup>以及模拟故障注入<sup>[20-22]</sup>。

### (1) 地面辐射测试

目前最常用的评估单粒子翻转效应的方法是地面辐射测试，该方法是利用粒子加速器对各种粒子进行加速之后撞击目标器件，实现在地面模拟空间辐射环境中的集成器件产生单粒子翻转效应，从而进一步进行测试。因为地面辐射测试得到的结果接近真实空间辐射环境下的结果，准确性更高，因此美国国家航空航天局(National Aeronautics and Space Administration,NASA)每年都要通过此方法来对航天器件进行筛选，并会将测试结果提供给相关科研人员。包括 Actel、赛灵思在内的几家大型 FPGA 生产商也会利用地面辐射测试的方法对自己的产品做相关的测试，并提供相应的测试报告。然而，地面辐射测试的成本过高、测试难度较大、易对被测器件造成损害等缺点也限制了它的应用范围。

### (2) 卫星搭载

卫星搭载是指将搭载有 SRAM 型 FPGA 器件的载体通过人造卫星发射至太空中，直接在真实的空间辐射环境中对目标器件的单粒子翻转效应敏感性进行测试。作为单粒子翻转效应最直接的评估方法，卫星搭载具有显著的优势：真实的太空辐射环境下，目标器件的测试数据也是最真实的。同时，此方法的缺点也是显而易见的：高昂的成本，漫长的测试周期、较差的可控性以及低水平的测试灵活性。

### (3) 模拟故障注入

模拟故障注入主要通过建立模型的方式来模拟单粒子翻转进而测试目标器件，

目前的模拟故障注入技术主要分为三种：故障仿真、静态故障分析以及故障注入。故障仿真是通过硬件描述语言来对某个简单的系统建立仿真模型，以此来模拟系统在不同的测试条件之下不同功能的特性，但是，复杂的模型建立过程以及较弱的可操作性，使其应用频率逐渐降低；静态故障分析主要应用于对模拟设计的敏感性测试上，其也是利用模型来对 FPGA 设计的敏感程度进行分析，具有较高的准确性；故障注入作为目前最普遍的模拟故障注入方法<sup>[23]</sup>，诞生于 20 世纪 70 年代，从验证系统的容错性逐步过渡到集成领域的研究，其主要方式是在目标器件的电路中注入特定的故障，通过对电路的输出进行分析，从而得到定量、定性的评估结果。

相比于前两种方法，故障注入技术投入的成本更低，可控性更强，因此使用的范围也更加广泛，受到各大高校和科研院所的亲睐，因此本论文将采用故障注入的方法对 SRAM 型 FPGA 的单粒子翻转效应进行研究。

### 1.2.2 故障注入技术研究现状

故障注入的方法多种多样，按照技术实现方式的不同，故障注入技术主要分为三类：软件故障注入、高能粒子辐射故障注入和硬件故障注入。

#### (1) 软件故障注入

软件故障注入通过对物理原型建立具体的故障模型后，改变目标器件的存储器以及寄存器内容，从而模拟器件发生软件或硬件故障。Na.J.W 的团队提出利用模拟器修改技术将单粒子效应故障注入目标器件的方法<sup>[24]</sup>；Nimara.S 等人通过注入命令脚本的方式来实现单粒子翻转故障注入<sup>[25]</sup>。国防科技大学通过故障脚本对电路进行故障注入<sup>[26]</sup>。软件故障注入投入的成本小、便于收集信息，并且不会损坏被测器件，然而该方法严重依赖于故障模型，可移植性差。

#### (2) 高能粒子辐射故障注入

高能粒子辐射故障注入通过粒子加速器对高能粒子进行加速后撞击目标器件，从而模拟空间辐射环境中的集成器件产生单粒子翻转效应的现象。Johnson.E 等人利用加速器对 XC2V1000FPGA 进行故障注入<sup>[27]</sup>；Swift.G.M 等人对 XQR2V6000FPGA 进行故障注入<sup>[28]</sup>；George.J 等人对赛灵思公司生产的 Virtex IV

系列 FPGA 进行故障注入<sup>[29]</sup>。高能粒子辐射故障注入的优势在于其高效性，但巨大的投入成本与繁复的操作过程都限制了对高能粒子辐射故障注入的研究。

### (3) 硬件故障注入

硬件故障注入是目前研究最为广泛的故障注入方法，其采取的手段主要通过物理层面来实现，例如，直接在目标器件的管脚施加电压或电磁干扰等方式将故障注入硬件结构，再对其进一步分析得到结果。硬件故障注入不需要根据实际的集成电路建立具体的故障注入模型，也能将故障注入目标器件，因此节省了大量的时间，然而，硬件故障注入需要投入较高的成本以及复杂的设备支持，并且会对目标器件造成损害。

国外对硬件故障注入的研究起步较早，Sterphone.L, Boragno.L 等人<sup>[30]</sup>基于硬件平台模拟空间辐射环境中引起的 SEU，利用图 1-1 所示的单粒子翻转故障注入工具接收 bit 文件、elf 文件、仿真周期数、和单粒子翻转故障注入次数，进而生成故障电路的 bit 文件、XMD 脚本和 Teraterm 日志文件，最后烧录至 FPGA 中完成单粒子翻转故障注入。txt 文件记录电路进行故障注入 FFT 运算的过程以及结果，比较 FFT 的运算结果与输出结果，分析评估故障注入电路受到单粒子翻转影响而产生的功能错误率。

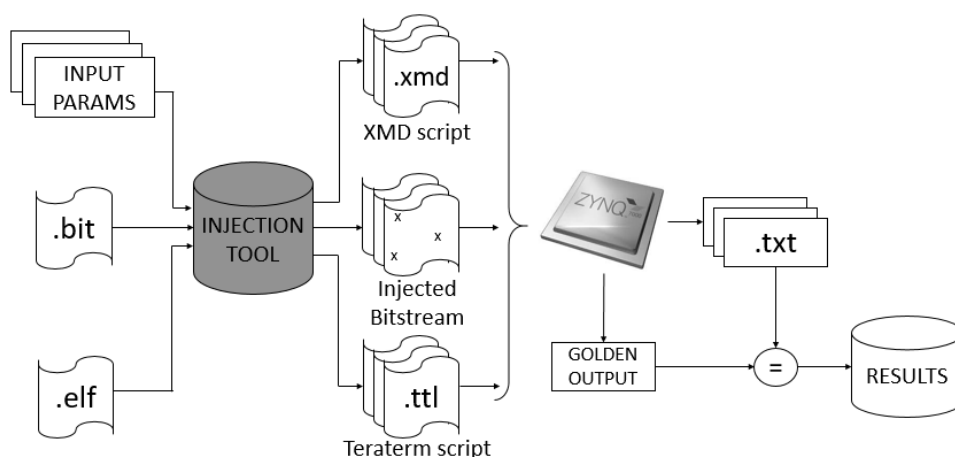


图 1-1 单粒子翻转故障注入工具<sup>[30]</sup>

Zhu.M, Song.N 等人<sup>[31]</sup>提出一种故障注入系统，此系统通过 FPGA 电路节点中的“0”和“1”来建立名为“FPGA-PTM”的概率转移矩阵模型，利用该模型来对节点功能的配置位数进行控制，然后通过 ISE 软件生成逻辑功能篡改后的用

户电路的比特流文件，将其与原始比特流文件进行比对，最终得出结果。

文献<sup>[32]</sup>将赛灵思公司应用广泛的 Virtex II 系列 FPGA 作为研究对象，提出如图 1-2 所示的单粒子效应测试方案。上位机通过 RS422 接口从控制 FPGA 接收回读文件与测试结果，控制 FPGA 完成对被测 FPGA 的配置、回读以及上传回读数据的任务。该测试方案主要完成对单粒子翻转的静态测试与动态测试，实现对时序逻辑电路中 BRAM 存储信息翻转情况的评估以及目标器件的单粒子翻转敏感性测试。

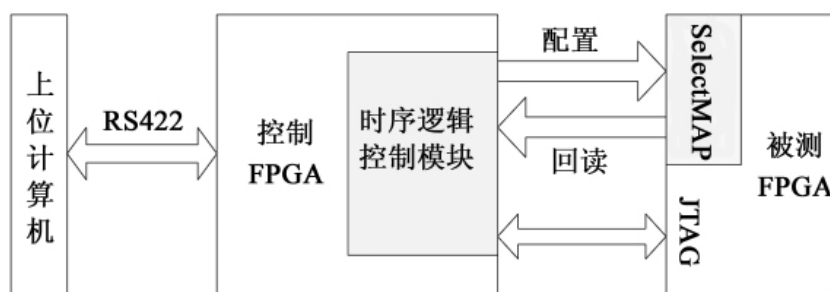


图 1-2 单粒子效应测试方案<sup>[32]</sup>

文献<sup>[33]</sup>中利用 ICAP 接口对 FPGA 配置寄存器进行回读，将回读的帧数据位翻转后回写，从而实现对 FPGA 中用户电路注入单粒子翻转故障，通过搭建 Virtex-6 FPGA 单粒子翻转故障注入平台，实现对 Virtex-6 FPGA 用户电路连续和多次注入单粒子翻转故障，进而准确评估用户电路在 Virtex-6 FPGA 上运行时对单粒子翻转的敏感性。但是通过这种方法只能随机打翻帧数据中的配置位，对于翻转后的配置帧数据产生的后果未知，无法有目的地针对用户电路中某个功能或者某个模块进行定向地故障注入。

文献<sup>[34]</sup>介绍了一种基于定向故障注入的 SRAM 型 FPGA 单粒子翻转效应评估方法。借助 XDL 工具，该方法解析了 Virtex-4SX55 型 FPGA 的帧地址与物理资源之间的对应关系；将电路网表中的资源按模块分组，利用部分重构技术分别对电路整体及各分组相关的配置帧进行随机故障注入，以评估电路整体及其子模块的抗单粒子翻转能力。该方法同样只是对 FPGA 进行随机单粒子故障注入，但是其通过解析帧地址与逻辑资源之间对应关系的想法值得本文借鉴，本文在此基础上进一步探究了帧地址与具体用户电路功能模块之间的关系，从而实现对指定模块

的单粒子翻转定向故障注入。

## 1.3 本文主要内容

针对 Kintex-7 系列 FPGA 实现故障定向注入主要存在“配置帧结构未知”“配置帧数量庞大”以及“难以实现指定部位的故障注入”三个难点。针对以上难点，本文从以下几个方面展开研究：

本论文分为 5 个章节，主要内容如下：

第 1 章为绪论。本章介绍了单粒子翻转定向故障注入方法研究的背景以及意义；比较了目前常用的单粒子翻转效应评估技术的优缺点，确定将故障注入方法作为本文的单粒子翻转评估方法；通过分析比较国内外故障注入方法研究的现状，确定本文实现单粒子翻转定向故障注入的思路。

第 2 章为 Kintex-7 系列 FPGA 单粒子翻转故障注入技术研究。首先介绍了 Kintex-7 系列 FPGA 的基本结构和工作原理；然后对常见的几种单粒子效应类型进行了分析和比较，明确了本文主要针对最普遍和最主要的单粒子翻转展开研究；以单粒子翻转为切入点，解析其对 Kintex-7 系列 FPGA 中各逻辑资源造成的故障，分析其产生的影响和后果；最后对单粒子翻转实现定向故障注入的难点进行了分析。

第 3 章针对 SRAM 型 FPGA 配置存储器中的配置帧展开研究。首先介绍了 FPGA 配置存储结构中的配置帧对用户设计电路的具体功能具有决定性的意义；针对 Kintex-7 系列 FPGA 配置帧结构未知的问题，本文通过 ICAP 接口回读的方式，读取 FPGA 中配置存储器中的配置帧数据，再进一步解析帧结构，分离出其中对应配置 RAM 资源的配置帧。

第 4 章为针对 Kintex-7 系列 FPGA 单粒子翻转定向故障注入的具体实现方法。在获取配置比特流中配置 RAM 所在的配置帧数据后，为进一步解决 Kintex-7 系列 FPGA 帧地址数据数量庞大的问题，借助基本位技术解析、分离其中感兴趣区域的帧数据，最后提出一种自动配置内存故障注入方法，将其翻译成故障注入器能够识别的帧地址的方法。通过对星载 FPGA 芯片中特定的图像处理功能进行

## 硕士学位论文

---

单粒子翻转故障注入，并通过对比使用此方法与不使用此方法的情景下单粒子翻转的故障率，评估方法的可行性。

第 5 章对本文的主要内容进行总结，并对未来基于本文结果进行的研究进行展望。

## 2 Kintex-7 FPGA 单粒子效应机理分析

持续暴露在空间辐射环境中，由于受到不同程度的高能粒子影响，航天设备上搭载的 SRAM 型 FPGA 器件极易发生各种类型的单粒子效应。本章将以 Kintex-7 系列 FPGA 为研究对象，探究 Kintex-7 系列 FPGA 的基本结构与工作原理，并通过分析 Kintex-7 系列 FPGA 的主要逻辑资源，对其实现逻辑功能的过程进行详细阐述，然后研究 FPGA 芯片产生不同类型单粒子效应的机理，并着重解析单粒子翻转对 FPGA 中不同资源模块引起的负面影响，最后对 Kintex-7 系列 FPGA 单粒子翻转故障定向注入的难点进行分析。

### 2.1 Kintex-7 FPGA 基本结构及工作原理

作为赛灵思公司采用二十八纳米制造工艺生产的高性能 SRAM 型 FPGA，Kintex-7 系列 FPGA 能够保证设计此节点实现最佳成本、性能、功耗平衡，相比于赛灵思公司生产的前几代 Kintex 系列 FPGA，其具备更强的性能、更低的成本以及更低的能耗<sup>[35]</sup>。

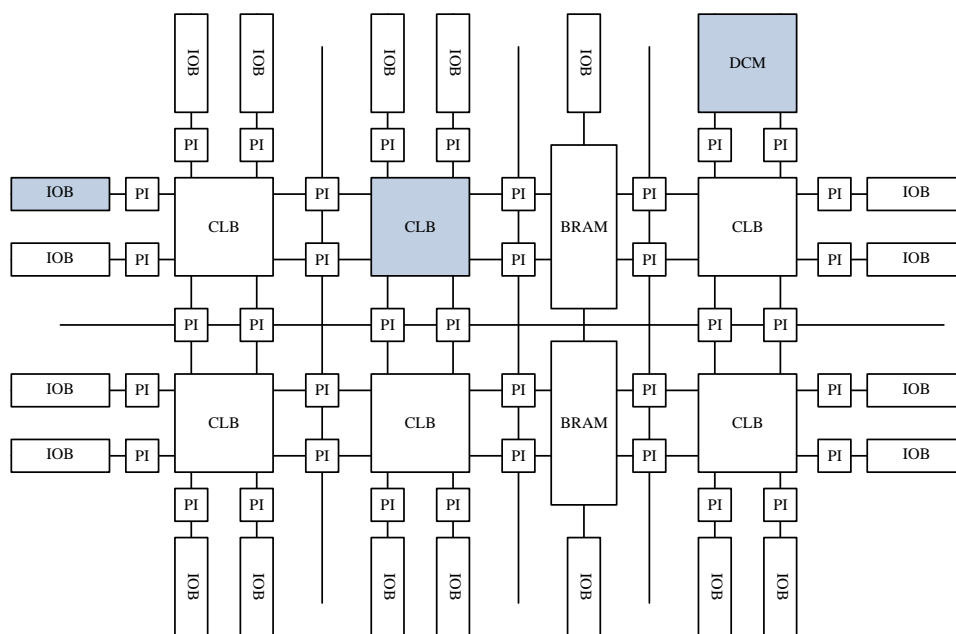


图 2-1 FPGA 芯片内部结构

Kintex-7 系列 FPGA 芯片内部逻辑资源结构如图 2-1 所示，其中主要包含可编程输入/输出块、可配置逻辑块、时钟管理模块、可编程内部互联资源、嵌入式块 RAM 以及内嵌功能单元模块<sup>[36]</sup>。

### (1) 可配置逻辑块 (CLB)

可配置逻辑块(CLB)是 Kintex-7 系列 FPGA 内最基本的也是最丰富的逻辑功能单元，主要用于完成组合逻辑电路与时序逻辑电路的设计<sup>[37]</sup>。如图 2-2 所示，Kintex-7 系列 FPGA 中每个 CLB 元件包含一对 SLICE 逻辑单元，其中，SLICE(0)位于 CLB 底部和左列，SLICE(1)位于 CLB 顶部和右列。X 编号用于标识 CLB 内每个 SLICE 的位置，以及所在的列，Y 编号用于标识 SLICE 所在行的位置。这两类 SLICE 互相之间没有直接连接，而是以列为单位，通过一个进位链将各自类型的 SLICE 相连。

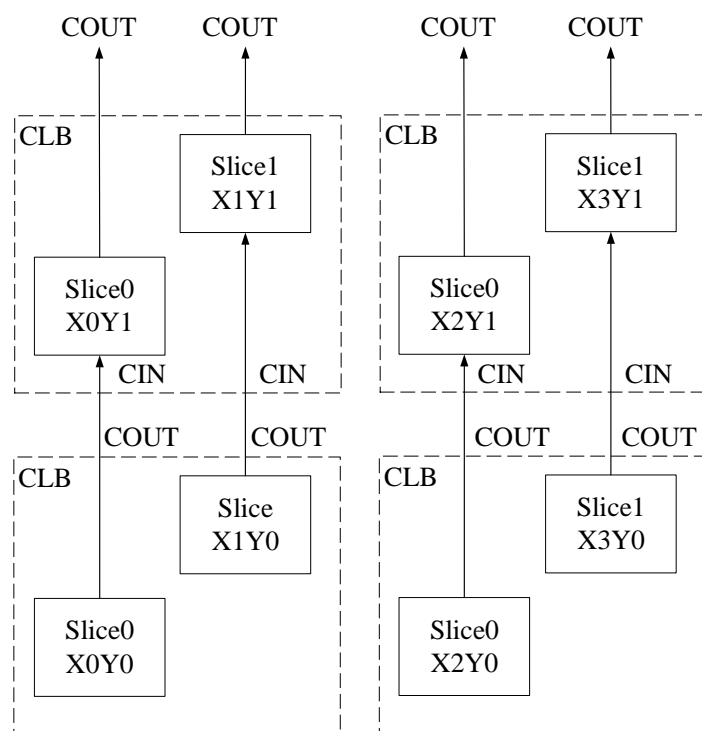


图 2-2 CLB 与 Slice 之间的行和列关系

Kintex-7 系列 FPGA 中每个 SLICE 都由四个 6 输入查找表(Look-up Table, LUT)、多路选择器(MUX)、四个触发器、进位链逻辑和四个可配置为锁存器的附加触发器组成，如图 2-3 所示。所有 SLICE 都通过这些单元来实现逻辑、数学运



算和 ROM 功能。特别的，部分 SLICE 还具备两个其他功能：利用 Distributed RAM 缓存数据和利用寄存器对数据进行移位处理。具备上述两种功能的 SLICE 称作 SLICEM，其他的则被称作 SLICEL。在广泛使用的 SLICE 资源中，接近三分之二为 SLICEL 资源，所以可以通过两个 SLICEL 或者一个 SLICEM 加一个 SLICEL 的组合来构成一个 CLB 资源。

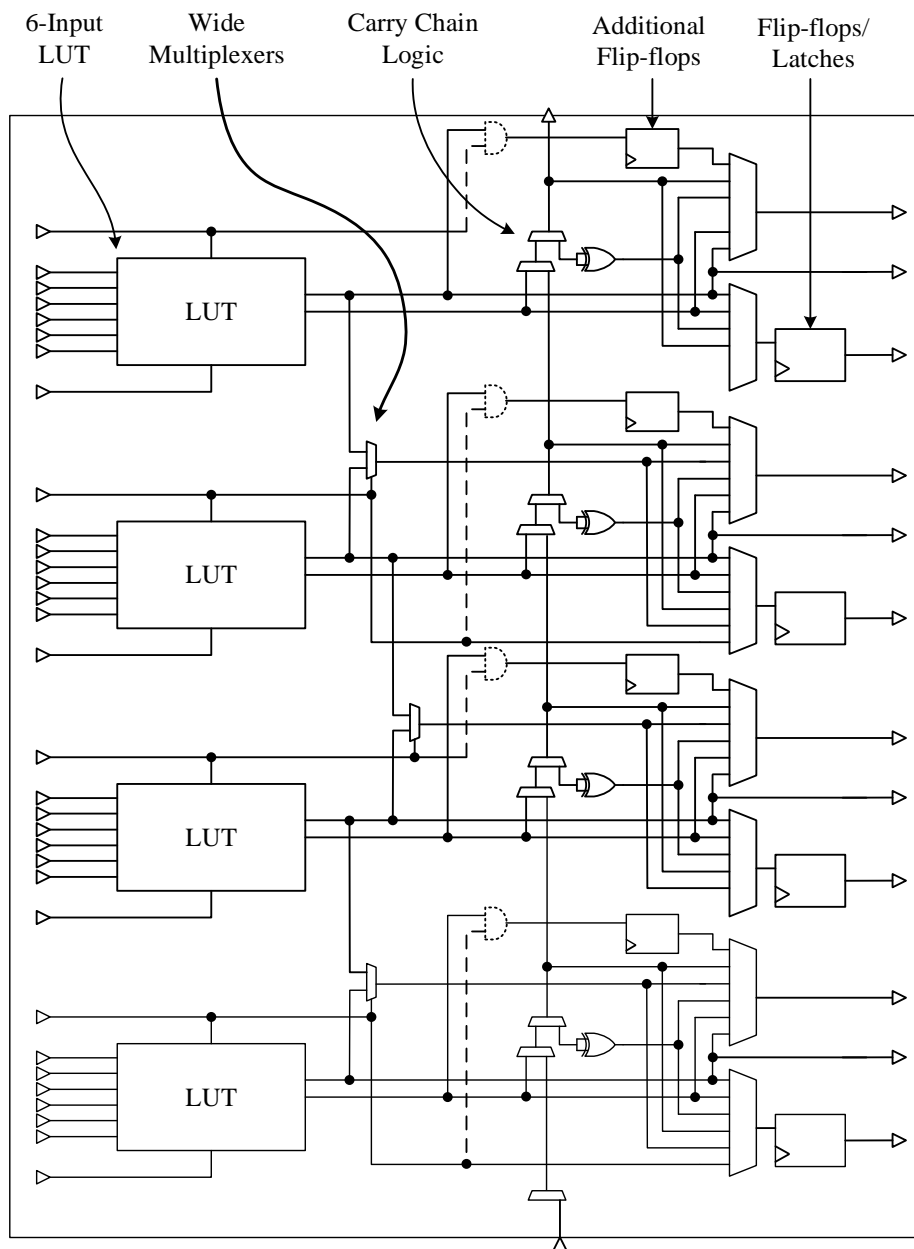


图 2-3 Kintex-7 FPGA SLICE 内部结构组成

对于数学中的多项式来说，不同的输入参数都能获得不同的输出结果，假如将输入数据集合与输出结果集合的对应关系通过映射表的形式进行保存，这样通过对应的数据输入就能在映射表中查找到预期的输出结果。LUT 实际上就可以理解为一张映射表，输入信号类似于地址，不同的地址映射不同的数据内容，每个输入信号都对应了一个固定的输出内容。LUT 的基本结构如图 2-4 所示，每一个晶体管的导通、截止都通过输入数据的‘0’和‘1’来进行控制，不同的输入数据驱动着该路线上的晶体管执行不同的动作，最终 LUT 会输出一个 SRAM 单元中存储的结果。

