工学硕士学位论文

LDPC 码在 SRAM 加固中的应用研究

DESIGN LDPC CODES FOR MULTIPLE BIT UPSETS MITIGATION IN SRAM

张艳晶

哈尔滨工业大学

2010 年 7 月

国内图书分类号：TN432
国际图书分类号：621.3.049.774

工学硕士学位论文

LDPC 码在 SRAM 加固中的应用研究

博 士 研 究 生：张艳晶
导　　师：肖立伊教授
申 请 学 位：工学硕士
学　　科：微电子学与固体电子学
所 在 单 位：微电子科学与技术系
答 辩 日 期：2010 年 7 月 01 日
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 Doctoral Degree in Engineering

DESIGN LDPC CODES FOR MULTIPLE BIT UPSETS MITIGATION IN SRAM

Candidate: YanJing Zhang
Supervisor: Prof. Xiao Liyi
Academic Degree Applied for: Master of Engineering
Speciality: Microelectronics and Solid-State Electronics
Affiliation: Dep. of Microelectronics Science and Technology
Date of Oral Examination: July, 2010
University: Harbin Institute of Technology

摘 要

随着集成电路复杂度的提高和处理数据能力的增强，芯片中存储器的比重越来越高，特征尺寸的下降又使存储器组合电路中发生单粒子瞬态（Single Event Transient，SET）软错误与阵列中发生单粒子翻转(Soft Event Upset，SEU)软错误的几率大致相当，如今如何消除这两类软错误已成为 SRAM 加固技术新的研究热点。

本文通过研究欧式几何空间（Euclidean-geometry，EG）和低密度单奇偶校验码(Low-Density Parity-Check，LDPC)的结构特征，设计了一种可用于 SRAM 的加固方案（即 EG_LDPC 码），通过采用并行大数译码和反馈环结构，使编译码电路在加固存储阵列的同时，本身也具有了探测纠正自身组合电路 SET 错误的能力（即自加固能力）。

该码是一类大数可译码，在译码器构建时本文提出一种算法，将 EG_LDPC 码的大数译码步数限制在两步以内，提高了译码器的速度，进而提出了压缩的 EG_LDPC 码，使每一个 EG_LDPC 码在不改变纠检能力下信息位可以任意收缩，以获得拥有合适信息位的码字。在分析系统可靠性时提出了一种计算加固存储器平均无故障时间（MTTF）的方法，该方法意义明确，运算简单。

在验证各个模块功能正确后，通过存储器行为模型和错误注入仿真分别对纠二检四(31,16)EG_LDPC 和纠四检五(42,16)EG_LDPC 码的可靠性和性能进行了仿真验证和分析，并与汉明码（Hamming），矩阵校验码（Matrix）和里德—穆尔码（Reed-Muller）进行了比较，结果显示 EG_LDPC 码加固的存储器可靠性较高，(42,16)EG_LDPC 码的 MTTF 比汉明码高出 3.19 倍，且与其它 ECC 码相比，仅需要少量额外面积、功耗和延迟的开销。

关键词 EG_LDPC；自加固；软错误；并行大数译码；MTTF

Abstract

As integrate circuit is becoming complex and its ability of dealing with data is changing more and more power, The proportion of SRAM is also greater than that of before, the number of SET in combinational logic circuit is close to that of SEU. As a result, efficient eliminating soft error and circuit hardening techniques for SRAM are paid more and more attention.

Through studying the structures of Euclidean geometry and low density Parity check code, we make a design which is also called EG_LDPC, it is could be used to harden SRAM. This design is different form the common code. There is parallel Majority Logic decoder and feedback-ring in this design, because of them the code is not only can avoid the SEU in memory array, but also can eliminate the SET which is taking place in its encoder and decoder.

The EG_LDPC is a one or more step majority logic code, we propound a arithmetic which can limit the number of the majority logic decoder's steps to two, because of this the speed of decoder is upgraded greater. And we further introduce curtate EG_LDPC code whose information bites can be altered randomly but keeping the original ability of correcting errors. When analyzing the reliability of the system we also introduce a means to count the MTTF of the hardened system which is definitude and simple.

After the functions of all kinds of modules are validated, we also analyze performance and reliability of EG_LDPC code through error-inject experiment. We also make a contrast with other codes including Hamming code Matrix code and Reed-Muller code. The result indicate that EG_LDPC code can achieve higher reliability with a little more area delay and energy overhead than the other codes, the MTTF of EG_LDPC code is indeed 3.19 times than that of Hamming code.

Keywords EG_LDPC, Harden-self , Soft Error, Parallel Majority Decoder, MTTF

目　　录

摘　要 I
Abstract II

第 1 章 绪论 1
1.1 课题背景 1
1.2 课题研究的目的和意义 2
1.2.1 软错误发生机理 2
1.2.2 SRAM加固的现状 4
1.3 纠错编码的发展 6
1.4 本文主要研究内容及结构 7
第 2 章 LDPC码概述 8
2.1 纠错编码基本概念 8
2.1.1 线性分组码 8
2.1.2 循环码 9
2.2 LDPC码简介 10
2.2.1 LDPC码定义 10
2.2.2 LDPC泰纳图表示 11
2.3 LDPC码常用构造方案 12
2.3.1 Gallager LDPC码[36] 12
2.3.2 Mackay LDPC码[37,38] 13
2.4 本章小结 14
第 3 章 LDPC码加固SRAM 15
3.1 编码器设计 16
3.1.1 码生成多项式 16
3.1.2 编码方案 20
3.2 译码器设计 22
3.2.1 大数逻辑译码 23
3.2.2 译码方案 26
3.3 探测器设计 32
3.4 压缩型EG_LDPC码 35
3.5 本章小结 38

第 4 章 可靠性和性能分析 39
4.1 编译码器功能仿真 39
4.2 故障注入仿真验证 40
4.3 纠错性价比与检错性价比 43
4.4 平均失效时间与平均可检测时间 45
4.5 本章小结 48
结 论 49
参考文献 50
哈尔滨工业大学硕士学位论文原创性声明 54
哈尔滨工业大学硕士学位论文使用授权书 54
致 谢 55

第1章 绪论

1.1 课题背景

目前集成电路的制造工艺已经进入纳米级阶段，其性能获得了大幅提高。但随着特征尺寸的减少、电源电压的降低和设计复杂度的增加，集成电路对空间辐射环境和地面噪声环境的影响也越发敏感[1]。自 1975 年报道了第一则由辐射引起了太空船电子器件工作发生故障[2]，人们开始认识到辐射和噪声环境对集成电路的影响，可靠性问题也成为集成电路设计者必须考虑的问题之一，尤其是在军事电子系统和宇航电子系统中使用的集成电路芯片，抗辐射指标要求更为严格。

之后人们开始关注集成电路与粒子辐射的关系。1978 年 Intel 公司研究人员在该公司产品中首次观察到地面辐射环境下 α 粒子引发的集成电路工作故障[3]。1979 年 Ziegler 在首次阐述了地面环境宇宙射线引发集成电路工作故障（称为软错误）的机理[4]。1993 年在一个商用航天器的计算机上发现了中子导致的软错误[5]。1996 年一项关于计算机日志文件的调查发现一台拥有 156Gbit DRAM 的超级计算机每天可能发生几次软错误[6]。在 2007 版国际半导体技术路线图 ITRS 中，软错误被列为集成电路可靠性所面临的主要挑战之一[7]。而且，随着集成电路特征尺寸的下降，抑制软错误不只局限于航天等领域，也适用于一般的民用领域。

自从上世纪七十年代人们发现软错误开始，各国都相继投入了大量资源来研究消除软错误的方法。日本从上世纪八十年代初开始对微电子器件的抗辐射加固方案进行研究，由东芝公司实施，主要从事硅 MOS，硅双极及砷化镓技术的开发研究。美国波音（Boeing）、霍尼威尔（Honneywell）等芯片提供商也已推出成熟的抗软错误产品，还有欧空局在这方面也处于领先地位。在国内针对软错误的研究单位主要有北京卫星环境工程研究所、兰州空间技术物理研究所、北京空间飞行器总体设计部、中国空间技术研究院控制与推进系统事业部、中国科学院空间科学与应用研究中心等单位。因国内研究起步较晚，水平与国外也有很大的差距。

1.2 课题研究的目的和意义

在集成电路中，出现的工作异常通常是由硬错误和软错误引起的[8]。制造工艺过程中的出现的缺陷或强辐射引起的高温致使器件烧毁而引起的错误称为硬错误，这种错误是不可逆的[9]，表现为在某个门或者位置上数据反复的出现错误，而软错误(Soft Error)则是由于粒子辐射等原因造成的数据信息暂时发生随机翻转的现象，它们均会严重影响集成电路的功能。

1.2.1 软错误发生机理

人们生活的环境中充满了各种辐射，大部分辐射是来自太阳粒子射线，只有少量是我们身边各种放射性材料以及电磁辐射造成的，如微电子产品封装材料中通常含有少量的放射性元素等等。当然随着高度、纬度以及时间的不同辐射粒子的种类和数量会有一定的差别，高空以中子辐射为主，而低海拔区域则以α粒子为主，它们都会影响集成电路正常工作，一般情况下中子辐射的能量比要α粒子大，对微电子器件的影响也比较严重。

太阳辐射中包含很多种粒子射线，但能引起电路产生软错误的主要有两种：低能中子粒子和高能中子粒子。当半导体材料中存在放射性元素（如硼 10 元素）时，来自太空的低能中子粒子有可能与其相遇并发生核反应，释放出大量α粒子从而引起软错误影响电路工作。而过去主要在高海拔地区才能发现的高能中子粒子，随着电路特征尺寸的减小，现在即使在海平面也可以发现它的踪迹，并且每个高能中子产生α粒子的能力可以几倍于低能中子，因此会严重影响电路的可靠性[10,11]。引起电路发生软错误的α粒子还有第三个来源——电路内部产生的α粒子，主要是半导体材料以及封装材料，互连材料中的放射性杂质（如铅）在衰变过程中释放出的α粒子，并且只需少量放射性杂质就会对电路的可靠性造成很大影响。由于α射线粒子的电离效应，当粒子穿入硅衬底时，随着入射深度的增加粒子能量会逐渐损失，并且发生裂变在其运动路径上会产生大量的电子—空穴对，如图 1-1 所示，当电子—空穴对发生在器件灵敏区时（即 P-N 结空间电荷区和少数载流子扩散长度范围内的区域），电子会被收集到带正电的扩散区域，而空穴被排斥流入衬底。以 NMOS 晶体管为例，当处于关断状态时栅极输入为低电平，漏极为高电平，衬底处于低电平，漏极和衬底之间的电场会将空穴压向衬底，电子被吸往漏极，从而沿着入射通路产生一个由漏极流向衬底的电流脉冲，该脉冲会严重影响电路正常工作。

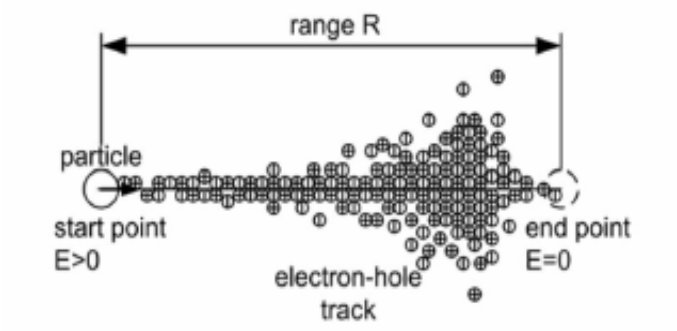


图 1-1 粒子裂变示意图

当电流脉冲发生在组合电路中时，由于其的暂态性不会对电路的稳态功能造成影响，经过短暂扰动之后，电路中各节点电压会恢复到正常值，这种现象称为单粒子瞬态效应（Single Event Transient,SET），但如果干扰脉冲在传输过程中最后为机械态元件所保存，同样有可能影响到电路正常工作。如果辐射粒子发生在存储器或者触发器等时序部件，或组合逻辑电路传播来的扰动脉冲被时序部件锁存，就会造成存储信息的失效，这称为单粒子翻转(Soft Event Upset，SEU)，如图1-2所示。为了降低SET和SEU的危害，通常可以采取下列两种途径：对电子部件进行加固防护或选用对单粒子不敏感的部件[12]。

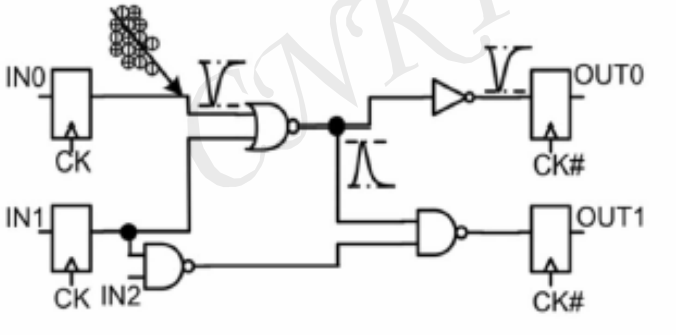


图 1-2 SET 对电路的影响

辐射环境对微电子器件影响还包括总剂量辐射效应和单粒子锁定效应，这两种效应均会对电路造成硬错误损伤。总剂量效应是指在电子器件的特性发生重大变化前，器件所能承受的总吸收能量级，超过这个能量级后器件就会出现永久故障，可采用抗辐射半导体工艺和辐射屏蔽来消除总剂量效应。单粒子锁定效应则是指当高能带电粒子穿过电路的PN结时，电离作用使其偏压，触通电路中的可控硅结构，并由于正反馈作用，在电源与地之间形成低电阻大电流的通路，当电流过大时会烧毁器件，造成永久失效。

1.2.2 SRAM 加固的现状

一般来说，随着特征尺寸的下降，集成电路 SEU 发生的几率应该有所增加，但当工艺尺寸发展到深亚微米领域后（<0.18um），集成电路中发生 SEU 的几率增加的并不明显，并且呈现出趋于饱和的趋势，这是由于特征尺寸不断缩小使电压降低趋于饱和，增加了相临节点之间短沟道效应的电荷共享，降低了节点电荷收集效率[13]。但由于在同一块晶元上可以放置更多的单元，当粒子以一定角度入射时可能与半导体晶元中的杂质发生反应，产生能量足够大的二级粒子，造成多位翻转（Multiple Bit Upsets, MBUs）[14]，MBUs 的宽度与粒子能量、入射角度、晶元搀杂类型和二级粒子的散射角度都有关系[15,16]。图 1-3 显示了在 0.18 微米随着粒子能量的不同引起 MBUs 的不同几率，可知进入深亚微米阶段后 MBUs 错误已不可忽视[17]。

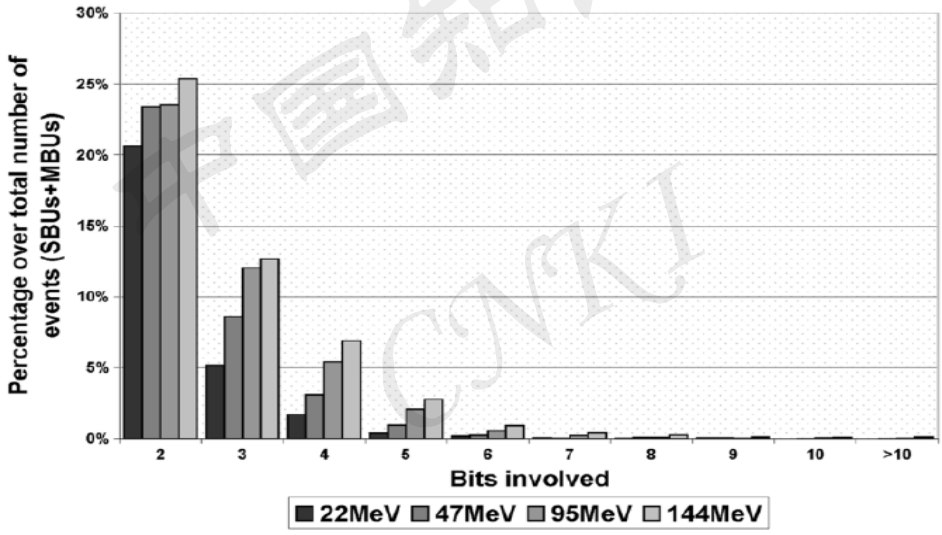


图 1-3 0.18um 工艺下粒子能量与多位翻转发生概率图

研究表明，多数超大规模集成电路中都包含了大量的存储器。在某些SOC系统中甚至超过60%的芯片面积[18]。硬错误和软错误均可引起半导体存储器工作出现故障[19]，但软错误发生的概率比硬错误发生的概率高5000倍以上[20]。因此，软错误已经是影响集成电路可靠性的主要因素[21,22]，为提高整个系统可靠性就必须采取有效途径来抑制存储器中的软错误。

为设计高可靠性的电路系统，目前也已有很多种方案，主要可分为以下三个级别：器件级、电路级和系统级。器件级加固主要是采用特殊工艺（如SOI、衬底重掺杂）或特殊材料（如高纯度封装材料）来减少电荷聚集或者漂

移效应，从而来减少软错误发生的几率。电路级则是采用特殊结构或冗余措施来设计基本单元或者是通过调整基本单元的参数来达到对系统的加固。对SRAM而言，器件级和电路级加固通常会给电路带来巨大的额外开销，因此常采用系统级加固方案，系统级方案又可以分为时间冗余、空间冗余和信息冗余，时间冗余即是对同一操作执行多次，对结果进行比较表决，显然不适合加固SRAM，空间冗余则是采用模块复制结果比较来判断，开销比较大也不适合。信息冗余则是通过检错码（Error Correcting Code，ECC）通过加入冗余位使码字本身具有了纠错检错能力，已发现的容错ECC码有很多种，并且纠检能力有所不同，可以由器件工作的辐射环境选择合适的ECC码。

从系统组成的角度来看，存储器受软错误影响的两个主要部分是存储阵列和组合电路，其中存储阵列受到ECCs等技术的保护，但由于电路规模急剧增大，工艺尺寸急剧下降，SET错误发生几率大大增加[23]，预计组合电路对系统软错误可靠性的影响从1992-2011年的阶段将上升9个数量级，到那时基本与未加任何保护的存储器持平。

已研究发现许多种ECC编码，但通常用来加固SRAM的有：汉明码（Hamming）、里德—穆尔（Reed_Muller）码以及矩阵校验（Matrix）码等。汉明码是一种线性分组码，可通过简单的查表机制来译码，经过适当的压缩可得到最小距离为4、纠单个差错并检测两个差错的码，因其码率高并且译码简单而得到广泛使用，但纠检错误的能力有限[24]。里德—穆尔码是另一类应用广泛的线性分组码，它的构造简单并且具有丰富的结构特性，能够纠正小于等于3的随机错误，但电路开销较大[25]。矩阵校验码能纠正两位随机错误，其码元由汉明和奇偶校验混合编码，但冗余位较高且译码复杂[26]。而为了消除组合电路中SET错误，通常采用三模冗余[27]或箝位二级管结构[28]等结构，这就需要更多面积、功耗和时序上的开销。

近来研究发现一类新的ECC码—欧式几何低密度单奇偶校验(Euclidean-geometry Low-Density Parity-Check，EG_LDPC)码，它是一种大数逻辑可乘的分组循环码，广泛应用于通信领域，其与别的ECC码不同之处在于其拥有自加固能力，即通过合理的设计容错电路结构，在加固存储器的同时，也可以消除自身编译码电路中的SET错误，文献[29]中将其用于字长非2的N次方的纳米存储器中验证了自加固能力。本文将其推广，应用于普通SRAM中，接下来章节

