基于 NAND FLASH 大容量立体封装芯片在嵌入式系统中的应用

叶振荣 1，王烈洋 1

珠海欧比特控制工程股份有限公司（519080）

摘要: 随着嵌入式系统越来越广泛的应用，嵌入式系统中的数据存储和数据管理已经成为一个重要的课题摆在设计人员面前。Flash 存储器目前已经逐步取代其它半导体存储元件，成为嵌入式系统中主要数据和程序载体,如何在有限的空间内实现大容量存储是目前各大厂商亟待解决的问题。应用的困难在于 NAND Flash 的管理和需要特殊的系统接口。本文介绍了一种利用 MCU 存储器管理接口结合 I/O 口来实现 NAND Flash 存储结构的搭建和管理的方法，并介绍底层的驱动程序。

An Application Based On NAND FLASH Bulk Stereo Package Chip in Embedded System

Abstract: As the embedded systems become more and more widely used, data storage and management has become an important issue. Flash memory is now gradually replacing other semiconductor storage element, become the primary carrier of data and procedures on the embedded system, it is a problem with the manufactures that how in the limited space within the mass storage. The NAND Flash management and the special interface is the main difficult. This paper describes a method that reach to build and management structure of NAND Flash storage which implemented with MCU memory management interface I/O port, and describes the underlying driver.

关键字: 嵌入式系统; 立体封装; 存储器; 系统接口

KEYWORDS: Embedded systems; Stereo package; Memory; System interface

0.引言

随着嵌入式系统越来越广泛的应用，嵌入式系统中的数据存储和数据管理已经成为一个不容忽视的课题。NOR Flash 和 NAND Flash 是现在市场上两种主要的非易失闪存技术，使用复杂的 I/O 口来存取数据，在 NAND Flash 器件上自始至终都必须进行虚拟映射。东芝公司在 1989 年发表的 NAND Flash 结构，像磁盘一样可以通过接口轻松升级。但是应用的困难在于 NAND Flash 的管理和需要特殊的系统接口，本文将就这一系列问题进行探讨。

1. 元件介绍

基于 sparc 架构的 MCU

S698-MIL 是珠海欧比特控制工程股份有限公司为了满足嵌入式应用而开发的 32 位 RISC 高性能嵌入式微处理器，它遵循 SPARC V8 构架。

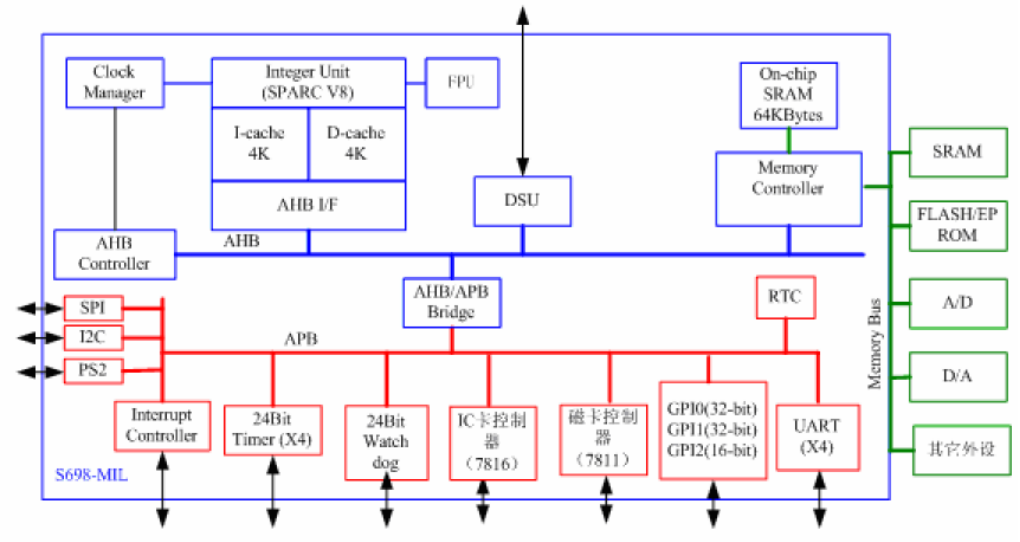


图 1 S698-mil 内部结构

S698-MIL 内部配置了 32 位整数处理单元(IU)，32/64 位浮点处理单元(FPU)。片内采用 32 位 AMBA 2.0 标准总线作为系统架构总线，外部总线支持 8 位、16 位、32 位。AMBA 总线配置了 80 个 GPIO 口(GPIO0 是 32 位，GPIO1 是 32 位，GPIO2 是 16 位)、3 路通用 UART 接口、1 路 16550 兼容 UART 接口，4 个 24 位定时器(TIMERS)、1 个实时时钟(RTC)、1 个看门狗、1 个 PS/2 接口、1 个 I2C 总线接口、1 个 SPI 总线接口、1 个三磁道磁卡接口，3 个智能卡接口等大量外设；丰富的片上外设资源使得 S698-MIL 的集成度和功能得到了大幅度的提高。另外，S698-MIL 还内嵌了 64KBytes 的 SRAM。

S698-MIL 支持片上调试功能。通过调试支持单元(DSU)，用户可以访问 CPU 内部所有寄存器和存储器资源，也可访问外部所有存储器和 I/O 外设，为基于 S698-MIL 的硬件/软件调试提供了方便。

