ruSolut
Data Investigation & Recovery Technology

VISUAL NAND RECONSTRUCTOR

The book

Part1. Fundamentals

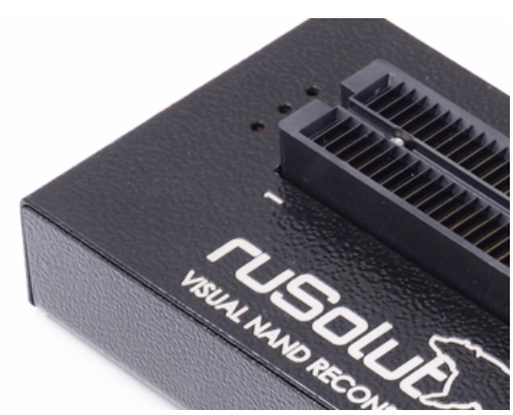
www.rusolut.com

Visual Nand Reconstructor hardware

The VNR hardware consists of flash memory reader and a set of adapters for different NAND chip packages.

The reader is connected up to the computer with a mini USB 2.0 plug. It has to be connected to start software and save data.

NAND adapter installation panel ZIF48 located on the top side.



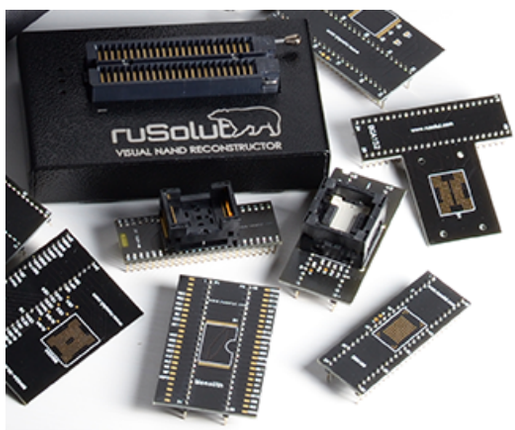
Adapter positioning key is marked by digit "1" on the reader and adapter.

Incorrect adapter installation may cause memory chip damage!

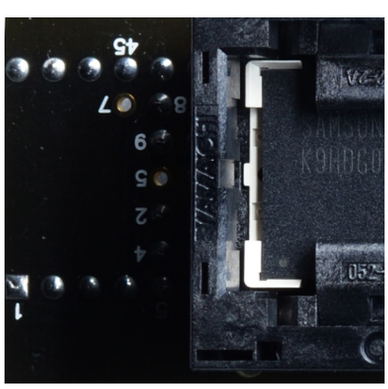
The reader has three LEDs. The green led means USB power. The yellow one means that the power is applied to the chip. The red one means the critical error – reader must be reconnected to USB port.

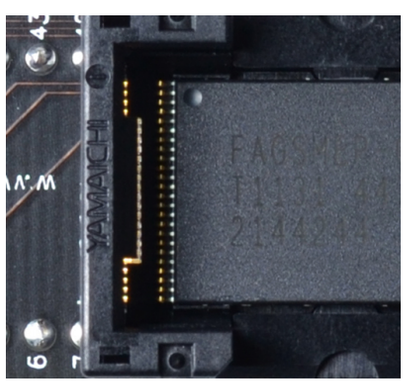
It's forbidden to put out the chip when the power is applied and the yellow is alight, it may cause memory chip damage.

The VNK kit contains two types of NAND adapters - easy chip installation (with socket) and chip soldering.



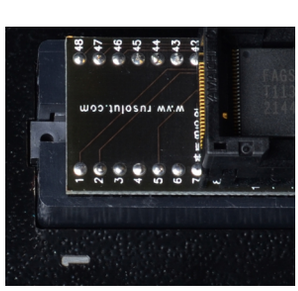
To connect memory chip to adapter it's necessary to keep up orientation about the Pin1 on the chip case, it is a point at corner.





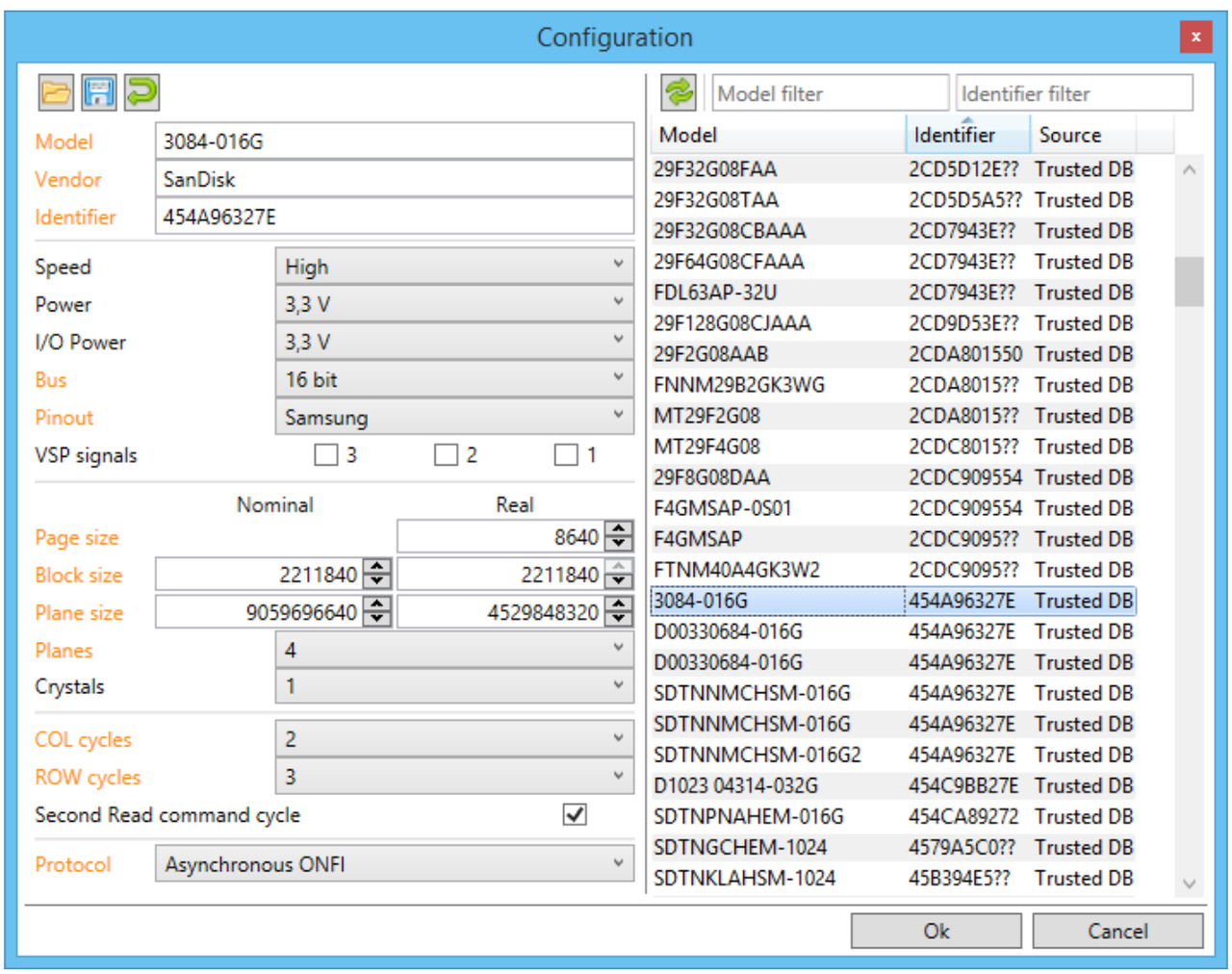
Wrong installation may cause memory NAND chip damage!