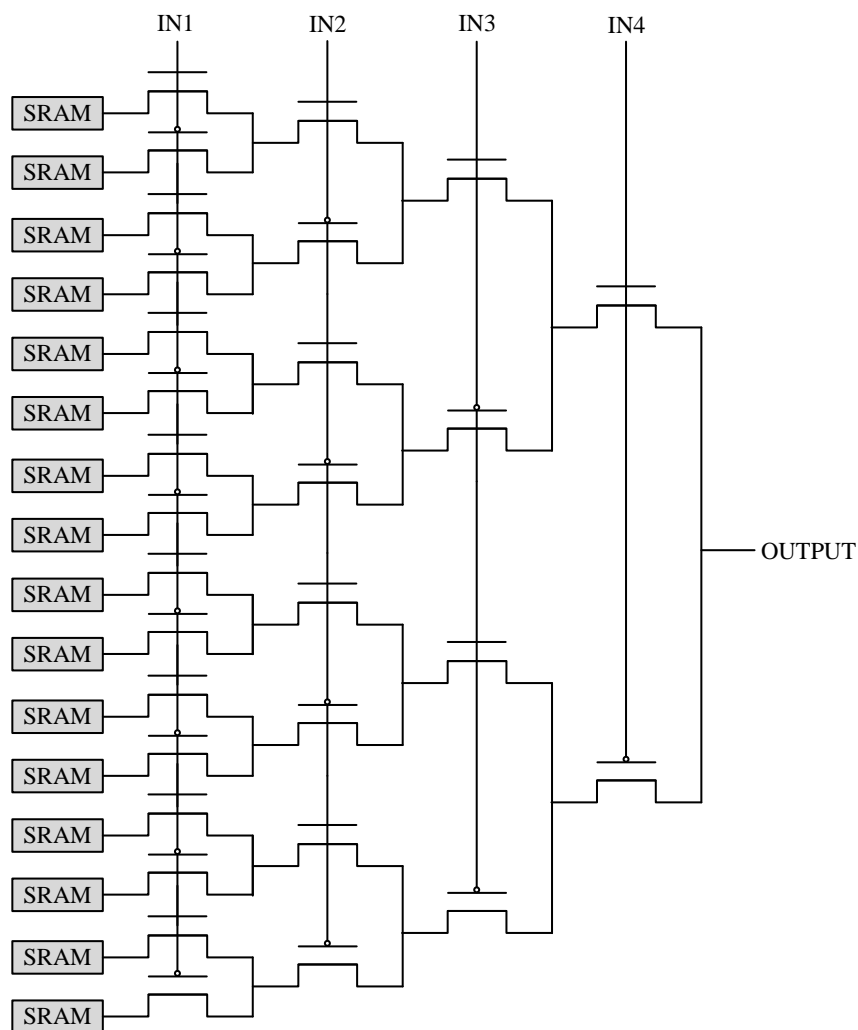


图 2-4 LUT 基本结构

接下来通过示例来详细介绍 LUT 的工作原理。如图 2-5 所示为一位二选一多路选择器的门电路结构，其中包括两个与门、一个或门和一个反相器。其中  $A_0$  和  $A_1$  为输入， $s$  表示选通信号，输出为  $y$ ，其布尔表达式如公式(2-1)所示。

$$y = A_0 \cdot \bar{s} + A_1 \cdot s \quad (2-1)$$

表 2-1 所示为二选一多路选择器的真值表，通过真值表可以将选择器映射到 LUT 中。三个输入信号  $A_0$ 、 $A_1$  和  $s$  能够随意组合为 8 种不同的输入数据，对应的 8 种输出结果通过 SRAM 单元分别进行存储，如图 2-6 a)所示为映射前的 LUT 结构。对照真值表中的数据，将结果映射到 LUT 中。例如， $A_0 = 0$ ， $A_1 = 0$ ， $s = 0$  时，输出结果  $y$  为对应的  $S_1$  中存储的值，即  $y = 1$ 。按照同样的方法，获得  $S_1 \sim S_8$  中对应的存储结果，最终获得如图 2-6 b)所示二选一多路选择器映射到 LUT 中的结果。

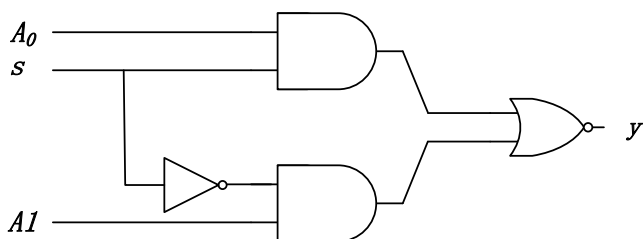


图 2-5 二选一多路选择器门电路

表 2-1 二选一多路选择器真值表

$A_0$	$A_1$	$s$	$y$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

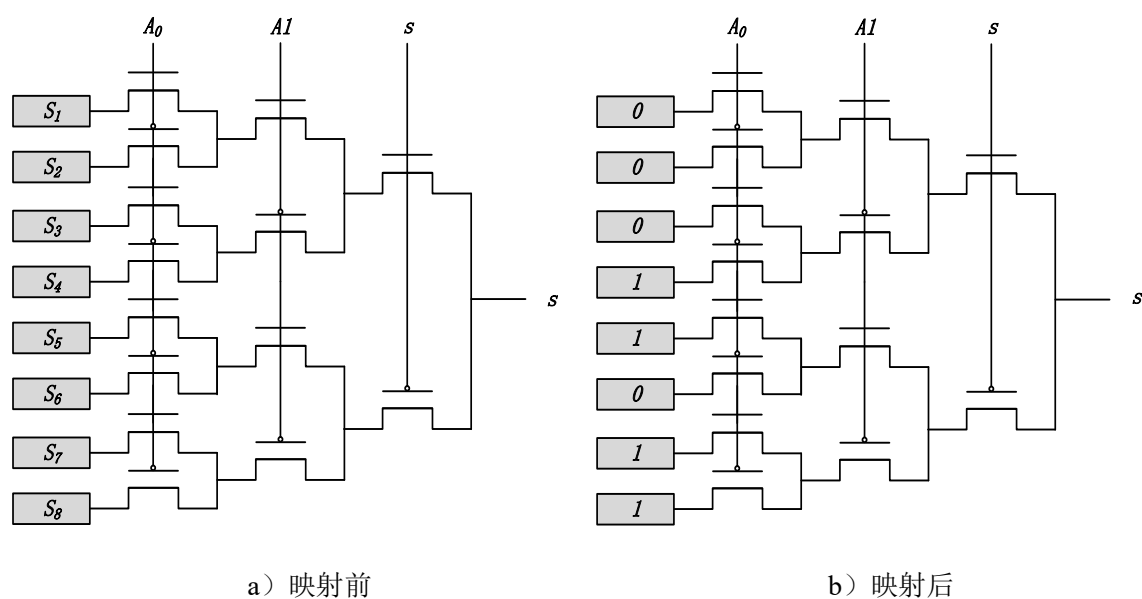


图 2-6 输入 LUT

## (2) 可编程输入/输出模块 (IOB)

可编程输入/输出模块 (IOB) 分布在 FPGA 的边缘，匹配各种标准的接口配置要求，如 LVTTTL、LVCMOS、PCI 和 LVDS 等，主要作用是为用户提供与 FPGA 的内部逻辑资源与外部设备之间建立数据交换的接口，提供接口电平转、输入缓冲、输出驱动、阻抗匹配、延迟控制等功能。

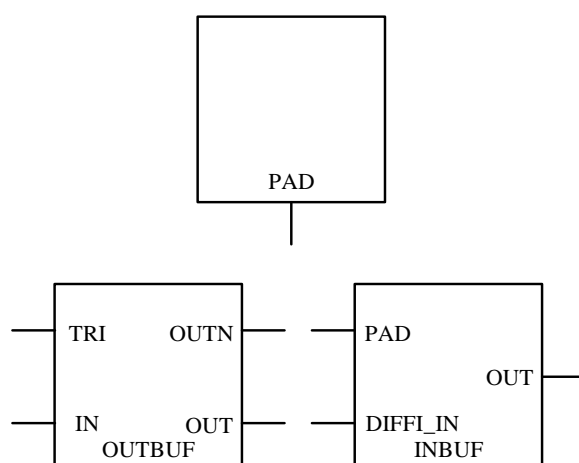


图 2-7 Kintex-7 系列 FPGA IOB 内部结构组成

为满足物理层与逻辑层的多种需求，如高速存储器、高速的 ADC/DAC 连接

以及传统接口，Kintex-7 系列 FPGA 对可编程输入输出块进行了改进提升。Kintex-7 系列 FPGA 中 IOB 主要由输入/输出 Buffer 和 PAD 组成，如图 2-7 所示。IOB 模块具有可编程性，通过软件按照不同的需求进行配置，进而符合各种标准的电气需求与输入输出物理特性；能够自动调整来匹配不同的阻抗、上下拉电阻；能够调整驱动电流的数值<sup>[38]</sup>。

### (3) 时钟管理模块 (CMT)

Kintex-7 系列 FPGA 内的每个时钟管理模块 (Clock Management Tile,CMT) 都包含一个混合模式的时钟管理器 (Mixed-mode Clock Manager,MMCM) 和一个相位锁相环 (Phase Locked Loop,PLL)。Kintex-7 系列 FPGA CMT 的核心是类似于 Virtex-5 和 Virtex-6 系列 FPGA 的体系结构，但是具备更强的功能和能力。如图 2-8 显示了各种时钟输入源和 MMCM/PLL 之间的连接关系，保证 CMT 为内部和外部提供精确的时钟。

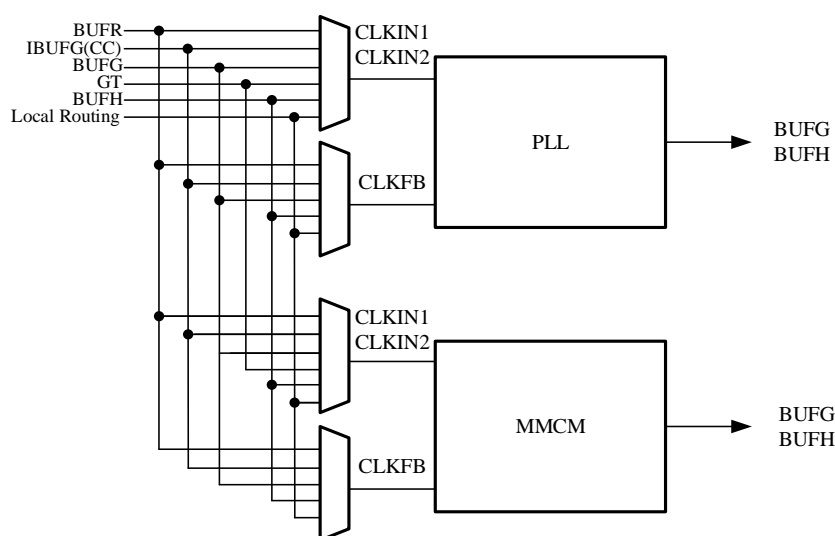


图 2-8 Kintex-7 系列 FPGA CMT 内部结构组成

### (4) 嵌入式块 RAM (BRAM)

Kintex-7 系列 FPGA 中均匀地排布了大量内嵌式块 RAM 资源，这极大地丰富了其使用的范围并提升了应用的灵活性。Kintex-7 系列 FPGA 中的 BRAM 存储高达 36Kbits 的数据，庞大的容量支持其配置为两个独立的 18Kb RAM 或者一个

36Kb RAM，其能够实现的功能包括：对高效的数据进行存储和缓冲、设置为简单双端口模式（Simple Dual-port, SDP）从而将数据位宽进行翻倍、配置为可编程的 FIFO、移位寄存器、LUT、ROM 等结构、内建可选的纠错电路等等，如图 2-9 所示为双端口 BRAM 结构。

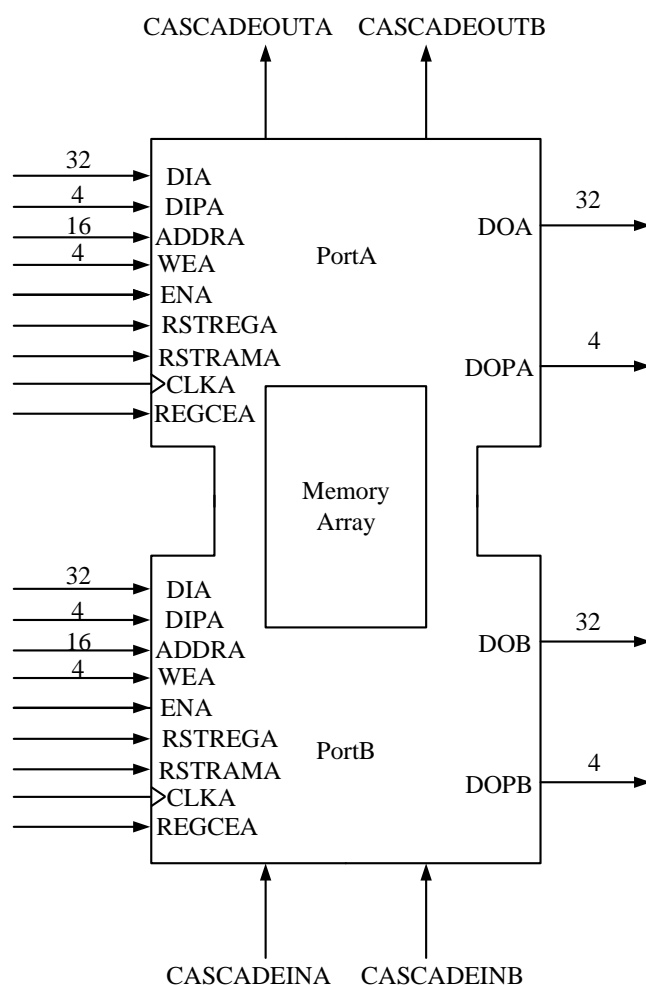


图 2-9 双端口 BRAM 结构组成

### (5) 可编程内部互联资源 (PI)

互联资源为 FPGA 内部各个逻辑资源元件，如 CLB、IOB、DSP 和块 RAM 之间搭建数据交换桥梁的可编程网络。互联也称布线，规则的排布在器件设备上，以最合理的方式进行连接。互联资源主要由三个部分组成：可编程开关矩阵、金属导线以及可编程开关点。金属导线作为通讯信号的载体，保证 FPGA 内部各个

模块之间的信号能够正常传输，可编程开关点和可编程开关矩阵则用来控制哪些模块之间能够进行通讯，调整信号传输的路径和方向。在 Kintex-7 系列 FPGA 中包含多种类型的布线资源，按照长度定义可以区分为：快速连接、单连接、双连接和四连接。快速连接将模块的输出输入相连；单连接分布在垂直和水平方向上，用于将相隔一个单位的可配置逻辑单元互相连接；双连接分布在四个方向上，将水平方向与垂直方向以及对角线相邻的单元相互连接；四连接分布在水平和垂直方向上，相隔四个单位的可配置逻辑单元互相连接或者将相隔两行和两列距离的可配置逻辑单元相连，如图 2-10 所示。

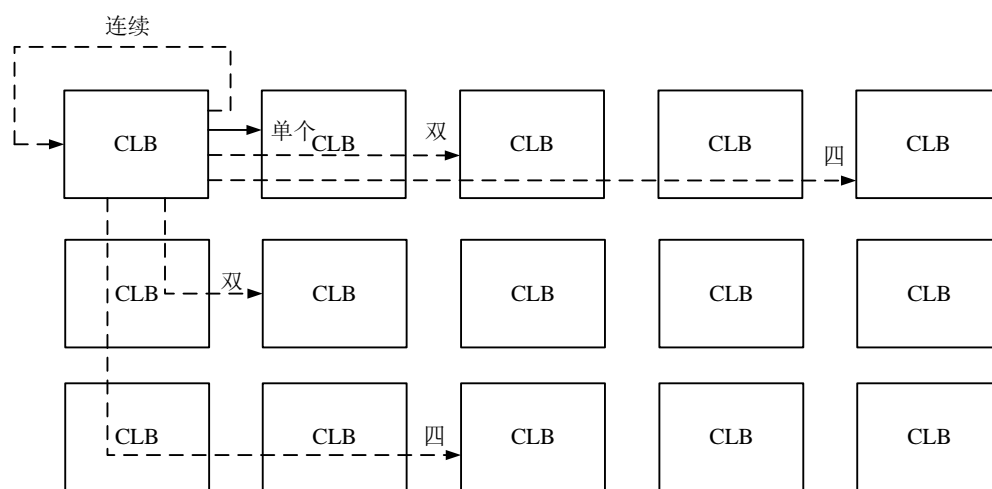


图 2-10 Kintex-系列 FPGA 不同类型的布线

## (6) 内嵌功能单元模块

内嵌功能模块主要包括锁相环 (DLL)、乘法器、数字信号处理器 (Digital Signal Processing,DSP)、软处理核等 (Soft Core)。日益完善的内嵌功能单元使得 FPGA 兼具软件和硬件共同设计的能力，并逐渐向系统级芯片 (System On Chip,SOC) 过渡。

## 2.2 Kintex-7 FPGA 单粒子效应

研究针对 Kintex-7 系列 FPGA 的故障注入方法，需要了解该类型 FPGA 造成故障的原因。Kintex-7 FPGA 对单粒子效应具有高度的敏感性，即使对其进

行相关的加固设计，仍然会受到空间环境中高能粒子的辐射影响。本小节，主要对 Kintex-7 系列 FPGA 产生的单粒子效应进行阐述。

空间领域中的设备由于处于不同的轨道中，因此受到不同程度的空间辐射影响，但都可以归结为两类：单粒子效应与总剂量效应(Total Ionizing Dose,TID)<sup>[39]</sup>。总剂量效应是一种长时间积淀的累积效应，而单粒子效应是由单个高能粒子辐射引发的短期效应，从时间角度考虑，SRAM 型 FPGA 在应用时更加关注短时间内引起的辐射影响，因此空间辐射效应的主要研究对象为单粒子效应<sup>[40]</sup>。

处于太空环境下工作的集成电路器件，在运行过程中，当其敏感区域因为遭到单个高能辐射粒子撞击而导致器件的状态发生非正常性变化的现象称作单粒子效应。单粒子效应产生的原因是，空间环境中的单个带电高能粒子击中集成器件的灵敏部位，因为发生电离作用产生了影响集成器件内部电荷平衡的额外电荷，导致器件的逻辑状态发生翻转，进而影响了电路的正常功能甚至使功能失效<sup>[41]</sup>。

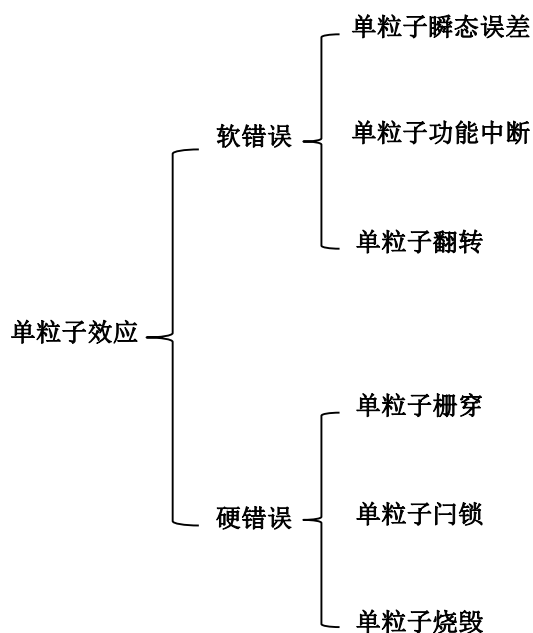


图 2-11 单粒子效应种类

单粒子效应多种多样，根据高能粒子对集成器件电路功能造成的影响是否是永久性的，可以分为硬错误与软错误。软错误是指集成器件由于单粒子效应造成电路功能发生暂时性故障，包括单粒子翻转、单粒子瞬态脉冲和单粒子功能中断



等；硬错误是指集成器件由于单粒子效应造成电路功能发生永久性故障，包括单粒子烧毁、单粒子栅穿和单粒子门锁等<sup>[42]</sup>，如图 2-11 所示。

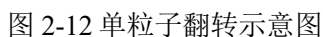
常见的几种单粒子效应类型分类如表 2-2 所示，其中描述了各种类型的单粒子效应发生的机理以及可能会出现的位置，其中 SRAM 型 FPGA 对单粒子翻转效应最为敏感，单粒子翻转也是造成 FPGA 故障最普遍、最主要的原因<sup>[43]</sup>。