S698-MIL 的应用软件开发环境具有很强的灵活性，除了可以使用欧比特公司提供的专用多任务嵌入式实时操作系统 ORION 外，开发者还可以选择如 RTEMS、VxWorks 等现今流行的嵌入式操作系统进行开发。

S698-MIL 可应用于包括税控收款机、银行 POS 机，电力系统等高端工业控制领域和消费电子领域以及高性能高可靠的航空、航天及武器领域。

1.2 NAND Flash 结构的存储芯片

目前的 Flash memory 主要包括以下两大类:针对程序和数据存储的 NOR flash；针对大容量存储的 NAND flash 。其中,NOR 的特点为芯片内执行(XIP ,eXecute In Place) ,这样应用程序可以直接在 flash 闪存内运行,不必再把代码读到系统 RAM 中,随机读取快、功耗低、稳定性高;而 NAND 的特点为容量大、写速度快、芯片面积小。

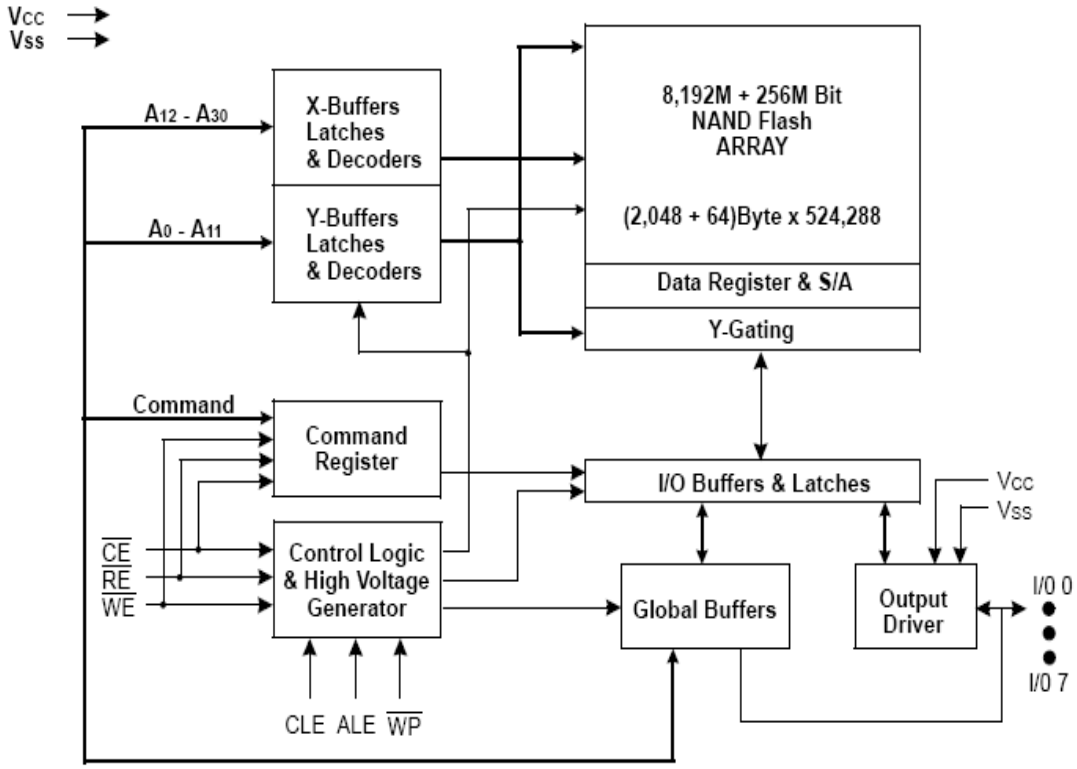


图 2 NAND Flash 的内部结构

VDNF64G08 是一个快速、高存储密度的随机访问存储器。它由 8 个 8G 位的 NAD FLASH 芯片堆叠而成,结构见图 2。整个模块采用堆叠技术，它们之间的互相连接线非常短，寄生电容小。这种芯片非常适用于高速、高性能、高容量的嵌入式系统中。

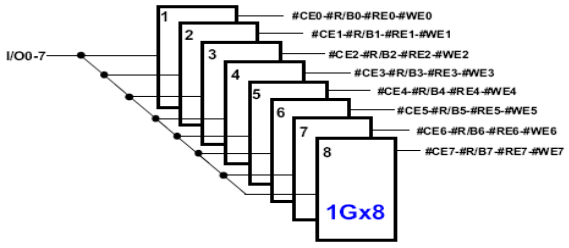


图 3 VDNF64G08 内部连接方式

VDNF64G08 对一个 2048(+64)字节的页进行典型的编程操作只要 200us 的时间，对一个 128k(+4K)大小的块进行擦除需要 1.5ms 的时间，对页中一个字节的读周期为 25ns。它的 I/O 管脚既作地址和数据的输入输出口，也作为命令的输入口。VDNF64G08 的片上写控制器能自动完成所有的编程、擦除功能包括产生所需的脉冲重复和内部数据校验。VDNF64G08 能擦除和编程百万次以上，并通过 ECC 或实时制定算法保证擦除和编程的可靠性。

