

# 硕士学位论文

## 基于 ECC 电路的软错误修复 和测试诊断 NBTI 错误方法研究

### **METHOD RESEARCH ABOUT CORRECTION FOR SOFT ERROR AND DETECTION FOR NBTI INDUCED ERROR BASED ON ECC CIRCUIT**

闻昌

哈尔滨工业大学

2016 年 6 月

国内图书分类号: TN432

学校代码: 10213

国际图书分类号: 621.3.049.774

密级: 公开

## 工程硕士学位论文

# 基于 ECC 电路的软错误修复 和测试诊断 NBTI 错误方法研究

硕 士 研 究 生: 闻昌

导 师: 肖立伊 教授

申 请 学 位: 工程硕士

学 科: 集成电路工程

所 在 单 位: 航天学院

答 辩 日 期: 2016 年 6 月

授予学位单位: 哈尔滨工业大学

Classified Index: TN432

U.D.C: 621.3.049.774

Dissertation for the Master's Degree in Engineering

**METHOD RESEARCH ABOUT CORRECTION FOR  
SOFT ERROR AND DETECTION FOR NBTI  
INDUCED ERROR BASED ON ECC CIRCUIT**

<b>Candidate:</b>	Wen Chang
<b>Supervisor:</b>	Prof. Xiao Liyi
<b>Academic Degree Applied for:</b>	Master of Engineering
<b>Speciality:</b>	Integrated Circuit Engineering
<b>Affiliation:</b>	School of Astronautics
<b>Date of Defence:</b>	June, 2016
<b>Degree-Conferring-Institution:</b>	Harbin Institute of Technology

## 摘 要

随着器件工艺尺寸的不断减小，静态随机存取存储器（Static Random Access Memory, SRAM）中相邻的存储单元之间的距离变得越来越近，对空间和地面中的质子、中子、 $\alpha$  粒子以及宇宙射线等电离辐射更加敏感。由一次电离辐射事件造成的单粒子翻转（Single Event Upset, SEU）和多位错误翻转（Multiple Bit Upset, MBU）都明显增加。存储器抗辐射翻转已经成为 SRAM 加固中的研究热点。

同时，由于 CMOS 超大规模集成电路的发展，器件等比例缩小、栅介质层厚度的不断减小，负偏置温度不稳定性（Negative Bias Temperature Instability, NBTI）效应对 PMOS 器件和电路的可靠性的影响变得越来越显著，成为影响器件及电路寿命不可忽略的因素之一。

本文在深入研究存储器存储单元单节点翻转和多位翻转以及 NBTI 效应对器件老化的影响的基础上，从电路级和系统级加固方法的角度出发，利用一步大数逻辑译码的低复杂度、较小的延迟以及较高的纠错检错能力，设计了针对 32 位存储器抗多位翻转的加固方案，实现了对存储器的 32 位存储数据纠错检错能力为 4 的存储器加固。

同时，为了使本文建立的一步大数逻辑译码能够检测 NBTI 引起的错误，本文提出基于 ECC 电路的二次检测方法，设计能够对 NBTI 错误和软错误进行检测区分的 SRAM 模块。同时设计 AHB 总线接口，使得增加了 NBTI 检测机制的 SRAM 能够在系统中集成使用，并对设计进行验证。

**关键词：**一步大数逻辑译码；存储器；SEU；MBU；NBTI；AHB

## Abstract

With the process of reducing the size of the device, a static random access memory (Static Random Access Memory, SRAM) distance between adjacent memory cells become increasingly close, and ground space of protons, neutrons more sensitive to particles and cosmic rays and other ionizing radiation. Single-particle ionizing radiation incident caused by the first flip (Single Event Upset, SEU), and multi-bit error flip (Multiple Bit Upset, MBU) were significantly increased. Flip has become the anti-radiation memory SRAM reinforcement of research.

Meanwhile, the proportion of CMOS VLSI development, and other devices shrink, decreasing the thickness of the gate dielectric layer, negative bias temperature instability (Negative Bias Temperature Instability, NBTI) effect on the PMOS device and circuit reliability effects become more pronounced, becoming one of the factors affect the life of devices and circuits can not be ignored.

On the basis of in-depth study of the impact a single memory cell node and a number of flip flip and NBTI effect of aging on the device, from the perspective of circuit-level and system-level departure reinforcement methods, the use of low complexity step majority logic decoding, a small delay and high error correction error detection capability, designed for 32-bit memory number of anti-rollover reinforcement program to achieve 32-bit data stored on the memory error detection error correction capability of 4 memory reinforcement.

Meanwhile, in order to make paper establishes step majority logic decoder can detect errors caused by NBTI proposed second ECC circuit detection method based on NBTI designed to detect errors and soft errors distinction SRAM module. AHB bus interface design at the same time, so as to increase the SRAM NBTI detection mechanism can be used in the system integration, and design verification.

**Keywords:** Step majority-logic decoding, SRAM, SEU, MBU, NBTI, AHB

# 目 录

摘 要.....	I
Abstract.....	II
第 1 章 绪 论.....	1
1.1 课题背景及研究的目的和意义 .....	1
1.2 国内外研究现状 .....	3
1.2.1 单粒子翻转 .....	3
1.2.2 存储器多位错误翻转 .....	5
1.2.3 存储器抗翻转加固技术 .....	6
1.2.4 负偏置温度不稳定性 .....	8
1.3 主要研究内容 .....	9
第 2 章 编码理论基础 .....	11
2.1 基本数学概念 .....	11
2.1.1 群 .....	11
2.1.2 域 .....	11
2.2 向量空间 .....	12
2.3 线性分组码 .....	13
2.3.1 线性分组码原理 .....	13
2.3.2 编码电路原理 .....	15
2.3.3 校正子与差错检测 .....	16
2.3.4 译码电路原理 .....	18
2.4 一步大数逻辑译码 .....	18
2.5 本章小结 .....	20
第 3 章 大数逻辑可译码加固 SRAM.....	21
3.1 一步大数逻辑译码的构造 .....	21
3.1.1 一类一步大数逻辑可译码 .....	21
3.1.2 一步大数逻辑可译 1 型 (63,37) DTI 码 .....	25
3.2 编码器设计 .....	28
3.3 译码器设计 .....	29
3.3.1 校验函数 .....	29

3.3.2 译码输出 .....	30
3.4 一步大数逻辑译码加固 SRAM 及验证 .....	31
3.4.1 存储器模型 .....	31
3.4.2 存储器加固方案 .....	33
3.4.3 故障注入及存储器加固验证 .....	34
3.5 本章小结 .....	35
第 4 章 检测 NBTI 错误的 SRAM 设计与验证 .....	37
4.1 测试诊断软错误和 NBTI 错误设计 .....	37
4.1.1 具有 NBTI 检测功能的 SRAM 的设计 .....	38
4.1.2 仿真验证 .....	41
4.2 SRAM 的 AHB 总线接口设计与系统级验证 .....	44
4.2.1 AHB 总线接口设计 .....	44
4.2.2 系统级验证 .....	47
4.3 本章小结 .....	50
结 论 .....	51
参考文献 .....	52
哈尔滨工业大学学位论文原创性声明及使用授权说明 .....	59
致 谢 .....	60

# 第1章 绪 论

## 1.1 课题背景及研究的目的和意义

目前,随着我国航天与核能技术的发展,集成电路以其体积小,功能强,重量轻等优点在军民生活中越来越多的被应用<sup>[1]</sup>。集成电路的工艺发展到目前为止,不论是芯片面积还是芯片体积都越来越小,这使得芯片的性能和功能都得到很大的提高<sup>[2][3]</sup>。但同时带来的可靠性问题也越来越突出。高能粒子入射到半导体器件内部,这将引发器件内的能量积累的改变,使得器件发生改变甚至失效,这种现象被称为单粒子效应<sup>[4]</sup>。经过长时间的观测,1975年在空间环境中单粒子效第一次被人们发现<sup>[5]</sup>。

半导体集成电路器件在空间辐射环境中会受到高能粒子的撞击,这些撞击将使得器件发生瞬态错误<sup>[6]</sup>。其中,这些错误大约占集成电路故障总量的90%<sup>[7]</sup>。卫星和航天器中产生的各种故障,超过40%与单粒子效应有关<sup>[8]</sup>。NASA在1996年发布了一份从1974年到1994年这20年中,所收集到的100多项器件失效的故障原因,其中,这些研究表明,约33%的故障是和电离辐射有关的,在电离辐射所引发的故障中,42%与SEU有关<sup>[9][10]</sup>。

卫星以及各种航天器存在的太空环境中,有大量的宇宙射线和带电粒子<sup>[11]</sup>,主要来源有如下几种:

### 1)、地球俘获带

地球俘获带又被称为范艾伦辐射带。其位于近地约800km的范围内。又分为内外辐射带,内带主要由30-100MeV的质子组成,外带主要由0.4-1MeV的电子组成<sup>[12]</sup>。

### 2)、太阳宇宙射线

太阳粒子事件(SEP)是一种随着时间而剧烈变化的行为。太阳系中的粒子扰动主要由太阳耀斑和日冕层的物质抛射(CME)构成<sup>[13]</sup>。太阳粒子主要为质子和 $\alpha$ 粒子。

### 3)、银河系宇宙射线

银河宇宙射线(GCR)主要是以质子和 $\alpha$ 粒子为主的、几乎涵盖所有元素的从太阳系以外而来的高能带电粒子<sup>[14]</sup>。这些宇宙射线大大增加了器件发生故障的概率。



集成电路的集成度越来越高，在电路中存储器是整个电路中的重要的组成部分<sup>[15]</sup>，而且集成的存储器越来越庞大。对微处理器的统计研究表明，其中存储器占到整个芯片面积的 30% 左右<sup>[16]</sup>；片上系统（System on Chip, SoC）中存储器的面积一般大于 60%，其中存在大量的 ROM 和 SRAM 以及 DRAM 和 FLASH 块<sup>[17]</sup>。文献[18]中研究表明，在集成电路中软错误的发生概率远远高于其他的错误类型，极大的影响器件的可靠性和稳定性。一次辐射事件造成多个存储器单元发生错误翻转，被称为存储器多位翻转<sup>[19]</sup>（Multiple Bit Upset, MBU），这种现象是由 Balke 在 1986 年中发布的关于 SRAM 在近极轨道的故障信息中被首次发现<sup>[20]</sup>。在其统计的总共 58 次辐射事件中，导致多个存储器单元发生故障的一共有 11 次，其中，影响单元数量的最大值达到了 4 位<sup>[20]</sup>。在超深亚微米时代，单位面积上集成的 SRAM 大大增加，这也使存储器之间的距离变得更近，从而将使得存储器件更容易发生 MBU 错误<sup>[21][22]</sup>。Heather Quinn 2005 年对 FPGA 中 MBU 效应的评估研究中<sup>[23]</sup>对赛灵思公司三种不同系列的 FPGA 芯片做了质子辐射实验，得到数据如表 1-1 所示。从表 1-1 中可以看到，器件发生 MBU 的概率会因为器件的工艺尺寸的减小而变大。而且器件尺寸还在不断的减小，根据 International Technology Roadmap for Semiconductor(ITRS)2012 年报告预测<sup>[24]</sup>如表 1-2 所示。

表 1-1 FPGA 芯片质子辐射实验结果

系列	工艺尺寸 (nm)	发生 MBU 的概率 (%)
Virtex	220	0.04
Virtex-II Pro	130	1.71
Virtex4	90	3.56

同时，随着半导体工艺的发展，器件的尺寸变得越来越小，这引发了另一个严重的问题，即器件的短沟道效应。为了减小短沟道效应对器件的影响，减小器件亚阈值斜率，增大驱动电流，必须使 MOS 器件的栅氧化层厚度和沟道长度同比例减小<sup>[12]</sup>。随着器件尺寸的不断减小，温度对器件可靠性的影响主要有热载流子效应和负偏置温度不稳定<sup>[25][26]</sup>。其中相比于热载流子效应对器件的影响，近年来研究表明对于 PMOSFET 产生的退化中，NBTI 老化效应越来越突出。它比由热载流子效应引起的 MOS 场效应晶体管寿命退化更为严重，也越来越多的引起人们的关注。

如图 1-1 所示为典型的 NBTI 效应示意图<sup>[24]</sup>。集成电路工作的特殊环境，如高温、高压以及辐射等对器件的性能和可靠性产生很大的影响，尤其进入超深亚微

米时代，NBTI 效应的影响也越来越显著<sup>[27]</sup>。所以研究 NBTI 以及软错误对器件，特别是存储器件的影响刻不容缓。

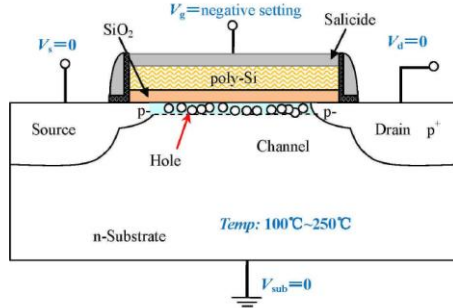


图 1-1 NBTI 效应剖面示意图<sup>[24]</sup>

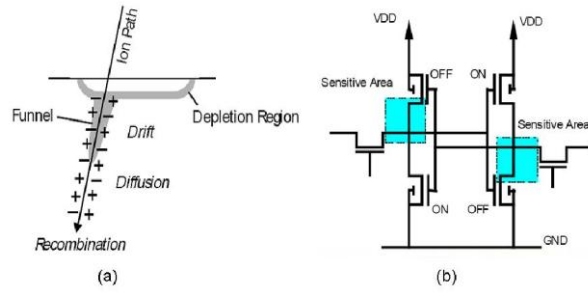
表 1-2 ITRS 预测<sup>[24]</sup>

Year of Production Requirement	2012	2013	2014	2015	2016	2017	2018	2019	2020
DRAM 1/2 Pich(nm)	31	28	25	22	20	18	16	14	13
L <sub>gate</sub> :for High-performance Logic(nm)	22	20	18	17	15.3	14.0	12.8	11.7	10.6
V <sub>dd</sub> :Power Supply Voltage(V)	0.87	0.85	0.82	0.80	0.77	0.75	0.73	0.71	0.68
EOT:Equivalent Oxide Thickness(nm)	0.92	0.88	0.84	0.8	0.76	0.72	0.68	0.65	0.62

## 1.2 国内外研究现状

### 1.2.1 单粒子翻转

最容易发生单粒子翻转效应的器件是利用双稳态进行存储的器件，比如 RAM，其次是 CPU，然后是其他接口电路<sup>[28]</sup>。随着工艺尺寸的减小，芯片集成度的增加，器件发生 SEU 故障的概率也在逐渐增大。


 图 1-2 漏斗效应以及 SRAM 中单粒子翻转示意图<sup>[24]</sup>

在一些高温高压以及辐射环境中，大规模集成电路受到干扰，尤其是存储器件，当受到高能粒子轰击后会使存储器件的逻辑数据发生翻转：原来的“1”变为“0”，原来的“0”变为“1”。这使得存储器件的数据发生变化将产生严重的后果，SEU 造成的逻辑错误是暂时性的不是永久性的，所以 SEU 引起的错误也被称为软错误<sup>[29][30]</sup>。

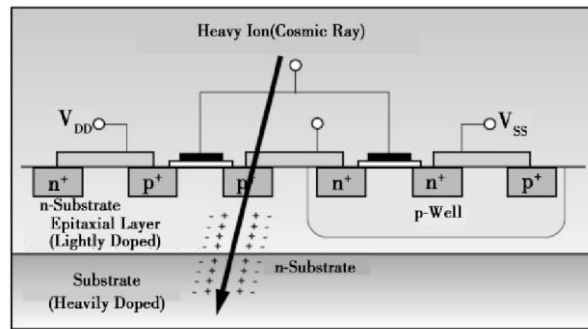
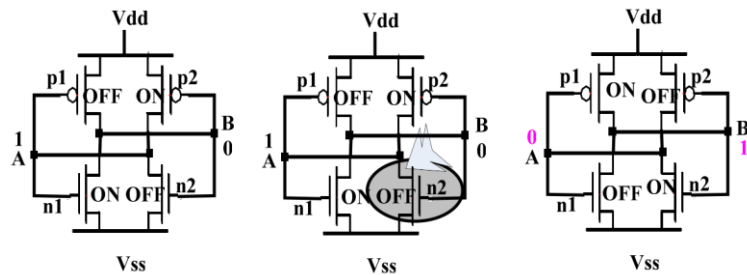

 图 1-3 重离子引发 SEU 示意图<sup>[25]</sup>

图 1-3 所示为单粒子效应示意图，当器件受到重粒子轰击后，在其行走路线上产生电子-空穴对，产生的这些电子和空穴将对器件的原始形态产生变化。当外界应力消失后，经过一段时间后，这些电子和空穴将慢慢消失中和，器件又恢复到原来的状态。


 图 1-4 SEU 对 SRAM 存储单元的影响<sup>[26]</sup>

如图 1-4 所示，每个 MOS 管组成的存储单元处于稳定状态，电路节点 A 存储数据“1”，电路节点 B 存储数据“0”。当受到粒子撞击后这种稳定状态将被打破，存储电路中会产生一个瞬间电流脉冲使得晶体管导通状态发生改变，使原来存储的数据发生翻转，结果使得节点 A 的数据变为“0”，节点 B 的数据变为“1”。

### 1.2.2 存储器多位错误翻转

存储器多位翻转（Multiple Bit Upset, MBU）是一种由一次辐射事件引起的存储器多位数据单元发生翻转的瞬态错误<sup>[31]</sup>。研究表明，质子、中子和重离子等辐射源是导致 MBU 产生的主要辐射来源<sup>[32][33]</sup>。而重离子的离化作用将使 MBU 发生的概率大大增加<sup>[20][34]</sup>。同时，器件的工艺尺寸不断减小，单位面积上集成的存储器件增多，这使得存储单元的临界电荷不断降低<sup>[35]</sup>，从而导致重离子和质子等更加容易导致 MBU 发生。表 1-3 表示了多位翻转占单粒子翻转的百分比统计，图 1-5 表示了 MBU 错误图样。

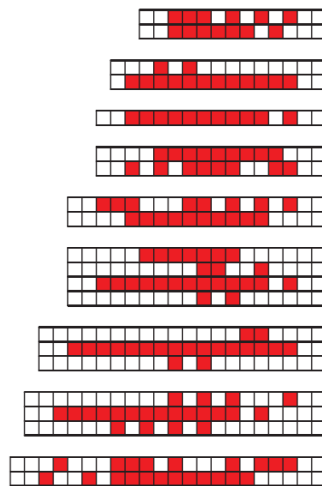


图 1-5 MBU 错误图样<sup>[36]</sup>

表 1-3 多位翻转占单粒子翻转的百分比<sup>[20][40]</sup>

工艺 尺寸 (nm)	总数	多位翻转位数				最大 位数
		2	3	4-8	>8	
180	0.5%	0.5%	0.0%	0.0%	<0.1%	2
130	1.2%	0.8%	0.2%	0.2%	<0.1%	7
90	1.9%	1.5%	0.2%	0.2%	<0.1%	7
65	2.4%	2.1%	0.1%	0.0%	<0.1%	10

表 1-3 （续表）

工艺 尺寸 (nm)	总数	多位翻转位数				最大 位数
		2	3	4-8	>8	
45	2.3%	1.9%	0.2%	0.1%	<0.1%	16
32	3.1%	2.6%	0.2%	0.3%	<0.1%	16
22	3.9%	3.0%	0.3%	0.1%	<0.1%	18

