ruSolut
Data Investigation & Recovery Technology

VISUAL NAND RECONSTRUCTOR

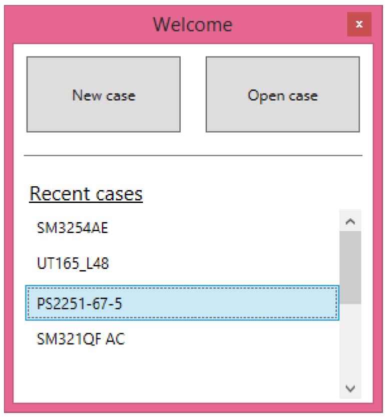
The book

Part2. Software

www.rusolut.com

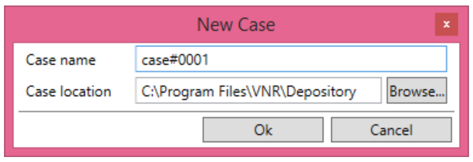
VNR software concept

The Visual Nand Reconstructor software uses a case management system. Each case is stored in a separate folder with physical images. There are three options at software start up - create new case, open case and open recent case.



SM3254AE
UT165_L48
PS2251-67-5
SM321QF AC

When creating a new case it's necessary to set the path to depository and the case number/name. The subfolder with case name will be created in depository.

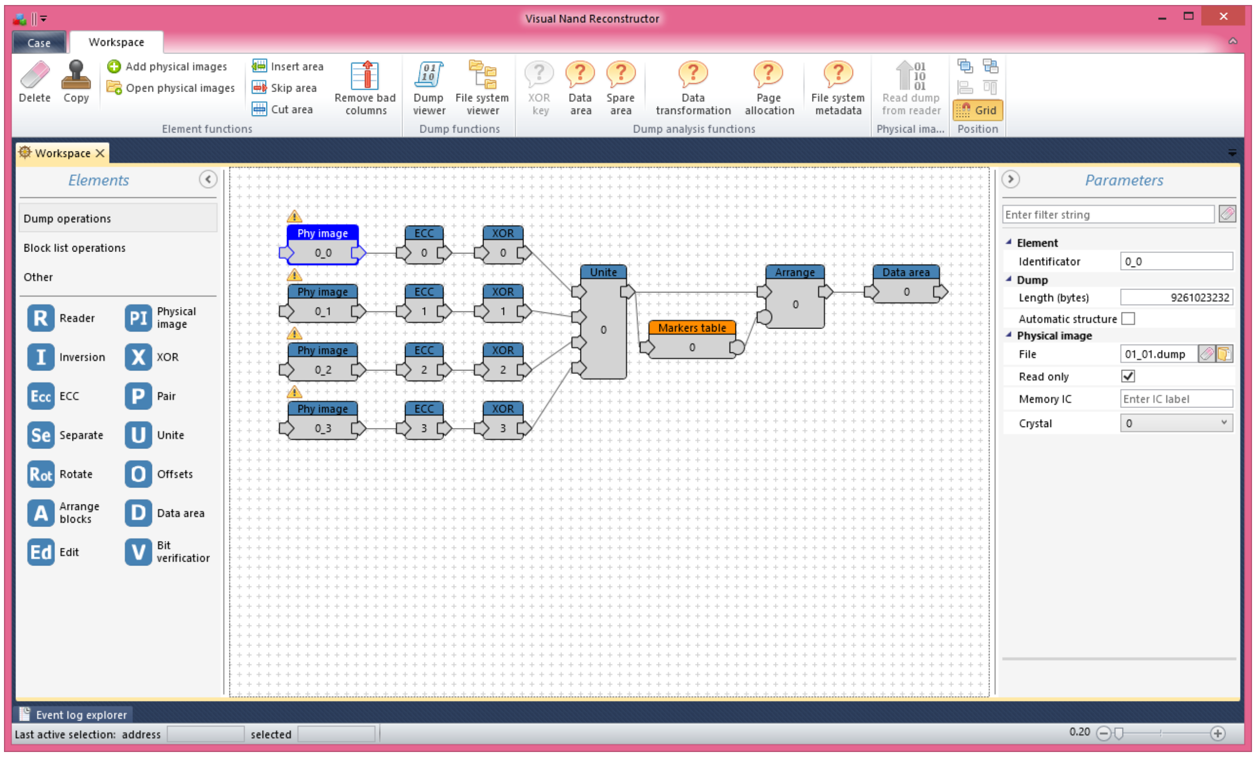


Case name: case#0001
Case location: C:\Program Files\VNR\Depository

The main window of Visual Nand Recontructor is divided into several zones:

Elements (left)
Workspace (center)
Parameters of element (right)
Toolbar (top)
Event log (bottom)

ruSolut
Data Investigation & Recovery Technology



Element
Identificator: 0_0
Dump
Length (bytes): 9261023232
Automatic structure: ☐
Physical image
File: 01_01.dump
Read only: ✓
Memory IC: Enter IC label
Crystal: 0

Elements represent the operations for transformation of the physical image of Nand memory chips. Elements are added to the workspace by Drag&Drop method. The set of elements is divided into several types: Dump operations, Block list operations, other elements.

Dump operations contain Reader and Physical image elements, also operations for physical image transformation: ECC, Inversion, XOR, Pair, Separate, Unite, Offsets, Arrange blocks, Data area, etc. Each element has a definite function, which allows transformation of the physical image in accordance with the controller's configuration.

Block list operations contain operations for work with the physical/virtual image on the blocks level. Markers table element is designed for image reconstruction from virtual to logical (translator).

Workspace is a work area, where Elements are added and also parameters and connections between them are set. Almost every element, except the Reader, has an input and output. Element connection is organized the way when each element virtually transforms the source element that connected to it's input. Some elements have two or more inputs, it is necessary for unite of virtual images (crystals or memory chips). All elements have one output, however it's possible to share it between several elements, for example for various hypothesis checking within same case. This ideology is close to the electronic circuit simulation, when every element emulates controller's electronic block.
Conversion from physical image to virtual and then to logical takes place in this workspace. When an element is added, it is necessary to connect it with a previous element, otherwise it won't have data source (except the Reader element). Connections between elements can be deleted and recreated, meanwhile other elements and their parameters stay untouched. To create a connection between elements it's necessary to click on output of source, then to input of the connecting element.

Parameters of element can be set to determine how the image will be transformed. Some of them depend on the number of chips and their physical parameters (page size, block size), some on controller's model (XOR key, ECC). Parameter set is adaptive and depends on the element. All parameters are divided into groups. All parameters are measured in bytes and can be entered in decimal or hexadecimal format and saved automatically while you fullfill appropriate fields.

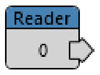
Toolbar includes automatic and manual physical image analysis modes and other functions. This tab is adaptive. Mode availability depends on the active element. Dump viewer tab contains a number of special modes for image browse, such as: Hex Viewer, Bitmap Viewer, Structure Viewer, Record Viewer. Modes can be used simultaneously and synchronously for unlimited number of physical images.

Events and errors which appear in work process are displayed in Event Log.

ruSolut
Data Investigation & Recovery Technology

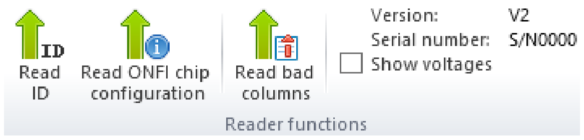
Elements

Elements are divided into Dump Operations and Block operations. The Reader and Physical image elements are containers of physical image (direct NAND access and dump file). All other Dump Operation elements virtually modify physical image image in accordance with parameters. Markers Table element sets parameters of translation of physical/virtual blocks into logical image.



The Reader element is a program module of NAND Reader. It is designed for realtime NAND memory access and physical image extraction. When the new case created the Reader is added automatically.

The set of functions for reader is available on the adaptive Toolbar.



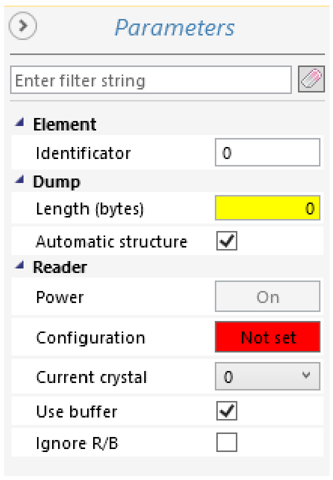
Version: V2
Serial number: S/N0000
☐ Show voltages

Read ID allows to get identifier of NAND chip.

Read ONFI chip configuration allows to read memory chip's parameters from special page, in case if NAND chip conforms to ONFI specification. This works well for Micron (0x2C) and Intel (0x89) chips.

Read bad columns allows to read defective columns (bytes) which were programmed in NAND chip at factory. This option is currently supported for some Sandisk and Toshiba NAND chips.

The Parameters tab contains settings of NAND chip access.


Parameters

Enter filter string

Element	
Identificator	0
Dump	
Length (bytes)	0
Automatic structure	✓
Reader	
Power	On
Configuration	Not set
Current crystal	0
Use buffer	✓
Ignore R/B	☐

NAND chip's access parameters must be set in Configuration from built-in database

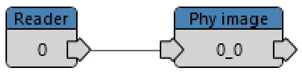
Current crystal represents the active crystal (CE) of NAND from which the dump will be read/accessed.

Power ON/OFF turns the power of NAND chip ON/OFF. Pressing the button performs the action displayed on it (not status!). When power is ON, the yellow LED of reader is alight.

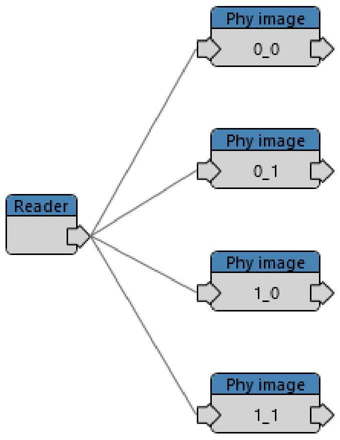
Ignore R/B function is used when analyzing new NAND chip configurations. It helps to avoid reader's hang up. However, it's not recommended to use it normally.

Detailed instruction how to use Reader element to access NAND and extract physical images can be found in the web article:
http://rusolut.com/direct-access-to-nand-and-physical-image-extraction/

The Physical Image element contains a binary copy of the NAND chip. When extracting physical image (dump reading), the RAW NAND data is recorded into a dump file on the disk.



Every crystal of NAND chip is represented by a Physical Image element when extracting dump. For example, in case of two NAND chips and two crystals per each, it is necessary to add 4 Physical Image elements.

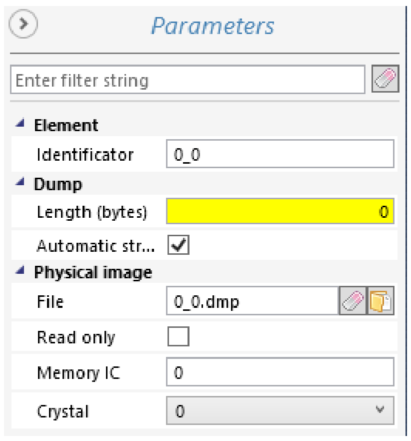


Physical Image element has input and output. The input is used for reader connection. The data goes out of reader to the Physical Image element and saved to dump file. On the output Physical Image connects with other elements for further transformation of physical image to virtual and logical.

The set of functions for Physical image element is available on the adaptive Toolbar.

Read dump operation starts physical image extraction process.

The Parameters tab contains settings of Physical image element.



Parameters

Enter filter string

Element

Identificator	0_0

Dump

Length (bytes)	0
Automatic str...	✓

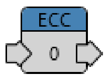
Physical image

File	0_0.dmp
Read only	☐
Memory IC	0
Crystal	0

File contains the path to file with physical image (dump file).

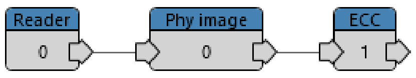
Read only is set to protect dump files from modification through hex viewer.

Crystal represents the crystal (CE) of NAND this pysical image belongs to.



The ECC element (Error Correction Code) allows to correct bit errors which appear during data recording/reading process in flash memory and also while physical image extraction. Uncorrected bit errors damage the user's data and corrupt files. The ECC decoder corrects errors on-fly, using code stored in pages of NAND chip.

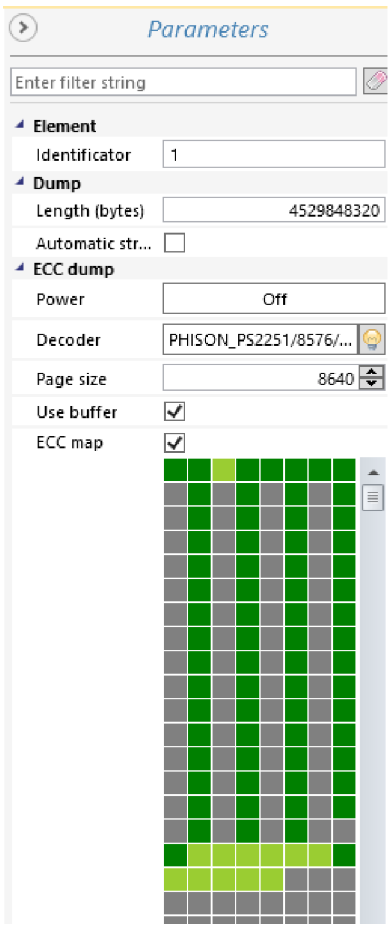
ECC element must be connected to the Physical Image, or after elimination of Bad Columns if they exist.



The set of functions for ECC element is available on the adaptive

Reread dump allows to re-read pages of the NAND chip, which can not be corrected due to too high level of errors. If the chip was read with the standard voltage 3.3V and the capability of ECC code is not enough for data correction, it's possible to reduce the voltage to 2.5V ...1.8V in order to reduce level of internal noise of chip. Having lowered the noise, this function does read only pages that were not corrected at previous read attempt. Multiple reread iterations are supported, if image was not ideally reread after first attempt. To use this option it's necessary to remove read only flag from physical image element. Reader must be connected to the Physical image elements.

The Parameters tab contains settings of ECC element.



Element	
Identificator	1
Dump	
Length (bytes)	4529848320
Automatic str...	☐
ECC dump	
Power	Off
Decoder	PHISON_PS2251/8576/...
Page size	8640
Use buffer	✓
ECC map	✓

Power ON/OFF turns ON/OFF bit error correction. Pressing the button performs the action displayed on it (not status!).

Decoder defines ECC code format used for particular controller. Different controllers use different code types. Decoder can be selected manually or automatically, by pressing on button

Page size must be adjusted to the NAND's page size.

ECC map allows to estimate the quality of data correction. One square represents one page.

Dark green = good page (no errors and no correction required)

Light green = corrected page (all errors corrected)

Red = bad page (too much errors, correction doesn't work due to code's power limit)

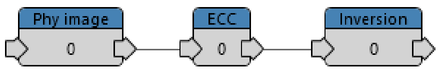
Grey = empty page (filled with FFFF)

Note: pressing on ECC map flag does not enable correction! In order to enable dump correction turn power ON.

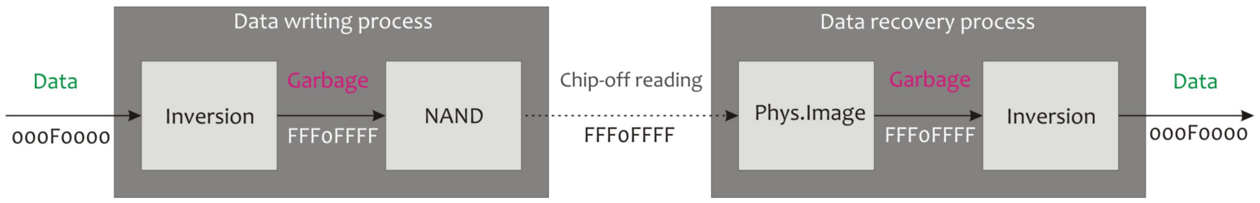
Inversion element performs the binary operation – NOT. Inversion converts the physical image in accordance with a simple binary rule:

Not 0 = 1
Not 1 = 0

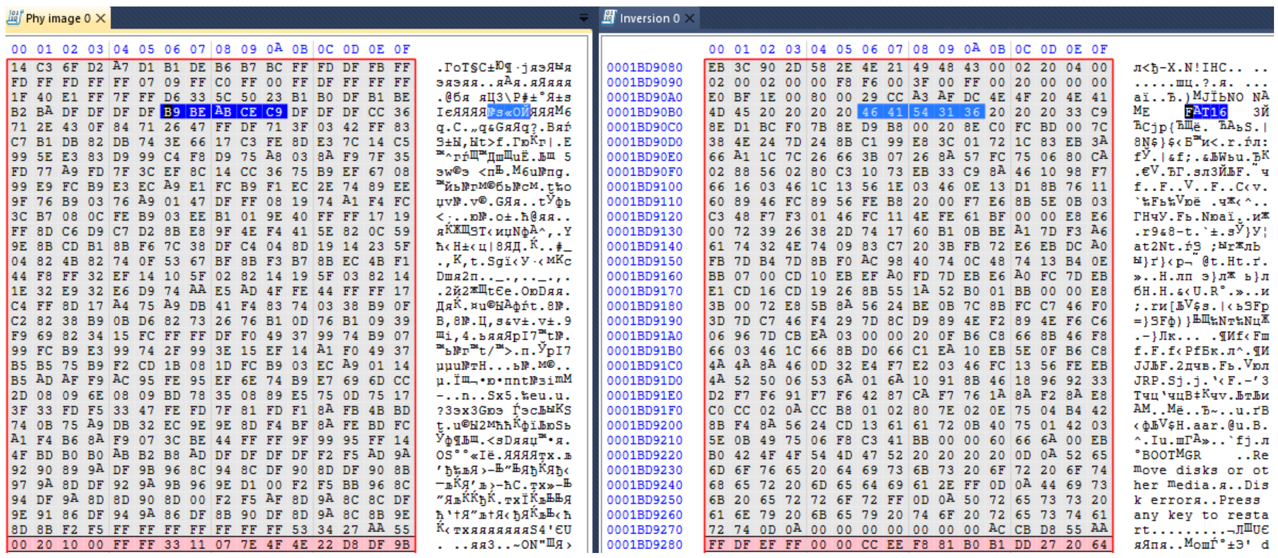
Some controllers invert data before recording into flash memory, to minimize the wear of memory cells. To convert inverted physical image to normal state, it's necessary to apply inversion.