表 2-2 单粒子效应的分类

类型	定义	易发生位置
单粒子翻转 SEU (Single Event Upset)	存储单元逻辑状态翻转	数据存储单元或指令
单粒子瞬态 SET (Single Event Transient)	单粒子入射混合逻辑电路引起的瞬态脉冲	大规模数字电路中的组合逻辑电路等
单粒子门锁 SEL (Single Event Latchup)	寄生可控硅结构中的大电流状态	含有 pnpn 结构的 CMOS 集成电路
单粒子烧毁 SEB (Single Event Burnout)	大电流导致器件烧毁	BJT 三极管、功率 MOSFET
单粒子栅穿 SEGR (Single Event Gate Rupture)	大电流击穿栅介质，漏栅永久短路	功率 MOSFET、非挥发性 SRAM 等

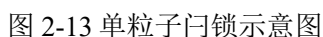
## (1) 单粒子翻转 (Single Event Upset,SEU)

单粒子翻转是太空环境中高能粒子辐射造成的单粒子效应中最典型也是最普遍的一种，其主要发生在半导体器件的锁存器、寄存器、触发器等存储单元，包括数据以及指令信息<sup>[44]</sup>。造成单粒子翻转的主要原因是：处于辐射、电磁环境相对恶劣的情况下时，集成电路器件的敏感区域受到单个高能辐射粒子的轰击，使得器件的逻辑状态发生翻转，由原始状态的“0”翻转为“1”，“1”翻转为“0”，进而导致系统功能异常甚至发生事故。但是，单粒子翻转造成的影响不是永久性的，且不会对器件本身造成损害，可以通过重新上电或者重新配置等操作使系统恢复正常。降低航天器件发生单粒子翻转效应概率的普遍方法是增加冗余度，对目标器件中易发生单粒子翻转的逻辑电路进行三模冗余处理。如图 2-12 所示为单粒子翻转示意图。



## (2) 单粒子闩锁 (Single Event Latchup, SEL)

单粒子门锁主要发生在含有 **pn-pn** 结构的 **CMOS** 结构中。当单个高能辐射粒子入射，撞击集成电路器件的敏感区域时，由于 **CMOS** 的特殊结构，使得流过的电流不断增大而产生瞬态电流，导致目标器件的功能紊乱的现象称作单粒子门锁。通过重新上电的方式可以清除器件中的单粒子门锁，但如果没有及时断电，不仅会影响集成电路器件的功能，严重时还会对器件本身造成永久性的不可逆的损伤。如图 2-13 所示为单粒子门锁示意图。



## (3) 单粒子烧毁 (SEB)

单粒子烧毁是一种“硬错误”，会对集成电路器件造成不可逆的永久性的故障。单粒子烧毁产生的原因与单粒子门锁相似，高能辐射粒子轰击集成电路器件产生的瞬态电流导致其敏感区域的电流不断增加，最终击穿晶体管，造成电路永久性损毁。

日益增长的卫星数量与复杂的功能需求，使得越来越多的集成电路器件被应用于航天系统中，由于集成电路器件对空间环境中高能辐射粒子的高度敏感性以及单粒子效应不可忽视的严重危害性，因此对 SRAM 型 FPGA 进行单粒子效应评估十分必要。本论文中主要对最典型、最普遍的单粒子翻转效应进行故障注入技术的方法研究。

## 2.3 单粒子翻转对 Kintex-7 FPGA 的故障解析

Kintex-7 系列 FPGA 芯片配置存储器中不同类型的资源发生 SEU 故障现象如图 2-14 所示，其中查找表、多路选择器与布线资源中由单粒子翻转效应产生的故障是持续的，可以通过重新上电或者重新配置恢复到正常状态；发生在时序逻辑资源中的单粒子翻转效应故障是瞬间的，下一个时钟到来后能够自动恢复到正确的逻辑状态，不需要手动操作。本小节主要对 Kintex-7 系列 FPGA 中 LUT、MUX、Buffer、布线资源发生的单粒子翻转故障进行解析。

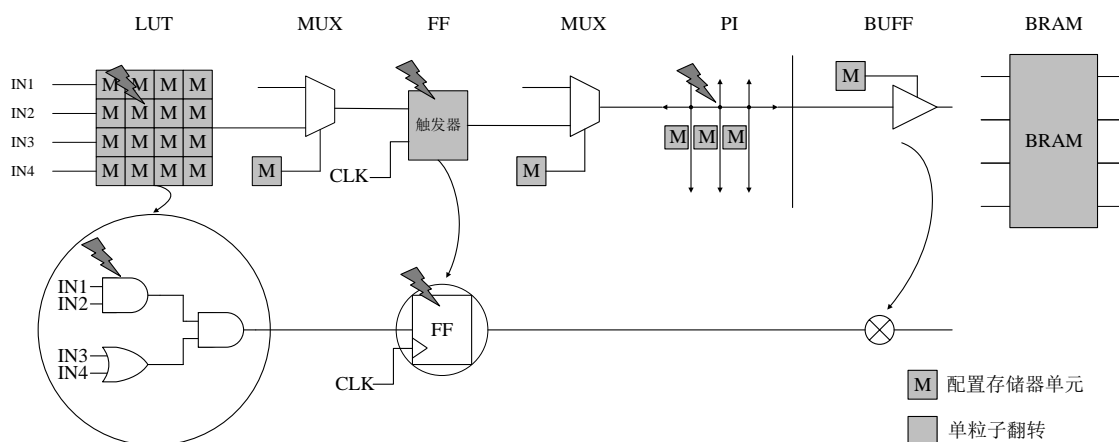


图 2-14 Kintex-7 系列 FPGA 配置存储器 SEU 故障<sup>[33]</sup>

## (1) LUT 的单粒子翻转

姚志斌等人研究了 FPGA 配置存储器中各类型资源对单粒子翻转的敏感性后发现，LUT 的位翻转饱和截面远大于其他类型资源，是单粒子翻转效应发生的主要区域<sup>[45]</sup>。查找表的主要功能是对数据进行储存、替代传统的门电路实现组合逻辑。前面小节中提到，LUT 实际上可以理解为一张映射表，输入信号类似于地址，不同的地址映射不同的数据内容，每个输入信号都对应了一个固定的输出内容。如图 2-15 所示为 FPGA 中 LUT 发生单粒子翻转的过程，高能辐射粒子轰击 FPGA 内部配置存储器中的查找表资源，造成 LUT 中的逻辑状态发生翻转，由原始状态的“0”变为“1”、“1”变为“0”。当输入信号还是原来的数据时，由于存储结果的 SRAM 单元发生单粒子翻转效应，导致最终的输出结果发生改变，由“1”变为“0”，得到错误的结果，进而可能对用户电路的逻辑状态造成影响，导致系统功能紊乱。

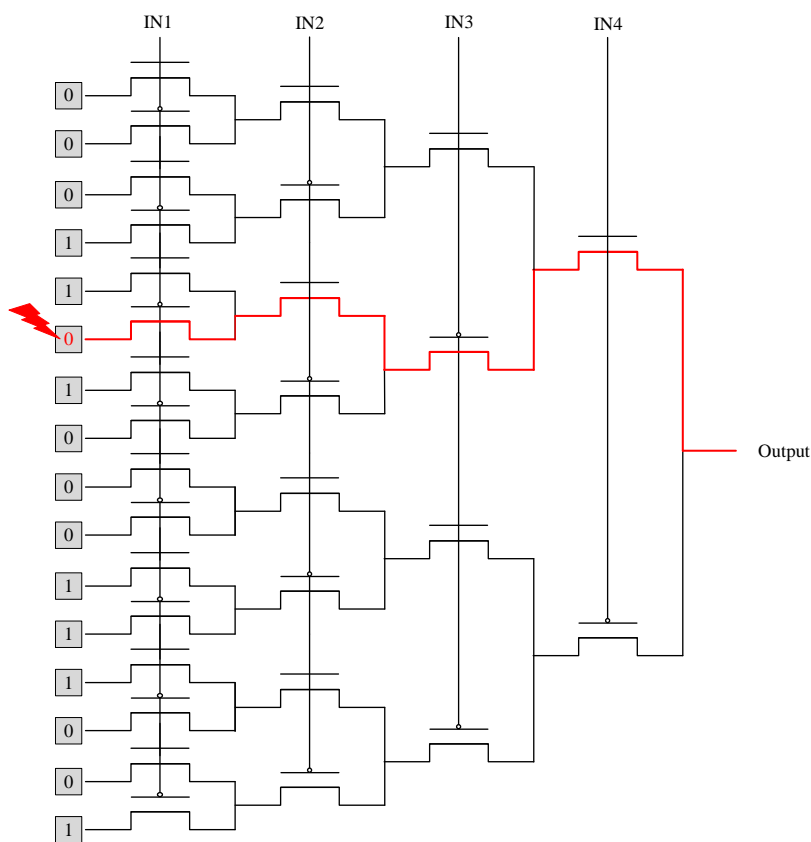


图 2-15 LUT 发生单粒子翻转

## (2) 多路选择器发生单粒子翻转

多路选择器是连接各个布线路径与 CLB 模块的桥梁，数字信号在进行传输的过程中，多路选择器能够从多路并行的输入数据中按照特定的条件从中选取符合要求的某一路数据输出。如图 2-16 所示为多路选择器发生单粒子翻转的过程，单粒子翻转导致配置存储器中的数据发生“0”→“1”或“1”→“0”的转变，改变多路选择器原始的输出数据，最终导致系统功能紊乱。

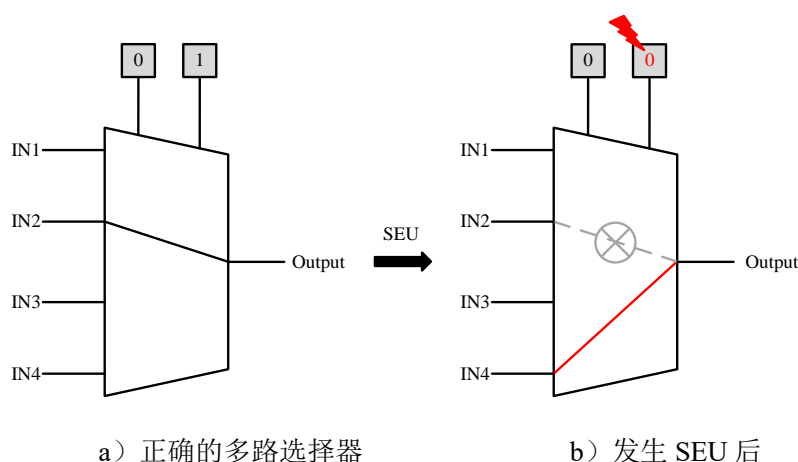


图 2-16 多路选择器发生单粒子翻转

## (3) 可编程内部互联资源 (PI) 发生单粒子翻转

内部互联资源能够将 FPGA 中位于不同位置的 CLB、时钟处理单元、BRAM 等资源连接起来，使它们相互通信、协调工作，进而完成相应的功能。Kintex-7 FPGA 中的可编程内部互联资源主要由可编程链路开关与不同长度的互联线组成。互联线在可编程链路开关处相互交接形成的结点称作可编程互联结点 (Programmable Interconnect Points, PIP)，用户能够以编程的方式通过 PIP 将线、逻辑资源以及输入输出接口互联，从而实现所需的逻辑功能。可编程互联结点实际上就是 SRAM 单元，当值为 0 时代表处于开路状态，值为 1 时代表处于连接状态。如图 2-17 所示为 PI 发生单粒子翻转的过程，可编程互联结点发生单粒子翻转时，主要会造成电路开路或短路两种故障情况。例如，可编程互联结点①数据由“0”变为“1”将导致原本不相连的线路 A1 与 B2 短路，由于短路后造成的驱动力与信号大小不匹配，可能会输出一个不稳定的信号，影响用户电路的正常功

能；②数据由“1”变为“0”将导致原本连接的线路 A3 与 D1 开路，同样也会输出一个不稳定的信号，导致用户电路功能紊乱。

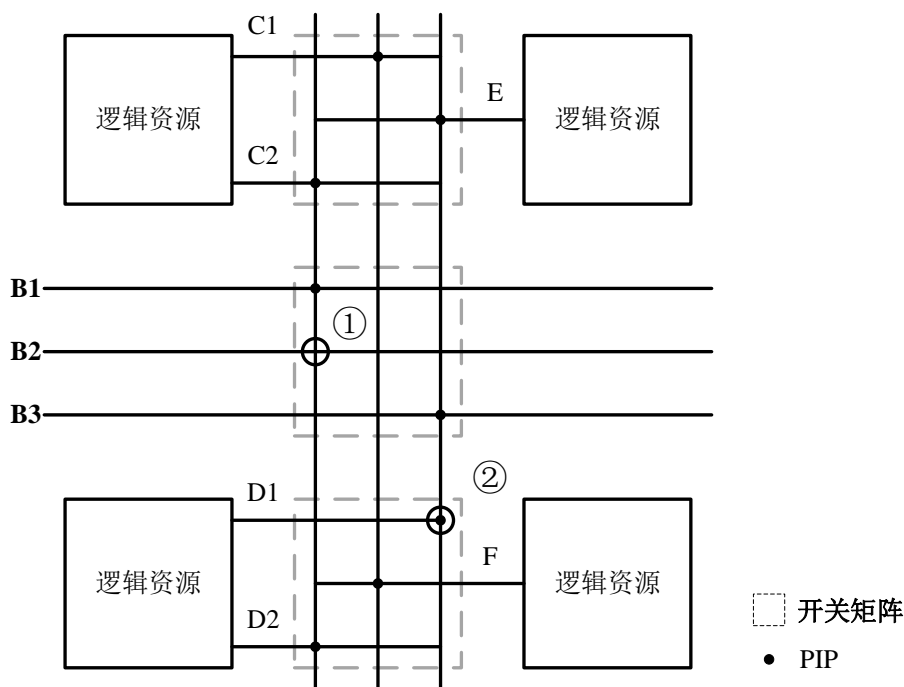


图 2-17 PI 发生单粒子翻转

## (4) 缓冲器发生单粒子翻转

缓冲器（Buffer）常被用于总线上，其主要作用是当总线无法驱动负载时，增加驱动能力；当缓冲器输出无法提供高阻状态时，阻隔前后信号。缓冲器具备三态输出的功能，三种输出结果分别为逻辑 0、逻辑 1 与高阻态，如表 2-3 所示，输出的结果由输入和使能决定。Kintex-7 系列 FPGA 的缓冲器分为输入与输出两种缓冲器，输入缓冲器用于暂存外部设备输入的数据，直至内部将其读取；输出缓冲器用于暂存内部送往外部设备的数据。工作速度差异明显的内部处理器与外部设备在缓冲器的作用下能够实现数据的同步传输。

表 2-3 三态缓冲器真值表

Input	Enable	Output
0	1	0
1	1	1
X	0	Z(high impedance)

如图 2-18 所示, 当高能粒子入射缓冲器敏感区域时, 造成器件中存储使能信号的 SRAM 逻辑单元发生单粒子翻转效应, 使能信号的原始逻辑状态被打翻, 改变了缓冲器的输出结果, 造成电路的时序紊乱, 进而导致系统功能失效。

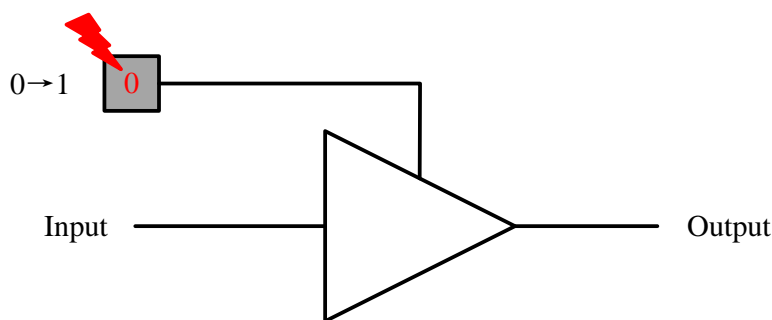


图 2-18 缓冲器发生单粒子翻转

### 2.4 单粒子翻转故障定向注入技术难点分析

故障注入技术是目前最普遍的 FPGA 单粒子翻转敏感性的评估方法。本论文将以 Kintex-7 系列 FPGA 芯片的配置存储器为目标, 通过故障注入的方法人为地将配置区的内容进行修改, 以此来模拟空间环境中的单粒子翻转效应。目前普遍采用的故障注入方法主要利用 ICAP 接口对 FPGA 配置寄存器进行回读, 将回读的帧数据位翻转后回写, 从而实现对 FPGA 中用户电路注入单粒子翻转故障。但是通过这种方法只能随机打翻帧数据中的配置位, 对于翻转后的配置帧数据产生的后果未知, 无法有目的地针对用户电路中某个功能或者某个模块进行定向地故障注入。另一种基于定向故障注入的 SRAM 型 FPGA 单粒子翻转效应评估方法, 借助 XDL 工具解析 FPGA 的帧地址与物理资源之间的对应关系, 根据二者之间存在的关系将用户电路中的资源按照模块进行划分, 然后利用部分重构技术分别对电路整体及各分组相关的配置帧进行随机故障注入, 从而对电路整体及其子模块的抗单粒子翻转能力进行评估。该方法同样只是对 FPGA 进行随机单粒子故障注入, 但是其通过解析帧地址与逻辑资源之间对应关系的想法值得本文借鉴, 本文在此基础上进一步探究了帧地址与具体用户电路功能模块之间的关系, 从而实现对指定模块的单粒子翻转定向故障注入。然而, 针对 Kintex-7 系列 FPGA 芯片实

现故障的定向注入主要有以下难点。

### （1）配置存储器帧结构未知

赛灵思公司设计的 ISE 开发工具能够生成与用户电路紧密相关的特有二进制文件即比特流文件，配置存储器对比特流文件进行调用和配置后将其映射成实际的电路进而实现用户设计的功能。配置比特流文件包含用户电路中所有可配置逻辑资源的信息，包括可配置逻辑块、可编程输入输出块、时钟管理模块、BRAM 互联以及 BRAM 内容块等。配置帧中不同的位对应着不同的逻辑资源配置信息，通过对这些位进行操作，能够控制不同逻辑资源块的可编程逻辑点实现相应的功能。然而，本文所使用的 Kintex-7 系列 FPGA 芯片赛灵思公司并未提供公开的配置帧结构具体信息，因此需要自行解析。

### （2）配置帧数量过于庞大

在执行故障注入时，通过修改配置文件来模拟由单粒子翻转引起的 FPGA 中信息的变化。对于每种类型的 FPGA，配置文件中的帧数是不同的。一般来说，特定的设计不能占用 FPGA 中所有的可编程资源，未配置的资源对系统的功能没有影响。Kintex-7 系列 FPGA 中的每个可配置资源都由一个或多个配置内存定义，配置内存被组织为一组帧，很像一个范围宽广的静态 RAM，因为大多数 FPGA 中的配置位数都很高，范围从几兆到超过千兆，例如，Kintex-7 系列 FPGA 具有 2 896 320 个位，所以将配置内存中的所有位进行故障注入，会花费巨大的时间成本，更加重要的是，并不是所有配置位上发生的单粒子翻转效应都会导致设计电路的逻辑功能发生改变，只有配置编程到 FPGA 中的特定应用程序使用的资源的这些位的子集才是相关的。因此，需要缩减故障注入的配置帧数量，避免对不相关的配置帧进行故障注入。

### （3）难以实现指定区域或功能模块的故障注入

常见的故障注入方法包括对所有配置数据帧中的位执行全局故障注入或是随机产生某些配置帧中的位执行随机故障注入。针对用户设计中的某个模块或是某个功能，我们需要单独对其进行单粒子翻转敏感特性的测试，人为的对该指定模块或功能执行故障注入，进而测试其可能产生的故障以及对系统造成的影响。但



是配置数据帧只是简单的“0”和“1”组成的集合，难以将其与用户设计电路中模块和功能的实际物理区域联系起来。因此。需要探究配置帧与特定模块和功能之间的关联，从而实现定向故障注入。

### 2.5 本章小结

本章主要围绕 Kintex-7 FPGA 芯片 SEU 故障注入技术展开研究。从 Kintex-7 FPGA 的基本结构以及工作原理入手，通过分析各逻辑资源模块的特殊结构特征，阐述空间辐射环境中的单粒子效应产生的原因。作为 SRAM 型 FPGA 配置存储器特有的单粒子效应，单粒子翻转是其中最普遍也是导致集成电路器件功能紊乱的主要原因。因此以 Kintex-7 FPGA 为目标，分析了配置存储器中各逻辑单元发生单粒子翻转的机理，以及单粒子翻转对它们产生的影响。最后，分析了单粒子定向故障注入方法实现的主要难点。

## 3 基于 ICAP 回读技术的配置帧结构解析方法

第二章中介绍了 SRAM 型 FPGA 由于高能粒子辐射影响会产生的多种单粒子效应,针对其中最典型、发生概率最高的单粒子翻转效应,从 FPGA 最基础的逻辑资源出发,逐个解析它们发生故障的原因并阐述了实现故障定向注入的难点。本章将针对 SRAM 型 FPGA 配置存储器中的配置帧展开研究。

对于 SRAM 型 FPGA,单粒子翻转效应会影响其中任何一个存储单元,其中配置 RAM 决定了用户电路的逻辑功能,为了实现对用户电路中指定功能模块的定向故障注入,需要明确其对应的配置 RAM 中的配置帧数据。但是,Kintex-7 FPGA 厂家并未明确给出具体的配置帧数据结构,需要设计有效的配置帧数据结构解析方案。针对配置存储器中帧结构未知的问题,本章设计一种基于 ICAP 回读技术的帧结构解析方法,首先通过 ICAP 读写控制模块向 ICAP 接口发送命令序列,实现对其配置存储器的回读,获取配置帧数据后,利用 FRAME\_ECC 输出至上位机,由上位机对其进行帧结构解析,分离出对应配置 RAM 资源的配置帧,以进一步缩小配置帧数量,为用户电路功能模块定向故障注入提供基础。