### 1.2.3 存储器抗翻转加固技术

对存储器中的加固主要可以分为以下几种：器件级、电路级、版图级、封装级和系统级等。各种加固方法有其各自的特点。

#### 1、器件级加固

器件级加固是使用不同的衬底材料，来达到抗辐射的能力<sup>[37]-[39]</sup>。典型的有：绝缘体上硅（Silicon On Insulator, SOI）器件，这种器件对抗表面效应、电离效应、位移效应以及单粒子效应都比较强，工艺相对比较成熟<sup>[40][41]</sup>；蓝宝石上硅（Silicon On Sapphire, SOS）器件，这种器件抗辐射能力强，但是芯片成品率低成本高而且易碎；SON（Silicon On Nothing）器件，这种器件控制短沟道效应和迁移率等要比 SOI 器件好，其在抗辐射加固方面的性能有着广阔的应用前景<sup>[42][43]</sup>。

#### 2、版图级加固

版图级加固是改变芯片的制造工艺和版图设计，从本源上解决电荷积累的原因。一般的版图级加固方法有：阱电极优化，阱电极是 CMOS 器件中除了漏、源和栅极之外的衬底电极，通过优化设计阱电极的位置、形状和大小等可以减小辐射对器件的影响<sup>[44]</sup>；隔离阱；反向衬底偏置（RBB）MOS 以及环形栅极（annular gate），如图 1-6 所示；版图单元交错，如图 1-7 所示等方法。

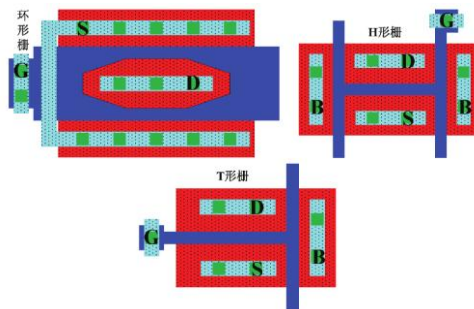


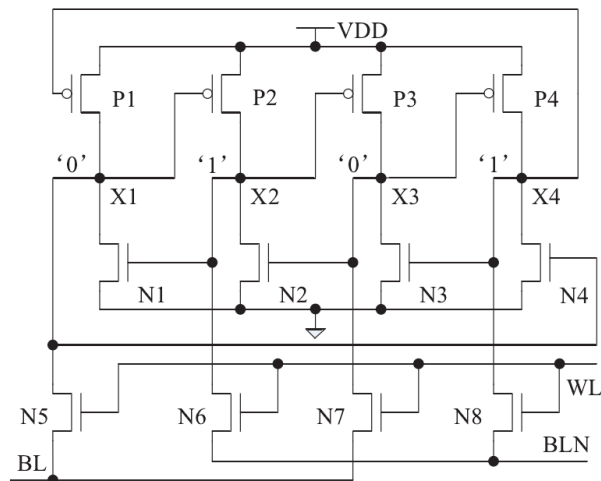
图 1-6 版图加固技术<sup>[32]</sup>

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32

 图 1-7 存储器版图位交错技术<sup>[36]</sup>

### 3、电路级加固

电路级加固是向电路结构中加入了时间或者空间的冗余，以及能从冗余信息中恢复原始信息的电路结构<sup>[45]</sup>。这种类型的加固方法对单粒子翻转有着很好的处理结果，典型的形式有双互锁单元（Dual Interlocked Cell, DICE），这种结构是一种 12 管的 SRAM 存储单元，如图 1-8 所示；三模冗余（Triple Modular Redundancy, TMR）和时间采样（Temporal sampling）这种技术利用电路的空间与时间冗余一般应用在 D 触发器上，具有一定的抗 SEU 与 SET 的能力；其余的还有增加固定电容；反馈去耦电阻；减小高阻抗节点<sup>[46]</sup>等方法。


 图 1-8 DICE 结构<sup>[45]</sup>

### 4、封装级加固

封装级加固是基于寻找新的抗辐射封装材料来改变封装工艺以达到抗辐射的

目的。典型的封装加固形式有：抗辐射外壳，使用 HighZ 材料做顶部和底部封装，共烧 Al 封装技术<sup>[45][46]</sup>；商用器件封装重构；抽真空封装，这种封装是为了避免封装器件内气体的电离辐射效应对器件的表面的影响，真空度一般为  $1 \times 10^{-1} \sim 1 \times 10^{-2} \text{ mmHg}$ ，不采用惰性气体封装<sup>[47]-[49]</sup>。

### 5、系统级加固

系统级加固中，存储器一般采用故障检测和修正技术（Error Detection and Correction, EDAC），如汉明码（Hamming）、双错误探测码（Single Error Correction and Double Error Detection, SEC-DED）它们可以修正一位错误，同时也可以探测两位错误，但是不能修正两位及两位以上的错误<sup>[50]</sup>。随着工艺尺寸的进步，器件沟道长度变短，多位错误已经在太空和核辐射环境中被发现，因此需要纠错能力更强的错误修正码（Error Correction Code, ECC）来对存储器进行加固。如 Reed-Solomon（RS）码<sup>[51]</sup>、Bose-Chaudhuri-Hocquenghem（BCH）<sup>[52]</sup>以及一步大数逻辑可译码（One-Step Majority Logic Decodable, OS-MLD）等。通过 EDAC 电路不需要额外的改变 MOS 管的结构以及封装结构等，能够大大的提高存储器及电路结构的抗干扰能力，从而提高了系统的可靠性；但是同时由于其编码和译码过程的复杂度也将带来系统的功耗、面积和延迟等方面的问题<sup>[9][53]</sup>。

#### 1.2.4 负偏置温度不稳定性

在 MOS 器件发展的早期，负偏置温度不稳定性（NBTI）效应就被发现并注意到，1967 年人们第一次观测到 NBTI<sup>[30]</sup>。NBTI 通常是指 PMOS 管在高温、强电场以及大的负栅压环境下，器件性能表现出的退化，典型的温度是在  $80-250^{\circ}\text{C}$  的范围内，图 1-9 为 NBTI 效应示意图。这种情况在器件老化和工作过程中都可能遇到，NBTI 效应引起的器件老化主要表现为对器件的关态电流（ $I_{off}$ ）增大，阈值电压（ $V_{th}$ ）负向漂移，亚阈值摆幅（ $S$ ）增大以及跨导（ $G_m$ ）和漏电流（ $I_{ds}$ ）减小等<sup>[27]</sup>。在模拟电路中 NBTI 效应会使晶体管在工作中受到不对称的偏置应力，这些偏置应力将会导致晶体管的参数失配。在数字电路的应用中，NBTI 老化效应会使时序电路中的信号产生延迟，这些信号延迟将产生时序漂移。同时晶体管的噪声容限也将会变小，更严重的将会导致器件失效。

相比于热载流子等研究比较成熟的方向，NBTI 相对来说是一个比较新的研究领域。在超深亚微米时代，VLSI 中的 NBTI 效应对器件的可靠性影响也越来越严重，国际上对 NBTI 效应的研究逐渐成为热点。国内外高校、公司以及 IC 生产厂家都对 NBTI 效应展开了研究。

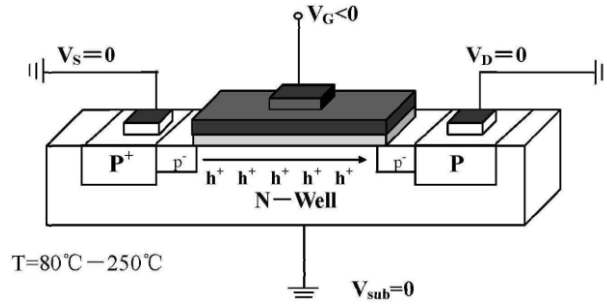


图 1-9 NBTI 效应器件模型示意图

目前对 NBTI 的研究领域主要集中在 NBTI 的退化现象、退化机理、动态 NBTI 效应的研究以及不同工艺条件对 NBTI 效应的影响等几个方面<sup>[27]</sup>，并且，在这些领域中都取得了一定的结果和进展。人们普遍认为，界面陷阱电荷的产生以及氧化层固定正电荷的产生是导致 NBTI 效应的主要原因， $H_2$  和  $H_2O$  是导致器件退化的主要物质。导致 NBTI 退化的机理主要是在 Si/SiO<sub>2</sub> 界面产生的界面陷阱和氧化层中正电荷的积累<sup>[24]</sup>。人们为此建立了大量的模型来研究，主要的模型有氢动力模型、电化学模型、反应-扩散（R-D）模型，其中，反应-扩散模型得到普遍的认可<sup>[33]</sup>。反应-扩散模型把产生 NBTI 的原因基于以下观点：在高温环境以及大的负偏置栅压环境下，PMOS 管反型层中的空穴受到热激发，这种空穴的能量很高，它将遂穿到硅/二氧化硅的界面处。这种热激发的空穴在与界面处的 Si-H 键产生相互作用从而生成 H 原子，这些 H 原子相互结合产生氢气从而被释放。而在器件的界面处会生成悬挂键。这些原因将导致器件的阈值电压向负向漂移，即阈值电压的绝对值将会变大。Jeppson、Svensson<sup>[35]</sup>以及 Rangan<sup>[37]</sup>等人，还有德国 Infineon 公司、TI 公司都对 NBTI 效应展开了大量的研究<sup>[38][39]</sup>。尽管人们开始重视 NBTI 效应产生的影响，但对 NBTI 的研究仍然有很大的空间和未知。与国外相比，我国对 NBTI 效应的研究相对来说要少很多，还有很多未知需要我们进一步探索研究。

### 1.3 主要研究内容

本文的主要研究任务是基于一歩大数逻辑可译码，对存储器抗单粒子翻转以及多位翻转进行加固设计，并对 SEU、MBU 等软错误和 NBTI 老化效应引起的错误进行检测区分，同时对软错误进行纠正。同时为加固的 SRAM 设计 AHB 总线接口，使得具有 NBTI 检测机制的 SRAM 能够在系统中集成使用。

本文共包括四章，以下分别对每一章节的内容安排做简要介绍：

第 1 章对本文的研究背景以及研究目的和意义做简单的介绍。并介绍单粒子翻转和多位错误翻转的来源、形式和危害以及负偏置温度不稳定性的简单介绍，



最后对存储器的抗辐射加固方法和 NBTI 效应的产生和危害做简要的介绍。

第 2 章主要介绍编码理论基础，对线性分组码原理进行介绍，同时介绍一步大数逻辑译码，为后续工作提供理论基础。

第 3 章主要介绍了选用的一步大数逻辑译码的构造，并对该大数逻辑译码进行编码设计和译码设计，并用 Memory Compiler 生成了一个宽度为 58 位，深度为 1024 位的存储器模型，对该存储器进行加固并设计故障注入模块来验证编码对存储器的加固结果。

第 4 章首先设计了应用前述一步大数逻辑译码进行的 SRAM 加固，并在此基础上设计增加对 NBTI 错误的检测机制，并设计 AHB 总线接口，将具有 NBTI 错误检测功能的 SRAM 放在总线上进行系统级验证。

最后，对本论文进行总结，同时对未来研究给出展望。

## 第2章 编码理论基础

本文通过 ECC 电路进行数据存储过程中的加固, 采用编码的方式进行存储加固, 在读出数据时对加固的编码进行译码并进行错误检测, 所以, 在此简要介绍与编码和译码有关的数学原理和概念, 并给出各个推导公式。

### 2.1 基本数学概念

#### 2.1.1 群

令  $G$  为一组元素的集合,  $G$  上的一个二元运算 (binary operation) 定义为一种规则: 对  $G$  中的每一对元素  $a$  和  $b$ , 由该规则在  $G$  中可以唯一确定第三个元素  $c = a + b$ 。这里称这种运算是封闭的。如果  $G$  上的二元运算对  $G$  中任意的  $a$ 、 $b$ 、 $c$  满足:

$$A * (b * c) = (a * b) * c \quad (2-1)$$

则称二元运算满足结合律 (association)。

一个定义了上述二元运算的集合  $G$  如果满足以下条件, 则称其为群:

1)、二元运算满足结合律;

2)、 $G$  中包含一个元素  $e$ , 使得对于  $G$  中任意的一个元素  $a$  都有

$$a * e = e * a = a \quad (2-2)$$

元素  $e$  称为  $G$  的单位元 (identity element)。

3)、对  $G$  中任意元素  $a$ , 在  $G$  中存在另外一个元素  $a'$ , 满足

$$a * a' = a' * a = e \quad (2-3)$$

元素  $a'$  称为  $a$  的逆元, 相同的,  $a$  也是  $a'$  的逆元。

如果群  $G$  在满足上述二元运算的同时满足式(2-4)的条件

$$a * b = b * a \quad (2-4)$$

则称群  $G$  是交换群 (commutative)。

#### 2.1.2 域

一般来说, 在一个集合中的元素, 对其进行加减乘除四则运算后, 其运算结果还是在这个集合中, 那么, 这个集合中的元素就被称为是域 (field)。域中元素的个数叫做域的阶 (order)。如果这个域中元素的个数是有限的, 那么这个域为有

限域 (finite field), 又被称为伽罗华域 (Galois Field, GF)。它与实数域以及有理数域有着不同的性质, 其中有限域中的二元域 (binary field), 记作  $GF(2)$ 。计算机与数字通信系统以及电路存储系统中, 人们广泛的采用编码理论, 而其中二元域有着举足轻重的作用。

表 2-1 模 2 加法

+	0	1
0	0	1
1	1	0

表 2-2 模 2 乘法

$\square$	0	1
0	0	0
1	0	1

如表 2-1 所示为模 2 加法规则表, 表 2-2 所示为模 2 乘法规则表。在二元运算中模 2 加法和模 2 乘法有着重要的作用, 加法和异或运算是相同的。二元域  $GF(2)$  通常被称为  $GF(2^m)$  的基域,  $GF(2^m)$  的特征值为 2。

## 2.2 向量空间

向量空间 (vector space) 又称为线性空间, 令  $V$  为定义了二元运算加法的集合,  $F$  是一个域, 其中的元素和  $V$  中的元素之间定义了一个乘法运算。若该运算满足以下条件:

- 1、 $V$  对于加法是一个交换群;
- 2、对  $F$  中的任意元素  $a$  和  $V$  中的任意元素  $v$ ,  $a \cdot v$  是  $V$  中的元素;
- 3、(分配率) 对  $V$  中的任意元素  $u$  和  $v$  及  $F$  中的任意元素  $a$  和  $b$ , 有

$$a \cdot (u + v) = a \cdot u + a \cdot v, \quad (a + b) \cdot v = a \cdot v + b \cdot v$$

- 4、(结合律) 对任意的  $v \in V$  和任意  $a, b \in F$ ,

$$(a \cdot b) \cdot v = a \cdot (b \cdot v)$$

设  $1$  为  $F$  的幺元, 则对任意的  $v \in V$ ,  $1 \cdot v = v$ 。则称集合  $V$  为域  $F$  上的一个向量空间。 $V$  中的元素称为向量 (vector),  $F$  中的元素称为标量 (scalar)。 $V$  中的加法称为向量加法 (vector addition), 将  $F$  中的一个标量与  $V$  中的一个向量相乘得到  $V$  中的一个向量的乘法称为标乘 (scalar multiplication) 或标积 (scalar product)。 $V$

中的加法单元位记为 0。

对于  $n$  维向量的线性相关性的主要性质如下所述：

1、如果  $\alpha_1, \alpha_2, \dots, \alpha_n$  是一组线性无关的向量，则从中任意取出若干个向量都是线性无关的（整体无关，部分必无关）。

2、向量的线性相关：向量空间中的一组向量  $v_1, v_2, \dots, v_n$ ，当存在  $n$  个不全为零的标量  $a_1, a_2, \dots, a_n$  使得  $a_1 v_1 + a_2 v_2 + \dots + a_n v_n = 0$ ，则称这组向量线性相关。相反的，若不存在  $n$  个不全为零的标量  $a_1, a_2, \dots, a_n$  使得  $a_1 v_1 + a_2 v_2 + \dots + a_n v_n = 0$ ，则称这组向量线性无关。

3、若  $r$  维向量组  $\alpha_{i1}, \alpha_{i2}, \dots, \alpha_{ir}$   $i=1, 2, \dots, s$  线性无关，则在每一个向量上再添加  $n-r$  个分量得到的  $n$  维向量组  $\beta_i = (\alpha_{i1}, \alpha_{i2}, \dots, \alpha_{ir}, \alpha_{i,r+1}, \dots, \alpha_{i,n})$ ， $i=1, 2, \dots, s$  也线性无关（线性无关的向量组其加维向量组也线性无关）。

4、向量正交：若  $\alpha = \alpha_0, \alpha_1, \dots, \alpha_{n-1}$  和  $\beta = \beta_0, \beta_1, \dots, \beta_{n-1}$  是  $V_n$  中的两个  $n$  维向量。内积定义为： $\alpha \times \beta = \alpha_0 \times \beta_0 + \alpha_1 \times \beta_1 + \dots + \alpha_{n-1} \times \beta_{n-1}$ ，若  $\alpha \times \beta = 0$ ，则称向量  $\alpha$  和  $\beta$  正交。

## 2.3 线性分组码

目前在大多数的数字通信系统和存储系统中都是以二进制数字 0 和 1 进行编码的。其中线性分组码（linear block codes）是一种广泛被应用的编码种类。其通过在输入数据中添加一定的冗余校验位对数据进行编码，然后在读取数据的时候通过译码器在不占用大量信息存储速率的情况下，对信道或存储器中的数据进行纠错检错。本节简单介绍线性分组码的构成原理。

### 2.3.1 线性分组码原理

当用线性方程组联系分组码时，若其信息码元与监督码元之间的关系为线性时，这种分组码称为线性分组码。在现在的通信系统和存储系统中都是使用二进制数字 1 和 0 开进行传输和存储。这些 1 和 0 组成了一个固定长度的信息。设这个信息的总长度为  $u$ ，有效信息位的总长度为  $k$ ，则该信息的冗余位为  $u-k$ ，而且这  $k$  个有效信息位一共可以组成  $2^k$  个信息。编码器按照编码规则把原始的有效信息转化为带有冗余校验位的二进制  $n$  维向量  $v(n > k)$ 。那么这  $n$  维的向量  $v$  就称为是信息  $u$  的码向量或者码字。所以，对于  $2^k$  个信息，相应的就有  $2^k$  个码字。如果要使分组码可用，那么这  $2^k$  个码字必须互不相同。信息  $u$  和向量  $v$  存在一一对应的关系。这些码字的集合称为一个分组码。

在线性分组码中，两个码字对应位上的数字不同的位数称为码字距离，简称距离。编码中各个码字之间的距离的最小值称为最小码距  $d$ ，最小码距决定了一个码检测随机错误和纠正随机错误的能力。最小码距  $d$  和纠检错能力有如下关系：

- 1)、为了检测  $e$  个错码，则要求最小码距  $d > e + 1$ ；
- 2)、为了纠正  $t$  个错码，则要求最小距离  $d > 2t + 1$ ；
- 3)、为了纠正  $t$  个错码，同时检测  $e$  个错码，则要求最小码距  $d > e + 1, e > t$ 。

### 2.3.1.1 生成矩阵

由前述可知，一个  $(n, k)$  线性码  $C$  是一个向量空间，这个向量空间的维数是  $k$ ，其属于  $n$  维空间的子空间，这里  $n > k$ 。那么，将有  $k$  个线性独立的码字  $g_0, g_1, \dots, g_{k-1}$ ，这些码字使线性码  $C$  中的每一个码字  $v$  都可以用这  $k$  个码字的线性组合来表示。可以用公式(2-5)表示其组成关系。

$$v = u_0 g_0 + u_1 g_1 + \dots + u_{k-1} g_{k-1} \quad (2-5)$$

式(2-5)中， $u_i = 0$  或  $1$ ， $0 \leq i < k$ 。以这  $k$  个线性独立的码字为行向量，可以得到一个  $k \times n$  阶矩阵如式(2-6)所示：

$$G = \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{k-1} \end{bmatrix} = \begin{bmatrix} g_{00} & g_{01} & g_{02} & \cdots & g_{0,n-1} \\ g_{10} & g_{11} & g_{12} & \cdots & g_{1,n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ g_{k-1,0} & g_{k-1,1} & g_{k-1,2} & \cdots & g_{k-1,n-1} \end{bmatrix} \quad (2-6)$$