The functional scheme of Inversion in working flash storage device and reverse process is represented below



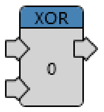
In hexadecimal format it looks like on the picture below. The inverted data is on the left, the original data is on the right.



```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
14 C3 6F D2 A7 D1 B1 DE B6 B7 BC FF FD DF FB FF   .ГоТ§С±Ю¶·jяэЯыя
FD FF FD FF FF 07 09 FF C0 FF 00 FF DF FF FF FF   эяэяя..яАя.яЯяяя
1F 40 E1 FF 7F FF D6 33 5C 50 23 B1 B0 DF B1 BE   .@бя яЦ3\P#±°Я±ѕ
B2 BA DF DF DF DF B9 BE AB CE C9 DF DF DF CC 36   IєЯЯЯЯ№ѕ«ОЙЯЯЯМ6
71 2E 43 0F 84 71 26 47 FF DF 71 3F 03 42 FF 83   q.C.„q&GяЯq?.Bяѓ
C7 B1 DB 82 DB 74 3E 66 17 C3 FE 8D E3 7C 14 C5   Ѕ±Ы,Ыt>f.ГюЌг|.E
99 5E E3 83 D9 99 C4 F8 D9 75 A8 03 8A F9 7F 35   ™^гѓЩ™ДшЩuЁ.Љщ 5
FD 77 A9 FD 7F 3C EF 8C 14 CC 36 75 B9 EF 67 08   эw©э <пЊ.Мбu№пg.
99 E9 FC B9 E3 EC A9 E1 FC B9 F1 EC 2E 74 89 EE   ™йь№гМ©бь№сМ.ţ‰o
9F 76 B9 03 76 A9 01 47 DF FF 08 19 74 A1 F4 FC   µv№.v©.GЯя..tЎфь
3C B7 08 0C FE B9 03 EE B1 01 9E 40 FF FF 17 19   <·..ю№.o±.ћ@яя..
FF 8D C6 D9 C7 D2 8B E8 9F 4E F4 41 5E 82 0C 59   яЌЖЩЗТ‹иµNфА^,.Y
9E 8B CD B1 8B F6 7C 38 DF C4 04 8D 19 14 23 5F   ћ‹Н±‹ц|8ЯД.Ќ..#_
04 82 4B 82 74 0F 53 67 BF 8B F3 B7 8B EC 4B F1   .,К,t.Sgï‹У·‹мКс
44 F8 FF 32 EF 14 10 5F 02 82 14 19 5F 03 82 14   Dшя2п.._.,.._.,.
1E 32 E9 32 E6 D9 74 AA E5 AD 4F FE 44 FF FF 17   .2й2жЩtЄe.OюDяя.
C4 FF 8D 17 A4 75 A9 DB 41 F4 83 74 03 38 B9 0F   ДяЌ.¤u©ЫАфѓt.8№.
C2 82 38 B9 0B D6 82 73 26 76 B1 0D 76 B1 09 39   B,8№.Ц,ѕ&v±.v±.9
F9 69 82 34 15 FC FF FF DF F0 49 37 99 74 B9 07   щi,4.ьяяЯpI7™t№.
99 FC B9 E3 99 74 2F 99 3E 15 EF 14 A1 F0 49 37   ™ь№г™t/™>.п.ЎpI7
B5 B5 75 B9 F2 CD 1B 08 1D FC B9 03 EC A9 01 14   µµu№тН...ь№.М©..
B5 AD AF F9 AC 95 FE 95 EF 6E 74 B9 E7 69 6D CC   µ.Їш¬•ю•ппt№ѕimМ
2D 08 09 6E 08 09 BD 78 35 08 89 E5 75 0D 75 17   -..n..Sx5.‰eu.u.
3F 33 FD F5 33 47 FE FD 7F 81 FD F1 8A FB 4B BD   ?3эx3Gюэ ЃэсЉыКS
74 0B 75 A9 DB 32 EC 9E 9E 8D F4 BF 8A FE BD FC   ţ.u©Ы2МћћЌфїЉюSь
A1 F4 B6 8A F9 07 3C BE 44 FF FF 9F 99 95 FF 14   Ўф¶Љщ.<ѕDяяµ™•я.
4F BD B0 B0 AB B2 B8 AD DF DF DF DF F2 F5 AD 9A   OS°°«Ië.ЯЯЯтx.љ
92 90 89 9A DF 9B 96 8C 94 8C DF 90 8D DF 90 8B   'ђ‰љЯ›–Њ”ЊЯђЌЯђ‹
97 9A 8D DF 92 9A 9B 96 9E D1 00 F2 F5 BB 96 8C   —љЌЯ'љ›–ћС.тх»–Њ
94 DF 9A 8D 8D 90 8D 00 F2 F5 AF 8D 9A 8C 8C DF   ”ЯљЌЌђЌ.тхЇЌљЊЊЯ
9E 91 86 DF 94 9A 86 DF 8B 90 DF 8D 9A 8C 8B 9E   ћ'†Я”љ†Я‹ђЯЌљЊ‹ћ
8D 8B F2 F5 FF FF FF FF FF FF FF 53 34 27 AA 55   Ќ‹тхяяяяяяяS4'ЄU
00 20 10 00 FF FF 33 11 07 7E 4F 4E 22 D8 DF 9B   . ..яя3..~ON"ШЯ›
```

```
          00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0001BD9080 EB 3C 90 2D 58 2E 4E 21 49 48 43 00 02 20 04 00   л<ђ-X.N!IHC.. ..
0001BD9090 02 00 02 00 00 F8 F6 00 3F 00 FF 00 20 00 00 00   .....шц.?.я. ...
0001BD90A0 E0 BF 1E 00 80 00 29 CC A3 AF DC 4E 4F 20 4E 41   aї..Ђ.)МЈЇЬNO NA
0001BD90B0 4D 45 20 20 20 20 46 41 54 31 36 20 20 20 33 C9   МЕ    FAT16   3Й
0001BD90C0 8E D1 BC F0 7B 8E D9 B8 00 20 8E C0 FC BD 00 7C   ЋСјр{ЋЩё. ЋАьS.|
0001BD90D0 38 4E 24 7D 24 8B C1 99 E8 3C 01 72 1C 83 EB 3A   8Ņ$}$‹Б™и<.r.ѓл:
0001BD90E0 66 A1 1C 7C 26 66 3B 07 26 8A 57 FC 75 06 80 CA   fЎ.|&f;.&ЉWьu.ЂК
0001BD90F0 02 88 56 02 80 C3 10 73 EB 33 C9 8A 46 10 98 F7   .€V.ЂГ.ѕл3ЙЉF.˜ч
0001BD9100 66 16 03 46 1C 13 56 1E 03 46 0E 13 D1 8B 76 11   f..F..V..F..С‹v.
0001BD9110 60 89 46 FC 89 56 FE B8 20 00 F7 E6 8B 5E 0B 03   `‰Fь‰Vюё .чЖ‹^..
0001BD9120 C3 48 F7 F3 01 46 FC 11 4E FE 61 BF 00 00 E8 E6   ГНчУ.Fь.Nюаї..иЖ
0001BD9130 00 72 39 26 38 2D 74 17 60 B1 0B BE A1 7D F3 A6   .r9&8-t.`±.ѕЎ}У¦
0001BD9140 61 74 32 4E 74 09 83 C7 20 3B FB 72 E6 EB DC A0   at2Nt.ѓЗ ;ыгЖлЬ
0001BD9150 FB 7D B4 7D 8B F0 AC 98 40 74 0C 48 74 13 B4 0E   ы}ґ}‹р¬˜@t.Ht.ґ.
0001BD9160 BB 07 00 CD 10 EB EF A0 FD 7D EB E6 A0 FC 7D EB   »..Н.лп э}лЖ ь}л
0001BD9170 E1 CD 16 CD 19 26 8B 55 1A 52 B0 01 BB 00 00 E8   бН.Н.&‹U.R°.»..и
0001BD9180 3B 00 72 E8 5B 8A 56 24 BE 0B 7C 8B FC C7 46 F0   ;.rи[ЉV$ѕ.|‹ьЗFр
0001BD9190 3D 7D C7 46 F4 29 7D 8C D9 89 4E F2 89 4E F6 C6   =}ЗFф)}ЊЩ‰Nт‰NцЖ
0001BD91A0 06 96 7D CB EA 03 00 00 20 0F B6 C8 66 8B 46 F8   .–}Лк... .¶Иf‹Fш
0001BD91B0 66 03 46 1C 66 8B D0 66 C1 EA 10 EB 5E 0F B6 C8   f.F.f‹PfБк.л^.¶И
0001BD91C0 4A 4A 8A 46 0D 32 E4 F7 E2 03 46 FC 13 56 FE EB   JJЉF.2дчв.Fь.Vюл
0001BD91D0 4A 52 50 06 53 6A 01 6A 10 91 8B 46 18 96 92 33   JRP.Sj.j.'‹F.–'3
0001BD91E0 D2 F7 F6 91 F7 F6 42 87 CA F7 76 1A 8A F2 8A E8   Тчц'чцВ‡Кчv.ЉтЉи
0001BD91F0 C0 CC 02 0A CC B8 01 02 80 7E 02 0E 75 04 B4 42   АМ..Мё..Ђ~..u.ґВ
0001BD9200 8B F4 8A 56 24 CD 13 61 61 72 0B 40 75 01 42 03   ‹фЉV$Н.aar.@u.B.
0001BD9210 5E 0B 49 75 06 F8 C3 41 BB 00 00 60 66 6A 00 EB   ^.Iu.шГА»..`fj.л
0001BD9220 B0 42 4F 4F 54 4D 47 52 20 20 20 20 0D 0A 52 65   °BOOTMGR    ..Re
0001BD9230 6D 6F 76 65 20 64 69 73 6B 73 20 6F 72 20 6F 74   move disks or ot
0001BD9240 68 65 72 20 6D 65 64 69 61 2E FF 0D 0A 44 69 73   her media.я..Dis
0001BD9250 6B 20 65 72 72 6F 72 FF 0D 0A 50 72 65 73 73 20   k errorя..Press 
0001BD9260 61 6E 79 20 6B 65 79 20 74 6F 20 72 65 73 74 61   any key to resta
0001BD9270 72 74 0D 0A 00 00 00 00 00 00 00 AC CB D8 55 AA   rt.........¬ЛШUЄ
0001BD9280 FF DF EF FF 00 00 CC EE F8 81 B0 B1 DD 27 20 64   яЯпя..МошЃ°±Э' d
```

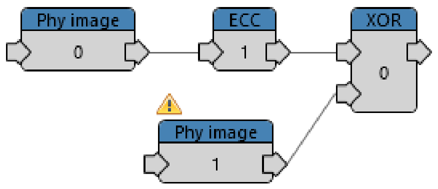
The Inversion element has no adjustable parameters.



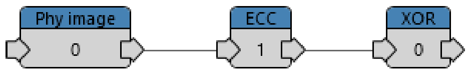
The XOR element decrypts the data that was scrambled (XOR'd) in controller's data transfer channel. This mathematical operation works according to the rule:

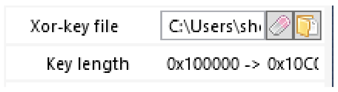
Data	XOR key	XOR'd data
0	0	0
0	1	1
1	0	1
1	1	0

The XOR element has 2 inputs and one output. The physical/virtual image is connected to upper input, the physical image element with XOR key loaded as file is connected to lower input. This method of connection of the XOR element is used for custom built XOR keys and new XOR key analysis and extraction.



When a XOR key is supported and available in VNR resources, it must be selected from Parameters tab of XOR element. Then the second lower input will automatically disappear.



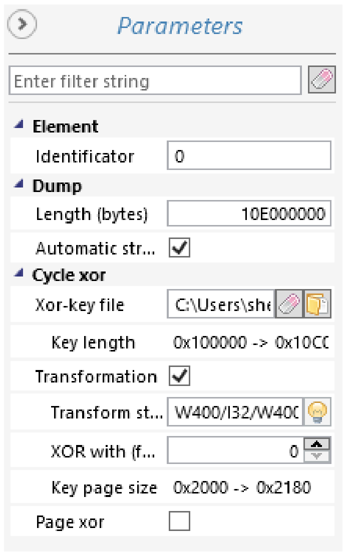


Xor-key file	C:\Users\sh
Key length	0x100000 -> 0x10C(

The Parameters tab contains settings of XOR element.

XOR-Key file contains the path to XOR key file.

Transformation flag is used for XOR key adjustment, according to page structure. There are 2 different XOR key formats - XOR Key for Data area only, and XOR Key for Data area and Spare area. Transformation flag is required for XOR Keys which applied to Data area only (this type is used in ~90% of all controllers). When XOR Key applied to Data area and Spare area, transformation flag must be unmarked.



Parameters

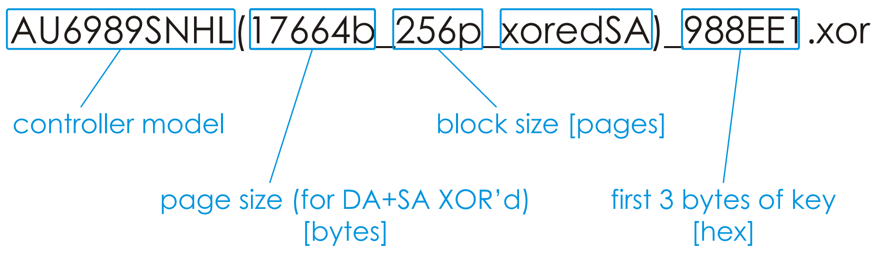
Enter filter string

Element	
Identificator	0
Dump	
Length (bytes)	10E000000
Automatic str...	✓
Cycle xor	
Xor-key file	C:\Users\she
Key length	0x100000 -> 0x10CC
Transformation	✓
Transform st...	W400/I32/W40C
XOR with (f...	0
Key page size	0x2000 -> 0x2180
Page xor	☐

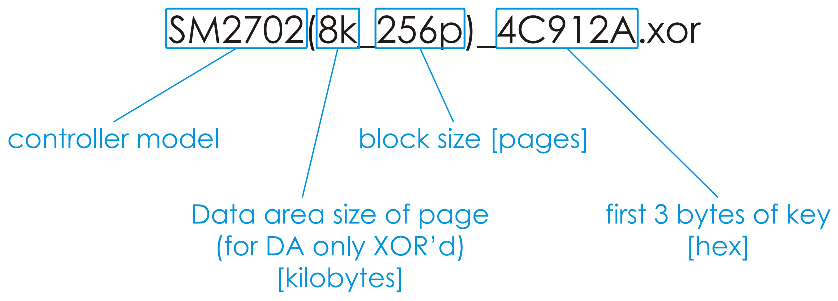
The button automatically adjusts the XOR key structure according the page structure (when XOR key for Data area only is used)

XOR Keys have two formats - for Data area only and for Data area with Spare area. All their parameters which should match to the given case are mentioned in XOR key file names.

XOR key for xor'd Data area and Spare area

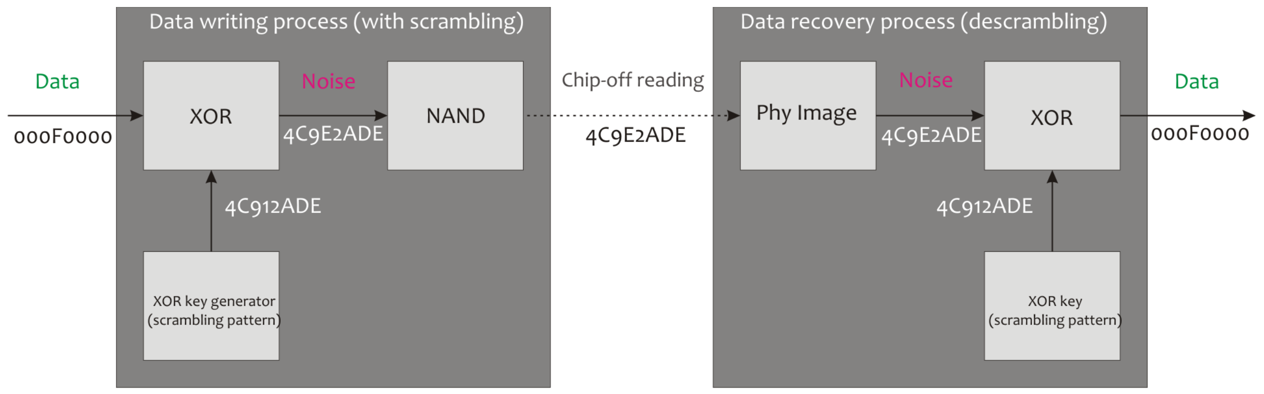

AU6989SNHL(17664b_256p_xoredSA)_988EE1.xor
controller model
block size [pages]
page size (for DA+SA XOR'd) [bytes]
first 3 bytes of key [hex]

XOR key for xor'd Data area only

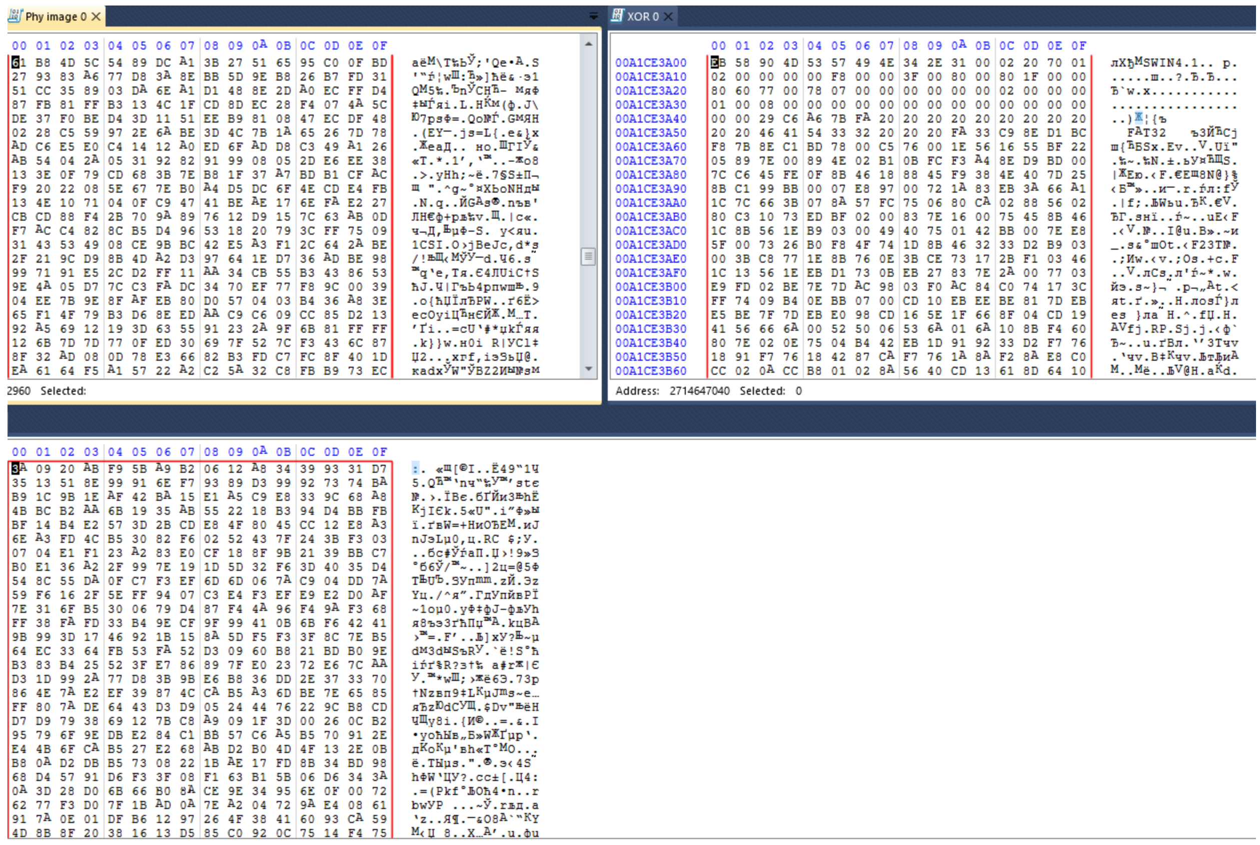

SM2702(8k_256p)_4C912A.xor
controller model
block size [pages]
Data area size of page (for DA only XOR'd) [kilobytes]
first 3 bytes of key [hex]

