

基于 BCH 纠错算法的编解码器设计与实现 *

王 莞^{1,2}, 魏 敬 和^{1,2}, 于 宗 光^{1,2}

(1.江南大学 物联网工程学院, 江苏 无锡 214122; 2.中国电子科技集团第 58 研究所, 江苏 无锡 214072)

摘要: 随着 NAND Flash 存储单元的快速发展, 存储密度增加使得器件的出错概率增加, 为此提出了一种优化的 BCH 编解码器结构, 编码和解码过程每个时钟周期可以并行处理 16 位数据, 其中译码电路中的伴随式模块、错误位置多项式模块与钱氏(Chien)搜索模块采取三级流水线结构, 纠错和检错阶段可以同时进行, 有效地提高数据的处理速度和纠错速度。在完成电路的 RTL 设计后利用 VCS 工具完成了电路的仿真验证, 结果表明在传输 8 192 bit 数据生成 672 校检因子情况下实现了 48 位纠错, 工作频率最高支持 200 MHz。

关键词: NAND Flash; BCH 码; 钱氏搜索; 流水线结构; 编解码

中图分类号: TN492

文献标识码: A

DOI: 10.16157/j.issn.0258-7998.212214

中文引用格式: 王莞, 魏敬和, 于宗光. 基于 BCH 纠错算法的编解码器设计与实现[J]. 电子技术应用, 2022, 48(5): 42-46.

英文引用格式: Wang Guan, Wei Jinghe, Yu Zongguang. Design and implementation of codec based on BCH error correction algorithm[J]. Application of Electronic Technique, 2022, 48(5): 42-46.

Design and implementation of codec based on BCH error correction algorithm

Wang Guan^{1,2}, Wei Jinghe^{1,2}, Yu Zongguang^{1,2}

(1. School of IoT Engineering, Jiangnan University, Wuxi 214122, China;

2. China Electronics Technology Group Corporation No.58 Research Institute, Wuxi 214072, China)

Abstract: With the rapid development of NAND Flash memory cells and the increase in storage density, the error probability of devices has increased. For this reason, an optimized BCH codec structure is proposed. The encoding and decoding process can process 16-bit data in parallel in each clock cycle. Among them, the syndrome module, error location polynomial module and Chien search module in the decoding circuit adopt a three-stage pipeline structure, and the error correction and error detection stages can be carried out at the same time, which effectively improves the data processing speed and error correction speed. After completing the RTL design of the circuit, the simulation verification of the circuit was completed by using the VCS tool. The results showed that 48-bit error correction was achieved when 8 192 bit data was transmitted to generate 672 check factors, and the maximum operating frequency was 200 MHz.

Key words: nand flash; BCH code; Chien search; pipeline structure; codec

0 引言

Nand Flash 是一种非易失性存储器, 与 NOR Flash 相比具有读写速度快和存储密度高等优势, 但由于 NAND Flash 本身结构特点, 其存储单元出现数据位翻转现象比 NOR Flash 中更常见^[1], 与此同时, 随着 NAND Flash 技术的飞快发展, NAND Flash 从 SLC 结构发展为 MLC 结构及现在的 TLC 结构, 每个存储单元可以存储 2 bit 以至更多的数据, 使得数据位之间的相互干扰变大, 进而导致出错概率增大, 随着工艺水平的不断提高, 超深亚微米下的电荷效应进一步增加了数据出错的可能性。因此, 在对 NAND Flash 存储数据时, 必须采用更高的纠错技术, 以提高存储的稳定性。文献[2]中采用一种 8 位并

行 BCH 编解码器, 但因为电路并行处理数据少, 影响处理速度, 文献[3]中设计一种纠错 16 位的 BCH 编解码器, 但纠错位数较少。文献[4]中设计一种校正 32 位出错位的 BCH 编解码器, 相比较纠错位数有所增加, 但还不能满足大容量存储的数据校正。本文设计一种 16 位并行 BCH 编解码器, 并且具有最高 48 位纠错能力, 纠错速度和纠错能力都有了进一步的提高。

