基于数位索引、开销近汉明的 q 元 SEC-DED 码

胡佳旭, Kenneth J. Roche

华盛顿大学, 西雅图, 华盛顿 98195-1560, 美国

2025 年 10 月 9 日

Abstract

本文提出了一种简单的 q 元线性码家族，具备单纠错-双检错能力。该码的冗余位与坐标索引的 p 进制数位直接关联（其中 $q = p$ 为素数）。对于码块长度 $n = p^r$，该构造仅需 $r + 1$ 个冗余位——即接近汉明码的开销——并支持一种基于索引的解码器。该解码器仅需单次扫描，即可根据校验信息，在常数时间内快速定位错误位置并计算出错误值。基于该原型码，我们进一步提出了两种扩展：A1 码通过移除特定冗余位提升信息率并支持变长编码；A2 码通过引入额外校验位与三重异或线性无关条件，将最小距离扩展至 4，实现单纠错-三检错。此外，我们阐述了该框架如何通过 n 重异或线性无关集加以推广，以构造距离为 $d = n + 1$ 的码，并在 $n = 5$ 时重构出了三元戈莱码——这既展示了其结构的通用性，也揭示了与最优经典编码之间意想不到的联系。

1 引言

在现代通信中，纠错编码用于检测并纠正由噪声信道引入的错误，从而确保数据传输的准确性。汉明码是一种广为人知的纠错码，能够纠正一个错误并检测两个错误，在二进制数据中具有广泛应用。沿用汉明码的思路，扩展到三进制系统，可以利用三态逻辑（0、1、2）更高的符号效率，在多级存储（如 NAND 闪存）和采用相位调制的光通信中实现更高密度的数据编码，在这些场景中二进制编码往往难以胜任。

本文研究三进制纠错码，探索其在非二进制系统中的独特性质。我们采用模运算的原理，并引入了一种新颖的结构性概念—— “逆索引成对结构”，这一结构上的改进使得在三进制框架下能够实现更高效的数据编码与纠错。在保持相同数量冗余三进制位（trits）的前提下，该新型编码方案可以多存储约 50% 的信息位。此外，我们还引入了 “三重线性无关集”（3-wise linearly independent set）的概念，以拓展最小汉明距离，从而构造出一种三进制的 SEC–TED 码。

本文结构安排如下：第二节概述了汉明码的三进制扩展，提出一个通过数位索引奇偶校验构建的三维 SEC–DED 原型码，并将该方法推广至任意素数进制。第三节引入模运算原理及纠错编码中的成对结构概念。第四节进一步探讨了成对结构在纠错中的应用，重点关注可变块长度、提升码率及扩展最小汉明距离等问题，同时给出 A1 码与 A2 码的理论证明与编码算法以验证其有效性。最后，第五节提出一种构建具有更高最小距离编码的方法，并以戈莱码作为实例加以说明。

通过本研究，我们不仅扩展了经典的汉明码，也为更广泛的非二进制纠错领域作出了贡献，同时提出了在更高维度与非二进制编码系统中的潜在研究方向。

为明确本方法的定位，我们概述推动 q 元编码发展的应用背景，随后将我们的方案与已有编码族进行对标，并提供比较性总结，再阐明本研究的具体贡献。

应用背景（为何研究 q 元编码） 多级存储与调制系统天然地运行在非二进制字符集上，例如：用于三元存储器的 $q = 3$ 进制、用于闪存/PCM 的 $q = 4/8$ 符号字母表，或编码调制中的 q 元星座图。在这些应用场景中，具备极简校验结构及基于索引的错误定位器的 SEC–DED 码，能够有效降低控制逻辑的复杂性、减少延迟并节省实现成本。

工作定位与创新性 本文提出的 SEC–DED 码族，在码长 $n = p^r$ 时仅需 $r + 1$ 个冗余位——该开销与同等码长下最优的 q 元汉明码（需 r 个校验位）非常接近。然而，与汉明码不同，本方案中的校验行直接对应于索引的 p 进制数位（外加一个全局和校验）。这一特性使得其解码器中的错误定位过程，不再是查表或代数搜索，而是一种显式的索引直接读取操作；同时，该结构也使得仅通过增加两个结构化的“分组”校验，即可实现向 SEC–TED 码的升级成为可能。与 SPC/乘积码相比，在相应码长下，本方案具有稍低的冗余度和更简洁的单遍解码器，尽管也放弃了乘积码在码距和迭代解码方面的部分优势。因此，我们将本方案定位为一个清晰的实现方案，它提供了一条通往 $d = 4$ 的模块化路径，而非宣称发现了一个新的最优码族。

简要对比

表 1 总结了本文方案与三种基线方案的校验开销、解码工作量及双错误处理行为的对比。(此处 N 表示总码长，R 表示冗余位数量，所列常数为指示性值；具体成本取决于实现细节。)

表 1: SEC–DED 方案与实现成本概览。示例：设 $q = p = 3$，r = 5：素数元-N=243; q 元汉明码-N = 121; SPC×SPC-$N = 256$(当 $m = 16$)。

Family	N	R	解码工作量	汉明距离：xEC–YED
素数元通用纠错码	p^r	r+1	单遍计算校正子；显式索引定位器 (O(1))	汉明距离 3：SEC–DED
q 元汉明	$\frac{q^r-1}{q-1}$	r	单遍计算校正子；代数/查表定位器 (O(1))	汉明距离 3：SEC–DED
SPC × SPC	m^2	$2m-1$	两轮计算（行与列校验）；坐标化定位器	汉明距离 4：SEC–TED

主要贡献

(i) 提出了一种数位索引的 q 元 SEC–DED 码，其在码长 $N = p^r$ 时仅需 $r + 1$ 个校验位，并提供一个显式的、基于索引的解码器。
(ii) 通过引入两个分组和校验与一个 3-wise 独立条件，构造了一个模块化的 SEC–TED 码变体（三元）。
(iii) 给出了形式化的校验矩阵描述（H 与 H'），并完成了其码距（$d = 3$ 与 $d \geq 4$）的理论证明。

2 原型三元单纠双检码

本研究主要涉及一种基于汉明码思路的三进制纠错码。我们在此简要介绍其最初版本，即一个 $[27, 23, 3]_3$ 码，它能够纠正单个错误，并检测最多两个错误。

2.1 编码构筑与冗余设计

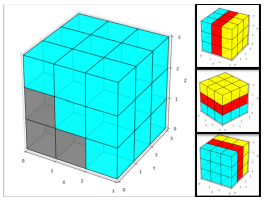


图 1: 三元汉明码的正方体布局示例。蓝色区域代表系数 0，红色对应 1，黄色对应 2。

立方体中剩余的 23 个位置作为信息位，其取值为 $-1, 0, +1$（等价于 $0, 1, 2$）。下面给出各冗余位的具体赋值规则：（其中 $*2$ 在模 3 下等价于 $* - 1$）

- v_{x_0} 位于 $(0, 0, 1)$: 使得 $\sum v_{(x_2,x_1,1)} + 2 \cdot \sum v_{(x_2,x_1,2)} \equiv 0 \pmod 3$.
- v_{x_1} 位于 $(0, 1, 0)$: 使得 $\sum v_{(x_2,1,x_0)} + 2 \cdot \sum v_{(x_2,2,x_0)} \equiv 0 \pmod 3$.
- v_{x_2} 位于 $(1, 0, 0)$: 使得 $\sum v_{(1,x_1,x_0)} + 2 \cdot \sum v_{(2,x_1,x_0)} \equiv 0 \pmod 3$.
- v_{mod3} 位于 $(0, 0, 0)$: 使得 $\sum v_{(x_2,x_1,x_0)} \equiv 0 \pmod 3$.

我们将 27 个位置排列成一个 $3 \times 3 \times 3$ 的立方体，或者看作是三张 3×3 的二维矩阵，共包含 27 个三进制位 (trits)。其中有 4 个冗余位 $v_{x_0}, v_{x_1}, v_{x_2}, v_{mod3}$ 用于纠错，分别位于位置 $(0, 0, 0)$、$(0, 0, 1)$、$(0, 1, 0)$

和 $(1,0,0)$，它们对应于第 0、1、3 和 9 个位置。当四个冗余位的取值确定后，我们完成了该纠错码的编码。注：后文将说明，该方法等价于计算所有信息位的“值 × 索引”之和，并在模 3 下使该加权和为 0，通过设置冗余位的值来实现这一目标。

2.2 示例

假设我们希望以此方法对一个长度为 23 的三元信息进行编码：**20,111,020,010,201,200,120,012**。

Algorithm 1: 原型码编码流程实例（$-1 \equiv 2$）：**20,111,020,010,201,200,120,012**

1: **初始化：** 将三进制信息按行优先顺序填入矩阵中，跳过索引为 0 以及 3^n（$n \in \mathbf{N}_0$）的格子（本例中为索引 0、1、3、9）。

$$\begin{bmatrix} v_{mod3} & v_{x_0} & -1 \\ v_{x_1} & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} v_{x_2} & -1 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix}$$

2: **计算并设置** v_{x_2}：$1 \cdot (v_{x_2} + (-1) + 1 + (-1) + 1) + 2 \cdot (-1 + 1 + (-1) + 1 + (-1)) = v_{x_2} - 2$

$$\begin{bmatrix} v_{mod3} & v_{x_0} & -1 \\ v_{x_1} & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} v_{x_2} = -1 & -1 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix}$$

为了满足 $v_{x_2} - 2 \equiv 0 \pmod 3$，根据 $-1 - 2 = -3 \equiv 0 \pmod 3$，解得 $v_{x_2} = -1$。