To install the adapter it's necessary to turn the lever up, insert the adapter accordingly with the key “1” and turn the lever down until it clicks.



NAND flash chips and its parameters

To extract the physical image (read dump) from memory chip to file it's necessary to read chip ID and to set reading parameters. VNR contains chip configs database. For most of them the parameters are defined automatically after ID reading. If the chip is new and it there's data sheet, it can be added to the base manually. For chips that satisfy the ONFI specification requirements, automatic settings detection is possible (Hynix, Sandisk, Intel, Micron, Spansion) by using ONFI page parameters.

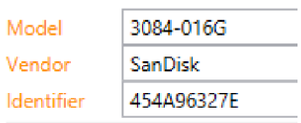


Instruction how to work with NAND chip can be found on the website:
http://rusolut.com/direct-access-to-nand-and-physical-image-exraction/

All the chip parameters can be tentatively divided into four groups – identification parameters, physical parameters, crystal geometry parameters, reading protocol parameters

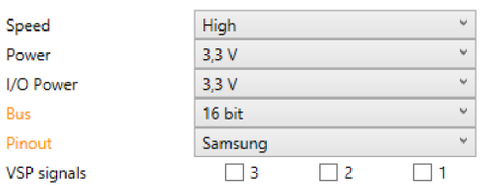
Identification parameters

model
vendor
identifier (ID)



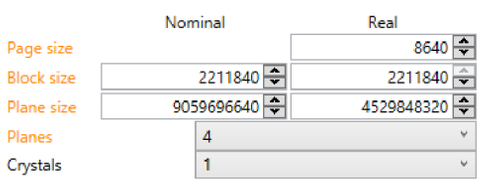
Physical parameters

speed
power
I/O power
bus



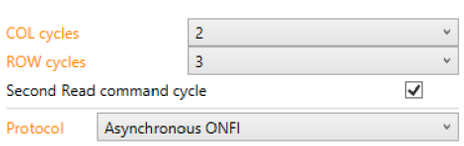
The crystal geometry parameters

page size (bytes)
nominal block size (bytes)
real block size (bytes)
nominal plane size (bytes)
real plane size (bytes)
number of planes in crystal (1/2/4)
number of crystals in the chip (1/2/4; CE pins)



Protocol parameters

COL cycles
ROW cycles
second read command cycle
data transfer protocol



Speed determines the signal frequency while working with flash memory chip.
VNR Reader supports 3 speed modes: High (~8Mb/s), Medium (~4Mb/s), Low(~1,5 Mb/s).

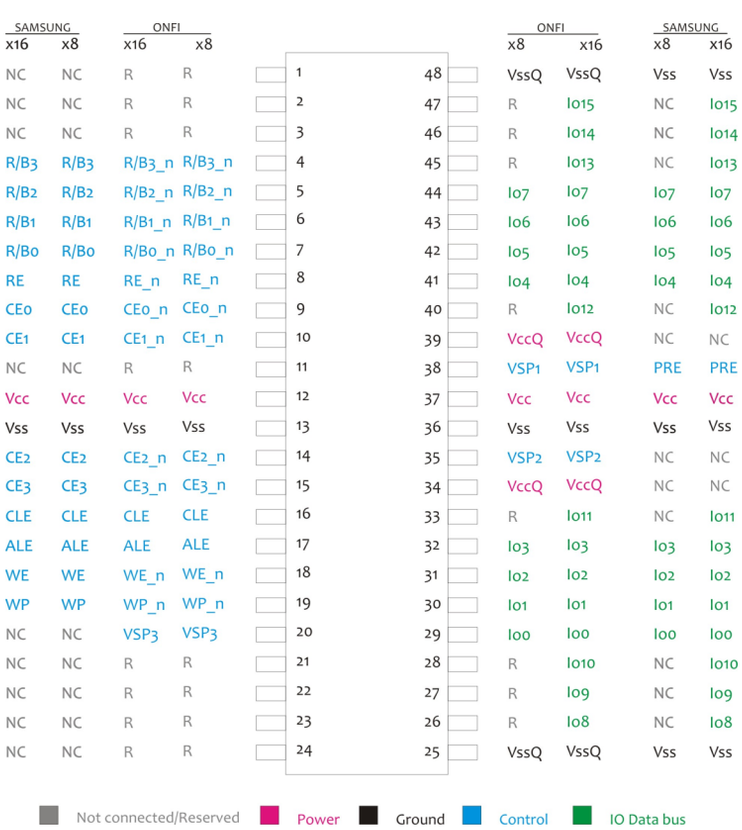
Power determines the voltage level on power input Vcc of flash memory chip (pins 12& 37)

I/O power determines the voltage level of I/O ports VccQ (pins 34& 39)
The VNR reader supports the following power levels:
1,6V; 1,8V; 2,0V; 2,3V; 2,5V; 2,7V; 3,0V; 3,3V; 3,6V; 4,0V.
The power adjustment is required for chips with a high number of bit errors. Lower power equals to lower noise and better data quality.

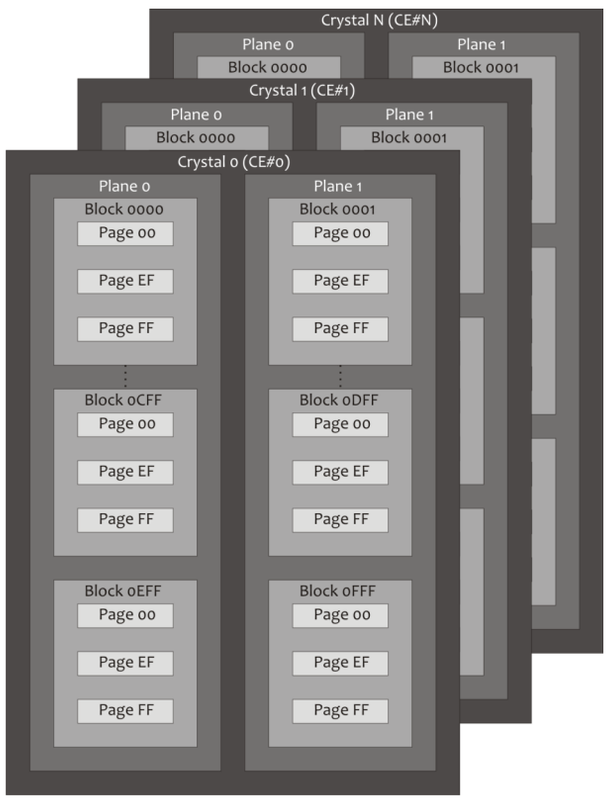
Bus determines the number of data transmission lines (IO bus). It can be 8 bit & 16 bit. Both of modes are supported on hardware level.

