

DOI:10.11918/202307075

纳米级 SRAM 多位翻转检纠错方法实现

薛国风^{1,2}, 安军社¹, 周昌义¹

(1. 中国科学院国家空间科学中心, 北京 100190; 2. 中国科学院大学, 北京 100049)

摘要: 为解决纳米级 SRAM (100 nm 以内工艺) 在航天工程应用中出现的多位翻转问题, 依据纳米级 SRAM 的翻转特性, 在传统串行编译码的基础上优化改进, 采用并行编译码的方式, 实现了基于 RS(12, 8, 4) 码的纳米级 SRAM 的加固方法, 在单时钟节拍内实现编译码输出。以 FPGA 为平台, 验证该加固方法的延时和纠错能力。测试结果表明: 与 Xilinx 自带的可检二纠一汉明码的块 RAM 相比, 本文提出的方法访问延时相近, 但纠错能力是汉明码的 5~8 倍; 与 FUEC-QUAEC、CLC 等编译码方法相比, 将连续 5 bit 翻转错误的纠正率提高到 100%。采用并行编译码实现的基于 RS(12, 8, 4) 码加固方法可用于纳米级 SRAM 抗多位翻转加固, 以较小的延时代价实现纠正一个码字 (48 bit) 内任意两个符号 (最多 8 bit) 内的错误, 可完全纠正空间单粒子环境中出现的单个字内连续 5 bit 翻转的错误。该加固方法可扩展应用到 CPU 外部存储器的访问控制以及 CPU 内部 cache 的加固, 以解决现有航天处理器采用检二纠一码无法纠正其 cache 多位翻转错误的问题。

关键词: 单粒子效应; 多位翻转; RS 编码; 纳米级 SRAM

中图分类号: TP399 **文献标志码:** A **文章编号:** 0367-6234(2025)09-0039-07

Implementation to mitigate multi-bit upset in nano-scale SRAM

XUE Guofeng^{1,2}, AN Junshe¹, ZHOU Changyi¹

(1. National Space Science Center, CAS, Beijing 100190, China;
2. University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: In order to solve the issue of multi-bit upset in nano-scale SRAM (processes below 100 nm) in aerospace applications, this study optimizes and improves traditional serial encoding and decoding methods based on multi-bit upset (MBU) characteristics of nano-scale SRAM. A parallel encoding and decoding approach is employed to implement a reinforcement method based on RS(12, 8, 4) code, enabling encoding and decoding outputs within one single clock cycle. The effectiveness of this reinforcement method in terms of delay and error correction capability is validated based on an FPGA platform. The test results show that, compared to the built-in Hamming code of Xilinx Block RAM, the proposed method has an equivalent output delay but with 5 to 8 times greater error detecting and correcting capability than those of Hamming code. Furthermore, when compared to encoding and decoding methods such as FUEC-QUAEC, CLC, the correction rate for consecutive 5-bit upset errors is elevated to 100%. Using parallel coding and decoding method, the implemented RS(12, 8, 4) code is effective for reinforcing multi-bit upset in nano-scale SRAM. At a minimum latency cost, it allows for the correction of any two symbols (up to 8 bits) within a single codeword (48 bits), fully correcting errors involving consecutive 5-bit upset within a single word in space radiation environment. The proposed MBU reinforcement method can be extended to external memory control interface or internal cache of CPUs addressing the issue of multi-bit upset errors in caches of existing aerospace processors that rely on single-error correction codes.

Keywords: single event effect; multi-bit upset; RS code; nano-scale SRAM

空间辐射环境及单粒子效应是影响星上产品特别是 SRAM 可靠性的重要因素。文献[1-4]通过实验发现, 在 180~20 nm CMOS SRAM 中多位翻转概率随着器件工艺尺寸的减小呈增大趋势。多位翻转错误无法采用传统的检二纠一算法规避, 从而更

易引发数据及系统错误。因此, 在纳米工艺下, 单粒子引发的多位翻转错误成为航天应用中必须要解决的问题。

为实现对单粒子翻转错误的免疫或者减缓, 通常从工艺、底层电路、体系结构 3 个层次对 RAM 进

收稿日期: 2023-07-26; 录用日期: 2023-11-08; 网络首发日期: 2024-12-19

网络首发地址: <https://link.cnki.net/urlid/23.1235.T.20241218.1811.004>

基金项目: 国家重点研发计划(2022YFF0503900)

作者简介: 薛国风(1984—), 女, 博士研究生; 周昌义(1965—), 男, 研究员, 博士生导师

通信作者: 周昌义, zhoucy@nssc.ac.cn

行加固。

65 nm 工艺下,全耗尽绝缘体上硅 (fully depleted silicon-on-insulator, FD-SOI) SRAM 的多位翻转 (multi-bit upset, MBU) 错误率仅为体硅工艺 SRAM 的三分之一到五分之一^[5-6]。但受限于 FD-SOI 工艺的高昂成本,航天应用较少。

在电路级,通过抗单粒子翻转的双互锁存储单元 (dual interlocked storage cell, DICE) 触发器或位交错技术进行加固。DICE 触发器和位交错技术对单位翻转具有很好的预防和纠错能力。但随着翻转阈值减小及电荷共享效应的出现,为纠正更多错误,位交错复杂度大增。同时,由于敏感节点距离靠近, DICE 触发器结构对于小尺寸电路也会无效,依然会产生多位翻转^[7]。

