

基于FPGA的NAND FLASH纠错编码方案与实现^{*}

孙晓磊 王红亮

(中北大学 电子测试技术国家重点实验室 太原 030051)

摘要: 为了解决 NAND FLASH 存储器数据读写过程中产生误码的问题,提出了一种基于 FPGA 的 NAND FLASH 纠错编码方案。方案采用 Micron 公司的 MT29F64G08ABAAA 存储芯片并由 FPGA 控制器控制进行数据存储,每 64 Bytes 产生 18 bits 汉明码校验码,将写入 FLASH 中的汉明码校验码和读取数据时生成的汉明码校验码比较便可确定误码发生的位置,对该数据位进行取反便可对误码进行纠错。测试结果表明,写入和读取 8 GBytes 数据若不采用纠错编码方案则在高低温和常温下测试均存在误码,采用纠错编码方案在 20 ℃、60 ℃和 -40 ℃环境温度下测试均无误码,实现了对 NAND FLASH 纠错的功能,提高了数据存储的准确度和稳定性。

关键词: FLASH;汉明码;FPGA;误码;编码;纠错

中图分类号: TP333.5 **文献标识码:** A **国家标准学科分类代码:** 510.99

NAND FLASH error correcting coding scheme and implementation
based on FPGA

Sun Xiaolei Wang Hongliang

(National Key Laboratory for Electronic Measurement Technology, North University of China, Taiyuan 030051, China)

Abstract: In order to solve the problem of error codes in the process of reading and writing NAND FLASH memory data, an FPGA-based NAND FLASH error correction coding scheme is proposed. The solution uses Micron's MT29F64G08ABAAA memory chip and is controlled by the FPGA controller for data storage. Every 64 Bytes generates 18 bits Hamming code check code, and writes the Hamming code check code in FLASH and the Hamming code check code generated when reading the data. The code comparison can determine the position where the error code occurs, and the error code can be corrected by inverting the data bit. The test results show that if the error correction coding scheme is not used for writing and reading 8 GBytes data, there are errors in the test at high and low temperatures and normal temperature. The error correction coding scheme is tested at 20 ℃, 60 ℃ and -40 ℃. Error-free, realizes the function of NAND FLASH error correction, and improves the accuracy and stability of data storage.

Keywords: FLASH; Hamming code; FPGA; error code; coding; error correction

0 引言

图像传感器技术的发展日新月异,图像传感器单位时间内产生的数据量越来越大,NAND FLASH 由于写入干扰和读取干扰的存在会使数据发生误码,如何实时正确存储采集回来的海量图像数据仍是近年来国内外许多研究机构 and 高校研究的热点之一^[1]。国内陈昭林等^[2]实现了基于 BCH 码的 NAND FLASH 纠错算法的设计;中北大学贾刘彬^[3]提出了汉明码校验码纠错方法进行 FLASH 数据纠错,每 512 Bytes 有 1 bit 误码可对其进行纠错;刘航航等^[4]

进行了 NAND FLASH 的关键技术研究,对比并提出了不同 NAND FLASH 架构应采用不同的纠错方法。国外 Lee 等^[5]根据 BCH 编解码器不会同时工作的原理提出了一种共享编码器和校正子模块的电路,降低了 BCH 解码器的硬件消耗;韩国 Kim 等^[6]提出了一种基于 LDPC 解码方案的重新编程方案,LDPC 解码方案在不擦除 FLASH 的情况下实现数据纠错,使用未纠错的信息来提高 LDPC 解码器的性能。本文介绍的基于 FPGA 的 NAND FLASH 纠错编码方案采用汉明码校验码纠错算法,相对于 BCH 码纠错算法实现方法更简单,对 NAND FLASH 数据读写速度影