1 BCH 编码基本原理

矩阵 BCH 码是纠正多个随机错误的循环码, 可以用生成多项式 $g(x)$ 的根来描述。它是一种有限域中的线性分组码, 广泛应用于存储和通信领域中的编码。定义如下: 给定任一有限域 $GF(q)$ 及其扩域 $GF(q^m)$, 其中 q 是素数或素数的幂, m 为某一正整数。若任一码元取自 $GF(q^m)$ 上的循环码 (n, k) , 其中 $n=2^m-1$, 其生成多项式 $g(x)$ 具有

* 基金项目: 国家自然科学基金(62174150)

2t 个连续根 $\{a^1, a^2, \dots, a^{(2t-1)}, a^2\}$ 时, t 为纠正位数, 则由生成多项式 $g(x)$ 生成的循环码称为 q 进制 BCH 码, 记为 (n, k, t) ^[5]。最常用的 BCH 码通常为二进制的 BCH 码, 二进制 BCH 码的码元都是由 0 和 1 构成的, 便于硬件电路的实现。本文以下讨论的 BCH 码是二进制 BCH 码, 二进制本原 BCH 码有以下重要参数, 码长为 $n=2^m-1$, 信息长度为 k , 纠错位数量为 t , 校检码长度为 $mt=n-k$, 对于本设计中, $k=1024$ B, $B=8192$ bit, $t=48$, $n=8192+48\times14=8864$, 因此 BCH 码为 $(8864, 8192, 48)$, 其中, 校检码长度为 672 bit。

2 BCH 并行编码器设计

BCH 编码可以按如下方式构造, 假定信息多项式为:

$$m(x)=m_{k-1}x^{k-1}+m_{k-2}x^{k-2}+\dots+m_0 \quad (1)$$

其中, $m_i \in GF(2)$, BCH 码系数通常由计算机离线生成, 首先找到生成多项式 $g(x)$, 记为:

$$g(x)=g_{n-k}x^{n-k}+g_{n-k-1}x^{n-k-1}+\dots+g_1x+g_0 \quad (2)$$

则 BCH 编码后的码组 $c(x)$ 为:

$$c(x)=x^{n-k}m(x)+(x^{n-k}m(x)) \bmod g(x) \quad (3)$$

其中校检多项式为:

$$r(x)=(x^{n-k}m(x)) \bmod g(x) \quad (4)$$

由 $n-k=672$, 故 $r(x)=(x^{672}m(x)) \bmod g(x)$ 进行 16 位并行编码, 记 $k=L \cdot m$, $L=512$, 则有:

$$r(x)=(((M_{511}x^{16}+M_{510})x^{16}+\dots+M_1)x^{16}+M_0)x^{672} \bmod g(x) \quad (5)$$

可以得到迭代计算的关键方程式:

$$r_i(x)=(r_{i-1}(x)x^{16}+M_i) \bmod g(x) \quad (6)$$

所以 BCH 编码的实质就是有限域内以生成多项式为模的除法问题, 是将 $x^{n-k}m(x)$ 对生成多项式 $g(x)$ 求模, 可以通过线性移位寄存器(LFSR)来实现, 16 位并行编码器原理结构如图 1 所示。

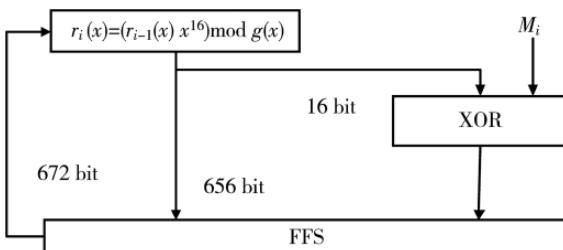


图 1 16 位并行编码器结构

16 位并行 BCH 编码器工作流程如下:

- (1) 在编码开始之前, 把所有寄存器初始化为 0;
- (2) 把 $r_i(x)=(r_{i-1}(x)x^{16}+M_i) \bmod g(x)$ 式做循环迭代运算, M_i 即为当前时刻要输入到 Nand Flash 中 16 位数据, 信息比特按照从高位到低位的方式依次进入。

(3) 迭代到 512 次的时候进入另一个阶段, 这时设置输入 $M_i=0$, 再继续迭代计算 42 次, 就完成并行编码的计算, BCH 编码寄存器储存的值即为最终校检位。

BCH 编码过程包含 3 个状态, 分别为空闲状态(IDLE)、

编码状态(ENC)、校检因子传输状态(WRITE)。在复位信号为低电平时, 状态机都被转入 IDLE 状态, 在时钟上升沿, 纠错使能信号 enable_ecc 为高电平, dir_sel 信号为低电平并且码字有效信号 data_in_valid 为高电平时转入 ENC 状态; 当 1024 B 数据编码完成后进入 WRITE 状态, 当完成校检因子传输后回到 IDLE 状态, 其状态转换图如图 2 所示。

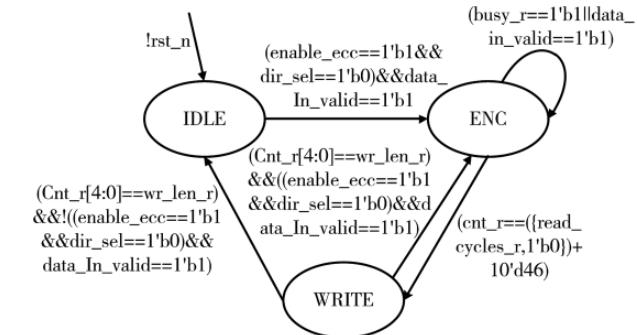
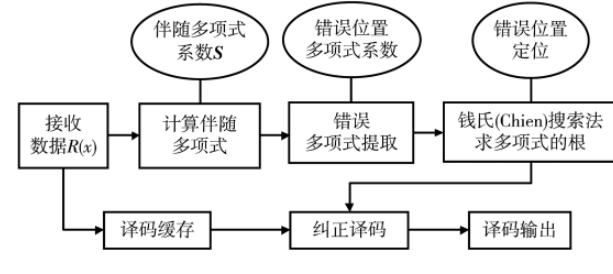


图 2 编码过程状态转换图

3 BCH 译码器设计

在所有的 BCH 译码算法中最常用是 Berlekamp – Massey 迭代算法(简称 BM 法), BCH 译码器首先通过计算校检因子实现错误检测, 然后通过无逆 BM 迭代算法和钱氏(Chien)搜索法实现纠错, 译码过程主要分为以下三个步骤完成^[6], BCH 译码流程如图 3 所示。



第一步: 由接收到的 $R(x)$ 计算伴随多项式 $S=(s_1, s_2, s_3, \dots, s_{2t})$, 求出伴随式系数。

第二步: 根据伴随多项式 S 求解错误多项式 $\sigma(x)=\sigma_t x^t + \sigma_{t-1} x^{t-1} + \dots + \sigma_1 x^1 + 1$, 进行错误多项式系数提取。

第三步: 求解 $\sigma(x)$ 的根, 利用钱氏(Chien)搜索法进行错误位置定位, 并进行错误纠正。

3.1 伴随多项式计算

假定发送的码字为:

$$C(x)=c_0+c_1x+c_2x^2+\dots+c_{n-1}x^{n-1} \quad (7)$$

接收到的码字多项式为:

$$R(x)=r_0+r_1x+r_2x^2+\dots+r_{n-1}x^{n-1} \quad (8)$$

假设错误多项式为:

$$E(x)=e_0+e_1x+e_2x^2+\dots+e_{n-1}x^{n-1} \quad (9)$$

理论上,任何一个接收码字 $R(x)$ 均可看作发送码字 $C(x)$ 与错误多项式 $E(x)$ 的叠加^[7],即:

$$R(x)=C(x)+E(x)=m(x)g(x)+E(x) \quad (10)$$

由于 $S^T=H^T$,其中 H 为校检多项式,因此伴随多项式可以表示为:

$$S^T = \begin{bmatrix} (a)^{n-1} & \cdots & (a)^{2t} & \cdots & a & 1 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ (a^2)^{n-1} & \cdots & (a^2)^{2t} & \cdots & a^2 & 1 \end{bmatrix} m(x)g(x) + \begin{bmatrix} (a)^{n-1} & \cdots & (a)^{2t} & \cdots & a & 1 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ (a^2)^{n-1} & \cdots & (a^2)^{2t} & \cdots & a^2 & 1 \end{bmatrix} e(x) \quad (11)$$

由于 $a^j (j=0, 1, 2, \dots, 2t)$ 是生成多项式 $g(x)$ 的根,因此,当 $x=a^j$ 时,可得到伴随多项式的系数为 $S_j=R(a^j)=m_j(a^j)g(a^j)+E_j(a^j)=E_j(a^j)$,伴随多项式 $S=(s_1, s_2, \dots, s_{2t})$ 的计算转换为 $R(x)$ 除以 $g(x)$ 的除法运算,相除之后的余式即为伴随式,如果伴随多项式中任意一个元素 S_j 不为 0,则表明有错误,否则没有错误。

伴随式电路硬件实现如下:

$$S_j=R(a^j)=\sum_{i=0}^{n-1} r_j(a^j)^i \quad (12)$$

上式展开得:

$$S_j=(\cdots(r_{n-1}(a^j)^{15}+r_{n-2}(a^j)^{14}+\cdots+r_{n-15}a^j+r_{n-16})(a^j)^{16}+\cdots+r_{16})(a^j)^{16}+r_{15}(a^j)^{15}+r_{14}(a^j)^{14}+\cdots+r_1a^j+r_0 \quad (13)$$

16 位并行伴随式电路结构如图 4 所示,每个时钟周期可以处理 16 bit 数据,其中,乘法器和加法器均为有限域乘法器和有限域的加法器,因为有限域的加法采用模 2 进行运算,故加法器为直接的异或运算。在 GF(2¹⁴) 域多项式的乘法过程和一般多项式乘法一样,在各项相加的时候按照模 2 规则进行,运算过程出现大于 a^{14} 项需要通过 GF(2¹⁴) 域上的本原多项式 $x^{14}+x^{10}+x^6+x^1$ 进行高位消除^[8]。首先计算 $S=(S_1, S_3, \dots, S_{2t-1})$,然后由于伴随系数满足 $(s_2=s_1^2, s_4=s_2^2, \dots, s_{2t}=(S_{2t-1})^2)$,通过计算得到 $S=(s_2, s_4, \dots, s_{2t})$ 。由此可得伴随多项式为:

$$S(x)=s_0+s_1x+s_2x^2+\cdots+s_jx^j \quad (14)$$

BCH 译码首先要进行伴随多项式求解,伴随多项式计算作为流水线的第一级,设计伴随式模块有三个状态

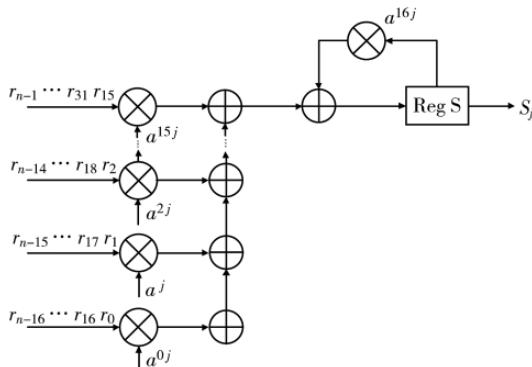


图 4 并行伴随式电路结构

分别为空闲状态(SYN_IDLE)、接收码字状态(READ)、伴随式计算状态(SYND_CALC),在复位信号为低电平时,无论伴随式模块在何种状态,都被转入 SYN_IDLE 状态,在时钟上升沿,纠错使能信号 enable_ecc 和码字有效信号 data_in_valid 为高电平时转入 READ 状态。当伴随式模块处 READ 状态,计数信号 cnt_syn_r 为 1024 时,进入 SYND_CALC 状态。当在伴随计算状态,如果 cnt_syn_r=6'b101110 并且 data_in_valid 为低电平时,表明伴随式计算完成,伴随式计算模块转入 SYN_IDLE 状态;如果伴随式系数有一个不为零,表明接收码字有错误,需要进入后续的错误位置多项式和钱氏搜索操作,伴随式计算状态转换如图 5 所示。

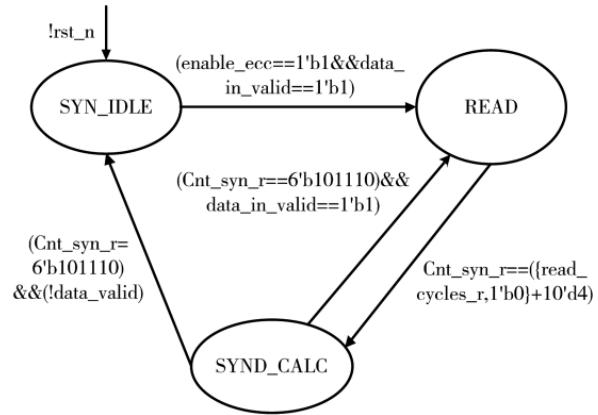


图 5 伴随多项式计算状态转换图

3.2 错误多项式提取

求解错误位置多项式是 BCH 译码过程中最重要的一步,首先构造错误位置多项式如下所示: $\sigma(x)=\sigma_0x^0+\sigma_1x^1+\cdots+\sigma_{t-1}x^{t-1}+\cdots+\sigma_1x^1+1$,其中,错误位置为该式的根,由数学推演,可知错误多项式的系数 $\sigma_t, \sigma_{t-1}, \dots, \sigma_2, \sigma_1$ 和伴随式有如下关系:

$$\begin{bmatrix} s_t & s_{t-1} & \cdots & s_1 \end{bmatrix} \begin{bmatrix} \sigma_0 \\ \vdots \\ \sigma_t \end{bmatrix} + \begin{bmatrix} s_{t+1} \\ \vdots \\ s_2 \end{bmatrix} = 0 \quad (15)$$

$$S(x)\sigma(x)=1+\sigma_2x^2+\sigma_4x^4+\cdots+\sigma_{2t-2}x^{2t-2}+\sigma_2x^2 \quad (16)$$

将 $S(x)$ 和 $\sigma(x)$ 带入上面方程式得到 $\sigma(x)$ 的关键方程^[9]:

$$S(x)\sigma(x)=\omega(x)(\bmod x^{2t+1}) \quad (17)$$

由于 BM 算法迭代效率,易于硬件实现,本文采用基于二进制 BCH 码的无求逆 BM 迭代算法求解该方程,首先选择一组合理的初始值 $\sigma^{(0)}(x)$ 和 $\omega^{(0)}(x)$,经过第一次迭代运算求得 $\sigma^{(1)}(x)$ 和 $\omega^{(1)}(x)$ ^[10],依次类推,由 $\sigma^{(i)}(x)$ 和 $\omega^{(i)}(x)$ 求得 $\sigma^{(i+1)}(x)$ 和 $\omega^{(i+1)}(x)$ 。迭代过程中定义第 $j+1$ 和第 j 步的差值为 d_j ,则有:

$$d_j=s_{j+1}+\sum_{i=0}^{\partial\sigma^{(j)}(x)} s_{j+1-i}\sigma_i^{(j)} \quad (18)$$

式中 $\sigma_i^{(j)}$ 是经过第 j 次迭代后得到的 x^i 项的系数有

$$\sigma^{(j)}(x) = 1 + \sigma_1^{(j)}x + \sigma_2^{(j)}x^2 + \cdots + \sigma_{D(j)}^{(j)}x^{D(j)}, \text{且 } D(j) = \partial_0 \sigma^{(j)}(x)。$$