在体系结构级,检纠错码可实现对错误纠正而无需改变器件的工艺和底层电路,是这一层级常用的单粒子翻转错误减缓技术。扩展汉明码可检两位错误,纠正一位错误。奇偶校验可检测多位错误,但不能纠正。这两种编码因其实现电路简单且延时可控,被广泛用于航天 SRAM 接口,以及宇航处理器内 cache 的检纠错。如 AT697F、BM8303、LS1E300 等宇航处理器的存储器访问接口均采用汉明码和奇偶校验码。然而,随着集成电路发展到纳米级,以上编码均不能解决 SRAM 或 CPU 内部 cache 在轨出现的多位翻转问题。

文献[8-9]分别针对 3 bit 连续错误和 3 bit 随机错误,提出了 MBEC 和 SEC-DAEC-TAEC 码。这两种编码在纠错能力提升的同时,冗余位数和传统检二纠一码保持一致。文献[10]采用 RM(2,5) 对 FPGA 的块 RAM 进行加固,可检测和纠正不超过

3 bit 的错误。文献[11]提出的矩阵码 FUEC-QUAEC 最高可纠正 4 bit 连续错误。文献[12]提出了两种低冗余的 SRAM 多位翻转纠错码,在 32 bit 信息位情况下,其码率(信息位长度/码字长度)分别高达 72.7% 和 78.0%,最多可纠正 4 bit 连续错误。文献[13]提出一种扩展汉明码和奇偶校验的组合码(CLC 码),该编码对 4、5 bit 均有一定的纠错能力。文献[14]提出了改进的矩阵码 FUEC-QUAEC,该编码对连续 4 bit 错误实现 100% 纠正。此外,对连续 5 bit 错误纠正率达到 60.8%。但以上研究对单个字内的连续 5 bit 错误不能实现 100% 纠正。文献[15-16]介绍了可单拍内实现的 RS(8, 4, 4) 码,该码可纠正信息域 5 bit 连续错误,但其码率仅为 50%,冗余资源占用率较高。然而,为达到较小的译码延时,其中仅实现了信息域或校验域的两个符号错误的纠正,对于同时发生在信息域和校验域两个符号的错误无法纠正。

针对以上问题,本文通过研究纳米级 SRAM 多位翻转的特性,基于检纠错码的方法,在较小延时与资源占用率的情况下实现对纳米级 SRAM 的多位翻转的检测和纠正。

1 纳米级 SRAM 多位翻转特性

工艺尺寸的减小导致产生单粒子翻转所需的临界电荷减少,使得纳米级 SRAM 的存储单元更易翻转。另外,伴随粒子入射过程中产生的电荷共享效应也是导致多位翻转的一个主要原因^[17]。研究人员通过重离子试验测试获得 90 和 65 nm 的多位翻转图样,如图 1^[17-18]所示。其中,深色为翻转位。

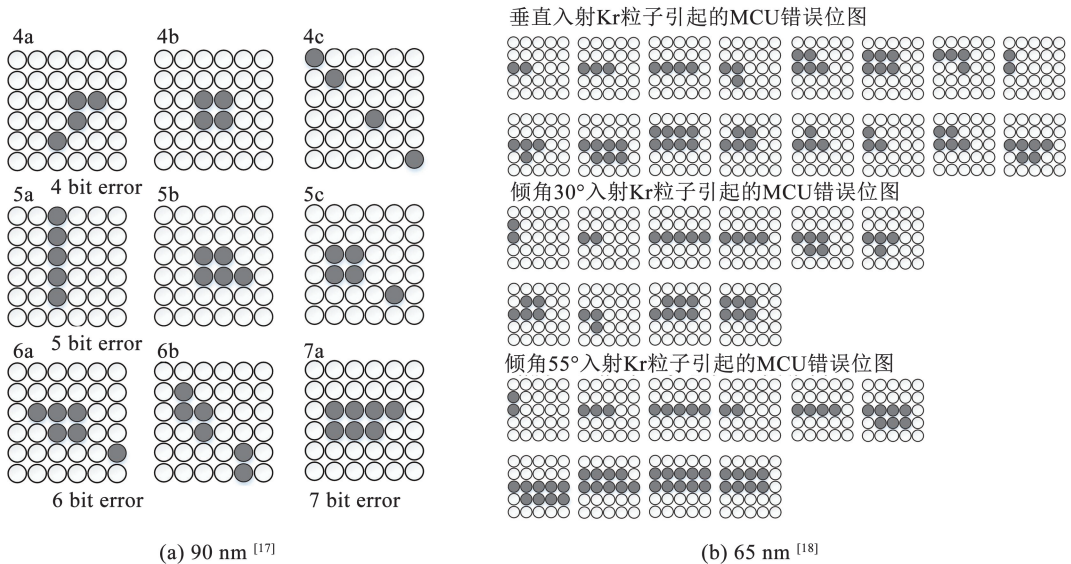


图 1 90 和 65 nm SRAM 多位翻转图样

Fig. 1 MCU patterns in 90 and 65 nm SRAM

依据惯例,定义 MCU (multi-cell upset) 为一次单粒子事件中产生的多个单元的翻转, MBU 为一次单粒子事件中在一个字内产生的多位翻转。从试验结果可知,纳米级 SRAM 多位翻转的特性如下:

1) 多位翻转成簇状排列。

2) 单行(列)不大于 5 bit,同时可能出现多行翻转的情况,累计多达 10 bit。

由以上特性可知,针对当前纳米级 SRAM,需要能够检测和纠正单个字内至少 5 bit 连续错误才可以保证单次访问 SRAM 数据的正确性。因此,本文将针对 5 bit 连续翻转错误,研究基于 RS(12,8,4) 码的检纠错编译码实现方法。

2 RS(12,8,4) 编译码的 FPGA 实现

RS 码是由 Reed 和 Solomon 于 1960 年提出的建立在伽罗华域 $GF(2^m)$ 上的 2^m 进制 BCH 码,被广泛用于通信和数据存储系统的差错控制,其原理参见文献[19]。本文在传统编译码实现方式上进行优化改进,研究可用于纠正宇航用纳米级 SRAM 多位翻转的 RS(12,8,4) 码。该码字由 12 个符号构成,每个符号 4 bit,码字总长 48 bit,信息域 32 bit,冗余校验域 16 bit。

2.1 并行编码实现

依据 RS 编译码原理,在 $GF(2^4)$ 上实现 k 个信息符号,可纠正 t 个符号的 RS($n, k, 2t$) 码,每个符号 4 bit。设信息多项式 $m(x) = m_{k-1}x^{k-1} + \dots + m_1x + m_0$,若取生成多项式为 $g(x) = (x + \alpha)(x + \alpha^2)(x + \alpha^3) \dots (x + \alpha^{2t})$,其系统码多项式为 $c(x) = m(x)x^{n-k} + r(x)$,由此可得冗余校验多项式为 $r(x) = [m(x)x^{n-k}] \bmod(g(x))$ 。

RS 编码电路是 k 级移位寄存器的乘加过程,需要 k 个时钟周期。在通信传输领域,可接受一定的延时,但在数字控制领域,CPU 访问控制延时越小,访问速率越高。因此,需要将编译码延时控制在一个时钟周期内。为减少编码延时,采用并行编码

实现,将码多项式表示为

$$C = MG = M[I|P] \quad (1)$$

$$P = \begin{bmatrix} x^{n-1} & \bmod(g(x)) \\ \vdots & \vdots \\ x^{n-k} & \bmod(g(x)) \end{bmatrix} \quad (2)$$

式中: M 为信息多项式的系数矩阵,其元素均为伽罗华域 $GF(2^4)$ 中的元素; G 为生成矩阵; I 为 k 阶单位矩阵; P 为生成冗余校验位的校验矩阵。

并行编码主要通过两步实现。1) 预先计算 P 值并存入 FPGA,供编码过程中直接调用。2) 根据伽罗华域的乘加运算法则,将乘加运算简化为加法和异或两种运算,并利用 FPGA 的并发特性,直接调用 P 矩阵并行运算。通过以上两步,将编码模块由原来的 k 级移位寄存器及乘加运算在两级 LUT 延时范围内实现,实现单时钟节拍的编码输出。

对于 RS(12,8,4) 码,根据本原多项式 $p(x) = x^4 + x + 1$,可生成伽罗华域 $GF(2^4)$ 上的所有 16 个元素。采用生成多项式 $g(x) = (x + \alpha)(x + \alpha^2)(x + \alpha^3)(x + \alpha^4)$,并通过长除法运算可求得 P 为

$$P^T = \begin{bmatrix} \alpha^6 & \alpha^{10} & \alpha^{13} & \alpha^3 & \alpha^{10} & \alpha^1 & \alpha^1 & \alpha^{13} \\ \alpha^{13} & \alpha^{14} & \alpha^{14} & \alpha^{12} & \alpha^{13} & \alpha^{11} & \alpha^7 & \alpha^6 \\ \alpha^3 & \alpha^{12} & \alpha^9 & \alpha^4 & \alpha^{13} & \alpha^5 & \alpha^8 & \alpha^3 \\ \alpha^5 & \alpha^8 & \alpha^{13} & \alpha^5 & \alpha^{11} & \alpha^{11} & \alpha^8 & \alpha^{10} \end{bmatrix} \quad (3)$$

信息 $M(x)$ 与 P 乘加运算后便可输出冗余信息 $R(x)$,完成编码。RS(12,8,4) 码符号排列如图 2 所示。其中,SY0 ~ SY11 表示编码后的一个符号。

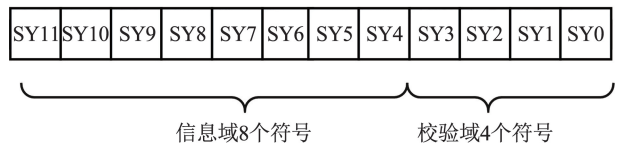


图 2 RS(12,8,4) 码符号排列

Fig. 2 Symbol arrangement of RS(12,8,4) code

2.2 并行译码实现

RS 码传统译码过程共分为 4 个步骤^[19],如图 3 所示。

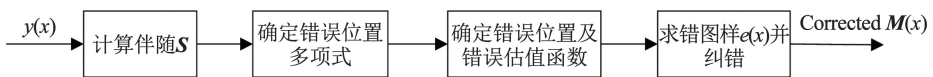


图 3 RS 码传统译码过程

Fig. 3 Traditional decoding process for RS code

传统译码采用 Berlekamp(BM)或欧几里得算法计算错误位置多项式,通过钱搜索算法求错误位置,采用 Forney 公式求取错误估值,并获得错误图样,最后纠正输出。欧几里得算法相比 BM 算法已简化许多,但仍然需要进行多次的除法迭代。另外,欧几