式(2-6)中  $g_i = (g_{i0}, g_{i1}, \dots, g_{i,n-1})$ ,  $0 \leq i < k$ 。如果  $u = (u_0, u_1, \dots, u_{k-1})$  是待编码的信息序列，则相应的码字可以由如下式(2-7)给出：

$$v = u \cdot G = (u_0, u_1, \dots, u_{k-1}) \cdot \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{k-1} \end{bmatrix} = u_0 g_0 + u_1 g_1 + \dots + u_{k-1} g_{k-1} \quad (2-7)$$

可以看出， $(n, k)$  线性码  $C$  是由矩阵  $G$  的行生成的。这里称矩阵  $G$  是码  $C$  的生成矩阵 (generator matrix)。因为生成矩阵的行向量是线性无关的，所以在  $(n, k)$  线性码中任意选择  $k$  个线性独立的码字均可以构成该码的生成矩阵。而这个线性码的组成也只和生成矩阵的行向量有关系。因此，为了使电路的复杂度降低，在编码器中仅仅存储生成矩阵  $G$  的  $k$  个线性无关的行向量就可以了，并根据输入信息  $u = (u_0, u_1, \dots, u_{k-1})$  构成  $k$  个行向量的一个线性组合。

对于线性分组码，它具有图 2-1 所示的结构。其中这  $k$  位信息就是原始的信息。

冗余校验信息是信息的总长度减去原始的信息得到的，称这  $n-k$  位信息为奇偶校验（parity-check）位，奇偶校验位是由信息位的线性和构成的。

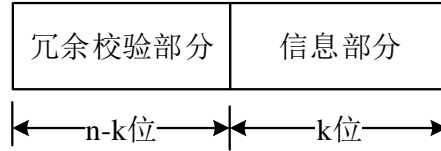


图 2-1 错误纠正码的系统结构

### 2.3.1.2 校验矩阵

在编码原理中，线性分组码的奇偶校验矩阵（parity-check matrix）与生成矩阵一样也尤为重要。奇偶校验矩阵记为  $H$ ，满足条件，即当且仅当  $v \cdot H^T = 0$  时，矩阵  $H$  被称为该码的奇偶校验矩阵。其中  $v$  是  $n$  维的向量，该向量是由生成矩阵  $G$  生成的  $(n, k)$  线性码  $C$  中的一个码字，该码被称为  $H$  的零空间。奇偶校验矩阵  $H$  的行向量有  $n-k$  个，这些个行向量的线性组合共有  $2^{n-k}$  种所有的组合构成线性码  $C$  的对偶码  $C_d$ ， $C_d$  为满足对任何  $v \in C$  和  $w \in C_d$ ，有  $v \cdot w = 0$  的  $(n, n-k)$  的线性码。因此，一个线性码  $C$  的奇偶校验矩阵是其对偶码  $C_d$  的生成矩阵。

式(2-8)为  $(n, k)$  线性码的  $k \times n$  阶生成矩阵  $G$ 。

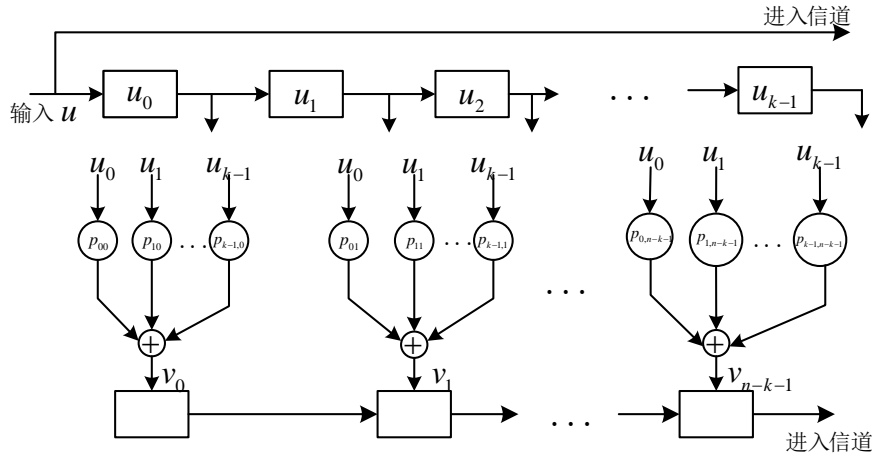
$$G = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ \vdots \\ g_{k-1} \end{bmatrix} = \begin{bmatrix} p_{00} & p_{01} & \cdots & p_{0,n-k-1} & 1 & 0 & 0 & \cdots & 0 \\ p_{10} & p_{11} & \cdots & p_{1,n-k-1} & 0 & 1 & 0 & \cdots & 0 \\ p_{20} & p_{21} & \cdots & p_{2,n-k-1} & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ p_{k-1,0} & p_{k-1,1} & \cdots & p_{k-1,n-k-1} & 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \quad (2-8)$$

其中， $p_{ij} = 0$  或  $1$ 。令  $I_k$  表示  $k \times k$  阶单位阵，则有  $G = [P \ I_k]$ 。它的奇偶校验矩阵如式(2-9)所示，式中  $P^T$  是矩阵  $P$  的转置矩阵。

$$H = [I_{n-k} \ P^T] = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & p_{00} & p_{10} & \cdots & p_{k-1,0} \\ 0 & 1 & 0 & \cdots & 0 & p_{01} & p_{11} & \cdots & p_{k-1,1} \\ 0 & 0 & 1 & \cdots & 0 & p_{02} & p_{12} & \cdots & p_{k-1,2} \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 & p_{0,n-k-1} & p_{1,n-k-1} & \cdots & p_{k-1,n-k-1} \end{bmatrix} \quad (2-9)$$

### 2.3.2 编码电路原理

由上述所知， $(n, k)$  线性分组码的编码电路可以由图 2-2 所示。


 图 2-2 通用  $(n, k)$  线性分组码编码电路

其中，图 2-2 中 “ $\rightarrow \square \rightarrow$ ” 表示一个移位寄存器（如触发器），“ $\oplus$ ” 表示模 2 加法器，其计算规则如表 2-1 所示。待编码信息  $u = (u_0, u_1, \dots, u_{k-1})$  同时移位进入信息寄存器和信道，在  $n-k$  个模 2 加法器的输出端便生成了  $n-k$  个校验位  $v_0, v_1, \dots, v_{n-k-1}$ ，这些校验位随即串行移位进入信道中。这里可以看到编码电路的复杂度随着码的分组长度的增加而线性增加。

### 2.3.3 校正子与差错检测

一个  $(n, k)$  线性码，其生成矩阵记为  $G$ ，奇偶校验矩阵记为  $H$ 。令  $v = (v_0, v_1, \dots, v_{n-1})$  表示要通过有噪声信道传输的码字， $r = (r_0, r_1, \dots, r_{n-1})$  为信道输出端接收到的码字。由于传输信道中的噪声， $r$  可能和  $v$  不相同，向量和

$$e = r + v = (e_0, e_1, \dots, e_{n-1}) \quad (2-10)$$

是一个  $n$  维向量，其中  $r_i \neq v_i$  时， $e_i = 1$ ，而  $r_i = v_i$  时， $e_i = 0$ 。该  $n$  维向量称为差错向量（error vector）或错误模式（error pattern），它直接指出了接收向量  $r$  和传输码字  $v$  两者之间不同的位置。 $e$  中的 1 是由于信道噪声引起的传输差错。由式(2-10)可知，接收向量  $r$  是传输码字和差错向量的向量和，即：

$$r = e + v \quad (2-11)$$

译码器对接收的数据进行译码输出的时候，首先需要判断读出数据是否包含差错，此时译码器需要计算如式(2-12)所示的  $(n-k)$  维向量。

$$s = r \cdot H^T = (s_0, s_1, \dots, s_{n-k-1}) \quad (2-12)$$

称  $s$  为  $r$  的校正子（syndrome）。码字  $r$  和校正子  $s$  有如式(2-13)所示关系。

$$s = \begin{cases} 0, & r \text{ 为码字} \\ 1, & r \text{ 不为码字} \end{cases} \quad (2-13)$$

因此当  $s \neq 0$  时，可以判定  $r$  不为码字，这时差错将被检测到；当  $s = 0$  时，可以判定  $r$  为码字，此时输出的数据是正确的。但是当  $s = r \cdot H^T = 0$  时，如果这时错误模式  $e$  等于某个非零码字， $r$  也可能发生错误。 $r$  的值为两个码字的和，此时有关系式  $r \cdot H^T = 0$ 。上述情况被称为漏检错误模式。通常译码器产生译码差错是由于发生了漏检错误模式，漏检错误模式的个数与信息中非零码字的个数相同，即  $2^k - 1$  个。校正子和差错向量之间有如式(2-14)所示关系。

$$s = e \cdot H^T \quad (2-14)$$

校正子提供了有关差错位的信息，因此可以用来纠正错误。 $(n, k)$  线性分组码的通用校正子电路如图 2-3 所示。

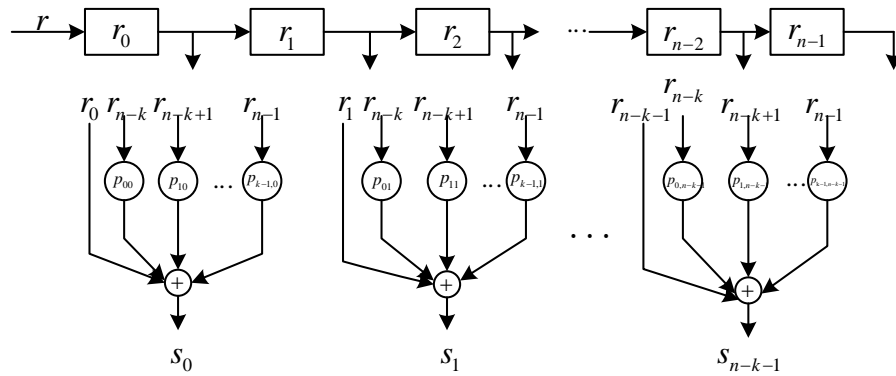


图 2-3 通用  $(n, k)$  线性分组码校正子电路

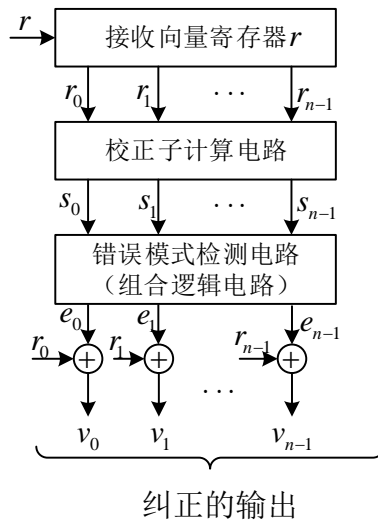


图 2-4 线性分组码通用译码器



### 2.3.4 译码电路原理

由上述可知，可纠正错误模式和校正子之间存在一一对应关系。利用这种一一对应关系可以得到信息接收端的译码步骤：

- 1、计算  $r$  的校正子  $r \cdot H^T$ ；
- 2、确定校正子等于  $r \cdot H^T$  的被假定为由信道引起的错误模式  $e_i$ ；
- 3、将接收向量  $r$  译为码字  $v^* = r + e_i$ 。

这种译码方案被称为校正子译码(syndrome decoding)或查表译码(table lookup decoding)。基于查找表方案的  $(n, k)$  线性分组码的通用译码器如图 2-4 所示。

### 2.4 一步大数逻辑译码

一个  $(n, k)$  循环码  $C$ ，其奇偶校验矩阵为  $H$ ，则  $H$  的行空间是一个  $(n, n-k)$  循环码，记为  $C_d$ ，它是  $C$  的对偶码。对于码  $C$  中的任意一个码字  $v$  和码  $C_d$  中的任意一个码字  $\omega$ ，其内积为零，即：

$$\omega \cdot v = \omega_0 v_0 + \omega_1 v_1 + \cdots + \omega_{n-1} v_{n-1} = 0 \quad (2-15)$$

式(2-15)称为奇偶校验方程 (parity-check equation)。可以看到，总共有  $2^{n-k}$  个这样的奇偶校验方程。

码  $C$  中的码字  $v$  发送后，令  $e = (e_0, e_1, \cdots, e_{n-1})$  和  $r = (r_0, r_1, \cdots, r_{n-1})$  分别为误差向量和接收向量，那么接收向量和码字之间的关系如式(2-16)所示。

$$r = v + e \quad (2-16)$$

$$A = \omega \cdot r = \omega_0 r_0 + \omega_1 r_1 + \cdots + \omega_{n-1} r_{n-1} \quad (2-17)$$

在对偶码  $C_d$  中的任意一个向量  $\omega$ ，这个向量和  $r$  的线性和如式(2-17)所示。该线性和被称为奇偶校验和 (parity-check sum)。若接收向量  $r$  是码  $C$  中的一个码字，那么该奇偶校验和  $A$  一定为 0；同理，若  $r$  不是  $C$  的码字，则奇偶校验和  $A$  可能不是 0。由式(2-15)和式(2-16)以及式(2-17)，可以得到校验和  $A$  与误差向量  $e$  中差错位之间的关系如式(2-18)所示。

$$A = \omega_0 e_0 + \omega_1 e_1 + \cdots + \omega_{n-1} e_{n-1} \quad (2-18)$$

如果系数  $\omega_i = 1$ ，则差错位  $e_i$  被校验和  $A$  所检验。

假设在对偶码  $C_d$  中存在  $J$  个向量

$$\begin{aligned} \omega_1 &= (\omega_{10}, \omega_{11}, \cdots, \omega_{1,n-1}) \\ \omega_2 &= (\omega_{20}, \omega_{21}, \cdots, \omega_{2,n-1}) \\ &\vdots \\ \omega_J &= (\omega_{J0}, \omega_{J1}, \cdots, \omega_{J,n-1}) \end{aligned} \quad (2-19)$$

其具有下述性质：

1、每个向量的第  $n-1$  个分量为 1，即：

$$\omega_{1,n-1} = \omega_{2,n-1} = \cdots = \omega_{J,n-1} = 1 \quad (2-20)$$

2、当  $i \neq n-1$  时，任何一个向量的第  $i$  个分量等于 1 的个数为 0 或者是 1。例如，若  $\omega_{1,i} = 1$ ，则有：

$$\omega_{2,i} = \omega_{3,i} = \cdots = \omega_{J,i} = 0 \quad (2-21)$$

这  $J$  个向量就被称作是在向量的第  $n-1$  个位置上正交 (orthogonal)，称此向量为正交向量。由这  $J$  个正交向量可以构造  $J$  个奇偶校验和：

$$\begin{aligned} A_1 &= \omega_1 \cdot r = \omega_{10}r_0 + \omega_{11}r_1 + \cdots + \omega_{1,n-1}r_{n-1} \\ A_2 &= \omega_2 \cdot r = \omega_{20}r_0 + \omega_{21}r_1 + \cdots + \omega_{2,n-1}r_{n-1} \\ &\vdots \\ A_J &= \omega_J \cdot r = \omega_{J0}r_0 + \omega_{J1}r_1 + \cdots + \omega_{J,n-1}r_{n-1} \end{aligned} \quad (2-22)$$

因为  $\omega_{1,n-1} = \omega_{2,n-1} = \cdots = \omega_{J,n-1} = 1$ ，这  $J$  个奇偶校验和与误差向量的差错位具有如式(2-23)所示的关系。

$$\begin{aligned} A_1 &= \omega_{10}e_0 + \omega_{11}e_1 + \cdots + \omega_{1,n-1}e_{n-2} + e_{n-1} \\ A_2 &= \omega_{20}e_0 + \omega_{21}e_1 + \cdots + \omega_{2,n-1}e_{n-2} + e_{n-1} \\ &\vdots \\ A_J &= \omega_{J0}e_0 + \omega_{J1}e_1 + \cdots + \omega_{J,n-1}e_{n-2} + e_{n-1} \end{aligned} \quad (2-23)$$

可见，差错位  $e_{n-1}$  能够被上述所有的校验和所检验。由于  $\omega_1, \omega_2, \cdots, \omega_J$  正交，且具有上述式(2-21)的性质，任意一个不是在  $e_{n-1}$  上的差错位，如果该差错位能够被校验和检验，那么该校验和的个数最多为 1。这  $J$  个奇偶校验和就称为是在差错位  $e_{n-1}$  上正交的。因为在对偶码中的任意一个向量的取值只有 0 和 1，即： $\omega_{i,j} = 0$  或 1。所以在  $e_{n-1}$  上正交的这  $J$  个奇偶校验和可以写成如式(2-24)的形式。

$$A_j = e_{n-1} + \sum_{i \neq n-1} e_i \quad (2-24)$$

如果校验和  $A_j$  中所有  $i \neq n-1$  的差错位均为 0，那么  $e_{n-1}$  的值就等于  $A_j$ （即  $e_{n-1} = A_j$ ）。由上所述，在  $e_{n-1}$  上正交的  $J$  个奇偶校验和可以对数据的接收向量的第  $n-1$  位译码。

如果误差向量  $e = (e_0, e_1, \cdots, e_{n-1})$  中的差错位的个数最多不超过  $J/2$ 。也就是说：误差向量  $e = 1$  的个数最多不超过  $J/2$ 。如果  $e_{n-1} = 1$ ，那么其他的非零差错位将会分布在不多于  $[J/2]-1$  个在  $e_{n-1}$  上正交的奇偶校验和中。因此在  $e_{n-1}$  上正交，且其值等于 1 的奇偶校验和的个数不少于  $J - [J/2] + 1$  个。如果  $e_{n-1} = 0$ ，在校验和中含有非零差错位的个数将不多于  $J/2$ 。因此，可以得到在  $e_{n-1}$  上正交的奇偶校验和的值等

于 0 的个数少于校验和总数的一半。所以，在  $e_{n-1}$  上正交的奇偶校验和的取值结果只有 0 和 1。若是 0 的个数比 1 的个数多，或者两者数量相等，那么  $e_{n-1}=0$ ；相反，若是 1 的个数比 0 的个数多，则  $e_{n-1}=1$ 。综上所述，若在差错位  $e_{n-1}$  上正交的奇偶校验和的绝对多数为 1，那么差错位  $e_{n-1}=1$ ；否则， $e_{n-1}=0$ 。

如果误差向量中的差错位的个数少于或等于  $J/2$ ，那么对差错位  $e_{n-1}$  进行译码的输出结果将是正确的。由于这种编码具有循环对称性，所以将数据进行循环移位之后，可以构造与第  $n-1$  位上完全相同的其他任意差错位上的译码。前述译码算法叫做一步大数逻辑译码 (one-step majority-logic decoding)。如果在任意一个差错位上正交的奇偶校验和的最大数值为  $J$ ，那么，通过一步大数逻辑译码，对接收向量中少于或等于  $J/2$  个差错的任意错误模式，都能够被纠正。

## 2.5 本章小结

本章对数学原理中的群和域做了简单介绍，并介绍了向量空间的性质，同时阐述了线性分组码的构成原理，生成矩阵和校验矩阵的关系，编码方法和译码方法，以及校正子与差错检测。本章最后介绍了一步大数逻辑译码的定义和生成规则以及它的纠检错能力。

## 第3章 大数逻辑可译码加固 SRAM

随着半导体工艺尺寸的不断减小，芯片单位面积上集成的器件逐渐增多，在一次辐射事件中，存储器中发生单粒子翻转的概率以及存储器中数据发生多位翻转的概率都在逐渐增大。同时，由于对存储器的长时间读写访问操作，存储器单元会随着时间产生老化，这将引起存储器数据产生变化甚至使存储器件失效。为了增加存储器件的稳定性和可靠性，由上一章节介绍的编码理论为根据，本章提出针对单粒子翻转和存储器多位故障翻转的存储器加固设计。