二进制 BCH 码迭代算法可以优化为如下步骤：

(1) 初始值 $\sigma^{(0)}(x) = 1, D(0) = 0, d_0 = s_1$ 开始迭代。

(2) 计算 $d_j = s_{2j+1} + \sum_{i=1}^{\partial_j \sigma^{(j)}(x)} s_{2j+1-i} \sigma_i^{(j)}$, 若 $d_j = 0$, 则有 $\sigma^{(j+1)}(x) = \sigma^{(j)}(x), D(j+1) = D(j)$; 若 $d_j \neq 0$, 则找出 j 之前的一行 i , 它的 $2i - D(i)$ 最大, 且 $d_i \neq 0$, 然后计算 $\sigma^{(j+1)}(x) = \sigma^{(j)}(x) - d_j d_i^{-1} x^{2(j-i)} \sigma_i^{(j)}(x)$ ^[12]。

(3) 计算 d_{j+1} , 重复步骤(2)进行迭代, 经过 t 次迭代后得到 $\sigma^{(t)}(x)$ 即为所求的 $\sigma(x)$ 。

3.3 钱氏搜索法定位错误位置

通过上面的 BM 算法, 能够得知错误位置多项式为 $\sigma(x) = \sigma_t x^t + \sigma_{t-1} x^{t-1} + \cdots + \sigma_1 x^1 + \sigma_0$, 如果直接对上式进行因式分解^[13], 将得到 $\sigma(x)$ 的分解等式:

$$\sigma(x) = \prod_{i=1}^t (1 - \beta_i x) = (1 - \beta_1 x)(1 - \beta_2 x) \cdots (1 - \beta_t x) \quad (19)$$

随着 t 的值变大时, 会发现求 $\sigma(x)$ 的因式分解越来越困难, 因此通常采用钱氏搜索法求出上式方程的根。钱氏搜索法原理如下^[14], 当 $x = 1/\beta_i$ 时, $\sigma(x) = 0$, 其中 x 的取值范围为 $\alpha^i, i = n-1, n-2, \dots, 0$ 。因此, 将 α^i 依次带入 $\sigma(x)$, 如果 $\sigma(x) = 0$, 则 α^i 为根, 说明第 i 位发生了错误^[15]。

BCH 译码过程第三级通过钱氏搜索法进行错误位置定位并进行纠错, 钱氏搜索过程包含四个状态分别为钱氏搜索空闲状态(CHIEN_IDLE)、接收码状态(CHIEN_NC)、搜索状态(CHIEN)、错误信息发出状态(STIG)。开始 chien_state_next 寄存器为 SYND_IDLE 状态, 在时钟上升沿, chien_err_r 信号变为高电平后, 状态机跳转到 CHIEN_NC 状态。当 cnt_chi_r 搜索计数信号满足时, chien_state_next

为 CHIEN 状态; 在钱氏搜索找到错误总数 chien_cnt_err_r 信号等于 cnt_exp_err_r 信号时, 完成钱氏搜索 chien_state_next 进入 STIG 状态, 在下一个时钟上升沿 dec1_full 信号满足为低电平时, chien_state_next 回到 CHIEN_IDLE 状态, 钱氏搜索状态转换如图 6 所示。