收稿日期:2021-06-09

^{*} 基金项目:山西省“1331 工程”重点学科建设计划(1331KSC)项目资助

响更小;相对于 LDPC 解码方案来说纠错能力更强;相对于 512 Bytes 纠错 1 bit 的汉明码校验码纠错,本方案采用的 64 Bytes 数据纠错 1 bit 的方案纠错率更高,现对 NAND FLASH 纠错编码方案进行深入研究。

1 FLASH 纠错编码系统设计

基于 FPGA 的 NAND FLASH 纠错编码方案主要实现对 NAND FLASH 的数据写入和读取过程中产生的误码进行纠错,计算机可以通过千兆以太网接口进行命令的发送和数据的回读。系统设计包括硬件设计和程序设计,硬件设计包括主控板设计和电源板设计;程序设计主要实现 FLASH 数据存储、千兆以太网数据回读、汉明码编解码与数据纠错等功能。

1.1 数据存储硬件系统设计

硬件系统整体设计原理如图 1 所示,包括电源板和主控板。

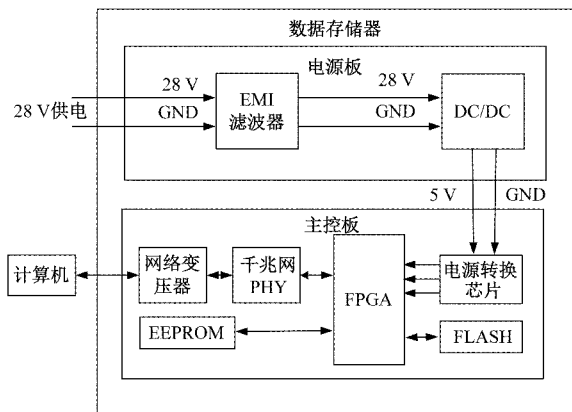


图 1 硬件系统整体设计原理

电源板采用 EMI 滤波器对供电电源进行滤波,防止纹波对系统产生干扰,采用 DC/DC 模块将 28 V 电压转换为 5 V 电压进行系统供电。主控板中采用 Xilinx 公司的 Spartan6 系列的 FPGA 处理器;千兆以太网的 PHY 芯片选用 88E1111,该千兆以太网芯片具有 MDI/MDIX 自动反转、极性自动校正以及双工/速率自动协商功能;选用 25LC640AT-I/SN 型号的 EEPROM 来存储千兆以太网接口的 IP 地址和端口号;Micron 公司的 MT29F64G08ABAAA 系列的 FLASH 存储芯片由于其读写速度快且在高温和低温的环境下可以稳定工作,因此本方案选用该型号的 FLASH 进行数据存储。

1.2 FLASH 控制与数据传输程序设计

本设计采用 FPGA 集成开发工具 ISE 进行程序的编写。程序总体设计原理如图 2 所示,信号源模块产生带有帧计数和帧结束标志的递增数以方便数据读取出来的数据分析;FIFO 用来缓存数据;编码模块将进出 FLASH 的数据生成汉明码校验码;解码模块将进出 FLASH 的数据生成的汉明码校验码进行比较,从而确定进出 FLASH 的

数据是否发生了误码以及误码发生的位置;纠错模块实现将数据从前边的 FIFO 读取数据并且进行纠错,然后传输到后边的 FIFO 中;千兆以太网控制程序实现千兆以太网和电脑的通信,包含数据发送和命令接收与解析。