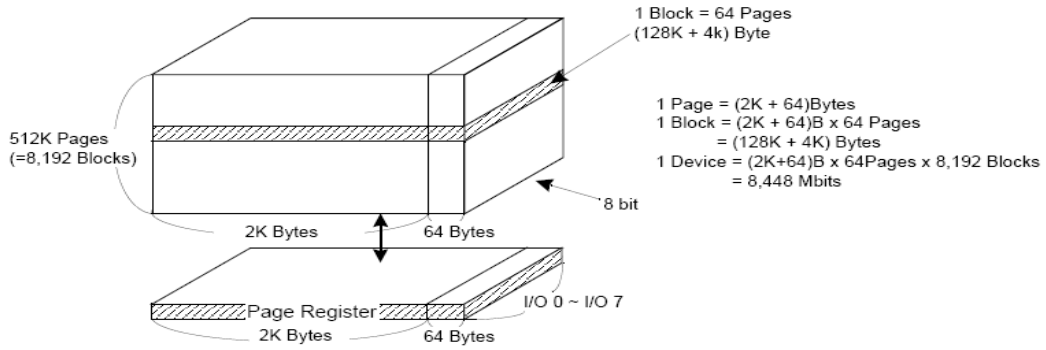


图 4 VDNF64G08 内部容量结构

VDNF64G08 地址是通过复用 8 个 I/O 送入芯片的。这样的设计显著减少了芯片的管脚数目，并为系统的升级带来了方便。在 CE 为低的时，把 WE 置低可以把 VDNF64G08 的命令、地址和数据通过 I /O 口写进去。数据在 WE 的上升沿写入芯片。命令锁存使能 CLE 和地址锁存使能 ALE 用来区分 I/O 口的数据是命令还是地址。所有的命令要用一个总线周期除了块擦除命令要用两个总想周期：一个周期用来做擦除建立，另一个周期在地址调入后做擦除执行。VDNF64G08 单片选有 1G(+32M)字节，需要 31 位的地址，地址分 5 个周期依次送入。页的读操作和页的编程操作都需要同样的 5 个地址周期紧跟在相应的命令输入之后。然而，在块擦除的操作中，只要 3 个地址期。不同的操作通过往命令寄存器写不同的命令来区分的。

信号	管脚序号			信号
#RB7	1		50	#WE6
#RB6	2		49	#WE7
#RB5	3		48	#CE6
#RB4	4		47	#CE7
#RB3	5		46	I/O0
#RB2	6		45	I/O1
#RB1	7		44	I/O2
#RB0	8		43	I/O3
#RE0	9		42	#CE5
#CE0	10		41	VSS
#CE1	11		40	VSS
#CE2	12		39	VSS
VCC	13		38	VCC
VSS	14		37	VCC
#CE3	15		36	#RE1
#CE4	16		35	#RE2
CLE	17		34	I/O4
ALE	18		33	I/O5
#WE0	19		32	I/O6
#WP	20		31	I/O7
#WE1	21		30	#RE3
#WE2	22		29	#RE4
#WE3	23		28	#RE5
#WE4	24		27	#RE6
#WE5	25		26	#RE7

表 1 VDNF64G08 管脚信号分布

2 硬件设计

VDNF64G08 芯片的管脚信息,如下表所示

管脚名称	功能
IO[7:0]	用于命令输入,地址输入,数据输入输出
CE[7:0]	芯片使能信号,低电平有效
RB[7:0]	芯片忙信号,低电平时芯片处于忙状态,与片选信号一一对应,集电极开路输出,使用时需上拉
WE[7:0]	写使能,低有效,与片选信号一一对应
RE[7:0]	读使能,低有效,与片选信号一一对应
CLE	命令锁存,高电平时,IO 数据作为命令输入到芯片内部
ALE	地址锁存,高电平时,IO 数据作为地址输入到芯片内部
WP	写保护,低有效
VCC	3.3v 电源输入
VSS	GND 连接
NC	无电气连接

表 2 VDNf64G08 管脚功能说明

由于 S698 没有专用的 NAND Flash 的专用接口,所有 NANDFlash 与 S698-mi1 的通信使用 GPIO 实现,NAND Flash 的 IO 口与 S698-mi1 的 GPIO0 低 8 位 GPIO0[7:0]连接,用于数据,地址,命令的传输.NAND Flash 的片选信号使用 S698-mi1 的低 3 位地址线 ADD[2:0],通过 3-8 译码实现,其中 3-8 译码器使用 S698-mi1 的 IOCS0 控制工作,防止对 VDNF64G08 的误操作.VDNF64G08 的读写信号直接与 S698-mi1 的读写信号相连,如果线路较长可以考虑串联一个 49.9R 的电阻.命令锁存 CLE,地址锁存 ALE 分别使用 S698-mi1 的地址线 ADD[3],ADD[4].状态信号 RB 上拉 10k 电阻到电源,不再与 S698-mi1 相连,VDNF64G08 的状态信息通过 IO 口读寄存器实现,以节约 S698-mi1 的端口资源.写保护端口上拉 10k 到电源,否则不可擦除及编程.相关连接见图 1.