Pinout determines the signal assignment of pins according to the picture below.

VSP signal state (vendor specific pin) determines the signal state of VSP1 (pin 38), VSP2 (pin 35), VSP3 (pin 20). Default state is 0, when flag on it equals to 1.



Flash memory has following physical structure:
NAND chip -> Crystal-> Plane-> Block-> Page



Page size determines minimal data size for read operation.

Block size (real) determines minimal data size for write and erase operations (PE cycles).

Block size (nominal) and real block size value are used only in the case of TLC (Tripple Level Cell) flash memory chips. Usually in those chips nominal block size is multiple of 4, but real block size is multiple of 3. For example, if nominal block size is equal 256 pages, real block size is equal 192 pages. The remaining 64 pages are addressed inside flash memory, but physically they are not exist. Therefore, for correct work with data it's necessary to consider these admissions in address space. There's similar situation with plane sizes. In all other cases, except TLC chips, Nominal size = Real size.

Plane size (real) determines capacity of crystal.

Plane size (Nominal) used in case of TLC memory, in all other cases its

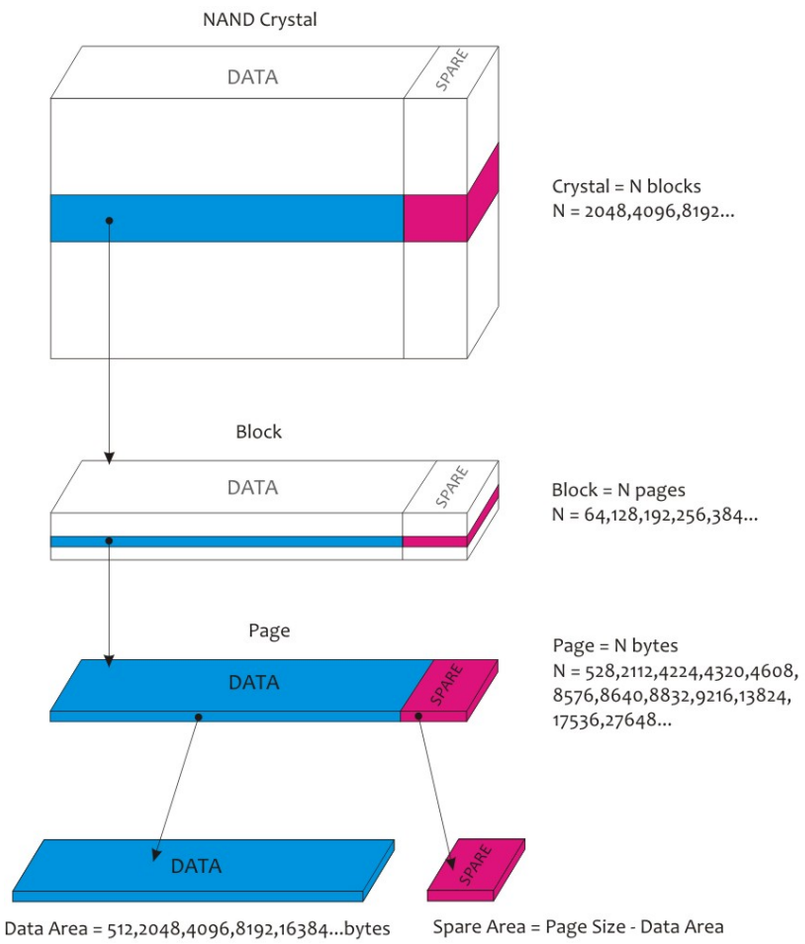
COL and ROW cycles determines how many bytes are necessary for all flash memory address space. By default it must be set as CLO=2, ROW=3.

Second read command cycle flag determines if it's necessary to use second cycle read command 0x30 according to JEDEC standard.

Protocol determines the memory chip data transfer method. There are 4 protocols:
- Asynchronous ONFI/SDR
- DDR
- WL Tripple address
- WL Tripple address + DDR

Data organization in NAND flash chips

According to the simplified physical structure NAND flash memory is divided into blocks and pages.



To describe the interior structure and functionality of the flash memory storage device, we use the following terms: physical block, virtual block, physical page, virtual page, data area (logical sector), spare area.

Physical block. Flash memory device is divided into physical memory blocks. They are minimal areas of memory, which can be erased for one erase operation. The physical block size depends on organization of particular flash memory and can be: 64, 128, 256, 512, 1024, 2048 Kbytes, or in case of TLC memory cells – 1.5, 3 Mbytes (the size is denoted excluding Spare Area).

Virtual block. A storage device on flash memory works by virtual memory blocks, which are organized from one or several physical blocks. It depends on the storage device organization and multi-plane page allocation scheme. Virtual block corresponds to a logical block of drive in accordance with its LBN (Logic Block Number), which is stored in its page spare area.

Physical Page. Each physical memory block divided into small memory areas which are operated by the controller as a minimal data volume for reading and writing. The page size depends on organization of concrete flash memory and can be 512, 2048, 4096, 8192, 16384 bytes (the size is denotes excluding service area).

Virtual page. It consists of several physical pages and corresponds to the logical page (in logical block) in accordance with the LPN number (Logic Page Number). Virtual block consists of virtual pages.

Data area (logical sector) is a minimal memory area, which is available to OS and can be available to Flash device as on block device memory. Usually logical sector size is 512 bytes. I order to read one sector controller needs to read whole page to buffer.

Spare area (SA). Is assigned for storing controller's service data and ECC code, and doesn't take part in logical address space (LBA) formation.

Flash controllers

Flash controllers are multifunctional devices which perform following functions:

- data reading and writing

- wear optimization of flash memory

- translation of physical chip space into logical and backwards (PBA-LBA; LBA-PBA)

The controller reads/records data into NAND memory by blocks. Because of memory cell low endurance, modern controllers optimise recording operation, transforming the user's data in a set manner.

Virtual block recording into physical blocks of flash memory chip carried non-linearly. That is, logical blocks with sectors of file-system are recorded in NAND's physical space inconsistently, in accordance with translation algorithm. In practice, it mix data inside the chip. It means that sectors with file-system structures such as MBR, FAT Tables can be recorded in the middle or in the end of memory chip unlike the linear HDD translation.

Different models of controllers have their own peculiarities. It makes the data recovery process from NAND memory nontrivial. Therefore, the main goal in chip-off digital forensics and data recovery is determination of controller's work and its algorithm parameters. Visual Nand Reconstructor emulates the controller and convert physical image into logical through virtual transformatons, from which it's possible to analyze user's data and recover files.