Every controller model may use one of 2-3 keys, depending on the page and block size of NAND chip

The functional scheme of XOR in working flash storage device and reverse process is represented below


Data writing process (with scrambling)
Data
000F0000
XOR
Noise
4C9E2ADE
NAND
4C912ADE
XOR key generator (scrambling pattern)
Chip-off reading
4C9E2ADE
Data recovery process (descrambling)
Phy Image
Noise
4C9E2ADE
XOR
Data
000F0000
4C912ADE
XOR key (scrambling pattern)

In hexadecimal format it looks like on picture below. The XOR'd data is on the left, the decrypted data is on the right, the XOR key is at the

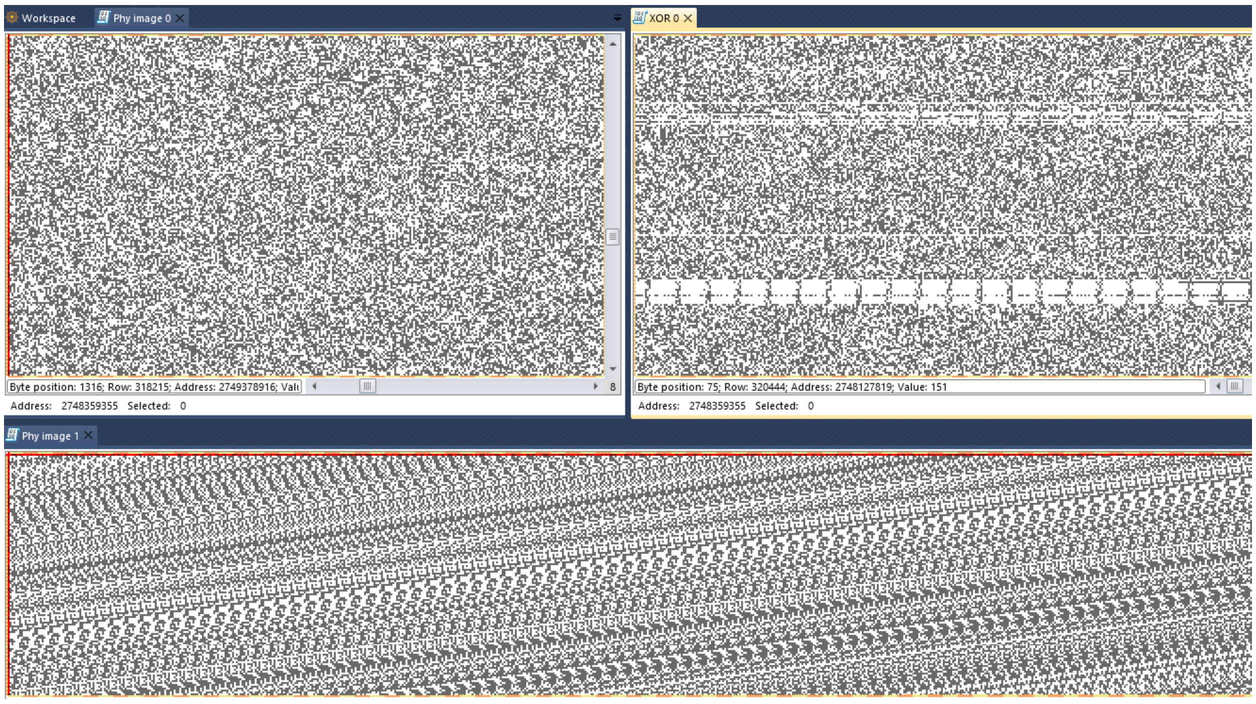


```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
61 B8 4D 5C 54 89 DC A1 3B 27 51 65 95 C0 0F BD   aëM\T‰ЬЎ;'Qe•A.S
27 93 83 A6 77 D8 3A 8E BB 5D 9E B8 26 B7 FD 31   '“ѓ¦wШ:Ћ»]ћё&·э1
51 CC 35 89 03 DA 6E A1 D1 48 8E 2D A0 EC FF D4   QM5‰.ЪnЎСНЋ- мяФ
87 FB 81 FF B3 13 4C 1F CD 8D EC 28 F4 07 4A 5C   ‡ыЃяi.L.НЌм(ф.J\
DE 37 F0 BE D4 3D 11 51 EE B9 81 08 47 EC DF 48   Ю7psФ=.Qo№Ѓ.GMЯH
02 28 C5 59 97 2E 6A BE 3D 4C 7B 1A 65 26 7D 78   .(EY—.js=L{.e&}x
AD C6 E5 E0 C4 14 12 A0 ED 6F AD D8 C3 49 A1 26   .ЖеаД.. но.ШГIЎ&
AB 54 04 2A 05 31 92 82 91 99 08 05 2D E6 EE 38   «T.*.1’‚‘™..-жо8
13 3E 0F 79 CD 68 3B 7E B8 1F 37 A7 BD B1 CF AC   .>.yHh;~ё.7§S±П¬
F9 20 22 08 5E 67 7E B0 A4 D5 DC 6F 4E CD E4 FB   щ ".^g~°¤XЬoNHды
13 4E 10 71 04 0F C9 47 41 BE AE 17 6E FA E2 27   .N.q..ЙGAs®.nъв'
CB CD 88 F4 2B 70 9A 89 76 12 D9 15 7C 63 AB 0D   ЛН€ф+pљ‰v.Щ.|c«.
F7 AC C4 82 8C B5 D4 96 53 18 20 79 3C FF 75 09   ч¬Д‚ЊµФ–S. y<яu.
31 43 53 49 08 CE 9B BC 42 E5 A3 F1 2C 64 2A BE   1CSI.O›jBeJc,d*s
2F 21 9C D9 8B 4D A2 D3 97 64 1E D7 36 AD BE 98   /!њЩ‹MўУ—d.Ч6.s˜
99 71 91 E5 2C D2 FF 11 AA 34 CB 55 B3 43 86 53   ™q‘e,Tя.Є4ЛUiC†S
9E 4A 05 D7 7C C3 FA DC 34 70 EF 77 F8 9C 00 39   ћJ.Ч|ГъЬ4pпwшњ.9
04 EE 7B 9E 8F AF EB 80 D0 57 04 03 B4 36 A8 3E   .o{ћЏЇлЂPW..ґ6Ё>
65 F1 4F 79 B3 D6 8E ED AA C9 C6 09 CC 85 D2 13   ecOyiЦЋнЄЙЖ.М…T.
92 A5 69 12 19 3D 63 55 91 23 2A 9F 6B 81 FF FF   ’Ґi..=cU‘#*џkЃяя
12 6B 7D 7D 77 0F ED 30 69 7F 52 7C F3 43 6C 87   .k}}w.н0i R|УCl‡
8F 32 AD 08 0D 78 E3 66 82 B3 FD C7 FC 8F 40 1D   Џ2...xгf‚iэЗьЏ@.
EA 61 64 F5 A1 57 22 A2 C2 5A 32 C8 FB B9 73 EC   кadxЎW"ЎBZ2Иы№sм
2960   Selected:
```

```
            00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00A1CE3A00  EB 58 90 4D 53 57 49 4E 34 2E 31 00 02 20 70 01   лXђMSWIN4.1.. p.
00A1CE3A10  02 00 00 00 00 F8 00 00 3F 00 80 00 80 1F 00 00   .....ш..?.Ђ.Ђ...
00A1CE3A20  80 60 77 00 78 07 00 00 00 00 00 00 02 00 00 00   Ђ`w.x...........
00A1CE3A30  01 00 08 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00A1CE3A40  00 00 29 C6 A6 7B FA 20 20 20 20 20 20 20 20 20   ..)Ж¦{ъ
00A1CE3A50  20 20 46 41 54 33 32 20 20 20 FA 33 C9 8E D1 BC     FAT32   ъ3ЙЋСj
00A1CE3A60  F8 7B 8E C1 BD 78 00 C5 76 00 1E 56 16 55 BF 22   ш{ЋБSx.Ev..V.Uї"
00A1CE3A70  05 89 7E 00 89 4E 02 B1 0B FC F3 A4 8E D9 BD 00   .‰~.‰N.±.ьу¤ЋЩS.
00A1CE3A80  7C C6 45 FE 0F 8B 46 18 88 45 F9 38 4E 40 7D 25   |ЖEю.‹F.€Eщ8N@}%
00A1CE3A90  8B C1 99 BB 00 07 E8 97 00 72 1A 83 EB 3A 66 A1   ‹Б™»..и—.r.ѓл:fЎ
00A1CE3AA0  1C 7C 66 3B 07 8A 57 FC 75 06 80 CA 02 88 56 02   .|f;.ЉWьu.ЂК.€V.
00A1CE3AB0  80 C3 10 73 ED BF 02 00 83 7E 16 00 75 45 8B 46   ЂГ.sнї..ѓ~..uE‹F
00A1CE3AC0  1C 8B 56 1E B9 03 00 49 40 75 01 42 BB 00 7E E8   .‹V.№..I@u.B».~и
00A1CE3AD0  5F 00 73 26 B0 F8 4F 74 1D 8B 46 32 33 D2 B9 03   _.s&°шOt.‹F23Т№.
00A1CE3AE0  00 3B C8 77 1E 8B 76 0E 3B CE 73 17 2B F1 03 46   .;Иw.‹v.;Os.+c.F
00A1CE3AF0  1C 13 56 1E EB D1 73 0B EB 27 83 7E 2A 00 77 03   ..V.лСs.л'ѓ~*.w.
00A1CE3B00  E9 FD 02 BE 7E 7D AC 98 03 F0 AC 84 C0 74 17 3C   йэ.s~}¬˜.p¬„At.<
00A1CE3B10  FF 74 09 B4 0E BB 07 00 CD 10 EB EE BE 81 7D EB   яt.ґ.»..H.лosЃ}л
00A1CE3B20  E5 BE 7F 7D EB E0 98 CD 16 5E 1F 66 8F 04 CD 19   es }ла˜H.^.fЏ.H.
00A1CE3B30  41 56 66 6A 00 52 50 06 53 6A 01 6A 10 8B F4 60   AVfj.RP.Sj.j.‹ф`
00A1CE3B40  80 7E 02 0E 75 04 B4 42 EB 1D 91 92 33 D2 F7 76   Ђ~..u.ґBл.‘’3Тчv
00A1CE3B50  18 91 F7 76 18 42 87 CA F7 76 1A 8A F2 8A E8 C0   .‘чv.B‡Кчv.ЉтЉиА
00A1CE3B60  CC 02 0A CC B8 01 02 8A 56 40 CD 13 61 8D 64 10   М..Мё..ЉV@H.aЌd.
Address:  2714647040   Selected:  0
```

```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
3A 09 20 AB F9 5B A9 B2 06 12 A8 34 39 93 31 D7   :. «щ[©I..Ё49“1Ч
35 13 51 8E 99 91 6E F7 93 89 D3 99 92 73 74 BA   5.QЋ™‘nч“‰У™’stє
B9 1C 9B 1E AF 42 BA 15 E1 A5 C9 E8 33 9C 68 A8   №.›.ЇBє.бҐЙи3њhЁ
4B BC B2 AA 6B 19 35 AB 55 22 18 B3 94 D4 BB FB   KjIЄk.5«U".i”Ф»ы
BF 14 B4 E2 57 3D 2B CD E8 4F 80 45 CC 12 E8 A3   ї.ґвW=+НиOЂEМ.иJ
6E A3 FD 4C B5 30 82 F6 02 52 43 7F 24 3B F3 03   nJэLµ0‚ц.RC $;У.
07 04 E1 F1 23 A2 83 E0 CF 18 8F 9B 21 39 BB C7   ..бс#ЎѓаП.Џ›!9»З
B0 E1 36 A2 2F 99 7E 19 1D 5D 32 F6 3D 40 35 D4   °б6Ў/™~..]2ц=@5Ф
54 8C 55 DA 0F C7 F3 EF 6D 6D 06 7A C9 04 DD 7A   TЊUЪ.ЗУпmm.zЙ.Эz
59 F6 16 2F 5E FF 94 07 C3 E4 F3 EF E9 E2 D0 AF   Yц./^я”.ГдУпйвPЇ
7E 31 6F B5 30 06 79 D4 87 F4 4A 96 F4 9A F3 68   ~1oµ0.yФ‡фJ–фљУh
FF 38 FA FD 33 B4 9E CF 9F 99 41 0B 6B F6 42 41   я8ъэ3ґћПџ™A.kцBA
9B 99 3D 17 46 92 1B 15 8A 5D F5 F3 3F 8C 7E B5   ›™=.F’..Љ]хУ?Њ~µ
64 EC 33 64 FB 53 FA 52 D3 09 60 B8 21 BD B0 9E   dМ3dыSъRУ.`ё!S°ћ
B3 83 B4 25 52 3F E7 86 89 7F E0 23 72 E6 7C AA   iѓґ%R?з†‰ a#rж|Є
D3 1D 99 2A 77 D8 3B 9B E6 B8 36 DD 2E 37 33 70   У.™*wШ;›жё6Э.73p
86 4E 7A E2 EF 39 87 4C CA B5 A3 6D BE 7E 65 85   †Nzвп9‡LКµJms~e…
FF 80 7A DE 64 43 D3 D9 05 24 44 76 22 9C B8 CD   яЂzЮdCУЩ.$Dv"њёH
D7 D9 79 38 69 12 7B C8 A9 09 1F 3D 00 26 0C B2   ЧЩy8i.{И©..=.&.I
95 79 6F 9E DB E2 84 C1 BB 57 C6 A5 B5 70 91 2E   •yoћЫв„Б»WЖҐµp‘.
E4 4B 6F CA B5 27 E2 68 AB D2 B0 4D 4F 13 2E 0B   дKoКµ'вh«T°MO...
B8 0A D2 DB B5 73 08 22 1B AE 17 FD 8B 34 BD 98   ё.TЫµs.".®.э‹4S˜
68 D4 57 91 D6 F3 3F 08 F1 63 B1 5B 06 D6 34 3A   hФW‘ЦУ?.cc±[.Ц4:
0A 3D 28 D0 6B 66 B0 8A CE 9E 34 95 6E 0F 00 72   .=(Pkf°ЉOћ4•n..r
62 77 F3 D0 7F 1B AD 0A 7E A2 04 72 9A E4 08 61   bwУP ...~Ў.rљд.a
91 7A 0E 01 DF B6 12 97 26 4F 38 41 60 93 CA 59   ‘z..Я¶.—&O8A`“KY
4D 8B 8F 20 38 16 13 D5 85 C0 92 0C 75 14 F4 75   M‹Џ 8..X…A’.u.фu
```

In Bitmap it looks like on following picture. The XOR'd data is on the left, the decrypted data is on the right, the XOR key is at the bottom



Byte position: 1316; Row: 318215; Address: 2749378916; Valu
Address: 2748359355 Selected: 0

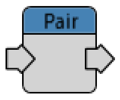
Byte position: 75; Row: 320444; Address: 2748127819; Value: 151
Address: 2748359355 Selected: 0

XOR'd data looks like the noise and has no patterns.

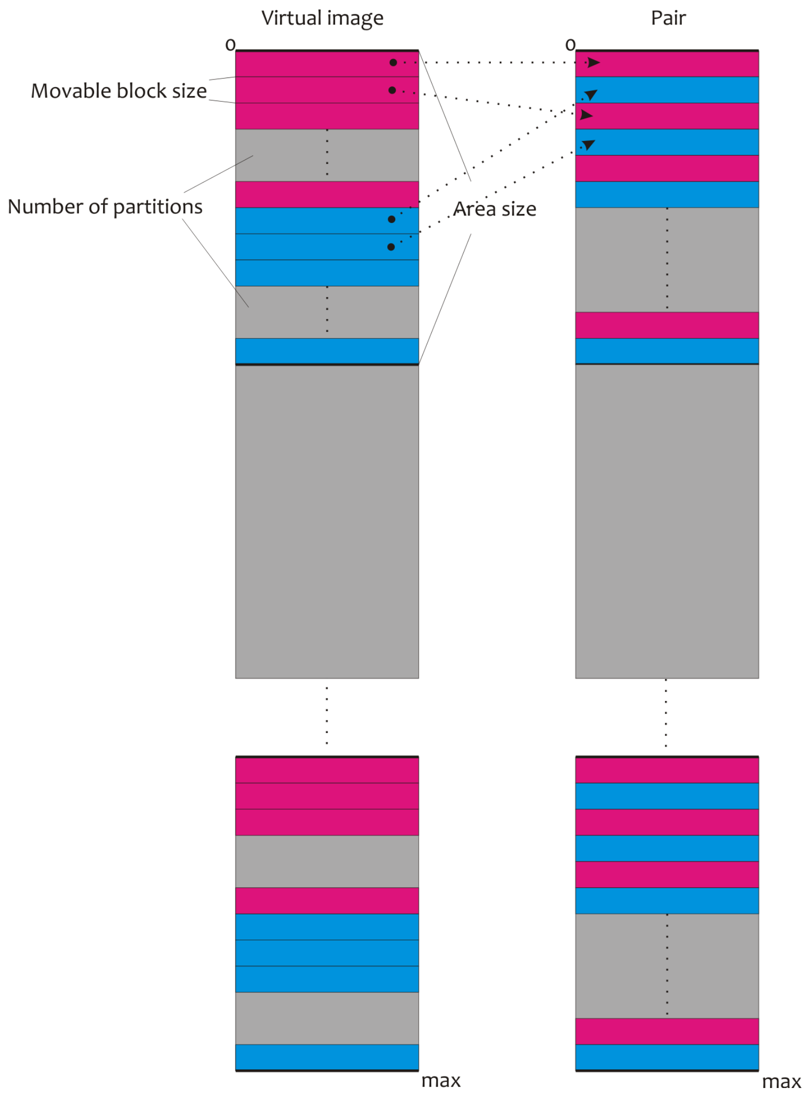
Decrypted or original data usually has different horizontal patterns.

