

## softpapa's Blog

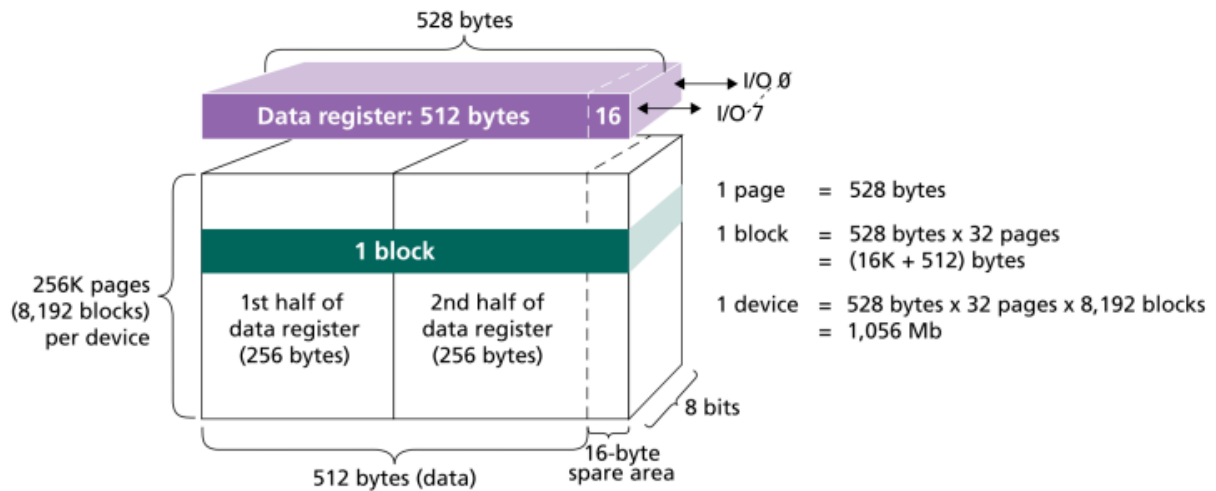
- [About Me](#)
- [Archive](#)
- [feeds](#)

over 4 years ago

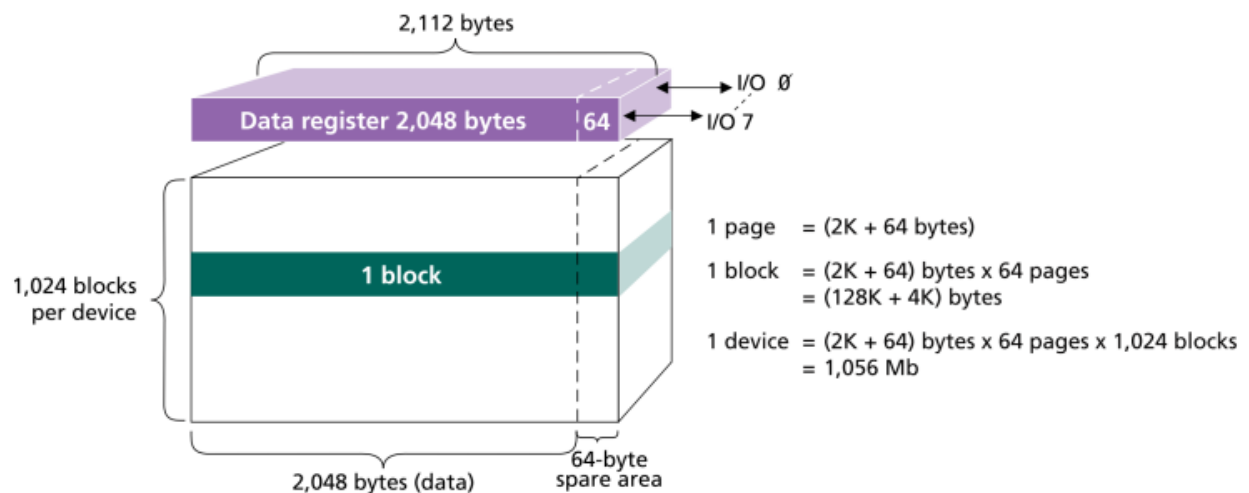
### Nand Flash 基本介紹

本文所有的內容請參考Figure 1和Figure 2，分別為small和large page 兩種型式的基本結構

**Figure 1: 1Gb NAND Flash Small-Block Array Organization**



**Figure 2: 1Gb NAND Flash Large-Block Array Organization**



文章內容基於2011 年的資訊，現在nand flash 單顆愈做愈大，也應該有新的技術出來，以後有機會再加吧

#### Page

為nand flash chip 讀取的最小單位，依顆粒製程不同有512, 2048, 4096, 8192 (bytes)，目前最常見的為2048 (bytes)