有详细探讨。

1.3 纠错编码的发展

伴随着信息技术的飞速发展，对各种设备可靠性的要求也不断提高。差错控制编码技术作为提高可靠性的简单而有效的手段，在数字存储和数字传输领域中显示出越来越重要的作用。且自纠错编码提出以来无论在理论还是在实际中都得到了飞速发展，其发展过程大致分为以下几个阶段：

20 世纪 40 年代，Hamming 提出了(7,4)Hamming 码，能纠正 7 个比特中所发生的单比特错误，之后 M.Golay 构造了(23,12)Golay 码，能纠正 3 个错误，极大的推动了编码理论的发展。

50 年代初，Reed 等人提出了 RM 码，与 Hamming 码和 Golay 码比起来在码字长度和纠错能力方面都有了很大的提高。该阶段主要研究各种有效的编译码方法，并且还提出了 BCH 码、RS 码等好码，它们可以同时纠正突发错误和随机错误。使纠错码从无到有得到了迅速的发展。

60 年代至 70 年代初，纠错编码得到迅速发展，不仅提出了各种有效的编码方法。而且讨论了与使用有关的各种问题，包括码的重量分布、译码错误概率和不可检错误概率计算等等，为纠错码的使用打下了坚实的基础。

70 年代初至 80 年代是纠错码的黄金发展期，使纠错码和具体的应用结合了起来。Goppa 码的构造引起了大量学者研究几何码的兴趣。这期间大规模集成电路的迅速发展也为纠错码的实用打下了坚实的物质基础，因而与使用相关的各种技术及有关问题得到了极大的关注，并取得了很多成果。

90 年代初，Berrou 等人提出了 Turbo 码[30,31]，它是一种信道编码理论界梦寐以求的可实用的好码，采用将卷积码和随机交织器相结合，并采用软输出迭代译码来逼近最大似然译码，由此得到了超乎寻常的优异功能。它的出现标志着信道编码理论研究进入了一个崭新的阶段，这在编码理论研究上具有里程碑的意义。但是 Turbo 码也有其缺点，译码复杂度较大，且在码长较长时，存在较大的时延并且还有 error-floor 效应。

Turbo 码的出现引发了对迭代译码算法研究的热潮，从而使人们重新发现：早在 1962 年 Gallager 提出的低密度单奇偶校验码（LDPC 码）[32,33]。

LDPC 码虽早就被提出，但在之后一段很长时间内没受到人们的重视，只有 Tanner 从图论角度研究过 LDPC 码。直到 Berrou 等提出 turbo 码之后，人们

才意识到 LDPC 码所具有的优越性和巨大的使用价值。目前，LDPC 码被认为是迄今为止性能最好的码，是当今信道编码领域的最令人瞩目的研究热点，近几年国际上对 LDPC 码的理论研究以及工程应用和 VLSI 实现方面的研究都已取得了重要进展。基于 LDPC 码的上述优异性能，其具有巨大的应用潜力[34]。

1.4 本文主要研究内容及结构

本文主要设计了可用于 SRAM 加固的欧式几何 LDPC 码加固方案，并提出一种算法，将多步欧式几何 LDPC 码的大数译码步数限制在两步以内，进而构造了压缩的 LDPC 码，最后提出了一种新型的被加固存储器平均失效时间（MTTF）的计算方案。本文的章节安排如下：

第一章绪论，介绍了该课题的背景以及研究的目的和意义，回顾了纠错编码的发展历史，并介绍了 LDPC 码的发展现状。

第二章主要介绍了有关 LDPC 码的基本概念和定理，以及 LDPC 码的多种构造方案。

第三章主要介绍了有自加固能力的 SRAM 容错结构，设计了欧式几何 LDPC 码的编码器、译码器和探测器的电路，并提出了压缩型欧式几何 LDPC 码的构造方案。

第四章通过存储器行为模型和错误注入仿真对采用欧式几何 LDPC 码加固的 SRAM 进行了可靠性和性能分析，并与其它加固方式进行了对比。

第2章 LDPC 码概述

2.1 纠错编码基本概念

LDPC 码作为一种 ECC 码，在实际中应用较多，但构造过程复杂，并需要较多的数学理论基础，本文接下来介绍有关 LDPC 码的基本概念和定理，方便大家理解和以后章节的引用。

2.1.1 线性分组码

在现在数字处理系统中，信源作为信息位输出端，输出一般为一系列二进制数字 0 和 1，这些二进制信息序列可被分成固定长度的消息分组，每个消息分组由 k 个信息位组成并记为u，因此共有2^k中不同的信息。编码器按照一定的规则可将输入的信息u转换为二进制 n 维向量v，且 n>k。此 n 维向量v就叫做消息u的码字，且消息u和码字v存在一一对应关系，因此，2^k种u对应有2^k中码字v，这2^k个码字的集合就叫一个分组码。

当 n 和 k 很大时，编码器则需在码库中存储2^k个长度为 n 的码字，这样编码器的复杂度将会非常高，但当分组码具有线性特征时则可大大降低编码复杂度。对于一个长度为 n，有2^k个码字的分组码，当且仅当其2^k个码字构成域 GF(2)上所有 n 维向量组成的向量空间的一个 k 维子空间时被称为线性(n,k)码。这时在其码字中可以找到 k 个线性独立的码字，g_0，g_1，…,g_{k-1}，使得 C 中的每个码字 v 都是这 k 个码字的一种线性组合，如式(2-1)所示：

$$v = u_0 g_0 + u_1 g_1 + ... + u_{k-1} g_{k-1} \tag{2-1}$$

式中，u_i=0 或 1，$0 \le i < k$。以这 k 个线性独立的码字为行向量，可得$k \times n$矩阵如式(2-2)所示：

$$G = \begin{bmatrix} g_0 \\ g_1 \\ \bullet \\ g_{k-1} \end{bmatrix} = \begin{bmatrix} g_{00} & g_{01} & g_{02} & \bullet & g_{0,n-1} \\ g_{10} & g_{11} & g_{12} & \bullet & g_{1,n-1} \\ \bullet & \bullet & \bullet & \bullet & \bullet \\ g_{k-1,0} & g_{k-1,1} & g_{k-1,2} & \bullet & g_{k-1,n-1} \end{bmatrix} \tag{2-2}$$

式中$g_i = (g_{i0}, g_{i1}, ..., g_{in-1})$，$0 \le i < k$。如果$u = (u_0, u_1, ..., u_{k-1})$是待编码的消息序列，则相应的码字应为$v = u \bullet G = u_0 g_0 + u_1 g_1 + \cdots + u_{k-1} g_{k-1}$。

显然 G 的行张成了(n,k)线性分组码 C，称矩阵 G 为 C 的生成矩阵。对于线性的分组码，存储器只需要存储 G 的 k 个行向量即可实现编码。

对于线性分组码，为便于译码一般都要求其具有系统形式，图 2-1 所示为系统形式的码字结构，可知其码字分为消息部分和冗余校验部分，消息部分由 k 个未经改变的原始信息位构成，冗余部分则是 n-k 个校验位，这些位均是消息位的线性和。

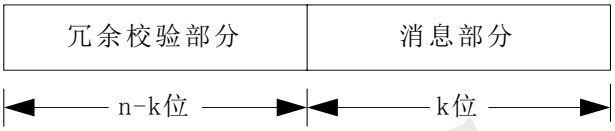


图 2-1 系统形式 ECC 码字结构

每一个(n,k)线性系统码均完全由 $k \times n$ 矩阵 G 确定，对任何一个 $k \times n$ 矩阵 G，均存在一个由 n-k 个线性独立的行向量组成的 $(n-k) \times n$ 矩阵 H，使 G 的行空间的任意向量与 H 的行向量正交，且任何与 H 的行正交的向量都在 G 的行空间中，称 H 为该码的奇偶校验矩阵。对于系统形式生成矩阵 $G=[P \quad I_K]$，奇偶校验矩阵 H 具有如式(2-3)所示形式，式中 P^T 是 P 的转置。

$$H=[I_{n-k} \quad P^T]=\begin{bmatrix} 1 & 0 & 0 & \bullet & 0 & p_{00} & p_{10} & \bullet & p_{k-1,0} \\ 0 & 1 & 0 & \bullet & 0 & p_{01} & p_{11} & \bullet & p_{k-1,1} \\ 0 & 0 & 1 & \bullet & 0 & p_{02} & p_{12} & \bullet & p_{k-1,2} \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ 0 & 0 & 0 & \bullet & 1 & p_{0,n-k-1} & p_{1,n-k-1} & \bullet & p_{k-1,n-k-1} \end{bmatrix} \tag{2-3}$$

2.1.2 循环码

循环码是线性分组码的一个重要子类，其包括有：BCH 码、里德—索罗门码、欧式几何码和射影几何码等等。这些码有两个主要的特点：一是通过带反馈连接的移位寄存器，该码的编码能很容易的实现；二是由于具有固有的代数结构，所以能找到多种实用的方法对该码进行译码。

如果对一个 n 维向量 $v=(v_0,v_1,...,v_{n-1})$ 的分量做一次向右的循环移位，可得到另一个 n 维向量 $v^{(1)}=(v_{n-1},v_0,...,v_{n-2})$，称上述操作为 v 的一次循环移位，式(2-4)是对 v 做 i 次向右循环移位所得的 n 维向量：

$$v^{(i)}=(v_{n-i},v_{n-i+1},...,v_{n-1},v_0,v_1,...,v_{n-i-1}) \tag{2-4}$$

对于一个(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。为了研究循环码的代数特性，码字 v 可写成式(2-5)所示的多项式形式：

$$v(x) = v_0 + v_1 X + v_2 X^2 + ... + v_{n-1} X^{n-1} \tag{2-5}$$

每个码字都对应一个次数等于或小于 n-1 的多项式。若 $v_{n-1} \neq 0$，则 v(x)的次数为 n-1，否则小于 n-1。码字 v 和多项式 v(x)之间一一对应，并称 v(x)为 v 的码多项式。

2.2 LDPC 码简介

LDPC 是一种线性分组码，并可分为循环 LDPC 码和准循环 LDPC 码，其拥有分组码的所有特征，纠检错误能力也由最小距离决定，但它有许多独特之处，本文简要介绍一下 LDPC 的特性。

2.2.1 LDPC 码定义

长度为 n 的线性分组 LDPC 码可以由生成矩阵 G 或校验矩阵 H 唯一确定。LDPC 码通过奇偶矩阵 H 的定义如下：

（1） 每一行含有 ρ 个 1；

（2） 每一列含有 γ 个 1；

（3） 任何两列之间位置相同的 1 的个数（以 λ 表示）不大于 1，即 $\lambda = 0$ 或者 $\lambda = 1$；

（4）与码长和 H 的行数相比，ρ 和 γ 都较小。

特性（1）和（2）表明奇偶校验矩阵 H 分别具有不变的行重 ρ 和列重 γ，特性（3）意味着在 H 中没有任何两行具有超过一个相同位置的 1。由于 ρ 和 γ 都小于码长和 H 中的行数，因此 H 中 1 的密度很小。正因为如此，H 被称为低密度单奇偶校验矩阵（low-density parity-check matrix），定义 H 的密度 r 为 H 中 1 的总数与 H 中的元素总数的比值，如式(2-6)（其中 n 为 H 的行数，J 为 H 的列数）：

$$r = \rho / n = \gamma / J \tag{2-6}$$

由上述定义给出的 LDPC 码被称为(γ,ρ)规则 LDPC，如果奇偶校验矩阵 H 的各行或者各列具有不同的重量，则称 LDPC 为非规则的，在本文中主要讨

论规则的 LDPC 码。

2.2.2 LDPC 泰纳图表示

线性码可以采用不同的方式来表示，最著名的图形化表示方法是网格表示，它使得采用基于网格的译码算法成为可能并且使译码的复杂度大大降低，线性分组码的另一种有用的图形化表示为泰纳图，泰纳图显示了码元与校验码元的奇偶校验和之间的关联关系[35]。

对于一个长度为 n 的线性分组码，设其奇偶校验矩阵 H 由 J 行组成，依次记为h_1， h_2，…,h_J。依此以构造一个图ς_T，包括两个顶点集γ_1和γ_2。第一个顶点集γ_1包括 n 个顶点，代表码的 n 个码元比特，记为v_0， v_1，…,v_{n-1}，被称为码元顶点。第二个顶点集γ_2包括 J 个顶点，代表 J 个奇偶校验和，记为s_1，s_2，…,s_J，这些顶点称为校验和顶点。当且仅当校验和s_j中包含码元比特v_l，码元顶点v_l与校验和顶点s_j通过一条边相连，可得任意两个码元顶点不相邻，同时任意两个校验和顶点也不相邻，因此图ς_T是一个二分图，并且码元顶点v_l的度等于包含v_l的校验和的个数，而校验和顶点s_j的度等于s_j所校验的码元比特数。式(2-7)为(6,2)LDPC 码的校验矩阵：

$$H=\begin{bmatrix}1&1&0&1&0&0\\0&0&1&1&1&0\\1&0&0&0&1&1\\0&1&1&0&0&1\end{bmatrix} \tag{2-7}$$

根据 H 可以画出其二分图表示形式，如图 2-3 所示。图中变量节点集合$(v_1,v_2.....v_6)$和校验节点集合$(c_1,c_2.....c_4)$内部不存在相连的边，但两类节点之间存在着连线。变量节点和校验节点之间存在连线意味着该变量比特参加了此校验式，也就是校验矩阵某一行中‘1’的位置。将每个节点上的连线数目称为该节点的度，图 2-2 中变量节点的度为 2，校验节点的度为 3。

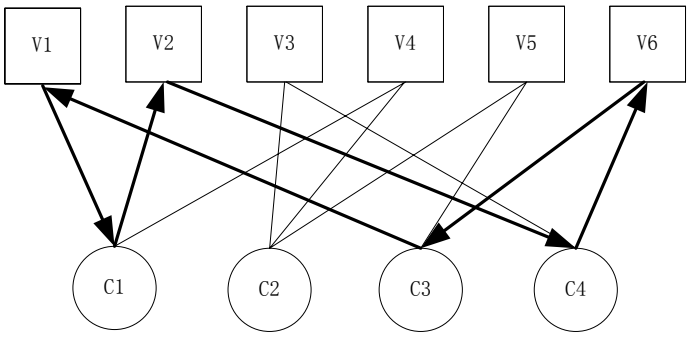


图 2-2 (6,2)LDPC 码的特纳图

图 2-2 中的(6,2)LDPC 是规则形式，图中所有码元顶点的度都相同且等于γ（奇偶校验矩阵的列重量），所有校验和顶点的度都相同且等于ρ（奇偶校验矩阵的行重量），这说明一个 LDPC 码的泰纳图ς_T中不包含长度为 4 的环，码的泰纳图中不包含长度很短的环这一点非常重要，因为短环限制了译码算法的性能。

2.3 LDPC 码常用构造方案

如何来构造 LDPC 码，目前已有多种方案，其中具有理论研究价值的主要包括：Gallager LDPC 算法，Mackay LDPC 算法以及林舒提出的组合数学构造方案等等，但它们各有利弊，本文接下来对各方案做一下简单介绍。

2.3.1 Gallager LDPC 码[36]

Gallager 给出了一种 LDPC 奇偶校验矩阵 H 的构造方法。对于给定的ρ和γ，令 k 为大于 1 的正整数，首先构造$k \times k\rho$的子矩阵$H_1, H_2, \ldots, H_\gamma$并组成矩阵 H。子矩阵的每列只有一个 1 且每行有ρ个 1。因此，每个子矩阵共有$k\rho$个 1。对于$1 \le i \le k$，H_1第i行的ρ个 1 都分布在$(i-1)\rho+1$到$i\rho$列中，其它子矩阵都仅仅是H_1的列置换，因此也称H_1为 H 的基矩阵，其形式如式(2-8)所示：

$$H_1 = \begin{bmatrix} 11\ldots1 & & & \\ & 11\ldots1 & & \\ & & * & \\ & & & * \\ & & & 11\ldots1 \end{bmatrix} \tag{2-8}$$

且H_1中每一行都有ρ个 1，共有 k 行，校验矩阵 H 如式(2-9)所示，π为

矩阵 H_1 的任一置换代替。

$$H=\left[H_1 \quad \pi_2(H_1) \quad * \quad * \quad \pi_\gamma(H_1)\right]^T \tag{2-9}$$

从 H 的构造过程可知，H 的子矩阵中任意两行对应列元素不会同时为 1，并且 H 的子矩阵中任意两列的对应行元素同时为 1 的次数至多为 1。由于 H 总的元素个数为 $k^2\rho\gamma$，1 元素的总数为 $k\rho\gamma$，，可知 H 的密度为 $1/k$。如果选择远大于 1 的 k，H 的密度就很小，是一个稀疏矩阵，H 是否具备 LDPC 定义中的条件(3)取决于对子矩阵 H_1 中 $\gamma-1$ 个列的置换选择。可惜 Gallager 并没有给出选择 H_1 列置换的方法、以构成其它子矩阵 $H_2,\ldots,H_\gamma$。因此，好的 LDPC 码，尤其是长码，需要用计算机搜索来寻找。式(2-10)是 $k=3$，$\rho=4$ 和 $\gamma=3$ 条件下利用 MATLAB 构造的校验矩阵，矩阵尺寸为 $k\gamma\times k\rho=9\times12$。

$$H=\begin{bmatrix}
1&1&1&1&0&0&0&0&0&0&0&0\\
0&0&0&0&1&1&1&1&0&0&0&0\\
0&0&0&0&0&0&0&0&1&1&1&1\\
0&1&0&0&0&0&0&1&0&1&1&0\\
0&0&1&1&0&0&1&0&0&0&0&1\\
1&0&0&0&1&1&0&0&1&0&0&0\\
1&0&1&0&0&0&0&1&1&0&0&0\\
0&1&0&1&0&0&0&0&0&1&1&0\\
0&0&0&0&1&1&1&0&0&0&0&1
\end{bmatrix} \tag{2-10}$$

后来 Tanner 对这种 LDPC 的构造方法进行了总结，之后 Mackay 又对其进行了改进提出了 Mackay LDPC 构造方案。

2.3.2 Mackay LDPC 码[37,38]

为了减少译码算法受到环的影响，Mackay 在校验矩阵中引入了一些重量为 2 的列，依此来降低整个校验矩阵的重量来减少二分图中环的数量，并取得了较好的效果。以下介绍 Mackay 随机 LDPC 码构造方法：

向一个大小为 $M\times N$ 的全零矩阵中插入元素 1，使所有列的列重为 γ，且行重均匀的保持为 ρ，同时要求任意两列之间元素 1 的交叠数量不超过 1，避免校验矩阵中 4 循环的存在。

MacKay 码的一个显著的缺点是它的结构无法充分保证编码的低复杂度。

在编码时通过对 H 进行高斯消去，得到校验矩阵的系统形式$H=[P|I]$，由此可得生成矩阵$G=[I|P^T]$。问题在于通过 G 进行编码的时候，子矩阵P^T通常不够稀疏，使得当码长较大的时候，编码的复杂度较高。

有限几何空间可以用来构造 LDPC 码[39]，由林舒首先提出来，依据几何空间不同其又可分为射影几何码（PG_LDPC）和欧式几何码（EG_LDPC），它们均是循环结构的 LDPC 码，其特点是纠检错性能较好，且译码较为方便。本文第三章将详细介绍如何根据欧式空间来构造 EG_LDPC 码，PG_LDPC 码与其的构造方法类似。