3: **计算并设置** v_{x_1}：$1 \cdot (v_{x_1} + 1 + 1 + 1 + (-1)) + 2 \cdot (1 + 1 + (-1) + 1 + (-1)) = v_{x_1} + 6$

$$\begin{bmatrix} v_{mod3} & v_{x_0} & -1 \\ v_{x_1} = 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix}$$

为了满足 $v_{x_1} + 6 \equiv 0 \pmod 3$，根据 $0 + 6 \equiv 6 \equiv 0 \pmod 3$，解得 $v_{x_1} = 0$。

4: **计算并设置** v_{x_0}：$1 \cdot (v_{x_0} + 1 + (-1) + 1 + (-1) + 1) + 2 \cdot (-1 + 1 + 1 + (-1)) = v_{x_0} + 1$

$$\begin{bmatrix} v_{mod3} & v_{x_0} = -1 & -1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix}$$

为了满足 $v_{x_0} + 1 \equiv 0 \pmod 3$，根据 $-1 + 1 \equiv 0 \pmod 3$，解得 $v_{x_0} = -1$。

5: **计算并设置** v_{mod3}：$\sum v_{(x_2,x_1,x_0)} = v_{mod3} - 1$

$$\begin{bmatrix} v_{mod3} = 1 & -1 & -1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix}$$

为了满足 $v_{mod3} - 1 \equiv 0 \pmod 3$，根据 $1 + -1 \equiv 0 \pmod 3$，解得 $v_{mod3} = 1$。

结语： 这样我们就得到了一个三元纠错码：**122,001,110,220,010,201,100,120,012**，能够纠正至多一个错误。当出现两个错误时，该码能够检测到错误的存在，但会误判为单个错误并给出错误的定位。因此，我们称它为单错误纠正、双错误检测（SEC-DED）码，其最小汉明距离为 3。

我们通过人为注入错误(将第八个三进制位从 1 改为 2)来测试解码器,接收序列变为 **122,001,120,220,**$\cdots$**,012**。下面是对该错误编码进行解码、纠正，并恢复原始信息的过程。

$$\begin{bmatrix} \mathbf{1} & \mathbf{-1} & -1 \\ \mathbf{0} & 0 & 1 \\ 1 & 1 \to -1 & 0 \end{bmatrix} \quad \begin{bmatrix} \mathbf{-1} & -1 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix}$$

注：将 *021* 上的 *1* 转换成*-1* 来模拟噪声干扰。

Algorithm 2: 原型码解码实例：**122,001,120,220,010,201,100,120,012**

1: 设：错误位于 $E=(x_2,x_1,x_0)$, 错误造成的改变为 Δv_E(只考虑最多一位错误的情况)

2: **检查 v_{mod3} 对应区域来判断 Δv_E:** $\sum v_{(x_2,x_1,x_0)} = -2 \equiv 1 \pmod 3$

因此，$\Delta v_E = +1 \mod 3 \equiv -2 \mod 3$ 存在三种可能情况：

- $\sum v_{(x_2,x_1,x_0)}$ +1 $\equiv 1 \pmod 3$ 由值变化 $-1 \to 0$ 或 $0 \to 1$ 引起。
- $\sum v_{(x_2,x_1,x_0)}$ -2 $\equiv 1 \pmod 3$ 由值变化 $1 \to -1$ 引起
- 存在两个或更多错误，但因已假设最多 1 个错误，故予以排除。

3: **检查 v_{x_2} 对应区域，判断 E 的 x_2 坐标:** $\sum v_{(1,x_1,x_0)} + 2 \cdot \sum v_{(1,x_1,x_0)} \equiv 0 \pmod 3$

错误并没有对 v_{x_2} 区域和产生影响。因此，错误位的 x_2 应为 0，$(0,x_1,x_0)$。

4: **检查 v_{x_1} 对应区域，判断 E 的 x_1 坐标:** $\sum v_{(x_2,1,x_0)} + 2 \cdot \sum v_{(x_2,2,x_0)} \equiv 2 \pmod 3$

根据之前冗余位 x_1 的值的设计，定义 A,B：$\sum v_{(x_2,1,x_0)} = A$ 与 $\sum v_{(x_2,2,x_0)} = B$，则有

$$(A + 2 \cdot B) \equiv 0 \pmod 3$$

根据错误位的 x_1 坐标和值，分为以下四种情况:

- $\big((A+2) + 2 \cdot B\big) \equiv 2 \pmod 3$
- $\big(A + 2 \cdot (B+1)\big) \equiv 2 \pmod 3$
- $\big((A-1) + 2 \cdot B\big) \equiv 2 \pmod 3$
- $\big(A + 2 \cdot (B-2)\big) \equiv 2 \pmod 3$

在步骤 3 中 v_{mod3} 的部分，我们得知错误的差值只能为 +1 或者-2，因此我们可以推断错误位 E 位于 B 对应的区域，即 $E=(x_2,2,x_0)$。结合步骤 4，$E=(0,2,x_0)$.

5: **检查 v_{x_0} 对应区域，判断 E 的 x_0 坐标:** $\sum v_{(x_2,x_1,1)} + 2 \cdot \sum v_{(x_2,x_1,2)} \equiv 1 \pmod 3$

根据之前冗余位 x_0 的值的设计，重新定义 A,B：$\sum v_{(x_2,x_1,1)} = A$ 与 $\sum v_{(x_2,x_1,2)} = B$，则有

$$(A + 2 \cdot B) \equiv 0 \pmod 3$$

根据错误位的 x_0 坐标和值，分为以下四种情况:

- $\big((A+1) + 2 \cdot B\big) \equiv 1 \pmod 3$
- $\big((A-2) + 2 \cdot B\big) \equiv 1 \pmod 3$
- $\big(A + 2 \cdot (B+2)\big) \equiv 1 \pmod 3$
- $\big(A + 2 \cdot (B-1)\big) \equiv 1 \pmod 3$

在步骤 3 中 v_{mod3} 的部分，我们得知错误的差值只能为 +1 或者-2，因此我们可以推断错误位 E 位于 A 对应的区域，即 $E=(x_2,x_1,1)$。结合步骤 4 和 5，$E=(0,2,1)$。

6: **订正:**

$$\begin{bmatrix} \mathbf{1} & \mathbf{-1} & -1 \\ \mathbf{0} & 0 & 1 \\ 1 & +1 & 0 \end{bmatrix} \quad \begin{bmatrix} \mathbf{-1} & -1 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix}$$

错误的 $v_{(0,2,1)} = -1$，回看步骤 3, 这是 $1 \to -1$ 的情况，所以原本的值为 1。

最终我们订正了纠错码得到 **12**2,**0**01,110,**2**20,010,201,200,120,012。随后移除所有的冗余位，我们得到原本的信息 **20,111,020,010,201,200,120,012**，与我们在 algorithm 1 最开始挑选的信息一致。

2.3 适用于任意素数进制的 SEC–DED 编码方法

该方法不仅适用于三元码，也适用于所有素数进制的 x 进制编码。使用 $n+1$ 个冗余位，该方法可以在总共 x^n 个位置中存储 $x^n - n - 1$ 个位信息，从而实现对单个错误的纠正以及对最多两个错误的检测。由此构成一个线性码，其形式为 $[x^n, x^n - n - 1, 3]_x$，或者等价地表示为 $[x^{n-1}, x^{n-1} - n, 3]_x$。

2.3.1 通用编码描述

Algorithm 3: 通用编码流程

1: **为总共 x^n 个位置从 0 到 x^n-1 分配编号**。

2: **将总共 x^n-n-1 个信息填入信息位：**

- 跳过编号 0 和所有编号为 x^i（其中 $i\in\mathbb{N}_0$ 且 $i<n$）的位置，这些位置预留作冗余位。
- 将信息填入所有剩余的位置。

3: **为冗余位编号 x^i 设定取值**：

- 将该冗余位记作 R_i，其取值记作 v_{x^i}。
- 将 x^i 用 x 进制表示为 $x_{n-1},\ldots,x_{i+1},x_i,x_{i-1},\ldots,x_0=0\cdots010\cdots0$。
- 为 v_{x_i} 取值，使得：

$$\left[\sum_{l=1}^{x-1} l\cdot\sum v_{(x_{n-1},\cdots,x_{i+1},l,x_{i-1},\cdots,x_0)}\right]\mod x=0.$$

其中 $\sum v_{(x_{n-1},\cdots,x_{i+1},l,x_{i-1},\cdots,x_0)}$ 为所有 x 进制编号中第 i 位等于 l 的位置所对应值的总和。(这个和会被 v_{x_i} 影响, 由于 $v_{x_i}=v_{(0,\cdots,0,1,0,\cdots,0)}$ 对应 l=1 的情况)

4: **为冗余位编号 0 设定取值 v_{modx}**：

- 为 v_{modx} 取值，使得：$\sum_{k=0}^{x^n-1} v_k \mod x=0$ 成立。

2.3.2 有效性证明

下面我们将证明，Algorithm 3 所生成的编码能够实现单错误纠正（SEC）。

证明. 假设仅有一位发生了错误，使得该位置的值变为 0 到 $x-1$ 之间的另一个整数。我们将其记作 v_i^{flaw}，以区别于原本的正确值 v_i。首先确认每一位的值的和，根据我们的设计：

$$\left(\sum v_i\right)\mod x=0$$

设：

$$\left(\sum v_i^{\text{flaw}}\right)\mod x=y$$

其中 $y\neq 0, y\in\mathbb{Z}^+, y<x$。因此，我们可以判断：该错误位置的值相较于原本的正确值，要么增加了 y，要么减少了 $x-y$。接下来，我们计算对应于第 x^i 位的区域的偏差加权和：

$$\left[\sum_{l=1}^{x-1} l\cdot\sum v^{\text{flaw}}_{(x_{n-1},\cdots,x_{i+1},l,x_{i-1},\cdots,x_0)}\right]\mod x=y_i$$