### 3.1 Kintex-7 FPGA 配置存储空间分析

#### 3.1.1 Kintex-7 FPGA 基本配置结构

Kintex-7 系列 FPGA 采用的是适应各个领域应用需求的多模块架构(Advanced Silicon Modular Block,ASMBL),该架构为用户以经济性、快速性为需求而实现针对特定领域的最佳性能和功能的平衡提供了平台<sup>[46]</sup>。ASMBL 架构是一种以器件子系统模块化框架为核心的架构,换句话说,ASMBL 凭借其特有的以列为单位的结构,实现了面向不同应用领域的 FPGA 平台,如图 3-1 所示,每一列中都由相同的具备特定性能的逻辑资源组成,如高速 I/O、全局时钟、存储器、DSP、嵌入式处理器、硬 IP 以及其他隐藏功能。通过将这些具有特定功能的列进行相应的组合,赛灵思公司开发出了适用于各式各样特定领域的专门 FPGA,包括逻辑密集型、存储密集型和处理密集型。

配置 RAM 是 FPGA 中重要的数据存储单元，包含所有 FPGA 配置存储器的配置存储空间中的数据决定了 FPGA 所具备的特定功能，如 I/O 资源的电压值，布局布线内容、锁存器（Latch）和触发器（Flip-Flop）值、查找表内容和 BRAM 值等。类似于 ASMBL 中基于列的结构，Kintex-7 系列 FPGA 配置存储空间是以配置帧为单位构成的，不同类型的逻辑资源包含的配置帧数量不同。

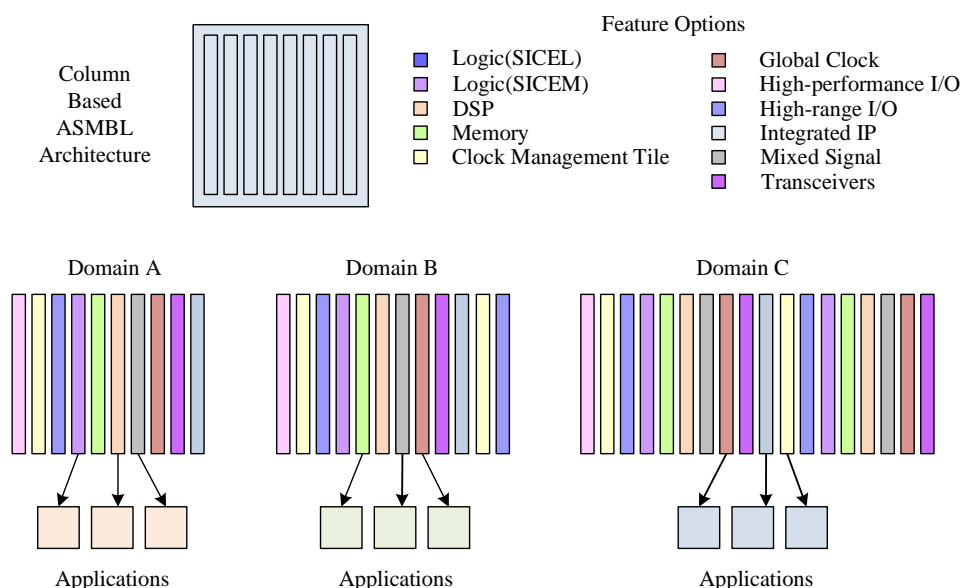


图 3-1 ASMBL 基本架构<sup>[46]</sup>

## 3.1.2 Kintex-7 FPGA 配置 RAM

通过外部注入的方法将 SEU 注入 FPGA 时需要考虑两点：一个是设计电路中使用的配置资源，另一个是比特流文件中帧的数量和帧地址之间的对应关系。Kintex-7 FPGA 中，可编程资源主要包括 CLB、IOB、CLK 和 BRAM，其中 CLB、IOB、CLK 属于配置 RAM<sup>[47,48]</sup>。配置 RAM 中存储的数据决定用户电路实现的逻辑功能，BRAM、分布式 RAM、触发器等存储单元用于存储用户电路运行状态下的数据信息。不同的设计电路占用的资源也不相同，本论文主要针对配置 RAM 中的存储单元进行单粒子翻转研究。

用户电路中每个模块实现的具体逻辑功能依赖于配置 RAM 中加载的配置比特流文件，如图 3-2 所示，配置层中配置 RAM 使用的逻辑资源之间不同的组合会

映射成为应用层中相应的逻辑电路。配置 RAM 中的逻辑资源以“0”和“1”组成的二维矩阵阵列的形式均匀的排布在器件上，每一个“0”和“1”都代表着可编程逻辑点。当配置 RAM 中的可编程逻辑点由于单粒子翻转发生改变时，将会影响原本正常的逻辑功能，导致系统出现功能紊乱产生严重后果。通过 SelectMAP 等外部接口配置或 ICAP 等内部配置接口将正确的配置比特流文件加载到配置 RAM 中能够对电路功能进行修复或更新。

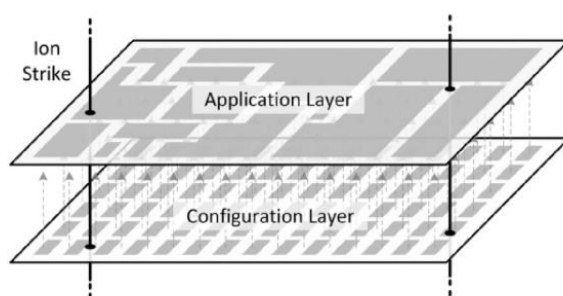


图 3-2 Kintex-7 FPGA 结构模型

表 3-1 帧地址寄存器描述

帧地址类型	位元索引	描述
保留位	[31:26]	无法被写入，保持全 0
块类型	[25:23]	有效的块类型： CLB、I/O、CLK、block RAM 互联资源（000） block RAM 内容（001） 正常的比特流文件不包含类型 011
上半区/下半区地址	22	选择上半部分的行或者下半部分的行 上半区：0 下半区：1
行地址	[21:17]	选择具体的某一行，行地址不论是在上半部分或者下半部分都是从靠近中心部分向远离中心部分增加
列地址	[16:7]	选择某一行之内的某一列，如一系列 CLB。列地址从左边的 0 开始一直增加到右边
次要地址	[6:0]	从一系列内选择一个存储单元地址线

Kintex-7 FPGA 的配置存储器由帧组成，帧是最小可寻地址单元，因此所有操作都必须以整个配置帧为单位进行。Kintex-7 系列 FPGA 中的所有帧的固定长度相同都是 3232 位，每一帧由 101 个 32 位的字组成，且每个配置帧都有唯一的一个地址与其对应。帧地址的描述如表 3-1 所示，主要包含五种类型：Block type、Top/Bottom、行地址、列地址以及列内帧地址。其中 Block type 有五种：块 0 为 CLB、IOB、CLK 以及 BRAM 互联资源；块 1 为 BRAM 内容块 0 代表的资源与用户电路的具体功能关系紧密，块 1 的 BRAM 内容是否发生变化并不会影响用户电路功能的正常运行，因此本文只针对块 0 进行故障注入。

帧结构解析的前提是明确 FPGA 中的帧具体排布方式，即帧的总数、具体的帧地址、帧地址类型以及不同帧地址类型的数量。Kintex-7 FPGA 手册中并未明确给出具体的帧地址分布，因此需要手动对其进行解析。本文利用内部配置访问接口（Internal Configuration Access Port, ICAP）从 Kintex-7 FPGA 的配置存储器中回读具体的帧地址，再进一步解析获取帧结构信息。

### 3.2 基于 ICAP 回读技术的帧结构解析方案

Kintex-7 系列 FPGA 通过配置接口与外部设备进行数据交互，主要包括三种接口：联合测试工作组（Joint Test Action Group, JTAG）、SelectMAP 与 ICAP。

JTAG 接口与 SelectMAP 接口能够读取和写入 Distributed RAM 中的数据，但是无法对块 RAM 逻辑资源进行操作。与 JTAG 接口与 SelectMAP 接口相比，ICAP 接口读取和写入数据的速度更快并且具备部分重配置的功能，更快的读写速度使得利用 ICAP 接口进行故障注入时具备更高的效率，同时部分重配置的特点能够针对性的对目标区域的逻辑进行单粒子翻转的故障注入，大大减少了全局注入的时间消耗。因此 ICAP 接口在模拟单粒子翻转故障注入相关实验时，扮演着越来越重要的角色，本章也将采用 ICAP 接口读取 FPGA 中的配置帧数据。读取出的配置帧数据需要进行输出，本章采用的是 Kintex-7 系列 FPGA 内嵌的一种名为 FRAME\_ECC 的逻辑电路，其通过 13 位的汉明校验码（Hamming Code）能够检测从配置存储器中回读的数据是否出现至多 2bit 的错误，并对出现 1bit 的错误进

行纠正。本课题主要采用的是 FRAME\_ECC 在回读配置数据的同时，能够输出 FAR 寄存器中的帧地址的功能。输出的配置帧数据通过上位机软件接收并进行结构解析后即得到理想的 Kintex-7 系列 FPGA 帧结构物理帧地址组织。

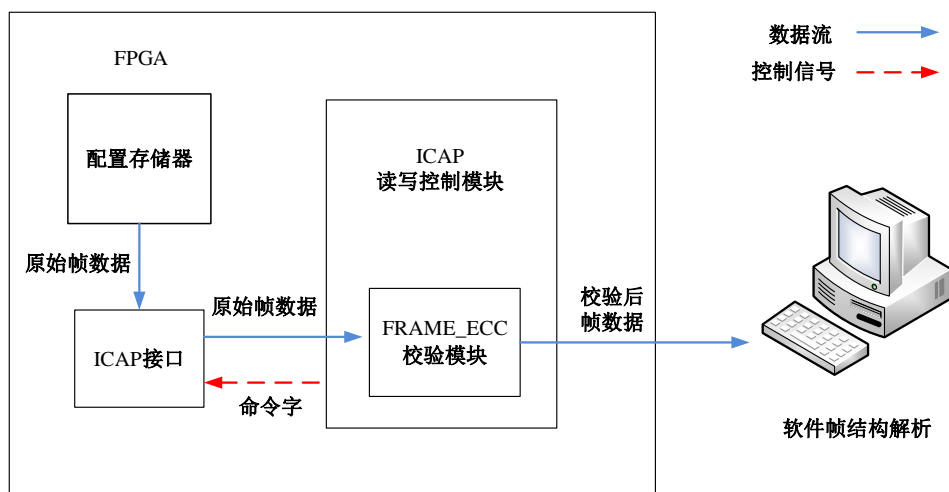


图 3-3 帧结构解析系统整体方案框图

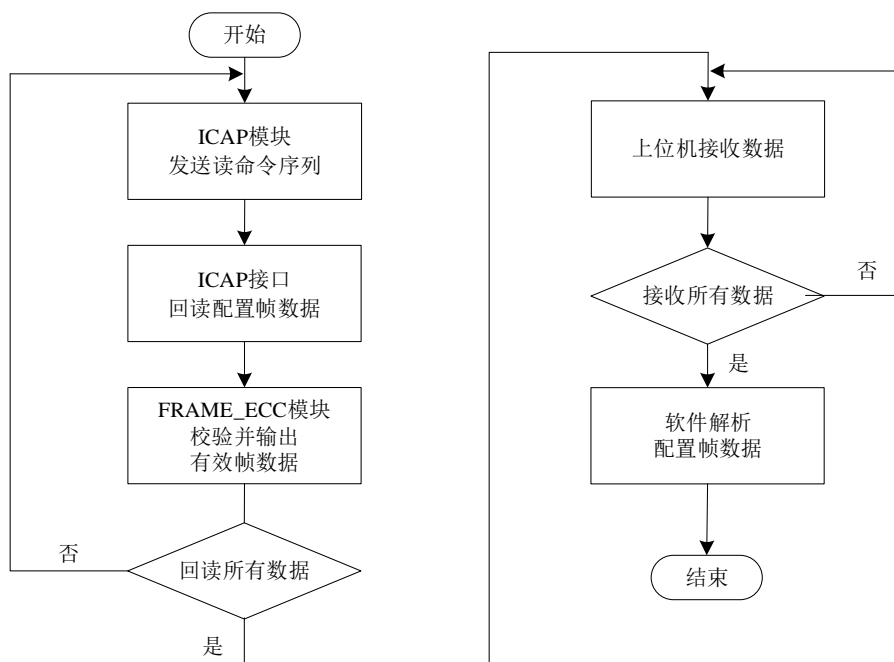


图 3-4 基于 ICAP 回读技术的帧结构解析流程图

本章设计一种针对 Kintex-7 系列 FPGA 配置 RAM 帧地址解析方法，其整体方案如图 3-3 所示。系统主要包括 ICAP 读写控制模块、FRAME\_ECC 模块。

ICAP 读写控制模块通过向 FPGA 内部的配置访问接口发送读写控制命令从而以帧为单位对配置存储器中的帧数据进行读取，读取的配置帧数据经过内部例化的原语 FRAME\_ECC 模块实现对配置 RAM 帧数据进行检错与纠错并输出帧物理地址。上位机接收输出的帧物理地址按照 FAR 寄存器中地址的格式进行整理分析后得到所需的配置 RAM 帧物理地址组织排布，具体流程如图 3-4 所示。

## 3.3 ICAP 读写控制模块设计

### 3.3.1 ICAP 功能描述

内部配置访问接口(Internal Configuration Access Port,ICAP)由赛灵思公司提供,用于 Kintex-7 FPGA 与外部器件进行数据交互,外部器件可通过 ICAP 接口对 FPGA 内部配置存储器进行写入与回读操作。用户利用基于 FPGA 的系统向设备发送读取的启动命令序列,设备在接收到读取配置内存的命令之后,将配置内存中的数据发送至 ICAP 接口。

用户通过控制读写控制信号低电平与高电平之间的转换,实现 ICAP 接口对 FPGA 内部配置存储区写入与读取操作的选择。如

图 3-5所示为ICAP接口状态切换时的时序关系图。需要注意的是,利用ICAP接口读取配置内存时,必须保证当前的状态为写入,FPGA 在接收到 ICAP 接口发送的读取启动命令序列以及帧地址数据后,再将 ICAP 切换为读取操作。

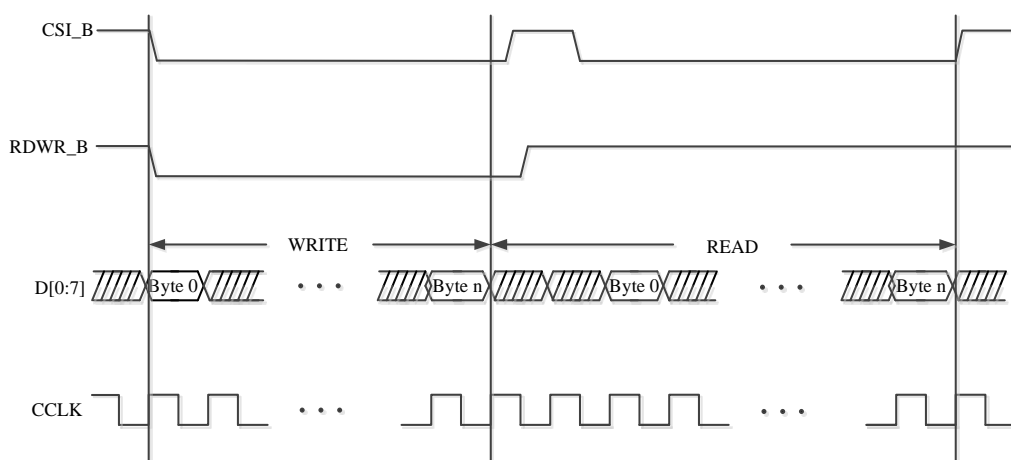


图 3-5 ICAP 接口时序关系图

需要特别注意的是，无论是通过 ICAP 接口向配置内存区域写入数据还是从配置内存区域中读取数据，都需要对数据进行位交换操作。如图 3-6 所示，显示了两个字节的数据（0xABCD）进行位交换的过程。

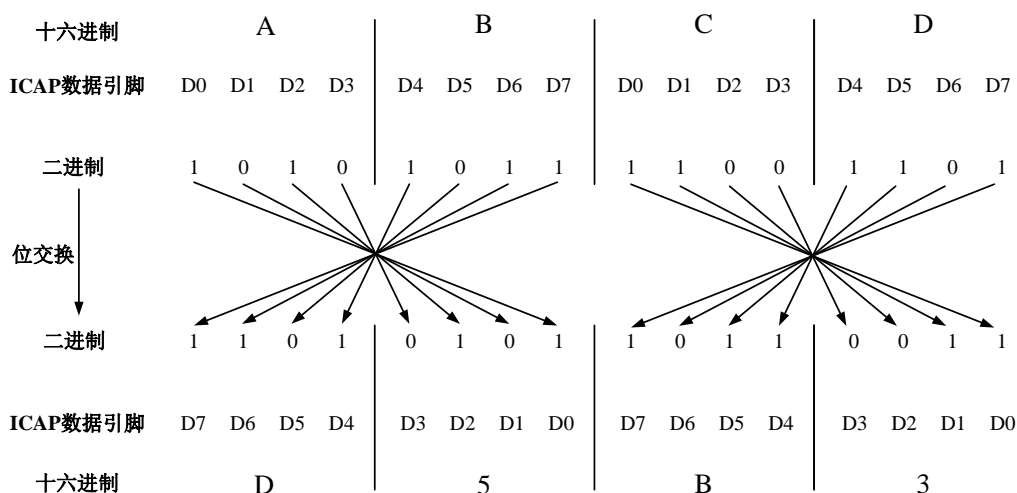


图 3-6 位交换操作示例

Kintex-7 系列 FPGA 的位交换以一个字节为单位进行操作，具体的位交换规则为：每个字节的最高有效位（Most Significant Bit,MSB）与最低有效位（Least Significant Bit,LSB）进行交换，次高位与次低位进行交换，以此类推。无论输入输出的数据方向如何，每个字节的 MSB 都会进入 D0 引脚：（1）如果数据进行位交换，则进入 D0 引脚的位是最右边的位；（2）如果数据不进行位交换，则进入 D0 引脚的位是最左边的位。数据是否进行位交换完全取决于应用程序。

在本论文中，通过 ICAP 接口向 FPGA 输入命令序列，然后通过 ICAP 接口读取 FPGA 配置内存中的帧地址数据，因为 ICAP 接口的位交换特性，所以在输入命令序列之前需要手动将命令序列进行位交换，同样读取的输出数据只有在经过位交换操作之后才是实际的帧地址信息。

### 3.3.2 Kintex-7 FPGA 配置寄存器读取

Kintex-7 系列 FPGA 所有的位流命令都是通过 ICAP 接口读取或写入配置寄存器来执行的。FPGA 的位流由两种包类型组成：包类型 1 和包类型 2。



## (1) 包类型 1

包类型 1 用来通过 ICAP 接口对 FPGA 配置寄存器进行读取和写入。包类型 1 包括包头和数据，在包头之后为包类型 1 的数据区域，包头区域的长度是 32 位，数据区域是以 32 位为 1 个字。表 3-2 所示为包类型 1 包头格式，Kintex-7 系列 FPGA 只使用了 14 位寄存器地址中的低 5 位，用于对配置寄存器中的相应地址进行数据的写入与回读操作，其余高 9 位为保留位，标记为“R”，本文设计中保留位用“0”表示；低 11 位用来设置 ICAP 接口写入或读取配置内存的具体数据个数；包类型固定为 001；操作码用来控制 ICAP 接口当前的操作，具体格式如表 3-3 所示。

表 3-2 包类型 1 包头格式

包类型	操作码	寄存器地址	保留位	字数
[31:29]	[28:27]	[26:13]	[12:11]	[10:0]
001	XX	R RRRR RRRR X XXXX	RR	XXX XXXX XXXX

表 3-3 操作码格式

操作码	功能
00	空操作
01	读操作
10	写操作
11	保留

## (2) 包类型 2

包类型 2 用来写长的数据块，由于其自身不具备地址位，因此不能单独使用，必须紧跟着包类型 1 之后发送数据，包类型 2 使用包类型 1 中的寄存器地址进行写入。表 3-4 所示为包类型 2 包头格式，与包类型 1 一致，包类型 2 的包头区域长度是 32 位，数据区域是以 32 位为一个字，其中低 27 位用于数据字计数。

表 3-4 包类型 2 包头格式

包类型	操作码	字数
[31:29]	[28:27]	[26:0]
010	RR	XXX XXXX XXXX XXXX XXXX XXXX XXXX

Kintex-7 系列 FPGA 开发板的配置寄存器如表 3-5 所示。不同类型的配置寄存

器具有不同的读写权限，并且各自对应独特的功能，用户能够通过 ICAP 的输入/输出端口向对应的地址中实现写入/读取数据的操作。其中帧地址寄存器（Frame Address Register, FAR）中存取配置帧所需的指定帧地址，Kintex-7 系列 FPGA 被分割成上下两个半区，帧地址寄存器的初始地址位于器件上半区中心最左侧，帧地址从 Minor Address 开始自动递增，接着依次从 Column Address、Row Address、Top/Bottom 和 Block Type 递增；帧数据输出寄存器（Frame Data Register, Output Register, FDRO）为只读寄存器，当从 FAR 寄存器中指定的地址开始读取配置帧的数据后，首先将回读的数据存放在该寄存器中；指令寄存器（Command Register, CMD）中存储与配置过程相关的指令，对全局信号进行控制并实现其他功能，每当帧地址寄存器中有新的数据输入时，就会执行指令寄存器中存储的指令，表 3-6 中列举了指令寄存器的指令和相应代码。