里得算法还需要使用除法运算所得的余式 $Z_0^{(p)}(x)$ 计算错误估值。传统译码过程均为串行操作,且除法迭代复杂并不适合 FPGA 运算。为方便 FPGA 并行计算,缩短译码延时,本文针对译码过程进行以下 5 项改进。

改进 1 预先存储用于计算伴随式 S 及校验码差值 ΔS 的矩阵到 FPGA, 简化计算。

改进 2 根据 ΔS 将错误分为无误、错误均在校验域、错误在校验域和信息域 3 类。

改进 3 推导错误位置多项式的一般表达式, 免去迭代运算, 便于 FPGA 并行计算。

BCH 码的译码过程即为求解主方程 $\sigma(x)S(x) \equiv Z_0(x) \pmod{x^{2t}}$ 的过程。欧几里得算法的核心是通过除法迭代的方法求 $S(x)$ 和 x^{2t} 的最大公因式, 从而求得错误位置多项式 $\sigma(x)$ 和错误估值 $Z_0(x)$ 。 $\sigma(x)$ 的根的倒数即为错误位置。

按照欧几里得译码算法^[19], 可以将除法余式表达为

$$Z_0^{(i)}(x) = \gamma^{(i)}(x)x^{2t} + \sigma^{(i)}(x)S(x) \quad (4)$$

递推关系为:

$$Z_0^{(i)}(x) = Z_0^{(i-2)}(x) - q_i(x)Z_0^{(i-1)}(x) \quad (5)$$

$$\sigma^{(i)}(x) = \sigma^{(i-2)}(x) - q_i(x)\sigma^{(i-1)}(x) \quad (6)$$

$$\gamma^{(i)}(x) = \gamma^{(i-2)}(x) - q_i(x)\gamma^{(i-1)}(x) \quad (7)$$

初值 $Z_0^{(-1)}(x) = x^{2t}, Z_0^{(0)}(x) = S(x), \gamma^{(-1)}(x) = \sigma^{(0)}(x) = 1, \gamma^{(0)}(x) = \sigma^{(-1)}(x) = 0$ 。

在第 i 次除法迭代中, 用 $Z_0^{(i-2)}(x)$ 除以 $Z_0^{(i-1)}(x)$, 得到商式 $q_i(x)$ 和余式 $Z_0^{(i)}(x)$ 。利用递推关系计算 $\sigma^{(i)}(x)$ 。当迭代进行到第 ρ 步时, 若满足式(8), 则迭代停止。

$$\deg Z_0^{(\rho)}(x) < \deg \sigma^{(\rho)}(x) \leq t \quad (8)$$

式中 $\deg Z_0^{(\rho)}(x)$ 为 $Z_0^{(\rho)}(x)$ 的次数。此时, $Z_0^{(i)}(x) = Z_0^{(\rho)}(x), \sigma(x) = \sigma^{(\rho)}(x)$ 。

若错误个数小于等于 t , 则一定存在满足式(8)的 ρ , 且 $\rho \leq t$ 。由此便可得到错误位置多项式 $\sigma(x) = \sigma^{(\rho)}(x)$ 。

对于 RS(12, 8, 4) 码, $t = 2$ 。因此, 当最多有两个符号错误时, 只需进行最多两次除法迭代即可求得 $\sigma^{(\rho)}(x)$, 计算过程如下:

当 $S_4 \neq 0$ 时, 执行以下迭代运算。

1) 令 $Z_0^{(-1)}(x) = x^4, Z_0^{(0)}(x) = S(x) = S_4x^3 + S_3x^2 + S_2x + S_1$ 。

2) $Z_0^{(-1)}(x)$ 除以 $Z_0^{(0)}(x)$ 得:

$$\sigma^{(1)}(x) = q_1(x) = \left(\frac{1}{S_4}x + \frac{S_3}{S_4}\right) \quad (9)$$

$$Z_0^{(1)}(x) = x^4 - q_1(x)S(x) \quad (10)$$

故, $Z_0^{(1)}(x) = \gamma^{(1)}(x) = \left(\frac{S_2}{S_4} + \frac{S_3^2}{S_4^2}\right)x^2 + \left(\frac{S_1}{S_4} + \frac{S_2S_3}{S_4^2}\right)x + \frac{S_1S_3}{S_4^2}$, 不满足式(8), 继续迭代。

3) $Z_0^{(0)}(x)$ 除以 $Z_0^{(1)}(x)$ 得

$$\sigma^{(2)}(x) = \left(\frac{1}{S_4}x + \frac{S_3}{S_4}\right) \times$$

$$\left[\frac{S_4^3}{S_2S_4 + S_3^2}x + \frac{S_4^2S_3(S_2S_4 + S_3^2) + S_4^3(S_1S_3 + S_2^2)}{(S_2S_4 + S_3^2)^2} \right] + 1 \quad (11)$$