回忆该加权和将所有 x^n 个数位划分为 $x-1$ 个组（由最左侧的求和符号决定），每一组包含所有在其第 i 位（在 x 进制表示中）等于某个 l 的位置编号（即形如 xx$\cdots l\cdots$xx 的编号）。

为简化表达，定义每组的和为 σ_j，其中 $\sigma_j^{\text{flaw}}=j\cdot\sum v^{\text{flaw}}_{(x_{n-1},\cdots,x_{i+1},j,x_{i-1},\cdots,x_0)}$，我们有

$$[1\cdot\sigma_1+2\cdot\sigma_2+\cdots+(n-1)\cdot\sigma_{n-1}]\mod x=0.$$

设：

$$[1\cdot\sigma_1^{\text{flaw}}+2\cdot\sigma_2^{\text{flaw}}+\cdots+(n-1)\cdot\sigma_{n-1}^{\text{flaw}}]\mod x=y_i.$$

定义 $\Delta\sigma_j=\sigma_j^{\text{flaw}}-\sigma_j$：

$$[1\cdot\Delta\sigma_1+2\cdot\Delta\sigma_2+\cdots+(n-1)\cdot\Delta\sigma_{n-1}]\mod x=y_i.$$

由于只有一个错误，该错误必定只会出现在某一个分组中：

$$\exists! \, j \in \{1, 2, \ldots, n-1\} \text{ such that } \Delta\sigma_j \neq 0$$

设 r 表示该错误所在的分组索引，去除所有其他项（其中 $\Delta\sigma_j = 0$），则有：

$$[r \cdot \Delta\sigma_r] \mod x = [r \cdot y] \mod x = y_i.$$

与此同时，由于 $\Delta\sigma_r \neq 0$，且错误的位置唯一。因此，$\Delta\sigma_r = y$（或 $-(x-y)$，因为它们对应相同的系数，得到相同的模值）。考虑定理 2.1（将在下一页中证明）：设 x 为质数，且 $x, y, z \in \mathbb{Z}^+$，满足 $y, z < x$，则存在唯一的 l，使得：

$$l \cdot z \mod x = y,$$

其中 $l \in \mathbb{Z}^+$，并且 $l < x$。

根据该定理，对于每个 r，恰好存在一个合法的值 l，因此该方法可以判断出错误位处于被 x^i 划分的哪一组中。由此，我们可以锁定错误三元组在 x 进制表示下索引的第 i 位。对所有位重复此过程，我们可以确定该错误的位置（以 x 进制表示）。再结合最开始我们得到的 y，我们便可以恢复其原始值。 □

Theorem 2.1.

假设 x 是一个质数，且给定 $x, y, z \in \mathbb{Z}^+$，其中 $y, z < x$。证明存在唯一的 $l \in \mathbb{Z}^+$，使得

$$l \cdot z \equiv y \pmod{x} \quad \text{且 } l < x.$$

证明.

⇒ 为了证明 l 的唯一性，为了反证，假设存在两个满足条件的 l 使得 $l_1 \cdot z \mod x = y$ 且 $l_2 \cdot z \mod x = y$。因此，对于某些整数 k_1 和 k_2，有：

$$\begin{cases} l_1 \cdot z = k_1 \cdot x + y \\ l_2 \cdot z = k_2 \cdot x + y \end{cases} \quad \text{且} l_1 \neq l_2 \tag{1}$$

将两式相减并整理，使 x 成为等式的一部分，我们得到：

$$\frac{(l_1 - l_2) \cdot z}{x} = (k_1 - k_2) \in \mathbb{Z} \tag{2}$$

由于 $(l_1 - l_2) \cdot z$ 被 x 整除，因此必须满足 $x \mid (l_1 - l_2)$ 或 $x \mid z$。由于 x 是质数，它不可能被 $(l_1 - l_2)$ 和 z 中的部分因子分别整除，因此排除了两者分别包含 x 的因子的可能性。

假设 $x \mid (l_1 - l_2)$。由于 $l_1, l_2 < x$，因此 $|l_1 - l_2| < x$。根据整除的定义，存在 $m \in \mathbb{Z}$ 且 $|m| < 1$，使得 $m \cdot x = l_1 - l_2$，这意味着 $m = 0$，从而 $l_1 - l_2 = 0$，这与 $l_1 \neq l_2$ 矛盾。

现在假设 $x \mid z$，由于 $z < x$ 且 $z \in \mathbb{Z}^+$，这意味着 $z = 0$，但 $0 \notin \mathbb{Z}^+$。因此不存在满足 $x \mid z$ 的 z。

因此，在这两种情况下都会产生矛盾，式（1）中的两个等式不可能在 $l_1 \neq l_2$ 时同时成立。因此，我们得到结论：最多存在唯一的 l。

⇐ 证明 l 的存在性：为了反证，假设存在某些 x, y_0, z，使得不存在满足 $l \cdot z \equiv y_0 \pmod{x}$ 的 l。考虑 $l \in \{1, 2, \cdots, x-1\}$ 有 $(x-1)$ 个可能的值。对每个 l_i，存在对应的 y_i 使得 $l_i \cdot z \equiv y_i \pmod{x}$，且 $y_i \neq y_0$。因此，y_i 只能从集合 $\{1, 2, \cdots, x-1\} \setminus \{y_0\}$ 中取值，即 $(x-2)$ 个有效值。根据抽屉原理，必然存在不同的 l_i 和 l_j 对应同一个 y_i，即 $l_i \neq l_j$ 但 $y_i = y_j$，这与前面 ⇒ 的唯一性结论矛盾。

因此，结合存在性与唯一性，我们可以得出结论：对于给定的 x, y, z，存在且仅存在唯一的 l 满足

$$l \cdot z \equiv y \pmod{x}.$$

□

3 奇偶校验异或法

3.1 模 3 异或与模 3 逆对的定义

在介绍基于模 3 异或操作的三元奇偶校验方法之前，我们需要先引入几个重要的定义。

Definition 1.

模 p 加法（$\oplus_p$）是一种特殊的加法运算，其在每一位上执行模 p 运算而不进行进位。为了简洁起见，我们将其称为 p 进制下的异或，或简称为 **模 p 异或**。对于任意整数 $A, B \in \mathbb{Z}$，将它们用相同长度为 $k+1$ 的 p 进制表示，其中 k 的取值取决于绝对值较大的那个数的位数。

$$A = a_k a_{k-1} \cdots a_1 a_0, \quad B = b_k b_{k-1} \cdots b_1 b_0,$$

$$A \oplus_p B = C = c_k c_{k-1} \cdots c_1 c_0, \quad \text{其中} c_i = (a_i + b_i) \mod p, \quad i = 0, 1, \ldots, k$$

模 p 异或满足以下性质：

1. **闭合性：** 对于所有 $a, b < (p)_{10}^k$，有 $a \oplus_p b < (p)_{10}^k$。
2. **交换性：** 对于所有 $a, b \in \mathbb{Z}$，有 $a \oplus_p b = b \oplus_p a$。
3. **结合性：** 对于所有 $a, b, c \in \mathbb{Z}$，有 $(a \oplus_p b) \oplus_p c = a \oplus_p (b \oplus_p c)$。
4. **单位元素：** 存在元素 $0 \in \mathbb{Z}$，使得对于所有 $a \in \mathbb{Z}$，有 $0 \oplus_p a = a \oplus_p 0 = a$。
5. **相反元素：** 对于任意整数 $a \in \mathbb{Z}$，存在整数 $b \in \mathbb{Z}$，使得 $a \oplus_p b = b \oplus_p a = 0$，其中 0 是单位元素。我们称 a 和 b 为模 p 逆对，是彼此的模 p 加法相反数。为简洁起见，我们记作 $a = -b$。

Note 1. 特殊乘法： 为了简化记号，我们定义一种类似乘法的运算：

$r \cdot a$ 表示将a 与自身进行r 次异或（基于其进制）运算，　$r \in \mathbb{N}_0$　（通常情况下，r 用十进制表示）

3.2 模 3 异或在原型单纠双检码中的应用

奇偶校验也可以通过异或的方式实现，这种方式在计算上更为清晰简便。仍以第 2.2 节中的信息为例：

$$\begin{bmatrix} 000 & 001 & 002 \\ 010 & 011 & 012 \\ 020 & 021 & 022 \end{bmatrix} \begin{bmatrix} 100 & 101 & 102 \\ 110 & 111 & 112 \\ 120 & 121 & 122 \end{bmatrix} \begin{bmatrix} 200 & 201 & 202 \\ 210 & 211 & 212 \\ 220 & 221 & 222 \end{bmatrix} \xrightarrow{\text{填入值}} \begin{bmatrix} v_{mod3} & v_{x_0} & 2 \\ v_{x_1} & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} v_{x_2} & 2 & 0 \\ 0 & 1 & 0 \\ 2 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 1 & 2 & 0 \\ 0 & 1 & 2 \end{bmatrix}$$

