WELCOME

Embedded System Hardware Architectures, *Introduction*

Lecture 4 May13th 2024

UCSC Silicon Valley Extension
PROFESSIONAL EDUCATION

Michael Wang

Student Services

E: extension@ucsc.edu
P: 408.861.3860

Schedule

Date:	Start Time:	End Time:	Meeting Type:	Location:
Mon, 04-01-2024	6:30 p.m.	9:30 p.m.	Flexible	SANTA CLARA / REMOTE
Mon, 04-08-2024	6:30 p.m.	9:30 p.m.	Flexible	SANTA CLARA / REMOTE
Mon, 04-15-2024	6:30 p.m.	9:30 p.m.	Flexible	SANTA CLARA / REMOTE
Mon, 04-22-2024	6:30 p.m.	9:30 p.m.	Flexible	SANTA CLARA / REMOTE
Mon, 04-29-2024	6:30 p.m.	9:30 p.m.	Flexible	SANTA CLARA / REMOTE
Mon, 05-06-2024	6:30 p.m.	9:30 p.m.	Flexible	SANTA CLARA / REMOTE
Mon, 05-13-2024	6:30 p.m.	9:30 p.m.	Flexible	SANTA CLARA / REMOTE
Mon, 05-20-2024	6:30 p.m.	9:30 p.m.	Flexible	SANTA CLARA / REMOTE
Mon, 06-03-2024	6:30 p.m.	9:30 p.m.	Flexible	SANTA CLARA / REMOTE
Mon, 06-10-2024	6:30 p.m.	9:30 p.m.	Flexible	SANTA CLARA / REMOTE

Fri 6:30pm May 10th

Fri 6:30pm May 17th

Wed 6:30pm May 22nd

Fri 6:30pm May 31st

(Memorial Day Weekend on May 27th – no class)

Agenda (1)

1. April 1st 6:30pm – Hybrid
   - Tear Down Analysis, Inter-IC communications
   - Basic Concepts (Serial vs Parallel bus)
   - Quiz 1 (closed Notes)

2. May 6th 6:30pm Hybrid

Microcontroller/ Microprocessor Systems / SoC Concepts

   - Quiz 2 (closed Notes)

Agenda (2)

3. May 10th Friday 6:30pm: Hybrid

- Volatile Memories (System and Architectures)
- Quiz 3 (closed Notes)

4. May13th Monday 6:30pm: Hybrid

- *Non-Volatile Memories (System and Architectures)*
- Quiz 4 (closed Notes)

Agenda (3)

5. May 17th Friday 6:30pm: Hybrid

- *Special Functions Circuitry in Embedded Processors ColdFireV1 v S5D9 Synergy v 8051: FPU, DMA*
- Quiz 5 (closed Notes)

6. May 20th Monday 6:30pm: Hybrid

- *Special Function Circuitry: GPIO, UART*
- *Hardware System Design Considerations: IC Packaging, IC Thermal Considerations*
- Quiz 6 (closed Notes)

Agenda (4)

7. May 22nd Wednesday 6:30pm: Hybrid

- *Embedded systems in Makers Faire DIY movement, single board computers, industry trends and best practices.*
- *Guest Speaker 1: Age of AI and the Transformation of Compute Infrastructure"*
- *Project 1 Presentation Due*
- Quiz 7 (Closed Notes)

8. May 31st Friday 6:30pm: Hybrid

- *Special Function Circuitry: PWM, WDT, PMIC*
- *Embedded System Design Methodology: FPGA, ASIC, Full-Custom Design, COTs*
- *Guest Speaker 2: "FPGA and SOMs"*
- Quiz 8 (Closed Notes)

(Memorial Day Weekend on May 27th – no class)

Agenda (5)

9. Jun 3rd Monday 6:30pm: Hybrid

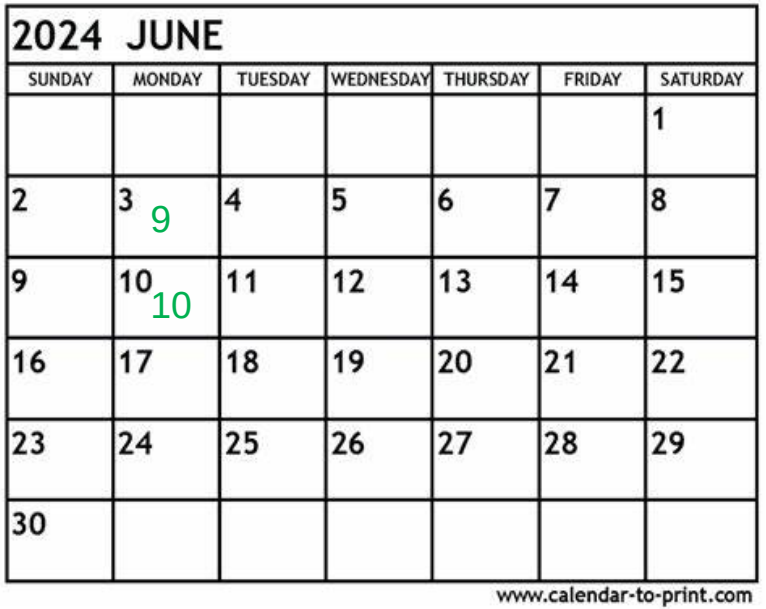
- *Special Function Circuitry: ADC , DAC, RTC*
- *Form Factor, System Benchmarking*
- *Guest Speaker 3: CM, EspressoBin / Dragon Board/ other SBCs “*
- Quiz 9 (Closed Notes)

10 Jun 10th Monday 6:30pm: Hybrid

- *CPU v GPU v TPU v NPU v VPU v XPU v DPU*
- *Software Considerations: OS, RTOS, Baremetal, Middleware*
- *Industry Case Studies*
- ***Project2 + Presentation Due,***
- ***Course Wrap Up***
- **Final Exam (Open Notes)**

ReCap on Schedule of classes

.



Project1 (Wednesday May 22nd)

Choose one of the following:

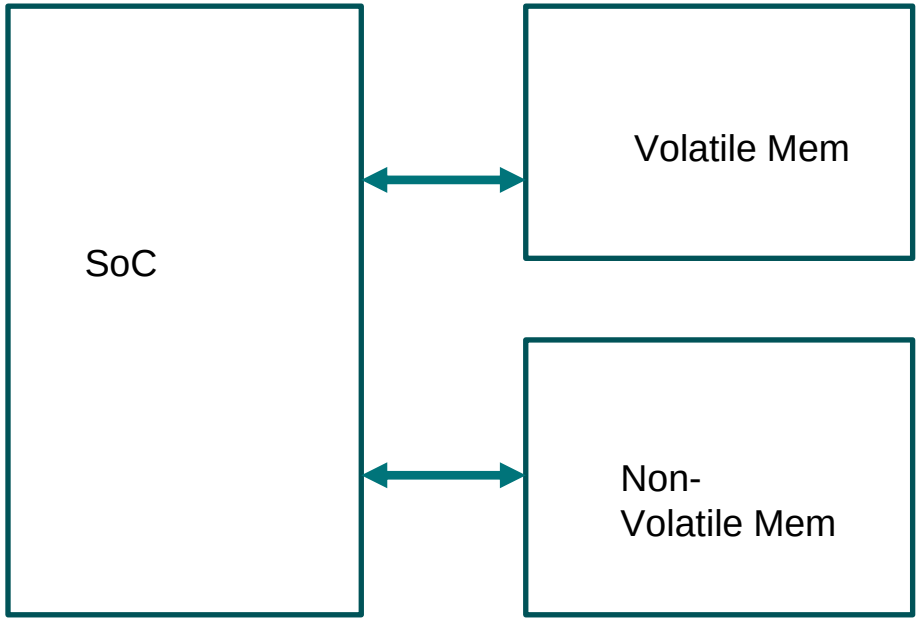
1. 4-page (double spaced) minimum Research on the Topic of your choice. Topic should be related to class materials eg Main Memory development over the last 20 years.
2. Read a book on class related material and write a 4-page book review. Review topics should summarize the contents of the book and include your feedback and recommendations.
3. Designing an embedded system on paper - Choose a hardware project of your choice and draft out the system functional block diagram (1 page). Explain why you choose the components, what is the functional purpose of the component (3 page).

Project2 (due Nov 20th -Last day of class)

Single Board Computer / uC Eval Kit Project:

On week # 7, you will be introduced to several SBC on the market, such as Rpi, Arduino, etc. Earlier on week # 2, you have been briefed on uC Eval kit/ Demo Kit. The purpose of the Project2 is to compare and contrast uC vs SBC requirements. What are the differences in processor performance, mem subsystem requirement, display(?) requirements, OS requirement, pricing, market application etc. Compare and contrast one uC kit vs a SBC.
This can be expanded to compare other topics such as DDR4 vs LPDDR4, etc.

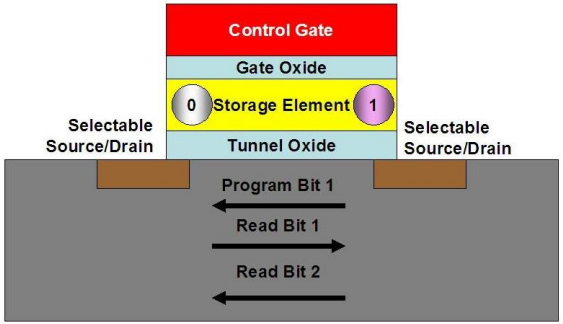
3-chip solution



Non-Volatile Memories

Non-Volatile Memories

- Technology behind NVM- FLASH and EEPROM



- Charge Trapping vs Floating Gate mechanisms
- In a charge trapping flash electrons are stored in a trapping layer just as they are stored in the floating gate in a standard flash memory, EEPROM, or EPROM. The key difference is that the charge trapping layer is an insulator, while the floating gate is a conductor.