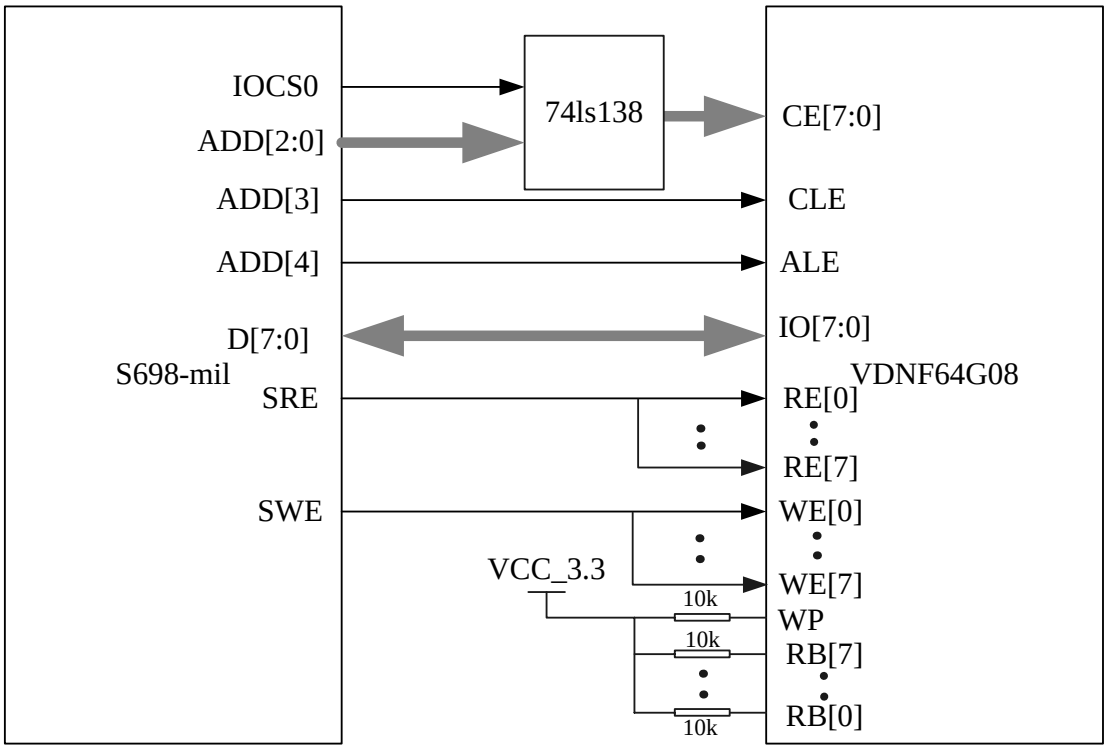


图 5 硬件连接简图

3 软件设计

根据前面的介绍,通过图 5 的硬件连接方式,可以实现对芯片的读写,擦除等控制操作,已可以满足各种场合的应用,S698-mi1 处理器是通过 GPIO 来操作控制 VDNF64G08 的,所以需要对底层操作需要比较了解,下面简单介绍一些常用操作的实现.

Function	1st Cycle	2nd Cycle	Acceptable Command during Busy
Read	00h	30h	
Read for Copy Back	00h	35h	
Read ID	90h	-	
Reset	FFh	-	O
Page Program	80h	10h	
Two-Plane Page Program(4)	80h---11h	81h---10h	
Copy-Back Program	85h	10h	
Two-Plane Copy-Back Program(4)	85h---11h	81h---10h	
Block Erase	60h	D0h	
Two-Plane Block Erase	60h---60h	D0h	
Random Data Input(1)	85h	-	
Random Data Output(1)	05h	E0h	
Read Status	70h		O
Read EDC Status(2)	7Bh		O
Chip1 Status(3)	F1h		O
Chip2 Status(3)	F2h		O

图 1 VDNF64G08 的指令表

S698-mil 芯片 GPIO0 对应的寄存器地址为 0x2000 0000　　0x27ff ffff,因为 VDNF64G08 芯片位宽是 8bit 的,所以需要把 MCU 的 GPIO 位宽也定义成 8bit,根据前面的硬件连接可以知道地址低 3 位对应到 VDNF64G08 芯片的片选信号,add[4]对应到 CLE,add[5]对应到 ALE,为了方便编程,这里做以下宏定义:

```
#define NF_ADDREG(CE)        (*(volatile unsigned char *) (0x20000010 + CE))
#define NF_CMDREG(CE)        (*(volatile unsigned char *) (0x20000008 + CE))
#define NF_DATAREG(CE)       (*(volatile unsigned char *) (0x20000000 + CE))
```

CLE	ALE	$\overline{CE}$	$\overline{WE}$	$\overline{RE}$	$\overline{WP}$	Mode	
H	L	L	⌐_⌐	H	X	Read Mode	Command Input
L	H	L	⌐_⌐	H	X		Address Input(5clock)
H	L	L	⌐_⌐	H	H	Write Mode	Command Input
L	H	L	⌐_⌐	H	H		Address Input(5clock)
L	L	L	⌐_⌐	H	H	Data Input	
L	L	L	H	⌐_⌐	X	Data Output	
X	X	X	X	H	X	During Read(Busy)	
X	X	X	X	X	H	During Program(Busy)	
X	X	X	X	X	H	During Erase(Busy)	
X	X(1)	X	X	X	L	Write Protect	
X	X	H	X	X	0V/Vcc(2)	Stand-by	

NOTE : 1. X can be V_{IL} or V_{IH}.
2. $\overline{WP}$ should be biased to CMOS high or CMOS low for standby.