不再分别计算每个冗余位所对应的区域，而是直接计算所有信息位的 “索引 * 值” 的模 3 异或和 $P_{message}$：

$$P_{message} = \bigoplus_{message} v_i \cdot (i)_3 = \begin{aligned} &2 \cdot 002 \oplus 1 \cdot 012 \oplus 1 \cdot 020 \oplus 1 \cdot 021 \oplus 2 \cdot 101 \oplus 1 \cdot 111 \oplus 2 \cdot 120 \\ &\oplus 1 \cdot 122 \oplus 2 \cdot 200 \oplus 1 \cdot 210 \oplus 2 \cdot 211 \oplus 1 \cdot 221 \oplus 2 \cdot 222 \end{aligned} = \mathbf{101}$$

冗余位 $v_{x_0}, v_{x_1}, v_{x_2}$ 的索引分别是 $001, 010, 100$。为他们取值，使得 $v_{x_2} v_{x_1} v_{x_0}$ 是 101 的相反数，即 202。

$$P_{all} = 000 = \bigoplus v_i \cdot (i)_3 = \bigoplus_{message} v_i \cdot (i)_3 \oplus (v_{x_0} \cdot 001) \oplus (v_{x_1} \cdot 010) \oplus (v_{x_2} \cdot 100) = 101 \oplus v_{x_2} v_{x_1} v_{x_0}$$

$$\begin{bmatrix} v_{mod3} = 1 & 2 & 2 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 2 & 2 & 0 \\ 0 & 1 & 0 \\ 2 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 1 & 2 & 0 \\ 0 & 1 & 2 \end{bmatrix}$$

最终为在 000 上的 v_{mod3} 取值，使所有值的和模 3(记作 P_{mod3}) 等于 0。因为 000 对应的值不影响 P_{all}，最终整个矩阵的异或和被设置为 000。

用这个特性，当仅有一个错误发生时，无论其出现在何处，整体的模 3 异或和都会从 000 偏移为该错误位的索引或其模 3 意义下的相反数。此时，我们可以利用 P_{mod3} 来判断属于哪种情况。因此，引入模 3 异或和后，我们能够更高效地对原型码进行编码与译码。

4 改进三元纠错码设计

我们的三元纠错码有两种优化方案：其一是 A1 码，移除特殊冗余位 v_{mod3}（亦称为 O），以提高码率并支持任意长度的信息编码；其二是 A2 码，引入一个新的特殊冗余位 E，将最小汉明距离扩展至 4，从而构造出单错纠正三错检测码（SEC–TED），同样支持任意长度的信息编码。

4.1 方案一 (A1)：更高信息率与自适应长度构造，精简索引三元纠错码

此前，我们仅考虑了信息长度为 $3^n - n - 1$ 形式的三元码构造方式。现在，借助模 3 逆对的性质，我们可以进一步扩展，使纠错码能够支持任意长度的信息。换句话说，我们可以构造一个 $[\frac{3^r-1}{2}, \frac{3^r-1}{2} - r, 3]_3$ 的线性码（r 是冗余位的数量），以适应任意长度的信息编码需求。

4.1.1 编码过程

对于任意三元信息，设长度为 $m \in \mathbb{Z}^+$，我们需要 n 个冗余位，存在唯一一个正整数 $n \in \mathbb{Z}^+$ 使得：

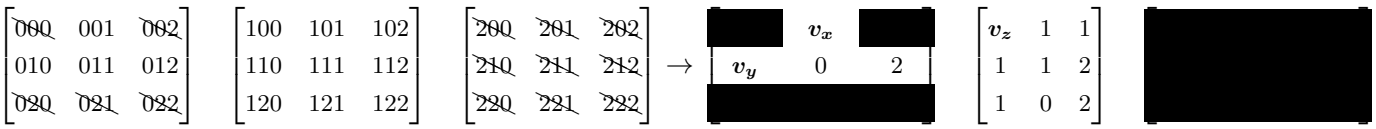
$$m \in (g(n-1), g(n)], \quad \text{其中} g(r) = \frac{3^r - 1}{2} - r$$

通过以下步骤，我们将长度为 m 位的信息编码为长度为 $m+n$ 位的 A1 纠错码：

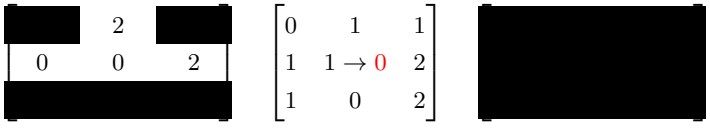
1. 构建矩阵，大小为 3^n.
2. **弃用**索引值为 0 以及所有其最高非零三进制位为 2 的索引。这样可以确保每一对模 3 逆对中只保留一个元素，即最高非零位为 1 的那个，因为除了 0 自身外，任意一对模 3 逆对中的两个数，其最高非零位必分别为 1 和 2。
3. 在剩余的索引中，跳过所有形如 3^i 的索引，其中 $i \in \mathbb{N}_0$ 且 $i < n$（这些位置预留作为冗余位）；然后按照索引的升序，将信息位填入其余未被弃用的位置中。
4. 填入所有信息后，弃用剩余所有的非冗余位。
5. 为冗余位取值，使得整个矩阵的异或和 $P_{all} = 0$。

4.1.2 示例：n=3

假设我们希望使用该方法对一个长度为 10 的三元信息进行编码，例如：”0, 211, 112, 102”。由于 $10 \in (g(2), g(3)] = (2, 10]$，我们选择 $n = 3$，并据此构建一个 $3 \times 3 \times 3$ 的三维矩阵。填入信息位，随后为冗余位取值。



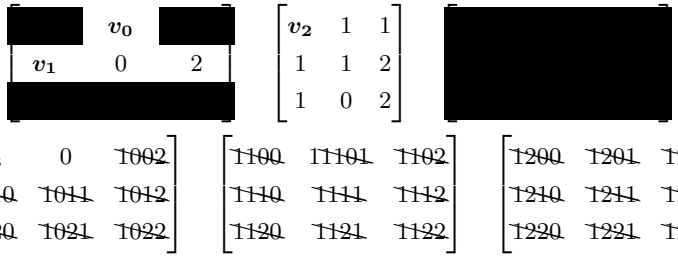
随后为冗余位赋值，使得整体异或和 $P_{all} = 0$。由于 $P_{message} = 001$，可以推出 $v_{x_2} v_{x_1} v_{x_0} = 002$。



注：将索引为 *111* 的位置上的 *1* 转换为 *0*，以模拟噪声干扰。

将索引为 111 的位置上的 1 修改为 0，以模拟噪声干扰。此时计算得到整体异或和 $P_{all} = 222$。在仅存在一个错误的前提下，该结果可能由索引 222 上的值发生 $+1$ 或 -2 所致，或由索引 111 上的值发生 -1 或 $+2$ 所致。由于在本编码方案中，索引 222 已被弃用，因此错误只能发生在索引 111 上。据此可定位错误并将索引 111 上的 0 恢复为 1，完成纠错过程。

4.1.3 示例：n=4



注：划掉的索引用于 $n = 4$ 范围内更长的信息，在这个示例中用不上。

更进一步，假设我们希望使用该方法对一个长度为 11 的三元信息进行编码，例如："02,111,121,020"，也就是在上一节的信息最后加一位 0。由于 $11 \in (g(3), g(4)] = (10, 36]$，我们选择 n=4，并据此构建一个 $3 \times 3 \times 3 \times 3$ 的四维矩阵。

4.2 方案二 (A2)：最小汉明距离 4, 单纠错三检测码（SEC-TED）

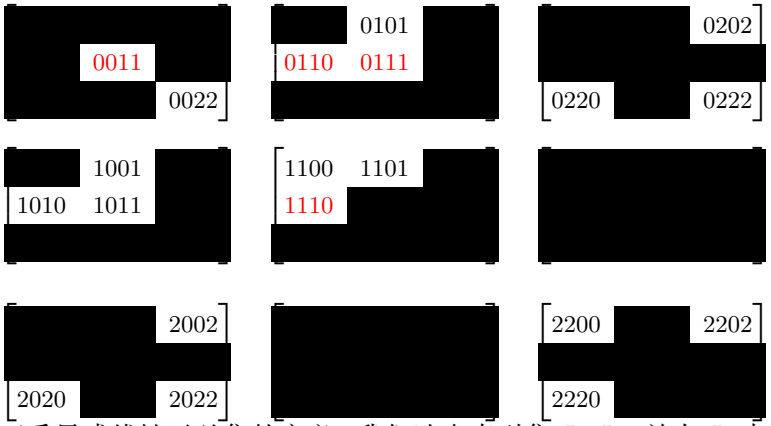
在上述精简索引的纠错码设计中，只要每个模 3 逆对中恰有一个索引被选中，编码便可正常运作。该优化方向通过减少冗余位数，提升了信息率。然而，其最小汉明距离仍为 3，无法实现对更多错误的纠正或检测能力。若能进一步利用三元异或下的线性无关性来精心选取索引集合，则有望提高最小距离，从而增强纠错性能。引入以下定义：

Definition 2. 三重异或线性无关集（三元）

若集合 L 满足以下性质，则称其为一个 **三重异或线性无关集**（*3-Wise XOR Linearly Independent Set*）：选出任意不同的三个元素 $A, B, C \in L$, $A \neq B \neq C \neq A$，使得：

$$\forall \alpha, \beta, \gamma \in \{0, 1, 2\}, \quad \alpha A \oplus \beta B \oplus \gamma C = 0 \iff \alpha = \beta = \gamma = 0.$$