NAND / NOR (1)

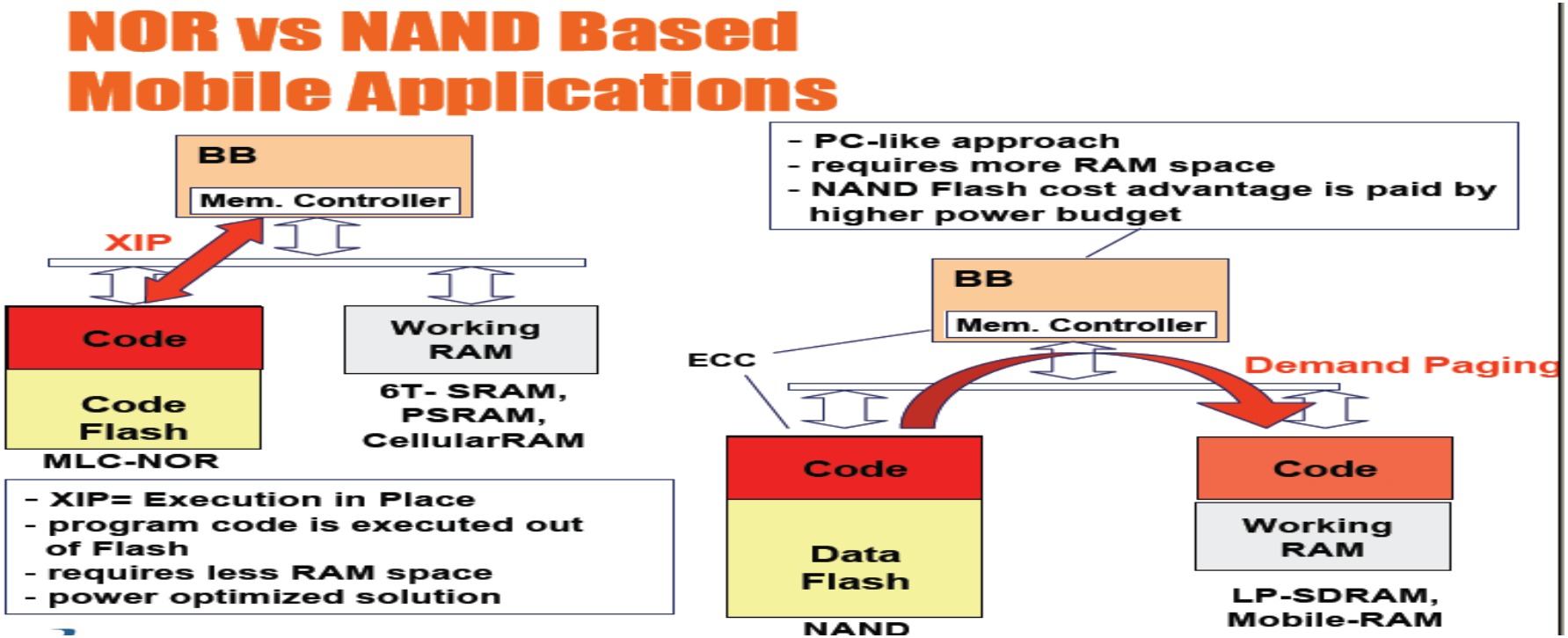
- NAND NOR
- ADVANTAGES: Fast writes/erases Random Access (Fast Read)

- Dis Adv: Slower Random Access Slower write/erase

Applications:
Data storage

Applications:
CODE Storage XIP
(Boot directly from NVM)

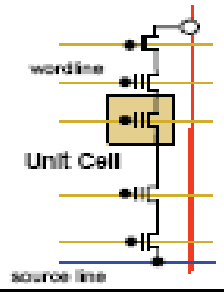
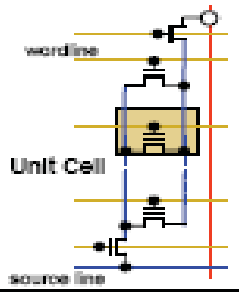
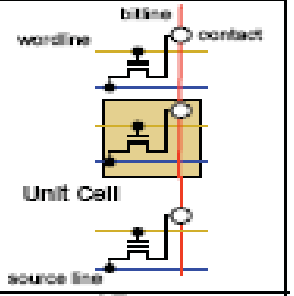
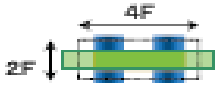
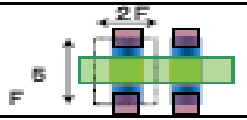
NAND / NOR (2)



Source attributed to–macronix.com

NAND / NOR (3)

Flash Memory Cell Comparison

	NAND	AND	NOR
Cell Array			
Layout			
Cross-section			
Cell size	$4F^2$	$8F^2$	$10F^2$

Source attributed to– WIKIPEDIA.COM

NAND (4)

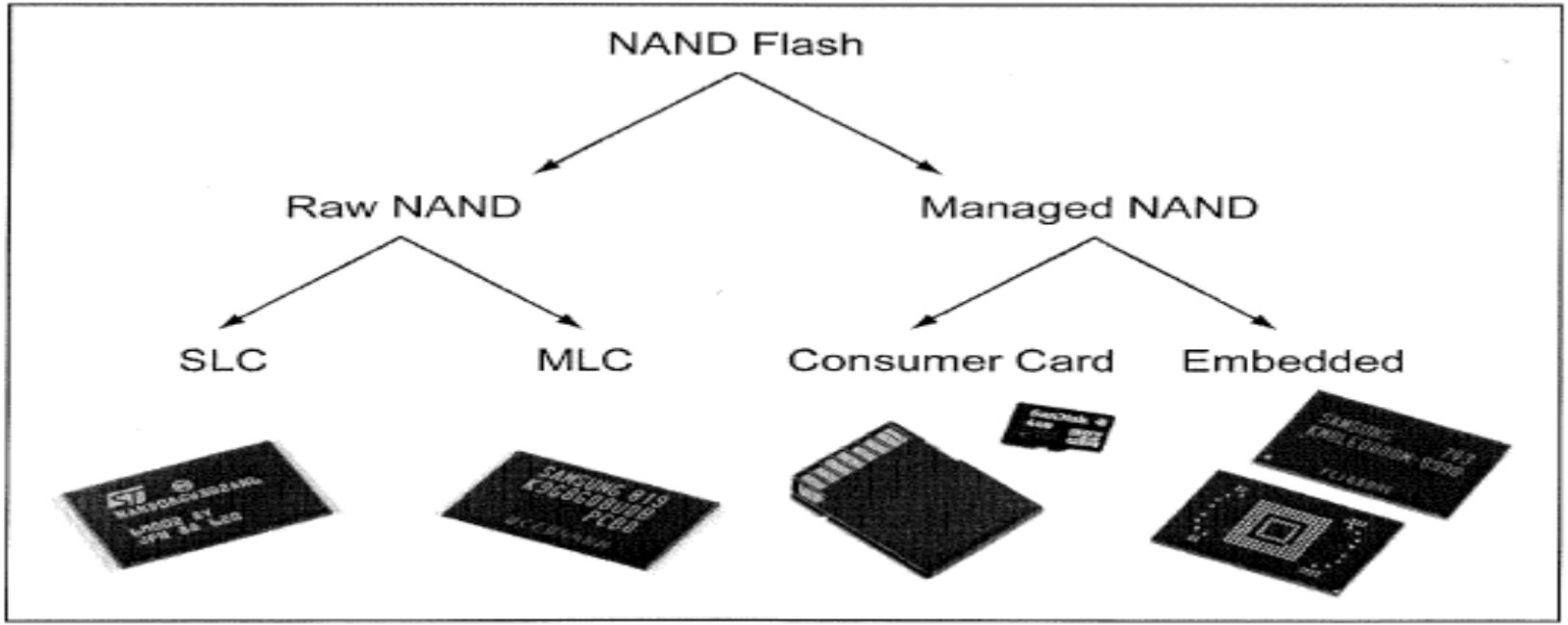
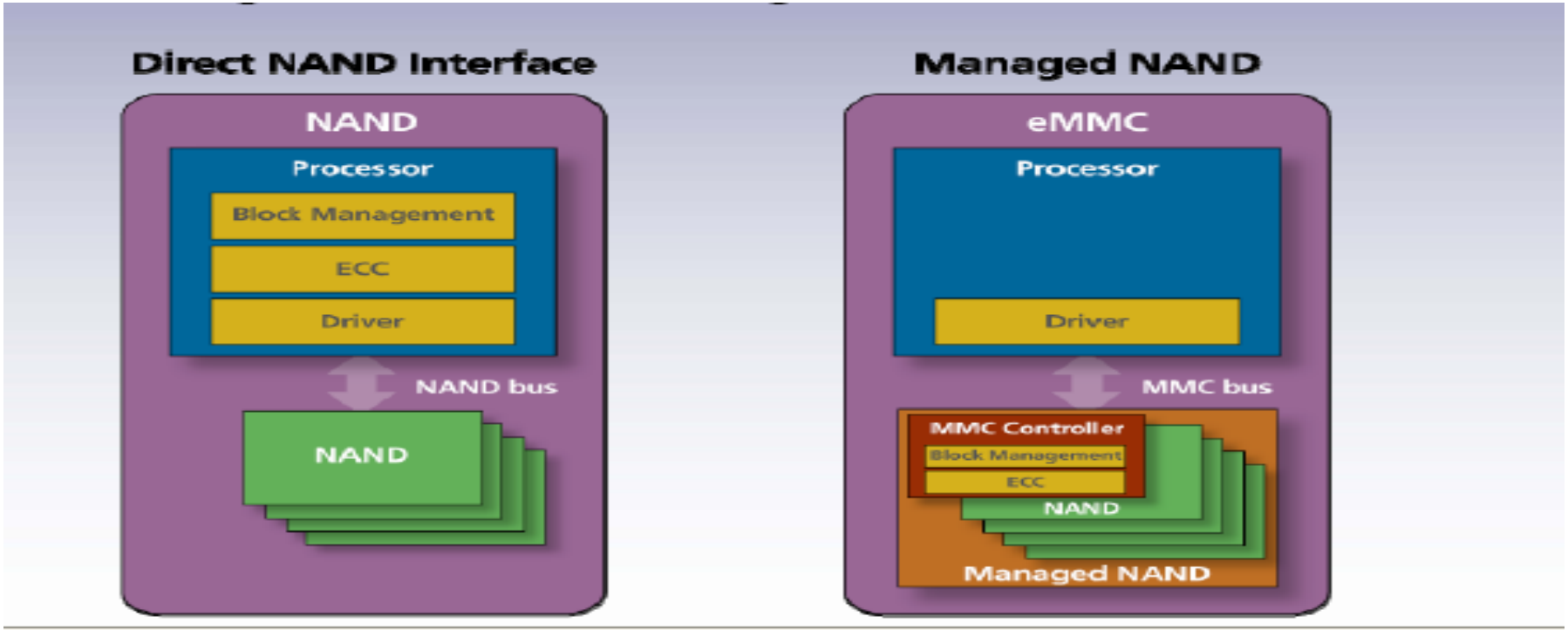