XOR key has very specific patterns, depending on controller it may have totally different look.

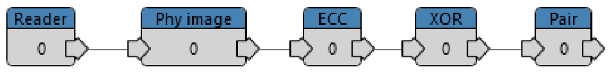
The online library of XOR key patterns can be found on the website: http://rusolut.com/xor-key-library/



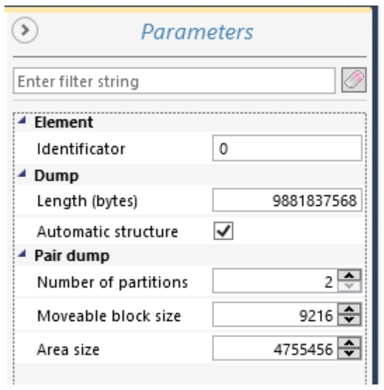
The Pair element performs a page reordering within virtual block, according to multi-plane block allocation scheme.



The Pair element must be connected to the virtual image at the end of scheme, if controller used multi-plane block allocation scheme (used in 90% of cases)



The Parameters tab contains settings of Pair element.



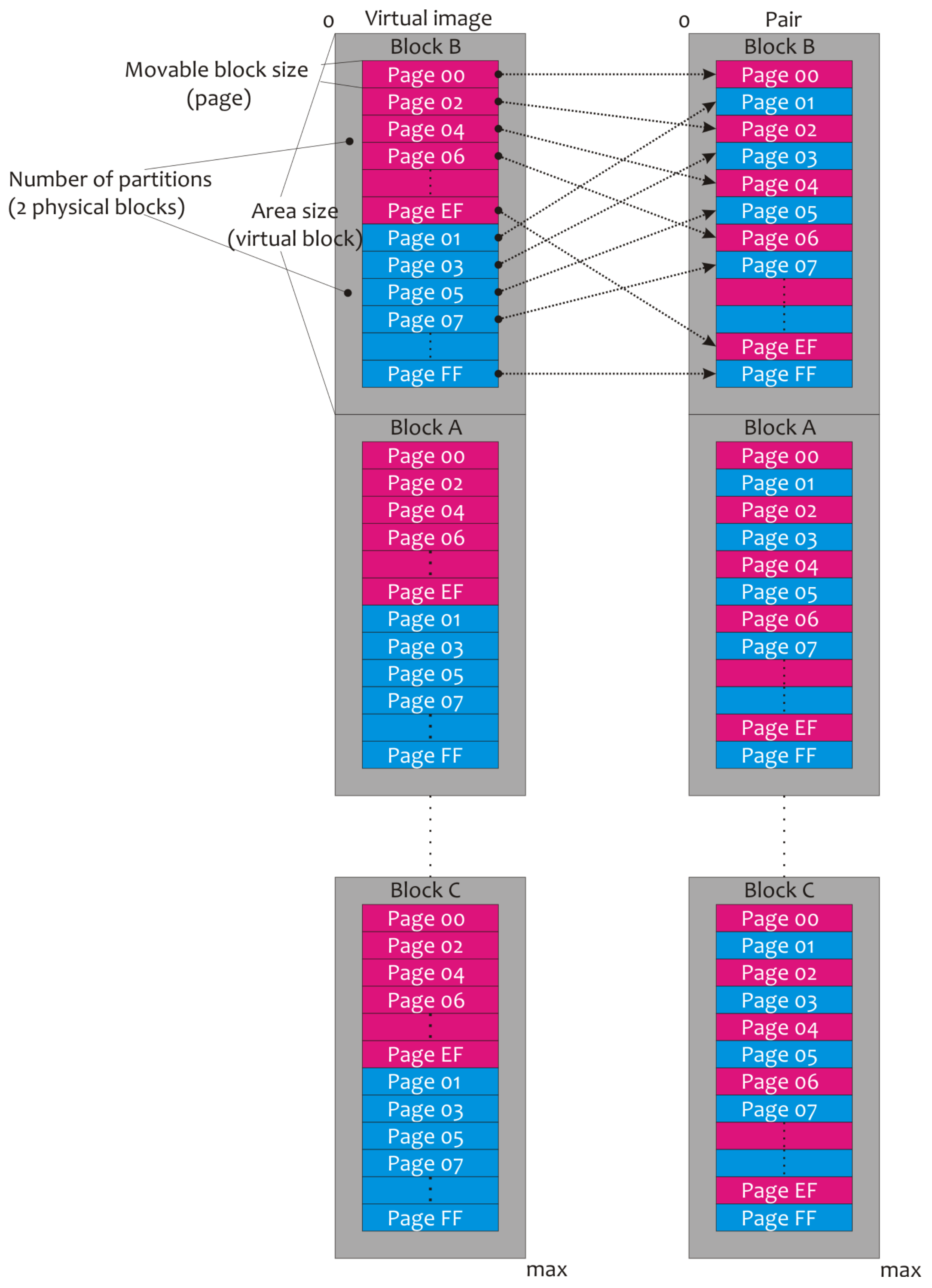
Element	
Identificator	0
Dump	
Length (bytes)	9881837568
Automatic structure	✓
Pair dump	
Number of partitions	2
Moveable block size	9216
Area size	4755456

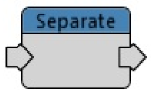
Number of partitions defines the number of parts on which the Area will be divided. Normally it's equal 2, sometimes 4, depending on how many physical blocks used for virtual block allocation (2-plane and 4-plane interleaving)

Movable block size determines the area size that is being reordered inside the Area size. Normally it's equal to page size of NAND chip.

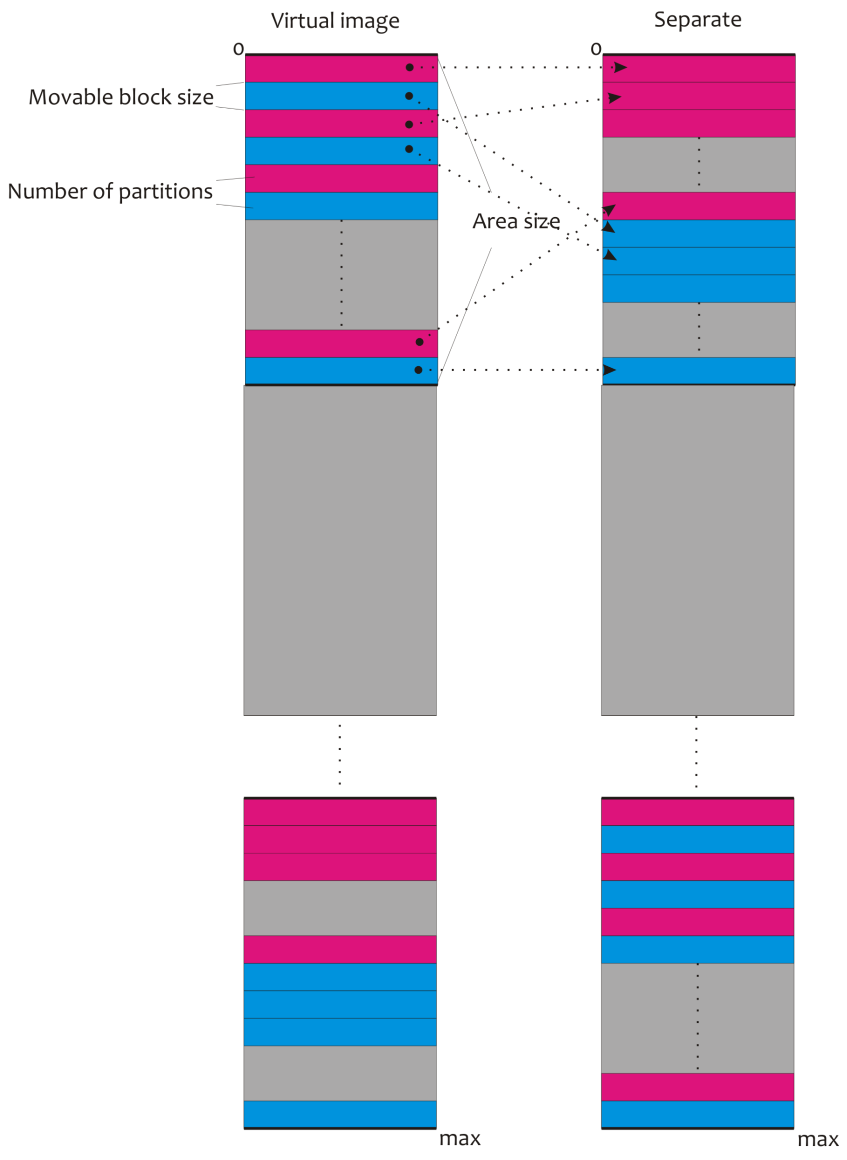
Area size defines the periodic area, inside of which there are reorderings of movable areas. Normally it's equal to virtual block size (virtual block consists of 2 or 4 physical blocks, depending on multi-plane block allocation scheme).

The Pair operation for 2-plane block allocation scheme (virtual block = 2 physical blocks) is shown below:

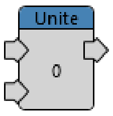




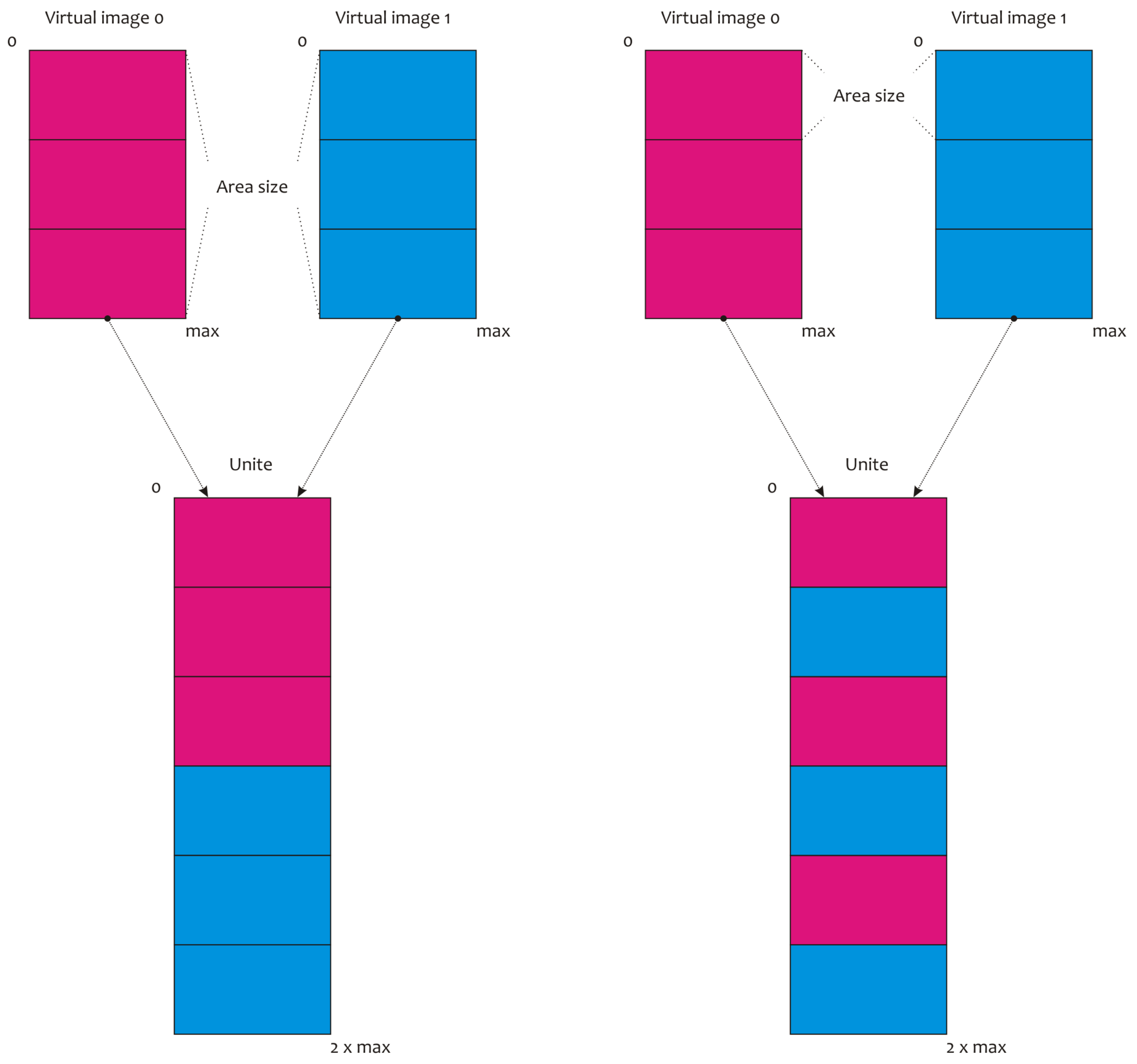
The Separate element performs a page reordering within virtual block, reverse to Pair element.



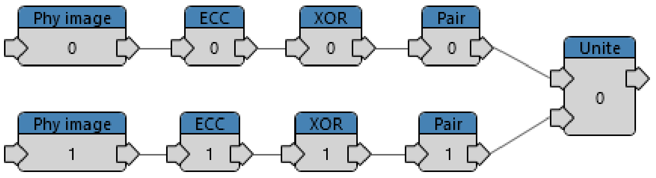
It has same parameters as Pair element



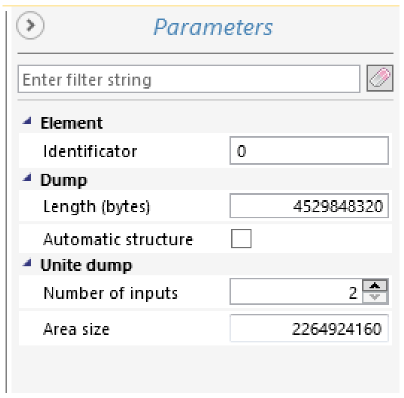
The Unite element joins virtual images together (dumps of crystals, NAND chips) with specific step. This operation is used to join NAND chips and crystals of one chip, also during Bad Column removal. When more than one physical image presented in case, they all must be united at the end of analysis to build the logical image, according to block allocation scheme (sequentially or parrallel).



The Unite element joins the virtual images at the end of analysis.



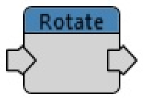
The Parameters tab contains settings of Unite element.



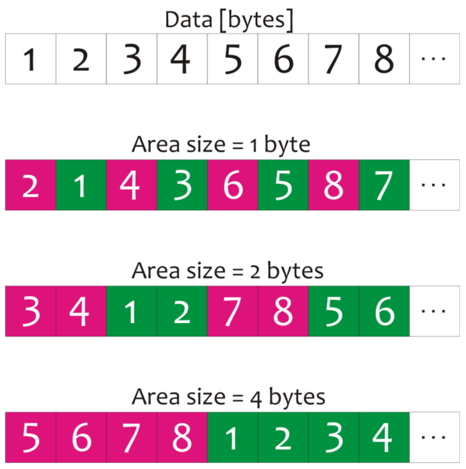
Element	
Identificator	0
Dump	
Length (bytes)	4529848320
Automatic structure	☐
Unite dump	
Number of inputs	2
Area size	2264924160

Number of inputs defines the number of dumps which will be united periodically (Area size).

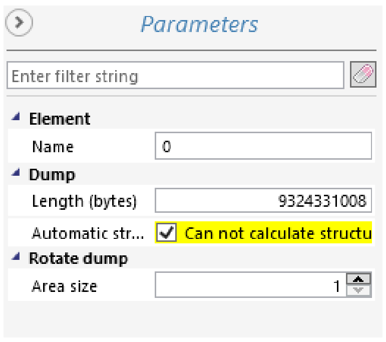
Area size defines the periodic area that is taken from several dumps and joint together. Normally it’s equal to page size (parallel) or dump size (sequentially). Sometimes it may be equal to 1 or 8 bytes, depending on controller’s block allocation scheme.



The Rotate element changes byte order (group of bytes) in the whole dump.



The Parameters tab contains settings of Rotate element.



Element

Name	0

Dump

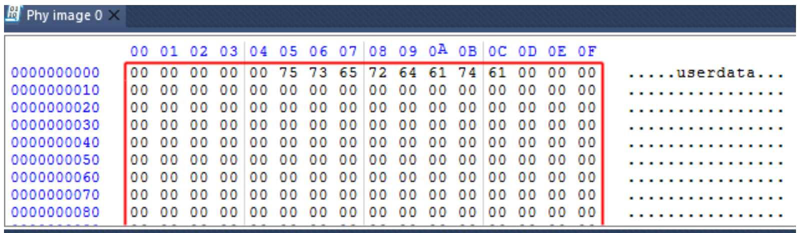
Length (bytes)	9324331008
Automatic str...	✓ Can not calculate structu

Rotate dump

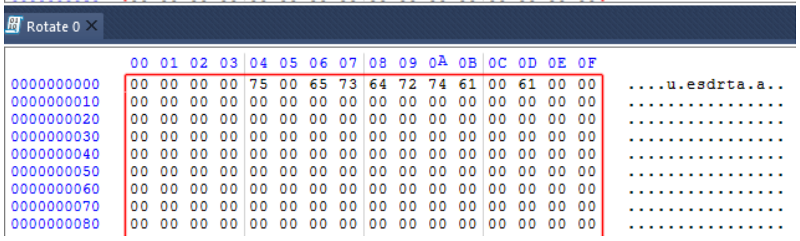
Area size	1

Area size defines the period of rotation, in

Area size = 1 byte

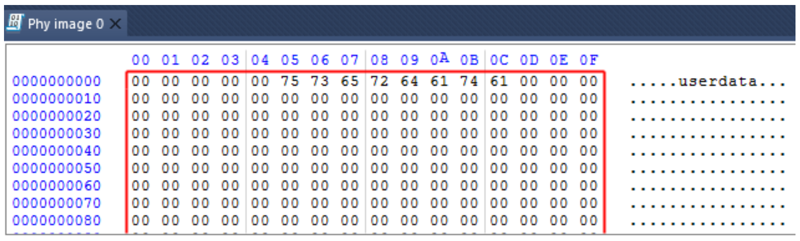

Phy image 0

```
            00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0000000000  00 00 00 00 00 75 73 65 72 64 61 74 61 00 00 00   .....userdata...
0000000010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000040  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
```

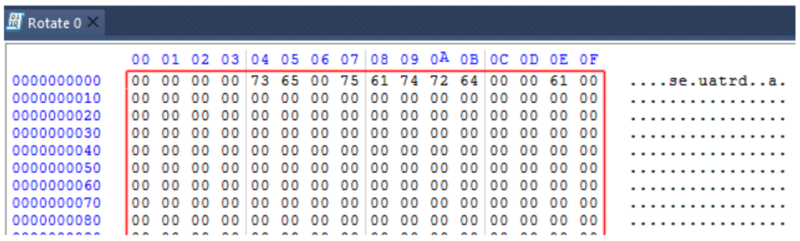

Rotate 0

```
            00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0000000000  00 00 00 00 75 00 65 73 64 72 74 61 00 61 00 00   ....u.esdrta.a..
0000000010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000040  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
```