PEG[40,41]算法是一种构造 Tanner 图的简单有效方法，主要是在某准则条件下通过加边的方式随机构造 LDPC 码，具体操作是在给定变量节点数目、校验节点数目和变量节点分布的条件下，逐步地在变量节点和校验节点之间加边线，选择加边时，要尽可能保持大的 girth，然后按规则接着放新的边线，直至结束。

2.4 本章小结

本章主要介绍了构造欧式几何低密度单奇偶校验码（EG_LDPC）所用到的的重要概念和原理，包括线性分组码、分组码最小距离以及循环码，重点介绍了 LDPC 码的定义和泰纳图表示方法，而且对 EG_LDPC 码的构造方案进行了简单的介绍，主要包括 Gallager LDPC 码、Mackay LDPC 码等。

第3章 LDPC 码加固 SRAM

本章介绍采用欧式几何低密度单奇偶校验码(EG_LDPC)进行 SRAM 加固的编译码算法以及电路实现。SRAM 容错结构如图 3-1 所示，令信息位的长度为 k，码字长为 n。

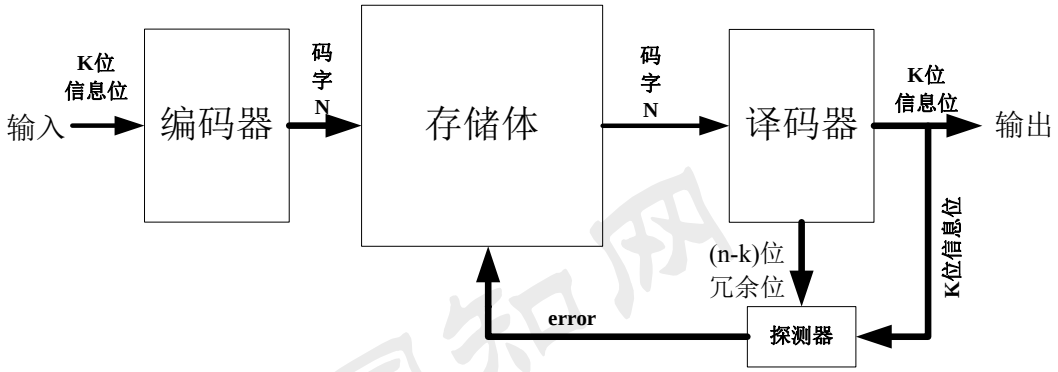


图 3-1 SRAM 加固框图

由图 3-1 可得，在存储体写周期，k 位信息位先输入编码器中，由编码器经编码加上额外的冗余位形成 n 位码字，然后码字 n 被存储在存储体中。当存储体读取数据时，n 位码字先由译码器进行译码修正，形成 k 位信息位的输出，同时被修复的 k 位信息位也和被修复的(n-k)位冗余位被送入检测器，产生检测信号，它可用来检测被修复的码字 n 是否还有不可修复的错误，当被修复的码字没错误时则为低电平，否则为高电平。

EG_LDPC 码拥有自加固特性(3.3 有详细介绍)，和图 3-1 的电路结构配合，不仅可以用来加固存储体，同时也使存储器周围的 EG_LDPC 码的编译码电路拥有了抗软错误能力。当编码器或存储体发生软错误时，在可纠正的范围内译码器均可以将信息位正确译出，且即使在存储体写电路中发生了软错误，译码器也可以纠正。译码器担任纠正功能，如果错误发生在译码器中，译码器也可以将绝大部分的错误屏蔽掉（因 EG_LDPC 码采用大数逻辑译码方案，3.2 节有详细介绍），即使少量错误传播到探测电路，探测电路也可以将错误探测出来，产生高电平探测信号。当错误发生在探测电路中时，因 EG_LDPC 码的自加固特性错误也可被检测出来，检测信号反馈回存储体中，再由存储器对数据重读来消除探测器中的 SET 错误。

3.1 编码器设计

对于(n,k)差错控制编码，编码器可以产生(n-k)位冗余位，一般编码器都是由生成矩阵构建的，对于 EG_LDPC 码也不例外。因此，构建 EG_LDPC 码编码器的重点是得到生成矩阵 G，而构建 G 的重点又是找到该码的生成多项式，本文接下来就来探讨如何得到(n,k)EG_LDPC 码的编码器。

3.1.1 码生成多项式

F 为一组元素的集合，在该集合中定义两种二元运算：加法‘+’和乘法‘•’。如果集合中元素满足：在加法下 F 是一个交换群（关于加法运算的单元称为 F 的零元，记为 0）；F 中的非零元素在乘法下也构成一个交换群（关于乘法运算的单元元称为 F 的幺元，记为 1）；且乘法对加法满足分配律，即对 F 中任意三个元素 a，b 和 c，满足式(3-1)

$$a \bullet (b + c) = a \bullet b + a \bullet c \tag{3-1}$$

则将集合 F 和这两种二元运算一起称为域。

由域的定义可得一个域至少包含两个元素，加法单位元和乘法单位元，域中元素的个数称为域的阶。一个含有有限个元素的域称为有限域（又名伽罗华域）。实数集在实数加法和乘法下就是一个域，该域有无穷多个元素，可以构建含有有限个元素的域。当 p 为素数时，依据上述域定义可得集合{0，1，2，…，p-1}在模 p 加法和模 p 乘法下是一个 p 阶域，因 p 为素数故称为素域，记做 GF(p)，研究发现对于任意素数 p，存在一个含 p 个元素的有限域，而且，对任意正整数 m，都可以把一个素域 GF(p)扩展为一个有 p^m 个元素的域，称为 GF(p)的扩域，记作 $GF(p^m)$。当 p 为 2 时，构成的域通常称为二元域 $GF(2)$，二元域及其扩域 $GF(2^m)$ 在编码理论中起着重要的作用，码的构造和译码的很大一部分内容都是围绕有限域建立起来的。

令 a 为 $GF(q)$ 中的非零元素，因 $GF(q)$ 中非零元素的集合在乘法下是封闭的，域中 a 的任意次幂必然也是 $GF(q)$ 中的非零元素，因为域中元素有限，a 的任意次幂不可能各不相同，因此，在 a 的幂序列的某处必然出现重复，也就是必然存在两个正整数 k 和 m 使得 $m > k$ 且 $a^k = a^m$，因此必然存在一个使得 $a^n = 1$ 的最小正整数 n，称 n 为该域元素 a 的阶。

在有限域 $GF(q)$ 中，如果 a 的阶为 $q-1$，则非零元素 a 被称为本原元，因

为，本原元的幂生成了$GF(q)$中的所有非零元素，任何一个有限域都有一个本原元。如在$GF(7)$中，计算整数 3 的幂，得到(3-2)式：

$$3^1=3，\ 3^2=3\bullet 3=2，\ 3^3=3\bullet 3^2=6，$$
$$3^4=3\bullet 3^3=4，\ 3^5=3\bullet 3^4=5，\ 3^6=3\bullet 3^5=1 \tag{3-2}$$

因此，整数 3 的阶是 6，从而 3 是$GF(7)$的本原元。

每一个伽罗华域都有唯一的一个本原元，并且有相应的域本原多项式，计算本原多项式需要考虑二元域$GF(2)$中元素多项式的计算，以 X 为变量，$GF(2)$中以元素为系数的多项式 $f(X)$ 有如式(3-3)的形式：

$$f(X)=f_0+f_1X+f_2X^2+...+f_nX^n \tag{3-3}$$

其中 $f_i=0$ 或 1，$0\le i\le n$。多项式的次数是系数非零的 X 的最高幂次，且$GF(2)$上的多项式可以按通常的方式进行多项式的加法、减法、乘法和除法运算。假定 $g(X)$ 的次数不为零，当 $f(X)$ 除以 $g(X)$ 时，可得到$GF(2)$上的唯一的一对多项式，分别称为商式$q(X)$和余式$r(X)$，如式(3-4)所示：

$$f(X)=q(X)g(X)+r(X) \tag{3-4}$$

对$GF(2)$上的多项式 $f(X)$，若其含有的项数为偶数，则它能被 $X+1$ 整除，$GF(2)$上的 m 次多项式 $p(X)$若不能被$GF(2)$上任意次数小于 m 且大于 0 的多项式整除，则称 $p(X)$在$GF(2)$上是不可约的。m 次不可约多项式 $p(X)$若满足 $p(X)$整除的 X^n+1的最小正整数 n 为$n=2^m-1$，则称 $p(X)$为该域的本原多项式。对于一个域本原多项式可能不只一个，但我们一般用项数最少的本原多项式。如$GF(2^5)$域的项数最少的本原多项式为 $p(X)=1+X^2+X^5$。本原多项式对伽罗华域很重要，因为其可以将二元域$GF(2)$扩展到$GF(2^{ms})$。

对一个在二元域$GF(2)$上的$(2^{ms}-1)$维向量$v=(v_0,v_1,...,v_{2^{ms}-2})$，可以将 v 的非零分量采用$GF(2^{ms})$上的非零元素来标号，对于$0\le i\le 2^m-2$，分量v_i用 α^i 来标号，因此α^i为元素v_i的位置数，这样就可以把$GF(2^{ms})$看做是域$GF(2^s)$上的 m 维欧式几何，$EG(m,2^s)$。欧式几何可以用来构建欧式几何低密度单奇偶校验码（EG_LDPC）。对于(31,16)EG_LDPC 码，其对应的欧式几何为$EG(5,2^1)$，$EG(5,2^1)$对应的伽罗华域为$GF(2^5)$，这时 m=5，令α为该空间本原元，将其代入域本原多项式中，则有 $p(\alpha)=1+\alpha^2+\alpha^5=0$，即

$1+\alpha^2=\alpha^5$。用该等式可以构造空间$GF(2^5)$，表 3-1 给出了$GF(2^5)$中的所有元素。在构造$GF(2^5)$元素的多项式时需要重复使用等式$1+\alpha^2=\alpha^5$，如式(3-5)和(3-6)所示：

$$\alpha^6=\alpha\bullet\alpha^5=\alpha\bullet(1+\alpha^2)=\alpha+\alpha^3 \tag{3-5}$$

$$\alpha^7=\alpha^2\bullet\alpha^5=\alpha^2(1+\alpha^2)=\alpha^2+\alpha^4 \tag{3-6}$$

表 3-1 $GF(2^5)$中元素多项式列表

幂表示	多项式表示	幂表示	多项式表示	幂表示	多项式表示
α^1	α^1	α^{11}	$1+\alpha+\alpha^2$	α^{21}	$\alpha^3+\alpha^4$
α^2	α^2	α^{12}	$\alpha+\alpha^2+\alpha^3$	α^{22}	$1+\alpha^2+\alpha^4$
α^3	α^3	α^{13}	$\alpha^2+\alpha^3+\alpha^4$	α^{23}	$1+\alpha+\alpha^2+\alpha^3$
α^4	α^4	α^{14}	$1+\alpha^2+\alpha^3+\alpha^4$	α^{24}	$\alpha+\alpha^2+\alpha^3+\alpha^4$
α^5	$1+\alpha^2$	α^{15}	$1+\alpha+\alpha^2+\alpha^3+\alpha^4$	α^{25}	$1+\alpha^3+\alpha^4$
α^6	$\alpha+\alpha^3$	α^{16}	$1+\alpha+\alpha^3+\alpha^4$	α^{26}	$1+\alpha+\alpha^2+\alpha^4$
α^7	$\alpha^2+\alpha^4$	α^{17}	$1+\alpha+\alpha^4$	α^{27}	$1+\alpha+\alpha^3$
α^8	$1+\alpha^2+\alpha^3$	α^{18}	$1+\alpha$	α^{28}	$\alpha+\alpha^2+\alpha^4$
α^9	$\alpha+\alpha^3+\alpha^4$	α^{19}	$\alpha+\alpha^2$	α^{29}	$1+\alpha^3$
α^{10}	$1+\alpha^4$	α^{20}	$\alpha^2+\alpha^3$	α^{30}	$\alpha+\alpha^4$

由表 3-1 得$GF(2^m)$域中有2^m个元素（包括$\alpha^\infty=0$和$\alpha^0=1$元素），且这2^m个元素可以分成 N 个共轭组，每个域的共轭元素组的个数各不相同，共轭组中包含的元素也各不相同。令β是$GF(2)$扩域中的一个元素，$f(X)$是一个以$GF(2)$中元素为系数的多项式，如果β是$f(X)$的一个根，则对任意$l\geq 0$，β^{2^l}也是$f(X)$的根，称元素β^{2^l}为β的共轭。如在$GF(2^5)$域中，有等式$\alpha^{31}=1$，本原元α的共轭如式(3-7)所示：

$$(\alpha^3)^2=\alpha^6,\ (\alpha^3)^{2^2}=\alpha^{12},\ (\alpha^3)^{2^3}=\alpha^{24},\ (\alpha^3)^{2^4}=\alpha^{48}=\alpha^{17} \tag{3-7}$$

表 3-2 是$GF(2^5)$中所有的共轭元素组。

表 3-2 $GF(2^5)$ 中所有的共轭元素组

$\alpha^1,\alpha^2,\alpha^4,\alpha^8,\alpha^{16}$	$\alpha^3,\alpha^6,\alpha^{12},\alpha^{24},\alpha^{17},$	$\alpha^5,\alpha^{10},\alpha^{20},\alpha^9,\alpha^{18}$
$\alpha^7,\alpha^{14},\alpha^{28},\alpha^{25},\alpha^{19}$	$\alpha^{11},\alpha^{22},\alpha^{13},\alpha^{26},\alpha^{21}$	$\alpha^{15},\alpha^{30},\alpha^{29},\alpha^{27},\alpha^{23}$

域中每个元素均有其对应的最小项多项式 $\phi(X)$，且共轭组元素的 $\phi(X)$ 相同，$GF(2^5)$ 域中 $\beta=\alpha^3$ 的最小项多项式为式(3-8)：

$$\phi(X)=(X+\alpha^3)(X+\alpha^6)(X+\alpha^{12})(X+\alpha^{24})(X+\alpha^{17}) \tag{3-8}$$

由表 3-1，将上面的等式右端乘积展开得 $\phi(X)=X^5+X^4+X^3+X^2+1$，下表 3-3 列出了 $GF(2^5)$ 域中所有项的最小项多项式：

表 3-3 $GF(2^5)$ 中元素最小项多项式

α	X^5+X^2+1
α^3	$X^5+X^4+X^3+X^2+1$
α^5	$X^5+X^4+X^2+X+1$
α^7	$X^5+X^3+X^2+X+1$
α^{11}	$X^5+X^4+X^3+X+1$
α^{15}	X^5+X^3+1

(n,k)EG_LDPC 码的生成多项式是由 $GF(2^{ms})$ 的根决定，有定理如下：令 α 为伽罗华域 $GF(2^{ms})$ 的一个本原元，h 为一个比 $2^{ms}-1$ 小的非负整数，长度为 $2^{ms}-1$ 的 EG_LDPC 码的生成多项式 $g(X)$ 以 α^h 作为根的充要条件如(3-9)所示：

$$0 \le \max_{0\le l<2} W_{2^s}(h^{(l)}) \le (m-\mu-1)(2^s-1) \tag{3-9}$$

其中 $h^{(l)}$ 为 $2^l h$ 除以 $2^{ms}-1$ 所得的余项，即 $2^l h=q(2^{ms}-1)+h^{(l)}$。显然，当且仅当 h 可以被 2^s-1 整除时，$h^{(l)}$ 可被 2^s-1 整除，且 $h^{(0)}=h$。式中的 μ 必须取整数，数值与空间 $EG(m,2^s)$ 及其空间中 EG_LDPC 码的信息位和冗余位有关，而且与该码译码复杂度有关，具体的数值可查阅相关文献。$W_{2^s}(h^{(l)})$ 表示的是 $h^{(l)}$ 的 2^s —重量，由上得 $h^{(l)}$ 是一个比 2^{ms} 小的非负整数，求 $h^{(l)}$ 的 2^s —重量时，首先要用 2^s 进制将 $h^{(l)}$ 按式(3-10)形式展开：

$$h^{(l)} = \delta_0 + \delta_1 2^s + \delta_2 2^{2s} + ... + \delta_{m-1} 2^{(m-1)s} \tag{3-10}$$

其中对于$0 \le i < m$有$0 \le \delta < 2^s$，$W_{2^s}(h^{(l)})$就是$h^{(l)}$的2^s进制展开的所有系数的实数和，即如式(3-11)所示：

$$W_{2^s}(h^{(l)}) = \sum_{i=0}^{m-1} \delta_i \tag{3-11}$$

$GF(2^s)$域形成的欧式几何$EG(5,2^1)$空间，可以构造(31,16)EG_LDPC 码，该码对应的μ为 2，由式(3-9)、(3-10)和(3-11)可得满足条件比 31 小的非负整数为 1、2、3、4、5、6、8、9、10、12、16、17、18、20 及 24，因此该码生成多项式$g(X)$的根如式(3-12)所示：

$$\alpha^1, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6, \alpha^8, \alpha^9, \alpha^{10}, \alpha^{12}, \alpha^{16}, \alpha^{17}, \alpha^{18}, \alpha^{20}, \alpha^{24} \tag{3-12}$$

由表 3-2 和 3-3 可得，元素$\alpha^1, \alpha^2, \alpha^4, \alpha^8$和$\alpha^{16}$的最小项多项式相同，元素$\alpha^3, \alpha^6, \alpha^{12}, \alpha^{24}$和$\alpha^{17}$的最小项多项式也相同，元素$\alpha^5, \alpha^{10}, \alpha^{20}, \alpha^9$和$\alpha^{18}$相同，因此该码的生成多项式$g(X)$如式 3-13 所示：

$$\begin{aligned} g(X) &= (X^5 + X^2 + 1)(X^5 + X^4 + X^3 + X^2 + 1)(X^5 + X^4 + X^2 + X + 1) \\ &= 1 + X + X^2 + X^3 + X^5 + X^7 + X^8 + X^9 + X^{10} + X^{11} + X^{15} \end{aligned} \tag{3-13}$$

由定理：如果$g(X)$是次数为 n-k 的多项式且为$X^n + 1$的因式，则$g(X)$生成一个(n,k)循环码，可得式 3-13 所得$g(X)$可以生成(31,16)循环的 EG_LDPC 码。

3.1.2 编码方案

由 3.1.1 可知(n,k)EG_LDPC 码，其生成多项式有如式(3-14)所示形式：

$$g(X) = g_0 + g_1 X + ... + g_{n-k} X^{n-k} \tag{3-14}$$

且 EG_LDPC 所有码字可以由 k 个码多项式$g(X)$，$Xg(X)$，…，$X^{k-1}g(X)$张成，如果以这 k 个码多项式所对应的 n 维向量作为$k \times n$矩阵的行向量，则得有如图 3-2 形式的矩阵 G，称为生成矩阵：

$$G=\begin{bmatrix} g_0 & g_1 & g_2 & . & . & . & . & . & g_{n-k} & 0 & 0 & 0 & . & . & 0 \\ 0 & g_0 & g_1 & g_2 & . & . & . & . & . & g_{n-k} & 0 & 0 & . & . & 0 \\ 0 & 0 & g_0 & g_1 & g_2 & . & . & . & . & . & g_{n-k} & 0 & . & . & 0 \\ . & & & & & & & & & & & & & & . \\ . & & & & & & & & & & & & & & . \\ . & & & & & & & & & & & & & & . \\ 0 & 0 & . & . & . & 0 & g_0 & g_1 & g_2 & . & . & . & . & . & g_{n-k} \end{bmatrix}$$