4.2.1 选取索引与冗余位，构建 $[22, 15, 4]_3$ 线性码 A2



$I_1 = \{O, 0011, 0101, 1001, 0110, 1010, 1100, 0111, 1011, 1101, 1110\}$

$I_2 = \{E, 0022, 0202, 2002, 0220, 2020, 2200, 0222, 2022, 2202, 2220\}$

$R = \{0011, 0110, 0111, 1110\} \subset I_1$

$I = I_1 \cup I_2$

$[O(OddAdjust), E(EvenAdjust)]$

根据三重异或线性无关集的定义，我们选出索引集 I_1, I_2，并在 I_1 中选出 R 作为普通冗余位，并引入 O, E 作为特殊冗余位。$I_1 \setminus \{O\}$ 与 $I_2 \setminus \{E\}$ 都是三重异或线性无关集合（将在第 4.2.4 节中统一证明）。我们从 I_1 中选取 $R = \{0011, 0110, 0111, 1110\}$ 作为常规冗余位。只需设置这些冗余位的取值，即可将整个编码矩阵的异或和 P_{all} 重置为 0000。接下来，为特殊冗余位 O 和 E 赋值，使得 I_1 与 I_2 中各索引对应值的总和 P_1, P_2 都能被 3 整除。我们将在后续小节中证明，此构造能纠正一个错误，检测两个错误，并在三处错误时提示错误存在。最终得到，A2 码，码长为 22，信息长度为 16，信息率为 0.727，最小汉明距离为 4，记作 $[22, 16, 4]_3$，即单纠错三检错（SEC–TED）码。

4.2.2 示例

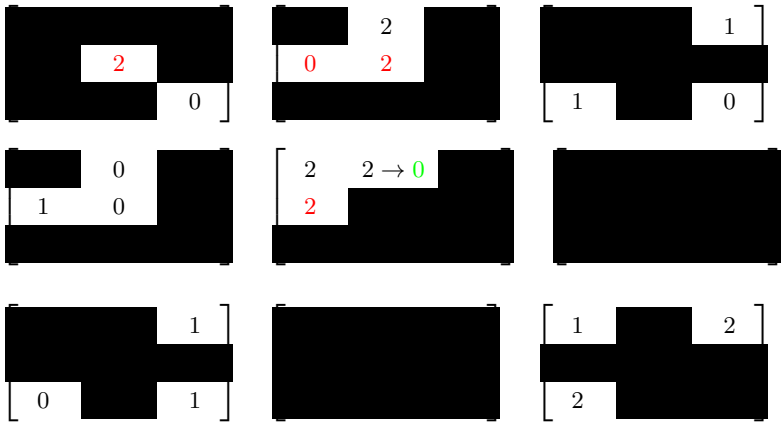
Algorithm 4: 编码 $[22,16,4]_3$ 线性码 A2：**0,211,001,022,101,122**

1: **初始化：**将 0,211,001,022,101,122 填入信息位，并计算 $P_{message}$（$I_{\text{message}} = I \setminus (R \cup \{O,E\})$）

$$P_{message} = \bigoplus_{i \in I_{\text{Message}}} = v_i \cdot i = \mathbf{1202}$$

$$= 2 \cdot 0101 \oplus 0202 \oplus 0220 \oplus 1010 \oplus 2 \cdot 1100 \oplus 2 \cdot 1101 \oplus 2002 \oplus 2022 \oplus 2200 \oplus 2 \cdot 2202 \oplus 2 \cdot 2220$$

2: **为冗余位赋值：**使整个矩阵的异或和 P_{all} 重置为 0000



$(R = \{0011, 0110, 0111, 1110\})$

$$P_{redundant} = \bigoplus_{i \in R} v_i \cdot i = \mathbf{2101}$$

$$= 2 \cdot 0011 \oplus 0 \cdot 0110 \oplus 2 \cdot 0111 \oplus 2 \cdot 1110$$

$$P_{all} = \bigoplus_{i \in I} v_i \cdot i = P_{redundant} \oplus P_{message}$$

$$= 2101 \oplus 1202 = \mathbf{0000}$$

$$\left[O = 2 \to 1, E = 0\right]$$

注：将 *1101* 上的 *2* 变为 *0*，将 *O* 上的 *2* 转化成 *1* 来模拟噪声干扰。

3: 最后为 O 和 E 赋值：

$$P_1 = (\sum_{i \in I_1 - O} i + O) \mod 3 = (13 + O) \mod 3 = 0 = (\sum_{i \in I_2 - E} i + E) \mod 3 = (9 + E) \mod 3 = P_2$$

得 $O = 2, E = 0$，将他们写在纠错码最后。成功编码了信息得到：**2,020,211,001,022,210,112,220**。

Algorithm 5: 解码 $[22,16,4]_3$ 线性码 A2：**2,020,211,001,022,210,112,220**

1: **模拟噪声干扰：**将 1101 上的 2 变为 0，将 O 上的 2 转化成 1：2,020,211,001,022,210,112,210。

2: **检查三个特征值:** 确认当前 P_{all}, P_1, P_2 的值。

$$P_{all} = \bigoplus_{i \in I} v_i \cdot i = 1101, \quad P_1 = \sum_{i \in I_1} v_i \mod 3 = 0, \quad P_2 = \sum_{i \in I_2} v_i \mod 3 = 0$$

3: **结论：** P_{all} 不等于零，推断出错误一定存在。假如错误只有一个，但 P_1, P_2 都等于零，单个错误无论位于 I_1 或 I_2 都无法符合当前特征值，**所以检测到至少有两处错误**。

4.2.3 扩展 A2 码与自适应长度构造

基于上述 A2 的设计思路，我们进一步构造出一类三元线性码。设 $f'(r)$ 表示在长度为 r 的范围内，所能构成的最大三重异或线性无关集合的元素数量。则该类码可以表示为：$[2f'(r)+2,\ 2f'(r)-r,\ 4]_3$，其中包括 r 个常规冗余位和 2 个奇偶校验冗余位。（尚需证明 $f(r)$ 确实等于所定义的 $f'(r)$）

$$n = \begin{cases} 2, & \text{if } r = 3, \\ \left\lfloor \frac{r}{2} \right\rfloor, & \text{if } 4 \le r \le 7, \\ \left\lceil \frac{r}{2} \right\rceil - 1, & \text{if } r \ge 8. \end{cases}$$

$$f(r) = \sum_{i=n}^{2n-1} \binom{r}{i}$$

r	3	4	5	6	7	8	9	⋯
f(r)	4	10	20	41	91	182	372	⋯
2f(r)+2	10	22	42	84	184	366	746	⋯
2f(r)-r	5	16	35	76	175	356	735	⋯
信息率	0.5	0.625	0.833	0.905	0.951	0.973	0.985	⋯

Algorithm 6: A2 通用编码流程

1: **构建 I_1 和 I_2:**($I = I_1 \cup I_2$ 是所有选中索引的集)

- 定义集合 I_1，其包含所有长度小于等于 r，且符合以下条件的的三元数，这些三元数仅由 0 和 1 组成，且恰好有 k 个 1，其中 $k \in [n, 2n-1]$。随后定义集合 I_2，其包含所有 I_1 中元素的模 3 相反数（即对每个位进行模 3 乘以 2）。
- I_1 加上奇偶校验冗余位 O, I_2 加上奇偶校验冗余位 E。

2: **将选定索引划分出冗余位 R 和信息位**:

- **标准情况：当 $r > 7 \vee r \in O$ （此处代表 r 是奇数）**
  - $R_j = \sum_{i=1+\left\lfloor \frac{j}{2} \right\rfloor}^{n+\left\lceil \frac{j}{2} \right\rceil} e_i \quad (\forall j < r, R_j \in I_1$, 因为 R_j 有刚好 n 或 n+1 位为 1)。
  - 定义集合 $R = \{R_1, R_2, \cdots, R_r\}$，其中 $|R| = r$，也就是常规冗余位的数量。
  - O, E 为奇偶校验冗余位。
  - I 中剩余的索引对应信息位。
- **其余情况：当 $r \le 7 \wedge r \in E$ （此处代表 r 是偶数）**
  - $R_j = \sum_{i=\left\lceil \frac{j}{2} \right\rceil}^{n+\left\lfloor \frac{j}{2} \right\rfloor} e_i \quad (\forall j < r, R_j \in I_1$，因为 R_j 有刚好 n 或 n+1 位为 1)。
  - 其余与情况一相同。

3: **为常规冗余位赋值**：在模 3 运算下，冗余位的线性组合可以获得任意标准基向量 e_i。

- **标准情况：(当 $r > 7 \vee r \in O$)**
  - $e_i = R_{2i-1} - R_{2i}, i \in [1, n]$
  - $e_j = R_{2(j-(n+1))+1} - R_{2(j-(n+1))}, j \in [n+2, r]$
  - $e_{n+1} = R_1 - \sum_{i=1}^{n} e_i$
- **其余情况：(当 $r \le 7 \wedge r \in E$)**
  - $e_i = R_{2i} - R_{2i+1}, i \in [1, n-1]$
  - $e_j = R_{2(j-n)} - R_{2(j-n)-1}, j \in [n+1, r]$
  - $e_n = R_1 - \sum_{i=1}^{n-1} e_i$
- 计算信息位异或和后，拆解成 r 个 “系数 * 标准基向量”，再一一调整 R_i 的值。

4: **为奇偶校验冗余位 O, E 赋值**：选出 v_O, v_E 使得

$$P_1 = (v_O + \sum_{i \in I_1 - \{O\}} v_i) \mod 3 = 0 = (v_E + \sum_{j \in I_2 - \{E\}} v_j) \mod 3 = P_2$$

自适应长度: 如果信息长度不是 $2 \cdot f(r) - r$ 的格式, 设信息长为 m, 则存在 n 符合 $m \in (f(n-1), f(n)]$。照常构建 I_1 和 I_2，但是移除 I_2 最后 $(2 \cdot f(n) - n - m)$ 个索引，不包括 E。其余步骤相同。