Flash Translation Layer (FTL)

All the flash devices on the physical layer of multilevel data storage system, use program layer of data management – Flash Translation Layer (FTL). FTL implements the block device emulation (HDD-like emulation) and performs following functions:

- Wear Leveling (WL) of memory cells

- Allocation of logical blocks with data in physical blocks (Block Mapping algorithm (BM))

- Bad Block Management (BBM) and Reserved Block Allocation

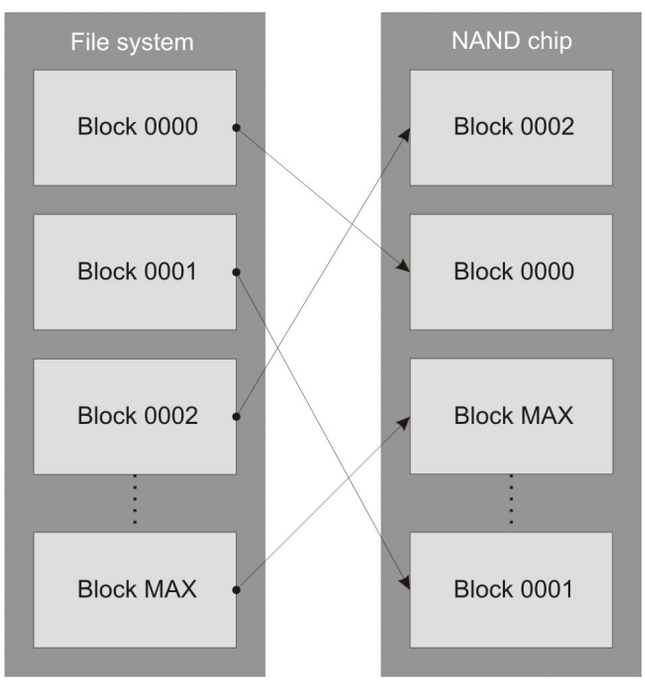
- NAND clean-up from obsolete data (Garbage collection (GC))

- Block recovery after storing obsolete data (Block Reclamation)

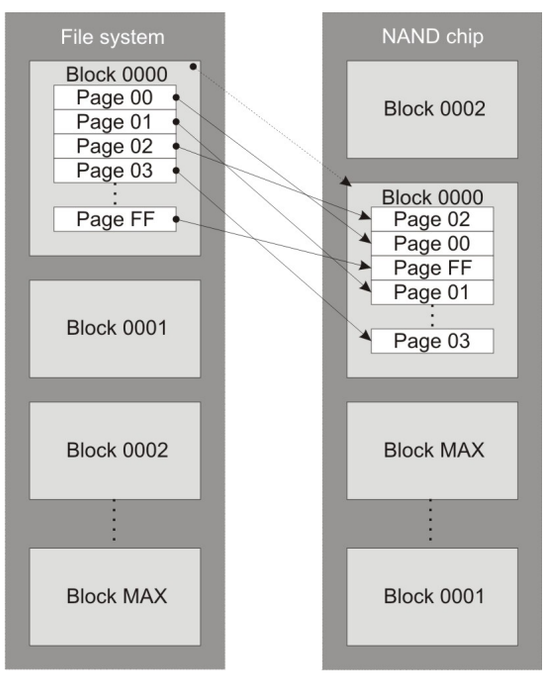
- Logical block sequence reconstruction after power loss (also at flash storage device initialization)

Wear Leveling algorithm is designed for the longevity of the NAND and wear level equalization of a memory cells (blocks) in the chip, relatively to the average, without reference to the block number. The main types of wear-leveling algorithms of memory cells are: Dynamic WL, Static WL, FAT Filter.

Block Mapping algorithm is the most important FTL function from a point of view of reverse data analysis, since this part is responsible for the allocation of logical sectors in physical pages/blocks of NAND chip. Different controllers use different translation schemes, but, in general, translation mechanism can be presented as translation of logical/virtual blocks into physical ones, and translation of virtual pages into physical. Most of controllers use only block translation (page translation is linear within block), some use block and page translation.

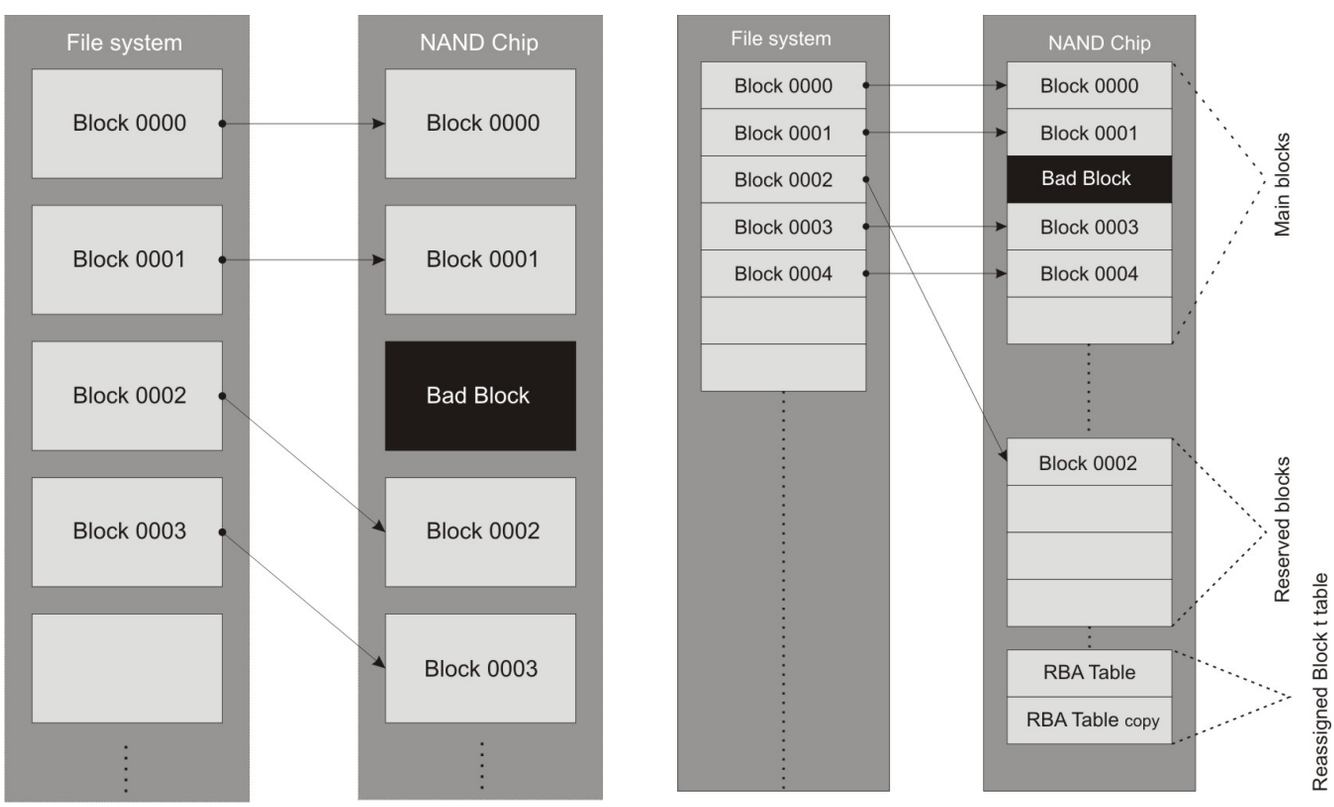


Non-linear translation of blocks with linear page translation



Non-linear translation of blocks with non-linear page

Bad Block Management algorithm controls physical blocks of NAND memory and detect bad blocks. Particularly, after erase and write operations, the ability of memory cells to store data can be measured using special method - verification by Error Correction Code (ECC), that is stored in Spare Area (SA) of each page. If the block failure is detected, FTL marks such block as Bad Block, records special information in SA and move data to another block. There are several algorithms of BB management. The most popular are: algorithm with BB omission and algorithm with BB reallocation from reserved area (similarly to P-list and G-list algoithms of HDD translator).