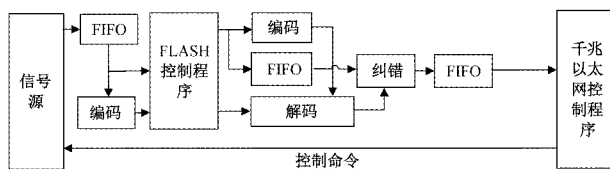


图 2 程序总体设计原理

FLASH 控制程序主要控制 FLASH 擦除和数据的读写。其工作流程如图 3 所示,系统上电后先对系统复位,然后进行坏块检测,将检测到的坏块地址存储到 RAM 中。等待擦除或者读数命令,若收到擦除命令则进行 FLASH 擦除,则判断 FLASH 的第 1 块是否为坏块,若为坏块则块地址加 1 并且判断下一块,否则判断前边 FIFO 中的数据是否有 8 kBytes 数据,若有 8 kBytes 的数据则将数据及其生成的汉明码存储到 FLASH 中,然后页地址加 1,每页依次写数据直至整块写完,每块依次写数据直至整片 FLASH 写满或者无数据需要存储;若收到读数命令,则首先判断 FLASH 的第 1 块是否为坏块,若为坏块则块地址加 1 并且判断下一块,否则读取 8 kBytes 的数据和其对应的汉明码校验码,每页依次读取直至整块读完,每块依次读取直至整片 FLASH 的数据读取完成。

2 纠错原理与 FPGA 实现

FLASH 主要分为 NAND FLASH、NOR FLASH 和 AG-AND FLASH,其中 NAND FLASH 根据技术不同可分为 SLC、MLC 和 TLC 3 种类型^[7],采用 SLC 技术研制的 FLASH 在一个存储单元中可以存储 1 bit 数据,采用 MLC 技术研制的 FLASH 在一个存储单元中可以存储 2 bits 数据,而采用 TLC 技术研制的 FLASH 在一个存储单元中可以存储 3 bits 数据^[8]。在 NAND FLASH 中对某一页进行读取和写入时,由于存储单元受到邻近存储单元升高电压的影响,使相邻的存储单元中已经存储的数据发生改变^[9],随着 NAND FLASH 工艺的发展与进步,同样大小的晶片上可以制造更多的存储单元,由此造成的写入干扰和读取干扰也越来越严重,因此需要采用 ECC 算法进行数据纠错^[10]。若采用 SLC 工艺研制生产的存储芯片,在写入干扰和读取干扰存在的情况下一次干扰只会使 1 bit 数据发生误码,在这种情况下采用汉明码可以有效降低误码率^[11]。若采用汉明码对 1 bit 误码进行纠错,则 64 Bytes 的数据会产生 16 bits 汉明码校验码。若采用 MLC 或 TLC 工艺生产的存储芯片,则在写入干扰和读取干扰存在的情况下一次干扰会使 2 bits 或 3 bits 数据发生误码,需要更为复杂的 BCH 校验码对误码进行纠错^[12]。若对 64 Bytes 中的 2 bits 或者 3 bits 误码进行纠错,则需 BCH 校验码 511 bits。因此采