4.2.4 A2 码最小距离 4 证明

最小汉明距离为 4 意味着任意两个合法码字之间至少有 4 位不同。这保证了 A2 码可以纠正任意一个错误，并能在最多三个错误位的情况下检测到错误的存在，这是单纠三检码的基本性质。

Theorem 4.1.
对于任意两个内容不同但长度相同的三元信息串 X 与 Y，将它们分别编码为相同长度的 A2 码 X' 与 Y'。则 X' 与 Y' 至少在 4 个三元位上不同。

根据上述编码流程，正确无误的 A2 码需满足以下三个校正子（Syndrome）条件：

$$P_1 = \sum_{i \in I_1} v_i \mod 3 = 0 \qquad P_2 = \sum_{i \in I_2} v_i \mod 3 = 0 \qquad P_{all} = \bigoplus_{i \in I} v_i \cdot i = 0$$

证明.
先证明距离至少为 4，分为四种情况讨论：

情况 1： X 与 Y 仅有一位不匹配

- 设：不同的那一位索引为 $A \in I \setminus (R \cup \{\mathrm{O}, \mathrm{E}\})$

子情况 1.1： $2A \in R$（即 A 的模 3 相反数是常规冗余位）

- 索引为 A 的信息位不同，且索引为 $2A$ 的常规冗余位不同（为了特征值 P_{all}）。
- 注意到 $2A \in R \subset I_1$，因此 $A \in I_2$
- 因为 I_1, I_2 各有一位不同，所以 X' 与 Y' 的奇偶校验冗余位 O 与 E 不同（为了特征值 P_1 与 P_2）。

$$X'_{\mathrm{O}} \neq Y'_{\mathrm{O}} \quad \text{and} \quad X'_{\mathrm{E}} \neq Y'_{\mathrm{E}}$$

- 总计不匹配的位数量为: $\underbrace{A}_{\text{msg}} + \underbrace{2A}_{\text{red}} + \underbrace{\mathrm{O}+\mathrm{E}}_{\text{chk}} = 4 \geq 4$ （成立）

子情况 1.2： $2A \notin R$

- 定义 $\Delta a \equiv X'_A - Y'_A \pmod 3$
- 三重异或线性无关使得:

$$\nexists B, C \in R \ (B, C \neq 2A \notin R) \quad \text{s.t.} \qquad \Delta a \cdot A \equiv \beta B \oplus \gamma C \pmod 3 \quad (\beta, \gamma \neq 0)$$

- 冗余位不同的位数至少为: 3
- 总计不匹配的位数量为: $\underbrace{1}_{\text{msg}} + \underbrace{3}_{\text{red}} = 4 \geq 4$ （成立）

情况 2： X 与 Y 仅有两位不匹配

- 设：不同的两位索引为 $A, B \in I \setminus (R \cup \{\mathrm{O}, \mathrm{E}\})$
- 定义：$\Delta a \equiv X'_A - Y'_A \pmod 3$，$\Delta b \equiv X'_B - Y'_B \pmod 3$

子情况 2.1： $B = 2A$ 且 $\Delta a = \Delta b$

- 使得 X',Y' 信息位异或和相等 $\Rightarrow$ X',Y' 冗余位完全相同
- 若 $A \in I_1 \Rightarrow B = 2A \in I_2$，若 $A \in I_2 \Rightarrow B = 2A \in I_1$。
- 则 A,B 分别对 O 和 E 造成影响，使得 X',Y' 的 O 和 E 因 P_1，P_2 不匹配:

$$\underbrace{A, B}_{\text{msg}} + \underbrace{\mathrm{O}, \mathrm{E}}_{\text{chk}} = 4 \geq 4 \quad \text{（成立）}$$

子情况 2.2: $B = 2A$ 且 $\Delta a \neq \Delta b$

- 定义 $\Delta a \cdot A \oplus \Delta b \cdot B \pmod 3 \equiv d \cdot A$ ($d \in \{1, 2\}, d \neq 0$ as $\Delta a \neq \Delta b$)
- 三重异或线性无关使得至少三位冗余位不匹配：

$$\nexists M, N \in R \setminus \{A, B\} \quad \text{s.t.} \quad dA \equiv \beta M \oplus \gamma N \pmod 3$$

- 总计不匹配的位数量为：$\underbrace{2}_{\text{msg}} + \underbrace{3}_{\text{red}} = 5 \geq 4$ （成立）

子情况 2.3: $B \neq 2A$

- 三重异或线性无关使得至少两位冗余位不匹配：

$$\nexists M \in R \setminus \{A, B\} \quad \text{s.t.} \quad \Delta a A \oplus \Delta b B \equiv \gamma M \pmod 3$$

- 总计不匹配的位数量为：$\underbrace{2}_{\text{msg}} + \underbrace{2}_{\text{red}} = 4 \geq 4$ （成立）

情况 3: X 与 Y 仅有三位不匹配

- 设：不同的三位索引为 $A, B, C \in I \setminus (R \cup \{\mathrm{O}, \mathrm{E}\})$
- 定义：$\Delta a \equiv X'_A - Y'_A \pmod 3$，$\Delta b \equiv X'_B - Y'_B \pmod 3$，$\Delta c \equiv X'_C - Y'_C \pmod 3$
- 从三重异或线性无关推断 X' 与 Y' 的信息位异或和不相等：

$$\nexists (\Delta a, \Delta b, \Delta c) \in \{1, 2\}^3 \text{ s.t. } \Delta a \cdot A \oplus \Delta b \cdot B \oplus \Delta c \cdot C \equiv 0 \pmod 3$$

- 总计不匹配的位数量至少为：$\underbrace{3}_{\text{msg}} + \underbrace{\geq 1}_{\text{red}} \geq 4$ （成立）

情况 4: X 与 Y 有四个及以上的位不匹配

- 即使不考虑冗余位也可直接得出不匹配的位数量至少为：$\underbrace{\geq 4}_{\text{message}} \geq 4$ （成立）

最小汉明距离为 4 的存在性：
存在以下信息 X, Y，使得他们的 A2 码 X', Y' 之间的距离为 4（仅有 4 位不同）：

$$X = 0,211,001,022,101,122 \quad X' = 2,020,211,001,022,210,112,220$$

$$Y = 0,021,021,022,001,122 \quad Y' = 2,000,221,021,022,200,112,220$$

X 与 Y 为长度相等的信息，但他们的 A2 码有刚好四位不同，意味着 A2 的最小距离不能大于 4。结合刚刚证明的最小距离至少为 4，A2 码的最小距离刚好为 4。 □

4.2.5 A2 纠错策略

回顾一下，A2 码需要满足以下三个校正子条件:

$$P_1 = \sum_{i \in I_1} v_i \mod 3 = 0 \qquad P_2 = \sum_{i \in I_2} v_i \mod 3 = 0 \qquad P_{all} = \bigoplus_{i \in I} v_i \cdot i = 0$$

那么通过计算 P_1, P_2, P_{all}，我们可以得知 A2 的错误情况。定义 X' 为纠错码，并暂时不考虑 3 个及以上错误的情况，则有以下表 2中策略。其中，校正子 P_1, P_2 可以反映对应的 I_1, I_2 中是否有错误存在，校正子 P_{all} 的值将反映错误的数量。

	A: $P_{\text{all}} = 0$	B: $P_{all} \in I \vee 2P_{all} \in I$	C: $P_{\text{all}}, 2P_{\text{all}} \notin I \cup \{0\}$
1: $P_1 = 0, P_2 = 0$	完美	两位错误	
2: $P_1 = 0, P_2 \neq 0$	$X_{\overline{E}} = X_E - P_2$	复合情况 1	两位错误
3: $P_1 \neq 0, P_2 = 0$	$X_{\overline{O}} = X_O - P_1$	复合情况 2	两位错误
4: $P_1 \neq 0, P_2 \neq 0$	两位错误		

表 2: 错误分布情况分析

- **4-**: 因为 P_1, P_2 都不等于零，这表示 I_1 和 I_2 中分别至少存在一个错误。因此，检测到存在两位或者更多的错误。
- **-C**: 在只有一个错误的情况下，不可能将 P_{all} 的值从 0 变为选中索引外的值。这表示存在两位或者更多的错误。
- **1A**: 这是一个完美码。由于该码的最小距离为 4，因此至少需要 4 个错误才可能将一个完美码其变为另一个完美码。
- **2A**: $P_2 \neq 0$ 表明至少有一个错误位于 I_2。而 $P_1 = 0$ 表明 0 或 2+ 位错误在 I_1。由于 $1 + 2 \geq 3$，我们只考虑 I_1 中没有错误的情况。随后，根据三重异或线性无关的性质：

  $$\nexists A, B, C \in I_2 \setminus \{E\} \text{ s.t. } A \neq B \neq C \neq A,\ \alpha, \beta, \gamma \in \{1, 2\}, \text{and } \alpha A \oplus \beta B \oplus \gamma C \equiv 0$$