图 3-2 (n,k)EG_LDPC 非系统形式生成矩阵

通常 G 并非系统形式，为了译码方便，可以通过行变换将其转化为系统形式，图 3-3 给出的是由(31,16)EG_LDPC 码的生成多项式 $g(X)$ 张成的码生成矩阵的系统形式。

$$G=\begin{bmatrix}
1&1&1&1&0&1&0&1&1&1&1&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&1&1&1&0&1&0&1&1&1&1&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&1&1&1&0&1&0&1&1&1&1&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&1&1&1&0&1&0&1&1&1&1&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
1&1&1&1&1&0&1&0&1&0&1&0&1&1&1&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&1&0&0&0&1&0&1&0&0&1&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
1&0&1&1&0&0&0&1&1&0&1&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\
1&0&1&0&1&1&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&1&0&1&0&1&1&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&1&0&1&0&1&1&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&1&0&1&0&1&1&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&1&0&1&0&1&1&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
1&1&1&1&0&0&0&0&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&1&1&1&1&0&0&0&0&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&1&1&1&1&0&0&0&0&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1\\
1&1&1&0&1&0&1&1&1&1&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{bmatrix}$$

图 3-3 (31,16)EG_LDPC 系统生成矩阵

由码生成多项式 $g(X)$ 以及生成矩阵 G 可以实现对循环码的编码，通常(n,k)EG_LDPC 码的编码由三步组成：（1）用 X^{n-k} 乘消息多项式 $u(X)$；（2）用 $X^{n-k}u(X)$ 除以 $g(X)$，得到余式 $b(X)$；（3）构造码字 $b(X)+X^{n-k}u(X)$。所有这三步都可由除法电路实现，如图 3-4 所示，它是一个(n-k)级移位寄存器，其反馈连接基于生成多项式 $g(X)=1+g_1X+g_2X^2+...+g_{n-k-1}X^{n-k-1}+X^{n-k}$，电路[43]，具体操作可查阅相关书目。

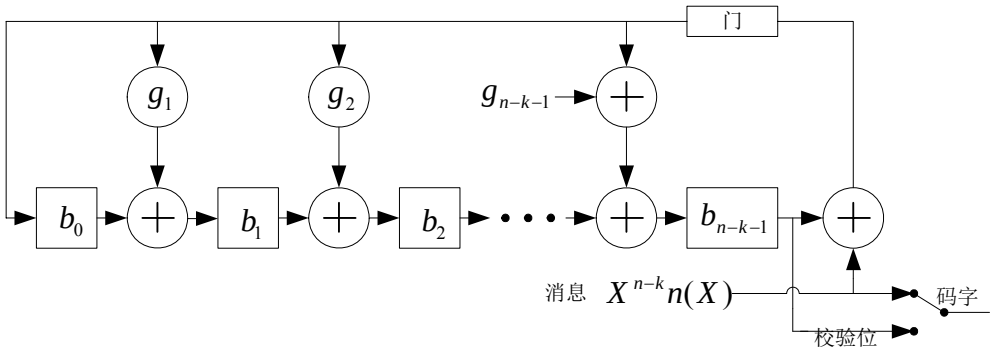


图 3-4 (n,k)EG_LDPC 多周期编码电路

图 3-4 所示编码器的硬件开销比较小，但对每个消息位均需要 k 个周期才可以实现编码，不适合加固 SRAM。因此本文采用如图 3-5 所示的编码方案：

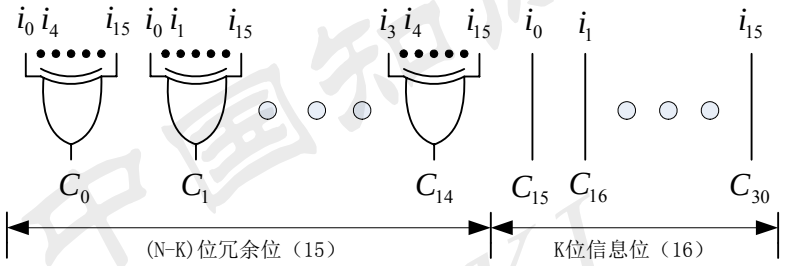


图 3-5 (31,16)EG_LDPC 编码电路

上图是依据图 3-3 矩阵搭建的(31,16)EG_LDPC 码编码器，令信息向量u_{16}为$u_{16}=(i_0,i_1,i_2,...,i_{15})$，码字 C 为$C_{31}=(C_0,C_1,...,C_{30})$。由图可得其分两部分，$C_0 \sim C_{14}$为冗余位，是由信息位通过异或操作得到的，$C_0 \sim C_{14}$分别对应图 3-3 生成矩阵 G 的前 15 列，$C_{15} \sim C_{30}$通过$i_0 \sim i_{15}$直连得到，因此该码是系统码，该编码方案电路简单并可实现单周期译码。

3.2 译码器设计

LDPC 码的译码方式大致可分为：软判决译码和硬判决译码。它们有着不同的纠错性能和译码复杂度。软判决解码算法性能较优，且纠错性能好，但运算复杂度也高[44,45]。在此我们选用硬判决译码中的大数逻辑来对 EG_LDPC 码进行译码，该方法运算简单而且复杂度低，适合对 SRAM 加固。本文接下来将介绍大数译码算法和译码电路。

3.2.1 大数逻辑译码

对于(n,k)EG_LDPC 的生成矩阵 G，其行向量可张成循环码 C，对于系统形式的生成矩阵$G=[P_{n-k,k}I_{k,k}]$，通过简单变换可得另个矩阵$H=[I_{n-k}P^T]$，H 矩阵行向量可张成另个(n,n-k)循环码C_d，对于码 C 中的任一码字v以及码C_d中的任一码字w，其内积为零，即如式(3-15)所示：

$$w \bullet v = w_0 v_0 + w_1 v_1 + ... + w_{n-1} v_{n-1} \tag{3-15}$$

H 矩阵在译码器设计中有重要作用，称 H 为奇偶校验矩阵。现在，假定 C 中的一个码字v被发送，令$e=(e_0,e_1,...,e_{n-1})$及$r=(r_0,r_1,...,r_{n-1})$分别为误差向量和接收向量，则有：$r=v+e$。对于码C_d中的任一向量w，可以根据接收向量构造如式(3-16)的线性和：

$$A = w \bullet r = w_0 r_0 + w_1 r_1 + ... + w_{n-1} r_{n-1} \tag{3-16}$$

如果接收向量r是码C中的一个码字，该线性和 A 一定为 0；反之，如果r不是码C中的码字，A 就可能不是 0。结合式(3-15)和(3-16)及$w \bullet r = 0$，可以得到有关线性和 A 与误差向量e中差错位之间的关系，如式(3-17)所示：

$$A = w_0 e_0 + w_1 e_1 + ... + w_{n-1} e_{n-1} \tag{3-17}$$

如果系数$w_l = 1$，则可以说差错位e_l被校验和 A 所校验，并且利用校验矩阵 H，通过合理构造校验和可以用来估计向量e中的差错位，如在码C_d中存在如下J个向量：

$w_1=(w_{10},w_{11},...,w_{1,n-1})$，$w_2=(w_{20},w_{21},...,w_{2,n-1})\ldots w_J=(w_{J0},w_{J1},...,w_{J,n-1})$

该向量组具有如下性质：

1）每个向量的第(n-1)个分量为 1，即：$w_{1,n-1}=w_{2,n-1}=...=w_{J,n-1}=1$

2）对于$i \neq n-1$，至多存在一个向量的第i个分量为 1。例如，如果$w_{1,i}=1$，则有$w_{2,i}=w_{3,i}=...=w_{J,i}=0$。

这 J 个向量被称为在第(n-1)个位置正交，并称它们为正交向量，由这 J 个正交向量可构造 J 个校验和，如式(3-18)所示：

$$A_1 = w_1 \bullet r，\ A_2 = w_2 \bullet r，\ \bullet\ \bullet\ \bullet，\ A_J = w_J \bullet r \tag{3-18}$$

由于$w_{1,n-1}=w_{2,n-1}=w_{J,n-1}=1$，且这 J 个奇偶校验和与误差向量的差错位具有如式(3-19)所示的关系：

$$A_1 = w_{10}e_0 + w_{11}e_1 + ... + w_{1,n-2}e_{n-2} + e_{n-1}$$
$$A_2 = w_{20}e_0 + w_{21}e_1 + ... + w_{2,n-2}e_{n-2} + e_{n-1}$$
$$\bullet \quad \bullet \quad \bullet$$
$$A_J = w_{J,0}e_0 + w_{J,1}e_1 + ... + w_{J,n-2}e_{n-2} + e_{n-1} \tag{3-19}$$

因此，差错位e_{n-1}能被上述所有的校验和所校验，并且任意一个非e_{n-1}的差错位至多被一个校验和所校验。这 J 个线性和被称为在差错位e_{n-1}上正交，由于$w_{i,j} = 0$或1，则上述在e_{n-1}上正交的校验和均具有如式(3-20)所示的形式：

$$A_j = e_{n-1} + \sum_{i \neq n-1} e_i \tag{3-20}$$

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$e_{n-1} = A_j$），利用该特点可以估计e_{n-1}，并对r_{n-1}接收位进行译码。

假定误差向量$e = (e_0, e_1, ..., e_{n-1})$中有不多于$[J/2]$个差错位（即$e$有不多于$[J/2]$个分量为 1）。如果$e_{n-1} = 1$，那么其它的非零差错位将会分布于至多$[J/2]-1$个在$e_{n-1}$上正交的校验和之中，因此，至少有$J-[J/2]+1$个或者多于一半的在$e_{n-1}$上正交的校验和等于$e_{n-1} = 1$。另一方面，如果$e_{n-1} = 0$，非零差错位将会分布于至多$[J/2]$个校验和中，因此，至少有$J-[J/2]$个，或至少一半的在$e_{n-1}$上正交的校验和等于$e_{n-1} = 0$。因此，$e_{n-1}$的值可以由在$e_{n-1}$上正交的校验和的绝对多数来决定，如果校验和无法给出绝对多数（即二者一样多），则差错位$e_{n-1} = 0$。

如果误差向量e中有不多于$[J/2]$个差错位，可以保证对e_{n-1}位的译码是正确的，如果可以构造在e_{n-1}上正交的J个校验和，那么基于该码的循环对称性，也可以构造出在任意差错位上正交的J个校验和，对其它差错位的译码与对e_{n-1}位的译码完全相同，上述译码算法即是一步大数逻辑译码。

以上的一步大数逻辑译码过程是以可以构造一个在某个差错位上正交的J个校验和的集合为基础的，同样，可以将正交于单个差错位的奇偶校验和的概念进行推广，利用多步大数逻辑判决器对更多的循环码进行译码。我们称由J个奇偶校验和A_1，A_2，…，A_J构成的集合正交于集合E（且含有 M 个差错位的集合$E = (e_{i1}, e_{i2}, ..., e_{iM})$，其中$0 \le i_1 < i_2 < ... < i_M < n$），当且仅当（1）$E$

中的每一个差错位e_{it}被每一个校验和A_j所校验，$1 \le j \le J$；（2）没有任何一个差错位被一个以上的校验和所检验。如下式(3-21)中四个奇偶校验和正交于集合$E = \{e_6, e_8\}$：

$$\begin{aligned} A_1 &= e_0 + e_2 + e_6 + e_8 \\ A_2 &= e_3 + e_4 + e_6 + e_8 \\ A_3 &= e_1 + e_6 + e_7 + e_8 \\ A_4 &= e_5 + e_6 + e_8 \end{aligned} \quad (3\text{-}21)$$

采用与一步大数逻辑译码所用的相同的推理方法，只要在错误模式e中仅存在$[J/2]$个或更少的差错，根据正交集合E的校验和A_1，A_2，…，A_J，就可以正确的确定E中差错位的和$e_{i1} + e_{i2} + ... + e_{iM}$，可以称之为附加校验和，因此可用于译码。

由此我们令E_1^1，E_2^1，…，E_i^1，…为适当地选择错误模式e中的一些差错位而得到的一组集合。令$S(E_i^1)$代表E_i^1中的差错位的模 2 和，假定对每一个集合E_i^1都可以构造出至少J个在该集合上正交的奇偶校验和。那么，和值$S(E_i^1)$可以由这J个正交的校验和来估计。该估计可以由一个以这J个正交的校验和作为输入的大数逻辑判决器来完成。$S(E_i^1)$的估计值就是大数逻辑判决器的输出：当且仅当一半以上的输入为 1 时输出为 1，否则输出为 0。如果在误差向量e中只有$[J/2]$个或更少的差错，判决值就是正确的。接下来，所判决的和$S(E_1^1), S(E_2^1), ..., S(E_i^1)$…，可被用来判决第二次所选集合$E_1^2$，$E_2^2$，…，$E_i^2$…，中的差错位的和，第二次选择的集合的大小比第一次选择的集合要小。假定对每一个集合E_i^2都可以构造至少J个或更多个在该集合上正交的奇偶校验和，如果在误差向量e中存在不多于$[J/2]$个差错，和值$S(E_i^2)$就可以由在E_i^2上正交的校验和来正确的估计。一旦和值$S(E_1^2)$，$S(E_2^2)$，…，$S(E_i^2)$，…被确定，就可以用它们来估计第三次所选集合，E_1^3，E_2^3，…，E_i^3…，中的差错位的和值，第三次选择集合的大小要比第二次选择的集合要小，由已知的校验和来估计校验和的过程被称为正交化过程。正交化过程一直进行，直到寻找到至少J个校验和，它们仅正交于一个差错位e_{n-1}。于是用这

些正交的校验和就可以估计e_{n-1}的值。根据码的循环结构，利用同一电路和相同的方法可以估计出其它差错位。如果对一个差错位进行译码判决需要进行L步正交化，则称此码是L步大数逻辑可译码。因为在正交化的每一步中都用大数逻辑判决器来对所选差错位的和值进行估计，所以译码总共需要L步大数逻辑判决器，而每一级所需判决器的数目是由码的结构所决定的。图 3-6 是一个L步大数逻辑译码器，其纠错过程可查阅相关书目。

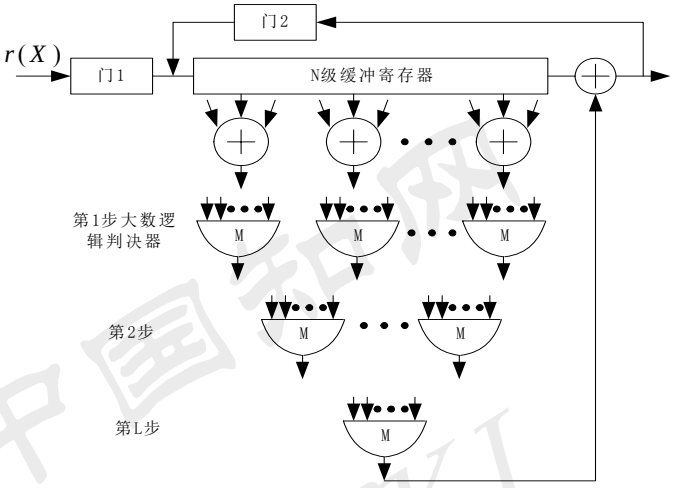


图 3-6 L步大数逻辑译码器

3.2.2 译码方案

由上述讨论可得，校验矩阵 H 与大数译码关系密切，译码的正交校验和矩阵$A^{(i)}$（包括$A_1, A_2, \ldots, A_J$；i为矩阵$A^{(i)}$的正交位置），是由 H 矩阵的各行张成的，以(31,16)EG_LDPC 码为例，图 3-3 为该编码的系统形式的生成矩阵 G，通过变换可得(31,16)EG_LDPC 校验矩阵 H 如图 3-7 所示：

$$H=\begin{bmatrix}
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&1&0&0&0&0&1&0&0&1\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&1&0&0&0&1&0&0&0&1&1&0&1\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&0&1&0&1&1&0&1&0&0&1&1&1&1\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&0&1&0&1&0&1&0&1&1&1&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&0&1&0&1&0&1&0&1&1&1\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&1&1&0&0&0&1&1&0&1&0&0&0&1&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&1&1&0&0&0&1&1&0&1&0&0&0&1\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&1&0&0&0&1&1&0&1&1&0&0&0&0&1\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&1&1&0&1&1&1&1&0&1&0&1&1&1&0&0&1\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&1&1&1&0&0&0&0&0&0&1&0&1&0&1&0&1\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&1&1&1&1&1&1&1&0&0&1&0&0&0&1&1\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&1&0&0&0&0&1&0&0&1&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&1&0&0&0&0&1&0&0&1&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&1&0&0&0&0&1&0&0&1&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&1&0&0&0&0&1&0&0&1&1
\end{bmatrix}$$

图 3-7 (31,16)EG_LDPC 校验矩阵 H

该码的 H 矩阵行向量张成的空间包含了 $EG(5,2^1)$ 中所有不通过原点的三维平面的关联向量。首先，我们将 $GF(2^5)$ 看做是几何 $EG(5,2^1)$。那么通过点 α^{30} 的一维平面由形式为 $\alpha^{30}+a\alpha^i$ 的点组成，其中 $a\in GF(2)$，总共有 29 个一维平面通过点 α^{30} 而不通过原点 $\alpha^{\infty}=0$，如下式(3-22)；

$$\{\alpha^{29},\alpha^{30}\}，\{\alpha^{28},\alpha^{30}\}，\ldots，\{\alpha^{i},\alpha^{30}\}，\{\alpha^{i-1},\alpha^{30}\}，\ldots，\{\alpha^{1},\alpha^{30}\} \quad (3\text{-}22)$$

从这 29 个一维平面中任意取出六个如式(3-23)所示：

$$\{\alpha^{29},\alpha^{30}\}，\{\alpha^{28},\alpha^{30}\}，\{\alpha^{27},\alpha^{30}\}，\{\alpha^{26},\alpha^{30}\}，\{\alpha^{25},\alpha^{30}\}，\{\alpha^{24},\alpha^{30}\} \quad (3\text{-}23)$$

对于每一个一维平面均有 J（J 的值如式(3-24)所示）

$$J=\frac{2^{(m-\mu)s}-1}{2^s-1}=\frac{2^{(5-2)\bullet 1}-1}{2^1-1}=6 \quad (3\text{-}24)$$

个不通过原点的二维平面在其上相交，每个二维平面都由形式 $\alpha^{30}+a\alpha^i+b\alpha^j$ 的点组成，其中 a 和 b 属于 $GF(2)$，且 $i\neq j$。如对于一维平面 $\{\alpha^{29},\alpha^{30}\}$，要求在该直线上相交的六个二维平面，重点是如何选择 α^i 和 α^j 的值，然后通过 a，b 的四种组合即可得到该二维平面。a 和 b 的四种组合如下所示：（0,0），（0,1），（1,0）和（1,1），该二维平面包含的四个位置数即为式(3-25)：

$$\begin{aligned}
&\alpha_1=\alpha^{30}+0\bullet\alpha^i+0\bullet\alpha^j \qquad \alpha_2=\alpha^{30}+0\bullet\alpha^i+1\bullet\alpha^j\\
&\alpha_3=\alpha^{30}+1\bullet\alpha^i+0\bullet\alpha^j \qquad \alpha_4=\alpha^{30}+1\bullet\alpha^i+1\bullet\alpha^j
\end{aligned} \quad (3\text{-}25)$$