On the picture logical blocks written into chip sequentially (lineary) for better understanding of principles. In real NAND chip they always mixed.

Garbage Collection and Block Reclamation algorithms are necessary for inactive data erasing in blocks of flash memory and staging them in queue for new data recording. This controller function presents a great interest for digital forensic analysis, because, when data is erased from a flash storage device, it is usually marked as garbage and remain within the chip indefinitely. It leads to the fact, that in many cases we can recover old versions of filesystem metadata, files or some text/image fragments, which are unavailable via the USB connection of working storage device.

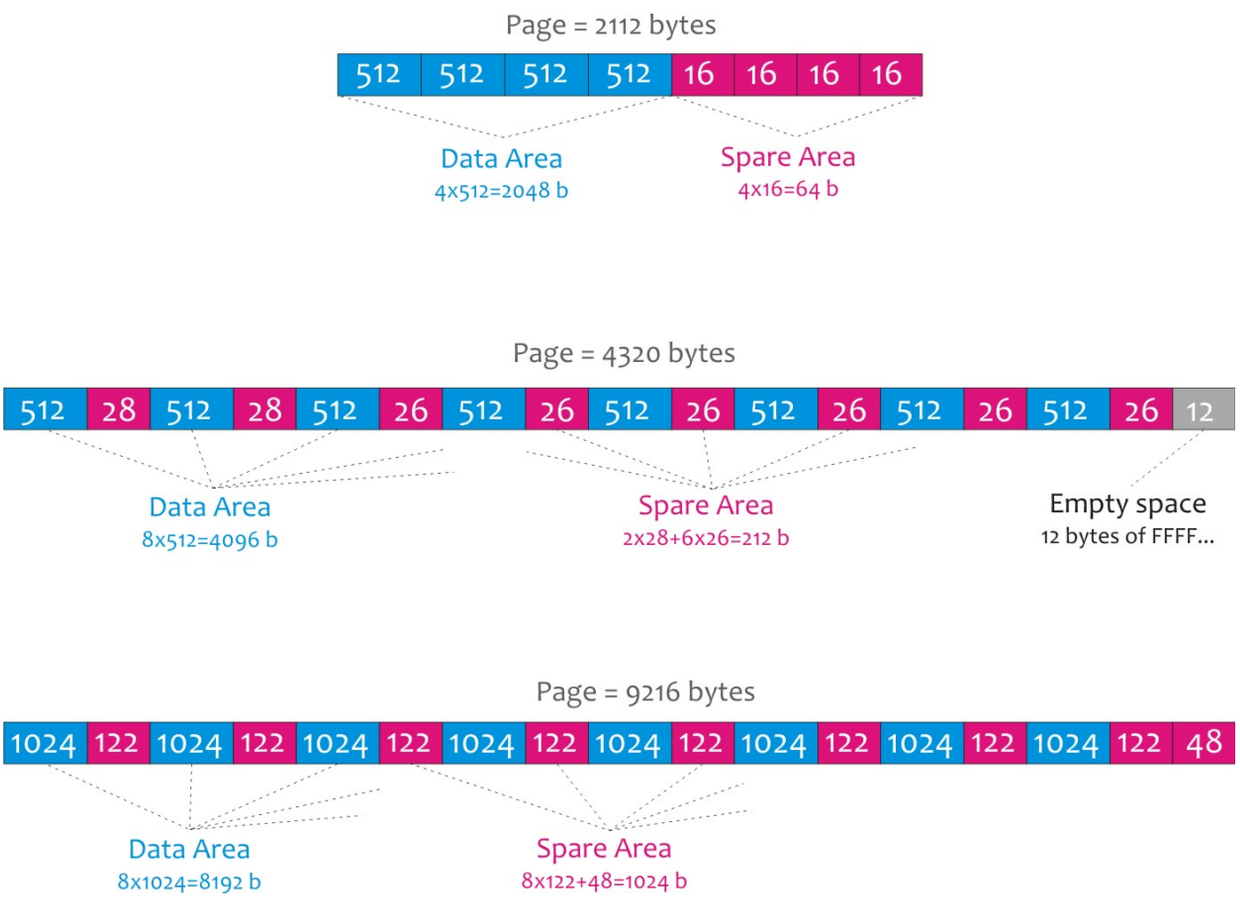
Visual Nand Reconstructor has a special mode for analysis of erased and obsolete blocks.

Page structure. Data & Spare areas

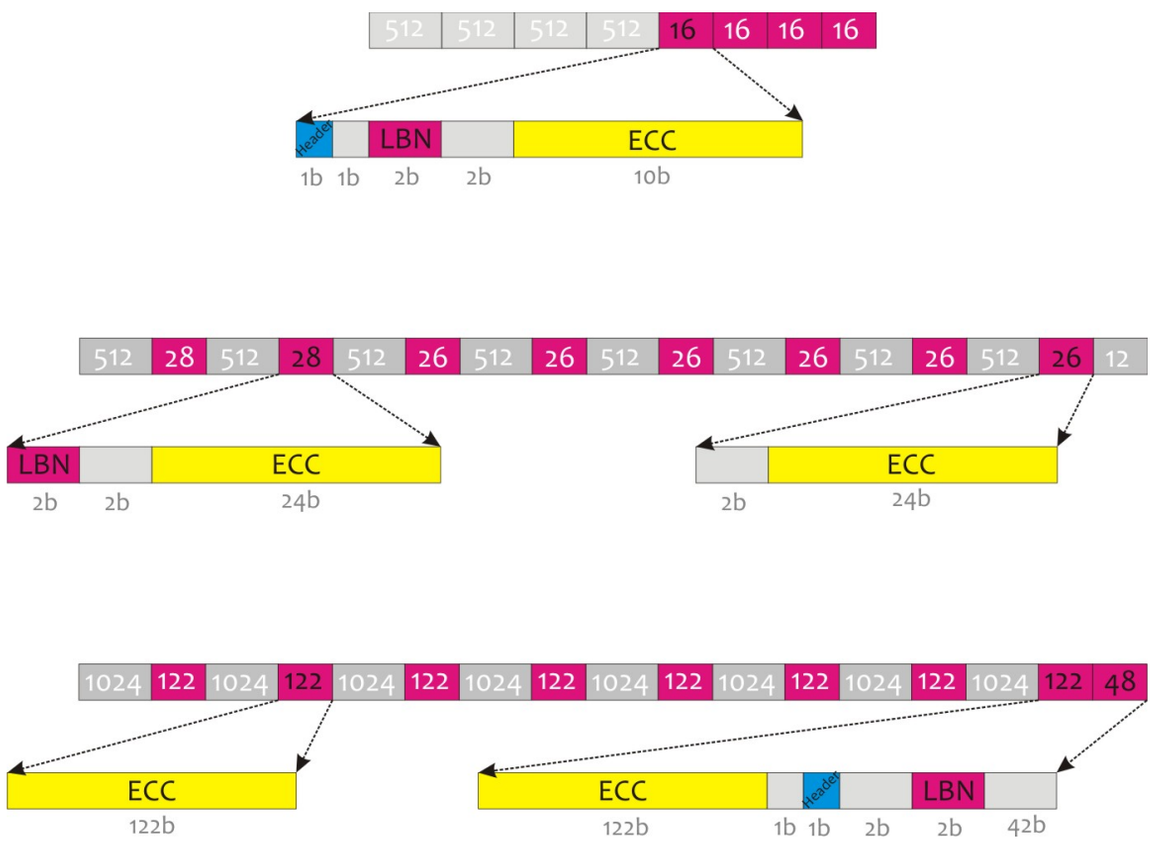
The blocks consist of pages. The page consists of 2 areas – Data Area and Spare Area. In Data Area the logical sectors with user data are recorded. Sector grouping method can be different, with SA alternation or successively located SA at the end of page. It depends on the firmware of controller. In the modern storage devices the way of grouping sectors with Spare Area alternation is used.