图 6 操作模式选择表

3.1 读芯片 ID

根据芯片指令表,可以知道读 ID 芯片只需要写入命令 0x90,然后在写入地址 0x0 即可,详细操作可以根据以下时序图进行:

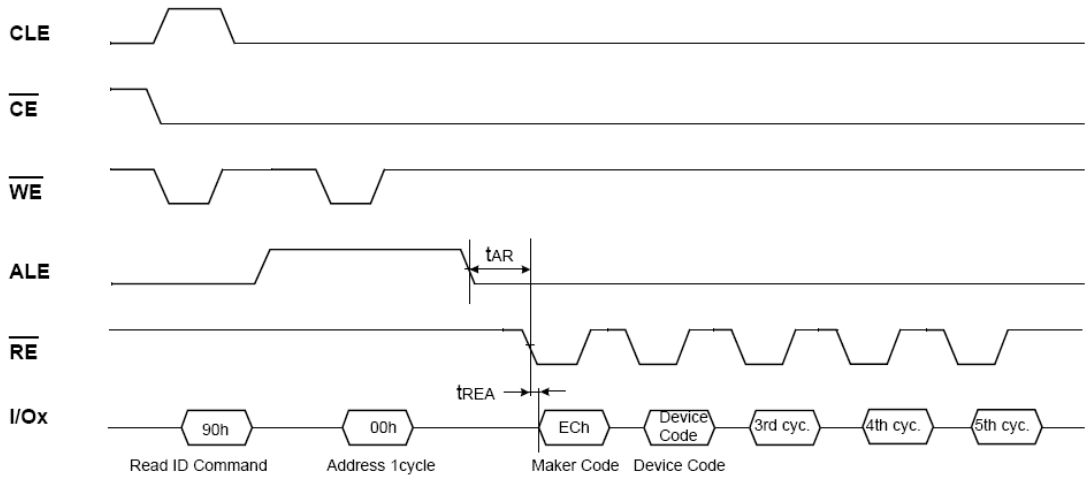


图 7 读 ID 时序

```
int readID(unsigned char CE,unsigned IDlength)
{
    unsigned char ID[],i;
    NF_CMDREG(CE) = 0x90;
    NF_ADDREG(CE) = 0x0;
    delayed(1);
    for(i = 0;i < IDlength;i++)
    {
        ID[i] = NF_DATAREG(CE);
    }
    i = i - 1;
    printf("CHIP CE d% ID:0x%x",ID[0]);
    while(i){
        printf("-%x",ID[IDlength - i]);
        i --;
    }
    printf("\n\r");
}
```

3.2 芯片坏块查询

由于 NAND Flash 的工艺不能保证 NAND 的 Memory Array 在其生命周期中保持性能的可靠，因此，在 NAND 的生产中及使用过程中会产生坏块。为了检测数据的可靠性，在应用 NAND Flash 的系统中一般都会采用一定的坏区管理策略，而管理坏区的前提是能比较可靠的进行坏区检测。检测流程如下图