Figure 1 – NAND Flash type categories.

Source attributed to–wikipedia.com

NAND (5)



Source attributed to–denali.com

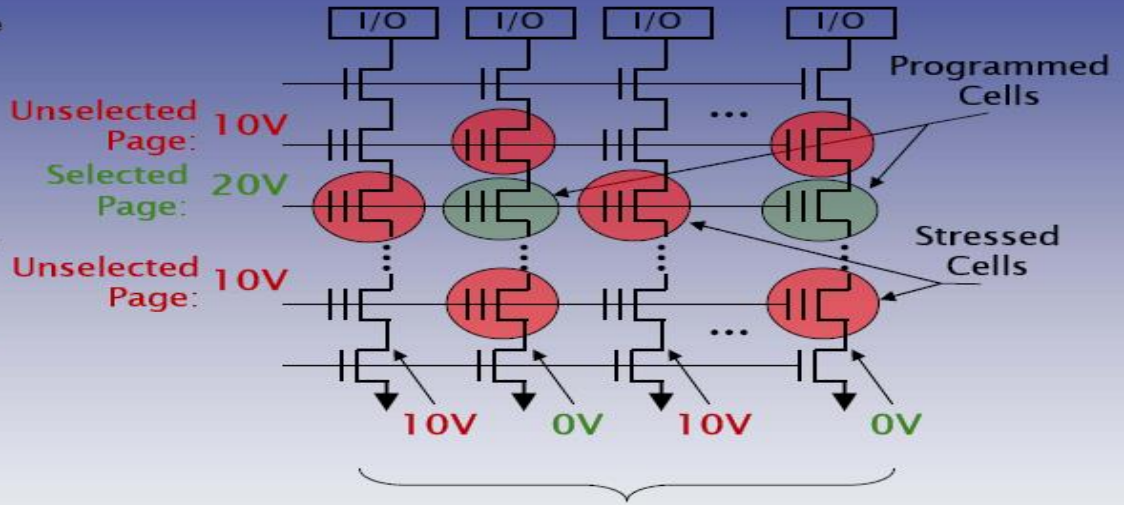
NAND (6)

Features	MLC	SLC
Bits per cell	2	1
Voltage	3.3V	3.3V, 1.8V
Data width (bits)	x8	x8, x16
Architecture		
Number of planes	2	1 or 2
Page size	2,112–4,314 bytes	2,112 bytes
Pages per block	128	64
Reliability		
NOP (partial page programming)	1	4
ECC (per 512 bytes)	4+	1
Endurance (ERASE / PROGRAM cycles)	<10K	<100K
Array Operations		
^{t}R (Max)	50µs	25µs
^{t}PROG (Typ)	600–900µs	200–300µs
^{t}BERS (Typ)	2ms	1.5–2ms

Source attributed to–denali.com

NAND (7)
Program Disturb

- Cells *not* being programmed receive elevated voltage stress
- Stressed cells
  - Are always in the block being programmed
  - Can either be on pages *not* selected, or in a selected page but not supposed to be programmed
- Charge collects on the floating gate causing the cell to appear to be weakly programmed
- Does not damage cells; ERASE returns cells to undisturbed levels
- Disturbed bits are effectively managed with error correction codes (ECC)
- Partial-page programming accelerates disturbance



Note: Circuit structures and voltages are representative only. Details vary by manufacturer and technology node.

Source attributed to–denali.com

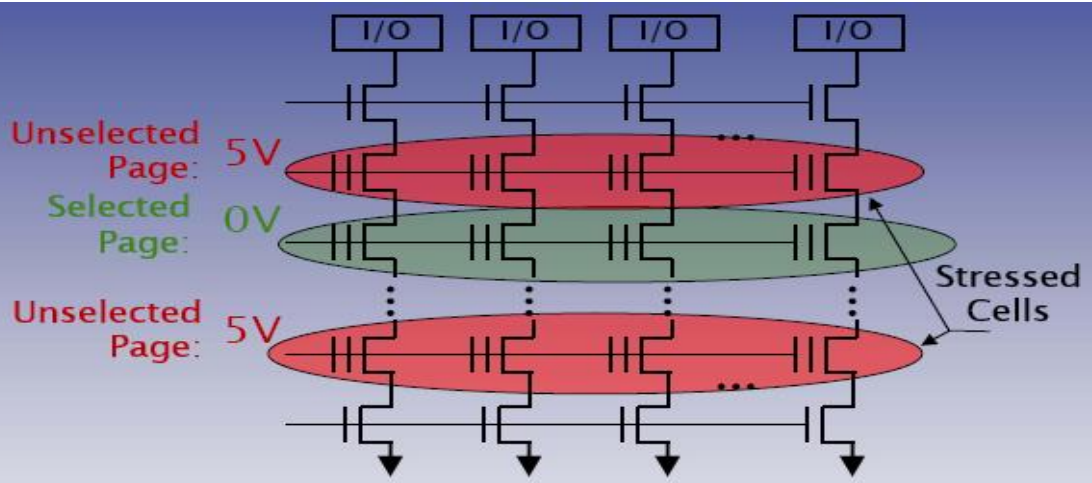
NAND (8)
Reducing Program Disturb

- Program pages in a block sequentially, from page 0 to page 63 (SLC) or 127 (MLC)
- Minimize partial-page programming operations (SLC)
- It is mandatory to restrict page programming to a single operation (MLC)
- Use ECC to recover from program disturb errors

Source attributed to–denali.com

NAND (9)
Read Disturb

- Cells *not* being read receive elevated voltage stress
- Stressed cells are
  - Always in the block being read
  - Always on pages not being read
- Charge collects on the floating gate causing the cell to appear to be weakly programmed
- Does not damage cells; ERASE returns cells to undisturbed levels
- Disturbed bits are effectively managed with ECC



Source attributed to–denali.com

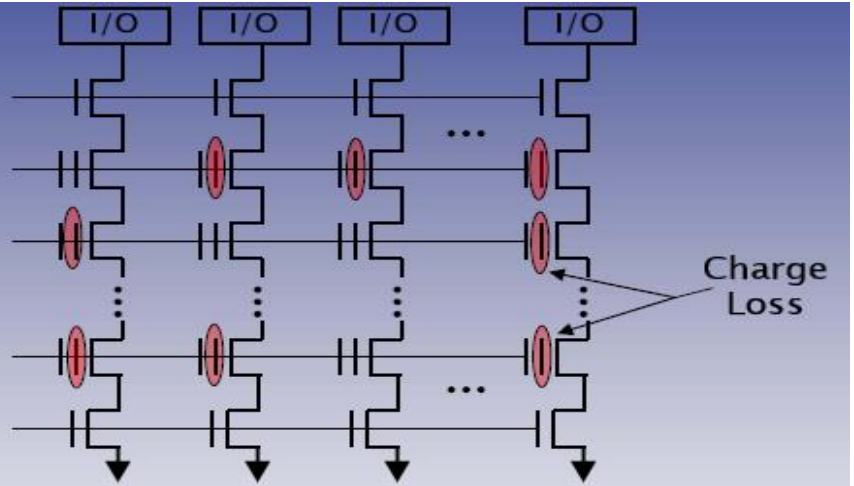
NAND (10)
Reducing Read Disturb

- Rule of thumb for excessive reads per block between ERASE operations
  - SLC – 1,000,000 READ cycles
  - MLC – 100,000 READ cycles
- If possible, read equally from pages within the block
- If exceeding the rule-of-thumb cycle count, then move the block to another location and erase the original block
- Establish ECC threshold to move data
- Erase resets the READ DISTURB cycle count
- Use ECC to recover from read disturb errors

Source attributed to–denali.com

NAND (11)
Data Retention

- Charge loss/gain occurs on the floating gate over time; device threshold voltage trends to a quiescent level
- Cell is undamaged; block can be reliably erased and reprogrammed