Different models of controllers use different internal structure of page. Page structure determination is one of the most important goals of the NAND's physical image analysis. This analysis comes to Data Area and Spare Area dedication and to determination of its sizes and offsets, relatively to beginning of the page (relatively to zero offset). All pages of one device have same structure. Data area is always multiple to 512b, the logical sector size inside the page can be 512, 1024, 2048 and more bytes. In most of modern controllers logical sector size 1024b is used (however file system works by 512b sector). It means that 2 logical sectors with data are recorded sequentially one by one. The Spare Area size can be different, but it's total size is no more than difference of Page Size and total Data Area size. It describes by formula [SA total = Page Size - Data Area]. In some cases Spare Area is not fully used and as a result empty areas, fulfilled by FF can appear in the end of the page.

The picture below represents examples of possible page structure



Internal structure of Spare Area also depends on controller's model. In general, the Spare area structure among all controllers may contain:
- LBN (Logical Block Number)
- LPN (Logical Page Number)
- Bank Number
- ECC (Error Correction Code)
- Block Header
- Block Write Counter



LBN (Logical Block Number)determines the belonging of virtual block and it's pages to the logical block of file system. In most cases 2 byte are given to LBN. Sorting virtual blocks in ascending order by LBN it's possible to reconstruct logical image. Usually block numeration begins from heximal 0000 and ends with maximum FFFF (or 01FF, 03FF, 07FF, 09FF, 0FFF – it depends on bank size), but sometimes high bits can be occupied by extra service information (1000...13FF).

LPN (Logical Page Number) determines the virtual page position inside the virtual block. More than half of controllers don't use LPN marker in spare area and record pages into the block sequentially (linear page translation). In case of non-linear translation pages are mixed up. In the moment of logical image reconstruction and data recovery, LPN marker allows to arrange them in correct ascending order.

Bank number determines to what bank one or another virtual block belongs. Typical bank can consist of 1024, 2048 or more blocks. Segmentation by banks shorten block addressing, because the LBN numeration in each bank begins with zero to maximum but is not prolonged. Some controllers give 1 byte in Spare Area to store Bank Number, but most manufacturers don't mark Bank Number in Spare Area. In the case presence of several blocks with the same LBN, belonging to some bank is determined by physical block address. In the end of each bank the reserve zone can be, where empty physical blocks for bad block reallocation are located.

ECC (Error Connecting Code) is used in all controllers, independently of manufacturer and use. When the logical sector (512, 1024, 2048 bytes) is going to be recorded in physical page, controller calculates the check sum of sector in accordance with a certain algorithm (mostly BCH algorithm) and records it in the SA. During the next page reading controller check and corrects the data in case of the error. The ECC code area size is directly proportional to the number of errors that the controller can correct and usually takes 3-15% of the logical sector size and 70-95% of the Spare Area size. Visual Nand Reconstructor has automatic ECC decoder for correcting reading errors, using ECC in Spare Area.

Block Header is one-byte marker that describes block purpose. There are several types of blocks:

- Main blocks (block contains user data)
- Replacement blocks (blocks with the updated user's data in cases when the old Main Block isn't overwritten)
- Log Blocks (blocks with the page updates, in case when the old data isn't overwritten)
- Bad Blocks (blocks which were marked on the factory as bad or worn during recording)
- Translation Table Blocks (blocks with the translation tables of controller)
- Reserved Blocks (blocks are assigned for bad blocks reallocation, usually empty FFFF)
- Firmware Blocks (blocks that contain Firmware of controller)
- Other system blocks (blocks that don't have any useful information or their purpose isn't deeply discovered yet)

Unused and reserved blocks usually do not contain any data and therefore not marked. Every controller manufacturer uses his own block header format. Its recognition is possible with the help of Visual Nand Reconstuctor instruments.

Block Write counter describes number of block write cycles.

Most controllers don't use all of the above Spare Area structures at the same time. Statistically average controller stores such structures as LBN, Block Header, ECC (sometimes LPN) in Spare Area. To analyze all described structures, the Bitmap mode of data visualization can be used.

Virtual Block allocation

Modern flash storage devices have high R/W speed and large capacity. It is achieved by multi-plane R/W operations within crystal or connecting two and more NAND chips to controller in parallel scheme (parallel page reading/recording from several chips/crystals/blocks, similar to RAID0). When portion of data sent by the host system, controller's buffer shapes the virtual block and then allocate it page by page between NAND chips/crystals/two neighboring physical blocks of crystal (multi-plane operation). The virtual block size depends on NAND chip's physical block size and the virtual block allocation scheme (data transfer channels).

There are several schemes of Virtual Block allocation:

- sequential allocation (virtual block equals to physical block of NAND chip and all memory chips/crystals/planes used sequentially by controller)