Area size = 2 bytes

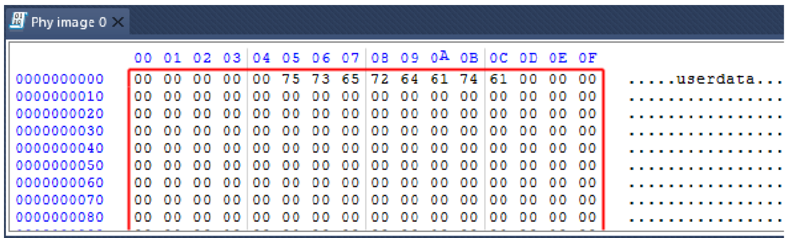

Phy image 0

```
            00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0000000000  00 00 00 00 00 75 73 65 72 64 61 74 61 00 00 00   .....userdata...
0000000010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000040  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
```

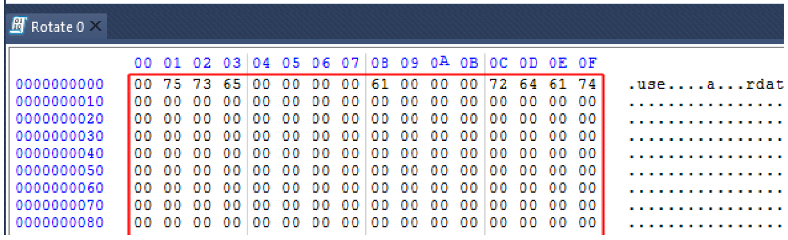

Rotate 0

```
            00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0000000000  00 00 00 00 73 65 00 75 61 74 72 64 00 00 61 00   ....se.uatrd..a.
0000000010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000040  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
```

Area size = 4 bytes


Phy image 0

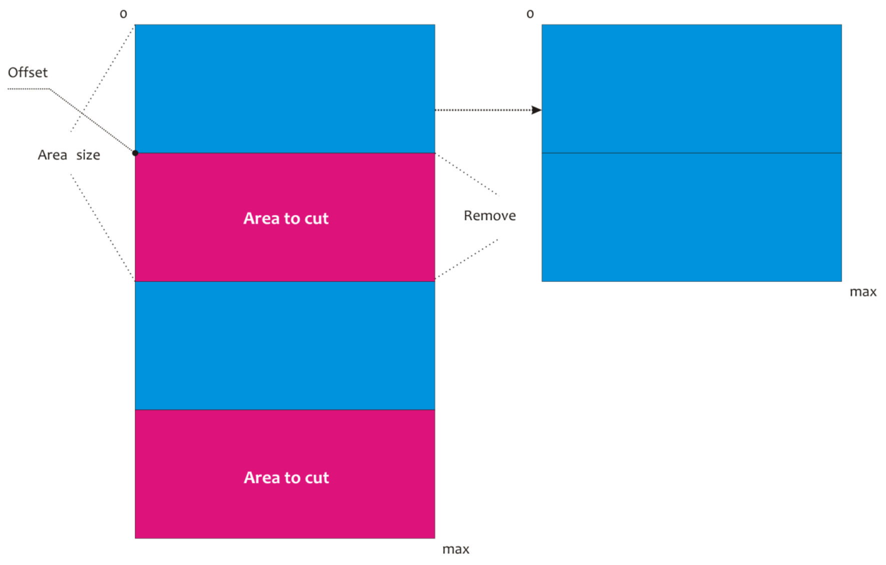
```
            00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0000000000  00 00 00 00 00 75 73 65 72 64 61 74 61 00 00 00   .....userdata...
0000000010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000040  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
```

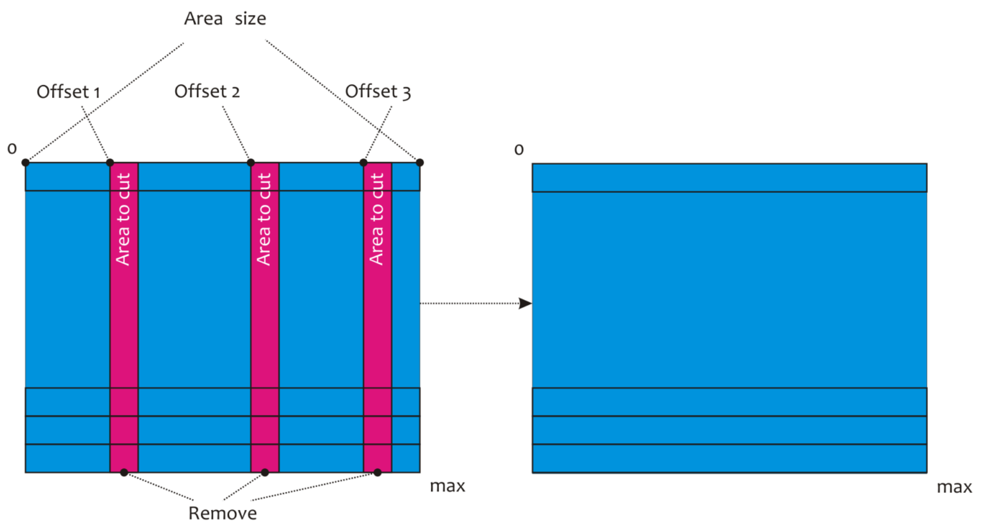

Rotate 0

```
            00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0000000000  00 75 73 65 00 00 00 00 61 00 00 00 72 64 61 74   .use....a...rdat
0000000010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000040  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0000000080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
```

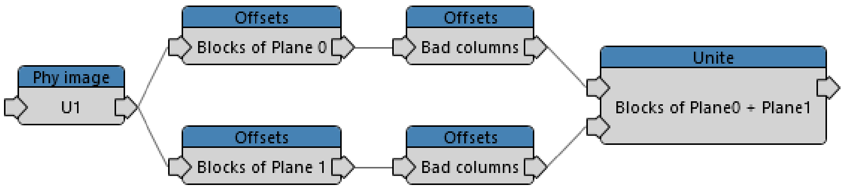
Offsets
0

The Offsets element allows to cut/add bytes at specific offsets within periodical areas.

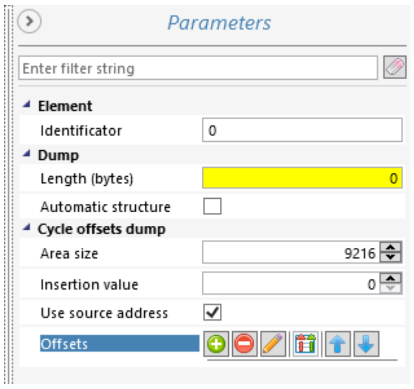




The Offset element used for Bad Column removal operation, as well as for some other non-standard transformations.



The Parameters tab contains settings of Offsets element.



Parameters

Enter filter string

Element
Identificator: 0

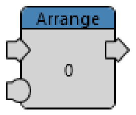
Dump
Length (bytes): 0
Automatic structure: ☐

Cycle offsets dump
Area size: 9216
Insertion value: 0
Use source address: ✓
Offsets

Area size defines the periodical area where offsets are added. It equals to physical block/page size during bad column removal operation.

Insertion value defines the pattern which will fill the added bytes (this option is not used for data recovery purpose usually).

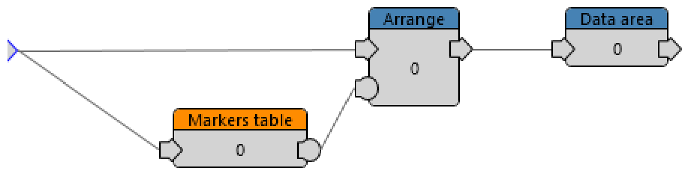
Offsets option is used for manipulations with the physical image - cut/add bytes, edit offset, remove bad column. Number of offsets is unlimited, they can be sonnected in sequence for complex transformations.



The Arrange blocks element is the container of logical image. It is the result of block translation from virtual image to logical through markers table.

The Arrange blocks element has two inputs and one output. The source element (XOR, Pair, Unite or any other) must be connected to the upper input, the Markers table element to lower input (it sets the blocks order - translator)

The Arrange blocks element must be added in combination with Markers table and Data Area elements.



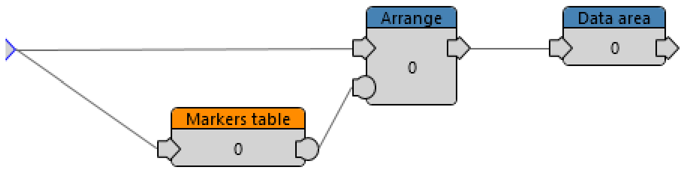
This element has no adjustable parameters.

The logical image is shaped inside the Arrange blocks element (including Data Area and Spare Area of pages), after analysis and block table creation via Markers table.

The Data area element is designed to extract the data area of page from logical image, stored in Arrange blocks element.

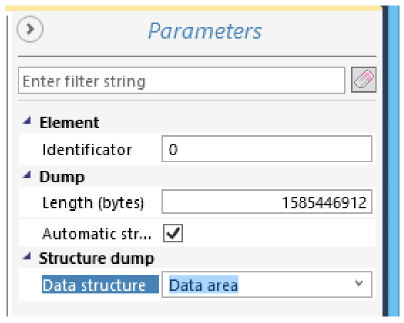
The Data area element has one input and output, it is the final element in process of physical image transformation. To save the user's data, the File System viewer must be opened on Data area element. It is possible to save the logical image to binary file from this element, for further analysis in forensic tools, using the functions from dump viewer menu (Save all).

The Data area element must be added in combination with Arrange blocks and Markers table elements.



The Parameters tab contains settings of Data area element.

Data structure defines the data area structure of page. it must be predefined in structure viewer and then choosen from this menu.



Parameters

Enter filter string

Element

Identificator	0

Dump

Length (bytes)	1585446912
Automatic str...	✓

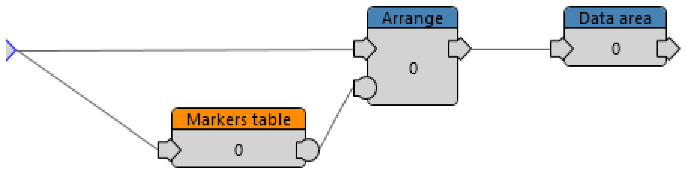
Structure dump

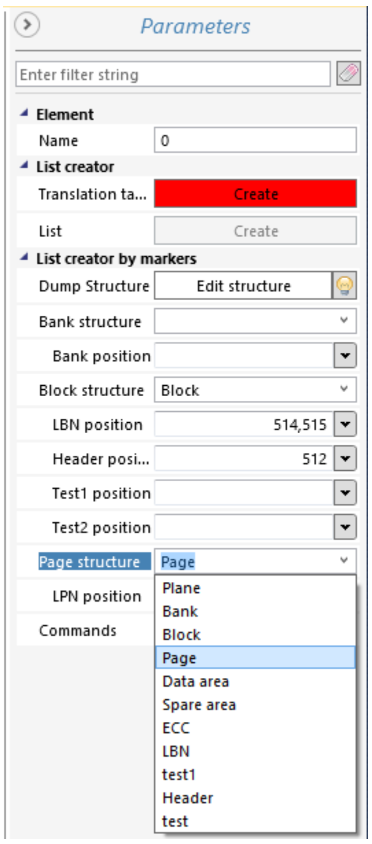
Data structure	Data area

Markers table
1

The Markers table element is designed for physical/virtual block translation into logical image, according to spare area of page and it's parameters (positions of different markers - LBN, Header, etc.). These parameters are set in Structure viewer mode at the step of dump structure analysis.

The Markers table element must be added in combination with Arrange blocks and Data area elements.





Parameters

Enter filter string

Element

Name	0

List creator

Translation ta...	Create
List	Create

List creator by markers

Dump Structure	Edit structure
Bank structure	
Bank position	
Block structure	Block
LBN position	514,515
Header posi...	512
Test1 position	
Test2 position	
Page structure	Page
LPN position	
Commands	

The Parameters tab contains settings of the Markers table element.

Dump structure must be copied from source-element using button

The Block structure is essential and it must be set from drop-down menu.

The Page structure is essential and it must be set from drop-down menu.

The Bank structure is optional, depends on how many banks NAND space is divided. It can be set from drop-down menu.

The LBN structure is essential and it must be set from drop-down menu. In case if LBN has reverse order (e.g.1025,1024), it can be changed manually in the LBN field.

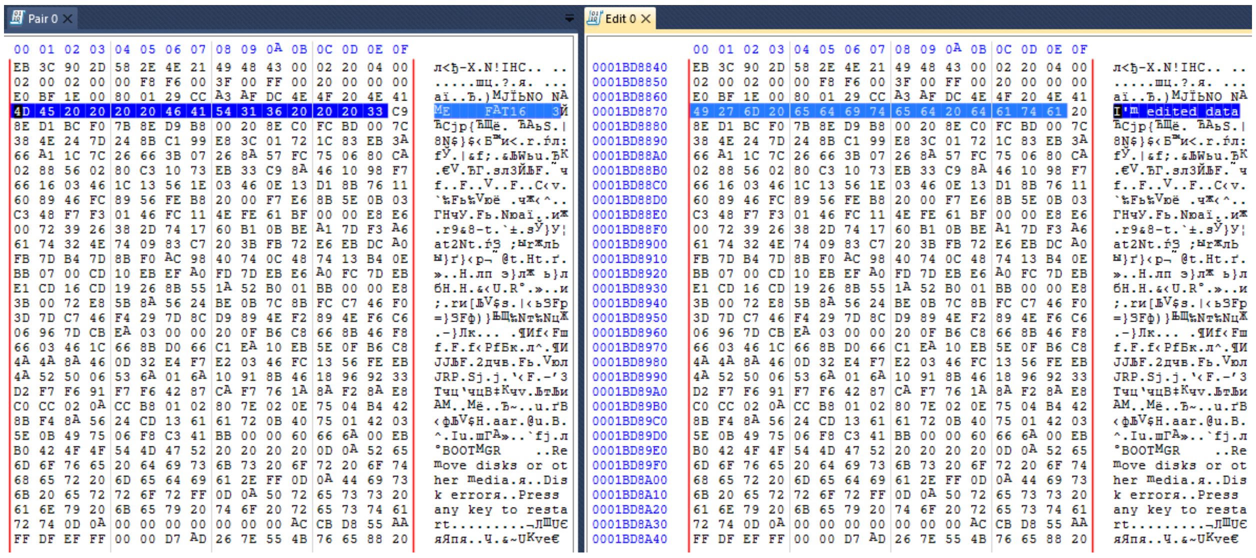
The Header structure is usually essential and it can be set from drop-down menu.

Other structures are optional and can be set from drop-down menu.

The Create translation table button is used to build table of virtual blocks (logical block distribution across physical blocks with mixed order) for further analysis in block table and block re-ordering and filtering.

The Create List button is used to build the logical image in Arrange blocks element (after sorting and filtering blocks according to LBN).

The Edit element virtually changes dump. This operation doesn't bring any changes directly to dump-file of physical image, it does virtual modification. It is useful in cases when some parameters of file system must be changed and other non-standard situations (e.g. to change sector size value in boot sector from 2048b to 512b).