### 3.1 一步大数逻辑译码的构造

由于器件尺寸的减小，单粒子翻转和多位翻转的概率大大增加以及 NBTI 效应引起的存储单元老化错误都是不定的，所以这里需要一种纠错检错能力范围大的编码以满足设计要求。

现有的多位错误修正码有 RS 码，BCH 码和 LDPC 码<sup>[20]</sup>等。一个  $(n, k)$  线性码  $C$ ，如果每个码字的循环移位仍是  $C$  的码字，则称其为循环码 (cyclic code)。循环码属于线性分组码的一种。循环码大多使用带反馈的移位寄存器（主要是线性时序电路），这使得要想获得循环码的编码和校正并不困难。而且循环码具有固有的代数结构，所以对于该码的译码计算方法相对来说较多。在通信系统的差错控制应用中，循环码被广泛的应用。1957 年，Eugene Prange 第一次进行了对循环码的研究。在此之后人们建立起多种循环码，包括：BCH 码、里德-索罗门 (Reed-Solomon) 码、欧式几何码、二次剩余码等。文献[54]中研究表明循环码在突发差错检测方面是非常有效的。本文选用一步大数逻辑可译码，对于某些分组码来说，它是一种简单有效的方法，尤其对于某些循环码。迄今为止，已经找到的大数逻辑可译码大部分是循环码。这一类重要的循环码大多是基于有限几何（欧式几何和射影几何）构造的。所以这类码也被称为有限几何码 (finite geometry codes)。

#### 3.1.1 一类一步大数逻辑可译码

在本节中，本文介绍一种基于对称结构而构造的一步大数逻辑可译循环码。令  $\alpha$  为伽罗华域  $GF(2^m)$  中的一个元素。通过伽罗华域  $GF(2^m)$  中的元素，用数字

标记扩展码  $C_e$  中向量  $v_e = (v_\infty, v_0, v_1, \dots, v_{2^m-1})$ ，其中  $\alpha^\infty = 0$ ， $\alpha^0 = 1$ ， $v_i = \alpha_i, 1 \leq i \leq 2^m - 1$ 。这些数字称为位置数 (location numbers)。令  $Y$  表示  $v_e$  的某个分量的位置，将  $v_e$  的分量从位置  $Y$  变换到  $Z = aY + b$ ，其中  $a, b$  为伽罗华域  $GF(2^m)$  中的元素，且  $a \neq 0$ 。该置换被称为仿射置换 (affine permutation)。一个仿射置换将一个具有  $2^m$  个分量的向量变化成另一个具有  $2^m$  个分量的向量。一个长度为  $2^m - 1$  的循环码，若其生成多项式满足定理 3-1 所给出的条件，则其具有双可迁不变 (doubly transitive invariant, DTI) 性。

定理 3-1: 令  $C$  为一个由  $g(X)$  生成的、长度为  $n = 2^m - 1$  的循环码。令  $C_e$  为由码  $C$  添加一位全局奇偶校验而得到的扩展码。令  $\alpha$  为伽罗华域  $GF(2^m)$  的一个本原。那么，扩展码  $C_e$  在仿射置换下具有不变性的充要条件是：对于码  $C$  生成多项式  $g(X)$  的每一个根  $\alpha^h$  和每一个在  $\Delta(h)$  中的每个  $h'$ ， $\alpha^{h'}$  也是  $g(X)$  的一个根，而  $\alpha^0 = 1$  不是  $g(X)$  的根。

在 2.4 节中介绍了一步大数逻辑译码的概念，其中参数  $t_{ML} = J/2$  被称为该一步大数逻辑译码的大数逻辑纠错能力 (majority-logic error-correcting capability)。令  $d_{\min}$  为该码的最小距离，如果要想使一步大数逻辑译码是有效的，那么，参数  $t_{ML} = J/2$  和一步大数逻辑译码的纠错能力  $t = (d_{\min} - 1)/2$  必须相等或接近。即， $J$  应当等于或接近于  $d_{\min} - 1$ 。

定义：具有最小距离  $d_{\min}$  的循环码被称为在一步中完全可正交 (completely orthogonalizable)，当且仅当可以构造出  $J = d_{\min} - 1$  个在某个差错位置上正交的奇偶校验和。

给定一个  $(n, k)$  循环码  $C$ ，对该码的大数逻辑译码可以在其一个差错位上构造出  $J$  个正交的奇偶校验和。首先，从该码的零空间  $C_d$  中取一组在最高位  $X^{n-1}$  上正交的  $J$  个向量  $\omega_1, \omega_2, \dots, \omega_J$ 。于是在差错位  $e_{n-1}$  上正交的  $J$  个奇偶校验和  $A_1, A_2, \dots, A_J$  可以由这  $J$  个正交向量和接收向量  $r$  构成。由公式(2-22)得，向量  $\omega_j$  表示了需要被加起来以构成校验和  $A_j$  的接收数位。 $J$  个校验和可以通过  $J$  个多输入模 2 加法器来计算。当计算出这  $J$  个校验和时，它们就可以被用来作为  $J$  个大数逻辑判决器的输入。当且仅当一半以上的输入不是 1 时，大数逻辑判决器的输出才是 0；否则，输出为 1。该输出为差错位  $e_{n-1}$  的估计值。图 3-1 所示的是一个通用的一步大数逻辑译码器。该译码器被称为 II 型一步大数逻辑译码器<sup>[54]</sup>，其纠错过程如下：

第一步：门 1 开启，门 2 关闭，将接收向量  $r$  存入缓存器中。

第二步：求出相应的接收位上的  $J$  个奇偶校验和，这些校验和在差错位  $e_{n-1}$  上正交。

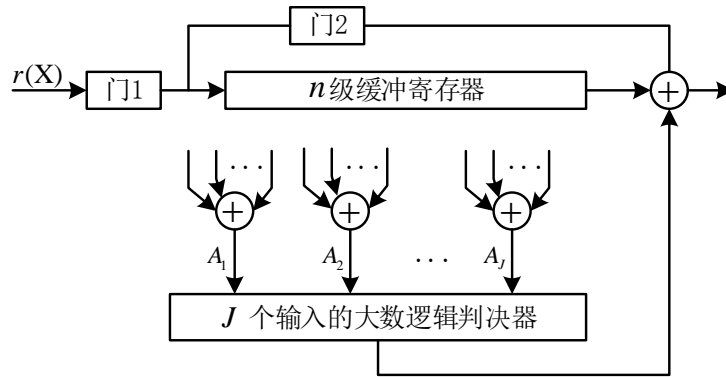


图 3-1 通用 II 型一步大数逻辑译码器

第三步：把这  $J$  个正交的校验和送到大数逻辑判决器。从缓冲器中读出第一个接收数位  $r_{n-1}$ ，并通过大数逻辑判决器的输出来纠正该接收位的差错。

第四步：在第三步的结尾处，保持门 2 开启，将缓存器向右移一位。则第二个接收数位将位于缓存器的最右端，第二个接收位的纠错过程和第一个接收位的纠错过程相同。译码器重复第二步和第三步。

第五步：通过相同的操作，对接收向量逐位进行译码，直到完成  $n$  次移位为止。

如果接收向量  $r$  含有  $J/2$  个或更少的差错，缓存器中所包含的向量就是发送码向量，而且在整个译码操作结束以后，大数逻辑判决器的输入将全部为 0。若大数逻辑判决器的输入不全为 0，那么就检测到一个不可纠正的错误模式 (uncorrectable error pattern)。

在一个差错位上正交的奇偶校验和还可以通过校正子来构造。正交向量  $\omega_1, \omega_2, \dots, \omega_j$  与校正子  $s$  以及校验和之间有如式(3-1)所示关系。

$$A_j = \omega_{j0}s_0 + \omega_{j1}s_1 + \dots + \omega_{j,n-k-1}s_{n-k-1} \quad (3-1)$$

式(3-1)中  $A_j$  表示为校验和，也就是校正子位的线性和，它的系数是正交向量  $\omega_j$  的前  $n-k$  个数位。根据公式(3-1)，下面介绍一步大数逻辑译码的另一种方法，其结构如图 3-2 所示，接收向量由校正子寄存器的右侧移入。该译码器被称为 I 型一步大数逻辑译码器<sup>[54]</sup>，其纠错过程如下：

第一步：将接收多项式  $r(X)$  移入校正子寄存器，按照通常的做法来计算校正子。

第二步：在差错位  $e_{n-1}$  上正交的  $J$  个奇偶校验和可以通过选取校正子中合适的数位进行求和得到，接下来将这  $J$  个奇偶校验和送到一个  $J$  个输入的大数逻辑判决器。

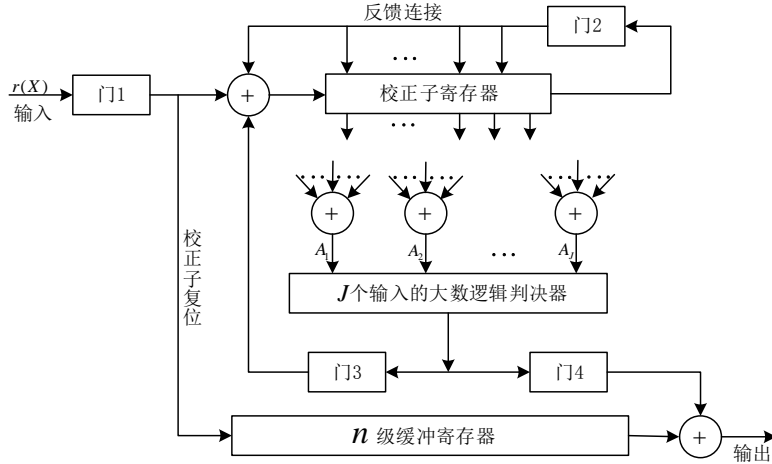


图 3-2 通用 I 型一步大数逻辑译码器

第三步：从缓存器中读出第一个接收数位，然后通过大数逻辑判决器的输出对其进行纠错。同时，将校正子寄存器移动一位（保持门 2 开启），并且  $e_{n-1}$  对校正子的作用也将消除（保持门 3 开启）。校正子寄存器中新的内容是修改后的接收向量向右循环移位一次后所得到的校正子。

第四步：利用第三步得到的新的校正子来对下一个接收位  $r_{n-2}$  进行译码。译码器重复第二步和第三步的操作。接收位  $r_{n-2}$  的纠错处理过程与第一个数位  $r_{n-1}$  的纠错过程完全一样。

第五步：通过相同的操作，译码器以同样的方式对接收向量  $r$  逐位译码，直到缓存器和校正子寄存器均进行了  $n$  步移位为止。

译码结束后，如果译码器的输出为一个码字，那么校正子寄存器的内容将全为 0。如果校正子寄存器的内容在译码结束后并非全部为 0，那么就检测到一个不可纠正的错误模式。如果我们希望只对接收的信息位进行译码，而忽略接收到的校验位，则缓存器只需存储  $k$  个接收信息位即可，它包含  $k$  级，在这种情况下，I 型和 II 型译码器所需的复杂度大致相当。

文献[54]研究表明，1 型双可迁不变（DTI）码的最小距离为  $L+1$ （ $L$  为向量重量），在一个差错位上可构造的正交奇偶校验和的数目具有式(3-2)所示上界。

$$\frac{2^m - 2}{L} = \frac{2^m - 1}{L} - \frac{1}{L} = J - \frac{1}{L} = J - 1 \quad (3-2)$$

因此，对于 1 型 DTI 码，在一个差错位置上可构造的正交奇偶校验和的数目等于其上界。由于  $J$  是奇数，则  $J/2 = (J-1)/2$ 。因此，0 型 DTI 码和 1 型 DTI 码具有相同的大数逻辑纠错能力。

一些一步大数逻辑可译 1 型 DTI 码如表 3-1 所示。

表 3-1 一些一步大数逻辑可译 1 型 DTI 码

$n$	$k$	$t_{ML}$		$n$	$k$	$t_{ML}$
15	9	1		2047	1211	11
	7	2			573	44
63	49	1		4095	3969	1
	37	4			3871	2
	13	10			3753	4
255	225	1			3611	6
	207	2			3367	32
	175	8			2707	17
	37	25			2262	19
	21	42			2074	22
511	343	3			1649	45
	139	36			1393	52
1023	961	1			1377	136
	833	5			406	292
	781	16			101	409
	151	46			43	682
	30	170				

表 3-1 中  $n$  表示信息总长度,  $k$  表示有效信息位数,  $n-k$  表示冗余位数,  $t_{ML}$  表示纠检错能力。

### 3.1.2 一步大数逻辑可译 1 型 (63,37) DTI 码

通常, 并没有一个简单的公式来给出一步大数逻辑可译 DTI 码 (0 型或 1 型) 的奇偶校验位的数目; 然而, 对于两个特殊情况, 可以得到  $n-k$  的准确计算公式 [54]:

情形 1: 对于  $m = 2sl$  和  $J = 2^l + 1$ , 长度为  $2^m - 1$  的 1 型 DTI 码的奇偶校验位的数目为:

$$n - k = (2^{s+1} - 1)^l - 1 \quad (3-3)$$

情形 2: 对于  $m = \lambda l$  和  $J = 2^l - 1$ , 长度为  $2^m - 1$  的 1 型 DTI 码的奇偶校验位的



数目为：

$$n-k = 2^m - (2^{\lambda} - 1)^l - 1 \quad (3-4)$$

短 DTI 码与 BCH 码相比较，效率基本相当。例如，存在一个 (63,37) 一步大数逻辑可译 1 型 DTI 码，该码能够纠正 4 个或更少的差错。具有相同长度的、能够纠正 4 个差错的 BCH 码是一个 (63,39) 码，该 BCH 码比 (63,37) 1 型 DTI 码多 2 个信息位；但是，(63,39) BCH 码的译码电路比 (63,37) 1 型 DTI 码的译码电路要复杂的多。对于较长的分组，在相同长度和相同纠错能力条件下，DTI 码的效率没有 BCH 码高，但是复杂度相对而言 DTI 码要简单的多。

文献[47]中研究表明，在一次辐射事件中，存储器中发生数据翻转的位数如图 3-3 所示，图中纵坐标为百分比率，横坐标为发生错误的位数。从图中可以看到存储器中数据发生翻转的位数主要集中在 4 位以下，其中发生 2 位错误的概率最大。

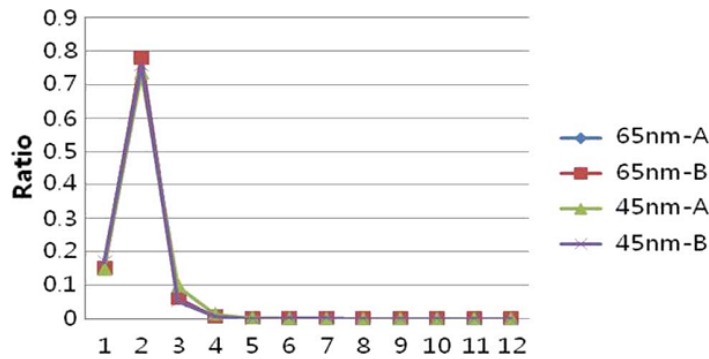


图 3-3 试验测试存储器发生翻转位数分布统计<sup>[47]</sup>

本文选择 (63,37) 码作为信息的纠错检错编码。由表 3-1 可以看到 (63,37) 码的纠检错能力  $t_{ML}=4$ 。在小于等于 4 的错误模式下，该码都可以进行纠错检错。根据前述编码规则可以计算得到该码的生成矩阵如图 3-4 所示。其中生成矩阵  $G$  的行向量是线性独立的， $G$  维数是一个  $37 \times 63$  的矩阵，右侧 37 列为一个  $37 \times 37$  的单位矩阵。

本文对 32 位数据进行存储加固，根据生成矩阵  $G$  的行向量是线性无关的性质，这里我们只需要用到图 3-4 中虚线左上角的  $32 \times 58$  阶矩阵即可，如图中虚线所示，这也降低了编码器的消耗。本文用的 (63,37) 码的校正子如表 3-2 所示，其中数字表示信息字的位置。

$$G =$$

图 3-4 (63,37) 码生成矩阵

表 3-2 (63,37) 码校正子

校正子	位置
$\omega_1$	$= 6+30+40+41+44+56+61+63$
$\omega_2$	$= 24+34+35+38+50+55+57+63$
$\omega_3$	$= 2+8+32+42+43+46+58+63$
$\omega_4$	$= 5+7+13+37+47+48+51+63$
$\omega_5$	$= 3+15+20+22+28+52+62+63$
$\omega_6$	$= 10+11+14+26+31+33+39+63$
$\omega_7$	$= 12+17+19+25+49+59+60+63$
$\omega_8$	$= 1+4+16+21+23+29+53+63$

根据 3.1.1 节中图 3-1 和图 3-2 以及一步大数逻辑可译码的编译码规则，得到 (63,37) 码的译码电路结构如图 3-5 所示。

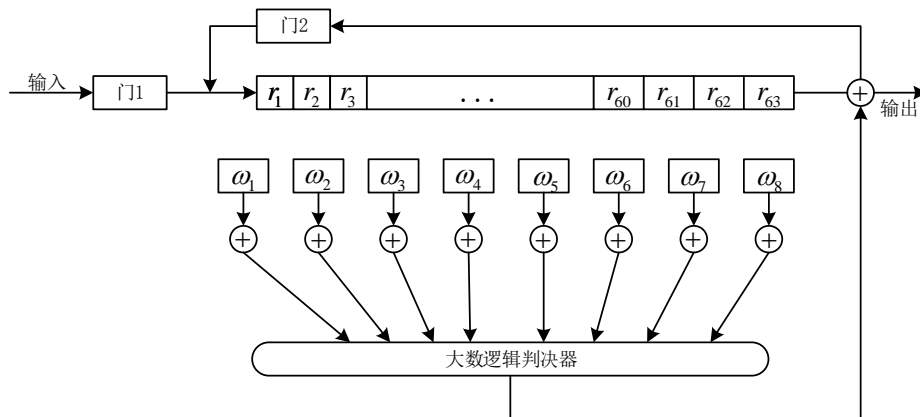


图 3-5 (63,37) 码译码电路

## 3.2 编码器设计

基于前述内容，这里对(63,37)码进行编码设计。该编码模块主要完成对数据的编码加固，使其具有一定的抗错误能力，其框图结构如图 3-6 所示。

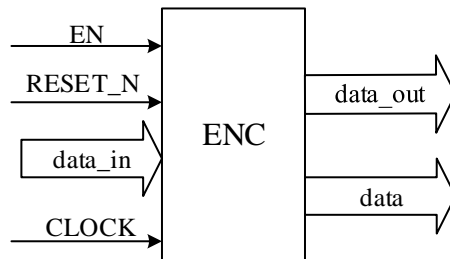


图 3-6 编码器结构

其信号定义如表 3-3 所示。从图 3-7 中可以看到，当  $EN=1'h1$  时，输入数据有效，在时钟上升沿将编码数据输出。根据码字  $C$ 、信息字  $i$  和生成矩阵  $G$  的关系： $C=i \cdot G$ ，其中生成矩阵如图 3-4 所示，可以看到当输入数据  $data\_in=32'h0ac0\_0230$  时，编码后的数据为  $data\_out=58'h3c6\_97be\_0ac0\_0230$  是正确的。