  结合 $P_{\text{all}} = 0$，至少需要四个错误才能使 P_{all} 保持为 0。因此，该错误一定位于 $E \in I_2$，且 P_2 代表它从原本正确值到错误值的变化。原本正确值 $\overline{E} : X_{\overline{E}} = (X_E - P_2) \mod 3$。
- **3A**: 与 2A 相似，该错误位于 $O \in I_1$，原本正确值 $\overline{O} : X_{\overline{O}} = (X_O - P_1) \mod 3$。
- **1B**: $P_1 = P_2 = 0$ 表明 I_1, I_2 分别各有 0 个或 2 个及以上的错误。由于 $0 \notin I$，且 $P_{\text{all}}, 2P_{all} \neq 0$ ，这表示码中存在至少一位错误。因此，可以排除 I_1 与 I_2 都为 0 个错误的情况，检测到至少两位错误。
- **复合情况 1(2B):** 定义 $f = P_2 \cdot P_{\text{all}}$。如果 $f \notin I_2$, 表示存在两位及以上的错误。如果 $f \in I_2$，则仅存在一位错误位于 $f \in I_2$，原本正确值为 $X_{\overline{f}} = (X_f - P_2) \mod 3$。
- **复合情况 2(3B):** 定义 $f = P_1 \cdot P_{\text{all}}$。如果 $f \notin I_1$，表示存在两位及以上的错误。如果 $f \in I_1$，则仅存在一位错误位于 $f \in I_1$，原本正确值为 $X_{\overline{f}} = (X_f - P_1) \mod 3$。

因此，该 A2 码可以在仅有两位错误时报告有两位或以上的错误存在。对于三位错误的情况，虽然以上策略会将其误认为某个一位错误情况，但根据最小距离为 4 的性质，A2 码仍然会检测到错误**存在**，即为单纠错三检测码。

4.2.6 线性无关索引集 I_1, I_2

索引集 I_1 与 I_2，从任意一个中取出三个不同的索引，他们在异或运算下线性无关。

Theorem 4.2.
选任意索引 $A, B, C \in I_1 \setminus \{O\}$，其中 $A \neq B \neq C \neq A$。则有：

$$\forall \alpha, \beta, \gamma \in \{1, 2\}, \alpha A \oplus \beta B \oplus \gamma C \neq 0$$

证明.
等价于

$$\forall \alpha, \beta \in \{1,2\}, \forall A, B \in I_1 \setminus \{O\} \text{ where } A \neq B,$$

$$\nexists C \in I_1 \setminus \{O, A, B\},\ \alpha A \oplus \beta B \equiv C$$

回顾一下我们如何构建 $I_1 \setminus O$:

- 定义集合 I_1，其包含所有长度小于等于 r，且符合以下条件的的三元数，这些三元数仅由 0 和 1 组成，且恰好有 k 个 1，其中 $k \in [n, 2n-1]$，$(n \geq 1)$。

设 A 有 a 位为 1，B 有 b 位为 1. 然后 A 和 B 有 l 位重叠（位对应的值同时为 1）。不失一般性，设 $l \leq a \leq b$, 且 $a, b \in [n, 2n-1]$。随后考虑 A 与 B 的四种异或线性组合。

- $A \oplus B = S_1$，S_1 有着 $a+b-l$ 位等于 1，有 l 位等于 2。因为 I_1 中 O 之外的元素都只由 0 和 1 组成，$S_1 \in I_1$ 需要 $\mathrm{l} = 0$。但是 S_1 需要 $a+b-l = a+b-0 = a+b \geq 2n$ 位等于 1，超出了为 I_1 设定的 1 的数量范围 (回顾，$a, b \in [n, 2n-1]$)。因此，$S_1 \notin I_1$。
- $A \oplus 2B = S_2$，S_2 有 a-l 位等于 1，b-l 位等于 2。如果 $S_2 \in I_1$，需要 $b-l = 0$，则 $b = l$。那么 $a - l \leq 0$，导致 $a - l \notin [n, 2n-1]$，相互矛盾。因此 $S_2 \notin I_1$。
- $2A \oplus B = S_3$，S_3 有 b-l 位等于 1，$a-l$ 位等于 2。如果 $S_2 \in I_1$，需要 $a-l = 0$，则 $a = l$。那么 $b - l = b - a \leq n-1$(回顾，$a, b \in [n, 2n-1]$). 因此，$b - l \notin [n, 2n-1]$，相互矛盾，等于 1 的位数不在选定范围内。因此 $S_3 \notin I_1$。
- $2A \oplus 2B = S_4$, S_4 有 l 位等于 1，$a+b-l$ 位等于 2。如果 $S_4 \in I_1$，需要 $a+b-l = 0$，则 $a+b = l$。但是根据 $0 \leq l \leq a \leq b$ 和 $a, b \geq n \geq 1$, 相互矛盾。因此 $S_4 \notin I_1$。

因为 $S_1, S_2, S_3, S_4 \notin I_1$，结论 $\nexists C \in I_1 \setminus \{O, A, B\},\ \alpha A \oplus \beta B \equiv C$ □

注：通过类似的步骤可以为 I_2 证明：选任意索引 $A, B, C \in I_2 \setminus \{E\}$，其中 $A \neq B \neq C \neq A$。则有：

$$\forall \alpha, \beta, \gamma \in \{1,2\}, \alpha A \oplus \beta B \oplus \gamma C \neq 0$$

4.3 评估

如果追求更高的码率，可以从 A2 中移除关于 O, E, I_2 的部分，只余 I_1。实际上结构与 A1 更为相似，但是选取索引的条件从双重异或线性无关集（每组模 3 逆对只取一位索引）变为三重异或线性无关集。构造出 $[10, 6, 4]_3$ 线性码，汉明距离相同的同时，其码率比 A2 更高。可以扩展为 $[f'(r), f'(r)-r, 4]_3$ 线性码，将其命名为 A2 稀疏版。

+ 可延展性: 移除 I_2 结构后，A2 稀疏版的结构可以通过选取索引为 n 重异或线性无关集来获取汉明距离为 n+1 的三元纠错码。而 A2 基础版会在错误分别存在于 I_1, I_2 时影响对校正子的分析。
+ 码率: A2 稀疏版的码率高于基础版。在冗余位数量均为 r+2 时，其信息位数量至多为 $f(r+2)-(r+2)$，高于基础版 $2 \cdot f(r) - r$ 的上限。
– 计算复杂度: 对于相同长度的信息位，A2 稀疏版需要从更大的索引空间中筛选出符合条件的集合。相比之下，A2 基础版因同时具有 I_1 和 I_2 结构，其有效索引更短（例如，111 对比 10111），这使得 A2 稀疏版在编码与解码时的计算开销均更高。

段落	名称	$[n,k,d]_q$	通用 $[n,k,d]_q$	xEC-yED	特点
2.1	原型	$[9,6,3]_3$	$[3^{r-1},3^{r-1}-r,3]_3$	单纠错–双检测	将二元汉明码的奇偶校验思路应用在三进制。
2.3	素数通用码	N/A	$[x^{n-1},x^{n-1}-n,3]_x$	单纠错–双检测	将原型的奇偶校验思路应用在任意素数进制。
4.1	A1/三元汉明码	$[13,10,3]_3$	$[\frac{3^{r-1}}{2},\frac{3^{r-1}}{2}-r,3]_3$	单纠错–双检测	基于原型，从每个模 3 逆对只选出一位索引，码率提升。
4.2	A2	$[22,16,4]_3$	$[2f'(r)+2,2f'(r)-r,4]_3$	单纠错–三检测	基于原型，选中索引分出 I_1,I_2 两组更加密集。计算效率提升，消耗降低。
4.3	A2 稀疏版	$[10,6,4]_3$	$[f'(r),f'(r)-r,4]_3$	单纠错–三检测	基于 A2，移除 O,E,I_2。精简后的 A2 稀疏版拥有更高的信息率。
5.1	三元戈莱码	$[11,6,5]_3$	N/A	双纠错–四检测	三元完美码

表 3: 文中相关纠错码总览

5 研究展望：最小汉明距离提升至 5 及以上

如 4.3 节所述，A2 稀疏版的框架允许通过选取 n 重异或线性无关集作为索引来构造汉明距离达 $n+1$ 的纠错码，从而具备纠正最多 $\lfloor\frac{n}{2}\rfloor$ 位错误的能力。

5.1 简要示例：$n=4$，双纠四检码 (DEC–QED) 与三元戈莱码

当 $n=4$, 选出四重异或线性无关集作为冗余位与信息位的索引，构造出最小距离为 5 的码。完美码 $[11,6,5]_3$ 三元戈莱码 [9] 的生成矩阵的每一列恰好符合我们对索引的要求。定义 I 为一个四重异或线性无关集，再划分为信息位集 M 和冗余位集 R:

$$I=\{00001,00010,00100,01000,10000,01122,10212,12021,12102,22110,22222\},$$

$$R=\{00001,00010,00100,01000,10000\},\quad M=I\setminus R=\{01122,10212,12021,12102,22110,22222\}.$$

随后，我们将基于四重异或线性无关集的框架来阐释戈莱码。

编码步骤:

1. 将信息 012,210 映射到 M 中的三元位:

$$\{01122:0,\ 10212:1,\ 12021:2,\ 12102:2,\ 22110:1,\ 22222:0\}.$$

2. 计算 M 中索引与对应值的异或和:

$$10212\oplus(12021\cdot 2)\oplus(12102\cdot 2)\oplus 22110=11202.$$

3. 为 R 中的冗余位索引赋值，使得 I 中索引与对应值的异或和 $P_{\text{all}}=0$。

$$11202\oplus 22101=00000,\qquad 22101=00001\oplus 00100\oplus(01000\cdot 2)\oplus(10000\cdot 2)$$

$$\{00001:1,\ 00010:0,\ 00100:1,\ 01000:2,\ 10000:2\}.$$

4. 完成编码: 10,122,012,210.