由于本算法不需要 $Z_0^{(2)}(x)$ 求错误估值, 因此免去 $Z_0^{(2)}(x)$ 的计算。

式(11)即为 RS(12, 8, 4) 码的错误位置多项式的一般表达式。求式(11)的根等价于求式(12)的根。本文定义式(12)为等价错误位置多项式。

$$\sigma(x) = (S_4S_2 + S_3^2)x^2 + (S_4S_1 + S_2S_3)x + (S_3S_1 + S_2^2) \quad (12)$$

当 $S_4 = 0$ 时, 重复迭代 1)、2)、3) 可推导出等价错误位置多项式为

$$\sigma(x) = S_3^2x^2 + (S_2S_3)x + (S_3S_1 + S_2^2) \quad (13)$$

该结果与式(12)中 $S_4 = 0$ 所得结果一致。

因此, 可将错误位置多项式简化为 $\sigma(x) = Ax^2 + Bx + C$, 其中 A, B, C 为伴随矩阵 S 的函数。采用 FPGA 的译码实现时, A, B, C 的值均可通过伽罗华域中两级乘加运算并行计算获得, 省去了迭代计算过程, 从而减少了运算延时。

改进 4 计算错误图样的一般表达式, 将错误位置代入后直接求取错误图样, 免去计算余式 $Z_0^{(\rho)}(x)$ 后再运算的过程。其推导过程如下:

设编码后的系统码多项式为 $c(x) = c_{11}x^{11} + \dots + c_1x + c_0$, 错误多项式为 $e(x) = e_{11}x^{11} + \dots + e_1x + e_0$, 传输读回的数据为 $y(x) = e(x) + c(x)$, 其系数可表示为

$$y_i = c_i + e_i, i = 0, 1, 2, \dots, 11 \quad (14)$$

重新计算 $y(x)$ 中信息位的校验值和冗余校验值的差值, 可得 ΔS_j , 可表示为

$$\Delta S_j = \sum_{i=0}^{11} y_i H_{(11-i)j} = \sum_{i=0}^{11} c_i H_{(11-i)j} + \sum_{i=0}^{11} e_i H_{(11-i)j} \quad (15)$$

式中: $j = 0, 1, 2, 3; H^T = [P^T | I_{4 \times 4}]$, I 为 4 阶单位矩阵。

根据错误分类, 考虑以下 4 种情况:

1) 有两个错误且均在信息域, 分别设为 e_m, e_n 。根据矩阵乘法, 将其代入式(15)得

$$\begin{cases} \Delta S_0 = e_m P_{m0} + e_n P_{n0} \\ \Delta S_1 = e_m P_{m1} + e_n P_{n1} \\ \Delta S_2 = e_m P_{m2} + e_n P_{n2} \\ \Delta S_3 = e_m P_{m3} + e_n P_{n3} \end{cases} \quad (16)$$

两个未知数 4 个方程, 任意取两个可求 e_m, e_n 。

联合前两个方程求解得:

$$e_m = \frac{\Delta S_0 + e_n P_{m0}}{P_{m0}}, e_n = \frac{\Delta S_0 P_{m1} + \Delta S_1 P_{m0}}{P_{n0} P_{m1} + P_{n1} P_{m0}}$$

2) 有两个错误。一个在信息域, 另一个在校验域, 分别设为 e_m, e_r , 其中 $r = 0, 1, 2, 3$ 。将其代入式(15), 得到两个方程组:

$$r = 3 \text{ 时, } \begin{cases} \Delta S_0 = e_m P_{m0} + e_r \\ \Delta S_1 = e_m P_{m1} \end{cases} \quad (17)$$

$$r \neq 3 \text{ 时, } \begin{cases} \Delta S_0 = e_m P_{m0} \\ \Delta S_j = e_m P_{mj} + e_j \end{cases} \quad (18)$$

解方程组可得 $e_m = \begin{cases} \frac{\Delta S_1}{P_{m1}}, r = 3 \\ \frac{\Delta S_0}{P_{m0}}, r \neq 3 \end{cases}$

e_r 为冗余校验域错误图样, 因此无需计算。

3) 仅有一个错误且在信息域时, 可求解

$$e_m = \frac{\Delta S_3}{P_{m3}}$$

4) 没有错误或所有错误均在冗余域。此时无需计算, 直接输出 32 bit 信息位。

通过以上方法优化错误多项式及错误估值的求解, 实现 RS(12, 8, 4) 的 FPGA 并行化译码, 减少串行译码带来的延时。

在航天工程中, 信息域和校验域分别存储在两个不同的 SRAM 片内, 其物理位置不相邻。由单粒子翻转特性可知, 一次单粒子事件导致 SRAM 一个字内的翻转位呈簇状排列, 即翻转位相邻。因此, 在

实际工程应用中, 信息域和校验域不会同时出错。为进一步减小译码延时, 可针对此特殊情况进行第 5 项改进。

改进 5 免去错误的精确位置计算, 简化错误位置和错误图样的计算逻辑。

如图 2 所示, RS(12, 8, 4) 系统码由 8 个信息符号和 4 个校验符号组成。在传统译码中, 计算错误在 12 个符号中的精确位置后方可根据错误估值计算错误图样。本文在前 4 项改进的基础上, 将错误位置划分为 [SY0, SY1, SY2, SY3]、[SY4, SY5, SY6, SY7]、[SY6, SY7, SY8, SY9]、[SY8, SY9, SY10, SY11] 4 个区域。根据错误相邻的规则, 错误一定位于这 4 个区域之一。