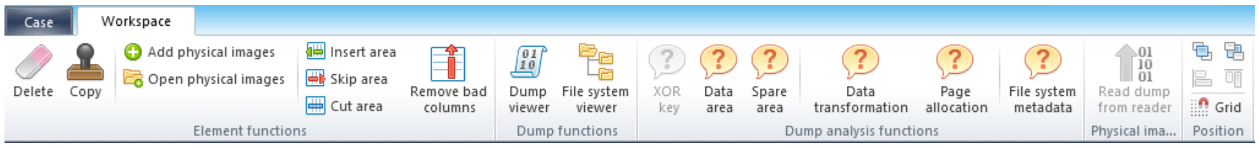
```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
EB 3C 90 2D 58 2E 4E 21 49 48 43 00 02 20 04 00   л<ђ-X.N!IHC.. ..
02 00 02 00 00 F8 F6 00 3F 00 FF 00 20 00 00 00   .....шц.?.я. ...
E0 BF 1E 00 80 01 29 CC A3 AF DC 4E 4F 20 4E 41   аї..Ђ.)МЈЇЬNO NA
4D 45 20 20 20 20 46 41 54 31 36 20 20 20 33 C9   ME    FAT16   3Й
8E D1 BC F0 7B 8E D9 B8 00 20 8E C0 FC BD 00 7C   ЋСјр{ЋЩё. ЋАьЅ.|
38 4E 24 7D 24 8B C1 99 E8 3C 01 72 1C 83 EB 3A   8N$}$‹Б™и<.r.ѓл:
66 A1 1C 7C 26 66 3B 07 26 8A 57 FC 75 06 80 CA   fЎ.|&f;.&ЉWьu.ЂК
02 88 56 02 80 C3 10 73 EB 33 C9 8A 46 10 98 F7   .€V.ЂГ.sл3ЙЉF.˜ч
66 16 03 46 1C 13 56 1E 03 46 0E 13 D1 8B 76 11   f..F..V..F..С‹v.
60 89 46 FC 89 56 FE B8 20 00 F7 E6 8B 5E 0B 03   `‰Fь‰Vюё .чж‹^..
C3 48 F7 F3 01 46 FC 11 4E FE 61 BF 00 00 E8 E6   ГНчу.Fь.Nюaї..иж
00 72 39 26 38 2D 74 17 60 B1 0B BE A1 7D F3 A6   .r9&8-t.`±.ѕЎ}у¦
61 74 32 4E 74 09 83 C7 20 3B FB 72 E6 EB DC A0   at2Nt.ѓЗ ;ыrжлЬ 
FB 7D B4 7D 8B F0 AC 98 40 74 0C 48 74 13 B4 0E   ы}ґ}‹р¬˜@t.Ht.ґ.
BB 07 00 CD 10 EB EF A0 FD 7D EB E6 A0 FC 7D EB   »..Н.лп э}лж ь}л
E1 CD 16 CD 19 26 8B 55 1A 52 B0 01 BB 00 00 E8   бН.Н.&‹U.R°.»..и
3B 00 72 E8 5B 8A 56 24 BE 0B 7C 8B FC C7 46 F0   ;.rи[ЉV$ѕ.|‹ьЗFр
3D 7D C7 46 F4 29 7D 8C D9 89 4E F2 89 4E F6 C6   =}ЗFф)}ЊЩ‰Nт‰NцЖ
06 96 7D CB EA 03 00 00 20 0F B6 C8 66 8B 46 F8   .–}Лк... .¶Иf‹Fш
66 03 46 1C 66 8B D0 66 C1 EA 10 EB 5E 0F B6 C8   f.F.f‹РfБк.л^.¶И
4A 4A 8A 46 0D 32 E4 F7 E2 03 46 FC 13 56 FE EB   JJЉF.2дчв.Fь.Vюл
4A 52 50 06 53 6A 01 6A 10 91 8B 46 18 96 92 33   JRP.Sj.j.‘‹F.–’3
D2 F7 F6 91 F7 F6 42 87 CA F7 76 1A 8A F2 8A E8   Тчц‘чцB‡Кчv.ЉтЉи
C0 CC 02 0A CC B8 01 02 80 7E 02 0E 75 04 B4 42   АМ..Мё..Ђ~..u.ґB
8B F4 8A 56 24 CD 13 61 61 72 0B 40 75 01 42 03   ‹фЉV$Н.aar.@u.B.
5E 0B 49 75 06 F8 C3 41 BB 00 00 60 66 6A 00 EB   ^.Iu.шГA»..`fj.л
B0 42 4F 4F 54 4D 47 52 20 20 20 20 0D 0A 52 65   °BOOTMGR    ..Re
6D 6F 76 65 20 64 69 73 6B 73 20 6F 72 20 6F 74   move disks or ot
68 65 72 20 6D 65 64 69 61 2E FF 0D 0A 44 69 73   her media.я..Dis
6B 20 65 72 72 6F 72 FF 0D 0A 50 72 65 73 73 20   k errorя..Press 
61 6E 79 20 6B 65 79 20 74 6F 20 72 65 73 74 61   any key to resta
72 74 0D 0A 00 00 00 00 00 00 00 AC CB D8 55 AA   rt.........¬ЛШUЄ
FF DF EF FF 00 00 D7 AD 26 7E 55 4B 76 65 88 20   яЯпя..Ч.&~UKve€
```

```
           00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0001BD8840 EB 3C 90 2D 58 2E 4E 21 49 48 43 00 02 20 04 00   л<ђ-X.N!IHC.. ..
0001BD8850 02 00 02 00 00 F8 F6 00 3F 00 FF 00 20 00 00 00   .....шц.?.я. ...
0001BD8860 E0 BF 1E 00 80 01 29 CC A3 AF DC 4E 4F 20 4E 41   аї..Ђ.)МЈЇЬNO NA
0001BD8870 49 27 6D 20 65 64 69 74 65 64 20 64 61 74 61 20   I'm edited data 
0001BD8880 8E D1 BC F0 7B 8E D9 B8 00 20 8E C0 FC BD 00 7C   ЋСјр{ЋЩё. ЋАьЅ.|
0001BD8890 38 4E 24 7D 24 8B C1 99 E8 3C 01 72 1C 83 EB 3A   8N$}$‹Б™и<.r.ѓл:
0001BD88A0 66 A1 1C 7C 26 66 3B 07 26 8A 57 FC 75 06 80 CA   fЎ.|&f;.&ЉWьu.ЂК
0001BD88B0 02 88 56 02 80 C3 10 73 EB 33 C9 8A 46 10 98 F7   .€V.ЂГ.sл3ЙЉF.˜ч
0001BD88C0 66 16 03 46 1C 13 56 1E 03 46 0E 13 D1 8B 76 11   f..F..V..F..С‹v.
0001BD88D0 60 89 46 FC 89 56 FE B8 20 00 F7 E6 8B 5E 0B 03   `‰Fь‰Vюё .чж‹^..
0001BD88E0 C3 48 F7 F3 01 46 FC 11 4E FE 61 BF 00 00 E8 E6   ГНчу.Fь.Nюaї..иж
0001BD88F0 00 72 39 26 38 2D 74 17 60 B1 0B BE A1 7D F3 A6   .r9&8-t.`±.ѕЎ}у¦
0001BD8900 61 74 32 4E 74 09 83 C7 20 3B FB 72 E6 EB DC A0   at2Nt.ѓЗ ;ыrжлЬ 
0001BD8910 FB 7D B4 7D 8B F0 AC 98 40 74 0C 48 74 13 B4 0E   ы}ґ}‹р¬˜@t.Ht.ґ.
0001BD8920 BB 07 00 CD 10 EB EF A0 FD 7D EB E6 A0 FC 7D EB   »..Н.лп э}лж ь}л
0001BD8930 E1 CD 16 CD 19 26 8B 55 1A 52 B0 01 BB 00 00 E8   бН.Н.&‹U.R°.»..и
0001BD8940 3B 00 72 E8 5B 8A 56 24 BE 0B 7C 8B FC C7 46 F0   ;.rи[ЉV$ѕ.|‹ьЗFр
0001BD8950 3D 7D C7 46 F4 29 7D 8C D9 89 4E F2 89 4E F6 C6   =}ЗFф)}ЊЩ‰Nт‰NцЖ
0001BD8960 06 96 7D CB EA 03 00 00 20 0F B6 C8 66 8B 46 F8   .–}Лк... .¶Иf‹Fш
0001BD8970 66 03 46 1C 66 8B D0 66 C1 EA 10 EB 5E 0F B6 C8   f.F.f‹РfБк.л^.¶И
0001BD8980 4A 4A 8A 46 0D 32 E4 F7 E2 03 46 FC 13 56 FE EB   JJЉF.2дчв.Fь.Vюл
0001BD8990 4A 52 50 06 53 6A 01 6A 10 91 8B 46 18 96 92 33   JRP.Sj.j.‘‹F.–’3
0001BD89A0 D2 F7 F6 91 F7 F6 42 87 CA F7 76 1A 8A F2 8A E8   Тчц‘чцB‡Кчv.ЉтЉи
0001BD89B0 C0 CC 02 0A CC B8 01 02 80 7E 02 0E 75 04 B4 42   АМ..Мё..Ђ~..u.ґB
0001BD89C0 8B F4 8A 56 24 CD 13 61 61 72 0B 40 75 01 42 03   ‹фЉV$Н.aar.@u.B.
0001BD89D0 5E 0B 49 75 06 F8 C3 41 BB 00 00 60 66 6A 00 EB   ^.Iu.шГA»..`fj.л
0001BD89E0 B0 42 4F 4F 54 4D 47 52 20 20 20 20 0D 0A 52 65   °BOOTMGR    ..Re
0001BD89F0 6D 6F 76 65 20 64 69 73 6B 73 20 6F 72 20 6F 74   move disks or ot
0001BD8A00 68 65 72 20 6D 65 64 69 61 2E FF 0D 0A 44 69 73   her media.я..Dis
0001BD8A10 6B 20 65 72 72 6F 72 FF 0D 0A 50 72 65 73 73 20   k errorя..Press 
0001BD8A20 61 6E 79 20 6B 65 79 20 74 6F 20 72 65 73 74 61   any key to resta
0001BD8A30 72 74 0D 0A 00 00 00 00 00 00 00 AC CB D8 55 AA   rt.........¬ЛШUЄ
0001BD8A40 FF DF EF FF 00 00 D7 AD 26 7E 55 4B 76 65 88 20   яЯпя..Ч.&~UKve€
```

This element has no adjustable parameters.

Toolbar

Toolbar contains a set of tools for work with elements. The set of tools is activated when one of the elements is chosen. Available modes depend on chosen element. All functions are divided by several groups



Element functions (available for all elements)

Reader functions (available for Reader element)

Physical Image functions (available for Physical image element)

ECC functions (available for ECC element)

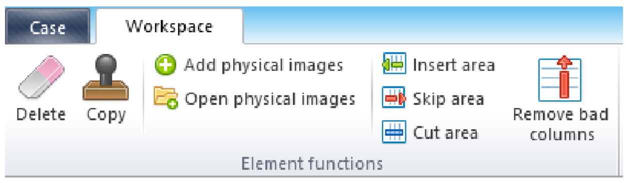
Dump analysis functions (available for elements except Markers table)

Dump functions (available for all elements except Markers table)

Block list operations (available for Markers table element)

Element functions

The group of Element functions contains Delete, Copy, Add physical image, Open physical image, Remove bad columns operations.



Delete operation is used to delete elements and connections between them.

Copy operation is used to clone elements with their parameters. It is used when the same operation applied to multiple dumps.

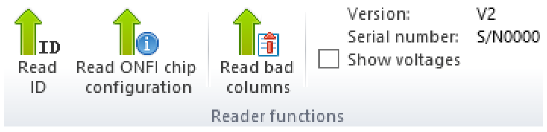
Add physical images operation is used to add empty physical image elements for further chip reading into it. It is used when the new task is being created.

Open physical images operation is used to import dumps to the task. It is used when the task is created from old dumps that have been previously read.

Remove bad columns is used to remove bad columns automatically, from extracted bad column table of chip (Read bad columns operation)

Reader functions

The group of Reader functions contains Read ID, Read ONFI chip configuration, Read bad columns operations.



Version: V2
Serial number: S/N0000
Show voltages

All operations available for Reader are described in element description (Elements section)

Physical Image functions

The group of Physical image functions contains Read dump from Reader operation.

All operations available for physical image element are described in element description (Elements section)

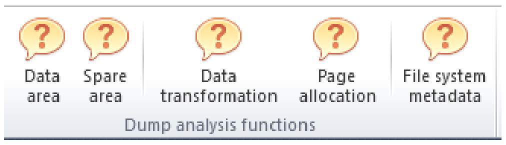
ECC functions

The group of ECC functions contains Reread dump operation.

All operations available for ECC are described in element description (Elements section)

Dump analysis functions

The group of Dump analysis functions contains Data area analysis, Spare area analysis, Data transformation analysis, Page allocation analysis, File system metadata analysis operations.



Data area analysis is used for automatic analysis of page structure. It is used on the stage of physical image structure description.

Spare Area analysis is used for statistical analysis of Spare Area. It is used for detection of LBN, Header and other structures.

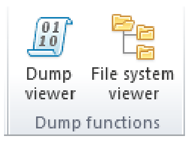
Data transformation analysis is used for automatic analysis of Inversion, Unite by byte, Rotation presence or absence. It is used for finding transformation of user's data in the controller-NAND data transfer channel.

Page allocation analysis is used for determination of type of the virtual block allocation, inside the image and also between chips/crystals. It is applied to single dump for analysis of Pair operation (Multi-plane block allocation) and to several selected images (serial/parallel block allocation schemes determination).

File system metadata analysis is used for search and analysis of file system structures (FAT table/MFT File records). It is used for analysis of page allocation inside/between chips. Analysis based on FAT tables/MFT record sequence.

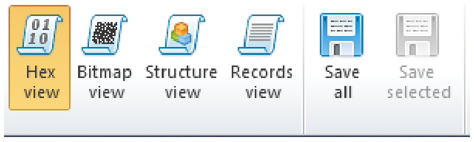
Dump functions

The group of Dump functions includes Dump viewer and File System viewer.

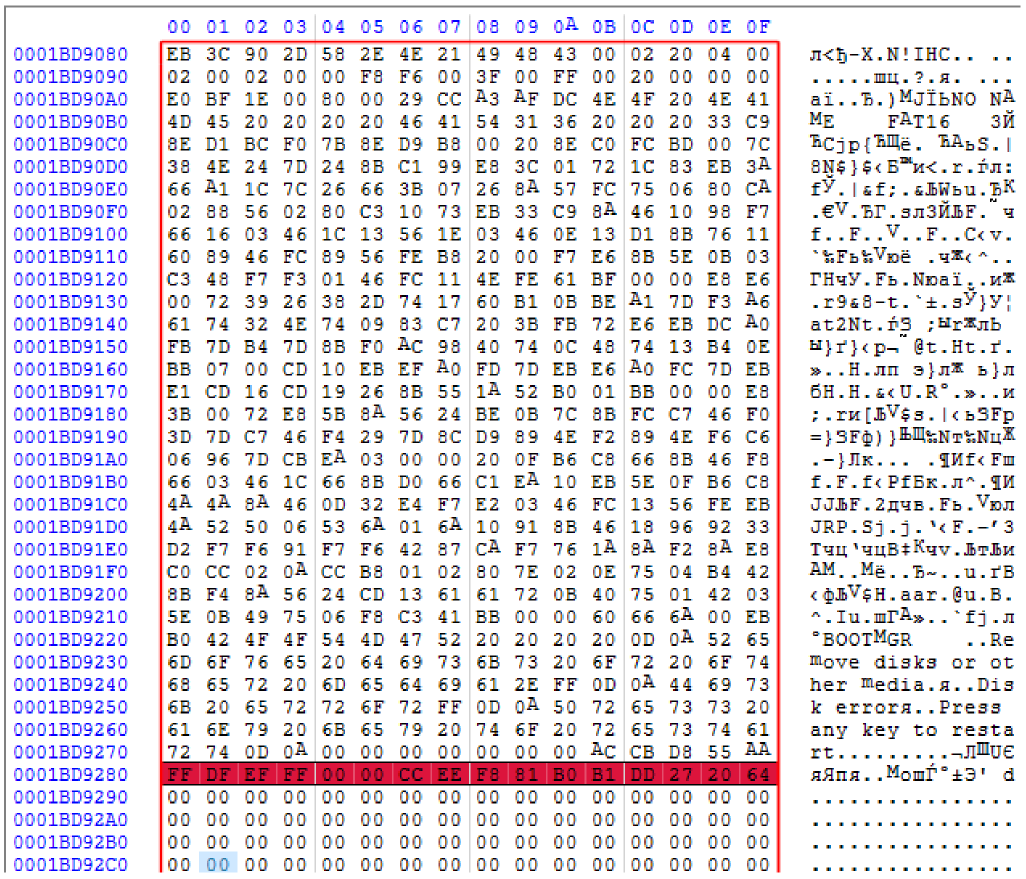


File System viewer is designed for file system browsing and file saving from the logical image.

Dump viewer is designed for visual analysis of the content of physical and virtual images in different data formats. It has several data respresentation and view modes.

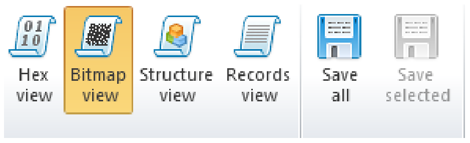


The Hex View mode is the typical hexadecimal representation of data in physical image.



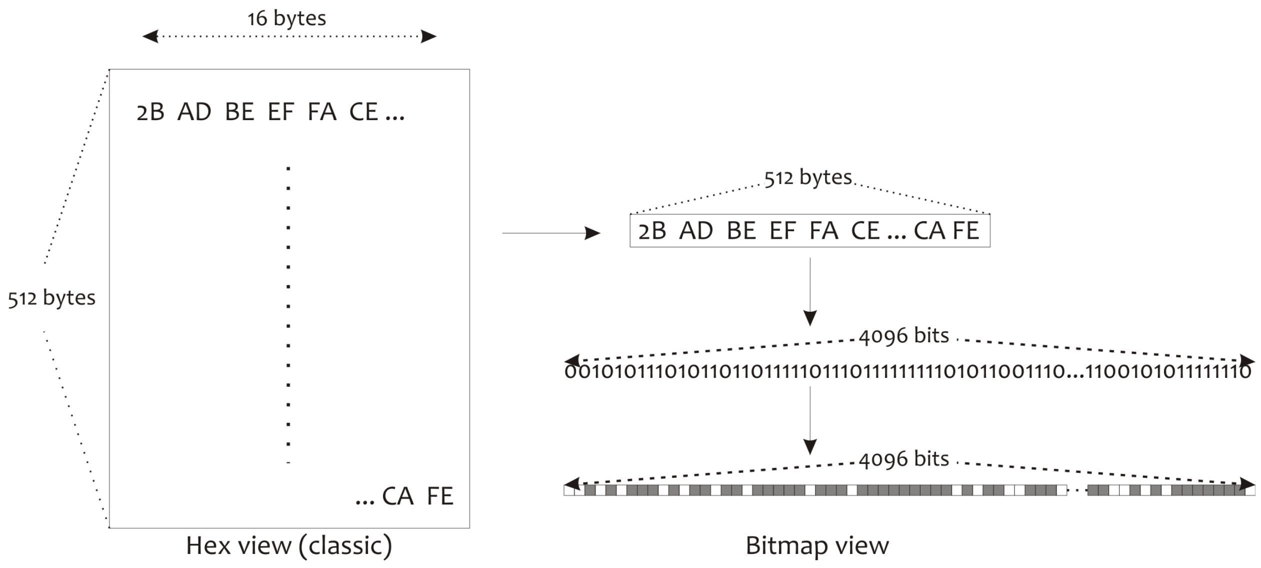
```
             00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0001BD9080   EB 3C 90 2D 58 2E 4E 21 49 48 43 00 02 20 04 00   л<ђ-X.N!IHC.. ..
0001BD9090   02 00 02 00 00 F8 F6 00 3F 00 FF 00 20 00 00 00   .....шц.?.я. ...
0001BD90A0   E0 BF 1E 00 80 00 29 CC A3 AF DC 4E 4F 20 4E 41   аї..Ђ.)МЈЇЬNO NA
0001BD90B0   4D 45 20 20 20 20 46 41 54 31 36 20 20 20 33 C9   ME    FAT16   3Й
0001BD90C0   8E D1 BC F0 7B 8E D9 B8 00 20 8E C0 FC BD 00 7C   ЋСјр{ЋЩё. ЋАьЅ.|
0001BD90D0   38 4E 24 7D 24 8B C1 99 E8 3C 01 72 1C 83 EB 3A   8N$}$‹Б™и<.r.ѓл:
0001BD90E0   66 A1 1C 7C 26 66 3B 07 26 8A 57 FC 75 06 80 CA   fЎ.|&f;.&ЉWьu.ЂК
0001BD90F0   02 88 56 02 80 C3 10 73 EB 33 C9 8A 46 10 98 F7   .€V.ЂГ.sл3ЙЉF.˜ч
0001BD9100   66 16 03 46 1C 13 56 1E 03 46 0E 13 D1 8B 76 11   f..F..V..F..С‹v.
0001BD9110   60 89 46 FC 89 56 FE B8 20 00 F7 E6 8B 5E 0B 03   `‰Fь‰Vюё .чж‹^..
0001BD9120   C3 48 F7 F3 01 46 FC 11 4E FE 61 BF 00 00 E8 E6   ГНчу.Fь.Nюaї..иж
0001BD9130   00 72 39 26 38 2D 74 17 60 B1 0B BE A1 7D F3 A6   .r9&8-t.`±.ѕЎ}у¦
0001BD9140   61 74 32 4E 74 09 83 C7 20 3B FB 72 E6 EB DC A0   at2Nt.ѓЗ ;ыrжлЬ 
0001BD9150   FB 7D B4 7D 8B F0 AC 98 40 74 0C 48 74 13 B4 0E   ы}ґ}‹р¬˜@t.Ht.ґ.
0001BD9160   BB 07 00 CD 10 EB EF A0 FD 7D EB E6 A0 FC 7D EB   »..Н.лп э}лж ь}л
0001BD9170   E1 CD 16 CD 19 26 8B 55 1A 52 B0 01 BB 00 00 E8   бН.Н.&‹U.R°.»..и
0001BD9180   3B 00 72 E8 5B 8A 56 24 BE 0B 7C 8B FC C7 46 F0   ;.rи[ЉV$ѕ.|‹ьЗFр
0001BD9190   3D 7D C7 46 F4 29 7D 8C D9 89 4E F2 89 4E F6 C6   =}ЗFф)}ЊЩ‰Nт‰NцЖ
0001BD91A0   06 96 7D CB EA 03 00 00 20 0F B6 C8 66 8B 46 F8   .–}Лк... .¶Иf‹Fш
0001BD91B0   66 03 46 1C 66 8B D0 66 C1 EA 10 EB 5E 0F B6 C8   f.F.f‹РfБк.л^.¶И
0001BD91C0   4A 4A 8A 46 0D 32 E4 F7 E2 03 46 FC 13 56 FE EB   JJЉF.2дчв.Fь.Vюл
0001BD91D0   4A 52 50 06 53 6A 01 6A 10 91 8B 46 18 96 92 33   JRP.Sj.j.‘‹F.–’3
0001BD91E0   D2 F7 F6 91 F7 F6 42 87 CA F7 76 1A 8A F2 8A E8   Тчц‘чцB‡Кчv.ЉтЉи
0001BD91F0   C0 CC 02 0A CC B8 01 02 80 7E 02 0E 75 04 B4 42   АМ..Мё..Ђ~..u.ґB
0001BD9200   8B F4 8A 56 24 CD 13 61 61 72 0B 40 75 01 42 03   ‹фЉV$Н.aar.@u.B.
0001BD9210   5E 0B 49 75 06 F8 C3 41 BB 00 00 60 66 6A 00 EB   ^.Iu.шГA»..`fj.л
0001BD9220   B0 42 4F 4F 54 4D 47 52 20 20 20 20 0D 0A 52 65   °BOOTMGR    ..Re
0001BD9230   6D 6F 76 65 20 64 69 73 6B 73 20 6F 72 20 6F 74   move disks or ot
0001BD9240   68 65 72 20 6D 65 64 69 61 2E FF 0D 0A 44 69 73   her media.я..Dis
0001BD9250   6B 20 65 72 72 6F 72 FF 0D 0A 50 72 65 73 73 20   k errorя..Press 
0001BD9260   61 6E 79 20 6B 65 79 20 74 6F 20 72 65 73 74 61   any key to resta
0001BD9270   72 74 0D 0A 00 00 00 00 00 00 00 AC CB D8 55 AA   rt.........¬ЛШUЄ
0001BD9280   FF DF EF FF 00 00 CC EE F8 81 B0 B1 DD 27 20 64   яЯпя..Момѓ°±Э' d
0001BD9290   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0001BD92A0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0001BD92B0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0001BD92C0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
```