因为该二维平面一定包含位置数α^{30}和α^{29}，由(3-25)可令α_1等于α^{30}，位置数α_2等于α^{29}，由表 3-1 可得式(3-26)：

$$\alpha^{30}+\alpha^{16}=\alpha+\alpha^4+1+\alpha+\alpha^3+\alpha^4=1+\alpha^3=\alpha^{29} \tag{3-26}$$

取$\alpha^j=\alpha^{16}$，α^i取除了α^{16}和α^{30}之外的所有位置数，将α^i的值和式$\alpha^j=\alpha^{16}$均代入式 3-25 中即可得六组二维平面，如式(3-27)所示：

$$\{\alpha^4,\alpha^{27},\alpha^{29},\alpha^{30}\}，\ \{\alpha^{20},\alpha^{26},\alpha^{29},\alpha^{30}\}，\ \{\alpha^8,\alpha^{22},\alpha^{29},\alpha^{30}\}$$
$$\{\alpha,\alpha^{25},\alpha^{29},\alpha^{30}\}，\quad \{\alpha^{18},\alpha^{21},\alpha^{29},\alpha^{30}\}，\ \{\alpha^{14},\alpha^{19},\alpha^{29},\alpha^{30}\} \tag{3-27}$$

对于每一个二维平面又各有六个三维平面在该二维平面上相交，且三维空间体具有如下的形式$\alpha^{30}+a\alpha^i+b\alpha^j+c\alpha^k$，其中 a，b，c 均属于$GF(2)$，且$i\neq j\neq k$，对每一个二维平面，寻找三维体的关键是找到每个二维平面相应的α^i，α^j和α^k，如对于$\{\alpha^4,\alpha^{27},\alpha^{29},\alpha^{30}\}$，因$\alpha^{30}+\alpha^{16}=\alpha^{29}$，$\alpha^{30}+\alpha^{25}=\alpha^{27}$，因此取$\alpha^i=\alpha^{25}$，$\alpha^j=\alpha^{16}$，并且取$\alpha^k$为除$\alpha^{25}$，$\alpha^{16}$和$\alpha^{30}$之外的所有值，如取$\alpha^k=\alpha^2$，则可得三维空间$\{\alpha^4,\alpha^7,\alpha^8,\alpha^{23},\alpha^{27},\alpha^{28},\alpha^{29},\alpha^{30}\}$，如表 3-4 所示：

同样道理，当α^k取其他值时，可得其他五个在$\{\alpha^4,\alpha^{27},\alpha^{29},\alpha^{30}\}$上相交但不通过原点的三维空间体，如式(3-28)所示：$\{\alpha^4,\alpha^5,\alpha^{11},\alpha^{13},\alpha^{24},\alpha^{27},\alpha^{29},\alpha^{30}\}$
$\{\alpha^0,\alpha^4,\alpha^9,\alpha^{18},\alpha^{21},\alpha^{27},\alpha^{29},\alpha^{30}\}$，$\{\alpha^2,\alpha^4,\alpha^{14},\alpha^{15},\alpha^{19},\alpha^{27},\alpha^{29},\alpha^{30}\}$
$\{\alpha^3,\alpha^4,\alpha^6,\alpha^{10},\alpha^{17},\alpha^{27},\alpha^{29},\alpha^{30}\}$，$\{\alpha^4,\alpha^{12},\alpha^{20},\alpha^{22},\alpha^{26},\alpha^{27},\alpha^{29},\alpha^{30}\}$ (3-28)

表 3-4 三维空间$\{\alpha^4,\alpha^7,\alpha^8,\alpha^{23},\alpha^{27},\alpha^{28},\alpha^{29},\alpha^{30}\}$算法

[a,b,c]	位置数α	[a,b,c]	位置数α
[0 0 0]	$\alpha^{30}+0\alpha^{25}+0\alpha^{16}+0\alpha^2=\alpha^{30}$	[1 0 0]	$\alpha^{30}+1\alpha^{25}+0\alpha^{16}+0\alpha^2=\alpha^{27}$
[0 0 1]	$\alpha^{30}+0\alpha^{25}+0\alpha^{16}+1\alpha^2=\alpha^{28}$	[1 0 1]	$\alpha^{30}+1\alpha^{25}+0\alpha^{16}+1\alpha^2=\alpha^{23}$
[0 1 0]	$\alpha^{30}+0\alpha^{25}+1\alpha^{16}+0\alpha^2=\alpha^{29}$	[1 1 0]	$\alpha^{30}+1\alpha^{25}+1\alpha^{16}+0\alpha^2=\alpha^4$
[0 1 1]	$\alpha^{30}+0\alpha^{25}+1\alpha^{16}+1\alpha^2=\alpha^8$	[1 1 1]	$\alpha^{30}+1\alpha^{25}+1\alpha^{16}+1\alpha^2=\alpha^7$

向量形式的矩阵 $A^{(4-27-29-30)}$ 如下图 3-8 所示：

$$\begin{bmatrix}
0&0&0&0&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&1\\
0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&1&0&0&0&1&1&0&1&1\\
0&0&1&0&1&0&0&0&0&0&0&0&0&0&1&1&0&0&0&1&0&0&0&0&0&0&0&1&0&1&1\\
0&0&0&1&1&0&1&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&1&1\\
1&0&0&0&1&0&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&0&1&0&1&1\\
0&0&0&0&1&1&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&1&0&1&1
\end{bmatrix}$$

图 3-8 $A^{(4-27-29-30)}$ 正交向量组

依据式(3-27)，同理可推导出矩阵 $A^{(8-22-29-30)}$， $A^{(1-25-29-30)}$， $A^{(20-26-29-30)}$，$A^{(18-21-29-30)}$ 和 $A^{(14-19-29-30)}$，且每个矩阵含有 $J=6$ 个行向量，利用六输入大数逻辑判决器可以求出和值 $S(E^{4-27-29-30})$， $S(E^{8-22-29-30})$， $S(E^{1-25-29-30})$，$S(E^{20-26-29-30})$， $S(E^{18-21-29-30})$ 和 $S(E^{14-19-29-30})$，之后再利用一级六输入大数逻辑判决器就可以将和值 $S(E^{29-30})$ 求出。

同理重复上述步骤可以求出和值 $S(E^{28-30})$， $S(E^{27-30})$， $S(E^{26-30})$，$S(E^{25-30})$ 和 $S(E^{24-30})$，利用六输入大数逻辑判决器就可以将和值 $S(E^{30})$ 求出来，然后就可以实现对位置数 α^{30} 上的码字正确译码。用图 3-6 所示的循环移动就可以实现对整个码字的译码操作。

以上实现的是三步大数逻辑译码，且为多周期译码，延迟，面积等各项开销都很大。本文提出一种算法，可将多步大数译码操作转换为两步，减小了各项开销，使各域的 EG_LDPC 码均适合加固 SRAM，算法流程如下所示：

(1)读取校验矩阵H；

(2)产生在位置数 α^i 和 α^{n-1} 上为1的矩阵Z：将H矩阵任意 N （$N \le m/2$）个行向量组合异或，且满足条件 $\alpha^i=1$、 $\alpha^{n-1}=1$、 $m+1 \le i \le n-2$；

(3)产生在 α^i 和 α^{n-1} 位置上的正交矩阵组 Di：对Z做 $J/2$ 层循环，选取行向量构造矩阵 Di，使满足在($\alpha^0 \sim \alpha^{i-1}$)和($\alpha^{i+1} \sim \alpha^{n-2}$)位置数上的值为0或只有一个为1，形如 $D1_{J/2\times n}=(Z_iZ_j\mathrm{K}\ Z_kZ_m)^T$， $D2_{J/2\times n}=(Z_iZ_j\mathrm{K}\ Z_kZ_l)^T$；

(4)产生正交矩阵 $D_{w\times n}=(Z_iZ_j\mathrm{K}\ Z_kZ_mZ_l)^T$：把 $Di_{J/2\times n}$ 中相互正交的向量组成矩阵D，其中 $w \le J$；

(5)产生在位置数 α^i 和 α^{n-1} 为1的矩阵Y：将H矩阵任意Q（ $N \le Q \le m$ ）个行向量组合二进制相加得矩阵Y，且在位置数 α^i、 α^{n-1} 的值为1；

(6)产生正交矩阵 $A^{(i\_n-1)}{}_{s\times n}=(Z_iZ_j\,\mathrm{K}\,Z_kZ_lZ_mY_iY_j)^T$：选择Y矩阵中与D矩阵各行向量在位置数 α^i、α^{n-1} 正交的向量 Y_iY_j 等，与D组成矩阵 $A^{(i\_n-1)}{}_{s\times n}$，其中 $s\le J$；

(7)产生在其它位置数上正交的矩阵：重复上述a-e步，找寻另外 $s-1$ 组分别在 $\alpha^{p1}\alpha^{n-1}$，$\alpha^{p2}\alpha^{n-1}\mathrm{K}\,\alpha^{p(s-1)}\alpha^{n-1}$ 上正交向量组 $A^{(p1\_n-1)}{}_{s\times n}$，$A^{(p2\_n-1)}{}_{s\times n}\mathrm{K}\;A^{[p(s-1)\_n-1]}{}_{s\times n}$，且 $pi>m$；

步骤(2)中取N≤$m/2$有利于减少步骤(3)循环使用的时间，在选择正交位置数时必须包含 α^{n-1}。在步骤(3)中当 $J/2$ 循环操作得到的向量组数过多时，可以提高到 $J/2+1$ 循环，这样得出维数为 $(J/2+1)\times n$ 的向量组 Di。步骤(4)得到的在 $\alpha^i\alpha^{n-1}$ 上正交的D矩阵如果有多个，应对每一组D矩阵均执行步骤(5)和(6)，可得到多个在 $\alpha^i\alpha^{n-1}$ 上的正交矩阵 $A^{(i\_n-1)}{}_{S1\times n}$、$A^{(i\_n-1)}{}_{S2\times n}\mathrm{K}\,A^{(i\_n-1)}{}_{SI\times n}$。(7)中得出的分别在 $\alpha^{pi}\alpha^{n-1}$ 上正交的矩阵也会有多个，需要从这S个正交矩阵组的每组中选出维数为S的矩阵作为二步译码的的正交矩阵组。

(31,16)EG_LDPC最小距离 J 为6，由上文分析得可采用三级大数逻辑进行译码，但逻辑深度太长不适合加固存储器，故本文采用两步大数逻辑译码方案。依据图3-9算法求得在(α^{29} α^{30})上正交的向量组 $A^{(29\_30)}$，其有四个向量如式(3-29)所示（h_i 表示H矩阵的第 i 行）：

$$\begin{aligned}
A1 &= h_{15} && =(00000000000000100011111000010011)\\
A2 &= h_2+h_3+h_8 && =(01100001000000010000000010100011)\\
A3 &= h_1+h_6+h_9+h_{10} && =(10000100110000000000000001000111)\\
A4 &= h_4+h_5+h_7+h_{11} && =(00011010001000000100000000001011)
\end{aligned}\qquad (3\text{-}29)$$

同理利用两步译码算法也可求出分别在位置(α^{28},α^{30})，(α^{27},α^{30})和(α^{26},α^{30})上正交的向量组 $A^{(28\_30)}$、$A^{(27\_30)}$ 和 $A^{(26\_30)}$，并且每个向量组均包含四个向量。

由正交向量组可构建译码器，如图 3-9 所示，(31,16)EG_LDPC 译码器由 7 个大数逻辑译码结构和一个探测器电路 S 组成，采用两步大数译码，且第 N-1 个大数译码结构的输入是第 N 个大数结构输入循环向右移动 5 位得到的，前六个子译码结构只对输入数据的最后五位进行译码（如 N1 只对 C1-C5 译码），N7 只对 C0 进行译码（避免重复译码），这种并行结构可以实现在单个

周期内译码，但硬件开销会相应的有所增加。

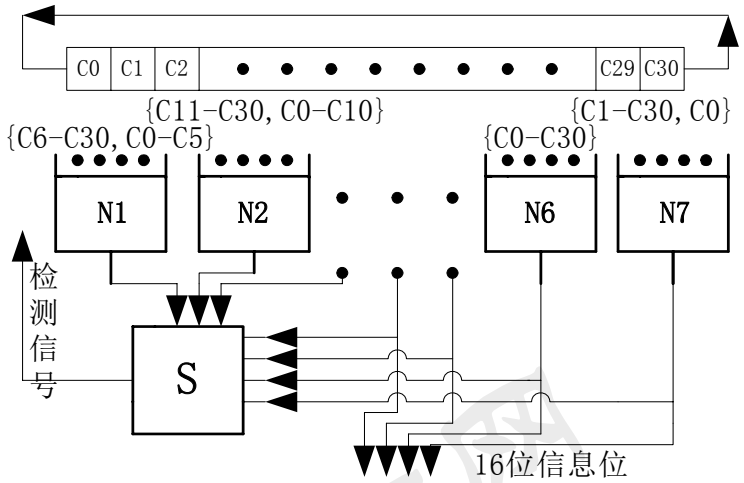


图 3-9 (31,16)EG_LDPC 译码检测器电路结构

图3-10是子大数逻辑N6的内部结构，其中 *Mi* 是大数选择电路，每一个大数逻辑结构均由5个大数选择电路组成（N7子结构中只包含一个大数选择电路），图中可得N6实现了对C26-C30的译码。

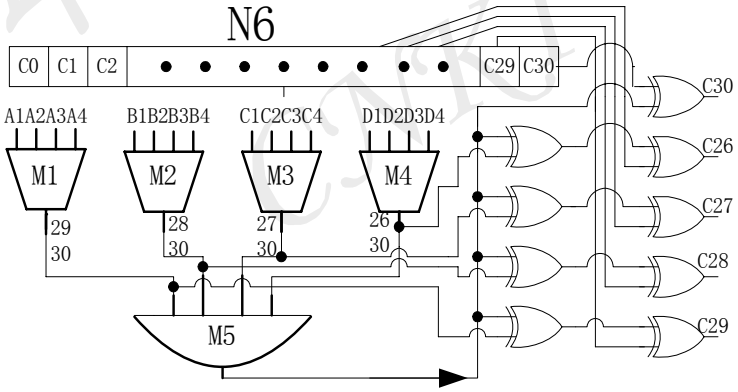
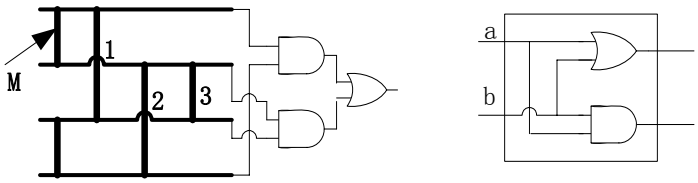


图 3-10 (31,16)EG_LDPC 大数逻辑译码电路

M1-M5的电路结构又如图3-11所示[46]，图3-11(a)是四输入大数逻辑判断电路，图中竖线是子电路，内部电路如图3-11(b)所示，当错误发生在3-11(a)中M处时，通过选择电路在末端会正确的屏蔽掉，同理可得到任意输入端的大数选择电路，该电路结构简单，易于实现。



(a) 四输入大数选择电路　　(b) 竖线电路

图 3-11 (31,16)EG_LDPC 大数选择电路

3.3 探测器设计

探测电路可以探测译码器输出的信息位是否正确，对于(n,k)EG_LDPC码，它会产生n个校正子，校正子位可产生探测信号并反馈回存储器，使EG_LDPC码具有了自加固的能力（即在加固存储体的同时，同时可探测并纠正本身编码器、译码器和校正子电路的软错误）。一个ECC编码是否具有自加固能力，与校正子位密切相关。

一个最小距离为J的ECC码，当错误图样的重量满足 $0 < e \le J-1$，如果这时校正子的重量至少为 $J-e$，则称该ECC编码有自加固能力。令ECC码最大纠正能力为E，最大检测能力为D，数据在存储传输中编码器、译码器和校正电路中发生的错误分别用 E_e、E_d 和 E_s 标记，存储器中的错误为 E_m。对于普通ECC码需要满足 $E_m \le E$ 且 $E_e = E_d = 0$，而对于自加固的ECC码则需满足 $E_e + E_m \le E$ 且 $E_e + E_m + E_d + E_s \le D$。假定自加固ECC中错误图样重量为 $E_e + E_m + E_d = e$，则校正子电路在可检测范围内发生错误的重量应满足 $E_s \le J-1-e$，因校正子电路中每个错误只影响一个校正子位，故这时校正子的重量至少为1，因此可以实现对存储体以及组合电路中错误的检测。

对于(n,k)EG_LDPC，其对应的欧式空间为EG(m, 2^s)，空间中任意一点均有 $\gamma = (2^{ms}-1)/(2^s-1)$ 条直线与其相交，码最小距离 $J = \gamma + 1$。对于重量为E的错误图样，其影响的直线数为 γE，考虑到直线可能经过两个错误点，故受影响的直线至少为 $\gamma E - C_E^2$，因此校正子的重量至少为：

$$W_S = \gamma E - C_E^2 = \frac{2\gamma E - E^2 + E}{2} \ge \frac{2\gamma E - 2E^2 + 2E}{2} \ge J - E \qquad (3\text{-}30)$$

对于 m 维欧式几何 $EG(m,2^S)$，令 F 为 $EG(m,2^S)$ 上的一个不通过原点的 μ

维平面，基于此 μ 维平面，可以构造 $GF(2)$ 上的一个向量如式(3-31)所示：

$$v_F = (v_0, v_1, ..., v_{2^{ms}-2}) \tag{3-31}$$

其中，如果第 i 个分量 v_i 的位置数 α^i 为平面 F 上的一点，则 v_i 为1；否则， v_i 为0。即 v_F 非零分量的位置数构成了 μ 维平面 F 上的点，向量 v_F 称为 F 的关联向量， μ 维平面 F 的关联向量简单地显示了平面 F 上的点，不通过原点的 μ 维平面的关联向量的一个非常有趣的结构特性就是其循环结构：不通过原点 μ 维平面 F 的关联向量经过循环移位就是另一个不通过原点 μ 维平面的关联向量。

校正子的关联向量可由 $V_F = (X^{2^{ms}-1})/G(x)$ 求得，之后将 V_F 扩展成校正矩阵 $[S]_{N\times N}$，且每一行都是上一行循环向左移动一位得到的，以(31,16)EG_LDPC为例，该码的关联向量由式(3-32)得到。

$$v_F = \frac{X^{2^{ms}-1}}{G(x)} = x^{16} + x^{12} + x^{11} + x^{10} + x^9 + x^4 + x + 1 \tag{3-32}$$

即为 $v_F = (1000111100010011)$，由 v_F 按式 3-2 可得矩阵 S 如图 3-12 所示：

$$\begin{bmatrix}
1&0&0&0&1&1&1&1&0&0&0&0&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&1&1&1&1&0&0&0&0&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&1&1&1&1&0&0&0&0&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&1&1&1&1&0&0&0&0&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&1&1&1&1&0&0&0&0&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&1&1&1&1&0&0&0&0&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&1&1&1&1&0&0&0&0&1&0&0&1&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&1&1&1&1&0&0&0&0&1&0&0&1&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&1&0&0&0&0&1&0&0&1&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&1&0&0&0&0&1&0&0&1&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&1&0&0&0&0&1&0&0&1&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&1&0&0&0&0&1&0&0&1&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&1&0&0&0&0&1&0&0&1&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&1&0&0&0&0&1&0&0&1&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&1&0&0&0&0&1&0&0&1&1&1\\
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&1&0&0&0&0&1&0&0&1\\
1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&1&0&0&0&0&1&0&0\\
0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&1&0&0&0&0&1&0\\
0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&1&0&0&0&0&1\\
1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&1&0&0&0&0\\
0&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&1&0&0&0\\
0&0&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&1&0&0\\
0&0&0&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&1&0\\
0&0&0&0&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&1\\
1&0&0&0&0&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1\\
1&1&0&0&0&0&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1\\
1&1&1&0&0&0&0&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1\\
1&1&1&1&0&0&0&0&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&1&1&1&1&0&0&0&0&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&1&1&1&1&0&0&0&0&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&1&1&1&1&0&0&0&0&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1
\end{bmatrix}$$