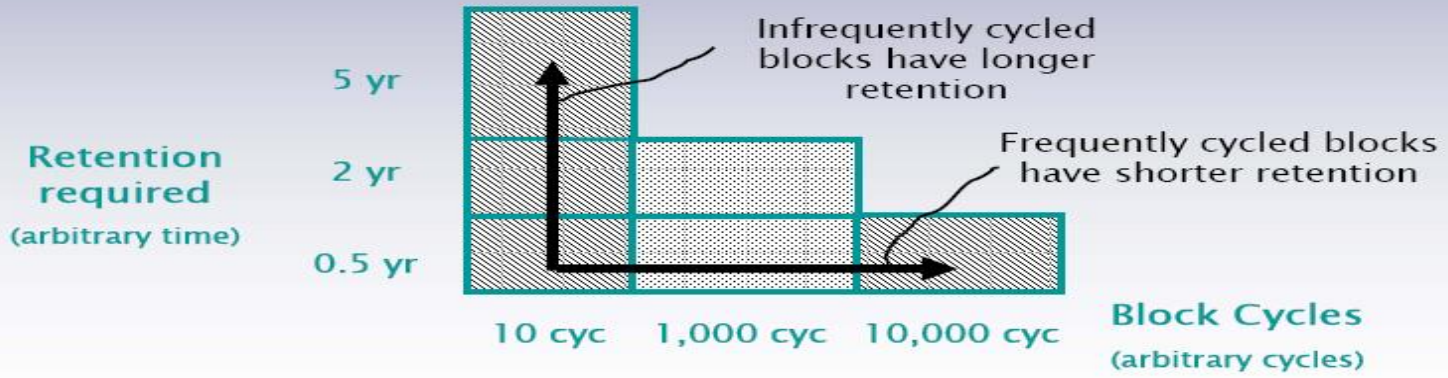


Note: Circuit structures and voltages are representative only. Details vary by manufacturer and technology node.

Source attributed to–denali.com

NAND (12)
Improving Data Retention

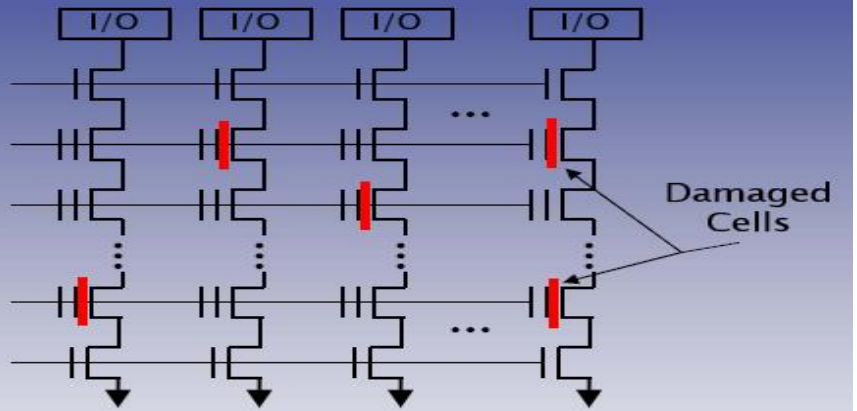
- Limit PROGRAM/ ERASE cycles in blocks that require long retention
- Limit READs to reduce read disturb



Source attributed to–denali.com

NAND (13) Endurance

- PROGRAM/ERASE cycles cause charge to be trapped in the dielectric
- Causes a permanent shift in cell characteristics—not recovered by erase
- Observed as failed program or erase status
- Blocks that fail should be retired (marked as bad and no longer used)



Note: Circuit structures and voltages are representative only. Details vary by manufacturer and technology node.

Source attributed to–denali.com

NAND (14)
Endurance Recommendations

- Always check pass/fail status (SR0) for PROGRAM and ERASE operations
  - Note: READ operations do not set SR0 to fail status
- If fail status after PROGRAM, move all block data to an available block and mark the failed block bad
- Use ECC to recover from errors
- Write data equally to all good blocks (wear leveling)
- Protect block management/meta data in spare area with ECC

Source attributed to–denali.com

NAND (15)

Block Size (pages)	Page Size (bytes)	Main Area (bytes)	Spare Area (bytes)
64	528	512	16
64	2112	2048	64
64	4224	4096	128
128	2112	2048	64
128	4224	4096	128
128	4314	4096	218

Table 1 – Typical block and page sizes.

Source attributed to – micron.com

NAND (16)

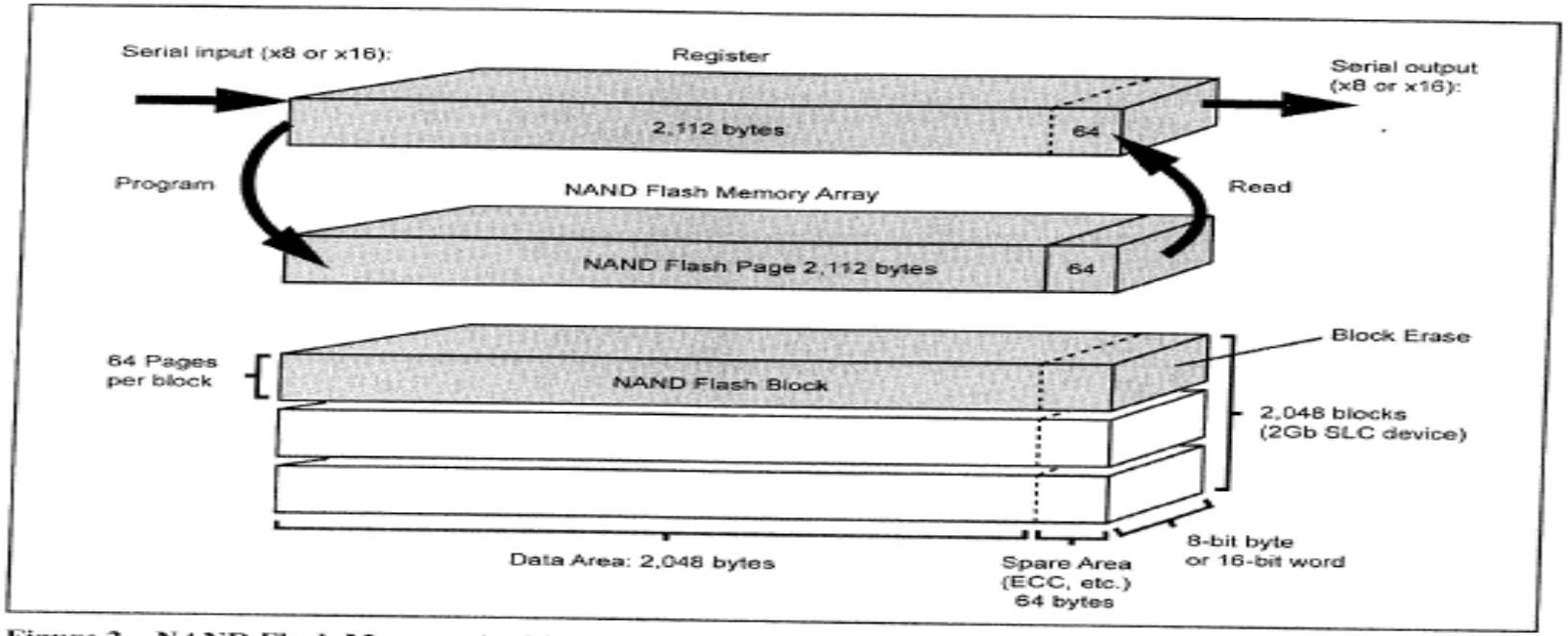
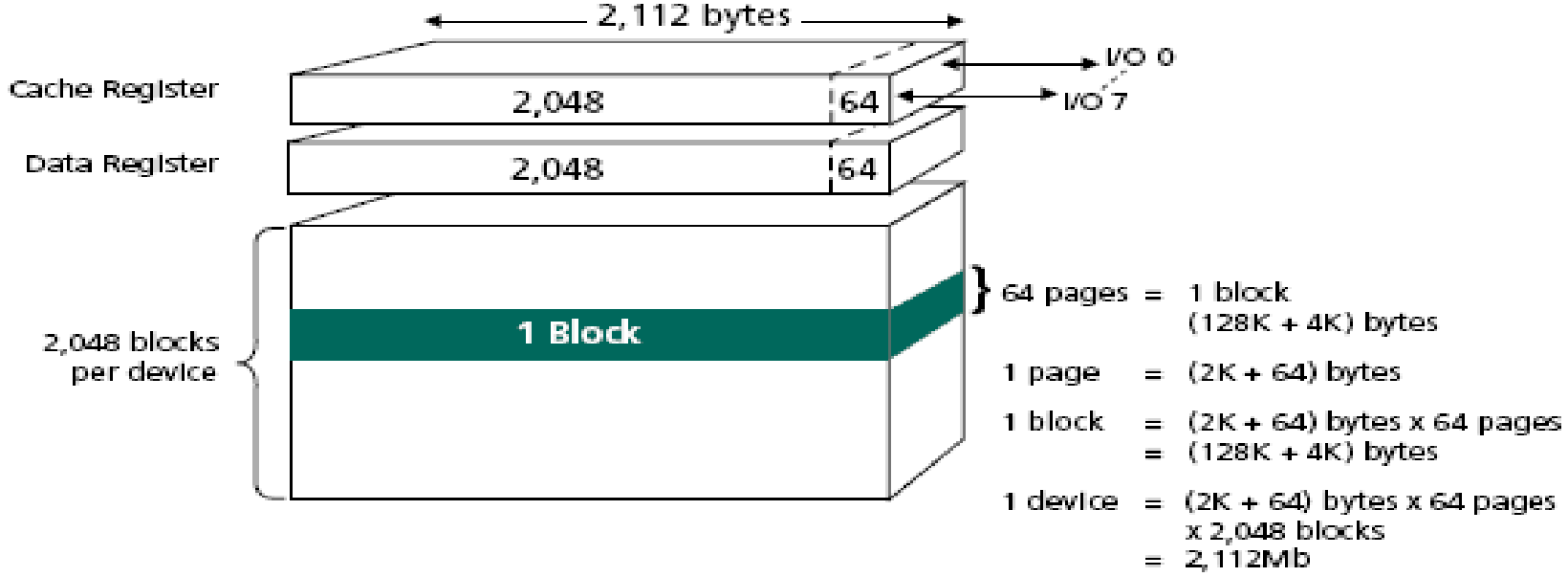


Figure 2 – NAND Flash Memory Architecture.