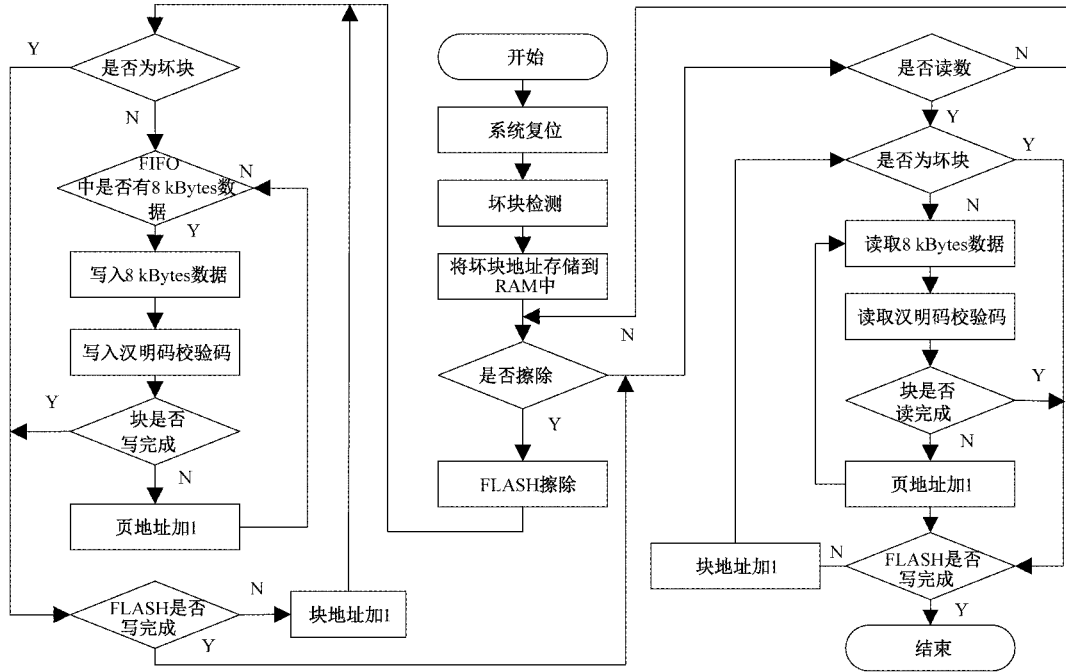


图3 FLASH 控制程序工作流程

用 BCH 校验码进行多 bit 纠错产生的校验码位数多,对 NAND FLASH 存储芯片的存储速率影响较大^[13]。由于本设计采用的 Micron 公司研制生产的 MT29F64G08ABAAA 存储芯片是采用 SLC 工艺研制生产的,因此采用汉明码校验码可以在对存储速率影响较小的情况下进行数据纠错。

2.1 编码原理

汉明码(Hamming code)是基于奇偶校验改进而来的,通过在数据后增加校验位来进行数据纠错^[14]。与奇偶校验不同的是,汉明码校验不仅可以判断数据的有效性,还可以确定数据误码的位置以达到数据纠错的作用,因此被广泛应用于数据存储器件中的数据纠错^[15]。

本设计采用 Micron 公司的 MT29F64G08ABAAA 芯片进行数据存储,该存储器每页有 8 192 Bytes 的存储空间可以用来存储数据,每页后有 448 Bytes 的附加存储空间用来存储坏块标志和校验码。

综合存储速率和纠错率等因素,本次设计以 64 Bytes 为一组形成一个列为 64、行为 8 的矩阵进行汉明码校验码的生成,生成的行或列汉明码校验码的位数计算公式为:

$$m = 2\log_2 n \quad (1)$$

其中, n 为矩阵中的行数或列数, m 为生成的行汉明码校验码位数或列汉明码校验码位数。

根据式(1)可以计算出每组数据生成的行校验码为 6 bits,每组数据生成的列校验码为 12 bits,将这 18 bits 校验码存储在 3 Bytes 的存储空间。这样每页数据就需要 384 Bytes 的存储空间用来存放校验码数据,可以在 64 Bytes 数据中有 1 bit 误码的情况下进行数据纠错,若 32 Bytes 数据生成一次汉明码校验码则需要 738 Bytes 存

储空间来存储校验码,这样每页后附加的存储空间不够;若 128 Bytes 生成一次汉明码校验码则降低了误码纠正率^[16]。因此,采用 64 Bytes 数据生成一次汉明码校验码可以在充分利用附加存储空间且对读写速度影响较小的情况下提高误码纠正率。

如图 4 所示,6 个列校验码用 $cc0 \sim cc5$ 表示,式(2)中 n 的范围为 $0 \sim 7$, $Bi(n)$ 为每一列中的各元素求异。

$$Bi(n) = [Bit(n), Byte0] \oplus [Bit(n), Byte1] \oplus [Bit(n), Byte2] \oplus \dots \oplus [Bit(n), Byte63] \quad (2)$$