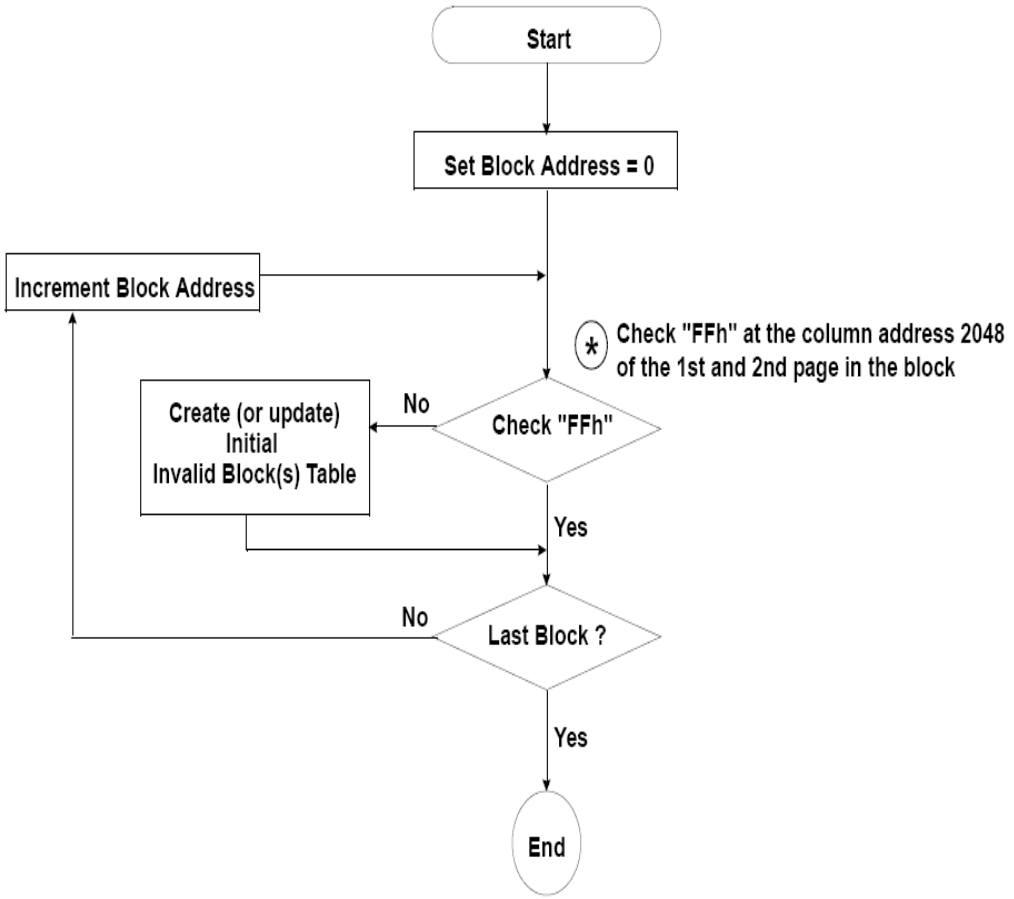
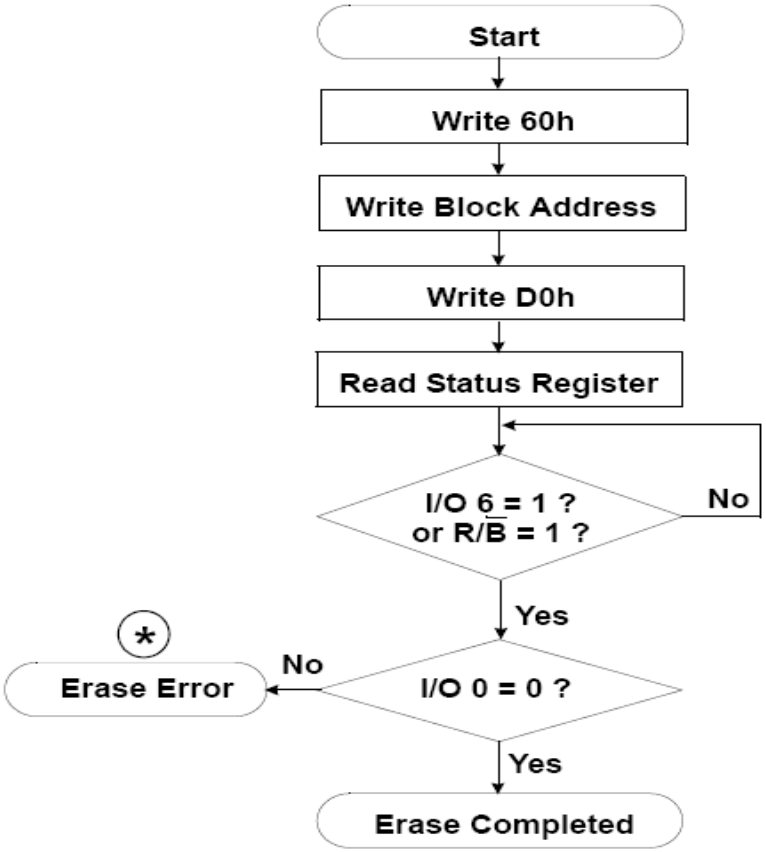


图 8 检测流程

3.3 芯片擦除操作

NAND Flash 的擦除操作是以块为基础进行的.只有已擦除的块才能编程,因为 NAND Flash 芯片的工艺特性决定了,芯片的 CELL 只能由 1 写成 0,而不能由 0 写成 1.芯片的擦除操作有 2 个命令周期和 3 个地址周期构成,其中列地址不需要输入,并且行地址的页地址也不会影响块擦除效果,即块擦除地址只有块地址有效,操作时序及流程图如下.



(*) : If erase operation results in an error, map out the failing block and replace it with another block.