Source attributed to – mciron.com

NAND/ NOR (17)



Source attributed to – micron.com

NAND (18)

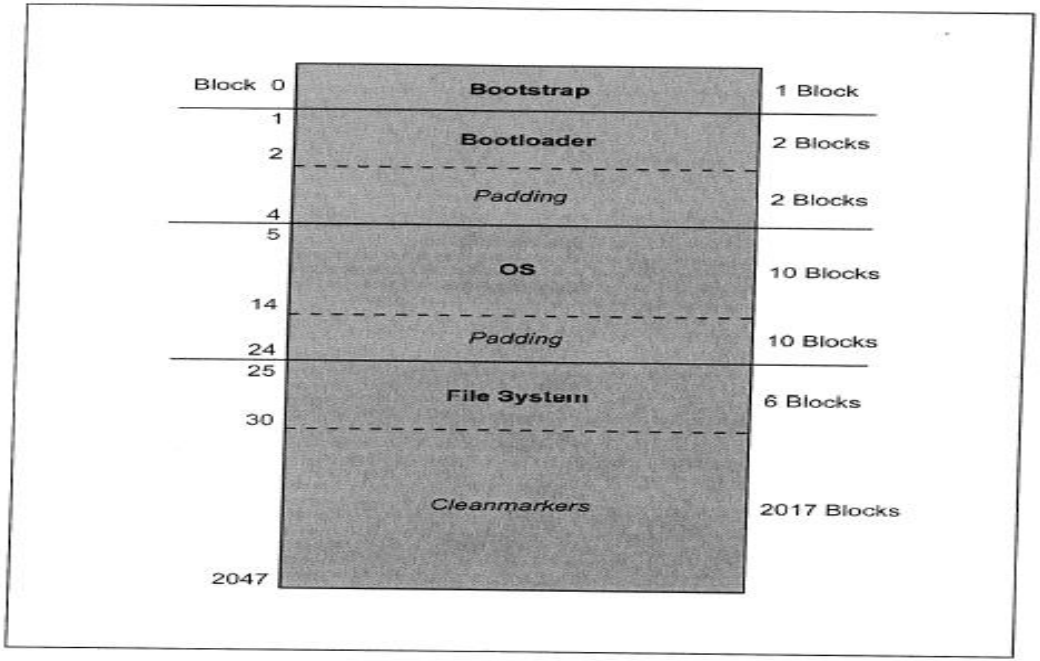


Figure 10 – Example NAND Flash Memory map for an embedded system.

Source attributed to – mciron.com

NOR (19)

- NOR Interface 41 pins

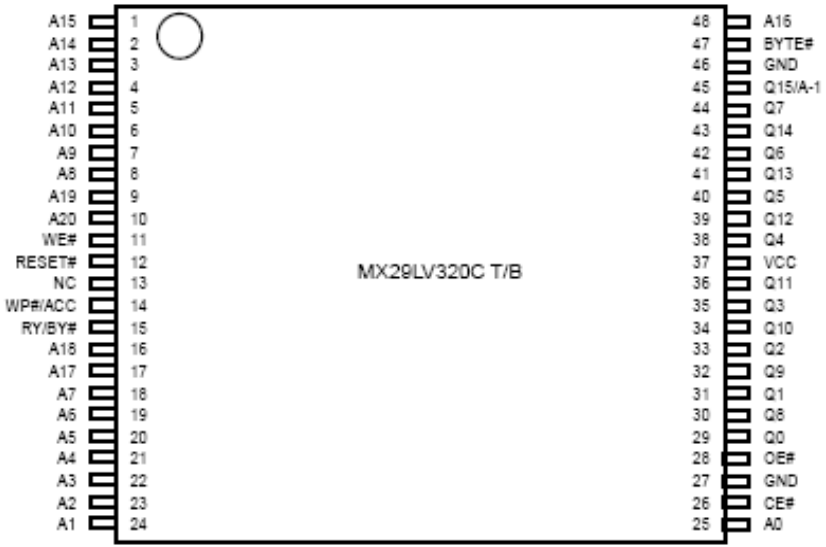
CE# - Chip Enable
WE# - Write Enable
OE#- Output Enable
D0toD15 – Data Bus
A0 to A20 – Address Bus
WP# - Write Protect

NOR(20)