# 硕士学位论文

表 3-5 Kintex-7 系列 FPGA 配置寄存器

寄存器名	读	写	地址	说明
CRC	YES	YES	0 0000	CRC 寄存器，用来存放位流写入的 CRC 值
FAR	YES	YES	0 0001	帧地址寄存器
FDRI	NO	YES	0 0010	帧数据寄存器，输入（写配置数据）
FDRO	YES	NO	0 0011	帧数据寄存器，输出（读配置数据）
CMD	YES	YES	0 0100	指令寄存器
CTL0	YES	YES	0 0101	控制寄存器 0
MASK	YES	YES	0 0110	控制寄存器 0 和控制寄存器 1 的掩码寄存器。 在写控制寄存器之前，根据掩码寄存器的值来决定是否对控制寄存器中对应的位进行写操作
STAT	YES	NO	0 0111	状态寄存器
LOUT	NO	YES	0 1000	串行菊花链输出寄存器
COR0	YES	YES	0 1001	配置选择寄存器 0
MFWR	NO	YES	0 1010	多帧写寄存器
CBC	NO	YES	0 1011	初始化 CRC 值寄存器
IDCODE	YES	YES	0 1100	器件 ID 寄存器
AXSS	YES	YES	0 1101	用户位流存取寄存器
COR1	YES	YES	0 1110	配置选择寄存器 1
WBSTAR	YES	YES	0 1111	热启动地址寄存器
TIMER	YES	YES	1 0000	看门狗定时器寄存器
BOOSTER	YES	NO	1 0110	启动历史状态寄存器
CTL1	YES	YES	1 1000	控制寄存器 1
BSPI	YES	YES	1 1111	BPI/SPI 配置选项寄存器

## 硕士学位论文

表 3-6 指令寄存器

指令	代码	描述
NULL	0 0000	该指令为一个空指令，当该指令写入指令寄存器后，配置控制逻辑不进行任何操作，配置过程停留在当前状态等待下一条指令。
WCFG	0 0001	该指令为写配置数据指令，在将配置数据写入到 FDRI 寄存器之前使用。
MFWR	0 0010	该指令为多帧写指令，用在压缩位流模式下，将一帧数据写入到多个地址。
LFWR	0 0011	该指令用来复位全局信号 GHIGH_B，在写完最后一帧配置数据之后用来激活所有的 FPGA 内部逻辑的互联。
RCFG	0 0100	该指令为读配置数据指令，在将配置数据读出到 FDRO 寄存器之前使用。
START	0 0101	该指令为启动指令，出发启动序列，启动序列在循环冗余校验（Cyclic Redundancy Check,CRC）成功并且 FPGA 接收到了 DESYNC 指令后执行。
RCRC	0 0111	该指令用来复位 CRC 指令，用来复位当前 CRC 值。
AGHIGH	0 1000	该指令用来置位全局信号 GHIGH_B，在写配置数据之前断开 FPGA 内部逻辑的互联，使所有的互联处于一个高阻状态，这样能防止写配置数据是出现竞争。该指令主要用于关闭（SHUTDOWN）模式下的回读或者重配置。默认情况下，GHIGH_B 信号的初始值为 0。
GRESTORE	0 1010	该指令用来对 FPGA 内部触发器进行初始化操作，过程和回读抓取相反。在配置数据写完之后，通过发送 GRESTORE 指令，给全局置位复位（GSR）信号一个脉冲，将内部触发器所对应的配置存储器中的值写入内部触发器中，从而完成 FPGA 内部触发器的初始化。
DESYNC	01101	该指令为解同步指令，用来复位同步成功信号，在配置的结尾处用来对 FPGA 进行解同步操作，在解同步完成之后，配置数据接口的值将被忽略，配置位流禁止写入内部数据总线。

Kintex-7 系列 FPGA 配置寄存器与配置存储器存在的最明显的区别是：配置

寄存器以字为单位存放数据，每个字由 32 位组成；而配置存储器由帧组成，帧是最小的可寻地址单元，每一帧由 101 个字组成，每个字包含 32 个 bit。因此实现了通过 ICAP 接口向 Kintex-7 系列 FPGA 芯片发送命令序列的基础上，为了获取 FPGA 的具体帧数据，需要从配置存储器中对应的地址中读取，其中特别要注意的是，从 FDRO 寄存器中读取数据时需要经过 Buffer 缓存一个帧的数据，因此需要丢弃读取的首帧数据。

总线宽度自动检测模式被插入到命令序列的开头，它能够自动检测配置总线宽度。配置逻辑仅检测总线的低 8 位，对照命令序列中指定的数据位宽，配置逻辑能够自动调整外部总线宽度进行匹配。因为它出现在同步命令字之前，而一个特殊的同步命令字用于允许配置逻辑在 32 位字边界对齐，所以必须先成功检测总线宽度，才能检测同步命令字。

### 3.3.3 ICAP 读写控制模块实现

ICAP 读写控制模块主要用于将位流命令序列按照 ICAP 接口的时序要求发送给配置控制逻辑，从而实现对配置存储器中帧数据的读取与写入，其接口信号如表 3-7 所示，其中包含调用 ICAP 接口，读取配置内存时例化 ICAP 模块的信号；控制 ICAP 接口写入/读取操作的选择信号；ICAP 接口对 FPGA 配置内存实行相关操作时的输入输出信号。

表 3-7 ICAP 接口信号

信号名	接口类型	位数	描述
clk	输入	1	ICAP 接口时钟
reset	输入	1	复位信号，高电平有效
start_icap	输入	1	启动信号
mode	输入	1	模式选择信号；低电平为读，高电平为写
frame_addr	输出	32	读写指定的帧地址
data_valid	输出	1	回读数据有效信号
word_cnt	输出	8	读写数据计数

ICAP 接口电路如图 3-7 所示。上述命令序列字在经过配置逻辑实现位交换之后，按照指定的顺序存放于 CMD ROM 中；CS ROM 与 READ ROM 分别用于存

放对应的片选信号与读控制信号；Data Counter 记录了从配置存储器中读取的配置数据字数，每完成一次数据读取操作，总共需要接受 202 个字的数据，其中有效的数据只包含后 101 个字，Counter\_r 与 Counter\_w 计数器分别用于记录已经写入与读取的命令序列字数，ICAP 状态机控制这些命令序列的数值，并给出读取和写入操作时的 ROM 地址。ROM 中读取与写入的命令序列基址由 start 信号译码后产生，在与上述计数器的数值相加后得到对应操作命令序列在 ROM 中的实际地址。

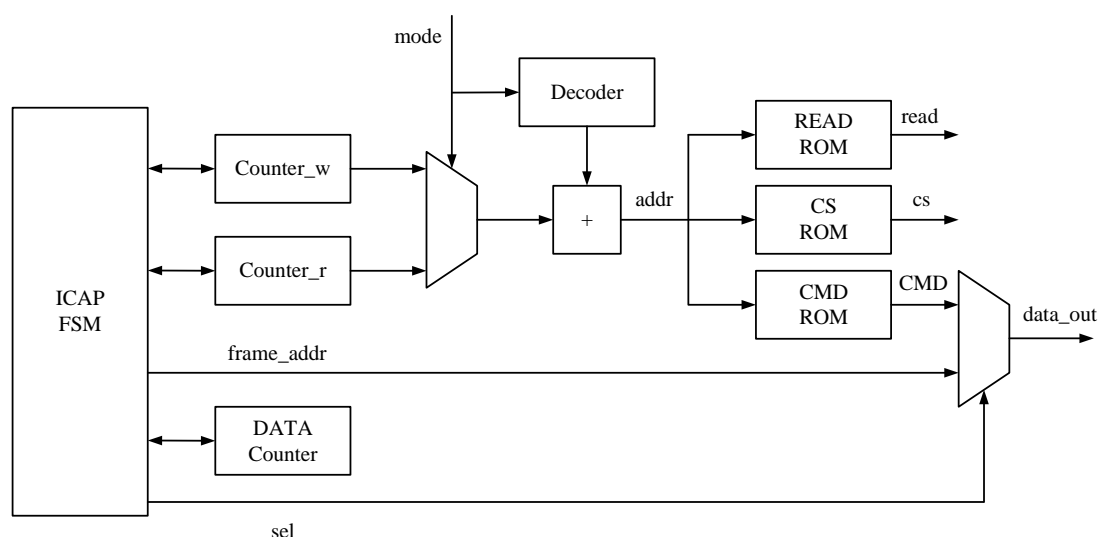


图 3-7 ICAP 接口电路

ICAP 接口读写控制模块的有限状态机转化关系如图 3-8 所示。处于初始状态时，判断 start\_icap 信号是否有效，同时控制器采样 mode 信号从而选择读取还是写入操作，当判断为读操作时，跳转到 READ\_CMD 状态，发送读取配置数据命令序列，当命令序列被接收并正确识别后，跳转到 READ\_DATA 状态等待接收读取的数据，完成后返回上一状态重新发送去同步命令序列，最后返回初始状态。当判断为写操作时，跳转至 WRITE\_CMD 状态并发送写操作命令序列，随后进入 WRITE\_DATA 状态向配置存储器中写入配置数据，完成后跳转回上一状态继续发送去同步等命令序列，最后返回初始状态，完成一次写配置存储器的操作。

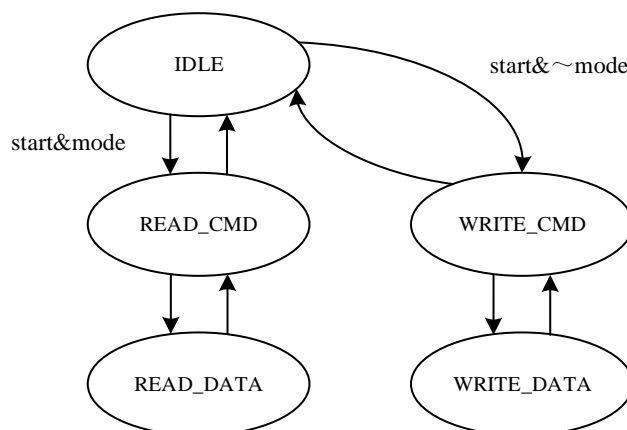


图 3-8 ICAP 读写控制模块状态转换图

## 3.3.4 功能验证

ICAP 接口读写控制模块功能上板结果如图 3-9 所示，图中为读配置存储器的结果。当外部给出激励信号帧地址信号 `frame_addr`、模式选择信号 `mode` 以及启动信号 `start_icap` 之后，开始进行读配置存储器操作。电路由初始状态 `IDLE` 跳转至 `READ_CMD` 状态，并通过输出信号 `data_out`、读信号 `read` 以及 `cs` 信号向 ICAP 接口发送读配置存储器命令，完成后跳转至 `READ_DATA` 状态等待接收从配置存储器中回读的帧数据，然后返回上一状态发送去同步命令，最后跳转回初始状态，完成一次读配置存储器操作。

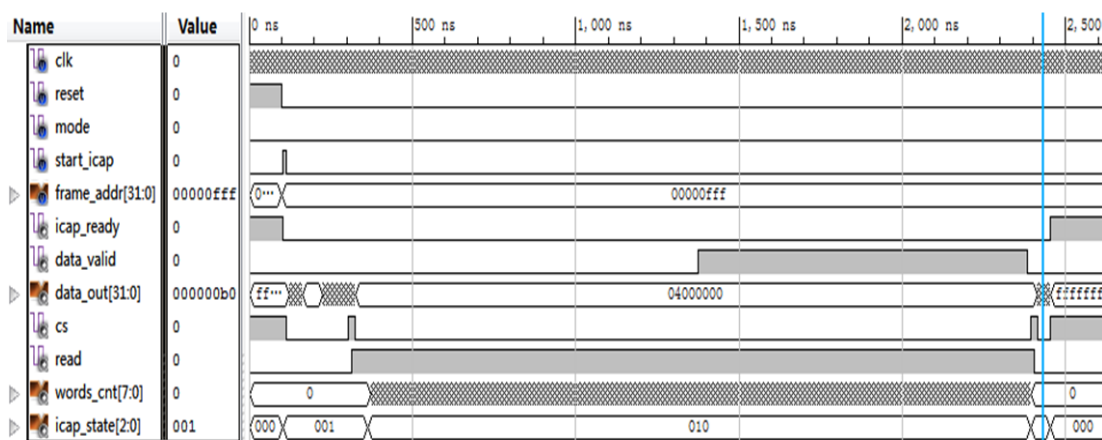


图 3-9 读配置存储器结果 1

当 ICAP 读写控制模块由写操作切换至读操作时，测试结果如图 3-10 所示。从图中能够清晰看出，时间处于 320ns 左右时，read 信号由低电平跳转为高电平，且当 cs 信号有效后，经过四个时钟周期的延迟，接收到第一个有效数据，计数器开始计数，并且之后每过一个时钟周期都会接收一个数据。当计数器达到 100 后，data\_valid 信号由低电平跳转为高电平，表示此时读出的配置存储器帧数据为有效数据，即之后的第 101 到第 201 个配置帧数据为有效数据。

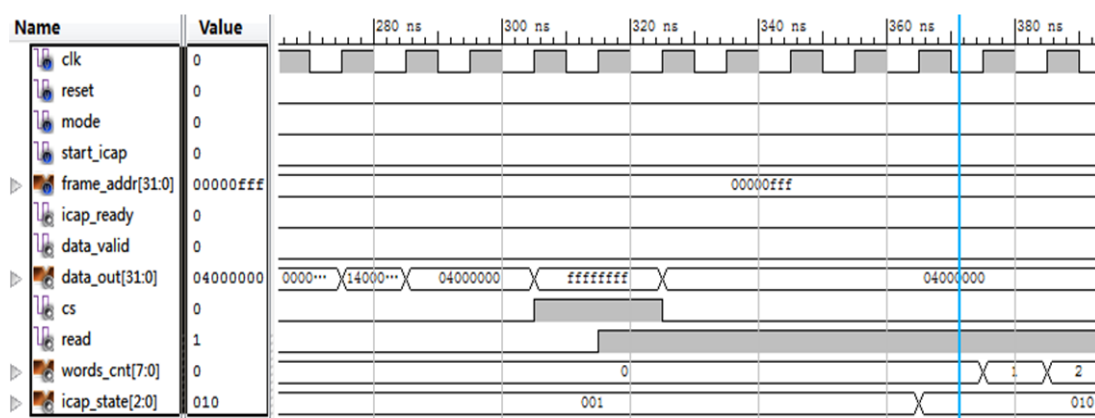


图 3-10 读配置存储器结果 2

## 3.4 帧地址解析模块设计

### 3.4.1 模块功能及实现

Kintex-7 系列 FPGA 内嵌了一种名为 FRAME\_ECC 的逻辑电路，其通过 13 位的汉明校验码（Hamming Code）能够检测从配置存储器中回读的数据是否出现至多 2bit 的错误，并对出现 1bit 的错误进行纠正。本课题主要采用的是 FRAME\_ECC 在回读配置数据的同时，能够输出 FAR 寄存器中的帧地址的功能。图 3-11 所示为 FRAME\_ECC 的对外接口。



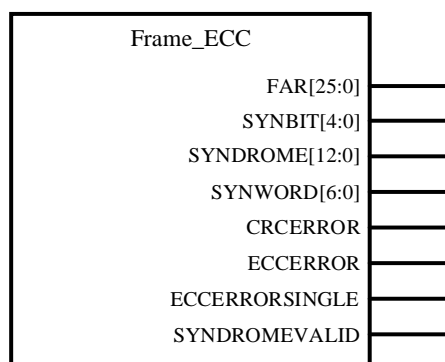


图 3-11 FRAME\_ECC 接口示意图

如前文所述，配置 RAM 中的帧地址并不是连续的。当帧解析模块完成一帧的检错后，需要对下一帧地址进行解析，如果只是简单的在初始地址的基础上进行加一，得到的下一帧地址可能是无效地址。因此为了正确地解析帧地址数据，需要首先确定有效帧地址的准确范围。

通过例化 FRAME\_ECC 的原语，在对每一帧配置数据回读的最后一个周期，输出当前配置帧的物理地址。如果将所有的配置数据尽数读取，就能按照顺序将所有帧的物理进行输出，得到有效帧地址的集合，从而能够确定配置数据的有效帧地址范围。获取有效帧地址范围的电路基本结构如图 3-12 所示。

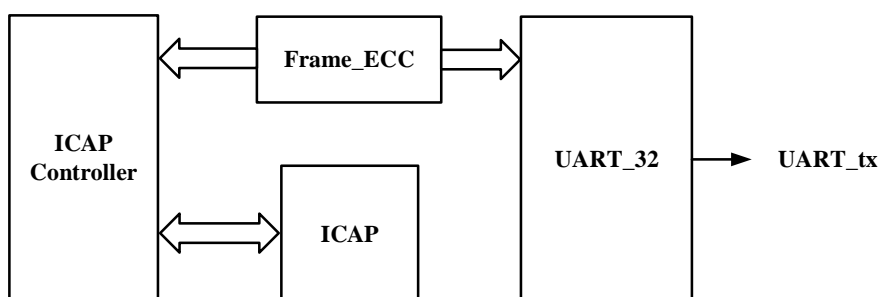


图 3-12 帧地址获取电路

### 3.4.2 帧地址结构解析及结果验证

ICAP 接口模块负责向 Kintex-7 系列 FPGA 发送回读命令序列，完成对配置存储器中帧数据的回读，每完成一次配置数据的回读，帧解析模块都会将配置帧

的物理地址通过 FAR 接口输出，UART\_32 模块将接收到的 32 位帧地址分成 4 次通过通用异步收发传输器(Universal Asynchronous Receiver/Transmitter,UART)传输到上位机中，上位机接收到部分数据如图 3-13 所示。

```

00 00 00 00 00 00 00 01 00 00 00 02 00 00 00 03 00 00 00 04 00 00 00 05 00 00 00 06
00 00 00 07 00 00 00 08 00 00 00 09 00 00 00 0A 00 00 00 0B 00 00 00 0C 00 00 00 0D
00 00 00 0E 00 00 00 0F 00 00 00 10 00 00 00 11 00 00 00 12 00 00 00 13 00 00 00 14
00 00 00 15 00 00 00 16 00 00 00 17 00 00 00 18 00 00 00 19 00 00 00 1A 00 00 00 1B
00 00 00 1C 00 00 00 1D 00 00 00 1E 00 00 00 1F 00 00 00 80 00 00 00 81 00 00 00 82
00 00 00 83 00 00 00 84 00 00 00 85 00 00 00 86 00 00 00 87 00 00 00 88 00 00 00 89
00 00 00 8A 00 00 00 8B 00 00 00 8C 00 00 00 8D 00 00 00 8E 00 00 00 8F 00 00 00 90
00 00 00 91 00 00 00 92 00 00 00 93 00 00 00 94 00 00 00 95 00 00 00 96 00 00 00 97
00 00 00 98 00 00 00 99 00 00 00 9A 00 00 00 9B 00 00 00 9C 00 00 00 9D 00 00 00 9E
00 00 00 9F 00 00 00 A0 00 00 00 A1 00 00 00 A2 00 00 00 A3 00 00 01 00 00 01 01
00 00 01 02 00 00 01 03 00 00 01 04 00 00 01 05 00 00 01 06 00 00 01 07 00 00 01 08
00 00 01 09 00 00 01 0A 00 00 01 0B 00 00 01 0C 00 00 01 0D 00 00 01 0E 00 00 01 0F
00 00 01 10 00 00 01 11 00 00 01 12 00 00 01 13 00 00 01 14 00 00 01 15 00 00 01 16
00 00 01 17 00 00 01 18 00 00 01 19 00 00 01 1A 00 00 01 1B 00 00 01 1C 00 00 01 1D
00 00 01 1E 00 00 01 1F 00 00 01 20 00 00 01 21 00 00 01 22 00 00 01 23 00 00 01 80
00 00 01 81 00 00 01 82 00 00 01 83 00 00 01 84 00 00 01 85 00 00 01 86 00 00 01 87
00 00 01 88 00 00 01 89 00 00 01 8A 00 00 01 8B 00 00 01 8C 00 00 01 8D 00 00 01 8E
    
```

图 3-13 有效地址数据

将接收的有效地址数据按照帧结构进行分析整理后，获取了 Kintex-7 系列 FPGA 的具体帧地址分布如图 3-16 所示。其主要由 Top 和 Bottom 两部分组成，Top 部分包含 4 行，Bottom 部分包含 3 行，每一行均有 103 列，而每一列中包含的配置帧数不固定，根据逻辑资源类型不同配置帧数目也不同。为了实现对配置 RAM 中帧地址实现故障注入功能，只需获取其中 Block Type 为 000 的帧数据，其中前 96 列由 CLB 列、CLK 列、IOB 列以及 BRAM 互联列组成，其余为 BRAM 内容列，本文只针对其中与配置 RAM 存储单元相关的帧数据集合进行研究，结合实验得到的有效帧数据集合，配置 RAM 资源共包含 21762 帧。

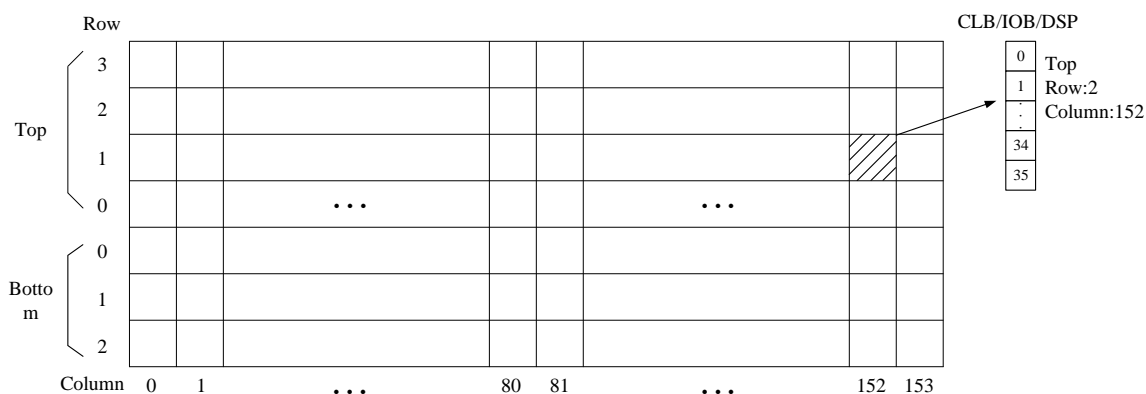


图 3-14 Kintex-7 系列 FPGA 帧地址组织方式

## 硕士学位论文

根据 Kintex-7 芯片用户手册可知，总共包含五种类型资源，各类型资源每列包含的配置帧数量不同：CLB 资源每列包含 36 帧、CLK 资源每列包含 28 帧、IOB 资源每列包含 42 帧、BRAM 互联资源每列包含 30 帧、BRAM 内容资源每列包含 128 帧。以上述内容以及帧地址寄存器含义为基础对配置帧数据进行二次结构解析，得到 Kintex-7 系列 FPGA 配置帧结构如表所示。