$cc0 \sim cc5$ 为对各列求出的异或结果进行计算,如式(3)~(8)所示,从而求出列校验码,计算方法可参考图 4。

$$cc0 = Bi0 \oplus Bi2 \oplus Bi4 \oplus Bi6 \quad (3)$$

$$cc1 = Bi1 \oplus Bi3 \oplus Bi5 \oplus Bi7 \quad (4)$$

$$cc2 = Bi0 \oplus Bi1 \oplus Bi4 \oplus Bi5 \quad (5)$$

$$cc3 = Bi2 \oplus Bi3 \oplus Bi6 \oplus Bi7 \quad (6)$$

$$cc4 = Bi0 \oplus Bi1 \oplus Bi2 \oplus Bi3 \quad (7)$$

$$cc5 = Bi4 \oplus Bi5 \oplus Bi6 \oplus Bi7 \quad (8)$$

其中,12 个行校验码用 $rc0 \sim rc11$ 表示。式(9)中 n 的范围为 $0 \sim 63$, $By(n)$ 为每一行中的各元素求异或。

$$By(n) = [Bit0, Byte(n)] \oplus [Bit1, Byte(n)] \oplus [Bit2, Byte(n)] \oplus \dots \oplus [Bit7, Byte(n)] \quad (9)$$

$rc0 \sim rc11$ 为对各行求出的异或结果进行计算,从而求出对应的行校验码, $rc0 \sim rc3$ 的计算方法如式(10)~(13)所示, $rc4 \sim rc11$ 计算方法可参考图 4 进行计算。

$$rc0 = By0 \oplus By2 \oplus By4 \oplus By6 \oplus \dots \oplus By56 \oplus By58 \oplus By60 \oplus By62 \quad (10)$$

$$rc1 = By1 \oplus By3 \oplus By5 \oplus By7 \oplus \dots \oplus By57 \oplus By59 \oplus By61 \oplus By63 \quad (11)$$

$$rc2 = By0 \oplus By1 \oplus By4 \oplus By5 \oplus \dots \oplus By56 \oplus By59 \oplus By62 \oplus By63 \quad (13)$$

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0								
Byte0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	By0	rc0	rc2	rc4	rc6	rc10		
Byte1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	By1	rc1						
Byte2	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	By2	rc0	rc3					
Byte3	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	By3	rc1						
Byte4	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	By4	rc0	rc2	rc5				
Byte5	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	By5	rc1						
Byte6	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	By6	rc0	rc3					
Byte7	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	By7	rc1						
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
Byte56	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	By56	rc0	rc2	rc4	rc7	rc11		
Byte57	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	By57	rc1						
Byte58	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	By58	rc0	rc3					
Byte59	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	By59	rc1						
Byte60	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	By60	rc0	rc2	rc5				
Byte61	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	By61	rc1						
Byte62	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	By62	rc0	rc3					
Byte63	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	By63	rc1						
	Bi7	Bi6	Bi5	Bi4	Bi3	Bi2	Bi1	Bi0								
	cc1	cc0	cc1	cc0	cc1	cc0	cc1	cc0								
	cc3		cc2		cc3		cc2									
	cc5				cc4											

图 4 汉明码行列校验图

2.2 解码原理

在汉明码解码过程中需要对读取出来的数据重新生成汉明码校验码,然后将重新生成的汉明码校验码和存储在 NAND FLASH 附加存储空间的汉明码校验码进行对比就可以确定 FLASH 存储数据是否发生了单 bit 误码或多 bit 误码,若发生单 bit 误码,则可以确定其误码的具体位置及对其进行纠错^[17]。

其具体计算方法为,将读取出来的汉明码校验码与读取出来的数据重新生成的汉明码校验码进行异或,若计算的结果为 0,则表明数据存储过程中没有发生误码;若计算结果中 18 个行列校验码中有 9 个校验码为 1,则表明 FLASH 数据存储过程中有 1 bit 的数据发生了误码,可以进行纠正;若出现其它情况,则可能是多 bit 误码或者汉明码校验码发生了误码,无法进行纠正。