图 3-12 (31,16)EG_LDPC 校正矩阵 S

探测电路由 S 矩阵搭建，如图 3-13 所示，经过译码后的码字为 $v=(v_0v_1...v_{30})$，如果译码正确，则有 $vS^T=0$，由此可得到 31 个校正子，且 S_i 是由 v 和 S 矩阵的第 i 行的向量乘积得到的，如图 3-14 中所示 $S_0=v_0\oplus v_4\oplus v_5\oplus v_6\oplus v_7\oplus v_{12}\oplus v_{15}\oplus v_{16}$。为了防止检测电路中发生的单个软错误影响到多个输出端口，要求探测电路中每个校正子生成电路没有逻辑共享单元。如果译码后的 v 仍有错误，或者检测电路内部发生了软错误，检测信号均不等于 0，如果检测信号不为 0，则可反馈回存储体中，请求对原始数据重新读取一次，这样可以消除检测电路本身发生的软错误，使检测电路具有了自加固能力。

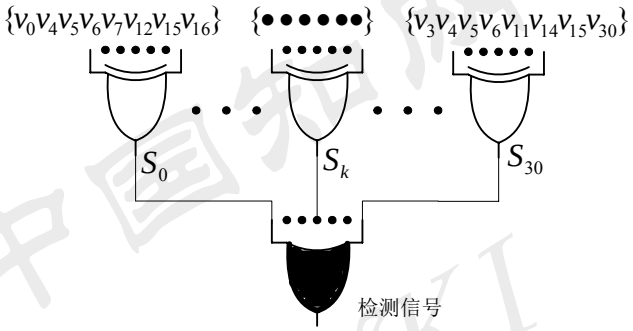


图 3-13 (31,16)EG_LDPC 校正子电路

图 3-13 中阴影部分为或门，如果软错误发生在该处，可能会影响的检测信号，因此要求对该处或门进行加固，在此我们选用冗余门结构[47]。图 3-14 是利用冗余门结构加固后的两输入或门电路，其基本思想是加多余的晶体管（如图中 $M1$ 和 $M2$）将 CMOS 电路中的 PMOS 块与 NMOS 块分开，这样就会产生四个输入（*Ap Bp An Bn*），其中 PMOS 部的输入又来驱动后一个门的 PMOS 部。这样就可以消除 SET 错误，因为粒子只有击中反偏的 PN 结时才可能产生瞬态电流，因此当一个门仅由 PMOS 构成时，其就不可能发生 1 到 0 的跳变，同样由 NMOS 构成的门不可能发生 0 到 1 的跳变，即使发生了瞬态电流，也可以通过 $M1$ 和 $M2$ 泄放掉。

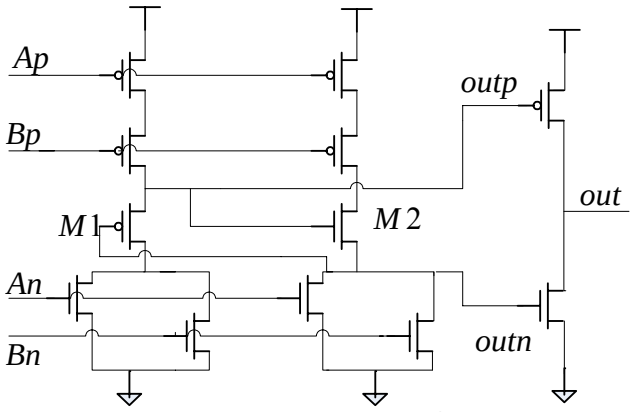


图 3-14 冗余门加固的与门电路

3.4 压缩型 EG_LDPC 码

由上文分析可得，$GF(2^S)$ 域上的 m 维欧式几何 $EG(m,2^S)$ 可用来构造 LDPC 码，且所构造的 LDPC 码的奇偶校验矩阵 H_{EG} 行向量是 $EG(m,2^S)$ 中直线的关联向量，列向量对应 $EG(m,2^S)$ 中的每个点，并且 H_{EG} 包含 J 个行和 $n=2^{ms}$ 个列，J 的值如式(3-33)所示：

$$J=\frac{2^{(m-1)s}(2^{ms}-1)}{2^s-1} \tag{3-33}$$

由于 $EG(m,2^S)$ 中的每条直线包含 2^S 个点，故 H_{EG} 的每一行的重量为 $\rho=2^S$。又由 $EG(m,2^S)$ 中的每个点与 $(2^{ms}-1)/(2^s-1)$ 条直线相交，可知 H_{EG} 的每一列的重量 $\gamma=(2^{ms}-1)/(2^s-1)$，且由该 H_{EG} 张成的可一步实现大数译码码字的重要参数如下：

长度：	$n=2^{2s}-1$
校验比特数：	$n-k=3^s-1$
最小距离：	$d_{\min}=2^s+1$
密度：	$r=2^s/(2^{2s}-1)$

表 3-5 分别列出了由空间 $EG(2,2^2)$、$EG(2,2^3)$ 和 $EG(2,2^4)$ 构造的(15,7)，(63,37)和(255,175)EG_LDPC 码，其中 k 为信息位长度，n 为码字长度，$d_{\min}$ 为该码的最小距离，t_{ML} 是该码可以纠正的比特位数，μ 参数与译码步数有关，采用大数逻辑译码算法需要 $\mu+1$ 步大数逻辑才可以实现译码。

表 3-5 部分 EG_LDPC 码表及相关参数

m	s	μ	n	k	$d_{\min}$	t_{ML}
2	2	0	15	7	4	2
2	3	0	63	37	8	4
2	4	1	255	175	16	8
•••	•••	•••	•••	•••	•••	•••

EG_LDPC 码不只是表 3-5 所列的几组，对同一个空间，也可以构造好多组 EG_LDPC 码，如 $EG(5,2^1)$ 空间可同时构造 $(31,26)$，$(31,16)$ 和 $(31,6)$ 的 EG_LDPC 码，当然，构造的方式不同，译码所需的步数也有所不同。

由表 3-5 可得其信息位均不是 2 的 n 次方，由几何空间 $EG(m,2^S)$ 构造的 LDPC 码共有 51 组，其中只有一组的信息位为 2 的 n 次方，即上文所讲的由 $EG(5,2^1)$ 空间所构造的(31,16)EG_LDPC 码，为了构造合适信息位的 EG_LDPC 码，本文介绍了压缩型 EG_LDPC 码，通过对生成矩阵和校验矩阵合理的改造，使码的信息位均为 2 的 n 次方，故可以用来加固 SRAM。

对于有限几何 LDPC 码，可以通过适当地删除其生成矩阵 G 或者奇偶校验矩阵 H 的若干列，从而压缩原始 LDPC 码，得到一个新的 LDPC 码，删除的列对应于码构造所基于的有限几何中适当选取的一个点集[48]。考虑在 m 维欧式几何 $EG(m,2^S)$ 上构造的 m 维循环 EG_LDPC 码，设 H_{EG} 为这种码的奇偶校验矩阵，H_{EG} 的行向量是 $EG(m,2^S)$ 中不通过原点的所有直线的关联向量，H_{EG} 的列向量对应 $EG(m,2^S)$ 中非零点的所有点。对于 $1\le q<2^s$，令 S 为 $EG(m,2^S)$ 中不包含原点的 q 个平行的 $(m-1)$ 维平面的集合。由于 $EG(m,2^S)$ 中每个 $(m-1)$ 维平面包含 $2^{(m-2)s}(2^{(m-1)s}-1)/(2^s-1)$ 条直线，所以 S 包含

$$J_1=2^{(m-1)s}(2^{(m-1)s}-1)q/(2^s-1) \tag{3-34}$$

条 $EG(m,2^S)$ 中不通过原点的完整直线，可以将 H_{EG} 对应于 S 中点的列删除，删除这些列后，对应于 S 中直线的 H_{EG} 的整行必定全为零，并且将这些行删除就可以得到一个新的矩阵 $H_{EG,S}$。新矩阵 $H_{EG,S}$ 的每一列与原矩阵 H_{EG} 的列有着相同的重量 $(2^{ms}-1)/(2^s-1)-1$，但是它们的行重量并不相同，对应于完全在 S 外的直线的行，重量为 2^S；而对应于部分属于 S 的直线的行，其重量小于

2^S。因此所得的$H_{EG,S}$是一个非规则低密度矩阵，其可以构造压缩的EG_LDPC 码。但压缩的列数是受限制的，不可以任意压缩，接下来考虑如何利用生成矩阵 G 构造可任意压缩信息位的 EG_LDPC 码。

考虑$EG(2,2^3)$空间构造的(63,37)EG_LDPC 码，利用第三章中生成矩阵的推导方法可求得(63,37)EG_LDPC 码的生成矩阵如图 3-15 所示：

图 3-15 (63,37)EG_LDPC 码生成矩阵

上图是(63,37)EG_LDPC 码系统形式的生成矩阵$G_{37\times63}$，通过它可以实现信息位为 37，码字长为 63 的编码器的构建。为了加固普通 SRAM，可以通过适当的删除$G_{37\times63}$矩阵中的若干行来得到合适的信息位，如将图 3-16 青色部分的五列五行删除则可得到信息位为 32、码长为 58 的(58,32)EG_LDPC 码的生成矩阵$G_{32\times58}$，同样如果将图中青色和黄色部分都删除则可得到(42,16)EG_LDPC 码的生成矩阵$G_{42\times16}$。用这种方式，可以构造任意信息位的 EG_LDPC 码的生成矩阵，但在构造时删除的列只能是原始矩阵中单位矩阵部分的列。因此被构造出来的码的冗余位与原始码的一定相同，如$G_{42\times16}$和$G_{32\times58}$的冗余位均与原始矩阵$G_{37\times63}$相同，均为 26 位，编码器电路与图 3-5 的电路拓扑结构相同。

在进行译码的时候，所用的校验矩阵均为原始矩阵$G_{37\times63}=\left[P_{37\times26}I_{37\times37}\right]$通过变换得到的矩阵$H=\left[I_{26\times26}P^{T}\right]$，并且采用一步大数译码即可实现译码操作，求得在$\alpha^{62}$上正交的译码向量如图 3-16 所示：

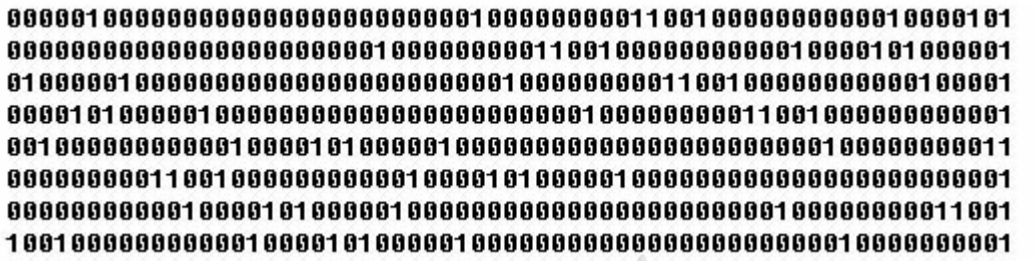


图 3-16 (63,37)EG_LDPC 码一步大数译码正交向量组

可得正交译码向量每行都有 63 位，在对(42,16)和(58,32)EG_LDPC 码进行译码时需要对码字用“0”位在末尾补齐，然后再进行大数逻辑译码，电路拓扑结构如图 3-9 和 3-10 所示，但大数选择逻辑需要用八输入选择器。

3.5 本章小结

本章主要介绍了拥有自加固能力的 SRAM 的容错电路结构，并以(31,16)EG_LDPC 码为例详细阐述了编码器生成矩阵 G 的构造及其编码器的电路实现，又提出了构建译码正交矩阵的算法流程，进而设计了适合加固 SRAM 的并行大数译码器，之后又对 EG_LDPC 码的自加固能力进行了理论分析，并构建了校正矩阵 *S* 以及相应的探测器电路，最后提出了压缩型 EG_LDPC 码的构建方案，使每一个 EG_LDPC 码在不改变纠检错能力条件下信息位可以任意收缩，以获得拥有合适信息位可用来加固 SRAM 的好码。

第4章 可靠性和性能分析

前文已经详细描述了EG_LDPC的编译码方案，接下来本章将对(31,16)和(42,16)的EG_LDPC码的可靠性和性能做仿真分析。验证编译码功能正确的同时并通过了存储器行为模型仿真，之后又用错误注入仿真对EG_LDPC码的纠正和检测的百分率进行了统计，评估了面积、延迟和功耗的开销，并与汉明码(Hamming)、矩阵码(Matrix)和里德—穆尔码(Reed_Muller)进行了比较。

4.1 编译码器功能仿真

本文设计的编码器、译码器和探测器均由硬件语言 Verilog 编写，并用仿真软件 Modelsim6.0 进行了功能仿真。图 4-1、4-2 分别是(31,16)EG_LDPC 码编码器和译码探测器的仿真结果。

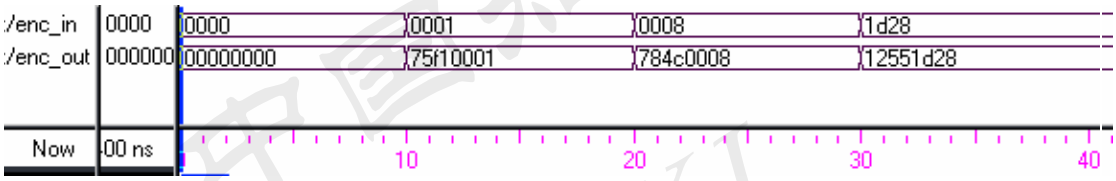


图 4-1 (31,16)EG_LDPC 码编码器功能仿真

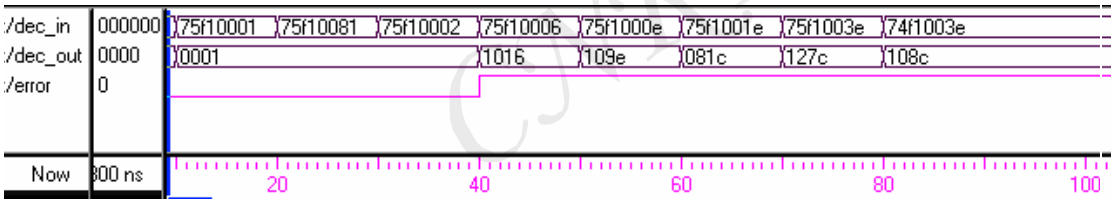


图 4-2 (31,16)EG_LDPC 码译码探测器功能仿真

图 4-1 中 enc_in 为编码器的输入，enc_out 为编码输出。图中对 enc_in 进行了四次输入数据的变换，如第二次采样数据为‘16’H0001’，输出为‘16’H75f10001’，与理论分析编码值相同，因此编码器功能正确。图 4-2 中 dec_in 为译码探测器的输入，dec_out 为输出，error 为探测器的探测信号，当其为‘0’时表示译码正确，否则译码错误。由图可得输入共有八个周期，令第一个周期输入正确码字‘16’H75f10001’，之后每个周期的码字比上个周期多注入一个错误，即这八个周期输入端数据注入的错误数目为{0,1,2,3,4,5,6,7}，已知码字‘16’H75f10001’正确的译码结果为‘16H0001’，由 dec_out 和 error 的仿真波形图可得该(31,16)EG_LDPC 码可纠正小于等于两个的随机错误，且探测能力

较强，甚至可探测七个错误，但对七位错误并不是 100%可探测。

图 4-3 和 4-4 分别是(42,16)EG_LDPC 码的编码器和译码器仿真波形图，各信号的意义与图 4-1 和 4-2 中的相同，并且 dec_in 的六个周期输入数据被注入错误数目分别为{0,1,2,3,4,5}，由 dec_out 和 error 结果显示(42,16)EG_LDPC 可以实现纠小于等于四个的随机错误。

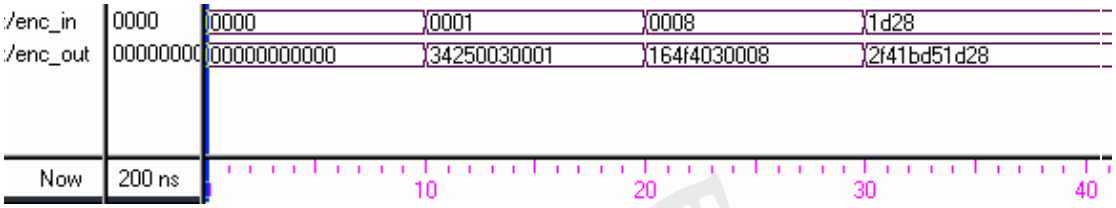


图 4-3 (42,16)EG_LDPC 码编码器功能仿真

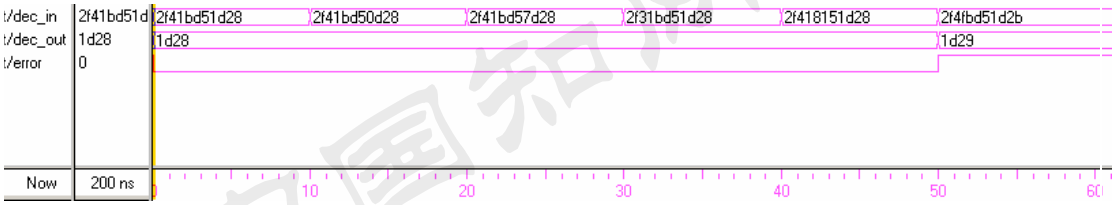


图 4-4 (42,16)EG_LDPC 码译码器功能仿真

4.2 故障注入仿真验证

本文对(31,16)和(42,16)EG_LDPC 码进行了存储器行为模型仿真分析。由编码器和译码探测器电路模块和生成的存储器模块，按图 3-1 所示的容错电路拓扑结构将各模块连接起来，之后对该模型进行了一读一写仿真操作。

图 4-5 是(31,16)EG_LDPC 码纠单位错误 SRAM 模型仿真图，图中只显示了重要信号的数据变换，图中 clk 为读时钟信号，在 clk 上升沿时存储器从数据文件中读取一个新的 16 位信息位，enc_in 是存储器的信息位的输入端（也是编码器的输入端），enc_out 是信息位经编码后形成的输出码字，之后 enc_out 直接存入到存储器中。数据在传输或者在存储器中可能会发生软错误，因此可以直接将两位错误注入码字中来模拟该行为，dec_in_w 即是注入错误错误的码字。之后 dec_in_w 由存储器读出，并在译码器和检测器中进行校正并检测，形成输出的信息位 dec_out 和探测信号 error。如图 4-5 在 54ns 时，clk 出现上升沿，信息位‘16’H2e45’被存储器读入，经编码形成码字‘16’H23182e45’，之后经注入错误得到有两位错误的码字‘16’H23182645’并被存入存储器中，在 58ns 时存储器读取错误码字，并对其进行译码形成输出信