#### Spare

以2k page 為例，每一個page在實體的結構中除了2k bytes的使用空間，還會有64 bytes額外的容量，供hw/sw儲存一些額外的資訊，通常會用來儲存ecc的data，來修正nand flash使用久後會有bit不能使用的問題 [Dashboard](#)

## OOB (out of boundary)

即指spare，sw比較常用這個名詞

## Small/Large

page 容量512 bytes 的稱 small page，大於等2k bytes 都稱為large page

## Block

為nand flash chip 寫入的最小單位，每64 page 即為一個block，因為page size 會有不同，block size 也就不同，一顆chip裡有多少block，是依nand flash的大小而定，但通常是2的冪次方。

## Read/Write operation

nand flash 讀寫方式異於一般的磁性儲性裝置。因為結構上的問題，如果要針對特定的page 寫入，只能把資料寫由1 -> 0，如果要0->1，那只能針對整個block (64 pages)做動作，此動作稱為”erase“，而寫入page的動作稱為”program“，一般nand chip的資料都會標明erase/program的時間，以[Micro MT29F2G08AABWP](#)為例，erase/program的時間分別為(2ms/300us)

- write  
標準寫入動作可分為下列幾步
  1. 將block 讀進ram
  2. 改變ram裡的資料
  3. erase block
  4. 依ram 裡block 的資料，program 64 個page
- read 則以page 為單位，無特殊的限制。

## Life cycle

nand flash 相較傳統磁性的儲存裝置有較低的資料可靠度和使用限制，主要為二個方面

- endurance  
erase block/program page 這個二動作是破壞性的，每個block erase的次數是有限制的，通常為1k~100k次
- retention  
這是指當資料儲放在nand中，可以保證有效多久，一般是10年。所以nand flash device 不適合當長時間的資料保存裝置

## Bad block

指在chip中資料讀取出來有可能會發生錯誤的block，依形成的原因可以分為下列二種

- factory-bad  
即出廠時即有問題的block，每個nand chip 都會標明這個型別的顆粒至少有多少block是可以用的，而且保證第一個block一定是好的，且第一個block在某個erase/program次數不會出問題(這是為了nand boot)。nand flash 出廠時會erase 所有的block，如經檢測為bad block，則會j把那個block的第一個page的oob的第一(Micro)或第六個byte(Samung, Toshiba and 其他)標示為0xFF以外的值，每顆nand flash 在使用之前應該要先把這個值備份出來，作為以後建立bad block table時使用
- worn-out  
即不斷使用後所產生的bad block

## ECC

ECC即error correct code，因為無法預測什麼時候會發生bad block，sw上會對每個page做ecc的修正機制以防止資料錯誤，這些修正的資料的會被存在同一個page的oob裡，一般用的演算法有下列三種，每種有不同的修正能力也需要不同的ecc修正資訊，可參考下圖。另外隨著page size愈來愈大，對ECC的能力要求來愈來愈高，依innodisk 資料，硬碟型的產品，單一page ECC能力要求到7x bits

- Hamming code
- reed-solomum
- binary BCH

Type	Hamming				Reed-Solomon				Binary BCH			
ECT <sup>1</sup>	Overhead Per Sector		Spare Area Usage		Overhead Per Sector		Spare Area Usage		Overhead Per Sector		Spare Area Usage	
	Bits	Bytes	64B	112B	Bits	Bytes	64B	112B	Bits	Bytes	64B	112B
1	13	2	13%	7%	18	3	19%	11%	13	2	13%	7%
2	-	-	-	-	36	5	31%	18%	26	4	25%	14%
4	-	-	-	-	72	9	56%	32%	52	7	44%	25%
8	-	-	-	-	144	18	113%	64%	104	13	81%	46%
10	-	-	-	-	180	23	144%	82%	130	17	106%	61%
14	-	-	-	-	252	32	200%	114%	182	23	144%	82%

Notes: 1) ECT = error correction threshold; number of correctable bits per 512-byte sector.