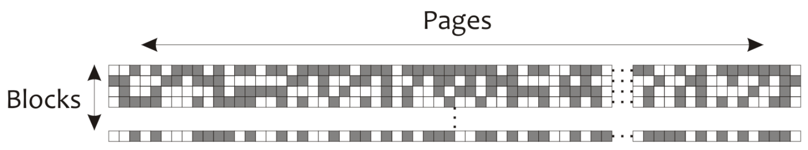
The whole hexadecimal array of bytes is divided by 16 bytes which is the industry standard.
The current address in hexadecimal format is displayed on the left, which means relative offset from the beginning of dump.
The byte address inside the line is displayed above with an accuracy to 1 byte.
In the center window there is user's data in hexadecimal representation.

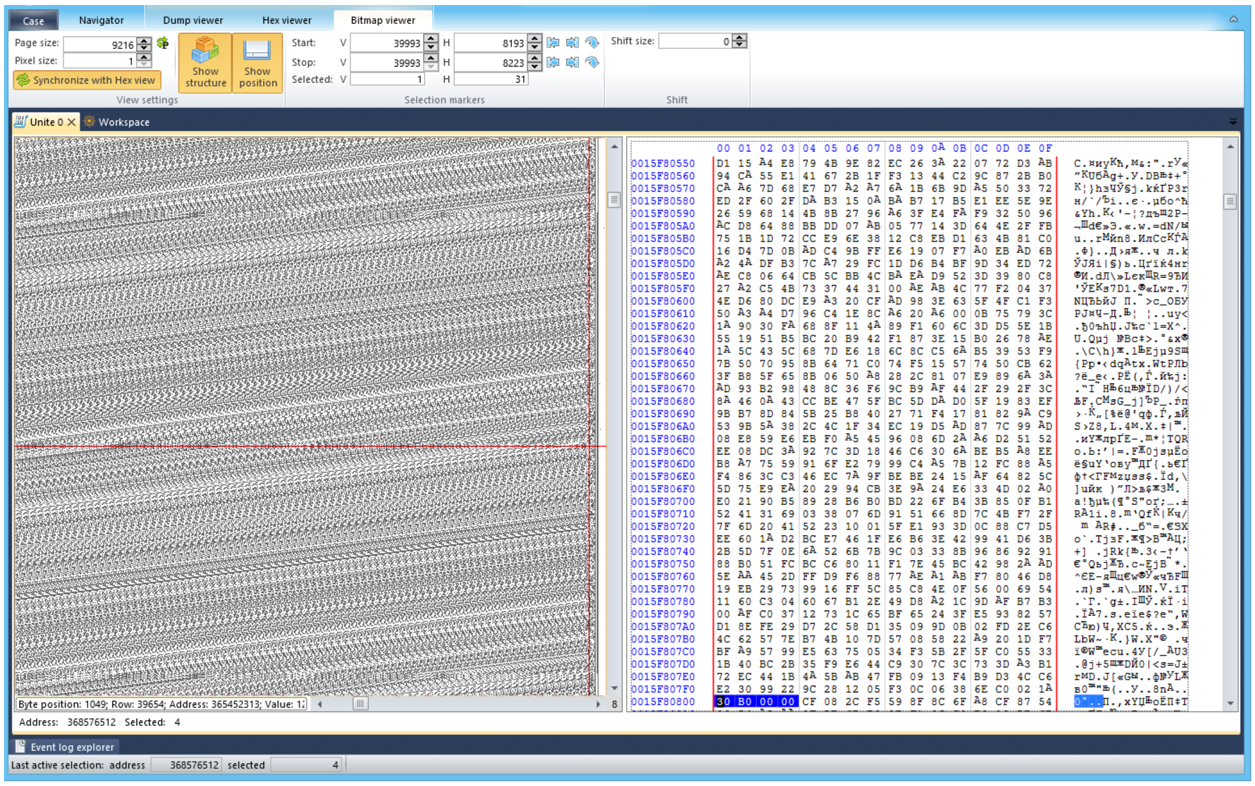


The Bitmap View mode is a tool for visualization of binary representation of dump. In this mode ~60-80% of dump analysis process takes place. It allows to analyze: data patterns, virtual block size, bad columns, ECC errors, XOR key, page structure, spare area structure, virtual block allocation scheme and many other patterns.

Transition from Hexadecimal to Bitmap data representation is performed in a following way







Page size: 9216
Pixel size: 1
Start: V 39993 H 8193
Stop: V 39993 H 8223
Selected: V 1 H 31
Shift size: 0

```
            00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0015F80550  D1 15 A4 E8 79 4B 9E 82 EC 26 3A 22 07 72 D3 AB   C.¤иуKћ,м&:".rУ«
0015F80560  94 CA 55 E1 41 67 2B 1F F3 13 44 C2 9C 87 2B B0   ”KЏбAg+.У.DBњ‡+°
0015F80570  CA A6 7D 68 E7 D7 A2 A7 6A 1B 6B 9D A5 50 33 72   K¦}hзЧЎ§j.kќЃP3r
0015F80580  ED 2F 60 2F DA B3 15 0A BA B7 17 B5 E1 EE 5E 9E   н/`/Ђi..є·.µбо^ћ
0015F80590  26 59 68 14 4B 8B 27 96 A6 3F E4 FA F9 32 50 96   &Yh.K‹'-¦?дъШ2P-
0015F805A0  AC D8 64 88 BB DD 07 AB 05 77 14 3D 64 4E 2F FB   ¬Шd€»Э.«.w.=dN/ы
0015F805B0  75 1B 1D 72 CC E9 6E 38 12 C8 EB D1 63 4B 81 C0   u..rMйn8.ИлCcKЃA
0015F805C0  16 D4 7D 0B AD C4 9B FF E6 19 07 F7 A0 EB AD 6B   .ф}..Д›яЖ..ч л.k
0015F805D0  A2 4A DF B3 7C A7 29 FC 1D D6 B4 BF 9D 34 ED 72   ЎJЯi|§)ь.Цґїќ4нr
0015F805E0  AE C8 06 64 CB 5C BB 4C BA EA D9 52 3D 39 80 C8   ®И.dЛ\»LєкЩR=9ЂИ
0015F805F0  27 A2 C5 4B 73 37 44 31 00 AE AB 4C 77 F2 04 37   'ЎEKs7D1.®«Lwт.7
0015F80600  4E D6 80 DC E9 A3 20 CF AD 98 3E 63 5F 4F C1 F3   NЦЂЊйJ П.˜>c_OБУ
0015F80610  50 A3 A4 D7 96 C4 1E 8C A6 20 A6 00 0B 75 79 3C   PJ¤Ч-Д.Ѣ¦ ¦..uy<
0015F80620  1A 90 30 FA 68 8F 11 4A 89 F1 60 6C 3D D5 5E 1B   .ђ0ъhЏ.J‰c`l=X^.
0015F80630  55 19 51 B5 BC 20 B9 42 F1 87 3E 15 B0 26 78 AE   U.Qµj №Bc‡>.°&x®
0015F80640  1A 5C 43 5C 68 7D E6 18 6C 8C C5 6A B5 39 53 F9   .\C\h}Ж.lѢEjµ9Sщ
0015F80650  7B 50 70 95 8B 64 71 C0 74 F5 15 57 74 50 CB 62   {Pp•‹dqAtx.WtPЛь
0015F80660  3F B8 5F 65 8B 06 50 A8 28 2C 81 07 E9 89 6A 3A   ?ё_e‹.PЁ(,Ѓ.й‰j:
0015F80670  AD 93 B2 98 48 8C 36 F6 9C B9 AF 44 2F 29 2F 3C   ."I˜ HѢ6цѢ№ЇD/)/<
0015F80680  8A 46 0A 43 CC BE 47 5F BC 5D DA D0 5F 19 83 EF   ЉF.CMsG_j]ЂP_.ѓп
0015F80690  9B B7 8D 84 5B 25 B8 40 27 71 F4 17 81 82 9A C9   ›·K„[‰ё@'qф.Ѓ,љЙ
0015F806A0  53 9B 5A 38 2C 4C 1F 34 EC 19 D5 AD 87 7C 99 AD   S›Z8,L.4M.X.‡|™.
0015F806B0  08 E8 59 E6 EB F0 A5 45 96 08 6D 2A A6 D2 51 52   .иYЖлрҐE-.m*¦TQR
0015F806C0  EE 08 DC 3A 92 7C 3D 18 46 C6 30 6A BE B5 A8 EE   o.Ь:'|=.FЖ0jsµЁo
0015F806D0  B8 A7 75 59 91 6F E2 79 99 C4 A5 7B 12 FC 88 A5   ё§uY'овy™ДҐ{.ь€Ґ
0015F806E0  F4 86 3C C3 46 EC 7A 9F BE BE 24 15 AF 64 82 5C   ф†<ГFMzµss$.Їd,\
0015F806F0  5D 75 E9 EA 20 29 94 CB 3E 9A 24 E6 33 4D 02 A0   ]uйк )”Л>љ$Ж3M.
0015F80700  E0 21 90 B5 89 28 B6 B0 BD 22 6F B4 3B 85 0F B1   a!ђµ‰(¶°S"oґ;….±
0015F80710  52 41 31 69 03 38 07 6D 91 51 66 8D 7C 4B F7 2F   RA1i.8.m'QfЌ|Kч/
0015F80720  7F 6D 20 41 52 23 10 01 5F E1 93 3D 0C 88 C7 D5    m AR‡.._б“=.€SX
0015F80730  EE 60 1A D2 BC E7 46 1F E6 B6 3E 42 99 41 D6 3B   o`.TjзF.Ж¶>B™AЦ;
0015F80740  2B 5D 7F 0E 6A 52 6B 7B 9C 03 33 8B 96 86 92 91   +] .jRk{Ѣ.3‹–†'`
0015F80750  88 B0 51 FC BC C6 80 11 F1 7E 45 BC 42 98 2A AD   €°QьjЖЂ.c~EjB˜*.
0015F80760  5E AA 45 2D FF D9 F6 88 77 AE A1 AB F7 80 46 D8   ^ЄE-яЩц€w®Ў«чЂFШ
0015F80770  19 EB 29 73 99 16 FF 5C 85 C8 4E 0F 56 00 69 54   .л)s™.я\…ИN.V.iT
0015F80780  11 60 C3 04 60 67 B1 2E 49 D8 A2 1C 9D AF B7 B3   .`Г.`g±.IШЎ.ќЇ·i
0015F80790  00 AF C0 37 12 73 1C 65 BF 65 24 3F E5 93 82 57   .ЇA7.s.eїe$?e“,W
0015F807A0  D1 8E FE 29 D7 2C 58 D1 35 09 9D 0B 02 FD 2E C6   CЋю)Ч,XC5.ќ..э.Ж
0015F807B0  4C 62 57 7E B7 4B 10 7D 57 08 58 22 A9 20 1D F7   LbW~·K.}W.X"© .ч
0015F807C0  BF A9 57 99 E5 63 75 05 34 F3 5B 2F 5F C0 55 33   ї©W™ecu.4У[/_AU3
0015F807D0  1B 40 BC 2B 35 F9 E6 44 C9 30 7C 3C 73 3D A3 B1   .@j+5шЖDЙ0|<s=J±
0015F807E0  72 EC 44 1B 4A 5B AB 47 FB 09 13 F4 B9 D3 4C C6   rMD.J[«GЫ..ф№УLЖ
0015F807F0  E2 30 99 22 9C 28 12 05 F3 0C 06 38 6E C0 02 1A   в0™"љ(..У..8nA..
0015F80800  30 B0 00 00 CF 08 2C F5 59 8F 8C 6F A8 CF 87 54   0°..П.,xYЏѢoЁП‡T
```

Byte position: 1049; Row: 39654; Address: 365452313; Value: 1

Address: 368576512 Selected: 4

Last active selection: address 368576512 selected 4

Toolbar with parameters of Bitmap mode is at in the upper part of the window.

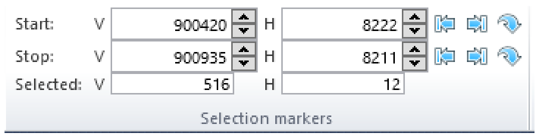
The content of dump in the visual binary representation is at the central part.

In the lower part of the window there is information about the current selection.

In Bitmap mode the pages are represented horizontally, the blocks vertically. This image representation shows the real physical structure of data in flash memory. The Bitmap mode is active, different measurements can be done there.

By pressing the mouse in the central part of the window where the binary content is, the selection tool is called, that allows to perform binary measurements. Pressing the left mouse button sets selection marker of beginning (red), pressing the right mouse button sets the end marker (violet). Measurements are made simultaneously - horizontally (inside of the page) and vertically (inside the block).

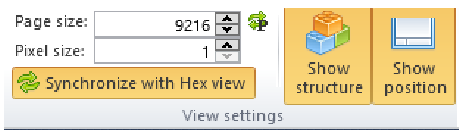
In the toolbar of Bitmap the selection markers show parameters of active selection.


Start: V 900420 H 8222
Stop: V 900935 H 8211
Selected: V 516 H 12
Selection markers

Start V – red horizontal marker-line. It describes vertical beginning-coordinate of area from the beginning of dump.
Stop V – violet horizontal marker-line. It describes vertical end-coordinate of area from the beginning of dump.
Selected V describes the size of selected vertical area in pages (horizontal lines). This marker is used for determination of the virtual block size.

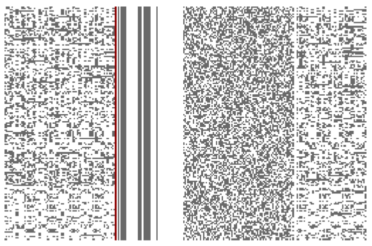
Start H – red vertical marker-line. It describes horizontal beginning-coordinate of area from the beginning of page.
Stop H – violet vertical marker-line. It describes horizontal end-coordinate of area from the beginning of page.
Selected H describes the size of selected horizontal area in bits (pixels within same line). This marker is used for analysis of page structure, determination of data and spare area size, bad columns analysis.

To use the Bitmap mode it's necessary to set correct page size (according to NAND configuration). In case if page size is set incorrectly, it shows garbage.

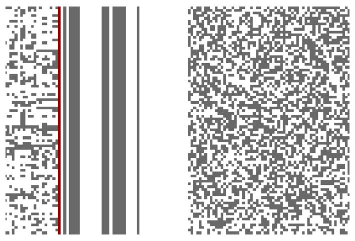

Page size: 9216
Pixel size: 1

Pixel size determines how many pixels used to display one bit.

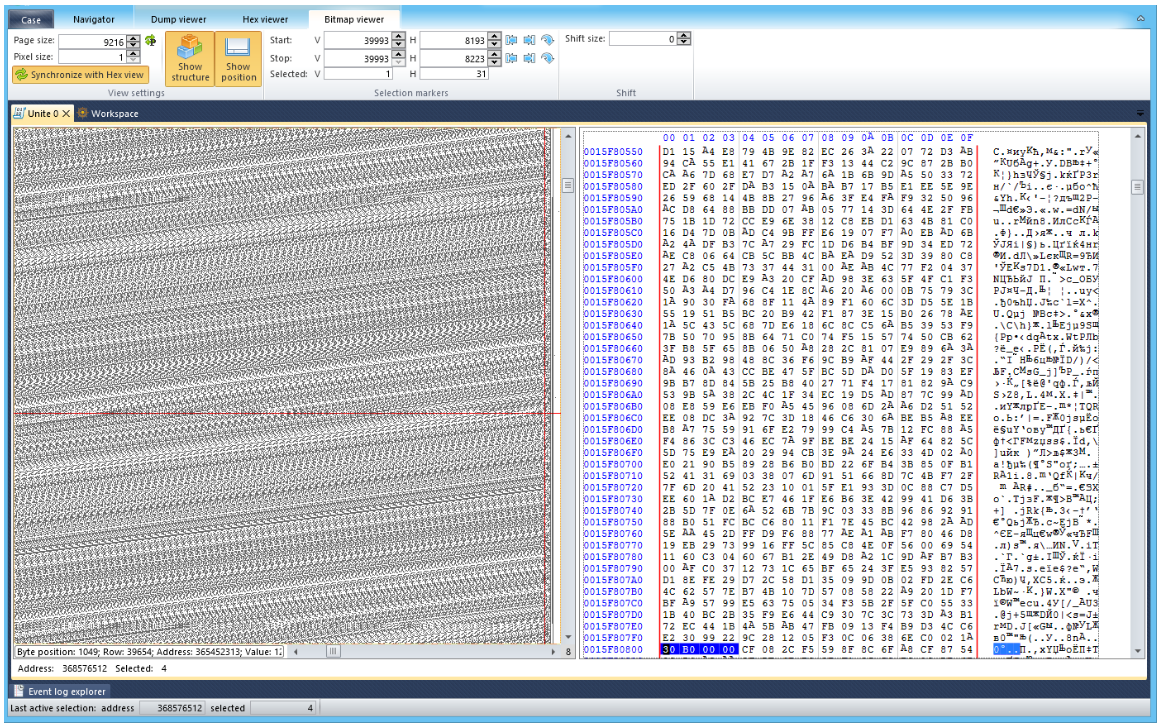
Set pixel size = 1 for analysis of page structure, virtual block size and bit errors.