息位‘16’H2e45’，与存储器在 54ns 时读入的数据一样，因此该码可以正确纠正单位的错误。同理，可以注入多位错误，验证该编码对多位错误的纠正能力。

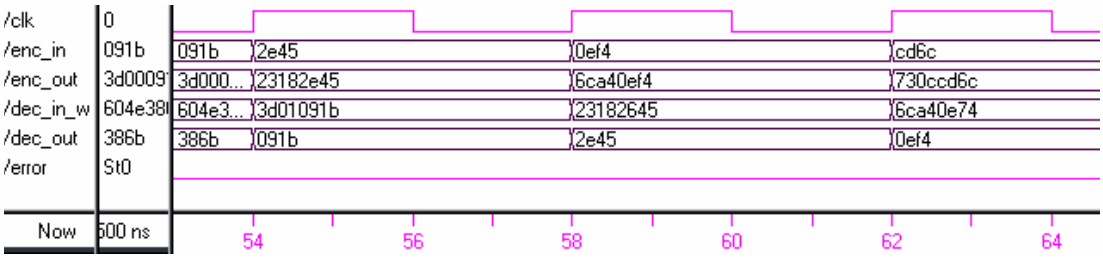


图 4-5 (31,16)EG_LDPC 码纠单位错误功能仿真

图 4-6 是(31,16)EG_LDPC 码纠两位错误 SRAM 模型仿真图，图中各信号的意义和图 4-5 中的一样。在 246ns 时，clk 出现上升沿，信息位‘16’Hb097’被存储器读入，经编码形成码字‘16’H12fcb097’，之后经注入错误得到有两位错误的码字‘16’H7a5659e7’并被存入存储器中，在 250ns 时存储器读取错误码字，并对其进行译码形成输出信息位‘16’Hb097’，且错误检测信号均是低电平，因此可以正确的纠正两位的随机错误。

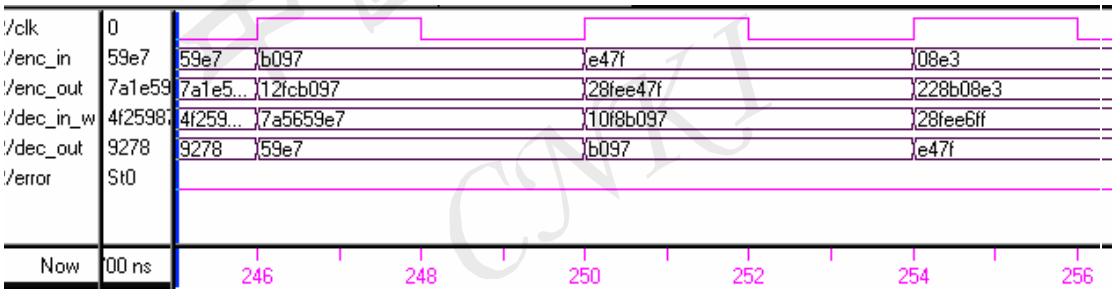


图 4-6 (31,16)EG_LDPC 码纠两位错误功能仿真

(31,16)EG_LDPC 码可实现检测三位错误的功能，图 4-7 是在 SRAM 单个字中注入三个错误的仿真图。在 306ns 时，存储器读入新的数据 enc_in 为‘16H8e1f’，经编码和错误注入，译码器在 310ns 时输出译码信息位为‘16Hef07’，这时 error 信号输出为高电平，检测译码信息位有误，因此可以正确的检测三个错误的情况，同样在 314ns 时 error 信号为低电平，可见该码有纠正部分三位错误的能力，表 4-1 利用错误注入对其纠正多位错误的能力进行了详细统计。

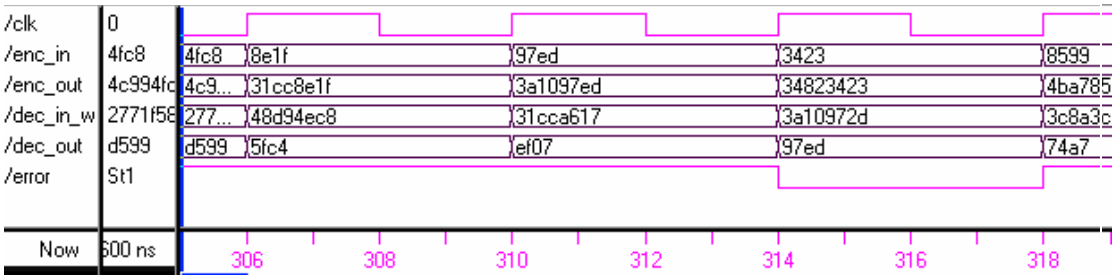


图 4-7 (31,16)EG_LDPC 码纠检三位错误功能仿真

当然，用同样的方法可以验证(42,16)EG_LDPC 码的纠检错误的能力，图 4-8 和 4-9 分别是在 SRAM 字中注入四个和五个错误之后，用(42,16)EG_LDPC 码纠检的波形图，图中信号的意义与图 4-5 中的一样，由图可得该码可以实现纠四检五的功能。

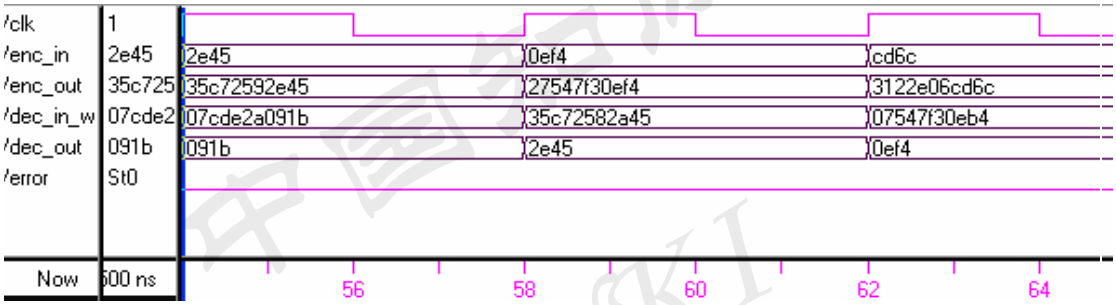


图 4-8 (42,16)EG_LDPC 码纠检四位错误功能仿真

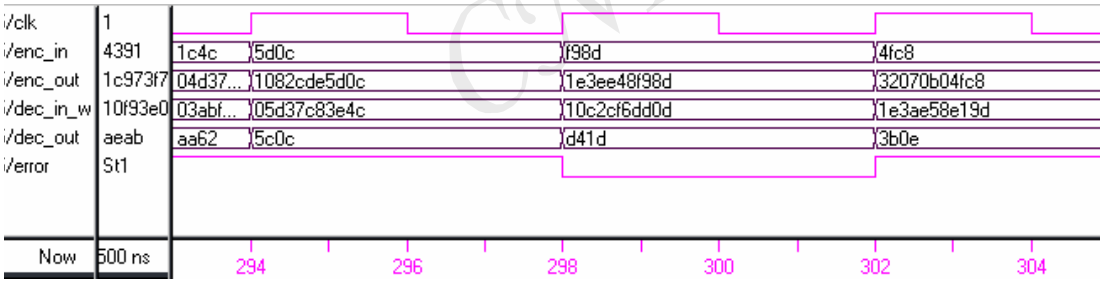


图 4-9 (42,16)EG_LDPC 码纠检五位错误功能仿真

在研究容错覆盖率时，本文采用16位容量为16M的存储器模型，仿真时对存储器中的每个字均注入相同数目的错误，以此来计算本编译码器对该数目软错误的纠正率和检测率。考虑到SEU和MBUs效应，需要求不同错误数目的纠正和检测率，因此对模型做多次故障注入仿真，且每次仿真每个字注入的错误数目都比前次仿真的数目多1。

该模型共进行了四次注入仿真，根据结果分别统计出了每个容错码对四个

以内随机错误的纠正率和检测率，如表4-1所示。表中第一列是在每个字中注入错误的数目，最后一行表示的是该容错码是否具有自加固能力（Y表示有，N表示没有），可得EG_LDPC码和RM码的故障检测率和故障纠正率都要比Matrix码和Hamming码高，并且在这五种差错码中(42,16)EG_LDPC码的性能最好，其可以纠正四个以内的随机错误，即使注入五个错误下，也可百分百检测出来，另外，虽然RM码的纠正率比(31,16)EG_LDPC码的纠正率高，但(31,16)EG_LDPC码的错误检测率高，并且EG_LDPC码同时具有自加固功能，而其它三种码均不具有。表4-2是信息位为16时各ECC码的码率，由表知Hamming码率最高，不过纠检错误能力只能达到纠一检二，(31,16)EG_LDPC码比RM和Matrix的码率略高，(42,16)EG_LDPC的码率最低但可以实现纠四检五，纠检能力最好，且对于大于五个的错误，检测率也可达90%以上。

表4-1 ECC检测率和纠正率列表

故障数目	(31,16)EG_LDPC		(42,16)EG_LDPC		Reed_Muller[26]		Matrix[26]		Hamming	
	检测率(%)	纠正率(%)	检测率(%)	纠正率(%)	检测率(%)	纠正率(%)	检测率(%)	纠正率(%)	检测率(%)	纠正率(%)
1	100	100	100	100	100	100	100	100	100	100
2	100	100	100	100	100	100	100	100	100	0
3	100	0.02	100	100	100	100	94.1	79.3	0	0
4	100	0.01	100	100	100	88.3	80.7	57.9	0	0
5	98.2	0	100	0.05	93.74	65.1	62.2	35.1	0	0
自	Y				N		N		N	

表4-2 ECC校验位码率列表(16位信息位）

	(42,16)EG_LDPC	(31,16)EG_LDPC	R_M	Mat	Ham
校验位	26	15	16	16	6
码率(R)	0.382	0.516	0.5	0.5	0.727

4.3 纠错性价比与检错性价比

为了分析各码纠检错误的开销，由编码器、译码器和探测器的硬件描述，用Synopsys工具Design Compiler综合可得出面积和关键路径延迟参数，Design Power工具可得到功耗参数，所用的工艺库均为SMIC的0.18微米工艺库。

各ECC码编码器和译码器的面积，功耗和延迟参数如表4-3所示，其中以Ham码为比较标准，(31,16)EG_LDPC码的面积、功耗和延迟分别比Ham码增加

了3.92、1.58和1.13倍，与同样可以纠正两位的Matrix码相比各项参数都高，但EG_LDPC码有自加固能力。(42,16)EG_LDPC码的各项参数都最大，但其可以纠四检五，而且检七位错误的能力也很强，纠检错误能力最强并且可以加固编译码电路。为了综合比较各ECC码的纠检错误能力和开销大小，可以利用纠错性价比—Correction Coverage Per Cost（CCC）来评估[26]：

$$CCC(N) == \frac{Correction\_Coverage(N)}{Cost} \tag{4-1}$$

在此Cost如下定义：

$$Cost == Power \bullet Delay \bullet Area \tag{4-2}$$

其中Correction_Coverage(N)为能纠正N个错误的百分率，Cost为面积、功耗和延迟的乘积，在此假定Hamming、Matrix和RM码均采用三模冗余加固，因此需要乘个加固系数T，同理可以得出检错性价比—Detection Coverage Per Cost（DCC）参数。

表4-3 ECC面积、功耗延迟参数列表

	面积（μm^2）		功耗（mw）		延迟（ns）	
Ham	28437	1	39.8	1	1.27	1
Mat[26]	99764	3.51	105.4	2.64	2.35	1.85
RM[26]	160944	5.66	172.3	4.33	3.25	2.55
(31,16)EG_LDPC	139835	4.92	102.7	2.58	2.71	2.13
(42,16)EG_LDPC	256639	9.01	194.6	4.89	3.24	2.55

图4-10描述了各编码的CCC参数，由图可得(31,16)EG_LDPC码的CCC比其它三种码都要高，虽其只能纠正两个错误，但却是MATR码的185%左右，比RM码更高，大约是其的740%。(42,16)EG_LDPC码CCC虽低于Matrix，但其可完全纠正四个以内的随机错误而Matrix只能纠两个错误，比起纠正3个错误的RM码，(42,16)EG_LDPC大约高出80%左右。

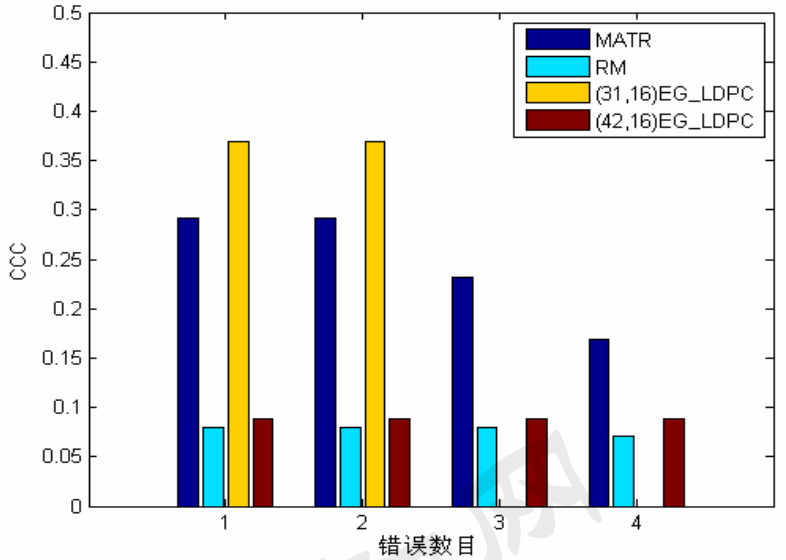


图 4-10 CCC 参数柱状图

图4-11是各编码DCC参数柱状图，可得(31,16)EG_LDPC的DCC参数保持均衡并且最高RM码最低，(42,16)EG_LDPC码的DCC虽然低，但其纠检能力强，可应用与可靠性要求较高的环境下。

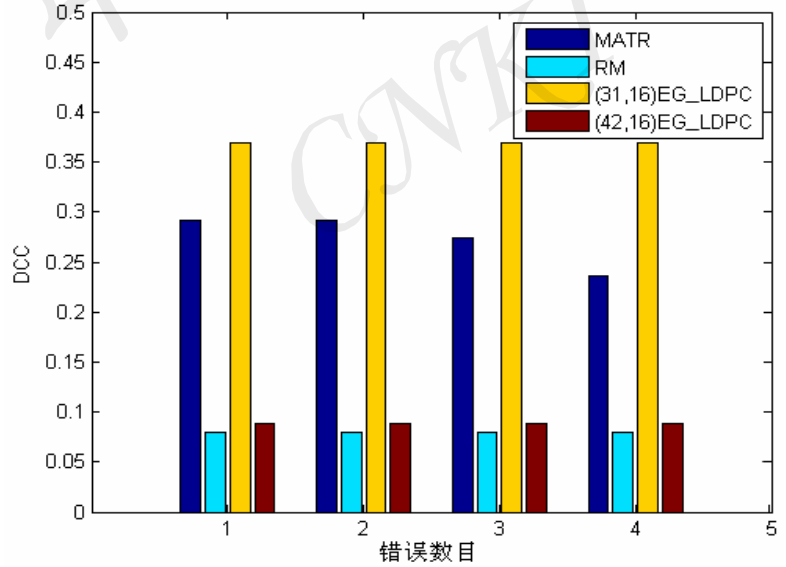


图 4-11 DCC 参数柱状图

4.4 平均失效时间与平均可检测时间

本文将对EG_LDPC码的可靠性进行分析，并且与Hamming码、Matrix码以及Reed_Muller码进行对比。在此我们用平均可纠正率和平均可检测率来进行

评估，并以平均可纠正率为例来介绍。首先，假定粒子辐射的能量符合柏松X-P(3)分布[49]，一次辐射引起的N位错误也符合该柏松分布并可求出其概率（分别用p1，p2，…pN表示）。

对于特定ECC码加固后的存储器，其单个字经N次辐射仍可被纠正的概率与码的纠错能力L有关（L为最大可纠正数），单个字经N次辐射仍可被纠正的概率用 $R(N)$ 表示，表达式如(4-3)所示：

$$R(N)=\sum_{i+j+\ldots+z\le L} p(i\_1)\bullet p(j\_2)\bullet\cdots\bullet p(z\_m) \tag{4-3}$$

且满足条件： $m=N$ 。在此， $p(z\_m)$ 是该字节第m次辐射引起 z 个错误的概率，由定义得 $p(z\_m)=p(z)$ ，如 $p(2\_5)$ 即为该字节第5次辐射发生两个错误的概率。为了保证经N次辐射后该字仍可被纠正过来，要求 $i+j+\ldots+z\le L$ ，且i，j和z可以相等。基于上述理论，经N次辐射该存储器仍可完全被纠正的概率 $J(N)$ 如式(4-4)所示：

$$J(N)=\sum_{i+j+\ldots z=N}\frac{C_Q^m}{Q^m}\bullet R(i\_1)\bullet R(j\_2)\bullet\ldots\bullet R(z\_m) \tag{4-4}$$

且满足条件： $m\le N$ ，Q为存储器的字容量。 C_Q^m 表示从存储器的Q个字中任意选取 m 个， $R(z\_m)$ 为这第 m 个字被辐射 z 次仍可被纠正的概率，即 $R(z\_m)=R(z)$ 。 $J(N)$ 即是存储器经过N次辐射的平均可纠正率。

在此我们用32个字容量的存储器进行分析，其信息位均是16位，对于不同的ECC码，每个字的位数会有所不同，单个字和存储器的可纠正率可分别由上述式(4-3)和(4-4)求得，存储器平均可纠正率如图4-12所示，定义字每次翻转的间隔 $\lambda=10^{-4}$ （次/天），由图4-12可用Matlab求出在任意时间点存储器的平均可纠正时间以及在任意时间点的可纠正率。图中可得(42,16)EG_LDPC码的平均可纠正率最高而Ham码的最低。图4-13是辐射能量满足X-P(6)分布的ECC可检测率分布，由图知(42,16)EG_LDPC码和(31,16)EG_LDPC码的检测能力最高。