NOR FLASH 32Mb

PIN CONFIGURATION

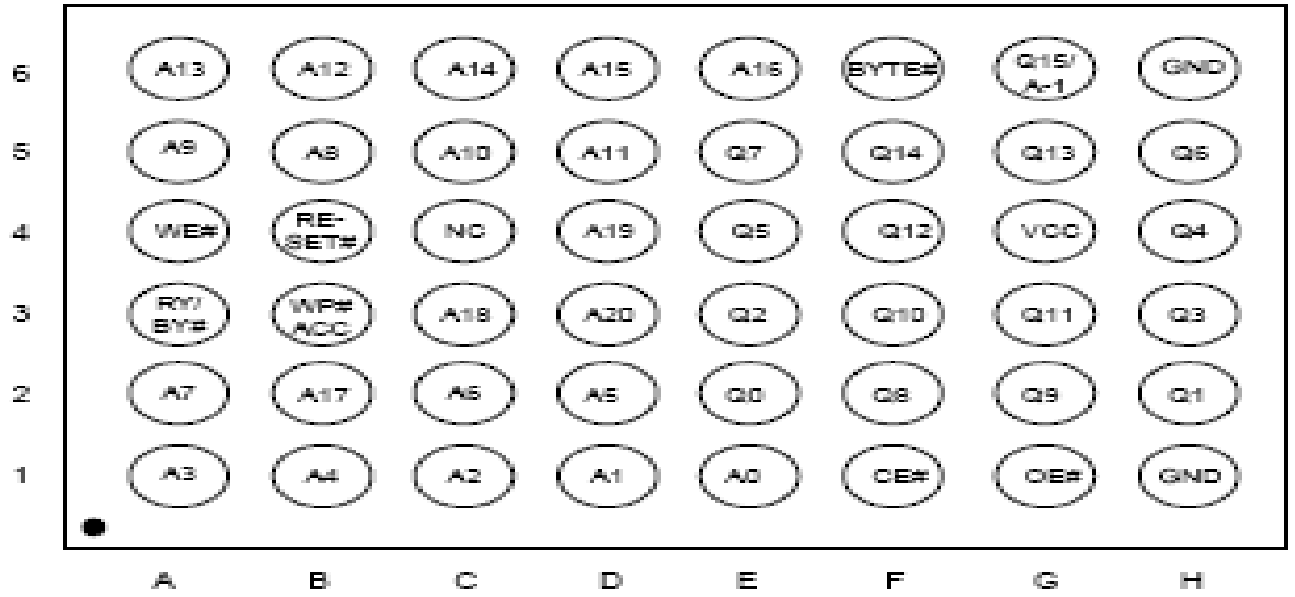
48 TSOP



Source attributed to – macronix.com

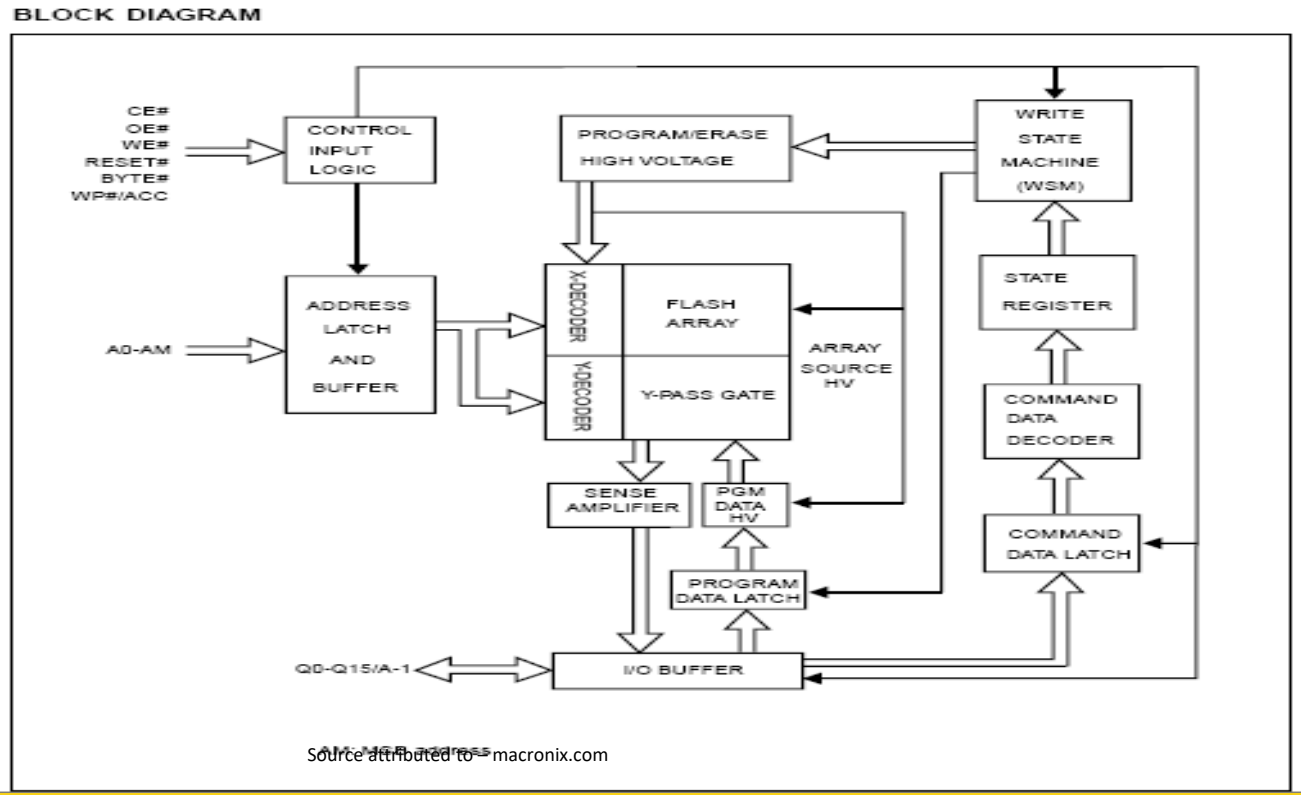
NOR (21) – NOR 32Mb CSP

48-Ball CSP 6mm x 8mm (Ball Pitch = 0.8 mm), Top View, Balls Facing Down



Source attributed to – macronix.com

NOR (22) – NOR BLK Diagram



Source attributed to – macronix.com

NAND (23)

- NAND Interface 24 pins

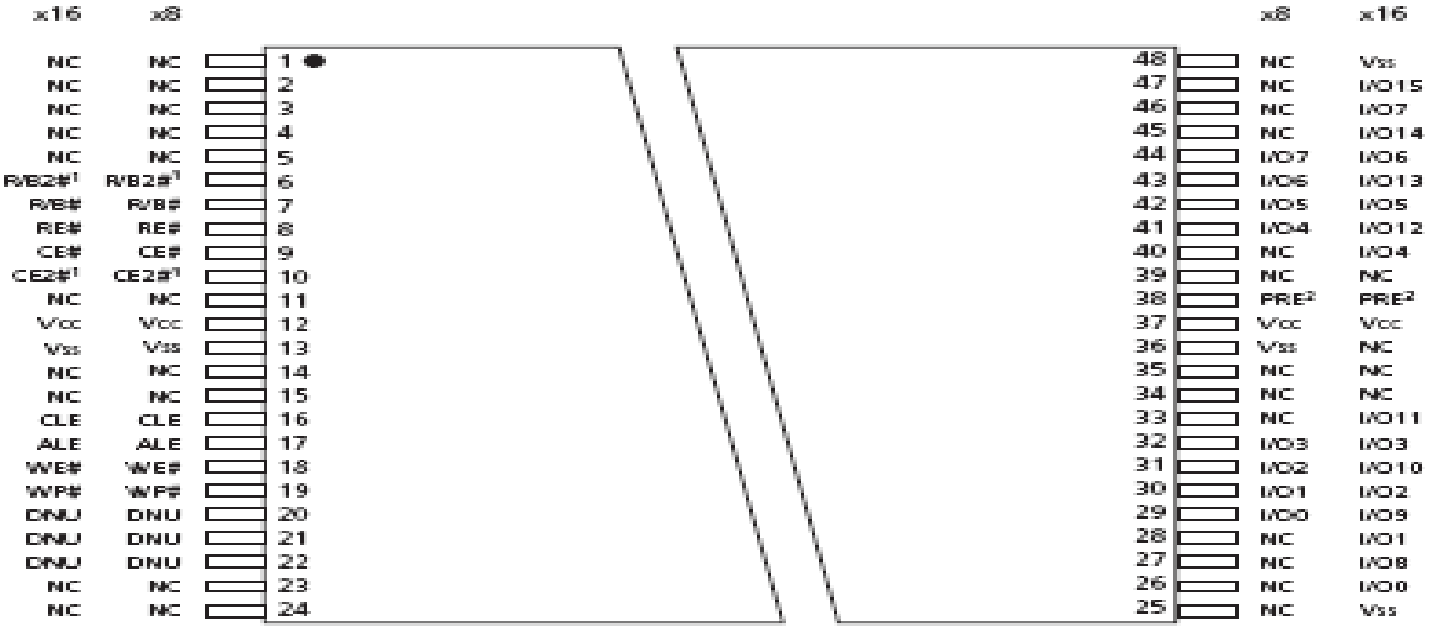
CE# - Chip Enable CLE – Command Latch Enable
WE# - Write Enable ALE – Addr Latch Enable
RE#- Output Enable R/B# - Ready/ Busy
I/O0-15 –IO Bus PRE- Power-on Read Enable

Source attributed to – micron.com

NAND (24)

NAND FLASH 2Gb

Figure 4: 48-Pin TSOP Type 1 Pin Assignments (Top View)



Source attributed to – micron.com

NAND (25)

Table 4: Array Addressing: MT29F2G08AxC

Cycle	I/O7	I/O6	I/O5	I/O4	I/O3	I/O2	I/O1	I/O0
First	CA7	CA6	CA5	CA4	CA3	CA2	CA1	CA0
Second	LOW	LOW	LOW	LOW	CA11[1]	CA10	CA9	CA8
Third	BA7	BA6	PA5	PA4	PA3	PA2	PA1	PA0
Fourth	BA15	BA14	BA13	BA12	BA11	BA10	BA9	BA8
Fifth	LOW	LOW	LOW	LOW	LOW	LOW	LOW	BA16

Notes: 1. If CA11 = "1" then CA[10:6] must be "0."
2. Block address concatenated with page address = actual page address; CAx = column address; PAx = page address, BAx = block address

Source attributed to – micron.com

NAND/NOR(26)

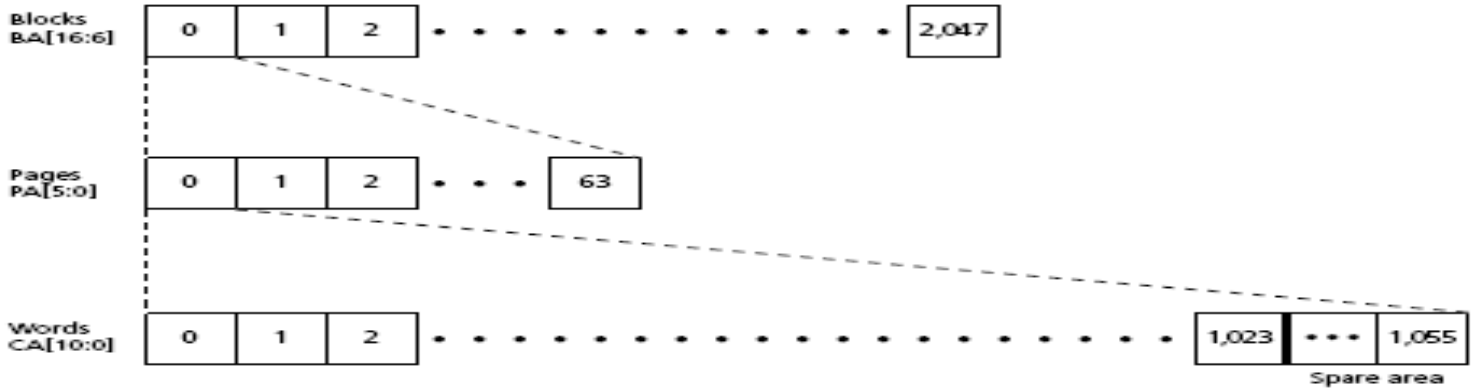


Table 3: Operational Example (x16)

Block	Page	Min Address in Page	Max Address in Page	Out of Bounds Addresses in Page
0	0	0x0000000000	0x000000041F	0x0000000420–0x0000000FFF
0	1	0x0000010000	0x000001041F	0x0000010420–0x0000010FFF
0	2	0x0000020000	0x000002041F	0x0000020420–0x0000020FFF
...	...	...	...	
2,046	62	0x01FFFE0000	0x01FFFE041F	0x01FFFE0420–0x01FFFE0FFF
2,047	63	0x01FFFF0000	0x01FFFF041F	0x01FFFF0420–0x01FFFF0FFF

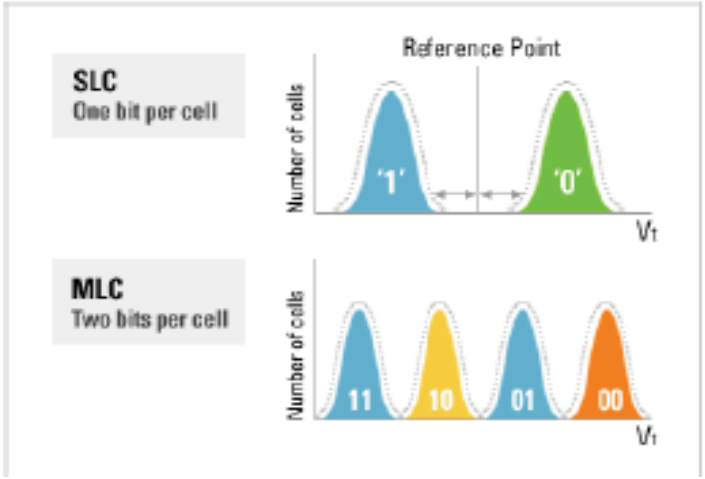
Source attributed to – micron.com

NAND/NOR(27)

What is Multi-level Cell?