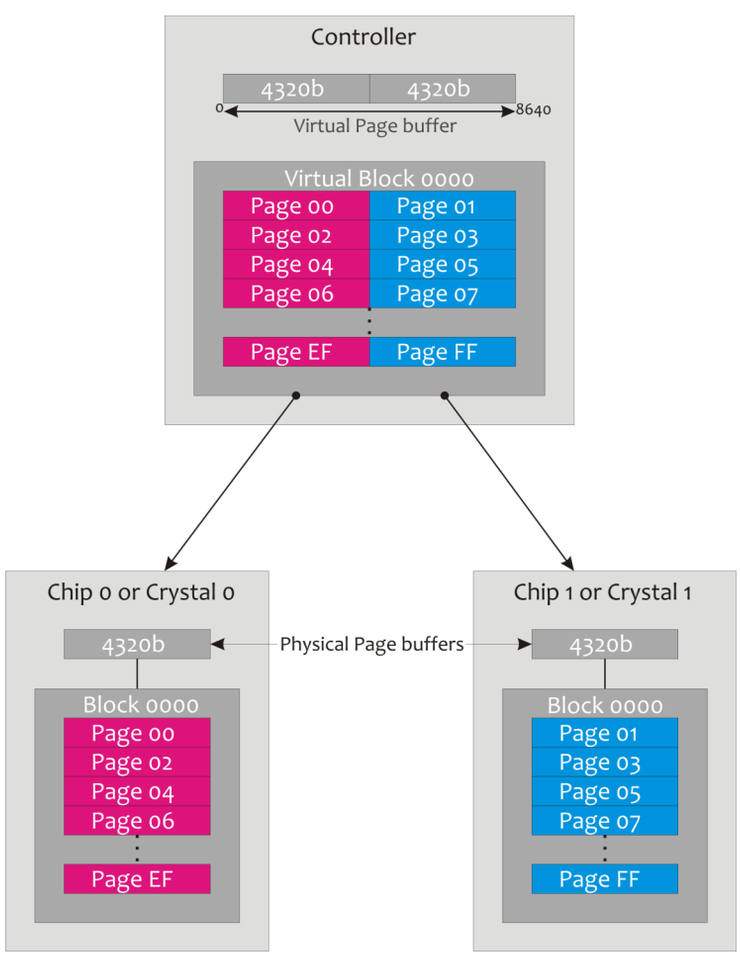
- parallel allocation (parallel data streams between NAND chips/crystals/two physical blocks of different planes)

- combined allocation (virtual block allocated in parallel between two physical blocks of crystal and sequentially between NAND chips/crystals)

If the Sequential allocation is used, the virtual block, which is formed in controller's buffer, equals to one physical block of NAND chip. In the certain moment of time controller works with one physical block of one chip/crystal, recording data page by page. Sequence of logical pages inside the block is continued. When free blocks are ended in chip/crystal, the controller switch to second chip/crystal. This scheme works same way as JBOD array of HDDs. At the end of analysis process all physical images of NAND chips must be united sequentially.

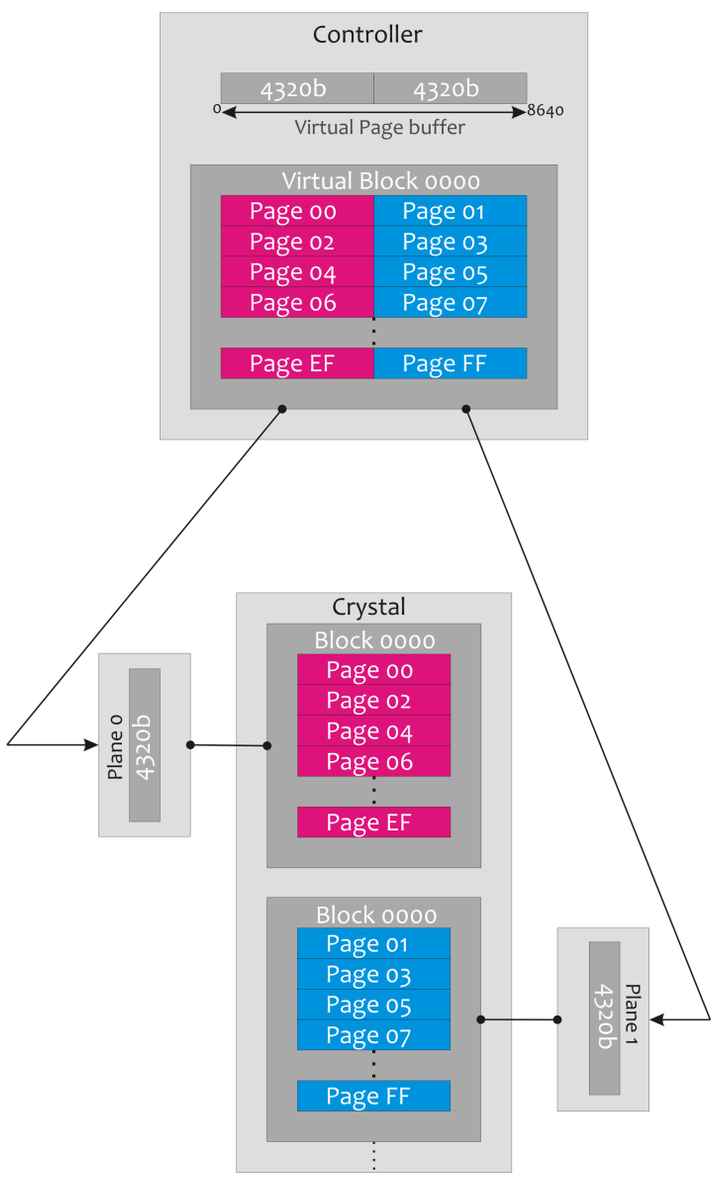
If the Parallel allocation is used, the virtual block equals to several physical blocks (2,4,8). The pages from buffer are recorded synchronously into the NAND chips/crystals/planes. The virtual block size depends on the number of chips/crystals/planes which take a part in parallel allocation, and physical block size of NAND chip.

The picture below presents parallel virtual block allocation between two NAND chips or two crystals of one chip.