解码示例: 接收到的码字为 10,122,012,222，计算校正子 P_{all}:

$$P_{\text{all}}=00221=1\times 22110\oplus 2\times 22222.$$

根据四重异或线性无关集的性质，该校正子 00221 有唯一分解 (仅用集内索引)，可定位错误位索引为 22110 与 22222，其幅度分别在原正确值上增加了 +1 与 +2。通过减去相应偏移量进行修正，还原正确码字 10,122,012,210.

当待编码的信息长度超过 6 时，可沿用前述思路：从长度 $r\geq 6$ 的索引范围（如 $000000,\cdots,222222$）中，构建规模更大的四重异或线性无关集，对其进行划分得到集合 R，进而获得元素更多的集合 M，以容纳更长的信息。构造的 n 重异或线性无关集规模越大，所得纠错码的码率也越高。这使其性能接近最优的完美码。然而，给定约束下该线性无关集的最大规模构造问题尚未完全解决，这为后续研究指明了方向。

6 总结与结论

在本文中，我们从 $[3^r, 3^r-(r+1), 3]_3$ 原型码出发，将奇偶校验的思路拓展到三进制系统。在此基础上，构建出适用于所有素数进制的通用一纠二检编码方案，并证明了其正确性。通过引入模三异或与逆对的概念，我们提升了编码与解码的效率。我们进一步优化了原型码，使其能够支持对任意长度信息进行编码。同时，提出两种扩展方向：其一，通过减少冗余位以提升信息率的 A1 纠错码；其二，选取 n 重异或线性无关索引集合来提升最小距离，牺牲信息率来纠正更多错误的 A2 纠错码。最后，我们提出了 n 重异或线性无关集可用于构造距离为 n+1 的线性码，这为设计具有任意大最小距离的码提供了一条可行路径。特别值得注意的是，当码距 d=5 时，此方法恰好对应着三元戈莱码，这有力地验证了我们方法的有效性与通用性。

我们的研究结果表明，奇偶校验的理念在三元系统中依然表现出色，能够利用二元系统中不存在的成对结构特性。本文所发展的技术为深入探索模运算原理的应用以及研究替代性编码策略铺平了道路。我们期望通过突破传统编码理论的边界，推动对非二进制编码在现代通信技术中应用的深入研究与持续关注。

7 补充材料

附录 A：通过权重为 3 的码字证明三元原型码最小距离的精确性

我们给出一个权重为 3 码字的具体实例，证明了这个带有切片奇偶校验与全局和校验，长度为 27 的三元原型码的最小距离恰好为 3。

Lemma 7.1 (三元原型码权重 3 示例)**.**
考虑索引集 $\{0,\dots,26\}$，我们将其中的每个索引视作一个维度为 3 的三进制向量 $\mathbf{i}=(i_2,i_1,i_0)\in\{0,1,2\}^3$，如 $19_{10}=201_3=(2,0,1)$。任取一个向量 $\mathbf{u}\in\{0,1,2\}^3$，并定义以下三个位置：

$$\mathbf{a}=\mathbf{u},\qquad \mathbf{b}=\mathbf{u}+\mathbf{1}\pmod 3,\qquad \mathbf{c}=\mathbf{u}+\mathbf{2}\pmod 3,$$

其中定义 $\mathbf{1}=(1,1,1)$，且加法是逐分量模 3 运算。若 $u=(0,1,2)$，则有 $a=(0,1,2)$，$b=(1,2,0)$，$c=(2,0,1)$。让 $\mathbf{a},\mathbf{b},\mathbf{c}$ 这三个位置对应的值设为 1，其余所有位设为 0。由此构造出的长度为 27 的向量是一个非零码字，其权重为 3（即恰好包含 3 个非零元）。

证明.
根据构造，对于每个数位维度 $j\in 0,1,2$，$\mathbf{a},\mathbf{b},\mathbf{c}$ 在该维度的坐标构成 $(0,1,2)$ 的一个排列。根据第 3.2 节中模 3 异或的性质，有 $1\cdot a\oplus 1\cdot b\oplus 1\oplus c=(v_{x_2},v_{x_1},v_{x_0})=(0,0,0)$，这表明所有切片奇偶校验均通过。同时，全局和校验 $v_{mod3}=1+1+1\equiv 0\pmod 3$ 亦为零。两个校正子结果均表明未检测到错误，而该向量本身非零且恰有三位非零，故其为一个权重为 3 的码字。

因此，该码的最小距离 $d_{\min}\le 3$。结合 2.3.2 节中针对此构造能纠正一位错误的证明（该证明已表明 $d_{\min}\ge 3$），我们最终得出结论：$d_{\min}=3$。 □

附录 B: 解码伪代码

本附录记录了原型码的单错误纠正解码器，以及距离为 4 的 A2 码（具备检测 ≥ 2 位错误能力）的清晰过程化伪代码。

B.1 针对原型码 $[27, 23, 3]_3$ 的单纠错解码器

Algorithm 7: 单纠错解码 (三元原型码)

Input: 接收码字 $r \in \mathbb{F}_3^{27}$。

Output: 估计码字 $\hat{c}$。

根据三个数位奇偶检验计算切片校正子 $S_0, S_1, S_2 \in \mathbb{F}_3$。
计算全局和校正子 $S_{\text{all}} \in \mathbb{F}_3$。
if $S_0 = S_1 = S_2 = S_{\text{all}} = 0$ **then**
    return $\hat{c} = r$ // 没有错误
else
    Let $\hat{\mathbf{i}} = (\hat{i}_2, \hat{i}_1, \hat{i}_0) \leftarrow f_{\text{locate}}(S_2, S_1, S_0) \in \{0, 1, 2\}^3$。 // 从切片校正子解码错误位置索引
    令 $\hat{e} \leftarrow S_{\text{all}}$ ($\mathbb{F}_3$ 上的符号错误)。 // 根据全局和推算错误值
    在位置 $\hat{\mathbf{i}}$ 处将 r 的值翻转 $-\hat{e}$ 以得到 $\hat{c}$。 // 将错误位还原
    return $\hat{c}$

注. 定位函数 f_{locate} 由校验矩阵决定：三元组 (S_2, S_1, S_0) 表示单个错误的三进制索引；用代数术语来说，它就是错误所击中的矩阵 H 的列。

B.2 针对 A2 稀疏码的单纠三检解码器

Algorithm 8: 单纠三检解码 (A2 稀疏码)

Input: 接收码字 $r \in \mathbb{F}_3^n$。

Output: 估计码字 $\hat{c}$ 或 **检测到存在多位错误**。

计算校正子 $S_{\text{idxXor}} \in \mathbb{F}_3^n, S_{\text{value}} \in \mathbb{F}_3$。
if $S_{idxXor} = S_{value} = 0$ **then**
    return $\hat{c} = r$ // 没有错误
else if $S_{idxXor} \neq 0$ **then**
    if $S_{loc} = S_{value} \cdot S_{idxXor} \in I \setminus \{0\}$ **then**
        Let $\hat{\mathbf{i}} = (\hat{i}_{n-1}, \cdots, \hat{i}_1, \hat{i}_0) \leftarrow f_{\text{locate}}(S_{loc}) \in \{0, 1, 2\}^n$。 // 从切片校正子解码错误位置索引
        令 $\hat{e} \leftarrow S_{\text{value}}$ ($\mathbb{F}_3$ 上的符号错误)。 // 根据全局和推算错误值
        在位置 $\hat{\mathbf{i}}$ 处将 r 的值翻转 $-\hat{e}$ 以得到 $\hat{c}$。 // 将错误位还原
        return $\hat{c}$
    else
        return 检测到存在多位错误
else
    Let $\hat{\mathbf{i}} = (\hat{i}_{n-1}, \cdots, \hat{i}_1, \hat{i}_0) = (0, \cdots, 0, 0)$。 // 该情况下错误索引必定为 0
    令 $\hat{e} \leftarrow S_{\text{value}}$ ($\mathbb{F}_3$ 上的符号错误)。
    在位置 $\hat{\mathbf{i}}$ 处将 r 的值翻转 $-\hat{e}$ 以得到 $\hat{c}$。
    return $\hat{c}$

注. 设 $S_{idxXor} = \hat{i}_{n-1} \cdots \hat{i}_1 \hat{i}_0$, 则定位函数 $f_{\text{locate}}(S_{idxXor}) = (\hat{i}_{n-1}, \cdots, \hat{i}_1, \hat{i}_0)$

附录 C：背景与相关研究

C.1 A1 & A2 码交互演示页面

访问地址：https://sltracer.github.io/ECC_Paper_Website_Demo/index_SEC_TED_cn.html

C.2 参考文献

[1] R. W. Hamming，“纠错与检错码”，《贝尔系统技术期刊》，第 29 卷，第 2 期，第 147-160 页，1950 年。

[2] F. J. MacWilliams 和 N. J. A. Sloane，《纠错码理论》，North-Holland 出版社，1977 年。

[3] W. C. Huffman 和 V. Pless，《纠错码基础》，剑桥大学出版社，2003 年。

[4] S. Lin 和 D. J. Costello，《差错控制编码》（第 2 版），Pearson 出版社，2004 年。

[5] R. E. Blahut，《数据传输中的代数编码》，剑桥大学出版社，2003 年。

[6] J. H. van Lint，《编码理论导论》（第 3 版），Springer 出版社，1999 年。

[7] P. Elias，“列表解码的纠错码”，《IEEE 信息论汇刊》，第 37 卷，第 1 期，1991 年。

[8] G. D. Forney, Jr.，“级联码”，MIT 出版社，1966 年。

[9] M. J. E. Golay，“数字编码笔记”，《IRE 会刊》，第 37 卷，第 657 页，1949 年 6 月。