对于在校验域 [SY0, SY1, SY2, SY3] 的错误无需译码。其余 3 类, 每类均有 4 个符号。设可能的错误值分别为 e_m, e_n, e_k, e_l , 错误图样采用式(19)求解, 其推导过程与式(16)相同。

$$\begin{cases} \Delta S_0 = e_m P_{m0} + e_n P_{n0} + e_k P_{k0} + e_l P_{l0} \\ \Delta S_1 = e_m P_{m1} + e_n P_{n1} + e_k P_{k1} + e_l P_{l1} \\ \Delta S_2 = e_m P_{m2} + e_n P_{n2} + e_k P_{k2} + e_l P_{l2} \\ \Delta S_3 = e_m P_{m3} + e_n P_{n3} + e_k P_{k3} + e_l P_{l3} \end{cases} \quad (19)$$

由 $m, n, k, t \in \{4, 5, 6, 7\}$ 或 $m, n, k, t \in \{6, 7, 8, 9\}$ 或 $m, n, k, t \in \{8, 9, 10, 11\}$ 代入式(19)得到 3 组方程。每组均为 4 个方程 4 个未知数, 可求唯一解。由于免去错误的精确位置计算, 译码延时被进一步减小。优化后的译码模块功能结构如图 4 所示。

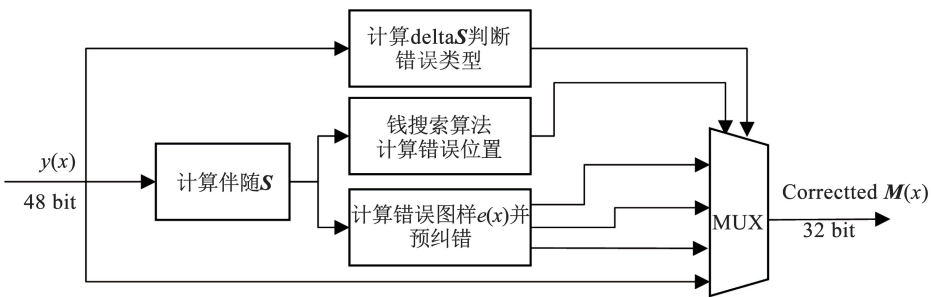


图 4 优化后的译码模块功能结构

Fig. 4 Functional structure of optimized decoding module

为方便后续比较, 将经过前 4 项改进后实现的 FPGA 译码模块命名为 RS12_8_4, 经所有 5 项改进后实现的译码模块命名为 RS12_8_4_sp。

2.3 纠错能力及性能

如前所述, RS(12, 8, 4) 码由 12 个符号组成, 高 32 bit 为信息域, 低 16 bit 为冗余校验域。图 5 为错误模式的 16 进制表示, 共计 78 类。其中, F 表示一个错误符号, 代表了一个符号内任意位翻转的情况 (共计 15 种)。

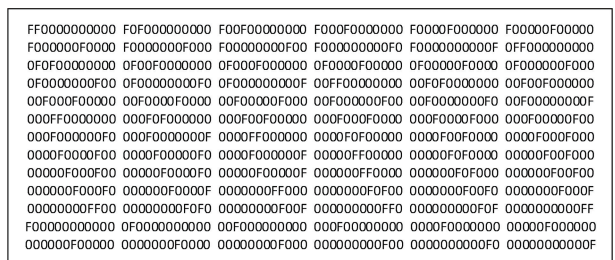


图 5 错误模式的 16 进制表示

Fig. 5 Error data mode in Hexadecimal

采用 Modelsim 仿真注入错误模式数据,对所有 78 类错误进行了验证。验证结果显示,译码后数据和原始数据一致,编码和译码正确无误。

由于单个字内的 5 bit 连续错误一定分布在两个符号内,属于 RS(12,8,4)的检纠错范围。因此,RS(12,8,4)对 5 bit 连续错误可实现 100% 纠正。

与汉明码、CLC^[13]、FUEC-QUAEC^[14]相比,本文实现的 RS(12,8,4)码纠错能力提升至 5 bit 连续错误的 100% 纠正,6~8 bit 连续错误纠正率分别提高至 76.7%、52.3%、26.8%。连续错误纠正率对比如图 6 所示。另外,在综合纠错能力和码率方面,本文实现的 RS(12,8,4)码也具有一定的优势,比较结果见表 1。

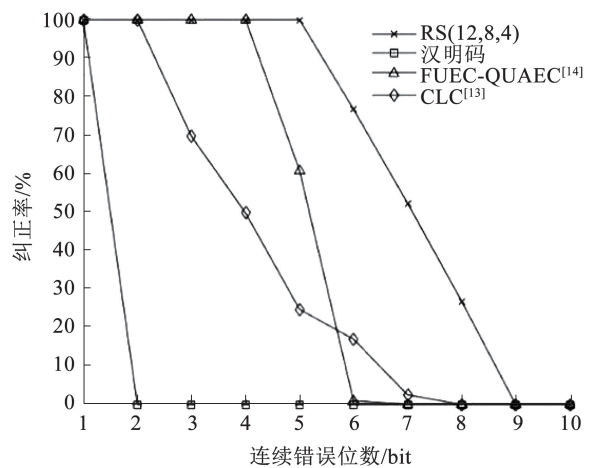


图 6 连续错误的纠正率对比

Fig. 6 Comparison of correction rates for burst error

表 1 不同类型纠错码的码率和纠错能力对比

Tab. 1 Comparison of code rates and error correction capabilities of different error correction codes