SLC stores two states (binary)

MLC stores four states



Source attributed to – micron.com

NAND/ NOR(28)

- Wear Leveling – a method that ensures even distribution of writing the same data on the media.
- Flash has typical limited endurance of 100K erase cycles.
- Typically handled by the file system whereby the same logical address is mapped to a different physical address on the flash.

NAND/ NOR (29)

Wear Leveling

- Wear leveling is a plus on SLC devices where blocks can support up to 100,000 PROGRAM/ ERASE cycles
- Wear leveling is imperative on MLC devices where blocks typically support less than 10,000 cycles
- If a block was erased and reprogrammed every minute, the 10,000 cycling limit would be exceeded in just 7 days!

  $$60 \times 24 \times 7 = 10{,}080$$

- Rather than cycling the same block, wear leveling involves distributing the number of blocks that are cycled

Source attributed to – denali.com

NAND/ NOR (30)
Wear Leveling – cont'd

- An 8Gb MLC device contains 4,096 independent blocks
- Using the previous example, if the cycles were distributed over 4,096 blocks, each block would be programmed less than 3 times (vs. 10,800 cycles if the same block is cycled)
- If perfect wear leveling was performed on a 4,096-block device, a block could be erased and programmed every minute, every day for 77 years!

$$\frac{10{,}000 \times 4{,}096}{60 \times 24} = \frac{40{,}960{,}000}{1{,}440} = 28{,}444 \text{ days} = 77.9 \text{ years}$$

- Consider static vs. dynamic wear leveling

Source attributed to – denali.com

NAND (31)

$/MB	DRAM	NAND flash
2000	$0.97	$1.35
2001	0.22	0.43
2002	0.22	0.25
2003	0.17	0.21
2004	0.17	0.10
2005e	0.11	0.05
change	-88.7%	-96.3%

Source attributed to – denali.com

NAND (32)

The Need for Over-provisioning NAND Flash Memory

NAND flash memory is unlike both random access memory and magnetic media, including hard disk drives, in one fundamental way: there is no ability to overwrite existing content. Instead, entire blocks of flash memory must first be erased before any new pages can be written.

With a hard disk drive (HDD), for example, that act of "deleting" files affects only the metadata in the directory. No data is actually deleted on the drive; the sectors used previously are merely made available as "free space" for storing new data. This is the reason "deleted" files can be recovered (or "undeleted") from HDDs, and why it is necessary to actually erase sensitive data to fully secure a drive.

With NAND flash memory, by contrast, free space can only be created by actually deleting or erasing the data that previously occupied any block of memory. The process of reclaiming blocks of flash memory that no longer contains valid data is called "garbage collection." Only when the blocks, and the pages they contain, have been cleared in this fashion are they then able to store new data during a write operation.

The flash storage processor (FSP) is responsible for managing the pages and blocks of memory, and also provides the interface with the operating system's file subsystem. This need to manage individual cells, pages and blocks of flash memory requires some overhead, and that in turn, means that the full amount of memory is not available to the user. To provide a specified amount of user capacity it is therefore necessary to over-provision the amount of flash memory, and as will be shown later, the more over-provisioning the better.

The portion of total NAND flash memory capacity held in reserve (unavailable to the user) for use by the FSP is used for garbage collection (the major use); FSP firmware (a small percentage); spare blocks (another small percentage); and optionally, enhanced data protection beyond the basic error correction (space requirement varies).

Even though there is a loss in user capacity with over-provisioning, the user does receive two important benefits: better performance and greater endurance. The former is one of the reasons for using flash memory, including in solid state drives (SSDs), while the latter addresses an inherent limitation in flash memory.

Source: http://www.gallagherpr.com/wp-content/uploads/2013/05/edn_Understanding_SSD_Overprovisioning.pdf

NAND (33)

Percentage Over-provisioning

The equation for calculating the percentage of over-provisioning is rather straightforward:

$$\text{Percentage Over-provisioning} = \frac{\text{Physical Capacity - User Capacity}}{\text{User Capacity}}$$

For example, in a configuration consisting of 128 Gigabytes (GB) of flash memory total, 120 GB of which is available to the user, the system is over-provisioned by 6.7 percent, which is typically rounded up to 7 percent:

$$\frac{128\ \text{GB} - 120\ \text{GB}}{120\ \text{GB}} = 0.067$$

Source: http://www.gallagherpr.com/wp-content/uploads/2013/05/edn_Understanding_SSD_Overprovisioning.pdf

NAND (34)

It is also important to note another factor that often causes confusion: a binary Gibibyte is not the same as a decimal Gigabyte. As shown in Figure 1, a binary GB is 7.37 percent larger than a decimal GB. Because most operating systems display the binary representation for both memory and storage, this causes over-provisioning to appear smaller because the actual number of bytes is 7.37 percent higher than the number of bytes displayed. This is why an SSD listed as providing 128 GB of user space can still function with 128 GB of physical memory. Using the calculation above, the over-provisioning amount would appear to be zero percent, which is impossible for NAND flash. In reality it is really over-provisioned closer to 0 + 7.37 percent.

	Binary	Decimal
Exponential Notation	2^30	10^9
Actual Number of Bytes	1,073,741,824	1,000,000,000
Naming Convention	Gibibyte (IEC)	Gigabyte (SI)
Typical Uses	Memory	Storage

IEC – International Electrotechnical Commission
SI – International System of Units

Figure 1. The difference between a binary Gigabyte and a decimal Gigabyte

Source: http://www.gallagherpr.com/wp-content/uploads/2013/05/edn_Understanding_SSD_Overprovisioning.pdf

NAND (35)

Test Environment

To isolate the over-provisioning variable, the tests were conducted on a single SSD with Toshiba MLC (multi-level cell) 24nm NAND flash memory controlled by an LSI SF-2281 flash storage processor. It is important to note that the FSP used employs the LSI DuraWrite™ technology that optimizes writes to flash memory, and utilizes intelligent block management and wear-leveling to improve reliability and endurance. These capabilities combine to afford over five years of useful life for MLC-based flash memory with typical use cases.

Previous testing performed by LSI revealed that entropy has an effect on performance only for SSDs without data reduction technology. For this reason, the red lines in the graphs showing the results for 100% entropy are labeled “Typical SSDs.” This series of tests, which used SSDs equipped with LSI DuraWrite data reduction technology, were designed to evaluate performance at different levels of both over-provisioning and entropy, and to specifically test the hypothesis that data reduction could improve performance at lower levels of entropy.

Test result data points are based on post-garbage collection, steady state operation. All preconditioning used the same transfer size and type as the test result (e.g. random 4KB results are preconditioned with random 4KB transfers until reaching steady state operation).

VDBench V5.02 was used as the main test software with IOMeter V1.1.0 providing cross-check verification. The test PC was configured with an Intel Core i5-2500K 3.30 GHz processor, the Intel H67 Express chipset, Intel Rapid Storage Technology 10.1.0.1008 (with AHCI Enabled); 4 GB of 1333 MHz RAM; and Windows 7 Professional (32-bit).

Source: http://www.gallagherpr.com/wp-content/uploads/2013/05/edn_Understanding_SSD_Overprovisioning.pdf

NAND (36)

Performance Test Results

Sequential writes were uniform across all tested over-provisioning ranging from zero to 75 percent. This flat performance derives from the nature of sequential writes to flash. As data is written to flash memory, it completely fills all of the pages in a block. When the drive becomes filled, blocks of data that are no longer valid need to be erased first via the garbage collection process, which it does by simply erasing entire blocks without needing to move (read then write) any individual pages that might otherwise still be valid. Because there are no incremental writes during garbage collection during this operation, there is no benefit from additional free space. With SSDs that use a data reduction technology like DuraWrite from LSI, the level of flat performance will increase as a function of the entropy (data randomness); the lower the entropy the higher the performance. In this situation, however, the increase in performance is due to the reduced writes being completed sooner and not from the additional free space.

Throughput performance for sustained 4KB random writes improved as the amount of over-provisioning increased. Additionally, for SSDs with DuraWrite data reduction technology, the throughput improvement also increased at all levels of entropy.

Figure 2 shows the results of this test. The reason why the increased over-provisioning improves performance for random writes is due to how garbage collection operates. As data is written randomly, the logical block addresses (LBAs) being updated are distributed across all the blocks of the flash. This causes a number of small “holes” of invalid data pages among valid data pages. During garbage collection those blocks with invalid data pages require the valid data to be read and moved to new empty blocks. This background read and write operation requires time to execute and prevents the SSD from responding to read and write requests from the host, giving the perception of slower overall performance.