表 3-3 编码器信号定义

信号	定义
EN	数据有效信号
RESET_N	复位信号
CLOCK	时钟信号

表 3-3 （续表）

信号	定义
data_in	输入数据
data_out	编码输出数据
data	输出数据

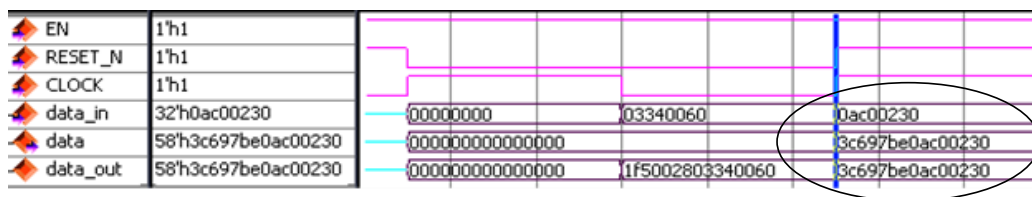


图 3-7 数据编码输出结果

### 3.3 译码器设计

译码模块主要对接收数据进行解码以判断数据是否有错误，若是在纠检错能力范围内的错误，对有错误的数据进行纠正，其结构框图如图 3-8 所示。

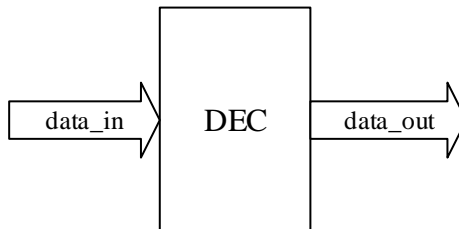


图 3-8 译码器结构

其信号定义如表 3-4 所示。

表 3-4 译码器信号定义

信号	定义
data_in	输入数据
data_out	输出数据

#### 3.3.1 校验函数

根据校正子( $\omega$ )、接收向量( $r$ )和奇偶校验矩阵( $H$ )之间的关系： $\omega = r \cdot H^T$ 。得到的(63,37)码的校正子如表 3-2 所示。图 3-9 所示校验函数的功能是：完成对

每一个数据位的校验，循环 $n$ 次，这里 $n=63$ 。其差错纠正过程如图 3-10 所示。

```

/*****
    循环移位 函数
*****/
function Cyc_shift;
input [1:63] data;
reg w_1,w_2,w_3,w_4,w_5,w_6,w_7,w_8;
reg [3:0] sum;
reg mj_out;
begin
    w_1 = data[6] +data[30]+data[40]+data[41]+data[44]+data[56]+data[61]+data[63];
    w_2 = data[24]+data[34]+data[35]+data[38]+data[50]+data[55]+data[57]+data[63];
    w_3 = data[2] +data[8] +data[32]+data[42]+data[43]+data[46]+data[58]+data[63];
    w_4 = data[5] +data[7] +data[13]+data[37]+data[47]+data[48]+data[51]+data[63];
    w_5 = data[3] +data[15]+data[20]+data[22]+data[28]+data[52]+data[62]+data[63];
    w_6 = data[10]+data[11]+data[14]+data[26]+data[31]+data[33]+data[39]+data[63];
    w_7 = data[12]+data[17]+data[19]+data[25]+data[49]+data[59]+data[60]+data[63];
    w_8 = data[1] +data[4] +data[16]+data[21]+data[23]+data[29]+data[53]+data[63];

    sum = w_1 + w_2 + w_3 + w_4 + w_5 + w_6 + w_7 + w_8;
    mj_out = (sum >= 'd5) ? 1 : 0;
    Cyc_shift = ^{data[63],mj_out};
end
endfunction
    
```

图 3-9 校验函数

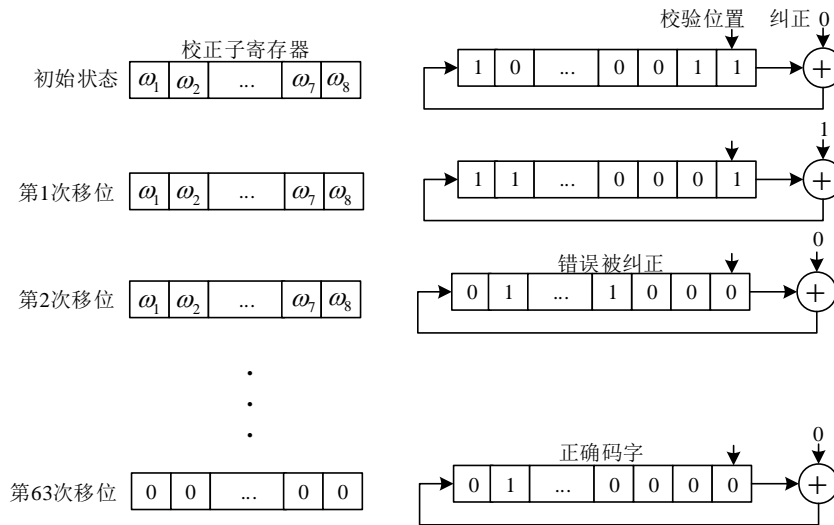


图 3-10 (63,37) 码的差错纠正过程示意图

### 3.3.2 译码输出

图 3-11 表示循环校验数据最后完成校验时的寄存数据，从图中读出此时的数据为： $\text{data\_temp\_0} = 63'h3ea0\_0500\_6680\_0c00$ 。

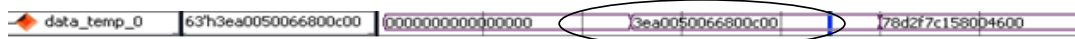


图 3-11 寄存变量值

根据(63,37)码的生成矩阵，32 位数据的编码后数据为 58 位，图 3-12 表示在校验过程中的多余五位的补零信号。



图 3-12 冗余位补零信号

图 3-13 表示译码的输入输出信号值，从图中看到输入输出数据一致。当输出为  $\text{data\_out} = 58'h1f5\_0028\_0334\_0060$  时，对应图 3-11 中  $\text{data\_temp\_0} = 63'h3ea0\_0500\_6680\_0c00$ 。去掉  $\text{data\_temp\_0}$  的高五位信号  $5'b00000$  后 58 位信号和输出信号  $\text{data\_out} = 58'h1f5\_0028\_0334\_0060$  结果一致。

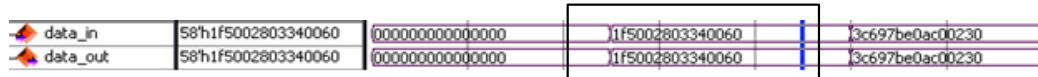


图 3-13 译码输入输出信号

### 3.4 一步大数逻辑译码加固 SRAM 及验证

#### 3.4.1 存储器模型

本文使用 Memory Compiler 生成了一个宽度为 58 位，深度为 1024 位的存储器模型。存储器模型框图如图 3-14 所示，信号定义如表 3-5 所示。

表 3-5 存储器信号定义

信号	位数	方向	定义
CLK	1	input	时钟信号
RETN	1	input	复位信号
WEN	1	input	读写控制信号
CEN	1	input	片选使能信号
A	10	input	地址线
D	58	input	写数据
Q	58	output	读数据

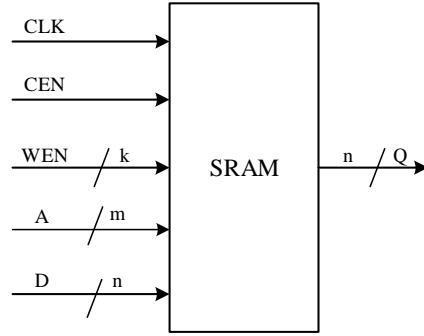
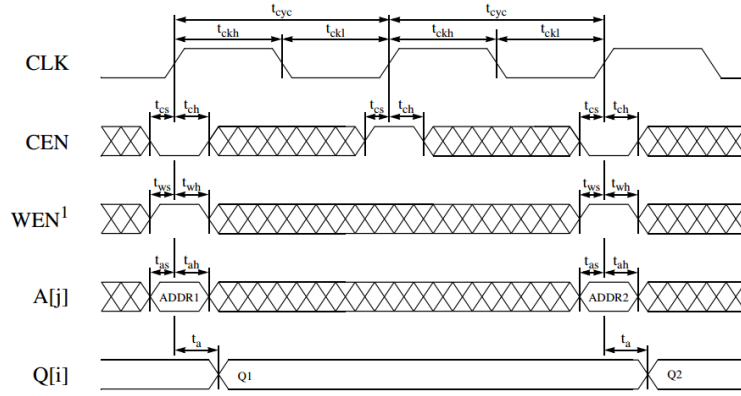
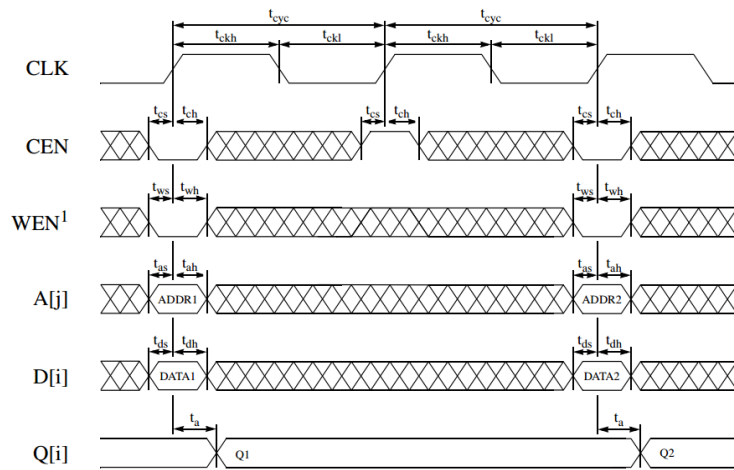


图 3-14 存储器模型框图

其中存储器的读写时序如图 3-15 所示。



a) 读时序



b) 写时序

图 3-15 存储器读写时序

存储器的读写由读写控制信号以及片选信号控制，如图 3-15 a) 所示，当片选有效 ( $CEN = 0$ )，读写控制信号为高电平 ( $WEN = 1$ ) 时，在时钟上升沿到来时，存储器根据地址信号  $A$ ，将相应的数据输出到  $Q$  端；如图 3-15 b) 所示，当片选有效 ( $CEN = 0$ )，读写控制信号为低电平 ( $WEN = 0$ ) 时，在时钟上升沿到来时，存储器根据地址信号  $A$ ，将数据输入端  $D$  的数据写到相应的存储位置。存储器的控制信号的操作说明如表 3-6 所示。

表 3-6 存储器控制信号与操作说明

RETN	CEN	WEN	操作
0	X	X	复位
1	1	X	片选无效，没有操作
1	0	0	写操作
1	0	1	读操作

### 3.4.2 存储器加固方案

本文应用一步大数逻辑译码对存储器进行加固，由前面章节设计的编码器和译码器来加固 SRAM，其设计方案如图 3-16 所示。ENC 为一步大数逻辑译码的编码电路，对存入存储器的数据进行编码加固；DEC 为译码电路，对存储器中读出的数据进行译码以得到原始数据，同时对数据进行纠错检错；RAM 为存储器建模模型。其端口信号定义如表 3-7 所示。

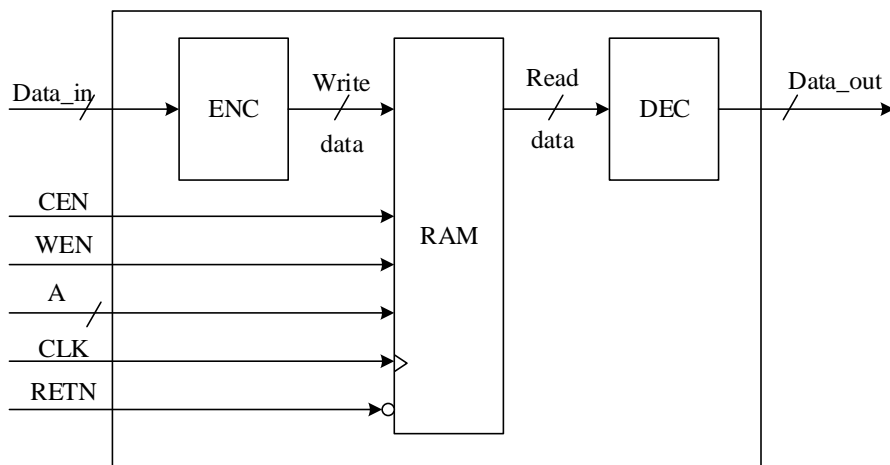


图 3-16 存储器加固设计框图

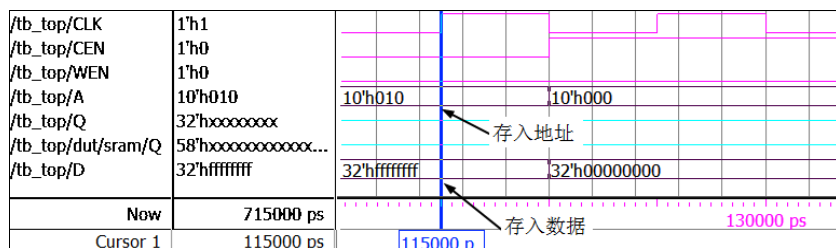
此设计方案中的输入数据  $Data\_in$  为 32 位数据，经过 (63,37) 码编码后进行存储，存入 RAM 中，最后当读出数据时，由译码电路进行译码，其纠错检错能力



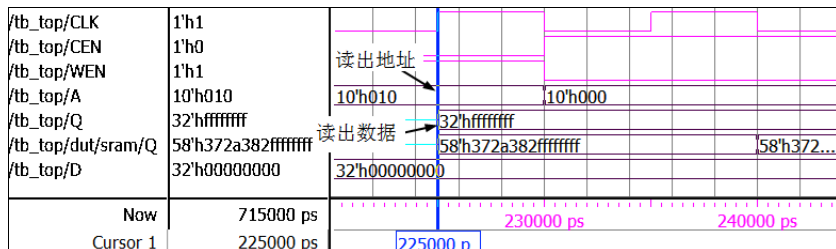
为 4 位。SRAM 加固之后的读写时序仿真图如图 3-17 所示。从仿真时序看，与图 3-15 比较，通过编码器和译码器加固的 SRAM 的读写时序与原来的读写时序没有发生变化，这是因为编码器和译码器都是组合逻辑电路，工作时不需要时钟沿触发并不占用时钟周期。

表 3-7 存储器信号端口定义

信号	定义
Data_in	输入数据
Data_out	输出数据
Write_data	编码后存储数据
Read_data	存储器读出数据



a) 写时序



b) 读时序

图 3-17 读写时序仿真结果

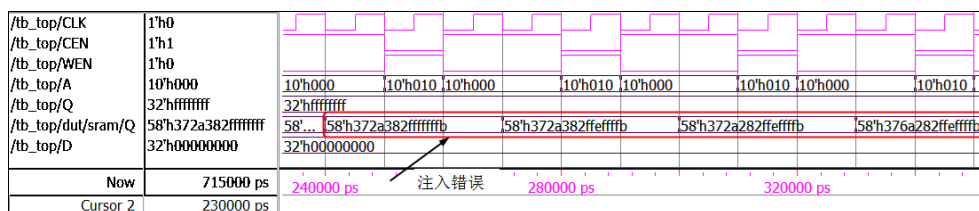
### 3.4.3 故障注入及存储器加固验证

由一步大数逻辑译码设计的编码器和译码器加固的 SRAM 能够实现对最多 4 位的错误纠错检错，本节将对加固的 SRAM 的纠错检错能力进行验证。

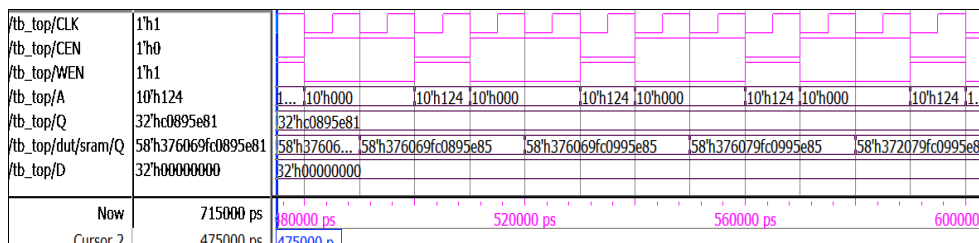
首先需要对设计的 SRAM 进行故障注入，故障注入模块用来模拟实际辐射环境中的单粒子翻转。Verilog 语言提供了 **force** 语句，能够对某一变量进行强制赋值而改变变量的原有值，与 **force** 语句相对应的 **release** 语句可以将被 **force** 语句强制

赋值的变量释放。因此，**force** 和 **release** 可以很好的满足模拟故障注入的需要。

图 3-18 给出了故障注入与纠错的仿真验证结果。如图 3-18 a) 所示，首先在地址  $A = 10'h010$  写入数据  $32'hffff\_ffff$ ，编码之后存入到存储器的值对应为  $58'h372\_a382\_ffff\_ffff$ ，然后分别在时刻 240ns 时刻注入一位错误（SRAM 的 Q 输出第 2 位翻转），在 270ns 时刻注入两位错误（SRAM 的 Q 输出第 2 位和第 20 位翻转），在 300ns 时刻注入三位错误（SRAM 的 Q 输出第 2 位，第 20 位和第 40 位翻转），在 330ns 时刻注入四位错误（SRAM 的 Q 输出第 2 位，第 20 位，第 40 位和第 50 位翻转）；从输出端 Q 看到输出结果一直保持为  $32'hffff\_ffff$ ，和输入数据一致。图 3-18 b) 中的地址和数据都是随机产生的，并对随机位置注入错误的仿真验证。验证结果显示，使用本文设计的一步大数逻辑译码加固的 SRAM 能够纠正多达四位的错误，而根据表 1-3 给出的单粒子翻转的错误统计，多位翻转数目集中在四位以下，加固的 SRAM 能够有效地预防单粒子翻转错误。



a) 地址 10'h10, 数据 32'hffff\_ffff



b) 地址 10'h124, 数据 32'hc089\_5e81

图 3-18 故障注入与纠错结果

### 3.5 本章小结

本章从错误修正码入手，选用一步大数逻辑译码，并对一步大数逻辑译码的编码和译码规则进行介绍。同时，根据研究结果，存储器发生错误翻转的位数主要集中在 4 位以下，所以选用 (63,37) 1 型 DTI 码作为本文的设计用码，其纠错检错能力  $t_{ML} = 4$ ，以满足单粒子翻转和存储器多位错误翻转的纠错检错能力。然后，本章对该码进行了编码器设计和译码器设计，并使用 Memory Compiler 生成

了一个宽度为 58 位，深度为 1024 位的存储器模型，并使用设计的编码器和译码器对其进行加固，然后对加固的存储器进行了读写时序仿真以及纠错能力验证。仿真结果表明，使用本文设计的一步大数逻辑译码加固的 **SRAM** 在不改变原有的读写时序的情况下能够实现多达四位的纠错能力，有效的提高了存储器件的抗辐射能力。下一章节在使用一步大数逻辑译码加固的 **SRAM** 的基础上，增加了对 **NBTI** 引起的错误的检测模块，使得 **SRAM** 不仅具有纠错能力，并且能够测试诊断 **NBTI** 错误。

## 第4章 检测 NBTI 错误的 SRAM 设计与验证

本文在前述章节介绍了编码理论，并使用一步大数逻辑译码对存储单元进行编码加固，由于器件尺寸的减小，NBTI 对存储器件的影响越来越显著。本章节在使用一步大数逻辑译码加固的 SRAM 的基础上，设计 NBTI 检测模块，使得 SRAM 能够检测和诊断 NBTI 引起的错误和软错误。同时为 SRAM 设计了 AHB 总线接口，使得增加了 NBTI 检测机制的 SRAM 能够在系统中集成使用。

### 4.1 测试诊断软错误和 NBTI 错误设计

在文献[55]和[56]中对 NBTI 效应的研究结果表明，NBTI 效应具有自愈合现象，应力消失后，器件放置一段时间对出现故障的存储单元不进行读写操作则存储单元会出现一定程度的自恢复，NBTI 效应对存储器的影响主要是引起存储器的读错误而不是写错误。