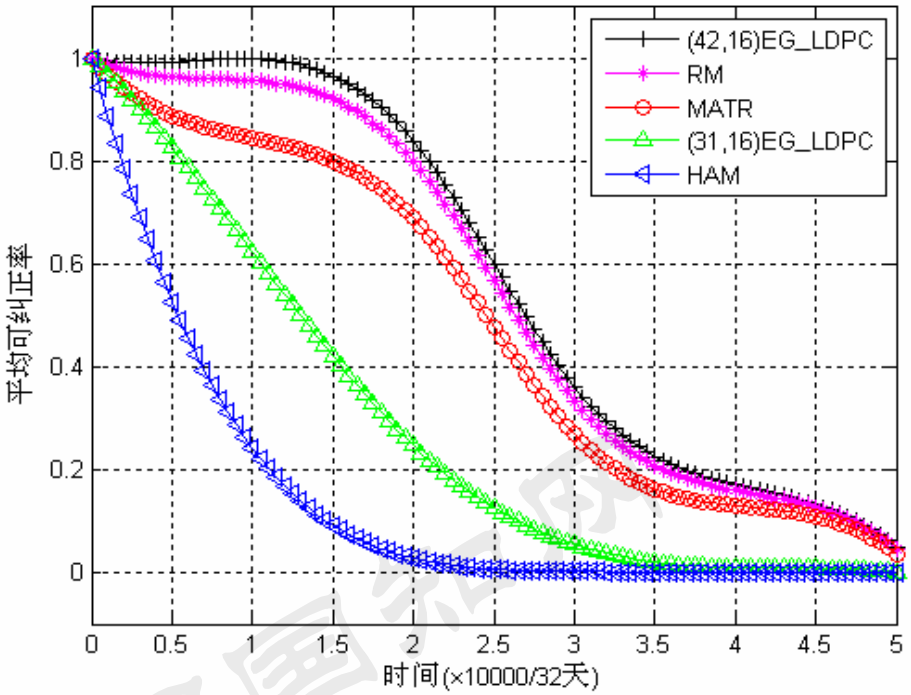


图 4-12 ECC 可纠正时间曲线图

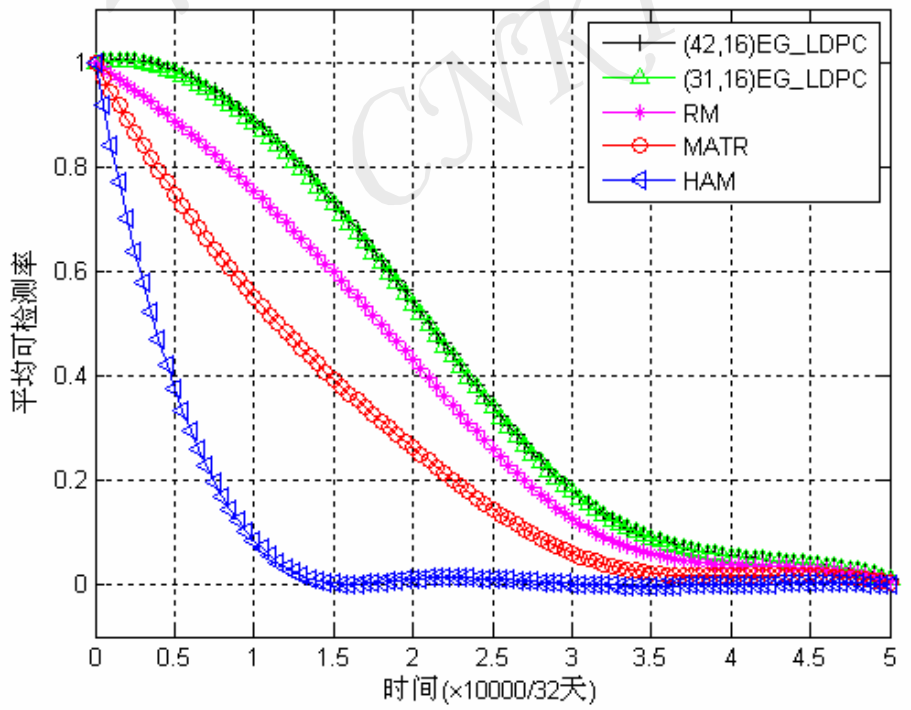


图 4-13 ECC 可检测时间曲线图

表4-4是ECC码的平均失效时间（MTTF）列表，可通过式4-5对每条曲线

进行积分得到[26]：

$$MTTF = \int_0^{\infty} J(t)\,dt \tag{4-5}$$

虽然 RM 码和 Matrix 比(31,16)EG_LDPC 码的平均纠正率高，但 RM 和 Matrix 不能纠正自身编译码电路的错误，由表也可得(42,16)EG_LDPC 码的 MTTF 参数最高，比 Matrix 码高出 18.9%，甚至比 Ham 码高出 3.19 倍。同理可以求出利用各 ECC 码加固存储器的平均可检测时间（Average Detection Time,ADT），表 4-5 是 ECC 码的平均可检测时间列表，可得(42,16)EG_LDPC 和(31,16)EG_LDPC 码要比 RM 码高 18%左右。

表4-4 能量X-P(3)平均无故障时间列表

MTTF	(42,16)EG_LDPC	(31,16)EG_LDPC	R_M	Mat	Ham
(单位：天)	918	450	878	772	219

表 4-5 能量 X-P(6)平均可检测时间列表

ADT	(42,16)EG_LDPC	(31,16)EG_LDPC	R_M	Mat	Ham
(单位：天)	691	686	584	429	151

4.5 本章小结

本章首先利用Modesim对(42,16)EG_LDPC和(31,16)EG_LDPC码的编译码器进行了仿真，验证了(42,16)EG_LDPC码纠四检五和(31,16)EG_LDPC码纠二检四的功能，之后按图3-1所示自加固容错电路结构将各模块搭接起来并对其进行了故障注入仿真验证，通过对存储体、组合电路的错误注入验证了该存储器对加固存储器的同时也可以自加固的性能，之后利用错误注入实验对EG_LDPC码的纠正率和检测率进行了分析，并利用平均纠错开销与平均检错开销评估了该码的各项消耗，最后利用平均无故障时间和平均可检测时间评估了被加固存储器的可靠性，且与汉明码、里德—穆尔码和矩阵码进行了比较。结果表明EG_LDPC码以较小的开销获得了较高的可靠性，而且在加固存储器的同时，是自身的编译码电路也有了抗软错误性能。

结　论

为确保存储器存取传输数据的可靠性，通常采用差错控制码来减少存储阵列中SEU错误，但随着特征尺寸的减少，差错控制编译码组合电路中发生SET错误的几率也越来越大，针对这一现象，本文设计了一种可同时消除SEU和SET错误的SRAM加固方案，本文取得的成果如下：

（1）介绍了本加固方案的电路拓扑结构，并阐述了由EG_LDPC码和反馈环结构使电路实现自加固能力的机理。文中以(31,16)EG_LDPC为例，通过对欧式几何空间和低密度单奇偶码的研究，给出了EG_LDPC码生成矩阵G的详细构造步骤，并给出了由矩阵G构造码器的方案。

（2）由生成矩阵G通过变换得到了校验矩阵H，并由两步译码算法得到了(31,16)EG_LDPC码的两步大数译码矩阵，实现了纠二检四的并行大数逻辑译码器，大大提高了译码速度，同时在可纠检错误能力范围内不仅可完全消除SEU错误，而且可将组合电路（包括编码器和译码器）中的SET错误屏蔽掉。

（3）由生成矩阵G推导得到校正矩阵S，矩阵S使EG_LDPC拥有了自加固能力并从理论上进行了推导验证，进而本文又提出了信息位可随意改变的压缩型EG_LDPC码构造方案，并实现了纠四检五的压缩型(42,16)EG_LDPC码。

最后，基于 SMIC0.18um 工艺，对上述两种码进行了可靠性和性能分析，并与其它 ECC 码进行了比较。结果显示，(31,16)EG_LDPC 码与 Matrix 和 RM 码相比，在面积、功耗以及延迟上开销都要小，平均纠错开销（CCC）分别是 Matrix 和 RM 的 185%和 740%，(42,16)EG_LDPC 码的平均纠错开销高于 RM 码而低于 Matrix 码，但它纠检错误能力较高，约是 Matrix 能力的两倍，并且(42,16)EG_LDPC 码拥有最高的平均失效时间 MTTF，比 Matrix 码高出 18.9%。可见利用 EG_LDPC 仅需要少量的额外开销就可以使编译码具有自加固能力，使电路系统的可靠性大大加强。

参考文献

1 Q. Zhao, K. Mohanram. Transistor Sizing for Radiation Hardening. IEEE 42nd Annual International Reliability Physical Symposium. Phoenbc, 2004: 310-315.(Z)

2 D. Binder, E.C. Smith, and A.B. Holman, “Satellite Anomalies from Galactic Cosmic Rays,” IEEE Trans. Nuclear Science, vol. 22, Dec. 1975: 2675-2680

3 T. May, M. Woods. Alpha-particle-induced Soft Errors in Dynamic Memories[J]. IEEE Transactions on Electron Devices, 1979, 26(1):2–9

4 J. F. Ziegler, W. A. Lanford. Effect of Cosmic Rays on Computer Memories[J]. SCIENCE, 1979, 206(16):776–788

5 J. Olsen, P.E. Becher, P.B. Fynbo, P. Raaby, and J. Schult, “Neutron-Induced Single Event Upsets in Static Rams Observed at 10km Flight Altitude,” IEEE Trans. Nuclear Science, vol. 40, Dec. 1993: 120-126

6 E. Normand, “Single Event Upset at Ground Level,” IEEE Trans. Nuclear Science, vol. 43, Dec. 1996: 2742-2750

7 S. I. Association. International Technology Roadmap for Semiconductors 2007 Edition[R]. Tech. rep., 2007

8 Rodney M. Goodman, Masahiro Sayano, "The Reliability of Semiconductor RAM Memories with On-Chip Error-Correction Coding" IEEE Transactions on information theory, VOL. 37, NO. 3, MAY 1991: 884-896

9 A.H.Johnston. Radiation Effects in Advanced Microelectronics Technologies. IEEE Transactions on Nuclear Science. 1998,45(3):1339-1354

10 R. C. Baumann. Radiation-Induced Soft Errors in Advanced Semiconductor Technologies. IEEE Transactions on Device and Materials Reliability. 2005: 5(3) 305~316

11 R. C. Baumann. Soft Errors in Advanced Semiconductor Devices–Part I: Three Radiation Sources. IEEE Transactions on Device and Materials Reliability. 2001: 17~22

12 Nguyen, H.T, Yagil, Y. A systematic approach to SER estimation and solutions.Reliability Physics Symposium Proceedings, 2003. 41st Annual. 2003 IEEE International.30 March-4 April 2003 Page(s):60~70

13 R. C. Baumann. “Radiation-Induced Soft Errors in Advanced Semiconductor Technologies” IEEE Transactions on Device and Materials Reliability.

2005,5(3):305-316.

14 G. C. Cardarilli, A. Leandri, P. Marinucci, M. Ottavi, S. Pontarelli, M. Re, and A. Salsano, “Design of a fault tolerant solid state mass memory,” IEEE Trans. Rel., vol. 52, no. 4, Dec. 2003: 476–491

15 Tipton, A.D., Jonathan A. Pellish, Patrick R. Fleming, Ronald D. Schrimpf, Robert A. Reed, Robert A. Weller, et al., “High Energy Neutron Multiple-Bit Upset”, Proc. of IEEE Int. Integrated Circuit Design and Technology, 2007. ICICDT'07 June 2007: 1-3

16 A.M. Chugg, M.J. Moutrie and R. Jones, “Broadening of the variance of the number of upsets in a read-cycle by MBUs”, IEEE Trans. on Nuclear Science, Volume 51, Issue 6, Part 2, Dec. 2004: 3701-3707

17 Yasuo Yahagi, Hironaru Yamaguchi, Eishi Ibe, Hideaki Kameyama, Masatoshi Sato, Takashi Akioka, Shigehisa Yamamoto “A Novel Feature of Neutron-Induced Multi-Cell Upsets in 130 and 180 nm SRAMs” IEEE Trans. on Nuclear Science, Vol. 54, No. 4, August 2007: 1030-1036

18 R. Rajsuman. Design and Test of Large Embedded Memories:An Overview. IEEE Design and Test of Computers. 2001: 18(3): 16-27

19 Rodney M. Goodman, Masahiro Sayano, "The Reliability of Semiconductor RAM Memories with On-Chip Error-Correction Coding" IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 37, NO. 3, MAY 1991: 884-896

20 D. Bossen, J. Tendler, K. Reick. Power4 System Design for High Reliability. IEEE Micro, 2002, 22(2):16–24

21 R. Baumann. Soft Errors in Advanced Computer Systems. IEEE Design & Test of Computers, 2005, 22(3):258–266

22 G. C. Cardarilli, A. Leandri, P. Marinucci, M. Ottavi, S. Pontarelli, M. Re, and A. Salsano, “Design of a fault tolerant solid state mass memory,” IEEE Trans. Rel., vol. 52, no. 4, Dec. 2003: 476–491

23 David G.Mavis and H.Eaton, SEU and SET Modeling and mitigation in deep submicron technologies, IEEE 45th Annual International Reliability Physics Symposium, Phoenix, , 2007: 293-305

24 G.L.Feng and K.K.Tzeng, On the Generalized Hamming Weights of Several Classes Of Cyclic Codes, IEEE Tran On Information Theory. Vol. 38.NO.3. May.1992: 1125-1130

25 Ofer Amrani and Y. Beery, Reed–Muller Codes: Projections onto GF(4) and Multilevel Construction, IEEE Tran On Information Theory. Vol. 47.NO.6.September.2001: 2560-2565
26 Costas Argyrides, Hamid R. Zarandi, Matrix Codes Multiple Bit Upsets Tolerant Method for SRAM Memories, 22nd IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems.2007: 340-348
27 Wei Chen, Rui Gong, Kui Dai and Fang Liu,"Two New Space-Time Triple Modular Redundancy Techniques for Improving Fault Tolerance of Computer Systems" The Sixth IEEE International Conference on Computer and Information Technology, 2006: 175-175
28 Rajesh Garg and Nikhil Jayakumar, Circuit-Level Design Approaches for Radiation-Hard Digital Electronics, IEEE Tran on very large scale Interation systems,VOL.17,NO.6,June 2009: 781-792
29 Helia Naeimi and Andre DeHon, Fault Secure Encoder and Decoder for NanoMemory Applications IEEE Transactions on very large scale integration(VLSI) systems, VOL.17, NO.4, APPIL 2009: 473-486
30 C.Berrou and A.Glavieux. Near optimum error correcting coding and decoding: turbo-codes[J]. IEEE Trans, Comm,Oct.1996,44(10):1261-1271
31 C.Berrou, A.Glavieux,and PThimajshima. Near Shannon limit error-comeding coding and decodingaurbo-codes[C]. In Proc,of ICC'93, May 1993,.1064-1070
32 D. J. C. Mackay and R. M. Neal. Near Shannon limit performance of low density parity check codes. Electronics Letters, Aug. 1996, 32: 1645-1646
33 M. C. Davey and D.J.C.Mackay. Low density parity check codes over GF(q). IEEE Commun. Let., June 1998, 2: 165-167
34 Bane Vasic, B. D. Ivan, Raymond K. Kostuk. Low Density Parity Check Codes and Iterative Decoding for Long-Haul Optical Communication Systems. IEEE Journal Lightwave Technology, 2005:21(2):438-446
35 S.YChung, G.D.Forney, T.J.Richardson, R.Urbanke. On the Design of Low Density Parity-Check Codes within 0.0045dB of the Limit. IEEE Communications Letters, 2001: 5(2): 58-60
36 R G Gallager. Low density parity-check codes. Cambridge, Mass: MTT Press 1963
37 D. J. C. MacKay.Good error correcting codes based on very sparse matrices. IEEE Trans Inform Theory,1999, Vo1.45, No.2. 399-431

38 D. J. C. MacKay, R Neal. Good error correcting codes based on very sparse matrices. Crgptography and Coding, Sth, IMA Conf., C. Boyd, Ed.，Lecture Notes in Computer Science. 1995: Sprinter. 100-111

39 B. Amman B. Honary, Y. Kou, J. Xu and S. Lin. Construction of Low-Density Parity-Check Codes Based on Balanced Incomplete Block Designs. IEEE Transactions on Information Theory. 2004, June, Vo1.50, No.6. 1257-1268

40 Xiao-Yu Hu,Dieter-Michael Arnold. Efficient Implementation of the Sum-Product Algorithm for Decoding LDPC Codes. Global Telecommunications Conference, 2001: 1036-1036

41 Xiao-Yu Hu, Evangelos Eleftheriou, Dieter-Michael Arnold.Regular and irregular progressive edge-growth Tanner graphs. Global Telecommunications Conference. 2001: 995-1001

42 J.Campello, D.S.Modha. Extended bit-filling and LDPC code design. IEEE Global Telecommunications Conference. 2001: Nov. 25-29

43 王新梅,肖国镇.纠错码原理与方法.西安电子科技大学出版社.1991.

44 Y. Kou, S. Lin, and M. P. C. Fossorier. Low Density Parity Check Codes based on Finite Geometries: A Rediscovery, in Proc. IEEE ISIT'00, Sorrento, Italy, 2000: 200

45 J. Zhang, and M. P. C. Fossorier. A modified weighted bit-flipping decoding of low-density Parity-check codes. IEEE Communications Letters, Vol. 8, No. 3, ,2004: 165-167

46 H. Naeimi and A.DeHon. Fault Secure Encoder and Decoder for Memory Applications. 22nd IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems. 2007: 409-417

47 Rajesh Garg and Sunil P. Khatri. A Novel Highly SEU Tolerant Digital Circuit Design Approach. IEEE International Conference on Computer Design. Oct. 2008: 14-20

48 Shu Lin and Daniel J.Costello, 差错控制编码. 晏坚 何元智等译 第 2 版 北京：机械工业出版社 2007

49 P. Reviriego, J.A. Maestro, Study of the effects of multibit error correction codes on the reliability of memories in the presence of MBUs. IEEE Trans. Dey Mater Reliab. VOL.9,NO 1, , March 2009: 31-39

哈尔滨工业大学硕士学位论文原创性声明

本人郑重声明：此处所提交的硕士学位论文《LDPC 码在 SRAM 加固中的应用研究》，是本人在导师指导下，在哈尔滨工业大学攻读硕士学位期间独立进行研究工作所取得的成果。据本人所知，论文中除已注明部分外不包含他人已发表或撰写过的研究成果。对本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。本声明的法律结果将完全由本人承担。

作者签字：张艳晶　　日期：2010 年 7 月 1 日

哈尔滨工业大学硕士学位论文使用授权书

《LDPC 码在 SRAM 加固中的应用研究》系本人在哈尔滨工业大学攻读硕士学位期间在导师指导下完成的硕士学位论文。本论文的研究成果归哈尔滨工业大学所有，本论文的研究内容不得以其它单位的名义发表。本人完全了解哈尔滨工业大学关于保存、使用学位论文的规定，同意学校保留并向有关部门送交论文的复印件和电子版本，允许论文被查阅和借阅，同意学校将论文加入《中国优秀博硕士学位论文全文数据库》和编入《中国知识资源总库》。本人授权哈尔滨工业大学，可以采用影印、缩印或其他复制手段保存论文，可以公布论文的全部或部分内容。

本学位论文属于（请在以下相应方框内打“√”）：

保密□，在　　年解密后适用本授权书

不保密☑

作者签名：张艳晶　　日期：2010 年 7 月 1 日

导师签名：韩玉伟　　日期：2010 年 7 月 1 日

致　谢

在此，首先要衷心感谢我的导师肖立伊教授两年来对我的培养、教育和关心。肖老师以其精深的学术水平和渊博的知识不断启发着我，以敬业的精神以及严谨的工作态度指导着并激励着我，当我遇到困难时，不断鼓励着我。肖老师为我的论文付出了巨大的心血，促使我科研水平的大大提高，使我受益匪浅，终身难忘。

衷心感谢来逢昌副教授、高志强副教授以精深的专业知识，强烈的责任心，热情和平易近人的态度长期教导并感染着我，使我的科研设计知识不断进步，并为我们创造了理想的科研环境，提供了许多便利条件。

感谢王进祥教授、李晓明老师、王永生老师和金文毅老师等为我们提供的帮助，方便了我们的学习和生活。同时感谢祝名、孙宇师兄对我的课题工作的耐心指导和帮助，他细致的工作和认真的工作态度对我影响巨大。同时感谢同一实验室的所有师兄师姐和同学，是他们陪伴我走过了两年难忘的时光，谢谢他们所给予的帮助和鼓励，让我难以忘怀这份情谊。

更要感谢我的父母和家人，他们付出了巨大的心血，含辛茹苦，为我做出了奋斗的榜样，是他们为我铺就了一条宽阔的求学教化之路。