Source: http://www.gallagherpr.com/wp-content/uploads/2013/05/edn_Understanding_SSD_Overprovisioning.pdf

NAND (37)

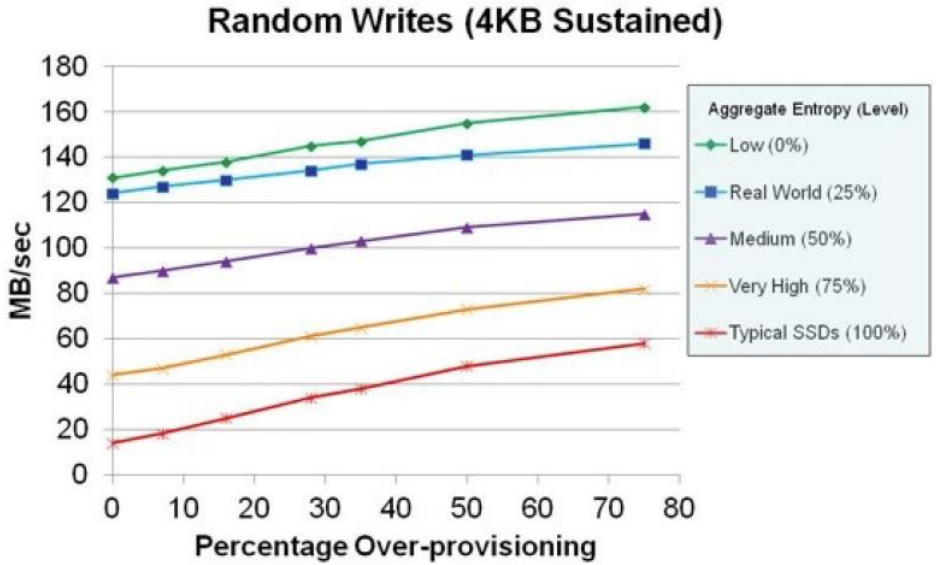


Figure 2. The effect of over-provisioning on write performance throughput

Source: http://www.gallagherpr.com/wp-content/uploads/2013/05/edn_Understanding_SSD_Overprovisioning.pdf

NAND (38)

The need for garbage collection and wear-leveling with NAND flash memory causes the amount of data being physically written to be a multiple of the logical data intended to be written. This phenomenon is expressed as a simple ratio called "write amplification," which ideally would approach 1.0 for standard SSDs with sequential writes, but typically is much higher due to the addition of random writes in most environments. With SSDs that have DuraWrite technology, the typical user experiences a much lower write amplification that is often on average only 0.5. Getting write amplification low is important to extending the flash memory's useful life.

Random write operations have the greatest impact on write amplification, so to best view the effect of over-provisioning on write amplification, tests were conducted under those conditions. As shown in Figure 3, write amplification for sustained 4KB random writes benefited significantly from a higher percentage of over-provisioning for SSDs that do not include DuraWrite technology. For SSDs that do include DuraWrite or a similar data reduction technology, the throughput improvement increased at a higher rate at higher levels of entropy.

Note also how the use of a data reduction technology like DuraWrite minimizes the benefits of over-provisioning for lower levels of entropy. When the entropy of the user data is low, DuraWrite is able to reduce the amount of space consumed in the flash memory. Because the operating system is unaware of this reduction, the extra space is automatically used by the flash storage processor as additional over-provisioning space. As the entropy of the data increases, the additional free space decreases. At 100 percent entropy the additional over-provisioning is zero, which is the same result as a "Typical SSD" (red line) that does not employ a data reduction technology. Referring again to Figure 3, a standard SSD with 28 percent over-provisioning would have the same write amplification as an SSD with DuraWrite technology at zero percent over-provisioning for data with an entropy as high as 75 percent.

Source: http://www.gallagherpr.com/wp-content/uploads/2013/05/edn_Understanding_SSD_Overprovisioning.pdf

NAND (39)

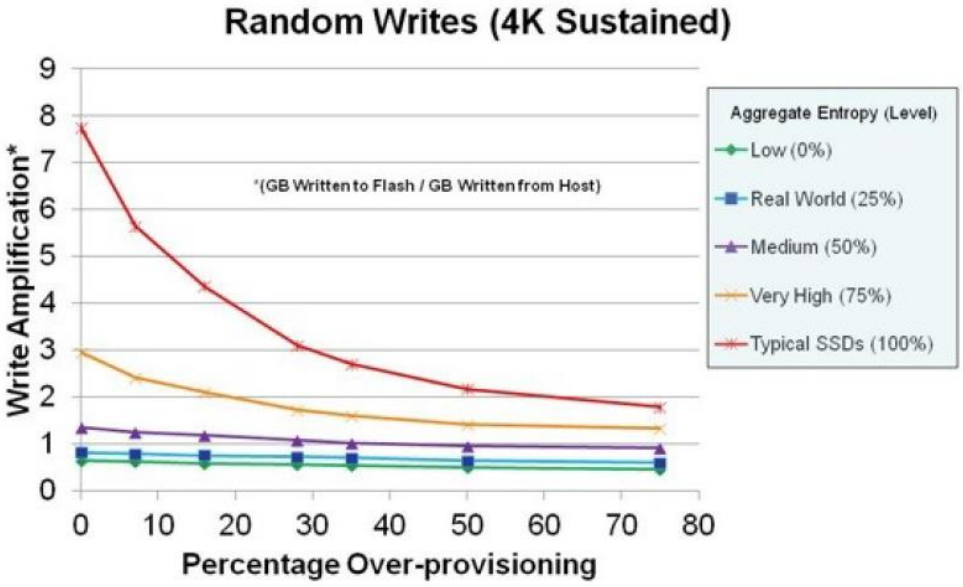


Figure 3. The effect of over-provisioning on write amplification

Source: http://www.gallagherpr.com/wp-content/uploads/2013/05/edn_Understanding_SSD_Overprovisioning.pdf

NAND (40)

With the advent of SSDs, and the need to manage them differently from traditional HDDs, a TRIM command was added to storage protocols to enable operating systems to designate blocks of data that are no longer valid. Until the SSD is informed the data is invalid with a write to a currently occupied LBA, it will continue to save that data during the garbage collection process, resulting in less free space and higher write amplification. TRIM enables the SSD to perform its garbage collection and free up the storage space occupied by invalid data in advance of future write operations.

Figure 4 shows the effect of the TRIM command on over-provisioning. For a “marketed” percentage of over-provisioning (28 percent in this example), the amount effectively increases after performing a TRIM operation. Note how the capacity originally designated as Free Space remains consumed as Presumed Valid Data by the SSD after being deleted by the operating system or the user until a TRIM command is received. In effect, the TRIM operation provides dynamic over-provisioning because it increases the resulting over-provisioning after completion.

Source: http://www.gallagherpr.com/wp-content/uploads/2013/05/edn_Understanding_SSD_Overprovisioning.pdf

NAND (41)

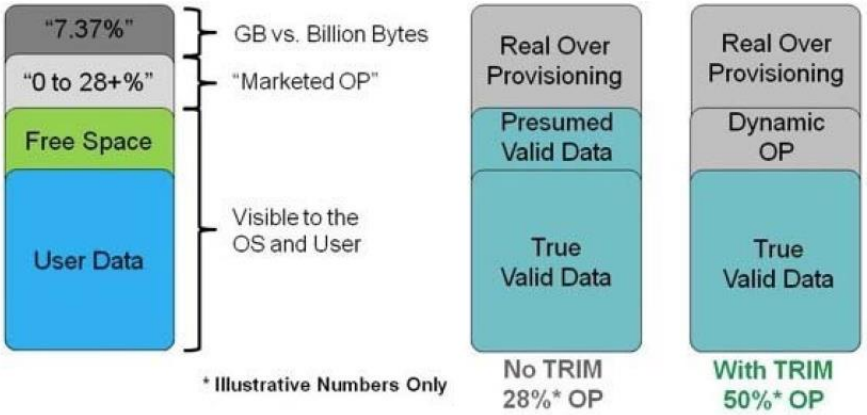


Figure 4. The effect of the TRIM command on over-provisioning percentage

Conclusion

The over-provisioned capacity of NAND flash memory creates the space the flash storage processor needs to manage the flash memory more intelligently and effectively. As shown by these test results, higher percentages of over-provisioning improve both write performance and write amplification. Higher percentages of over-provisioning can also improve the endurance of flash memory and enable more robust forms of data protection beyond basic error correction.

Source: http://www.gallagherpr.com/wp-content/uploads/2013/05/edn_Understanding_SSD_Overprovisioning.pdf

NAND (42)

Only SSDs that utilize a data reduction technology, such as DuraWrite in the LSI SandForce flash storage processors, can take advantage of lower levels of entropy to improve performance based on the increase in “dynamic” over-provisioning.

Owing to the many benefits of over-provisioning, a growing number of SSDs now enable users to control the percentage of over-provisioning by allocating a smaller portion of the total available flash memory to user capacity during formatting. With increased capacities based on the ever-shrinking geometries of NAND flash memory technology, combined with steady advances in flash storage processors, it is reasonable to expect that over-provisioning will become less of an issue with users over time.

Source: http://www.gallagherpr.com/wp-content/uploads/2013/05/edn_Understanding_SSD_Overprovisioning.pdf

Embedded Flash MCU

Manufacturer	Renesas	Infineon	Microchip	Fujitsu	ST Micro	Toshiba	Atmel
Part Number	SH7211F	SAK-XC2267-96F66L82	PIC32MX440F128L	MB91F318R	STM32F103	TMPM330FDFG	AT32UC3A0512-ALUT
Product Family	SuperH 32-bit MCU	XC2000 32-bit MCU	MIPS-based 32-bit Flash MCU	FR60 32-bit RISC MCU	Medium-density performance line, ARM based 32-bit MCU	TMPM330 32-bit RISC Micro-processor	AVR32 UC 32-bit MCU
Target Applications	A wide range of applications	Automotive applications	USB	Embedded control in TV or PDP	A wide range of applications	A wide range of applications	Battery/USB powered applications
Processor Core	SH-2A	C166SV2	MIPS32 M4K Core	FR60	ARM Cortex-M3	ARM Cortex-M3	AVR32 UC
CPU Clock	160MHz	66MHz	80MHz	40MHz	72MHz	40MHz	66MHz
Embedded Flash	512KB	768KB	512KB	1MB	128KB	512KB	512KB
Embedded RAM	32KB	82KB	32KB	48KB	20KB	32KB	64KB
Power Supply	1.5V CPU, 3.3V IO, 5V A/D Converter	3.0~5.5V	2.3~3.6V	1.8/3.3V	2.0~3.6V	2.7~3.6V	Single 3.3V or dual 1.8/3.3V
Package	144-LQFP	100-LQFP	100-TQFP	176-LQFP	48-LQFP	100-LQFP	144-LQFP

Source attributed to– renesas.com

Next Class:
Friday May 17th
6:30pm PST