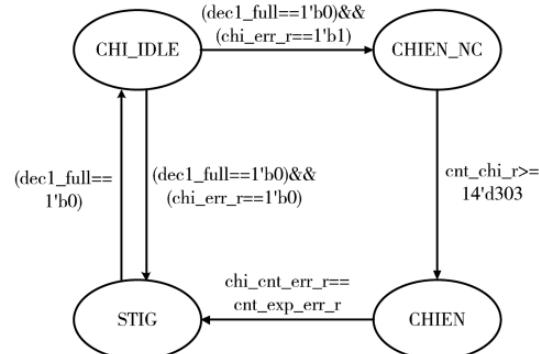


图 6 钱氏搜索状态转换图

4 仿真与验证结果分析

利用 Verilog 语言完成上述电路模块的 RTL 级设计, 在 Linux 系统下利用 VCS 工具进行电路的仿真与验证。发送 8 192 比特信息位, 通过 BCH 并行编码器电路生成 672 位校检码, 编码过程仿真波形图如图 7 所示。完成编码后本文通过后门访问 FIFO 寄存器把 1、3、5、7、9、11 字节的数据翻转, 共产生 48 位错误, 然后再通过 BCH 译码器分别进行伴随式因子的计算, 错误多项式系数提取, 最后通过 Chien 搜索法找到发生了错误位置的定位并正了错误值, 伴随多项式计算与错误多项式提取仿真波形图如图 8 所示, 钱氏搜索仿真如图 9 所示。

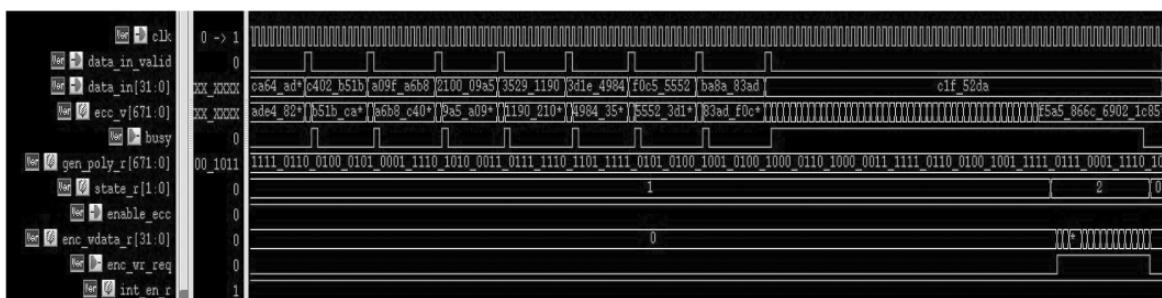


图 7 并行编码过程仿真波形图

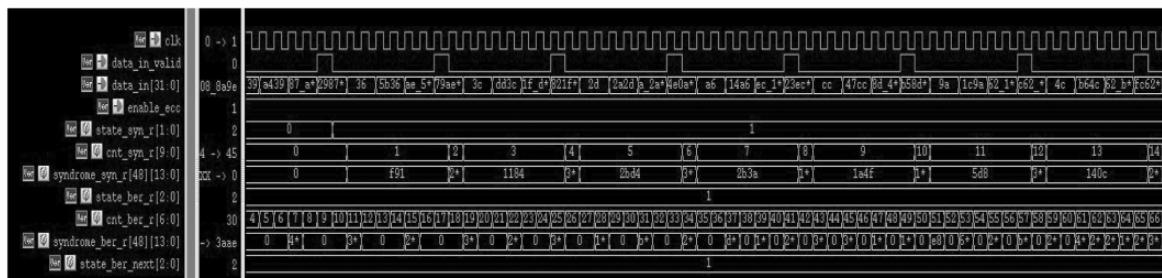


图 8 伴随多项式计算与错误多项式提取仿真波形图



图 9 钱氏搜索仿真波形图

5 结论

本文对编译码电路采用高效 16 并行电路结构,与此同时译码过程采用流水线结构,BCH 解码器可以在从内存中读取数据时检测错误,提出了一种对于 Nand Flash 的 48 bit 纠错能力解决方案,同时减少资源的消耗,减少编解码和解码周期。与其他文献中编解码中的数据传输速度、工作频率与纠错能力对比如表 1 所示。今后在此基础上研究 AES 加密算法,实现对数据的加密和解密方面有进一步研究。

表 1 纠错能力及频率对比

	本文	文献[2]	文献[3]	文献[4]
数据速率	16 位并行	8 位并行	8 位并行	8 位并行
频率/MHz	200	50	125	
纠错能力	48 位	32 位	16 位	32 位