在 6 bits 列校验码中,若有 1 bit 数据发生了误码,则存储前后 $cc4$ 和 $rc5$ 的计算结果必有一个为 1,同理, $cc2$ 和 $cc3$ 、 $cc0$ 和 $cc1$ 中各自的对应分组中也必有一个为 1,通过此二分法就可以确定误码具体发生在了哪一行。同理,依照此方法可以确定误码所在的行,得知了行位置和列位置后只需要对该位置的数据求反即可对其进行纠正。其对应的行位置计算公式如式(14)所示,列位置计算公式如

式(15)所示。

$$er_cc_addr = cc(5) \& cc(3) \& cc(1) \quad (14)$$

$$er_rc_addr = rc(11) \& rc(9) \& rc(7) \& rc(5) \& rc(3) \& rc(1) \quad (15)$$

其中, er_cc_addr 为一个误码的 3 位列地址, er_rc_addr 为一个误码的 6 位行地址。

2.3 FPGA 实现过程与分析

通过 ISE 开发工具以及 VHDL 语言编写程序并且进行测试,基于 FPGA 的 NAND FLASH 纠错编码方案中的核心模块主要有 4 个,FLASH 控制程序模块主要实现数据和汉明码校验码的写入和读取,编码模块将数据以 64 字节为一组生成汉明码校验码,解码模块通过进出 FLASH 的汉明码校验码对比从而确定误码所在位置,而纠错模块是对应误码位置的数据进行纠正。

将数据写进 FLASH 中进行存储,每 64 Bytes 产生一组汉明码校验码,其中列汉明码校验码有 6 bits(cc),行汉明码校验码有 12 bits(rc),将他们存储在 3 Bytes 的存储空间中,第 1 个字节高 6 位存放 cc ,第 2 个字节高 6 位存放 rc 的高 6 位,第 3 个字节的高 6 位存放 rc 的低 6 位。如图 5 所示为采用 ISE 的 Chipscope 工具抓取到的第 1 个误码的行汉明码校验码和列汉明码校验码。

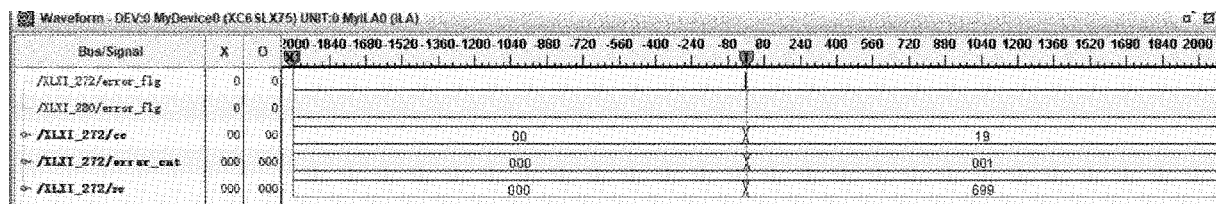


图 5 第 1 个误码对应的行列汉明码校验码

当FLASH数据读数时出现了第1个误码,对应的 $cc=19(16进制)=011001(二进制)$, $rc=699(16进制)=011010011001(二进制)$,对应的列地址为 $er_cc_addr=cc(5)\&cc(3)\&cc(1)=010=2$,行地址为 $er_rc_addr=rc(11)\&rc(9)\&rc(7)\&rc(5)\&rc(3)\&rc(1)=011010=26$,即第26个字节(从0字节开始计数)第2位反了(从0位开始计数)。然后停止数据读取,对已经读取出来的数据进行数据分析和数据查看。数据软件分析后指出了误码的位置及其原始数据和误码数据,图6为通过HexEdit查看的上位机回读的原始数据。