编码名称	信息位/bit	冗余位/bit	码率/%	纠错能力
汉明码	32	7	82.1	1 bit 错误纠正率 100%
MBEC ^[8]	16	7	69.6	3 bit 随机错误纠正率 100%
SEC-DAEC-TAEC ^[9]	32	7	82.1	3 bit 连续错误纠正率 100%
RM(2,5) ^[10]	16	16	50.0	3 bit 错误纠正率 100%
FUEC-QUAEC ^[11]	32	14	69.6	4 bit 连续错误纠正率 100%
Efficient 4-bit burst ECC ^[12]	32	9	78.0	4 bit 连续错误纠正率 100%
CLC ^[13]	16	24	40.0	4,5 bit 错误纠正率分别为 49.8%、24.7%
FUEC-QUAEC ^[14]	16	9	64.0	4,5 bit 错误纠正率分别为 100%、60.8%
RS(8,4,4) ^[15-16]	16	16	50.0	5 bit 连续错误纠正率 100%, 并且对 6~8 bit 错误也有一定的纠正能力
本文 RS(12,8,4)	32	16	66.7	

调用 Xilinx 自带汉明码的块 RAM 模块测试,并与采用本文实现的 RS(12,8,4)码加固的块 RAM 模块进行对比,其延时相近,约为 15 ns。针对航天应用的特殊情况,若采用改进的 RS12_8_4_sp 加固后,其延时缩短至 11.657 ns。

根据 RS12_8_4_sp 的实现原理,其无法纠正同时出现在信息域和校验域的错误。但在 32 bit 的信息域,依然对 5 bit 连续错误可 100% 纠正,6~8 bit 连续错误的纠正率分别为 77.8%、53.8%、28.0%。对校验域的任意两个符号(最多 8 bit)的错误可实现 100% 纠正。

根据文献[3],90 和 65 nm 工艺 SRAM 多位翻转比例分别为 5% 和 55%,且多位翻转均为 2~5 bit 错误,未发现 6 bit 及以上的错误。依据文献[20],65 nm SRAM 器件在轨单粒子翻转率 R 为 $1.29 \times 10^{-6}/(\text{bit} \cdot \text{d})$ 。对于 65 nm 工艺,容量为 1 Mbit 的 SRAM,若采用汉明码进行加固,其平均无不可纠正错误的运行时间约为 1.34 d,即

$\frac{1}{1\ 024 \times 1\ 024 \times R \times 0.55}$ d。若采用本文实现的方

法加固,则可纠正 SRAM 在轨运行期间产生的所有多位翻转错误,确保其长期稳定地运行。

3 应用扩展

本文提出的检纠错码实现方法可用于纳米级 SRAM 和纳米级 FPGA 中的块 RAM 加固,亦可扩展到 FPGA 或 CPU 对外部存储器的控制逻辑的加固。该加固方法可替代现有 CPU 存储控制器的检二纠一加固方法,以解决传统扩展汉明码无法纠正多位翻转的问题。同样,在访问频率要求不大于 80 MHz 的处理器 cache 中加入 RS(12,8,4)纠错码,可极大地提升处理器 cache 的纠错率。在现有的工艺尺寸下可解决宇航 CPU cache 中的多位翻转问题。

4 结论

1)通过研究纳米级 SRAM 在空间辐照环境中多位翻转特性,实现了一种可检测和纠正 5 bit 连续错误的 RS(12,8,4)码。通过优化改进传统串行编

译码,采用并行编译码的方法在单时钟节拍内完成编译码输出,降低了编码和译码的延时。

2)结合航天工程应用中信息域和校验域通常存储在不同的SRAM颗粒的情况,在确保纠正率100%的情况下,针对信息域5 bit连续错误实现的编译码可进一步降低延时。

3)采用并行编译码实现的RS(12,8,4)检纠错码具有较强的纠错能力,对更小尺寸工艺的SRAM可能出现的6~8 bit连续错误纠正率分别为77.8%、53.8%、28.0%。

4)从冗余信息占用率角度分析,与具有同等纠错能力的采用RS(8,4,4)检纠错实现方法相比,本文实现的编译码具有较高的码率,即同样的纠错能力所耗费的冗余信息更少。