参考文献

- [1] 雷水艳,焦继业,陈亚南.一种优化的 BCH 编解码器的设计与实现[J].计算机与数字工程,2019,47(9):2335-2338.
- [2] 程文韶.一种基于 BCH 码的 NAND Flash 控制器的研究与设计[D].武汉:华中科技大学,2013:42-55.
- [3] 蔡二龙.并行 BCH 编解码的快速实现方法[D].西安:西安电子科技大学,2015:42-53.
- [4] 郭晨阳.一种纠错 32 位 BCH 码的硬件实现[J].信息技术,2015(5):117-120.
- [5] 陈昭林,张晋宁,沈辉.基于 BCH 码的 NAND Flash 纠错算法设计与实现[J].电子测量术,2017,40(3):127-132.
- [6] 廖宇翔.基于 NAND flash 主控制器的 BCH 纠错算法设计与实现[D].哈尔滨:哈尔滨工业大学,2014.
- [7] 许芮铭.基于 SOPC 的 NAND Flash 控制器设计[D].成都:电子科技大学,2016:30-45.
- [8] 龙永萍.基于 UVM 的 Flash 控制器模块验证[D].杭州:杭州电子科技大学,2018:22-28.

- [9] 蒙宇霆,袁海英,丁冬.基于 AXI 总线的可配置 LVDS 控制器设计与验证[J].电子技术应用,2021,47(6):40-45,56.
- [10] 刘洋,李杰,李金强,等.一种基于 RCRF+BCH 算法的 NAND FLASH 纠错方案的 FPGA 设计与实现[J].空军工程大学学报(自然科学版),2020,21(6):46-52.
- [11] LI T, LEI Z. A novel multiple dies parallel NAND Flash memory controller for high-speed data storage[C]//International Conference on Electronic Measurement & Instruments(ICEMI). Beijing: IEEE, 2017: 6-11.
- [12] KUMAR A, ARDESHANA J, JAGTAP S. Design & verification of ONFI compliant high performance NAND flash-controller[C]//International Conference on Recent Trends in Electronics Information & Communication Technology (RTEICT). IEEE, 2016: 942-945.
- [13] POTEY S D, DHANDE P M. Error detection and correction capability for BCH encoder using VHDL[C]//IEEE International Conference for Convergence in Technology(I2CT). IEEE, 2019: 1-4.
- [14] SHIEH S. Concatenated BCH and LDPC coding scheme with iterative decoding algorithm for Flash Memory[J]. IEEE Communications Letters, 2015, 19(3): 327-330.
- [15] XIE Z, ZHANG X. Reduced-complexity key equation solvers for generalized integrated interleaved BCH decoders[J]. IEEE Transactions on Circuits and Systems, 2020, 67(12): 5520-5529.

(收稿日期:2021-10-09)

作者简介:

王莞(1995-),男,硕士研究生,主要研究方向:SoC 设计。

魏敬和(1975-),男,博士,研究员,主要研究方向:系统集成电路、可重构 ASIC 电路。

于宗光(1964-),男,博士,研究员,主要研究方向:集成电路设计、硬件木马检测。



扫码下载电子文档

版权声明

经作者授权，本论文版权和信息网络传播权归属于《电子技术应用》杂志，凡未经本刊书面同意任何机构、组织和个人不得擅自复印、汇编、翻译和进行信息网络传播。未经本刊书面同意，禁止一切互联网论文资源平台非法上传、收录本论文。

截至目前，本论文已经授权被中国期刊全文数据库（CNKI）、万方数据知识服务平台、中文科技期刊数据库（维普网）、DOAJ、美国《乌利希期刊指南》、JST 日本科技技术振兴机构数据库等数据库全文收录。

对于违反上述禁止行为并违法使用本论文的机构、组织和个人，本刊将采取一切必要法律行动来维护正当权益。

特此声明！

《电子技术应用》编辑部

中国电子信息产业集团有限公司第六研究所