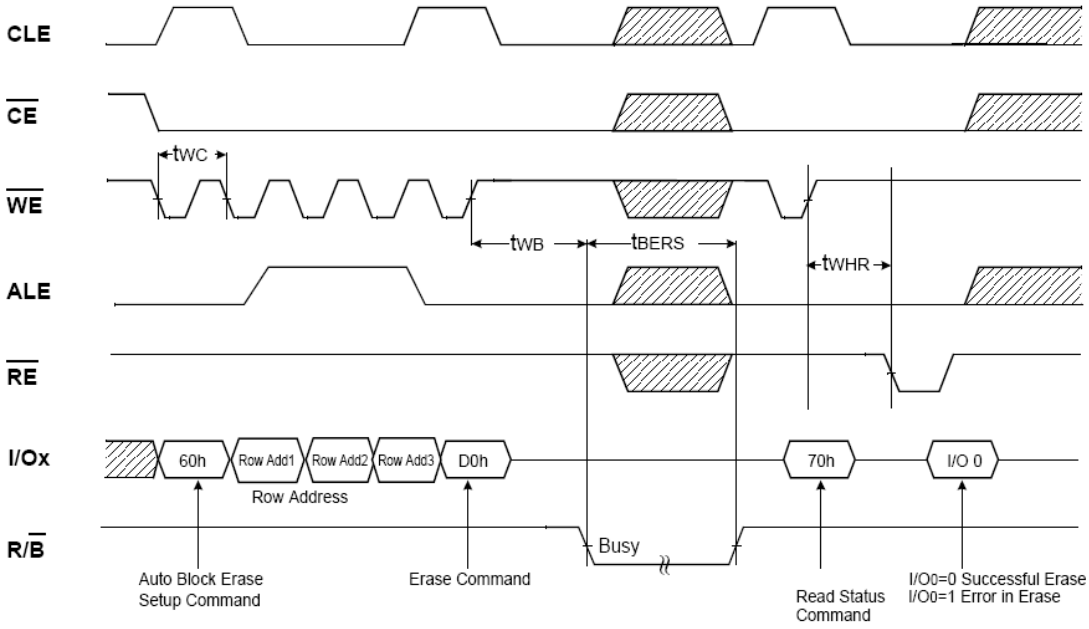


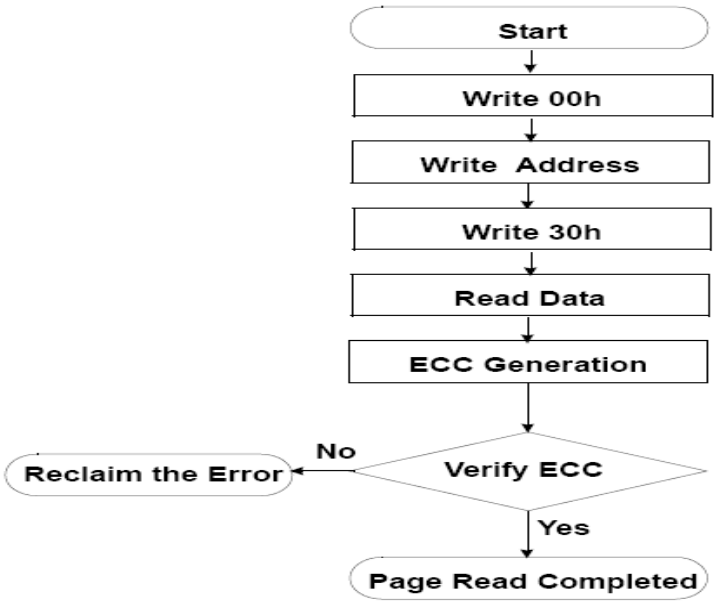
图 9 块擦除时序

3.4 芯片页读操作

页读操作通过将 00h 指令写入指令寄存器，接着写入 5 个地址（2 个列地址，3 个行地址），最后写入 30h 指令来启动。一旦页读指令被器件锁存，下面的页读操作就不需要再重复写入指令了。

写入指令和地址后，处理器可以通过对信号线 R/ 的分析来判断该操作是否完成。如果信号为低电平，表示器件正 “忙”；为高电平，说明器件内部操作完成，要读取的数据被送入了数据寄存器。外部控制器可以在以 50ns 为周期的连续 脉冲信号的控制下，从 I/O 口依次读出数据。

连续页读操作中，输出的数据是从指定的列地址开始，直到该页的最后一个列地址的数据为止。操作时序及流程图如下.



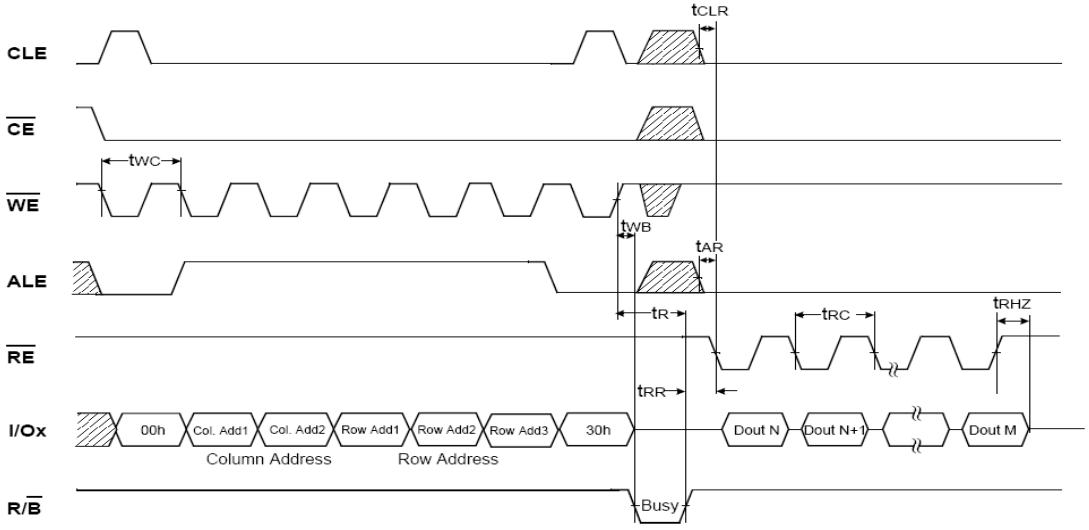


图 10 页读时序

3.5 芯片页编程操作

VDNF64G08 芯片的写入操作也以页为单位。写入之前必须先擦除，否则写入将出错。

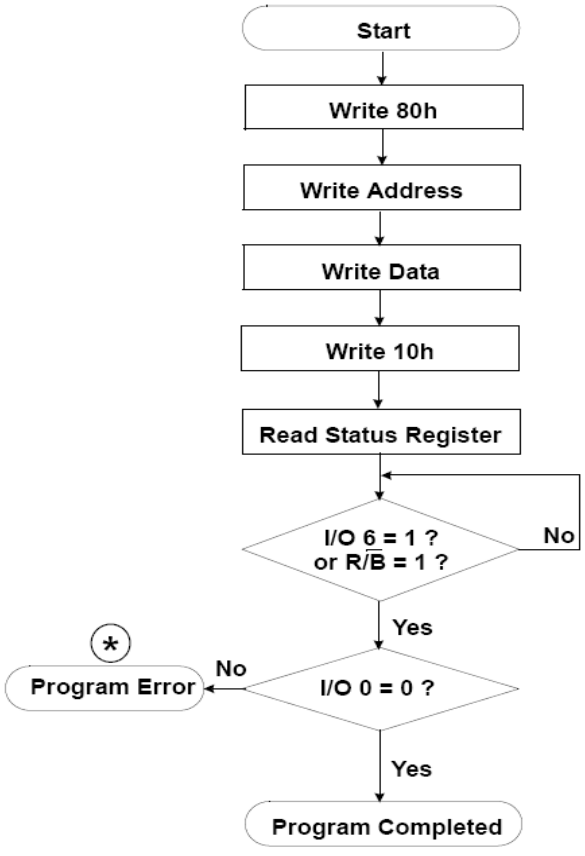
页写入周期总共包括 3 个步骤：写入串行数据输入指令（80h），然后写入 5 个字节的地址信息，最后串行写入数据。