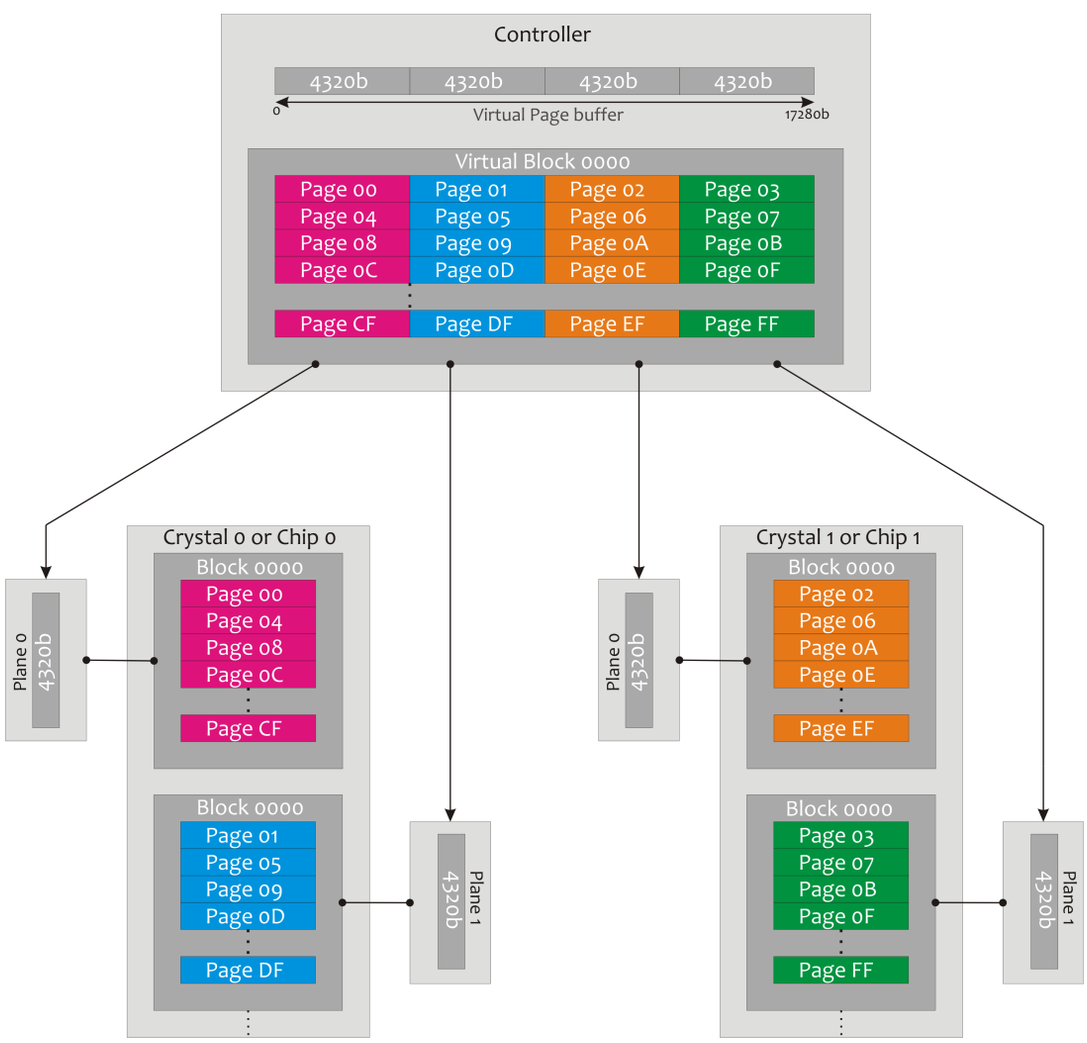
In case if controller used multi-plane R/W operations, the virtual block equals to 2 physical blocks of NAND. Controller recorded two pages at the same time into two neighbour blocks, that belong to different planes of one crystal. This is possible because each plane is independent memory cell array with its own page buffer. There's a special Pair operation in VNR, that rearrange pages sequentially and remove multi-plane allocation influence.

The picture below represents parallel virtual block allocation between two blocks of one crystal (multi-plane opreations).



If the Combined allocation is used, different serial and parallel allocation combinations are possible, depending on the number of NAND memory chips and crystals in each chip. E.g. parallel allocation between blocks of one crystal and sequential between crystals/chips, or all together in parallel. The virtual block size depends on the number of chips/crystals, and physical block size of flash memory chip.

The picture below represents combined virtual block allocation, parallel between two blocks of one crystal (multi-plane opreations) and parallel between crystals/NAND chips.



The virtual block allocation scheme determination is one of the most important steps in logical image reconstruction, because it's necessary to reconstruct structure of virtual block, joining physical images together. To analyze virtual Block allocation scheme we can use file system structures, such as FAT tables, FAT folders, NTFS File records and others. We can use LBN markers, because they are identical for every page of one virtual block. It means that presence of the same LBN in several crystals/chips by same physical address tells that parallel allocation scheme of virtual block would be used. When NAND chips/crystals have different LBN markers at the same physical address, a sequential allocation scheme would be used.

Data transformations. Inversion and Scrambling (XOR)

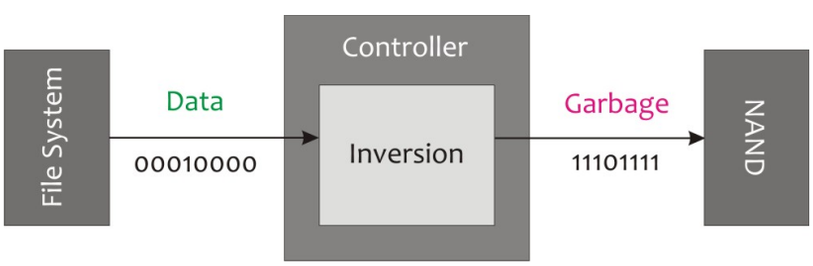
Most of modern controllers transform the data before recording it into the NAND memory chip. This is done to minimize the memory cell wear, due to specific disadvantages of high density flash chips, such as TLC chips. The typical operations are Inversion and Scrambling (XOR).

Inversion is used by some controllers to minimize write cycles, and by that to reduce wear of memory cells. This operation is performed on the bit level. It is a mathematical operation "NOT".

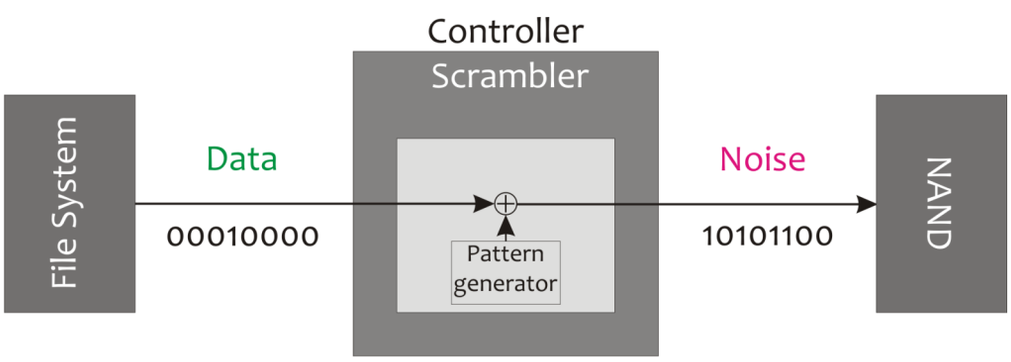
Invert 0 = 1
Invert 1 = 0

The user data that coming to controller's buffer is inverted and then recorded into NAND memory chip. During the recovery process it's necessary to return back the original shape of data, using the same action, that controller used.

Inversion [Inverted (Data)] = Data



Scrambling (XOR) converts the user data to binary noise. This transformation is applied almost in all modern controllers. This is because modern TLC memory chips are exposed to data storage degradation and instability, when they are recorded with specific user data which have patterns. To prevent it, the controller merges user data with special binary sequence (XOR key), which converts any data to noise. The scrambling pattern (XOR key) isn't stored inside of the controller, it is generated while writing/reading on-fly. In case of controller damage, the key is lost and data inside the chip remain scrambled (noise). The scrambling pattern isn't unique, all controllers of one model use the same one. To convert noise to data it's necessary to use one of the XORkeys from the Visual Nand Reconstruction resources. In many cases it's possible to extract key directly from physical image of NAND chip with user's data. The key structure also depends on page structure, because in most cases only Data Area of page is scrambled.



The Visual Nand Reconstructor has special tools for reverse engineering of scramblers of new controllers.

Next part is coming soon

ruSolut
Data Investigation & Recovery Technology

Contact Us

Rusolut Sp. z o.o.

Address: 49 Kasprzaka st., Warsaw, Poland, 01-249

Phone: +48531999777
Email: info@rusolut.com
Web: www.rusolut.com

Useful links

Case samples
http://rusolut.com/case-samples/

Latest technology in our blog:
http://rusolut.com/blog/

F.A.Q.
http://rusolut.com/f-a-q/