### Wear leveling

使得每個block可以寫入的次數可以大致相同延長nand使用的機制，通常由軟體或韌體負責，有分為Dynamic和static兩種方式

- Dynamic  
在要寫入資料的時候，挑選比較少使用的block來寫入，原本的那個block就回收作為以後用，這方式的缺點為，假設磁碟中有2000個block，其中1500個儲存read only data，那可以用來置換block就只剩500個。
- Static  
收集的統計數據，在沒讀寫的時候偷偷的把資料交換到較少使用的block，此方式可以避免dynamic的缺點，即使是read-only data 所佔據的block 也可以釋放出來使用  
由此可知nand flash based 的儲存裝置如果可用的空間等於實際的空最好不要裝滿資料，留些額外的空間做為替代用的block，可延長使用期限。但是現今實際上的產品，會限縮可使用空間。實際256GB 可用的只有240GB。

### Chip ID Definition Table

每顆nand chip 容量的相關值，可以透過“read id”這個命令讀出來，舊的格式有5bytes(目前常見)，新的格式有8 bytes(目前找不到)，下圖為舊格式bit field的定義

**ID Definition Table****90 ID : Access command = 90H**

	Description
1 <sup>st</sup> Byte	Maker Code
2 <sup>nd</sup> Byte	Device Code
3 <sup>rd</sup> Byte	Internal Chip Number, Cell Type, Number of Simultaneously Programmed Pages, Etc
4 <sup>th</sup> Byte	Page Size, Block Size, Redundant Area Size, Organization, Serial Access Minimum
5 <sup>th</sup> Byte	Plane Number, Plane Size

**3rd ID Data**

	Description	I/O7	I/O6	I/O5	I/O4	I/O3	I/O2	I/O1	I/O0
Internal Chip Number	1							0	0
	2							0	1
	4							1	0
	8							1	1
Cell Type	2 Level Cell					0	0		
	4 Level Cell					0	1		
	8 Level Cell					1	0		
	16 Level Cell					1	1		
Number of Simultaneously Programmed Pages	1			0	0				
	2			0	1				
	4			1	0				
	8			1	1				
Interleave Program Between multiple chips	Not Support		0						
	Support		1						
Cache Program	Not Support	0							
	Support	1							

**4th ID Data**

	Description	I/O7	I/O6	I/O5	I/O4	I/O3	I/O2	I/O1	I/O0
Page Size (w/o redundant area )	1KB							0	0
	2KB							0	1
	4KB							1	0
	8KB							1	1
Block Size (w/o redundant area )	64KB			0	0				
	128KB			0	1				
	256KB			1	0				
	512KB			1	1				
Redundant Area Size ( byte/512byte)	8						0		
	16						1		
Organization	x8		0						
	x16		1						
Serial Access Minimum	50ns/30ns	0				0			
	25ns	1				0			
	Reserved	0				1			
	Reserved	1				1			

**5th ID Data**

	Description	I/O7	I/O6	I/O5	I/O4	I/O3	I/O2	I/O1	I/O0
Plane Number	1					0	0		
	2					0	1		
	4					1	0		
	8					1	1		
Plane Size (w/o redundant Area)	64Mb		0	0	0				
	128Mb		0	0	1				
	256Mb		0	1	0				
	512Mb		0	1	1				
	1Gb		1	0	0				
	2Gb		1	0	1				
	4Gb		1	1	0				
	8Gb		1	1	1				
Reserved		0						0	0

**BBT**

即*Bad Block Table*，用來記錄整個nand flash 顆粒中，有那些block 是壞掉的。每個nand flash 在出廠後，會先被測試機台scan 一次，把壞的block資訊寫在某個特定的地方。做板子的廠商需要先把牠讀出來記起來，不然重新erase就不見了

[← sudo service blog restart Nand Flash BBT Support in Linux →](#)

Like 61

Tweet



- July 23, 2013 22:52
- [Permalink](#)

Copyright © 2013 softpapa . Powered by [Logdown](#).  
Based on work at [subtlepatterns.com](#).