表 3-8 Kintex-7 系列 FPGA 每行的帧结构

资源类型	每列帧数	列个数	主地址
CLB	36	71	其他
CLK	28	12	其他
IOB	42	2	0、95
BRAM 互联	30	4	1、24、49、94
BRAM 内容	128	7	96-102

起点 1	帧长 8	步进 1	显示 等长帧十六进制
<input type="checkbox"/> no_Bram.bit			
21741	00 00 00 00 00 00 00 00		
21742	00 07 D4 00 00 00 00 00		
21743	00 00 00 00 00 00 00 00		
21744	00 00 00 00 00 00 00 00		
21745	00 00 00 14 82 00 00 02		
21746	00 20 00 14 82 00 00 02		
21747	00 20 00 14 82 00 00 00		
21748	02 20 00 14 82 00 00 00		
21749	02 00 02 14 82 00 00 02		
21750	02 28 40 00 01 40 40 00		
21751	00 00 00 00 00 00 00 00		
21752	00 00 2A 00 00 00 00 00		
21753	00 00 00 00 00 00 00 00		
21754	00 00 00 00 00 00 00 00		
21755	00 00 00 00 00 00 00 00		
21756	00 00 00 6A 00 00 00 00		
21757	00 00 00 00 00 00 00 00		
21758	00 50 00 00 00 00 00 00		
21759	00 00 00 00 00 22 00 00		
21760	00 00 00 00 00 00 00 00		
21761	00 00 00 00 00 00 00 00		
21762	00 08 00 00 2E 00 00 4C		
21763	00 00 00 00 00 00 00 00		
21764	00 00 00 00 00 00 00 00		
21765	00 00 00 00 00 00 00 00		
21766	00 00 00 00 00 00 00 00		
21767	00 00 00 00 00 00 00 00		
21768	00 00 00 00 00 00 00 00		
21769	00 00 00 00 00 00 00 00		
21770	00 00 00 00 00 00 00 00		

起点 1	帧长 8	步进 1	显示 等长帧十六进制
<input type="checkbox"/> Bram.bit			
21741	00 00 00 00 00 00 00 00		
21742	00 07 D4 00 00 00 00 00		
21743	00 00 00 00 00 00 00 00		
21744	00 00 00 00 00 00 00 00		
21745	00 00 00 14 82 00 00 02		
21746	00 20 00 14 82 00 00 02		
21747	00 20 00 14 82 00 00 00		
21748	02 20 00 14 82 00 00 00		
21749	02 00 02 14 82 00 00 02		
21750	02 28 40 00 01 40 40 00		
21751	00 00 00 00 00 00 00 00		
21752	00 00 2A 00 00 00 00 00		
21753	00 00 00 00 00 00 00 00		
21754	00 00 00 00 00 00 00 00		
21755	00 00 00 00 00 00 00 00		
21756	00 00 00 6A 00 00 00 00		
21757	00 00 00 00 00 00 00 00		
21758	00 50 00 00 00 00 00 00		
21759	00 00 00 00 00 22 00 00		
21760	00 00 00 00 00 00 00 00		
21761	00 00 00 00 00 00 00 00		
21762	00 08 00 00 2E 00 00 4C		
21763	00 00 00 00 00 00 00 00		
21764	00 00 00 00 00 00 00 00		
21765	C2 00 64 00 00 86 86 AA		
21766	BE 8C A0 00 00 00 00 00		
21767	00 00 60 BE 00 00 00 00		
21768	86 9E 9A A0 00 00 A4 00		
21769	00 00 00 00 00 00 00 8A		
21770	A6 A6 76 AA E6 CA E4 92		

a)无 BRAM 内容资源

b)包含 BRAM 内容资源

图 3-15 Kintex-7 系列 FPGA 配置位流文件

---

$$= 3096 \times 7 = 21762$$

(3-2)

图 3-16 Kintex-7 系列 FPGA 帧地址分布

## 3.5 本章小结

在SRAM型FPGA多种类型的存储单元中，配置RAM存储单元数量占比最高，发生单粒子翻转的概率最大，且其中存储的数据决定了用户电路的逻辑功能，一旦配置RAM的存储单元发生单粒子翻转可能导致电路设计功能异常甚至失效。因此本章主要对Kintex-7系列FPGA中的配置RAM展开研究。

针对 Kintex-7 系列 FPGA 配置存储器中帧结构未知的问题，对配置存储器中的帧物理组织开展研究，提出了解析配置帧结构的方法，并给出配置比特流中的帧地址排列顺序。通过内部配置访问接口 ICAP 回读 FPGA 中帧地址寄存器，利用帧解码器 FRAME\_ECC 在回读配置帧数据时将具体的物理帧地址输出。获取所有的帧地址数据后，按照不同的帧地址类型进一步解析整理得到 Kintex-7 系列 FPGA 器件具体的帧地址分布，提取出其中与配置 RAM 资源对应的配置帧。

### 4 基于基本位的 FPGA 单粒子翻转定向注入技术

第三章中介绍了 SRAM 型 FPGA 中最主要的存储单元配置 RAM，由于其在 FPGA 中巨大的数量占比以及决定用户电路功能的重要地位，提出了一种将其从配置帧数据中提取出来的方法。本章将介绍如何通过故障定向注入的方法对 Kintex-7 系列 FPGA 中目标区域实现单粒子翻转。

本章在获取配置比特流中配置 RAM 所在的配置帧数据后，为进一步解决 Kintex-7 系列 FPGA 帧地址数据数量庞大的问题，借助基本位技术解析、分离其中感兴趣区域的帧数据，最后提出一种将其翻译成故障注入器能够识别的帧地址的方法，利用 SEM IP 故障注入器完成故障的定向注入，并与随机故障注入方法进行比对测试。

#### 4.1 Kintex-7 FPGA 用户电路基本位提取

在上面一章中，通过内部配置访问接口 ICAP 完成了对 Kintex-7 系列 FPGA 配置存储器中配置帧数据的回读，经过进一步的解析整理，获取了具体的帧地址分布。传统的全局故障注入方法会对配置存储器中所有的配置帧进行遍历式的故障注入，在每一帧数据单粒子翻转故障注入时间固定的情况下，平均故障时间与配置数据帧的数量成正比，因此通过减少配置帧故障注入的数量能有效降低故障注入的总耗时。实际上，FPGA 器件配置存储器中并不是任何配置位发生单粒子翻转都会对设计电路的逻辑功能造成干扰，本节中将与设计电路中逻辑功能紧密联系的配置位定义为基本位(Essential Bits)，同时包含基本位的配置帧则为关键帧。关键帧中的配置数据与设计电路中可配置逻辑资源 CLB 紧密相关，当关键帧中出现单粒子翻转时，将会影响设计电路的正常功能，严重时甚至破坏硬件结构。本小节介绍一种赛灵思公司提供的基本位技术，通过对用户设计中的层次感兴趣区域进行定义并识别与定义的用户逻辑相关的基本位，从而减少有效故障时间(Failures In Time,FIT)，缩小故障注入的范围，实现定向故障注入的目的。

### 4.1.1 基本位技术概述

Kintex-7 系列 FPGA 中的每个可配置资源都由一个或多个配置内存定义，配置内存被组织为一组帧，很像一个范围宽广的静态 RAM，因为大多数 FPGA 中的配置位数都很高，范围从几兆到超过千兆，例如，Kintex-7 系列 FPGA 具有 2 896 320 个配置位，所以将配置内存中的所有位进行故障注入，会花费巨大的时间成本，更加重要的是，并不是所有配置位上发生的单粒子翻转效应都会导致设计电路的逻辑功能发生改变，只有配置编程到 FPGA 中的特定应用程序使用的资源的这些位的子集才是相关的，这些位被称为基本位（Essential Bits）。

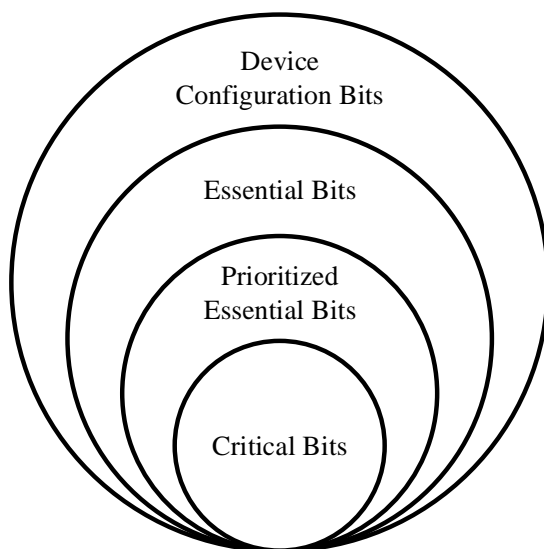


图 4-1 配置位、基本位、优先基本位、关键位关系图<sup>[49]</sup>

忽略 FPGA 中未使用部分资源上可能引发过电流或造成其他干扰的最终短路结构电路，因为任何非基本位中的 SEU 都不太可能产生功能错误或故障。此外，由于逻辑掩蔽、时间掩蔽或对应用程序的有效数据输入，即使其中一些基本位上发生 SEU 也可能不会产生功能错误或故障。为了增强通过基本位技术提供的收益，赛灵思提供了一种优先过滤基本位列表的方法，该方法允许用户根据用户定义的设计感兴趣的层次区域来优先过滤基本位文件，使用这种方法过滤得到的比特位称为优先基本位（Prioritized Essential Bits）。然而，对于特定的应用程序存在一些配置内存位，它们发生 SEU 肯定会导致功能错误或故障，这些位被称为关键位

(Critical Bits)。对于典型的电路设计，预计只有 5%至 10%的基本位是关键位。  
图 4-1 显示了上述配置存储器位类别之间的关系。

## 4.1.2 基本位实现流程

创建和使用优先基本位技术的流程如图 4-2 所示。设计流程中的步骤如下：

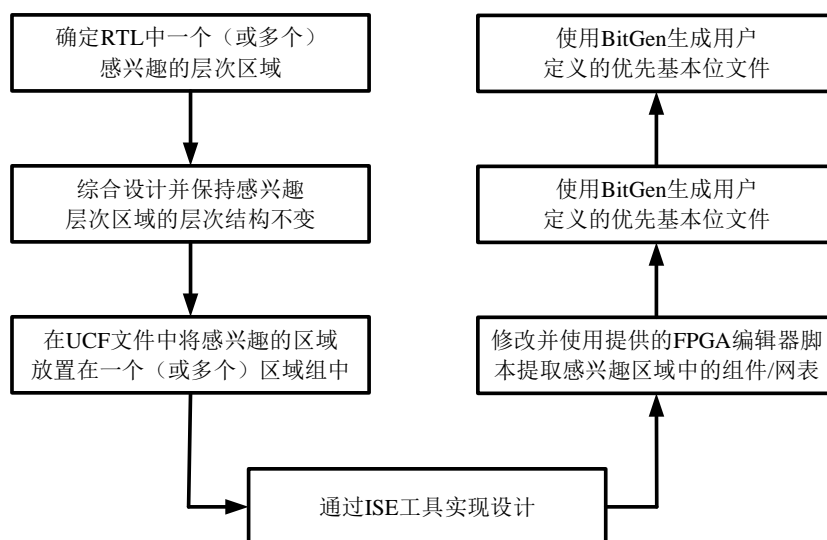


图 4-2 基本位技术设计流程

### (1) 确定 RTL 中感兴趣的层次区域并综合设计

在 RTL 级别代码中，用户对电路设计中起到关键性作用的某几个设计层次区域进行识别，然后对 RTL 进行分区，从而将这些区域划分至单独的模块中。识别和划分取决于用户设计和 SEU 的缓解要求，这是流程中最重要也是最困难的一步。

为了将用户设计中感兴趣的层次区域与其他非关键层次区域明显的区别开来，需要在设计中添加 RTL 级的约束(Keep Hierarchy)。该约束与 RTL 设计中指定的分层块（Verilog 模块）相关。如果在综合过程中保持层次结构，ISE 设计工具将使用 Keep Hierarchy 在整个实现过程中保持层次结构。

如图 4-3 所示，假设用户电路设计中的顶层模块中存在 I1、I2 和 I3 三个子模块，其中，只有 I2 模块是用户感兴趣的模块，为了将它与另外两个模块区分开来，需要使用基本位技术，首先将三个模块分别独立在各自的层次区域中，然后通过赛灵思工具（XST）中如图 4-4 所示语法生成 I2 模块的独立的基本位信息文件。



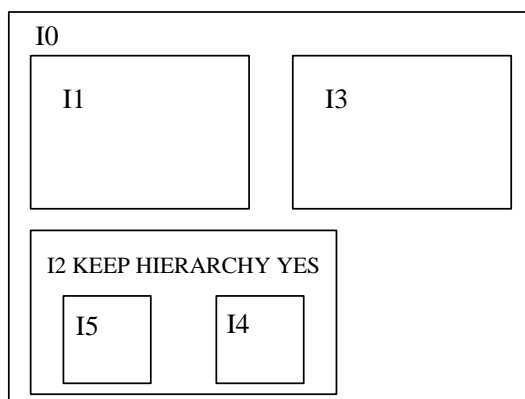


图 4-3 层次结构示意图

VHDL Syntax:

```
attribute keep_hierarchy : string ;
```

```
attribute keep_hierarchy of instant_name : label is "yes"
```

Verilog Syntax :

```
(* keep_hierarchy = "yes"*)
```

图 4-4 赛灵思综合工具层次约束语法

## (2) 为感兴趣的层次区域创建区域约束

在上一步的基础上，将确定的感兴趣区域进一步放入一个或多个 AREA GROUPs 中，作为主要的电路设计中指定区域的逻辑配置方法，AREA GROUPs 能够将用户感兴趣的区域与其它区域进行隔断，该方法通过将设计区域定义为物理区域的方式，来对感兴趣的区域作映射、打包、布局和布线处理。

用于定义 AREA GROUPs 的用户约束文件（UCF）语法如图 4-5 所示：

```
INST "X" AREA_GROUP = groupname;  
AREA_GROUP "groupname" RANGE = range;  
AREA_GROUP "groupname" GROUP = CLOSED;  
AREA_GROUP "groupname" PLACE = OPEN;
```

图 4-5 用户约束文件（UCF）语法

其中，“X”为 INST，表示的是 RTL 代码中定义的区域名称。“groupname”表示为用户设计中逻辑区块的名称，该逻辑区块将由打包工具关联在一起，并由布局工具定位在区域分组内。特别注意的是，设计中毫无关联的逻辑需要通过语句 AREA\_GROUP 中的 GROUP=CLOSED 来禁止工具进行打包，如果需要将无关的

逻辑布局在区域中需要利用语句中的 PLACE=OPEN。

用户约束文件创建完成后，利用 ISE 软件实现设计从而生成布局布线网表。

### (3) 修改提供的 FPGA 编辑器脚本，提取感兴趣区域组件

通过 ISE 设计工具在标准流程下完成设计后，得到脚本文件 essential.scr，为了满足脚本用于选择感兴趣区域中的组件和网表的要求，需要通过运行指令'fpga\_edline routed.ncd mapped.pcf -p essential.scr'来对脚本进行修改，从而达成使用设计中感兴趣区域的目的。

例如，设计提供的 FPGA 编辑器脚本的内容如图 4-6 所示：

```
clear
select -k comp *example_mon/example_mon_piso/*
select -k comp *example_mon/example_mon_siso/*
select -k net *example_mon/example_mon_piso/*
select -k net *example_mon/example_mon_siso/*
list
clear
exit
```

图 4-6 FPGA 编辑器脚本内容

在该脚本中，将选择已经识别的感兴趣区域的组件和网表。其中感兴趣的组件和网表分别为：`*example_mon/example_piso/*`与`*example_mon/example_siso/*`。完成上述操作后会生成一个日志文件，其中清晰地列举了设计中用户选择的感兴趣的区域组件和网表。

### (4) 使用 BitGen 生成按优先级排列的基本位文件。

得到包含感兴趣区域中组件和网表的日志文件后，BitGen 接收创建的日志文件，并生成 EBD 文件，其中仅包含被识别为优先基本位的位。

调用 BitGen 的命令语句如图 4-7 所示：

```
-g essentialbits : yes
-g essentialbitsfilter :
{none|mask|enable}
-g essentialbitsfilterfile : <filename>
```

图 4-7 BitGen 命令语句

其中, filename 为上一步骤生成的日志文件中的感兴趣的组件或网表名称。如果 essentialbitsfilter 设置为 enable, BitGen 的操作将生成与设计实体相关的基本位, 这些实体的名称仅与 essentialbitsfilterfile 选项中指定的名称相匹配。如果 essentialbitsfilter 设置为 mask, 生成的基本位将排除名称与 essentialbitsfilterfile 选项中指定的名称匹配的设计实体相关的位。如果 essentialbitsfilter 设置为 none, 则生成的基本位包含与整个设计相关的位。

与包含设计所有位的位流文件不同, EBD 文件中仅包含一个完整设计中的基本位部分, EBD 文件中的数据是由 ASCII 字符集编码的字符组成的文件, 每行有 32 个非“0”即“1”的字符组成, 其中,“0”代表设计中的非基本位,“1”代表设计中的基本位, 换句话说, EBD 文件包含与设计相关联的设备配置位的子集, 因此可以执行可配置逻辑块中的注入和 FPGA 的布线。薛晓良通过对 ISE 工具 BitGen 中的选项进行不同的设置, 分别获取了包含配置单元内容的.ebc 文件及其掩码文件.ebd 和仅包含回读数据的.rbd 文件, 对比.rbd 文件与.ebc 文件后, 发现两个文件中 Block Type 为 0 的部分完全一致<sup>[50]</sup>。然而关键的问题是, EBD 文件只是一个简单的“0”和“1”的集合, 因此需要一种方法来提取文件中的关键字符并将其转换为目标器件的注入地址。接下来, 将介绍一个能够提供设计子模块相关基本位的工具—自动配置内存故障注入, 利用它我们就可以针对想要测试的特定部件进行故障注入。

### 4.2 自动配置内存故障注入

自动配置内存故障注入目的是将从基于 SRAM 的 FPGA 中感兴趣区域中提取的配置内存基本位进一步做处理, 将其转换为赛灵思 SEM IP 控制器的注入地址。通过这种方式, 当被测设计 (Device Under Test,DUT) 是庞大系统的一部分时, 例如微处理器的翻译查找缓冲区 (Translation Lookaside Buffer,TLB), 自动配置内存故障注入可以使用 SEM IP 将故障注入该系统指定的区域中。自动配置内存故障注入方法实现单粒子翻转定向故障注入的流程如图 4-8 所示

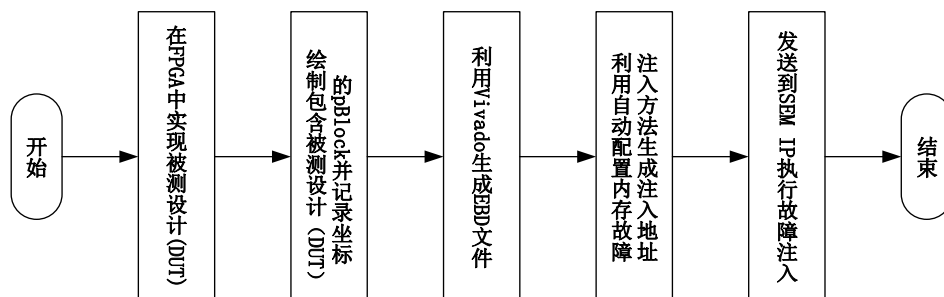


图 4-8 单粒子翻转定向故障注入流程

## 4.2.1 自动配置内存故障注入概述

软错误缓解（Soft Error Mitigation, SEM）IP 为赛灵思公司自主设计的一个模块，通过对 FPGA 中的配置存储区不间断地进行扫描，快速检测出其中存储单元值出现的变化并使用纠错码（Error-Correcting Codes, ECCs）及时对已经出现单粒子翻转的存储单元的值进行纠正，从而保护其生产的 7 系列或 UltraScale 系列 FPGA，防止其发生单粒子翻转效应。此外它还具有将错误注入配置内存中指定帧的某一位的功能。如图 4-9 所示，说明了将自动配置内存故障注入方法集成到 FPGA 设计的可靠性分析中，其中 SEM IP 用作故障注入器。

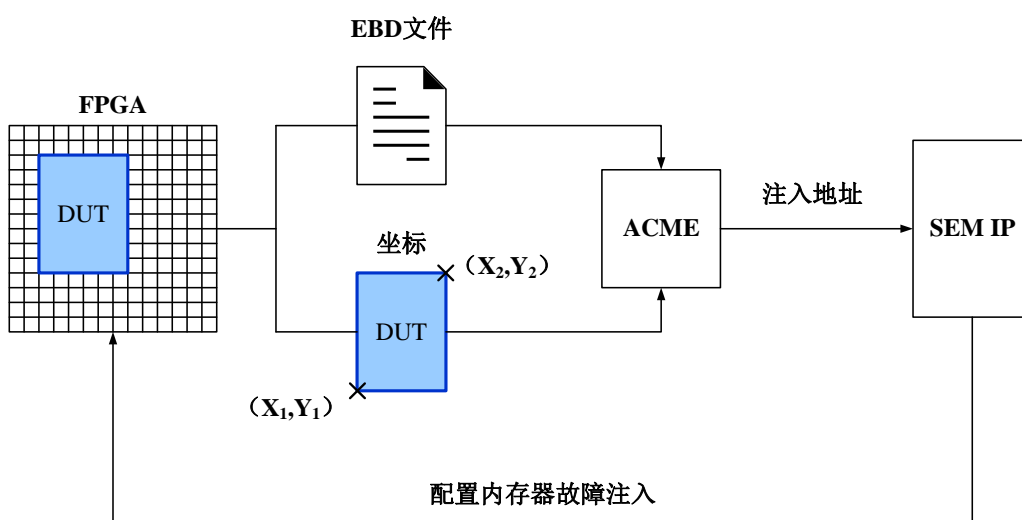


图 4-9 自动配置内存故障注入集成到 FPGA 设计中

从图中可以明显看出，生成 SEM IP 的故障注入地址需要具备两个前提条件；

1) 包含比特文件中所有基本位的 EBD 文件；2) FPGA 区域中被测设计所在的坐标。将这两个条件作为输入，自动配置内存故障注入能够从 EBD 文件中提取与被测设计基本位对应的文件行，然后通过实现赛灵思公司提供的公式将其转换为 SEM IP 的注入地址。这些公式将在后续小节进行详细解释。