参考文献

- [1]TOSAKA Y, EHARA H, IGETA M, et al. Comprehensive study of soft errors in advanced CMOS circuits with 90/130 nm technology [C]//IEEE International Electron Devices Meeting. San Francisco: IEEE, 2004: 941. DOI: 10.1109/IEDM/2004.1419339
- [2]SEIFERT N, SLANKARD P, KIRSCH M, et al. Radiation-induced soft error rates of advanced CMOS bulk devices[C]//44th Annual IEEE International Reliability Physics Symposium. San Jose: IEEE, 2006: 217. DOI: 10.1109/RELPHY/2006.251220
- [3]MANIATAKOS M, MICHAEL M K, MAKRIS Y. Multiple-bit upset protection in microprocessor memory arrays using vulnerability-based parity optimization and interleaving[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2015, 23(11): 2447. DOI: 10.1109/TVLSI.2014.2365032
- [4]KATO T, YAMAZAKI T, SAITO N, et al. Neutron-induced multiple-cell upset in 20-nm bulk SRAM: angular sensitivity and impact of multi-well potential perturbation[J]. IEEE Transactions on Nuclear Science, 2019, 66(7): 1381. DOI: 10.1109/TNS.2019.2900629
- [5]CANNON E H, GORDON M S, HEIDEL D F, et al. Multi-bit upset in 65 nm SOI SRAMs[C]//46th Annual 2008 IEEE International Reliability Physics Symposium. Phoenix: IEEE, 2008: 195. DOI: 10.1109/RELPHY.2008.4558885
- [6]GASIOT G, SOUSSAN D, AUTRAN J, et al. Muons and thermal neutrons SEU characterization of 28 nm UTBB FD-SOI and bulk eSRAMs [C]//2015 IEEE International Reliability Physics Symposium. Monterey: IEEE, 2015. DOI: 10.1109/IRPS.2015.7112677
- [7]周昕杰,张黎莉,王韬,等.基于28 nm体硅CMOS工艺位交型抗辐射触发器加固结构[J].固体电子学研究与进展,2022,42(4):306
ZHOU Xinjie, ZHANG Lili, WANG Tao, et al. A radiation hardened flip-flop with interlaced critical nodes in 28 nm CMOS technology[J]. Research & Progress of SSE, 2022, 42(4): 306. DOI:10.19623/j.cnki.rpsse.2022.04.011
- [8]GAMINI S, YELAMANCHILI A. Random error detection and correction codes for multiple bits [C]//2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT). Bangalore: IEEE, 2022: 1. DOI: 10.1109/GCAT55367.2022.9972031
- [9]SAIZ-ADALID L, 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 (VLSI) Systems, 2015, 23(10): 2332. DOI: 10.1109/TVLSI.2014.2357476
- [10]钟敏. SRAM型FPGA的SEU容错技术研究[D].北京:中国科学院大学,2021
ZHONG Min. Research on SEU fault tolerance technology of SRAM FPGA[D]. Beijing: University of Chinese Academy of Sciences, 2021. DOI:10.27543/d.cnki.gkgdk.2021.000020
- [11]ARGYIDES C A, REVIRIEGO P, PRADHAN D K, et al. Matrix-based codes for adjacent error correction[J]. IEEE Transactions on Nuclear Science, 2010, 57(4): 2106. DOI: 10.1109/TNS.2010.2043265
- [12]LI Jiaqiang, XIAO Liyi, REVIRIEGO P, et al. Efficient implementations of 4-bit burst error correction for memories[J]. IEEE Transactions on Circuits and Systems II-Express Briefs, 2018, 65(12): 2037. DOI: 10.1109/TCSII.2018.2817390
- [13]CASTRO H D S, DA SILVEIRA J A N, COELHO A A P, et al. A correction code for multiple cells upset in memory devices for space applications [C]//14th IEEE International New Circuits and Systems Conference (NEWCAS). Vancouver: IEEE, 2016: 7604783. DOI: 10.1109/NEWCAS.2016.7604783
- [14]GRACIA-MORÁN J, SAIZ-ADALID L, GIL-TOMÁS G, et al. Improving error correction codes for multiple-cell upset in space applications[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2018, 26(10): 2132. DOI: 10.1109/TVLSI.2018.2837220
- [15]张颖. SRAM抗SEU加固单元设计与ECC编码实现[D].西安:西安电子科技大学,2015
ZHANG Ying. Design and implementation of SEU tolerant robust cell and ECC encoding on SRAM[D]. Xi'an: Xidian University, 2015
- [16]余艺. 差错控制编码在BRAM及固态存储系统中的设计与应用[D].成都:电子科技大学,2020
YU Yi. Design and application of error control coding in BRAM and solid-state storage systems[D]. Chengdu: University of Electronic Science and Technology of China, 2020. DOI: 10.27005/d.cnki.gdzku.2020.004433
- [17]KUMAR S, CHO M, EVERSON L, et al. Analysis of neutron-induced multi-bit upset clusters in a 14-nm flip-flop array[J]. IEEE Transactions on Nuclear Science, 2019, 66(6): 918. DOI: 10.1109/TNS.2019.2911540
- [18]赵元富,王亮,岳素格,等.纳米级CMOS集成电路的单粒子效应及其加固技术[J].电子学报,2018,46(10):2511
ZHAO Yuanfu, WANG Liang, YUE Suge, et al. Single event effect and its hardening technique in nano-scale CMOS integrated circuits[J]. Acta Electronica Sinica, 2018, 46(10): 2511. DOI: 10.3969/j.issn.0372-2112.2018.10.027
- [19]LIN S, COSTELLO D. 差错控制编码[M].2版.晏坚,何元智,潘亚汉,等译.北京:机械工业出版社,2007:156
LIN S, COSTELLO D. Error control coding[M]. 2nd ed. YAN Jian, HE Yuanzhi, PAN Yahan, et al. translate. Beijing: China Machine Press, 2007: 156
- [20]汪波,王佳,刘伟鑫,等.星载ASIC芯片单粒子效应检测及在轨翻转率预估[J].半导体检测与设备,2019,44(9):728
WANG Bo, WANG Jia, LIU Weixin, et al. Single event effect detection and on-orbit upset rate estimation for spaceborne ASIC chip[J]. Semiconductor Testing and Equipment, 2019, 44(9): 728. DOI:10.13290/j.cnki.bdtjs.2019.09.013