串行写入的数据最多为 4096 字节，它们首先被写入器件内的页寄存器，接着器件进入一个内部写入过程，将数据从页寄存器写入存储宏单元。

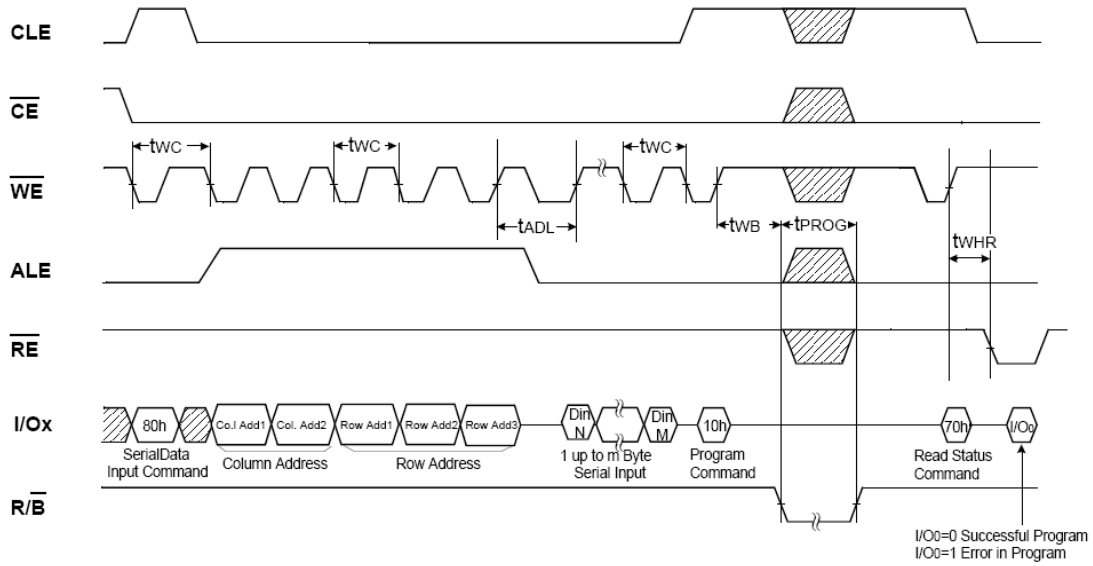
串行数据写入完成后，需要写入“页写入确认”指令 10h，这条指令将初始化器件的内部写入操作。如果单独写入 10h 而没有前面的步骤，则 10h 不起作用。10h 写入之后，内部写控制器将自动执行内部写入和校验中必要的算法和时序，这是系统控制器就可以去做别的事了。

内部写入操作开始后，器件自动进入“读状态寄存器”模式，在这一模式下，当 RE/CE 为低电平时，系统就可以读取状态寄存器。系统可以通过检测 R/B 的输出，或读状态寄存器的状态位（I/O7）来判断内部写入是否结束。在器件进行内部写入操作时，只有读状态寄存器指令和复位指令会被响应。当页写入操作完成，应该检测写状态位（I/O1）的电平。

内部写校验只对 1 没有成功地写为 0 的情况进行检测。指令寄存器始终保持着读状态寄存器模式，直到其它有效的指令写入指令寄存器为止。操作时序及流程图如下.



(*) : If program operation results in an error, map out the block including the page in error and copy the target data to another block.



NOTES : tADL is the time from the $\overline{WE}$ rising edge of final address cycle to the $\overline{WE}$ rising edge of first data cycle.

图 11 页编程时序

4 结束语

VDNF64G08 是一个快速、高存储密度的随机访问存储器。整个模块采用堆叠技术，它们之间的互相连接线非常短，寄生电容小。在研发初期理论论证和生产之后的实验数据表明，这种芯片非常适用于高速、高性能、高容量的嵌入式系统中，并得到用户的好评。

参考文献

[1]珠海欧比特控制工程股份有限公司. VDNF64G08-F 使用说明书. 2013.

[2]H M Peitel,P J Deitel.C How to program,second Edition.蒋才鹏等译.C 程序设计教程.北京: 机械工业出版社，2000

[3]珠海欧比特控制工程股份有限公司. VDNF64D08-K 使用说明书. 2013.

[4]珠海欧比特控制工程股份有限公司. S698-T 芯片用户手册. 2011

[5]夏宇闻.Verilog 数字系统设计教程[M].北京: 北京航空航天大学出版社，2003

[6]SPARC International Inc. The SPARC Architecture Manual. Version 8 [K]