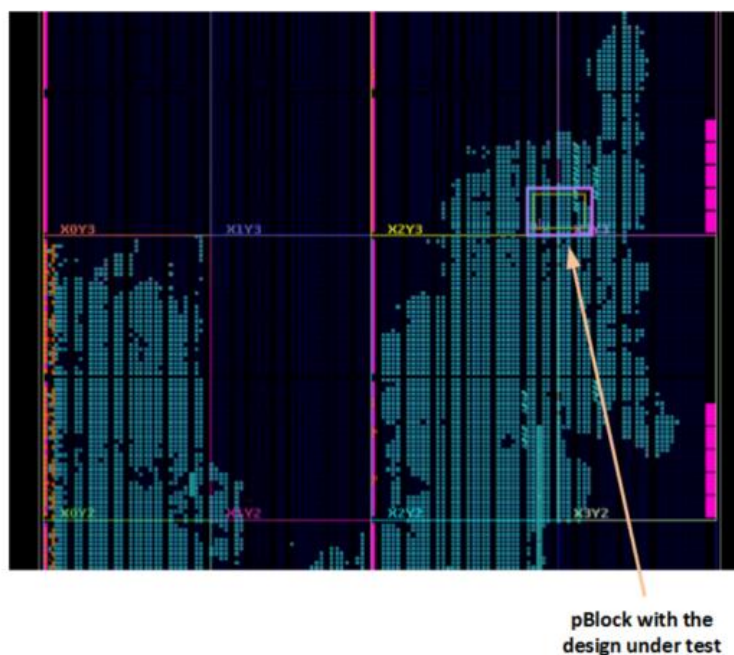


图 4-10 floorplan 中包含 DUT 的 pBlock

假设我们需要将故障注入基于 SRAM 型 FPGA 中某个感兴趣的区域中，例如微处理器上的 TLB 模块所在的区域，必须获得其所在的物理地址，进而利用 SEM IP 完成故障注入。首先，为了确定 TLB 模块使用的资源数量，必须成功地将微处理器部署到指定的 FPGA 目标体系结构中，然后，如图 4-10 所示，在利用 PlanAhead 工具生成的 FPGA 的 floorplan 视图中绘制出将 TLB 模块相关资源包围的放置块（Placement Block, pBlock）。

pBlock 是一个手动布局的矩形区域，旨在对 FPGA 中特定区域的逻辑单元进行面积约束，并将对应的逻辑单元进行区分后部署到 FPGA 的物理区域中。pBlock 的矩形区域大小限定了其中包含的 FPGA 逻辑资源的数量；pBlock 的矩形区域位置规定了其在 FPGA 物理区域的实际位置；pBlock 矩形区域中使用的 FPGA 逻辑资源种类限定了其所包含的逻辑资源种类。通过在 floorplan 视图中限

定两组坐标，绘制出 pBlock 在 FPGA 中的具体位置，并将选取的坐标进行记录，作为后续自动配置内存故障注入方法的输入参数。利用记录的两组坐标，自动配置内存故障注入方法能够从 EBD 文件中提取与 pBlock 限定的 FPGA 指定区域相对应的配置内存基本位。值得一提的是，利用 pBlock 将那些不属于目标模块的资源排除在外，是非常明智的，通过这种方式，避免了将故障注入其他无关模块中，防止出现干扰，同时也减少了故障注入耗费的时间。获取生成的 EBD 文件与记录的 pBlock 坐标后，可以利用自动配置内存故障注入将两者作为输入参数来获取 TLB 模块的注入地址，将该地址发送到 SEM IP 以实现故障注入。

## 4.2.2 生成故障注入地址

自动配置内存故障注入方法在使用的过程中，其内部主要的动作包括以下两部分：1) 从 FPGA 区域至 EBD 文件领域的转换；2) EBD 文件转换为注入地址。

第一步是从 EBD 文件中提取与被测设计相对应的基本位所在的行，如上一节所述，其中需要利用 pBlock 记录的坐标，自动配置内存故障注入方法基于 FPGA 区域和 EBD 文件行之间存在的线性关系来实现这一步骤。通过这种方式，每个 FPGA 时钟域都可以由线性斜率截距方程定义，该方程使用 pBlock 的坐标来求取其 EBD 行。

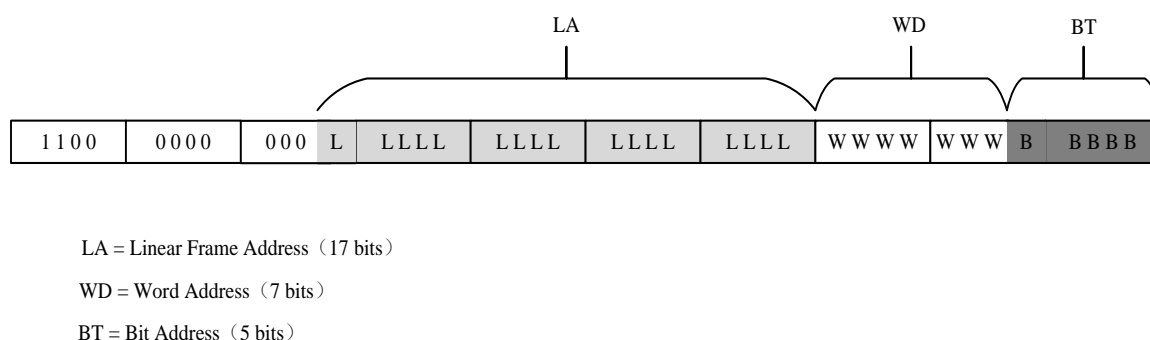


图 4-11 故障注入命令

第二步中，在获得了目标模块对应的 EBD 文件行基础上，通过赛灵思公司为其 7 系列和 UltraScale 系列 FPGA 开发的一些方程式，将其进一步转换为 SEM IP 的故障注入地址。每个故障注入地址必须通过其监视器接口发送到 SEM IP，监视

器接口通过 USB 转 UART 转换器与用户管理的计算机中的串行终端进行连接。当接口与上位机之间建立了连接，就可以向 SEM IP 发送不同的指令来控制其完成相应的动作，其中一条指令就是控制 SEM IP 完成故障注入。该指令的结构如图 4-11 所示。

通过故障注入命令，明确指定 SEM IP 在特定位置完成位翻转操作，因此有必要了解配置存储器中的线性帧地址分布。如前文所述，EBD 文件中的数据是由 ASCII 字符集编码的字符组成的文件，每一行都由 32 个非“0”即“1”的字符组成。每一行代表的是所选设备的配置内存帧的一个字。对于 Kintex-7 系列 FPGA 而言，每个配置帧由 101 个字组成，即 EBD 文件的 101 行。因此，利用公式(4-1)、公式(4-2)、公式(4-3)能够分别获取 17 位线性帧地址 (Linear Frame Address, LA)、7 位字地址 (Word Address, WD) 以及 5 位位地址 (Bit Address, BT)。

$$LA = (EBD\ line - (EBD\ line\ mod\ 101))/101 \quad (4-1)$$

$$WD = EBD\ line\ mod\ 101 \quad (4-2)$$

$$BT = 31 - character\ position \quad (4-3)$$

其中，*EBD line* 是 EBD 文件中基本位所在行的行号，*character position* 是每个基本位在文件中所处的位置，*mod* 代表的是模运算。如前文所述，每个 EBD 行都由 32 个非“0”即“1”的字符组成，这些字符都是一个字中的 32 个位，其中最低有效位 (Least Significant Bit, LSB) 位于行的最右侧。因此，通过将 31 减去 EBD 行中的每个字符‘1’就能轻易地获得 BT。

接下来将通过示例来详细说明如何获取如图 4-11 所示的故障注入命令。我们假设在 EBD 文件的第 134 行中有以下 32 个字符：1000 0000 1000 0000 0000。由于该 EBD 行的 32 个字符中存在两个‘1’，即两个基本位，因此我们将获得两个故障注入命令。通过公式(4-1)与公式(4-2)，计算得到  $LA = 0$ ， $WD = 33$ （二进制为 0100001）。然后，我们需要计算 BT 中的 5 位来完成每个故障注入命令。第一个‘1’位于最高有效位 (Most Significant Bit, MSB) 位置，即 *character position* = 0，因此  $BT = 31$ （二进制为 11111）；第二个‘1’位于 *character position* = 8 的位置，因此  $BT = 23$ （二进制为 10111）。最后，将 WD 的 7 位和 BT 的 5 位连接起

来，就可以得到如下十六进制数：010000111111=0x43F、010000110111=0x437。按照图 1-1 所示的命令格式，将上述获得的两个十六进制数代入后，就得到了完整的故障注入命令，十六进制表示为 C00000043F 和 C000000437，通过这两个命令就能将故障注入用户设计中这两个基本位所在的部分。

结合本小节开头解释的步骤 1) 中的内容和前面的方程式，就能生成 SEM IP 方法的故障注入地址，从而完成仅仅对属于设计中特定子模块的那些基本位中注入故障，达到定向故障注入的目的

## 4.3 定向故障注入测试及结果分析

### 4.3.1 定向故障注入测试环境

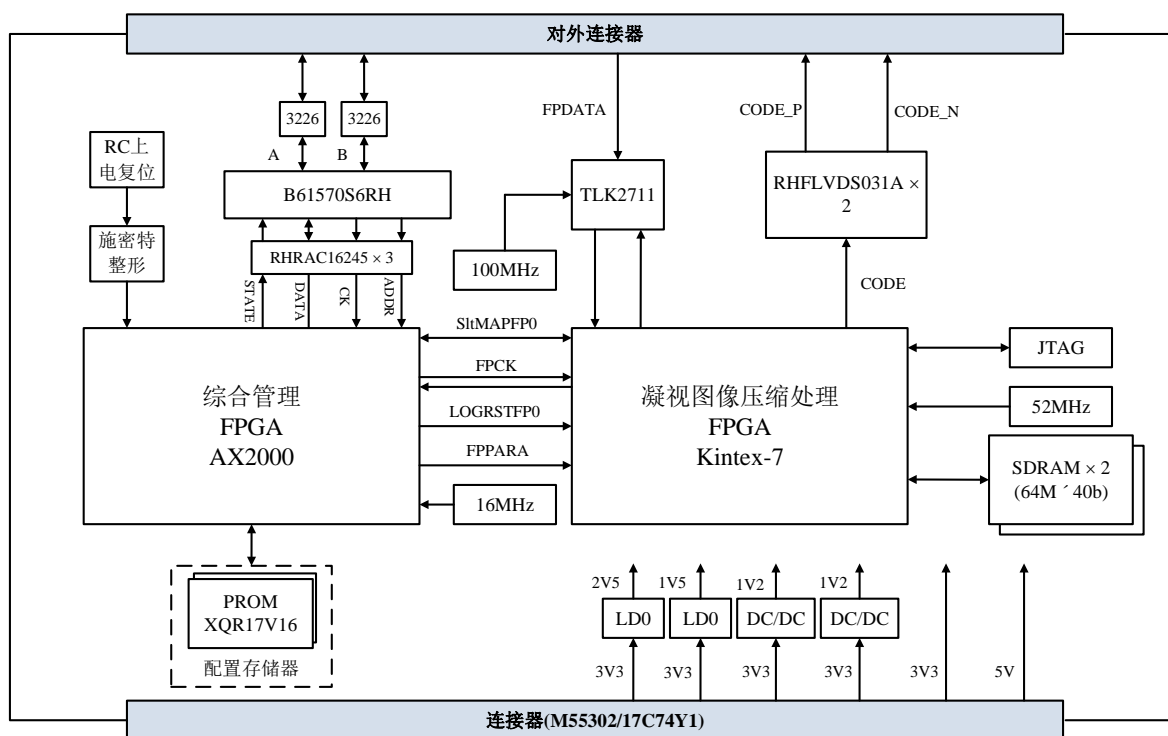


图 4-12 Kintex-7 系列 FPGA 星载图像处理平台硬件框架



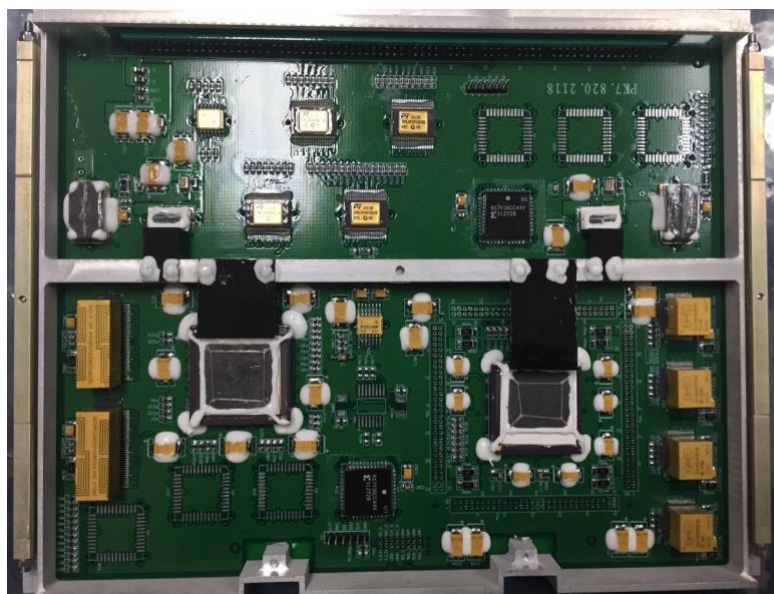


图 4-13 Kintex-7 系列 FPGA 星载图像处理平台实物图

测试实验采用课题组自行开发研制的基于 FPGA 的星载图像处理平台，如图 4-12 与图 4-13 所示，该平台接口丰富、功能完善，符合本课题实验要求。平台上主要搭载了两片 FPGA，一片为赛灵思公司生产的 Kintex-7 系列 FPGA，主要用于负责图像压缩处理工作；另一片为 Actel 公司生产的反熔丝型 FPGA，主要用于实现对 Kintex-7 系列 FPGA 的控制与监测功能。

地面测试平台包括：

(1)PC 端测试平台：基于 VS+QT 开发的数据压缩解码软件，实现与 FPGA 之间的串口通讯，提供对压缩码流的实时解码功能，解码过程中进行实时图像显示和解码信息输出；

(2)JTAG 调试器：用于实现 PC 端 FPGA 开发工具对 FPGA 的配置；

(3)impact 工具：向 FPGA 中加载配置位流文件。

### 4.3.2 定向故障注入结果及分析

为了验证本文提出的单粒子翻转定向故障注入方法的可行性与优势，在基于 Kintex-7 FPGA 的星载图像处理平台中的帧头帧尾检测模块、帧间压缩模块、组帧模块三个用户电路进行验证，并与传统单粒子翻转注入方法<sup>[33]</sup>进行对比测试。通过与不使用自动配置内存故障注入方法的场景进行比较从而显示自动配置内存

故障注入提供的高效率以及对于注入的控制作用的优势。

### (1) 本文定向故障注入方法

①**帧头帧尾检测模块电路故障注入**：首先在凝视相机图像数据压缩逻辑中选用帧头帧尾检测模块电路作为被测设计，采用本文提出的自动配置内存故障注入方法实现单粒子翻转定向故障注入，图 4-14 显示了凝视数据压缩 FPGA 设计的部分布局图，其中被测设计电路被布局在白色框图中，即底部的第一行至第四行的第 19 列到第 23 列，占据四个 CLB 列的共 400 帧，帧地址共四段，分别为：0x00101D00 到 0x00101D63、0x00101D80 到 0x00101DE3、0x00101E00 到 0x00101E63、0x00101E80 到 0x00101EE3。利用 Vivado 开发工具生成包含基本位的 EBD 文件，解析 EBD 文件中包含被测设计的“1”所在的位置，即被测设计中的基本位，将其转换为实际的物理帧地址列表，从物理帧地址列表中提取只包含 pBlock 的基本位生成 SEM IP 的故障注入地址。数据压缩处理单元界面显示故障注入结果如图 4-15 所示，压缩码流解码信息出错，压缩图像出现花块。

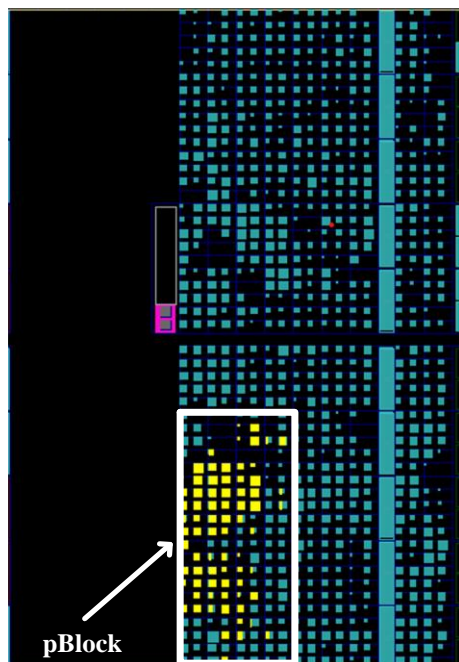


图 4-14 帧头帧尾检测模块作为 DUT 电路的布局图

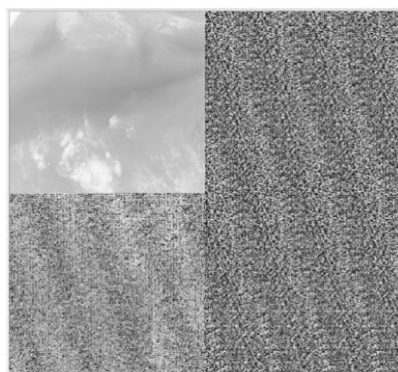


图 4-15 帧头帧尾检测模块定向故障注入结果

②**帧间压缩模块电路故障注入**：然后在凝视相机图像数据压缩逻辑中选用帧间压缩模块电路作为被测设计，采用本文提出的自动配置内存故障注入方法实现单粒子翻转定向故障注入，图 4-16 显示了凝视数据压缩 FPGA 设计部分的布局图，其中被测设计电路被布局在白色框图中，即底部的第二行至第五行的第 24 列，占据一个 CLB 列的共 36 帧，帧地址为 0x00106E40 到 0x00106E63。。采用相同方法将其转换为实际的物理帧地址列表，从物理帧地址列表中提取只包含 pBlock 的基本位生成 SEM IP 的故障注入地址。数据压缩处理单元界面显示故障注入结果如图 4-17 所示，压缩码流解码信息出错，压缩图像出现花块。

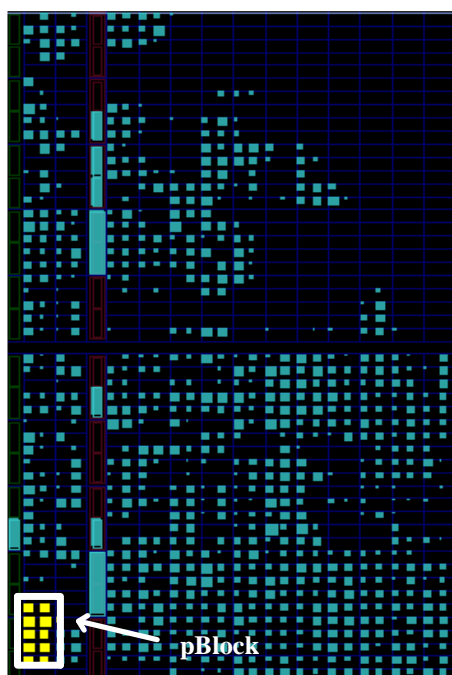


图 4-16 帧头帧尾检测模块作为 DUT 电路的布局图

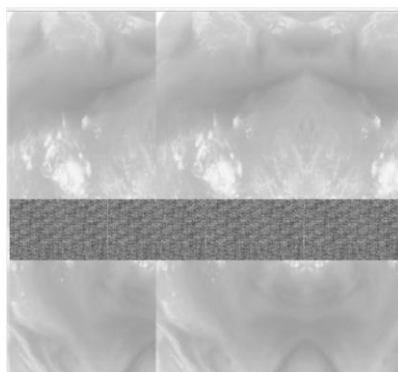


图 4-17 帧间压缩模块定向故障注入结果

③**组帧模块电路故障注入**：最后在凝视相机图像数据压缩设计中选用了组帧模块电路作为被测设计，采用本文提出的自动配置内存故障注入方法实现单粒子翻转定向故障注入，图 4-18 显示了凝视数据压缩 FPGA 设计部分的布局图，其中被测设计电路被布局在白色框图中，即底部的第一行至第四行的第 26 列到第 29 列，占据四个 CLB 列的共 180 帧，帧地址共四段，分别为：0x00110D00 到 0x00110D23、0x00110D80 到 0x00110DA3、0x00110E00 到 0x00110E23、0x00110E80 到 0x00110EA3。采用相同方法将其转换为实际的物理帧地址列表，从物理帧地址列表中提取只包含 pBlock 的基本位生成 SEM IP 的故障注入地址。数据压缩处理单元界面显示故障注入结果如图 4-19 所示，压缩码流解码信息出错，压缩图像出现花块。

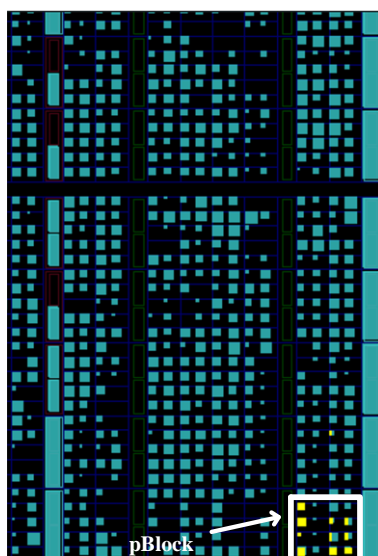


图 4-18 组帧模块作为 DUT 电路的布局图

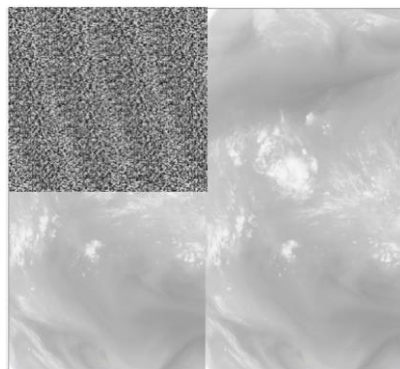


图 4-19 组帧模块定向故障注入结果

### (2) 传统的手动篡改配置位流文件注入故障方法

单粒子翻转故障随机注入主要通过对 FPGA 进行重配置来实现，手动修改配置比特流文件中的配置位，然后通过接口下载到目标 FPGA 存储器中，随机故障注入流程如图 4-20 所示。利用赛灵思公司发布的开发工具 Vivado 生成凝视压缩图像处理 FPGA 对应功能的配置比特流文件，通过 JTAG 接口下载到 FPGA 中，观察地面测试平台的数据压缩处理，功能运行正常。模拟单粒子翻转手动将配置位流文件中的比特位随机打翻，如图 4-21 所示，红色部分为手动篡改内容。将修改后的配置比特流文件通过 JTAG 接口下载到 FPGA 中，观察数据压缩单元压缩图像是否正常。

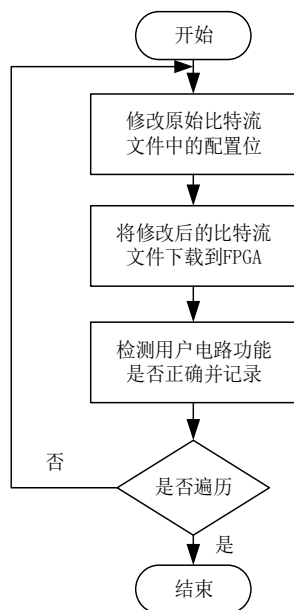


图 4-20 单粒子翻转随机故障注入流程



---

**Figure 1.** The relationship between the number of children and the probability of having a child who is at least 10 years old. The figure shows two sets of curves: one set for the probability of having a child who is at least 10 years old, and another set for the probability of having a child who is at least 10 years old and is also a girl. The curves are plotted against the number of children (0 to 10) and the probability of having a child who is at least 10 years old (0 to 1). The curves show that the probability of having a child who is at least 10 years old increases as the number of children increases, and the probability of having a child who is at least 10 years old and is also a girl decreases as the number of children increases.