在现在的存储器件中，ECC 电路被广泛的用于检测和修正软错误，但是，对于 NBTI 引起的错误和软错误的不同之处在于：当存储单元出现数据错误时，如果该错误是软错误，则可以通过对存储单元写一个互补的逻辑值的方法，通过这种方法，存储单元的数据错误将被检测到并被修正；如果该错误是由于 NBTI 效应引起的存储单元读错误，则该存储单元不能通过写互补的逻辑值的方法进行修正。

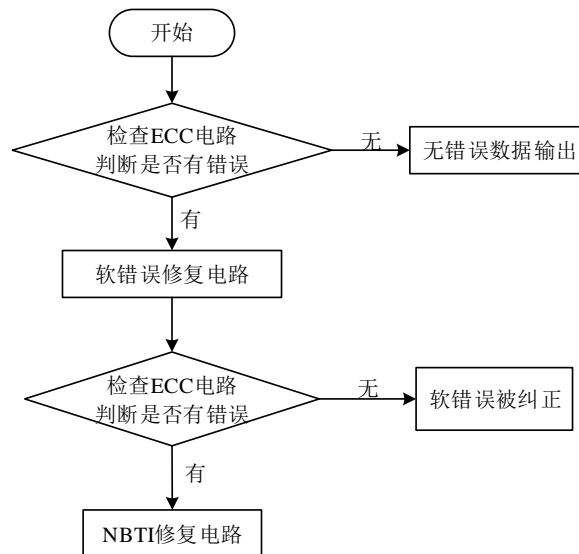


图 4-1 基于 ECC 电路检测 NBTI 引起的 SRAM 读错误和软错误流程图

对于 NBTI 引起的存储单元错误，本文提出一种基于 ECC 电路的写互补逻辑值的二次检测方法来探测 NBTI 引起的错误。该模块的思路流程图如图 4-1 所示。

首先，存储器在读出数据时检测 ECC 电路，若是没有发现错误，那么此存储单元存储的数据为正常值；若是发现存储单元数据有错误，则启用软错误修复电路对数据进行检测纠正，然后再进行第二次检测；在第二次检测中若是发现存储单元数据被修正，则说明第一次检测 ECC 电路探测到的是一个软错误，并且该错误被修正；若是在第二次检测中发现还是存在错误，则表明探测到的存储单元错误类型不是软错误，则可以告知系统，系统会采取相应的措施。

#### 4.1.1 具有 NBTI 检测功能的 SRAM 的设计

本文设计的具有 NBTI 检测功能的 SRAM 的设计结构框图如图 4-2 所示，端口信号说明在表 4-1 中。

表 4-1 NBTI 检测功能 SRAM 端口信号说明

信号	位数	方向	定义
CLK	1	input	时钟信号
RETN	1	input	复位信号
CEN	1	input	片选使能信号
WEN	1	input	读写控制信号
BNTI_EN	1	input	NBTI 检测使能
A	10	input	地址线
D	58	input	写数据
Q	58	output	读数据
NBTI	1	output	NBTI 检测结果输出
DATA_READY	1	output	高电平指示输出数据 Q 可以被读取

图 4-2 中的 ENC 和 DEC 分别为前述章节设计的一步大数逻辑可译 1 型(63,37) DTI 码的编码器和译码器，其纠检错能力为 4 位；SRAM 为存储器；NBTI CHECK 为检测 NBTI 引起的错误模块，其中信号 ctr1、ctr2 和 ctr3 分别控制编号为 1、2 和 3 的数据选择器；数据选择器 1 的作用是在进行回写数据时对输入编码器的数据进行选择；数据选择器 2 和 3 的作用分别是当数据进行回写时对存储器的片选使能信号和读写控制信号进行选择。

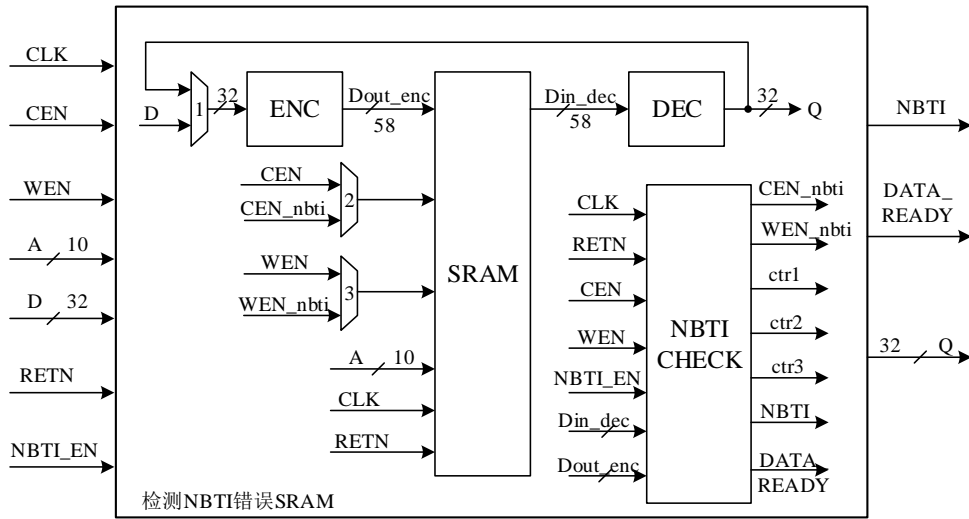
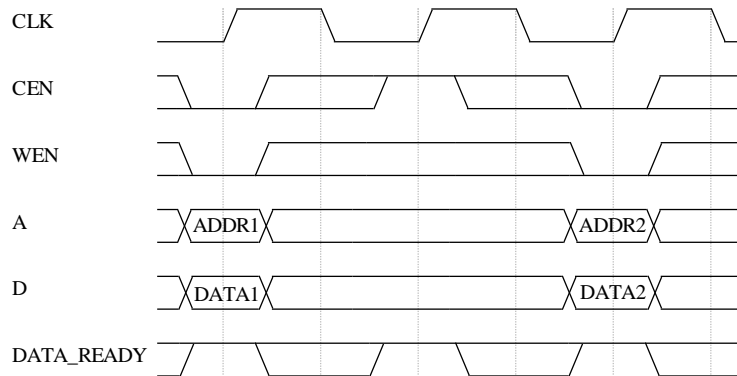
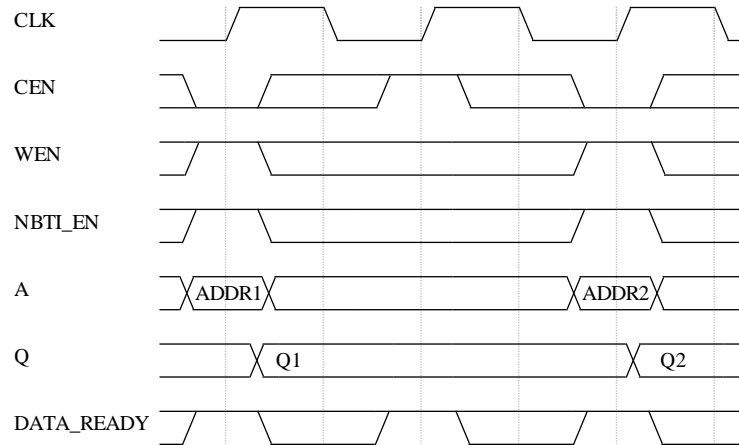


图 4-2 具有 NBTI 检测功能的 SRAM 的设计结构框图

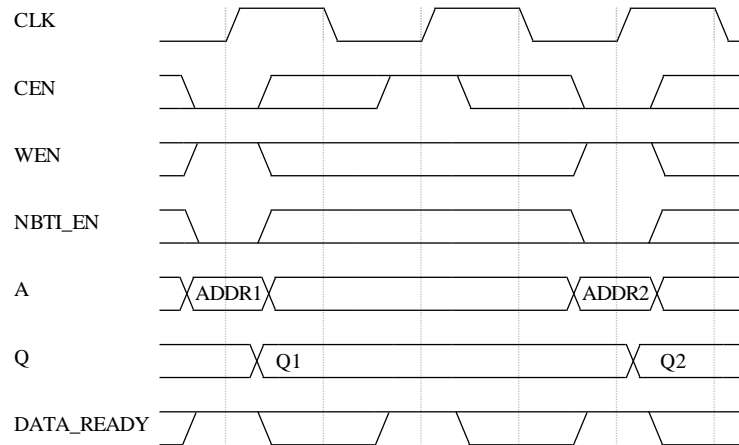
由于在检测 NBTI 错误的过程中存在数据回写至存储器的过程，整个数据的读写时序将发生变化。如图 4-3 所示，给出了本文设计的具有 NBTI 检测功能的 SRAM 读写控制时序。首先对于写时序，如图 4-3 a) 所示，由于写操作时不需要进行 NBTI 检测，因此写操作时序与原来 SRAM 的写操作时序相同。对于读操作存在三种可能：(1) 不启用 NBTI 检测 ( $NBTI\_EN = 1$ )，(2) 启用 NBTI 检测，未发现错误，(3) 启用 NBTI 检测，检测到错误。相应的操作时序分别如图 4-3 b) c) d) 所示，对于第 (1) (2) 种情况由于不用检测 NBTI 错误或者没有检测到错误，因此不需要对存储器进行回写判断，所以操作时序与原有的操作时序相比没有发生变化；而对于第 (3) 种情况，发现错误之后需要进行回写，然后再将数据读出，判断错误是否被回写修正，数据在进行回写和重读的过程中，DATA\_READY 信号会被拉低。



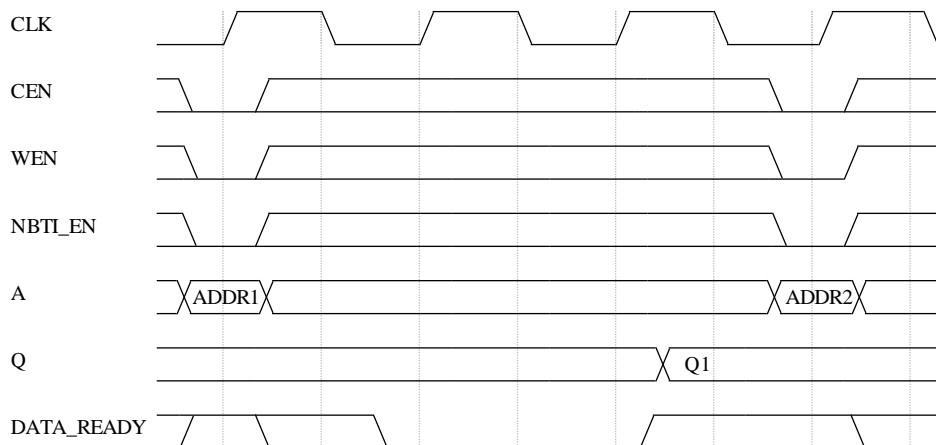
a) 写时序



b) 读时序 ( $NBTI\_EN = 1$ )



c) 读时序 ( $NBTI\_EN = 0$ , 未检测到错误)



d) 读时序 ( $NBTI\_EN = 0$ , 检测到错误)

图 4-3 具有 NBTI 检测功能的 SRAM 读写控制时序

### 4.1.2 仿真验证

由于 NBTI 的检测需要对存储器进行回写判断, 需要分别对 NBTI 错误和软错误进行故障注入的建模。对于 NBTI 错误的故障注入使用 3.4.3 节提到的 **force** 语句进行建模, 因为 **force** 语句可以强制改变变量的值, 而不会通过重写改变变量的值, 除非使用 **release** 语句将变量释放, 因此, **force** 语句能够很好的模拟 NBTI 错误的故障注入。对于软错误, 使用如图 4-4 所示的方案进行软错误故障注入建模, 故障注入使用 **fault\_en** 控制, 当 **fault\_en** 为 0 时, SRAM 数据输入端从编码器读取, 当 **fault\_en** 为 1 时, SRAM 数据输入端从 **fault\_i\_D** 读取, **fault\_i\_D** 数据宽度为 58 位, 与编码器输出相比是部分位翻转的, 实现软错误故障注入。

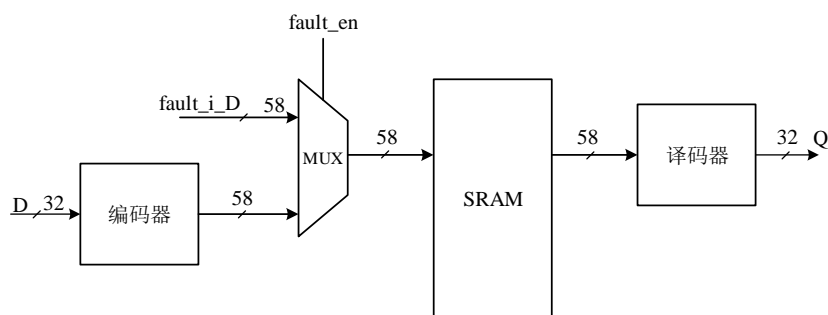


图 4-4 软错误故障注入原理框图

本文使用 ModelSim 对设计的具有 NBTI 检测功能的 SRAM 进行了功能仿真和验证, 仿真结果如图 4-5 至图 4-8 所示。其中, 图 4-5 为写操作仿真, 在 115ns 时向地址 10'h26d 写入数据 32'hfff\_fff, 数据经过编码后的 58 位数据为 58'h372\_a382\_fff\_fff, 这 58 位数据 58'h372\_a382\_fff\_fff 存入 SRAM 单元的 10'h26d 地址中; 图 4-6 为读操作仿真, 此时未注入故障, 图 4-6 a) 和 b) 分别为 NBTI 检测未使能和使能情况下的仿真结果; 图 4-7 显示了在 240ns, 对 SRAM 进行了故障注入, 此时注入的错误为 3 位, 故障注入后, 存储在 SRAM 中的值变成了 58'h372\_a282\_ffef\_fffb; 图 4-8 为故障注入之后的读操作, 图 4-8 a) 和 b) 分别显示了 NBTI 检测未使能和使能情况下的读操作仿真结果, 当 NBTI 检测未使能时, DATA\_READY 信号一直为高, 数据在第二个时钟周期就可以读取, 当 NBTI 检测使能时, DATA\_READY 信号会拉低两个周期, 在第四个时钟的上升沿时为高电平, 数据需要在第四个时钟周期进行读取, 并且输出信号 NBTI 的输出结果为 1, 因为检测到了不能回写修复的 NBTI 错误。图 4-9 所示为软错误注入以及 NBTI 检测, 在仿真时间 475ns 时刻注入软错误故障, 随后进行 NBTI 检测读出数据, 从仿真图形看, 首先检测到数据错误, 然后进行回写, 回写之后数据修正, NBTI 最后



输出为 0，并没有检测到 NBTI 错误，而是软错误。

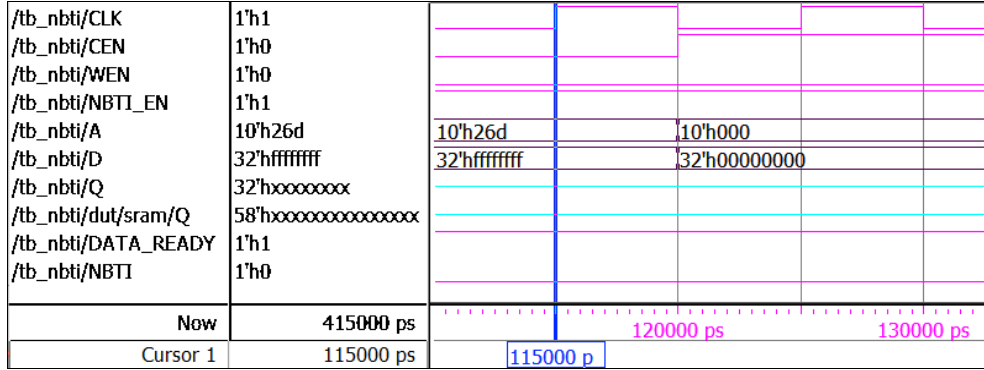
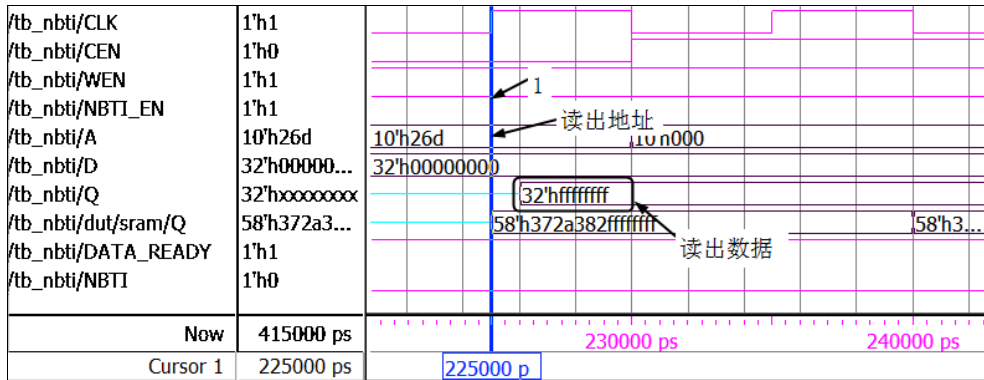
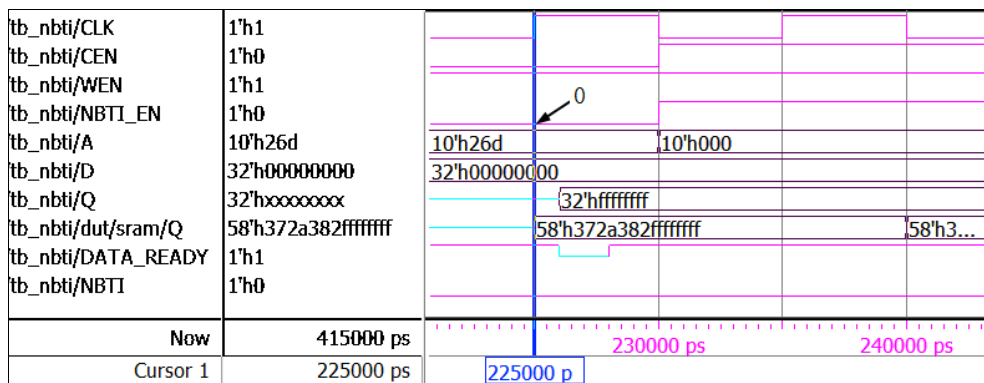


图 4-5 写操作仿真波形



a) NBTI 检测未使能 (NBTI\_EN = 1)



b) NBTI 检测使能 (NBTI\_EN = 0)