08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F	70	71	72	73	74	75	76	77	78	79
82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F	90	91	92	93	94	95	96	97	98	99
A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9
C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9
E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9
02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19
22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30	31	32	33	34	35	36	37	38	39
42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55	56	57	58	59
62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F	70	71	72	73	74	75	76	77	78	79
82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F	90	91	92	93	94	95	96	97	98	99
A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9
C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9
E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9
02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19
22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30	31	32	33	34	35	36	37	38	39
42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55	56	57	58	59
62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F	70	71	72	73	74	75	76	77	78	79
82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F	90	91	92	93	94	95	96	97	98	99
A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9
C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9
E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9
02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19
22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30	31	32	33	34	35	36	37	38	39
42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55	56	57	58	59
62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F	70	71	72	73	74	75	76	77	78	79

图6 上位机回读的原始数据

根据图6可以看出,数据设置为32 Bytes一行进行查看,数据在1A(26)的位置应为B4,实为B0,也就是第2位发生了误码,和汉明码计算出来的一致,该现象证明汉明码校验码可以检测到误码及其对应的位置。在程序中加入纠错模块,将行列地址对应的数据位取反便可以进行数据纠错。

3 测试结果与分析

3.1 数据读写测试结果

为了测试NAND FLASH数据读写过程中是否产生误码以及误码发生的数量,用带有帧计数和帧结束标志的递增数将8 GBytes的NAND FLASH写满,同时在NAND FLASH中存储汉明码校验码。NAND FLASH写满以后进行读取数据测试,在读取数据的过程中经过编码、解码以及纠错程序进行误码检测以及误码纠错。数据读取出来经过数据分析软件进行数据分析,从而确定误码发生的数量,如图7所示为加上汉明码校验码编码纠错功

能的数据分析结果,8 GBytes的数据读取出来经数据分析软件分析没有误码,可以说明基于FPGA的NAND FLASH纠错编码方案可以对误码进行纠正。



图7 加上汉明码校验码数据分析结果

3.2 高低温试验与存储数据误码分析

为了测试恶劣环境对NAND FLASH在有无汉明码校验码纠错功能情况下误码率的影响,分别测试了有无汉明码校验码纠错功能的不同温度下的误码,进行高低温循环试验的温度范围为 $-40^{\circ}\text{C}\sim+60^{\circ}\text{C}$,其中温度变化率为 $10^{\circ}\text{C}/\text{min}$,高低温和常温测试是以热透为原则分别在不同温度下通电2 h后进行读写测试,共做了12轮循环试验,其中前8轮为缺陷剔除试验,后4轮为无故障检验循环试验。图8为高低温试验环境,表1为有汉明码校验码不同温度下的12次误码测试,表2为无汉明码纠错不同温度下的12次误码测试。



图8 高低温试验环境

表1 有汉明码纠错高低温测试误码

温度/ $^{\circ}\text{C}$	测试次数											
	1	2	3	4	5	6	7	8	9	10	11	12
60	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0
-40	0	0	0	0	0	0	0	0	0	0	0	0

表2 无汉明码纠错高低温测试误码

温度/℃	测试次数											
	1	2	3	4	5	6	7	8	9	10	11	12
60	76	82	58	66	61	72	65	75	64	68	62	73
20	35	42	43	36	45	39	37	40	37	44	41	43
-40	84	78	99	88	92	93	89	80	91	82	90	79

试验结果表明,表1为有汉明码校验码不同温度下的12次误码测试,误码率均为0;表2为无汉明码纠错不同温度下的12次误码测试,高温60℃的平均误码率为 7.97×10^{-9} ,常温20℃的平均误码率为 4.68×10^{-9} ,低温-40℃的平均误码率为 10.14×10^{-9} 。因此,低温比高温对NAND FLASH的误码率影响更大,而常温下的误码率相对更低。相对于无汉明码校验码纠错的NAND FLASH数据读写测试,有汉明码校验码可以有效降低误码率,可以适应恶劣的应用环境,提高了数据存储系统的环境适应性。该设计每64 Bytes可以对1 bit数据进行纠错,因此最高误码纠错率可达 1.95×10^{-3} 。通过表中的数据及试验数据分析可以得知,将汉明码校验码用于NAND FLASH数据存储领域可以显著提高数据存储系统的稳定性和可靠性。