效果也不如后者质量高。为了解决这个问题，在不使用自动配置内存故障注入方法的情况下，需要执行更多次数的故障注入，才能保证实际对帧头帧尾检测模块产生影响的故障数目与使用自动配置内存故障注入方法时相同。巨大的时间消耗使得许多实验无法正常实施，如果为了缩短实验时间而强行缩短故障注入的次数，将无法保证获得满足质量要求的实验结果。综上所述，自动配置内存故障注入在保证高质量实验结果的前提下，能够大大缩减故障注入的次数和实验耗费的时间，并且对于目标实验模块，其在整个用户设计中所占的比例越小，使用自动配置内存故障注入获得的收益就越高，因为更小的模块意味着整个故障注入过程更强的可控性和更高的效率。

更加关键的问题是，进行故障注入时如果不使用自动配置内存故障注入方法，大量注入的故障不会对目标模块帧头帧尾检测模块产生实际的影响，反而有可能将故障注入设计中的其他模块，导致原本运行正常的模块出现功能紊乱。这些不是注入帧头帧尾检测模块中的故障导致的错误，是由用户设计中还未被检测的其他模块产生的，它们也会影响系统的功能。换句话说，由于其他模块造成的干扰，我们无法针对帧头帧尾检测模块的可靠性进行严格的描述。但是，通过自动配置内存故障注入方法，我们不仅能够在保证目标模块区域中实际故障注入数目满足要求的情况下减少故障注入时间，还可以避免将故障注入用户设计的非目标模块区域中造成全局实验结果的扭曲。

因此，使用自动配置内存故障注入意味着：1）能够获得质量更高的测试结果；2）需要更少的时间来进行实验。在许多需要执行大量故障注入活动的情况下，自动配置内存故障注入能发挥更强的作用。

#### 4.4 定向故障注入技术局限性分析

本文提出的定向故障注入方法使用基本位数据（EBD）文件和 pBlock 之间的关系来生成故障注入的特定地址，这种关系基于通过逆向工程获得的几个参数，这些参数近似于线性行为，这种近似足以描述 FPGA 中实现的隔离模块的可靠性。然而，在更复杂、更现实的场景中，感兴趣的模块是更大设计的一部分，并被部署在其他

模块附近，检测到近似会导致不太准确的结果和不良的副作用，这是因为在 FPGA 的其他资源中注入了不属于测试设计的部分，以及在冗余输入/输出布线中注入。利用设计的 EBD 文件和 pBlock 的 slice 坐标来确定 SEM IP 的注入地址，然而，这种基于切片（slice-based）的坐标系不足以确定要转换的 EBD 行的精确范围，该坐标系要求通过逆向工程程序估计每个 FPGA 时钟域的斜率和截距值，从而实现近似化，这种近似化会导致 EBD 行范围不精确的误差不断累积，这意味着与 FPGA 配置内存中目标帧相邻的其他帧也被转换为注入地址。例如，被测设计（DUT）与用户电路中其他模块靠得很近，这将意味着通过 SEM IP 进行故障注入时，同时也会将故障注入其他模块中，如果设计接近 SEM IP 本身，甚至会产生不良的故障注入副作用。

对于 DUT 独立的用户设计，会多次将故障注入相同的输入/输出布线资源中，产生相同的效果。这种现象是 FPGA 器件固有的，在实际的平台上部署 DUT 时会发生。然而，当重点是使用 FPGA 来对设计的防护能力进行测试时，冗余注入是不希望发生的事件，因为它将会扭曲最终结果的可靠性。

为了解决 FPGA 固有特性的限制并提高工具的精度，需要进一步深入研究 FPGA 的体系结构，将基于网格的平铺坐标系替换基于切片的坐标系，并对定向故障注入方法的内部原理进行修改，以支持不同的坐标系。

### 4.5 本章小结

本章主要围绕在 Kintex-7 系列 FPGA 器件上实现单粒子翻转的故障定向注入方法展开研究。针对其中几个难点提出了解决方法。

首先，因为 Kintex-7 系列 FPGA 器件配置帧数据数量数以百万计，为了大幅缩减需要处理的配置帧数量，通过对基本位技术的深入研究，剔除配置帧数据中占绝大部分的与用户电路设计功能毫无关联的配置位，分离出只包含配置编程到 FPGA 中的特定应用程序使用的资源的配置位的子集，即基本位的 EBD 文件。

然后，由于得到的仅包含基本位的 EBD 文件只是简单的“0”和“1”构成的 ASCII 字符集，因此为了进一步将其翻译成故障注入器能够识别的配置帧地址，提出一种自动配置内存故障注入的方法。在 Kintex-7 系列 FPGA 的 floorplan 视图



中人为的界定了一个矩形区域，将该区域的顶点坐标与 EBD 文件通过方程式计算后，得到了故障注入器 SEM IP 的故障注入指令，通过该指令完成对 FPGA 用户电路特定模块的单粒子翻转故障注入。

最后，基于星载图像处理平台对本文提出的定向故障注入方法与随机故障注入方法进行比对实验，结果表明使用定向故障注入方法不仅能够保证对目标模块区域实现单粒子翻转故障注入，而且有效故障注入的准确率可达随机故障注入方法的 9 倍。

## 5 总结与展望

### 5.1 本文工作总结

本文研究了一种针对 Xilinx Kintex-7 系列 SRAM 型 FPGA 单粒子翻转定向注入方法，将故障注入 FPGA 中的配置位来模拟真实空间环境下发生的 SEU 现象。以 Kintex-7 系列 FPGA 中的配置位为目标进行单粒子翻转故障注入主要存在“配置帧结构未知”“配置帧数量庞大”以及“难以实现指定部位的故障注入”三个难点，针对以上难点本文完成的主要工作内容如下：

对于 SRAM 型 FPGA，单粒子翻转效应会影响其中任何一个存储单元，其中配置 RAM 决定了用户电路的逻辑功能，针对 Kintex-7 系列 FPGA 器件帧地址具体分布情况未知的问题，通过内部配置访问接口 ICAP 回读 FPGA 中帧地址寄存器，利用帧解码器 FRAME\_ECC 在回读配置帧数据时将具体的物理帧地址输出。获取所有的帧地址数据后，按照不同的帧地址类型进一步解析整理得到 Kintex-7 系列 FPGA 器件具体的帧地址分布，提取出与配置 RAM 资源对应的配置帧。

针对 Kintex-7 系列 FPGA 器件配置帧数据数量过于庞大的问题，为了大幅缩减需要处理的配置帧数量，通过对基本位技术的深入研究，剔除配置帧数据中占绝大部分的与用户电路设计功能毫无关联的配置位，分离出只包含配置编程到 FPGA 中的特定应用程序使用的资源的配置位的子集，即基本位的 EBD 文件。

针对难以实现指定部位单粒子翻转故障注入的问题，由于得到的仅包含基本位的 EBD 文件只是简单的“0”和“1”构成的 ASCII 字符集，因此为了进一步将其翻译成故障注入器能够识别的配置帧地址，提出一种自动配置内存故障注入的方法。在 Kintex-7 系列 FPGA 的 floorplan 视图中人为的界定了一个矩形区域，将该区域的顶点坐标与 EBD 文件通过方程式计算后，得到了故障注入器 SEM IP 的故障注入指令，通过该指令完成对 FPGA 用户电路特定模块的单粒子翻转故障注入。

最后，在基于 Kintex-7 FPGA 的某星载图像压缩平台中的帧头帧尾检测模块、帧间压缩模块、组帧模块三个用户电路进行验证，实验结果表明，定向故障注入

方法能对指定功能模块完成故障注入，并且有效故障注入数可达随机故障注入方法的 9 倍。

### 5.2 未来工作展望

在本文研究成果的基础上，仍存在尚未解决的问题，未来需要开展的研究工作如下：

（1）对于 DUT 独立的用户设计，会多次将故障注入相同的输入/输出布线资源中，产生相同的效果。这种现象是 FPGA 器件固有的，在实际的平台上部署 DUT 时会发生。然而，当重点是使用 FPGA 来对设计的防护能力进行测试时，冗余注入是不希望发生的事件，因为它将会扭曲最终结果的可靠性。为了解决 FPGA 固有特性的限制并提高方法的精度，需要进一步深入研究 FPGA 的体系结构，将基于网格的平铺坐标系替换基于切片的坐标系，并对自动配置内存故障注入的内部原理进行修改，以支持不同的坐标系。

（2）在实现对目标功能模块的单粒子翻转故障注入后，还应考虑后续对 SEU 敏感程度高的模块开展辐射容错研究，进行 SEU 减缓系统设计，降低重要模块的故障率。

### 参考文献

- [1] Benevenuti F, Chielle E, Tonfat J, Tambara L, Kastensmidt F L, Zaffari C A, et al. Experimental Applications on SRAM-Based FPGA for the NanosatC-BR2 Scientific Mission[C]. IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), 2019: 140-146.
- [2] Zhang R, Xiao L, Li J. An adjustable and fast error repair scrubbing method based on Xilinx essential bits technology for SRAM-Based FPGA[J]. IEEE Transactions on Reliability, 2019, 69(2): 430-439.
- [3] Siegle F, Vladimirova T, Ilstad J. Availability analysis for satellite data processing systems based on SRAM FPGAs[J]. IEEE Transactions on Aerospace and Electronic Systems, 2016, 52(3): 977-989.
- [4] Sogoyan Armen V, Smolin Anatoly A, Chumakov Alexander I. Single event effects qualificatoin of integrated circuits[J]. Bezopasnost informacionnyh tehnology, 2020, 27(1).
- [5] Moll Michael. Displacement Damage in Silicon Detectors for High Energy Physics[J]. IEEE Transactions on Nuclear Science, 2018, 65(8).
- [6] Liu Rui, Ferlet Cavois Veronique, Evans Adrian, Chen Li, Li Yuanqing, Glorieux Maximilien, et al. Single Event Transient and TID Study in 28 nm UTBB FDSOI Technology[J]. IEEE Transactions on Nuclear Science, 2017, 64(1).
- [7] Fan Y Y, Cai X D, He C H. On-orbit single event effect of the digital signal processor of the alpha magnetic spectrometer and discrepancy analysis for the rate prediction[J]. IEEE Transactions on Nuclear Science, 2018, 65(5): 1140-1146.
- [8] Li J, Choutko V, Xiao L. Single event upset analysis: On-orbit performance of the alpha magnetic spectrometer digital signal processor memory aboard the international space station[J]. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 2018, 885: 98-104.
- [9] Kaul H, Anders M, Hsu S. Near-threshold voltage (NTV) design: Opportunities and challenges[C]. Proceedings of the 49th Annual Design Automation Conference. 2012: 1153-1158.

- [10] Ravi Dontaraju, S. Bhujanga Rao. An innovative Technique to Protect a Register File from Multiple-bit-upsets Implemented on FPGA.[J]. International Journal of Recent Technology and Engineering (IJRTE),2019,8(1).
- [11] Ebrahimi, Mojtaba, Rao, Parthasarathy Murali B, Seyyedi, Razi, et al. Low-Cost Multiple Bit Upset Correction in SRAM-Based FPGA Configuration Frames[J]. IEEE transactions on very large scale integration (VLSI) systems,2016,24(3).
- [12] Hyungmin Cho, Kon-Woo Kwon. Modeling Application-Level Soft Error Effects for Single-Event Multi-Bit Upsets[J]. IEEE Access, 2019, 7.
- [13] 徐鑫龙. 抗辐射 SRAM 测试系统的设计与实现[J]. 中国集成电路, 2016, 25(03): 52-55.
- [14] 胡孔阳, 胡海生, 刘小明. 三模冗余在高性能抗辐射 DSP 中的应用[J]. 微电子学与计算机, 2019, 36(03): 58-60.
- [15] 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(06): 1367-1370.
- [16] Quinn, Heather, Wirthlin, Michael. Validation Techniques for Fault Emulation of SRAM-based FPGAs[J]. IEEE Transactions on Nuclear Science, 2015, 62(4).
- [17] Baig Hasan, Lee Jeong A, Siddiqui Zahid Ali. A Low-Overhead Multiple-SEU Mitigation Approach for SRAM-based FPGAs with Increased Reliability[J]. IEEE Transactions on Nuclear Science, 2014, 61(3).
- [18] Garcı R, Biskup B, Brugger M. SEU measurements and simulations in a mixed field environment[J]. IEEE Transactions on Nuclear Science, 2013, 60(4): 2469-2476.
- [19] Höeffgen S K, Metzger S, Steffens M. Investigating the effects of cosmic rays on space electronics[J]. Frontiers in Physics, 2020, 8: 318.
- [20] Hachaj T, Bibrzycki Ł, Piekarczyk M. Recognition of cosmic ray images obtained from CMOS sensors used in mobile phones by approximation of uncertain class assignment with deep convolutional neural network[J]. Sensors, 2021, 21(6): 1963.
- [21] Alía Rubén García, Bonaldo Stefano, Brugger Markus, Danzeca Salvatore, Ferrari Alfredo, Frost Christopher, et al. Single Event Effect cross section calibration and application to quasi-monoenergetic and spallation facilities[J]. EPJ Nuclear Sciences & Technologies, 2018, 4.

## 硕士学位论文

---

- [22] Rubén García Alía, Markus Brugger, Salvatore Danzeca, Francesco Cerutti, Joao Pedro de Carvalho Saraiva, Reiner Denz, et al. Single event effects in high-energy accelerators[J]. Semiconductor Science and Technology, 2017, 32(3).
- [23] Wang X, Xu Z. A Novel Fault Injection Algorithm for Safety Analysis[C]. 2012 Spring Congress on Engineering and Technology. IEEE, 2012: 1-4.
- [24] Na J, Lee D. Simulated fault injection using simulator modification technique[J]. ETRI Journal, 2011, 33(1): 50-59.
- [25] Nimara S, Amaricai A, Popa M. Sub-threshold cmos circuits reliability assessment using simulated fault injection based on simulator commands[C]. 2015 IEEE 10th Jubilee International Symposium on Applied Computational Intelligence and Informatics. IEEE, 2015: 101-104.
- [26] 吴杰. 基于 VHDL 的故障注入工具的研究与实现[D]. 哈尔滨工业大学, 2013.
- [27] Johnson E, Caffrey M, Graham P. Accelerator validation of an FPGA SEU simulator[J]. IEEE Transactions on Nuclear Science, 2003, 50(6): 2147-2157.
- [28] Swift G M, Rezgui S, George J. Dynamic testing of Xilinx Virtex-II field programmable gate array (FPGA) input/output blocks (IOBs)[J]. IEEE Transactions on Nuclear Science, 2004, 51(6): 3469-3474.
- [29] George J, Koga R, Swift G. Single event upsets in Xilinx Virtex-4 FPGA devices[C]. 2006 IEEE Radiation Effects Data Workshop. IEEE, 2006: 109-114.
- [30] Sterpone L, Boragno L, Codinachs D M. Analysis of radiation-induced SEUs on dynamic reconfigurable systems[C]. 2016 11th International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC). IEEE, 2016: 1-6.
- [31] Zhu M, Song N, Pan X. Mitigation and experiment on neutron induced single-event upsets in SRAM-based FPGAs[J]. IEEE Transactions on Nuclear Science, 2013, 60(4): 3063-3073.
- [32] 宋凝芳, 朱明达, 潘雄. SRAM 型 FPGA 单粒子效应试验研究[J]. 宇航学报, 2012 (6): 836-842.
- [33] 王志敏. Virtex-6 FPGA 单粒子翻转快速故障注入平台设计[D]. 哈尔滨工业大学, 2020.

## 硕士学位论文

---

- [34] 卢凌云, 徐宇, 李悦, 李天文, 蔡刚, 杨海钢. 基于定向故障注入的 SRAM 型 FPGA 单粒子翻转效应评估方法[J]. 微电子学, 2017, 47(01): 135-140.
- [35] Zibo Wang, Wei Chen, Zhibin Yao, Fengqi Zhang, Yinhong Luo, Xiaobin Tang, et al. Proton-induced single-event effects on 28 nm Kintex-7 FPGA[J]. Microelectronics Reliability, 2020, 107.
- [36] Al-Khafaji A Q, Al-Gailani M F, Abdullah H N. FPGA Design and Implementation of an AES Algorithm based on Iterative Looping Architecture[C]. 2019 IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin). Berlin: IEEE, 2019: 1-5.
- [37] Ebrahimi M, Sadeghi R, Navabi Z. LUT Input Reordering to Reduce Aging Impact on FPGA LUTs[J]. IEEE Transactions on Computers, 2020, 69(10): 1500-1506.
- [38] 舒德刚. SRAM 型 FPGA 单粒子翻转定向故障注入技术研究[D]. 哈尔滨工业大学, 2019.
- [39] 赵兴, 栗伟珉, 程向丽, 王曦煜, 李昂阳. 航天空间环境单粒子效应研究[J]. 电子制作, 2021(13): 87-89.
- [40] 李铁虎. 深亚微米和纳米级集成电路的辐照效应及抗辐照加固技术[D]. 西安电子科技大学, 2018.
- [41] 齐春华. CMOS 存储单元电路抗单粒子翻转加固设计研究[D]. 哈尔滨工业大学, 2018.
- [42] 吕浩. 单粒子效应系统级故障注入仿真方法研究[D]. 西安电子科技大学, 2021.
- [43] 余永涛, 陈煜海, 余俊杰, 龙伊雯, 罗军, 王小强等. SRAM 型 FPGA 单粒子效应测试方法及试验验证[J]. 航天器环境工程, 2021, 38(05): 534-540.
- [44] 刘鸿瑾, 李天文, 粮时楠. 一种新型单粒子翻转加固 SRAM 单元[J]. 半导体技术, 2018(12): 941-948.
- [45] 姚志斌, 范如玉, 郭红霞, 王忠明, 何宝平, 张凤祁等. 静态单粒子翻转截面的获取及分类[J]. 强激光与粒子束, 2011, 23(03): 811-816.
- [46] ASMBL——创新下一代平台 FPGA[J]. 今日电子, 2004(05): 28-29.
- [47] She X, Li N. Reducing Critical Configuration Bits via Partial TMR for SEU

## 硕士学位论文

---

- Mitigation in FPGAs[J]. IEEE Transactions on Nuclear Science, 2017, 64(10): 2626-2632.
- [48] Mandal S, Paul R, Sau S. A Novel Method for Soft Error Mitigation in FPGA Using Modified Matrix Code[J]. IEEE Embedded Systems Letters, 2016, 8(4): 5-68.
- [49] M Grosso, H Guzman-Miranda, M A Aguirre. Exploiting Fault Model Correlations to Accelerate SEU Sensitivity Assessment[J]. IEEE Transactions on Industrial Informatics, 2015. 9(1): 142-148.
- [50] 薛晓良. SRAM 型 FPGA 在辐照环境下的容错技术研究[D]. 中国科学院大学(中国科学院光电技术研究所), 2019.