图 4-6 未注入故障的读操作仿真波形

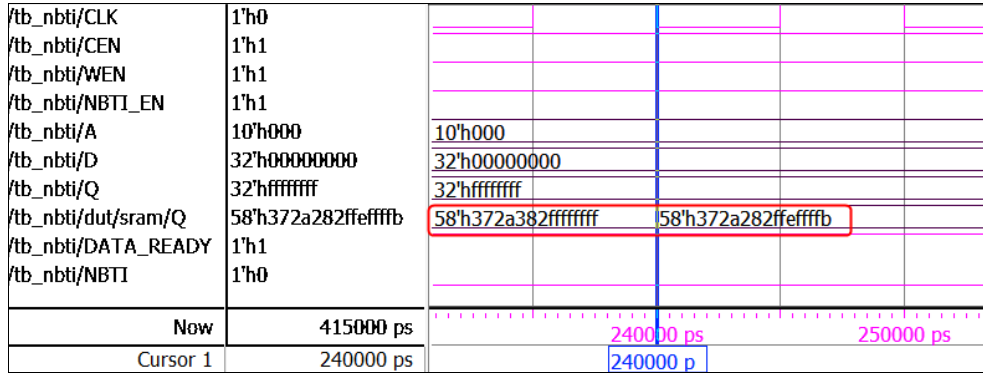
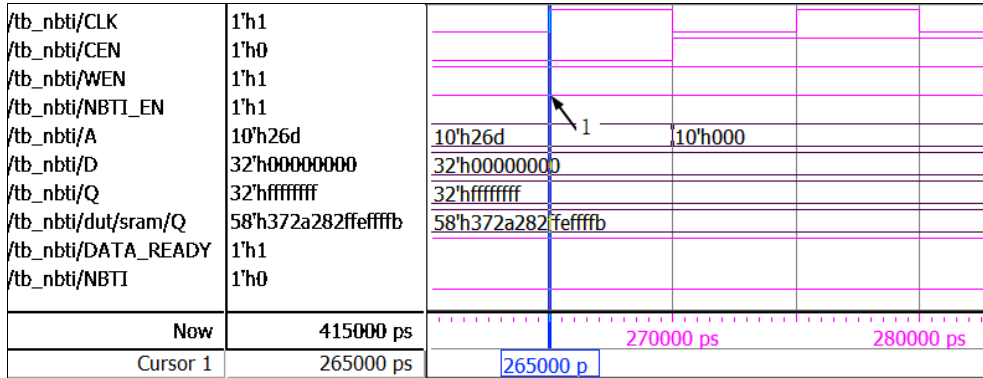
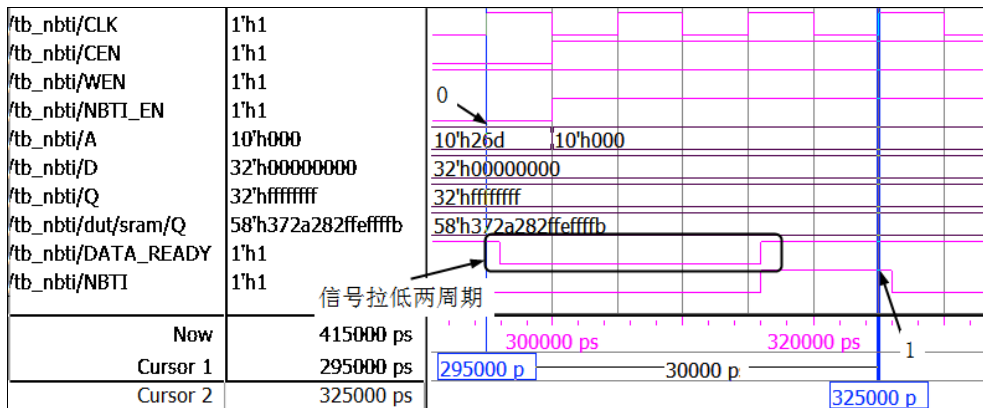


图 4-7 故障注入



a) NBTI 检测未使能 (NBTI\_EN = 1)



b) NBTI 检测使能 (NBTI\_EN = 0)

图 4-8 注入故障之后读操作时序

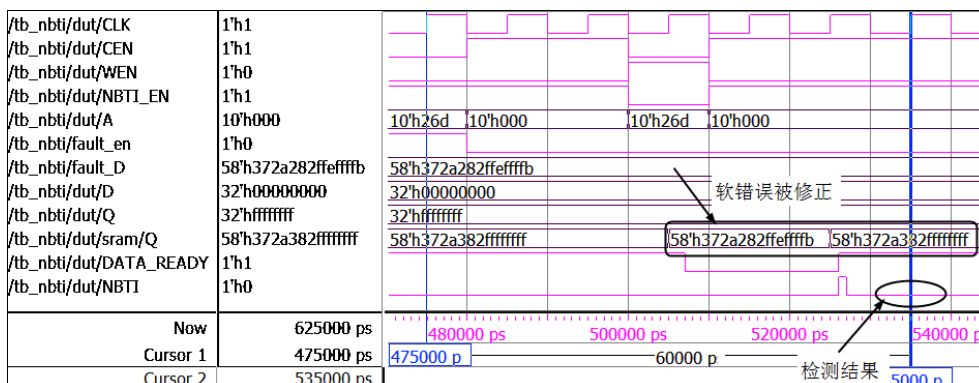


图 4-9 软错误注入与 NBTI 检测仿真波形

从仿真结果看，具有 NBTI 检测功能的 SRAM 不仅能够抵抗单粒子翻转以及多位错误翻转，又能够对 NBTI 错误作出诊断。设计时考虑到实际应用的情况，增加了 NBTI\_EN 控制端口，NBTI 检测功能可以通过 NBTI\_EN 端口进行启用或关闭，因为 NBTI 效应在器件使用的后期老化时出现的频率较高，而器件使用前期则很少出现。因此，在器件使用前期，可以关闭 NBTI 检测功能（NBTI\_EN = 1），在器件使用一段时间后可以选择开启 NBTI 检测功能（NBTI\_EN = 0），NBTI 检测功能开启后，可以通过 NBTI 输出端口判断器件是否出现 NBTI 错误而采取相应的解决措施。

## 4.2 SRAM 的 AHB 总线接口设计与系统级验证

SRAM 在实际系统应用中需要挂载至总线，与其它模块进行通信，并且 SRAM 通常挂载在高速总线上<sup>[57]</sup>。前一节介绍了检测 NBTI 错误的 SRAM 的设计与验证，从时序仿真结果看，与原有的 SRAM 时序相比存在差别。因此，本节首先为 SRAM 设计 AHB 总线接口，然后使用 AHB 总线的 BFM 对 SRAM 进行系统级的仿真验证，以验证本文设计的具有 NBTI 检测功能的 SRAM 的实用性。

### 4.2.1 AHB 总线接口设计

AHB 总线是一种片内高速总线，主要挂载高性能以及高时钟频率的功能模块，比如片内高速存储器、DMA、总线桥等<sup>[57]</sup>。AHB 总线传输过程分为地址周期和数据周期，两个周期分离可以实现流水传输，从而提高传输效率。AHB 总线支持多种传输方式，包括单个读写传输、不定长度的猝发读写传输以及固定长度的猝发读写传输等。

AHB 总线接口的设计需要满足总线规范的接口标准，如图 4-10 所示为 AHB

总线规定的从设备总线接口。

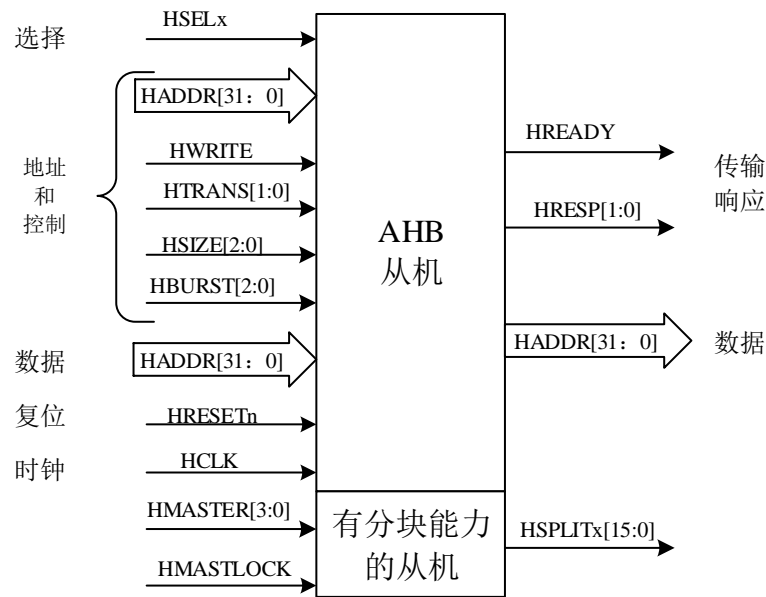


图 4-10 AHB 总线从设备标准接口

表 4-2 功能寄存器的描述

寄存器	位数	地址分配	读写权限	描述
NBTI_EN_REG	1	0x400	可读可写	NBTI 检测使能控制寄存器
				通过写操作，控制 NBTI 检测功能启用或关闭
				通过读操作，查看 NBTI 检测功能是否启用或关闭
NBTI_REG	可配置	0x404	可读可写	NBTI 检测结果状态寄存器
				记录检测到 NBTI 错误的次数
				通过读操作，查看检测到的 NBTI 错误次数
				通过写操作，清零或赋初值

SRAM 的 AHB 总线接口的设计框图如图 4-11 所示，其中接口模块一部分端口为 AHB 总线从设备的标准端口，挂接到总线与总线通信，另一部分端口与 SRAM 模块连接。一般接口模块中存在多个寄存器，用于控制不同的功能以及数据和地址的缓存等，如图 4-12 所示，每个的寄存器分配不同的地址，能够进行读写控制，

本文设计的接口模块包含两个特殊功能寄存器，寄存器的描述在表 4-2 中做了说明。本文使用的 SRAM 大小为 1K，分配的地址范围为 0x000~0x3ff。

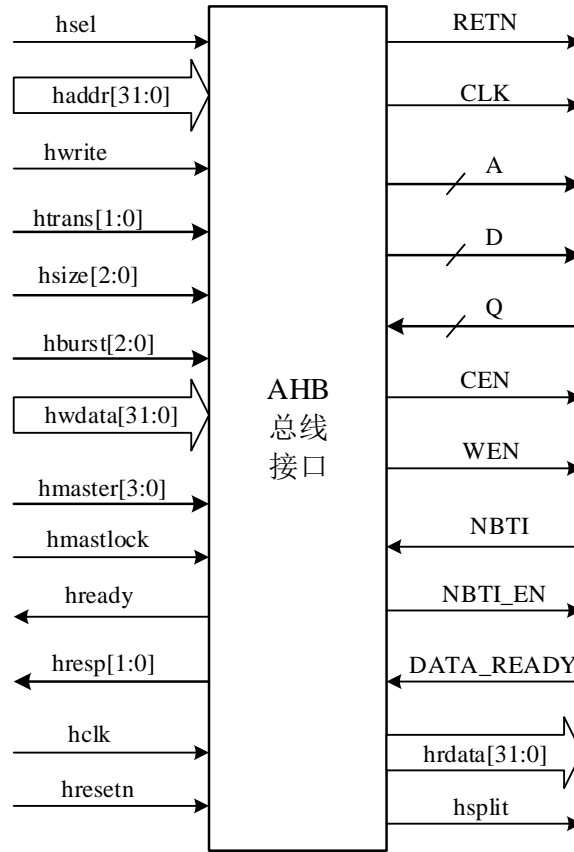


图 4-11 SRAM 的 AHB 总线接口的设计框图

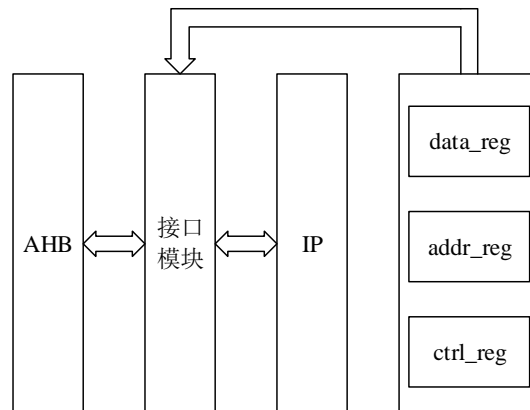


图 4-12 接口模块结构图

NBTI\_REG 状态寄存器的位数可以配置，这样可以根据实际应用按照需求设定位数。例如，系统设定 NBTI 错误累计出现 10 次时，就要采取相应的修复措施，那么 NBTI\_REG 寄存器的位数就可以设定为 4 位，最多能够记录 15 次 NBTI 错误。NBTI\_REG 状态寄存器的使用也可以按照实际需求，例如，可以将 NBTI\_REG 当做中断源，当 NBTI\_REG 寄存器记录次数超过设定的阈值时，启动中断程序，进行 NBTI 修复；或者系统通过读取 NBTI\_REG 寄存器的值，判断 NBTI\_REG 寄存器的值超过设定的阈值时进行 NBTI 修复操作。

#### 4.2.2 系统级验证

实验室基于 AMBA 总线的 VIP 模型，搭建了基于 AMBA 总线设计的 IP 核的验证环境，本文使用了该验证环境对设计的 SRAM 进行了系统级验证。基于 VIP 模型搭建的验证环境如图 4-13 所示。

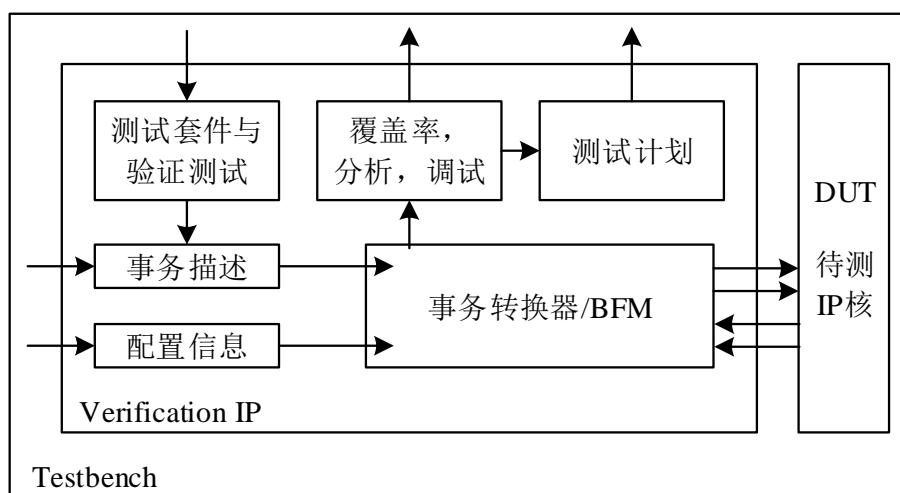


图 4-13 基于 VIP 模型搭建的验证环境

仿真结果如图 4-14 至图 4-18 所示。首先，如图 4-14 所示，向地址 0x10、0x14、0x18、0x1c 分别写入数据 32'h1111\_1111、32'h2222\_2222、32'h3333\_3333、32'h4444\_4444。此时，存储器中存储的数据为这 32 位数据经过编码器之后对应的 58 位数据。

如图 4-15 所示为从地址 0x10、0x14、0x18、0x1c 读出数据，与图 4-14 写入数据进行比较，在各个地址中都正确的读到了数据。

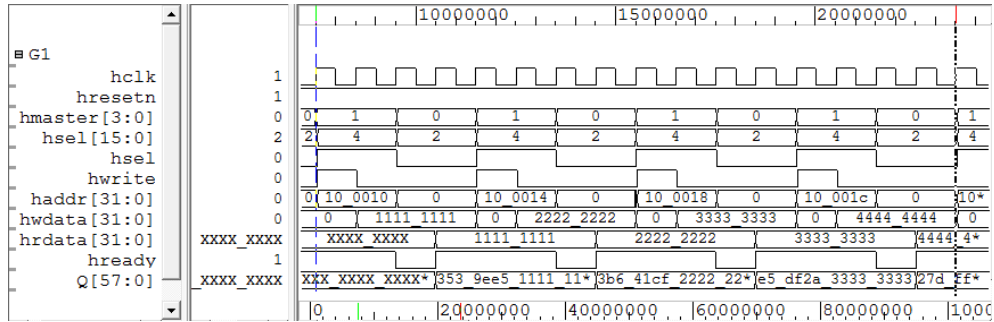


图 4-14 写数据

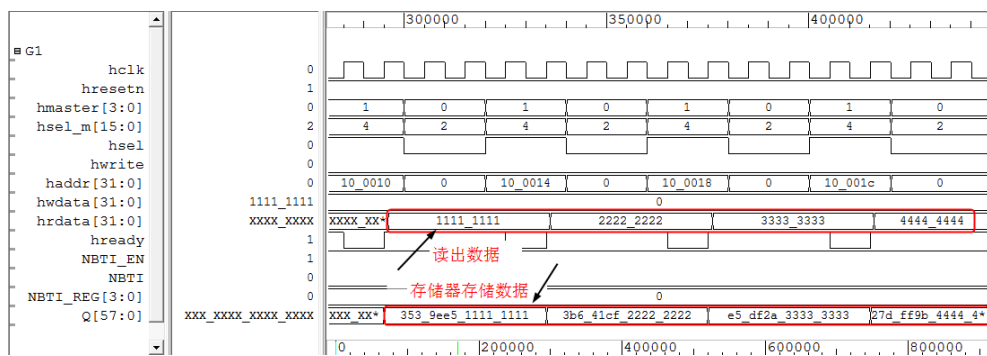
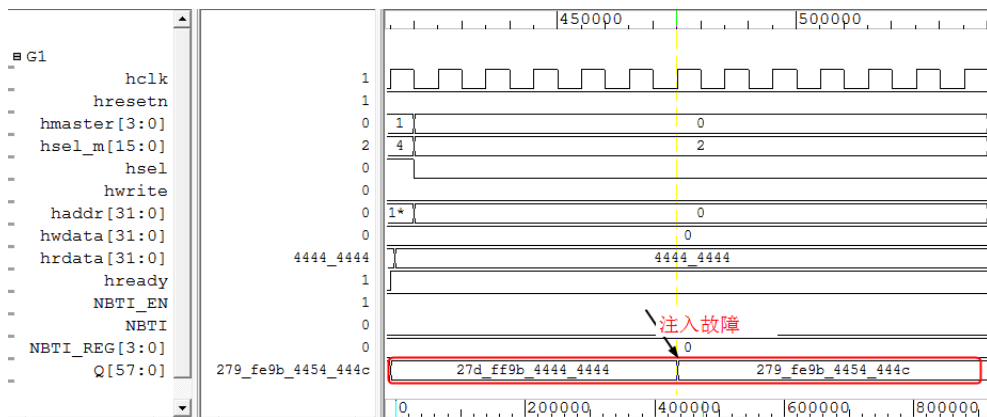
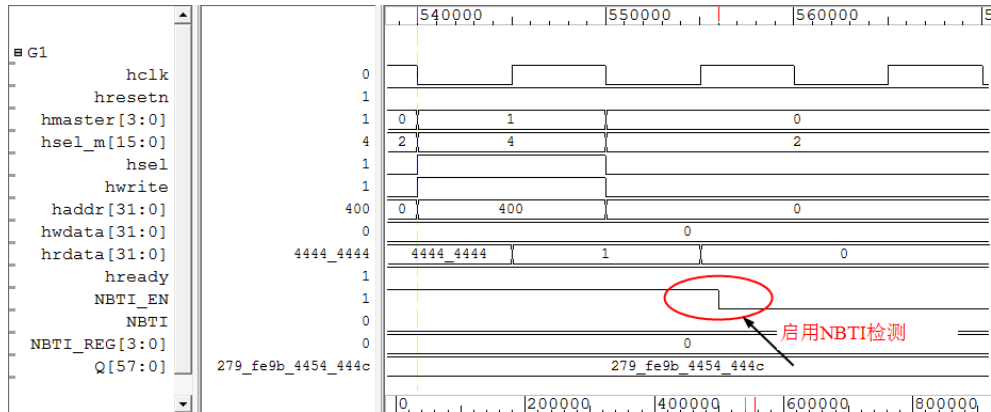


图 4-15 读数据

如图 4-16 a) 所示，为在某一时刻对存储器注入故障；然后，通过向寄存器 NBTI\_EN\_REG 写入 0，启动 NBTI 检测，图 4-16 b) 所示。



a) 注入故障



b) 启用 NBTI 检测

图 4-16 故障注入及检测

如图 4-17 所示为注入故障之后的数据读取结果，从图中可以看到存储器中存储的是故障注入后的数据，这些数据是经过检测回写的数据，由于注入的故障是 NBTI 错误，所以存储器中的数据没有被修正。NBTI 为高电平，表示检测到 NBTI 错误类型。读出数据是经过译码器译码后的数据，由于注入的错误个数为 4 位，在 (63,37) 码的纠错检测能力范围内，所以输出的结果和图 4-14 进行比较，可以看到结果是正确的。