4 结 论

本文针对NAND FLASH存储器存储数据的过程中产生误码的问题,结合NAND FLASH的不同技术类型,设计了基于FPGA的NAND FLASH纠错编码方案。该方案以FPGA为控制核心进行数据产生、控制FLASH数据存储和回读。系统每64 Bytes生成18 bits汉明码校验码,18 bits汉明码校验码经过编码、解码和对比特纠错便可纠正1 bit误码,因此最高误码纠错率最多可达 1.95×10^{-3} 。实验结果表明,加上汉明码校验码纠错功能的存储器在高低温和常温下均无误码,有效提高了NAND FLASH存储系统的稳定性和可靠度。采用汉明码校验码进行纠错相比于BCH码纠错来说实现方法简单,但是也存在不足之处,如64 Bytes中存在2 bits误码便无法进行纠错,故不能应用于MLC和TLC类型的FLASH纠错中。因此该方案还有一些需要改进和完善之处,以便适用于更多应用场景。

参考文献

- [1] 赵越,余红英,王一奇.一种高速数据存储方法的设计与验证[J].数据采集与处理,2021,36(2):384-390.
- [2] 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].电子测量技术,2017,40(3):127-132.
- [3] 贾刘彬.基于Flash阵列的高速存储及可靠性设计[D].太原:中北大学,2017.
- [4] 刘航航,李晓颖,杨云刚,等. NAND Flash存储关键技

术[J]. 弹箭与制导学报,2016,36(1):161-164.

- [5] LEE Y, YOO H, YOO I et al. 6.4 GB/s multi-threaded BCH encoder and decoder for multi-channel SSD controllers[C]. Solid-State Circuits Conference Digest of Technical Papers, 2012:426-428.
- [6] KIM J, JUNG J, PARK S. An improved LDPC ECC based on system level reprogramming for MLC NAND flash[J]. Journal of Semiconductor Technology and Science, 2020, 20(1):63-75.
- [7] 肖佳.基于NAND FLASH的大数据高速存储系统的设计与实现[D].西安:西安电子科技大学,2014.
- [8] 吴梦雨.基于NAND Flash的错误特性分析[D].杭州:杭州电子科技大学,2019.
- [9] 彭福来,于治楼,陈乃阔,等.面向NAND Flash存储的纠错编码技术概述[J].计算机与现代化,2017(11):35-40.
- [10] 周文.NAND闪存写入(编程)干扰研究[D].苏州:苏州大学,2013.
- [11] 刘传奇.NAND Flash纠错策略优化研究[D].武汉:华中科技大学,2018.
- [12] 文丰,王雨婷,薛志超,等.某记录器存储模块关键技术的研究[J].电子测量技术,2020,43(22):137-141.
- [13] 程龙.基于NAND FLASH存储器的BCH编解码技术研究[D].太原:中北大学,2015.
- [14] 陈佳楠,马永涛,李松,等.嵌入式存储器动态故障诊断数据压缩设计[J].电子测量与仪器学报,2020,34(7):203-209.
- [15] 郭杰.嵌入式高速存储器中数据质量控制方法分析[J].电子测量技术,2019,42(18):118-122.
- [16] 王轩,常亮,李杰.基于FPGA的NAND Flash ECC校验系统设计与实现[J].电子设计工程,2018,26(18):190-193,199.
- [17] DING C, FAN C, ZHOU Z. The dimension and minimum distance of two classes of primitive BCH codes[J]. Finite Fields and Their Applications, 2017(45):237-263.

作者简介

孙晓磊,硕士研究生,主要研究方向为高速数据采集与存储。

E-mail:1484696352@qq.com

王红亮(通信作者),教授,博士生导师,主要研究方向为测试系统集成、微纳器件与系统、超声换能器及系统等。

E-mail:wanghongliang@nuc.edu.cn