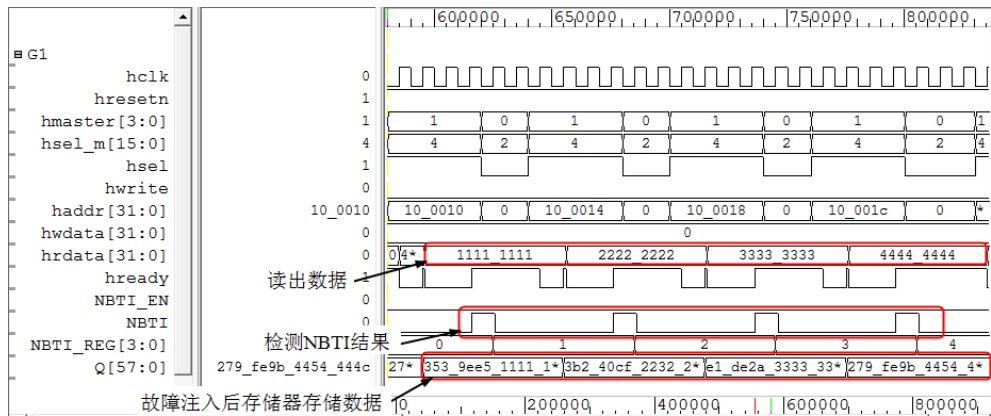


图 4-17 故障注入之后读数据

如图 4-18 所示，因为检测到了四次 NBTI 错误，因此从寄存器 NBTI\_REG 读取的值为 4。通过使用 VIP 模型搭建的验证环境，完成了本文设计的能够检测 NBTI 错误的 SRAM 的系统级验证



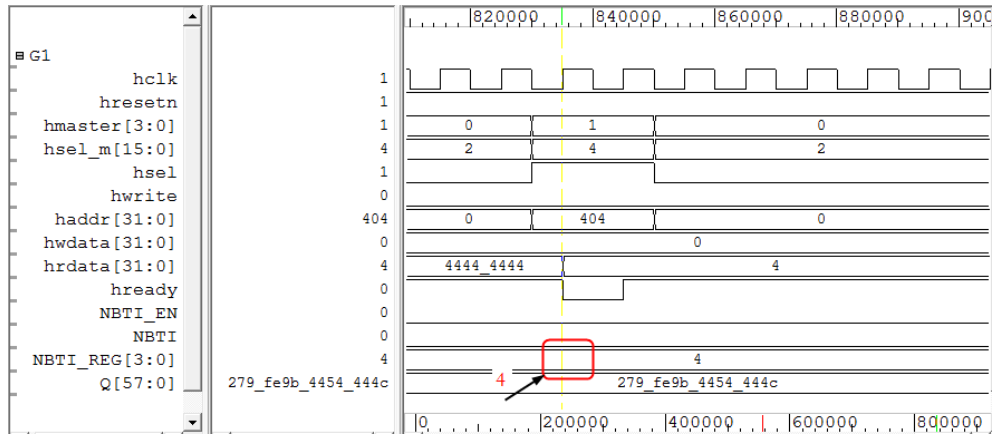


图 4-18 读 NBTI\_REG 值

### 4.3 本章小结

本章在对前述章节设计的一步大数逻辑可译 1 型 (63,37) 码加固的 SRAM 基础上, 提出了基于 ECC 电路的二次检测方法, 用来检测 NBTI 引起的错误和软错误。并设计了检测 NBTI 错误的模块, 验证结果表明设计的具有 NBTI 错误检测功能的 SRAM 能够正确的进行数据写入和读取。同时, 本章在 AHB 总线上对该设计进行系统验证。仿真验证结果表明, 本文设计的对 SRAM 的二次检测机制, 能够对 NBTI 引起的错误和软错误进行区分。

## 结 论

随着集成电路的发展，工艺尺寸的进步，存储器在空间辐射和核辐射等环境中发生单粒子翻转和多位错误翻转的概率逐渐增大。同时由于 NBTI 效应对短沟道器件的影响，在存储器元件中由于 NBTI 效应引起的老化错误也越来越得到重视。本文主要针对存储器进行系统级编码加固并验证以及对 NBTI 错误的检测。

主要工作包括以下几个方面：

1、通过对一次辐射事件中存储器发生错误的位数进行分析，采用一步大数逻辑译码，对存储器进行加固。选用纠错检错能力为 4 的 1 型 (63,37) DTI 码，根据其编译码规则，计算得到其生成矩阵和校验矩阵并得到相应的校正子。

2、对 (63,37) 码进行编码器和译码器设计，对存储单元的数据进行加固。同时，使用 Memory Compiler 建立了一个宽度为 58 位，深度为 1024 位的存储器模型。并对该存储器模型进行故障注入，来模拟实际辐射环境中对存储器的影响，并得到仿真结果。成功的实现将 32 位数据进行编码加固存储，在数据读出过程中随机注入错误，在 4 位及以下的错误模式下，设计的译码器能够检测并纠正。

3、在基于一部大数逻辑译码加固的 SRAM 的基础上，本文提出基于 ECC 电路的二次检测的方法，用来检测 NBTI 引起的错误和软错误的错误模式。并设计 AHB 总线接口，将设计的带有 NBTI 检测功能的 SRAM 在系统中进行验证。仿真结果表明，本文设计的带有 NBTI 检测功能的 SRAM 能够检测 NBTI 引起的错误模式。

## 参考文献

- [1] 刘征,孙永节,李少青,梁斌. SRAM 单元单粒子翻转效应的电路模拟[J]. 半导体学报,2007,28(01):138-141.
- [2] 王忠明. SRAM 型 FPGA 的单粒子效应评估技术研究[D].北京:清华大学,2011: 1-22.
- [3] Wu W, Seifert N. MBU-Calcul: A compact model for Multi-Bit Upset (MBU) SER estimation[C]. Monterey : Reliability Physics Symposium. IEEE, 2015: SE.2.1-SE.2.6.
- [4] Raine M, Hubert G, Gaillardin M, et al. Monte Carlo Prediction of Heavy Ion Induced MBU Sensitivity for SOI SRAMs Using Radial Ionization Profile[J]. Nuclear Science IEEE Transactions on, 2011, 58(6):2607-2613.
- [5] Li T, Yang H, Cai G, et al. A CMOS Triple Inter-Locked Latch for SEU Insensitivity Design[J]. IEEE Transactions on Nuclear Science, 2014, 61(6):3265-3273.
- [6] Pinsky L, Hoang S M, Idarraga-Munoz J, et al. Summary of the first year of medipix-based space radiation monitors on the ISS[C]. Montana : IEEE Aerospace Conference. 2014:1-8.
- [7] Rajsuman R. Design and Test of Large Embedded Memories: an Overview[J]. IEEE Design and Test of Computers, 2001,18(3):16-27.
- [8] 梁斌.数字集成电路中单粒子瞬态脉冲的产生与传播[D].长沙:国防科技大学, 2008 : 1-13.
- [9] Neuberger G, Lima F G d, Kastensmidt D, et al. An Automatic Technique for Op-timizing Reed-Solomon Codes to Improve Fault Tolerance in Memories[J]. IEEE Design & Test of Computers, 2005, 22(1): 50-58.

- [10] 刘必慰.集成电路单粒子效应建模与加固方法研究[D].长沙:国防科技大学, 2009.1-25.
- [11] E. Normand. Single Event Upset at Ground Level[J]. IEEE Transactions on Nuclear Science, 1996,43(6):2742-2750.
- [12] 李忠贺,刘红侠,郝跃. 超深亚微米 PMOS 器件的 NBTI 退化机理[J]. 物理学报,2006,55(02):820-824.
- [13] Pirich R, Weir J, Leyble D, et al. Effect of radiation on the molecular and contamination properties of silicone-based coatings[C]. Farmingdale : Systems, Applications and Technology Conference (LISAT), 2011 IEEE Long Island. IEEE, 2011:1-5.
- [14] Garg R, Nagpal C, Khatri S P. A fast, analytical estimator for the SEU-induced pulse width in combinational designs[C]. Anaheim : Design Automation Conference. ACM, 2008:918-923.
- [15] Juliano Benfica, Bruno Green, Bruno C. Porcher, et al. Analysis of SRAM-Based FPGA SEU Sensitivity to Combined EMI and TID-Imprinted Effects[J]. IEEE Transactions on Nuclear Science, 2016, 63(2):1294-1300.
- [16] Kuznetsov N V, Nikolaeva N I, Nymmik R A, et al. Comparison of the Models of Charged Particle Fluxes in Space[C]. Moscow: European Conference on Radiation and ITS Effects on Components and Systems. IEEE, 2015:1-4.
- [17] Rajsuman R. Design and Test of Large Embedded Memories: an Overview[J]. IEEE Design and Test of Computers, 2001,18(3):16-27.
- [18] D. Bossen, J. Tendler, K. Reick. Power4 System Design for High Reliability[J]. IEEE Micro, 2002,22(2):16–24.
- [19] Gaspard N J, Jagannathan S, Diggins Z J, et al. Technology Scaling Comparison of Flip-Flop Heavy-Ion Single-Event Upset Cross Sections[J]. IEEE Transactions on Nuclear Science, 2013, 60(6):4368-4373.

- [20] 李家强. 基于正交拉丁码的存储器抗多位翻转设计[D].哈尔滨:哈尔滨工业大学,2014:1-45.
- [21] P. Roche, G. Gasiot, K. Forbes, V. O'Sullivan. Comparisons of Soft Error Rate for SRAMs in Commercial SOI and Bulk Below the 130-nm Technology Node[J]. IEEE Transaction on Nuclear Science, 2003,50(6):2046-2054.
- [22] Alia R G, Biskup B, Brugger M, et al. SEU Measurements and Simulations in a Mixed Field Environment[J]. IEEE Transactions on Nuclear Science, 2013, 60(4):2469-2476.
- [23] Heather Quinn, Paul Graham. Radiation-Induced Multi-Bit Upsets in SRAM-Based FPGAs[J]. IEEE Transactions on Nuclear Science, 2005,52(6): 2455-2461.
- [24] 张月. PMOSFET 器件 NBTI 效应的机制与模型研究[D].西安:西安电子科技大学,2014:1-36.
- [25] 邱金娟,徐宏杰,潘雄,朱明达. SRAM 型 FPGA 单粒子翻转测试及加固技术研究 [J]. 电光与控制,2011,18(08):84-88.
- [26] 董巍. 面向片上存储应用的高性能抗辐射纠错码编码机制的研究与实现[D]. 上海:上海交通大学,2009:1-17.
- [27] 曹艳荣. 微纳米 PMOS 器件的 NBTI 效应研究[D].西安:西安电子科技大学,2009:1-23.
- [28] Chenouf A, Djeddar B, Benadelmoumene A, et al. Reliability analysis of CMOS inverter subjected to AC & DC NBTI stresses[C]. Algiers :International Design & Test Symposium. 2014:142-146.
- [29] Ghane M, Zarandi H R. Gate Merging: An NBTI Mitigation Method to Eliminate Critical Internal Nodes in Digital Circuits[C]. Heraklion :Euromicro International Conference on Parallel, Distributed, and Network-Based Processing. IEEE, 2016:786-791.

- [30] Arai K, Mardiyanto R. Eye Based HCI with Moving Keyboard for Reducing Fatigue Effects[C]. Las Vegas, NV : Eighth International Conference on Information Technology: New Generations. IEEE, 2011:417-422.
- [31] Lee Y, Kim T. A fine-grained technique of NBTI-aware voltage scaling and body biasing for standard cell based designs[C]. Yokohama : Design Automation Conference (ASP-DAC), 2011 16th Asia and South Pacific. IEEE, 2011:603-608.
- [32] 韩郑生,赵元富.抗辐射集成电路概论[M].北京:清华大学出版社,2011:1-152.
- [33] Jha N K, Ramgopal Rao V. A new oxide trap-assisted NBTI degradation model[J]. IEEE Electron Device Letters, 2005, 26(9):687-689.
- [34] Rofeh J, Sodhi A, Payvand M, et al. Vertical integration of memristors onto foundry CMOS dies using wafer-scale integration[C]. San Diego, CA : Electronic Components and Technology Conference. IEEE, 2015:957-962.
- [35] T. Grassler, B. Kaczer, W. Goes, et al. The Paradigm Shift in Understanding the Bias Temperature Instability from Reaction–Diffusion to Switching Oxide Traps[J]. IEEE Transactions on Electron Devices. 2011, 58(11): 3652-3666.
- [36] 郭靖. SRAM 存储器抗单粒子翻转加固设计技术研究[D].哈尔滨:哈尔滨工业大学,2015:1-56.
- [37] Rangan S, Mielke N, Yeh E C C. Universal recovery behavior of negative bias temperature instability [PMOSFETs][C]. Washington, DC, USA : Electron Devices Meeting, 2003. IEDM '03 Technical Digest. IEEE International. 2003:14.3.1-14.3.4.
- [38] Reddy V, Krishnan A T, Marshall A, et al. Impact of negative bias temperature instability on digital circuit reliability[C]. Reliability Physics Symposium Proceedings, 2002. IEEE, 2002:31-38.
- [39] Thewes R, Brederlow R, Schlunder C, et al. Device reliability in analog CMOS applications[C]. Washington, DC, USA : Electron Devices Meeting, 1999. IEDM

Technical Digest. International. 1999:81-84.

- [40] Eishi Ibe, Hitoshi Taniguchi, Yasuo Yahagi, Ken-ichi Shimbo, and Tadanobu Toba. Impact of Scaling on Neutron-Induced Soft Error in SRAMs From a 250 nm to a 22nm Design Rule[J]. IEEE Transactions on Electron Devices, 2010,7(57):1527-1538.
- [41] Schwank J.R., Ferlet-Cavrois V., Shaneyfelt M.R. et al., Radiation effects in SOI technologies[J], IEEE transactions on nuclear science, 2003, 50(3): 522-538.
- [42] Pretet J., Monfray S., Cristoloveanu S. et al., Silicon-on-Nothing MOSFETs: performance short-channel effects and backgate coupling[J]. IEEE transactions on electron devices, 2004, 51(2): 240-245.
- [43] Black J D, Ball II D R, Robinson W H, et al. Characterizing SRAM Single Event Upset in Terms of Single and Multiple Node Charge Collection[J]. IEEE Transactions On Nuclear Science, 2008, 55(6): 2943-2947.
- [44] Olson B.D., Amusan O.A., Dasgupta S. et al., Analysis of parasitic PNP Bipolar transistor mitigation using well contacts in 130nm and 90nm CMOS technology[J]. IEEE transactions on nuclear science, 2007, 54(4): 894-897.
- [45] Calin T, Nicolaidis M. Upset Hardened Memory Design for Submicron CMOS Technology[J]. IEEE Transactions On Nuclear Science, 1996, 43(6): 2874-2878.
- [46] 黄晔. 同时针对 SEU/SET/MBU 的 MOS 集成电路抗辐射加固技术研究[D].上海:上海交通大学,2009:1-20.
- [47] Saiz-Adalid L J, Reviriego P, Gil P, et al. MCU Tolerance in SRAMs Through Low-Redundancy Triple Adjacent Error Correction[J]. IEEE Transactions on Very Large Scale Integration Systems, 2015, 23(10):1.
- [48] Strobel D J, Czajkowski D R. Radiation shielding of integrated circuits and multi-chip modules in ceramic and metal packages: WO, US5635754[P]. 1997.
- [49] 李致远. 半导体器件辐射效应及抗辐射加固 [J]. 现代电子技术, 2015, 38(10): 1-4.

术,2006,19:138-141.

- [50] Bota S A, Torrens G, Alorda B, et al. Cross-BIC Architecture for Single and Multiple SEU Detection Enhancement in SRAM Memories[C]. Corfu : 2010 IEEE 16th International On-Line Testing Symposium (IOLTS). Corfu: IEEE, 2010: 141-146.
- [51] Neuberger G, Lima Fd, CarroL,etal. A Multiple Bit Upset Tolerant SRAM Memory[J]. ACM Transactions on Design Automation of Electronic Systems, 2003, 8(4):577-590.
- [52] Reviriego P, Argyrides C, Maestro J A,etal. Improving Memory Reliability Against Soft by Using Block Parity[J]. IEEE Transactions On Nuclear Science,2011,58(3):981-986.
- [53] Argyrides C. Improved Decoding Algorithm for High Reliable Reed Muller Coding[C]. Hsin Chu, Taiwan : 2007 IEEE International SOC Conference. Hsin Chu: IEEE, 2007: 95-98.
- [54] Shu Lin,Daniel J.Costello,Jr. Error Control Coding[M].Bei Jing:China Machine Press,2007:1-221.
- [55] Liu B, Chen C H. Testing, diagnosis and repair methods for NBTI-induced SRAM faults[C]. Austin, TX : IEEE International Conference on Ic Design & Technology. IEEE, 2014:1-4.
- [56] Kunhyuk Kang, Keejong Kim, Islam, A.E, et al. Characterization and Estimation of Circuit Reliability Degradation under NBTI using On-Line IDDQ Measurement[J]. 2007, 107(6):358-363.
- [57] 李玲,王祖强. AHB 总线仲裁器的设计[J]. 电子工程师,2005,31(01):23-25.





# 哈尔滨工业大学学位论文原创性声明及使用授权说明

## 学位论文原创性声明

本人郑重声明：此处所提交的学位论文《基于 ECC 电路的软错误修复和测试诊断 NBTI 错误方法研究》，是本人在导师指导下，在哈尔滨工业大学攻读学位期间独立进行研究工作所取得的成果，且学位论文中除已标注引用文献的部分外不包含他人完成或已发表的研究成果。对本学位论文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。

作者签名： 闻昱 日期：2016 年 7 月 1 日

## 学位论文使用权限

学位论文是研究生在哈尔滨工业大学攻读学位期间完成的成果，知识产权归属哈尔滨工业大学。学位论文的使用权限如下：

(1) 学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文，并向国家图书馆报送学位论文；(2) 学校可以将学位论文部分或全部内容编入有关数据库进行检索和提供相应阅览服务；(3) 研究生毕业后发表与此学位论文研究成果相关的学术论文和其他成果时，应征得导师同意，且第一署名单位为哈尔滨工业大学。

保密论文在保密期内遵守有关保密规定，解密后适用于此使用权限规定。

本人知悉学位论文的使用权限，并将遵守有关规定。

作者签名： 闻昱 日期：2016 年 7 月 1 日  
导师签名： 曹玉坤 日期：2016 年 7 月 1 日

## 致 谢

时光荏苒，不知不觉两年的硕士学习生活就要结束了。在毕业设计即将完成之际，在此我衷心的感谢在我学习阶段对我提供指导和帮助的老师、同学和朋友们。他们在我学习遇到困难时给予我帮助和鼓励，为我能顺利完成学业提供了巨大的帮助。

首先，我要感谢我的导师肖立伊教授，肖老师严谨的治学态度和科学的工作方法给了我极大的帮助和影响。肖老师和蔼可亲，在学习上悉心指导我完成毕业设计工作，生活上也细心备至，给予我很大的关心和帮助。在这过程中锻炼了我独立的思考和学习能力，让我受益匪浅。

其次，我要感谢王进祥老师、来逢昌老师、王永生老师、付方发老师以及微电子中心的其他老师在我硕士学习期间对我课业的指导和帮助。他们认真的治学态度，精益求精的科研态度都是我学习的榜样。

然后，我还要感谢实验室李杰师兄和柳姗姗师姐，在我做毕业设计期间，在设计方案和思路上给我提出了宝贵的意见和耐心的指导。在我撰写论文期间，张荣升师兄、李林哲师兄和曹雪兵师兄对我提供了热情的帮助。同时还有王天琦师兄、齐春华师兄、还有我的同学：兰风宇、于雨情等，在我遇到困惑时，给予我及时的帮助与关心，实验室和谐、温馨、互助的氛围给了我很多感动，让我有信心有勇气面对今后更大的挑战。在此我向他们表达我的感激之情。

另外，我还要特别感谢我的家人，是他们养育和教导了我，他们的理解和支持使我能够在学校专心完成我的学业。

最后对参加论文评审的各位教授和老师表示衷心的感谢。