Set pixel size = 2 for spare area and bad columns analysis.



Synchronize with Hex View option allows to navigate through the data in two modes simultaneously – Hex View and Bitmap View.



Page size: 9216
Pixel size: 1
Synchronize with Hex view
Start: V 39993 H 8193
Stop: V 39993 H 8223
Selected: V 1 H 31
Shift size: 0

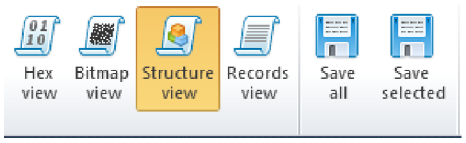
```
            00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0015F80550  D1 15 A4 E8 79 4B 9E 82 EC 26 3A 22 07 72 D3 AB
0015F80560  94 CA 55 E1 41 67 2B 1F F3 13 44 C2 9C 87 2B B0
0015F80570  CA A6 7D 68 E7 D7 A2 A7 6A 1B 6B 9D A5 50 33 72
0015F80580  ED 2F 60 2F DA B3 15 0A BA B7 17 B5 E1 EE 5E 9E
0015F80590  26 59 68 14 4B 8B 27 96 A6 3F E4 FA F9 32 50 96
0015F805A0  AC D8 64 88 BB DD 07 AB 05 77 14 3D 64 4E 2F FB
0015F805B0  75 1B 1D 72 CC E9 6E 38 12 C8 EB D1 63 4B 81 C0
0015F805C0  16 D4 7D 0B AD C4 9B FF E6 19 07 F7 A0 EB AD 6B
0015F805D0  A2 4A DF B3 7C A7 29 FC 1D D6 B4 BF 9D 34 ED 72
0015F805E0  AE C8 06 64 CB 5C BB 4C BA EA D9 52 3D 39 80 C8
0015F805F0  27 A2 C5 4B 73 37 44 31 00 AE AB 4C 77 F2 04 37
0015F80600  4E D6 80 DC E9 A3 20 CF AD 98 3E 63 5F 4F C1 F3
0015F80610  50 A3 A4 D7 96 C4 1E 8C A6 20 A6 00 0B 75 79 3C
0015F80620  1A 90 30 FA 68 8F 11 4A 89 F1 60 6C 3D D5 5E 1B
0015F80630  55 19 51 B5 BC 20 B9 42 F1 87 3E 15 B0 26 78 AE
0015F80640  1A 5C 43 5C 68 7D E6 18 6C 8C C5 6A B5 39 53 F9
0015F80650  7B 50 70 95 8B 64 71 C0 74 F5 15 57 74 50 CB 62
0015F80660  3F B8 5F 65 8B 06 50 A8 28 2C 81 07 E9 89 6A 3A
0015F80670  AD 93 B2 98 48 8C 36 F6 9C B9 AF 44 2F 29 2F 3C
0015F80680  8A 46 0A 43 CC BE 47 5F BC 5D DA D0 5F 19 83 EF
0015F80690  9B B7 8D 84 5B 25 B8 40 27 71 F4 17 81 82 9A C9
0015F806A0  53 9B 5A 38 2C 4C 1F 34 EC 19 D5 AD 87 7C 99 AD
0015F806B0  08 E8 59 E6 EB F0 A5 45 96 08 6D 2A A6 D2 51 52
0015F806C0  EE 08 DC 3A 92 7C 3D 18 46 C6 30 6A BE B5 A8 EE
0015F806D0  B8 A7 75 59 91 6F E2 79 99 C4 A5 7B 12 FC 88 A5
0015F806E0  F4 86 3C C3 46 EC 7A 9F BE BE 24 15 AF 64 82 5C
0015F806F0  5D 75 E9 EA 20 29 94 CB 3E 9A 24 E6 33 4D 02 A0
0015F80700  E0 21 90 B5 89 28 B6 B0 BD 22 6F B4 3B 85 0F B1
0015F80710  52 41 31 69 03 38 07 6D 91 51 66 8D 7C 4B F7 2F
0015F80720  7F 6D 20 41 52 23 10 01 5F E1 93 3D 0C 88 C7 D5
0015F80730  EE 60 1A D2 BC E7 46 1F E6 B6 3E 42 99 41 D6 3B
0015F80740  2B 5D 7F 0E 6A 52 6B 7B 9C 03 33 8B 96 86 92 91
0015F80750  88 B0 51 FC BC C6 80 11 F1 7E 45 BC 42 98 2A AD
0015F80760  5E AA 45 2D FF D9 F6 88 77 AE A1 AB F7 80 46 D8
0015F80770  19 EB 29 73 99 16 FF 5C 85 C8 4E 0F 56 00 69 54
0015F80780  11 60 C3 04 60 67 B1 2E 49 D8 A2 1C 9D AF B7 B3
0015F80790  00 AF C0 37 12 73 1C 65 BF 65 24 3F E5 93 82 57
0015F807A0  D1 8E FE 29 D7 2C 58 D1 35 09 9D 0B 02 FD 2E C6
0015F807B0  4C 62 57 7E B7 4B 10 7D 57 08 58 22 A9 20 1D F7
0015F807C0  BF A9 57 99 E5 63 75 05 34 F3 5B 2F 5F C0 55 33
0015F807D0  1B 40 BC 2B 35 F9 E6 44 C9 30 7C 3C 73 3D A3 B1
0015F807E0  72 EC 44 1B 4A 5B AB 47 FB 09 13 F4 B9 D3 4C C6
0015F807F0  E2 30 99 22 9C 28 12 05 F3 0C 06 38 6E C0 02 1A
0015F80800  30 B0 00 00 CF 08 2C F5 59 8F 8C 6F A8 CF 87 54
```

Byte position: 1049; Row: 39654; Address: 365452313; Value: 12

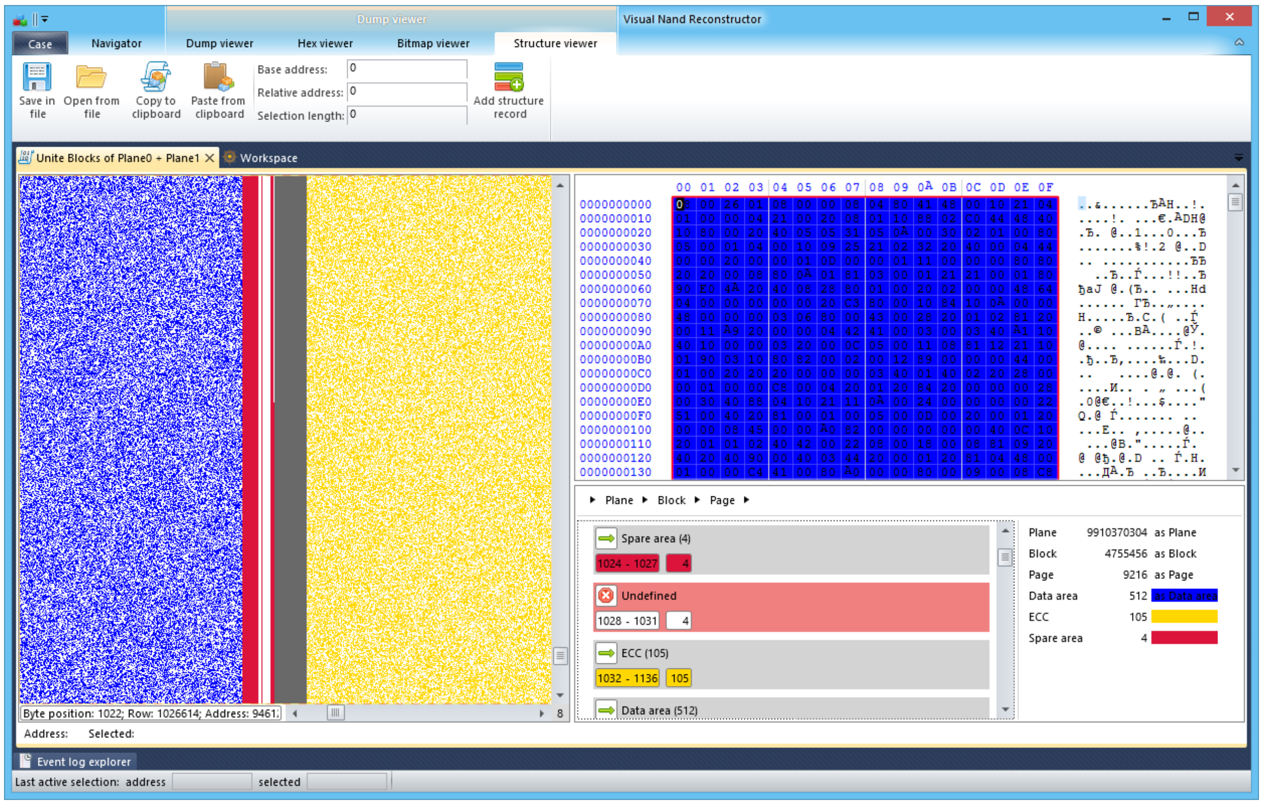
Address: 368576512 Selected: 4

Last active selection: address 368576512 selected 4

Different patterns of NAND chips and Bitmap usage can be found in the web article:
http://rusolut.com/binary-patterns-in-nand-flash-memory/



The Structure Viewer is a tool for the physical image structure description and visualization. Analysis and correct description of dump structure is the first and important step in the process of logical image reconstruction. To analyze the structure the Bitmap Viewer is used, and for describing structures and their sizes – Structure Viewer. All modes can be opened simultaneously in one window from Dump Viewer.



Base address: 0
Relative address: 0
Selection length: 0

```
            00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0000000000  08 00 26 01 08 00 00 08 04 80 41 48 00 10 21 04  ..&......ЂAH..!.
0000000010  01 00 00 04 21 00 20 08 01 10 88 02 C0 44 48 40  ....!. ...€.ADH@
0000000020  10 80 00 20 40 05 05 31 05 0A 00 30 02 01 00 80  .Ђ. @..1...0...Ђ
0000000030  05 00 01 04 00 10 09 25 21 02 32 20 40 00 04 44  .......%!.2 @..D
0000000040  00 00 20 00 00 01 0D 00 00 01 11 00 00 00 80 80  .. ............ЂЂ
0000000050  20 20 00 08 80 0A 01 81 03 00 01 21 21 00 01 80    ..Ђ..Ѓ...!!..Ђ
0000000060  90 E0 4A 20 40 08 28 80 01 00 20 02 00 00 48 64  ђаJ @.(Ђ.. ...Hd
0000000070  04 00 00 00 00 00 20 C3 80 00 10 84 10 0A 00 00  ...... ГЂ..„.....
0000000080  48 00 00 00 03 06 80 00 43 00 28 20 01 02 81 20  H.....Ђ.C.( ..Ѓ
0000000090  00 11 A9 20 00 00 04 42 41 00 03 00 03 40 A1 10  ..© ...BA....@Ў.
00000000A0  40 10 00 00 03 20 00 0C 05 00 11 08 81 12 21 10  @.... ......Ѓ.!.
00000000B0  01 90 03 10 80 82 00 02 00 12 89 00 00 00 44 00  .ђ..Ђ,....‰...D.
00000000C0  01 00 20 20 20 00 00 00 03 40 01 40 02 20 28 00  ..   ....@.@. (.
00000000D0  00 01 00 00 C8 00 04 20 01 20 84 20 00 00 00 28  ....И.. . „ ...(
00000000E0  00 30 40 88 04 10 21 11 0A 00 24 00 00 00 00 22  .0@€..!...$...."
00000000F0  51 00 40 20 81 00 01 00 05 00 0D 00 20 00 01 20  Q.@ Ѓ....... ..
0000000100  00 00 08 45 00 00 A0 82 00 00 00 00 00 40 0C 10  ...E.. ,.....@..
0000000110  20 01 01 02 40 42 00 22 08 00 18 00 08 81 09 20   ...@B."....Ѓ.
0000000120  40 20 40 90 00 40 03 44 20 00 01 20 81 04 48 00  @ @ђ.@.D .. Ѓ.H.
0000000130  01 00 00 C4 41 00 80 A0 00 00 80 00 09 00 08 C8  ...ДA.Ђ ..Ђ....И
```

Plane ▸ Block ▸ Page ▸

Spare area (4)
1024 - 1027 | 4

Undefined
1028 - 1031 | 4

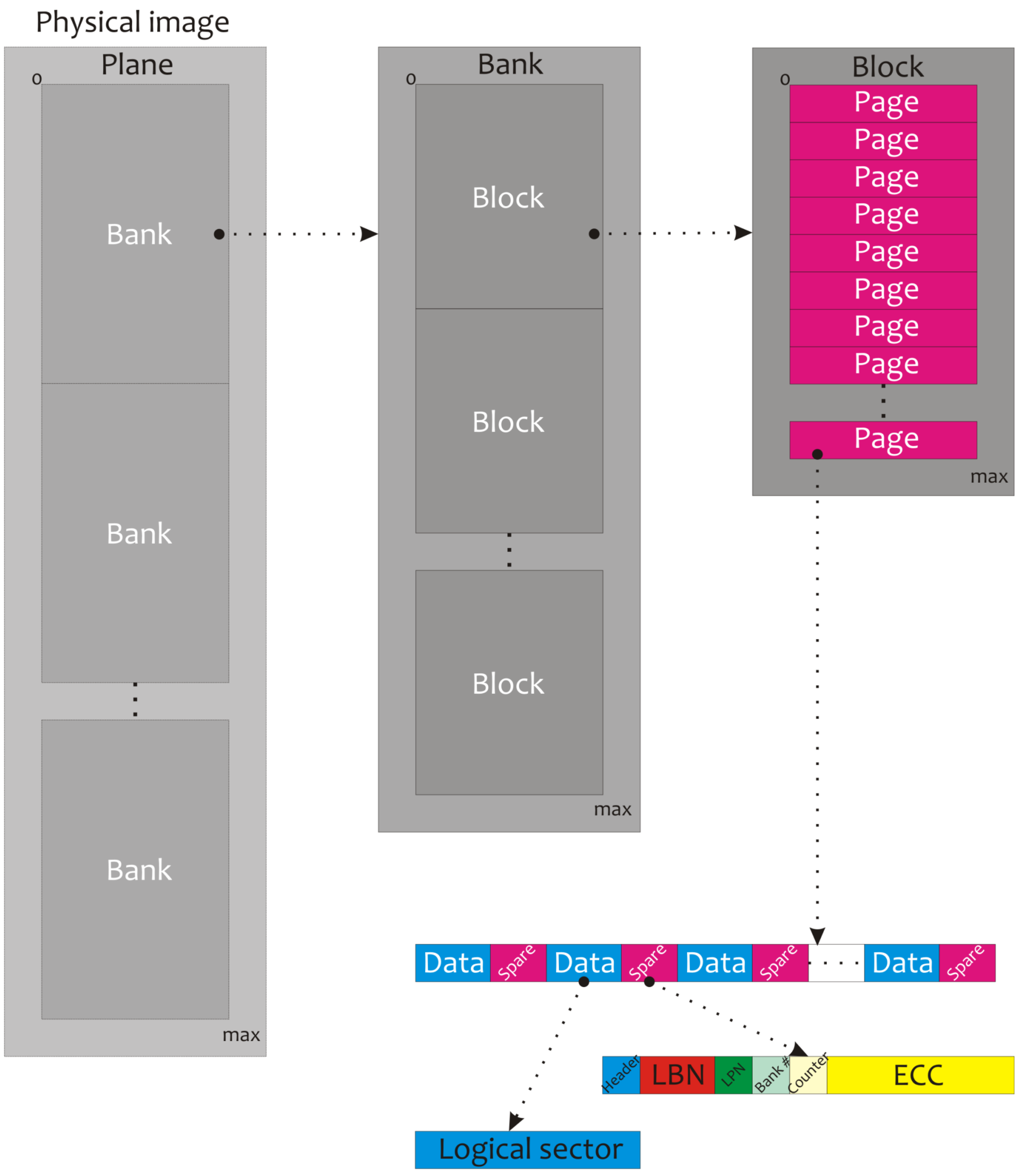
ECC (105)
1032 - 1136 | 105

Data area (512)

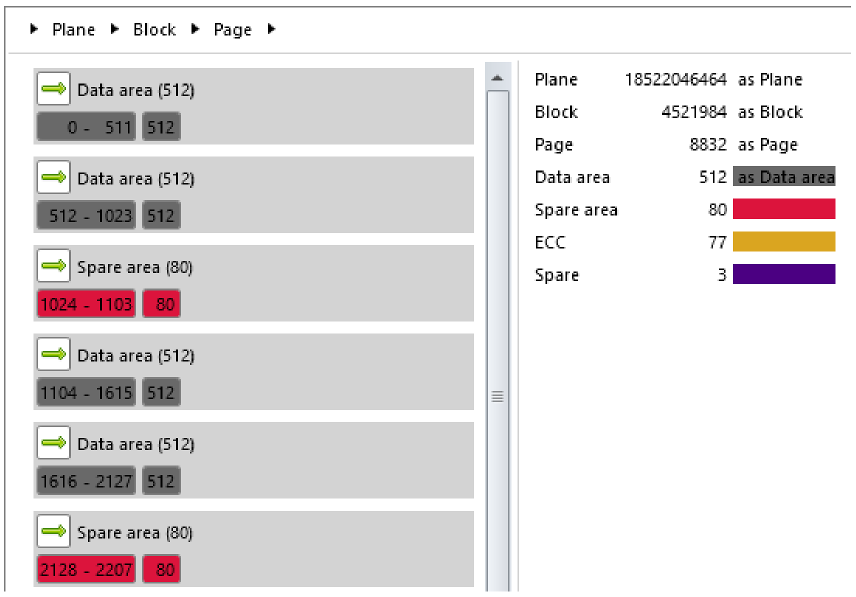
Plane	9910370304	as Plane
Block	4755456	as Block
Page	9216	as Page
Data area	512	as Data area
ECC	105	🟨
Spare area	4	🟥

Byte position: 1022; Row: 1026614; Address: 9461.
Address: Selected:
Last active selection: address selected

The physical image has multi-level structure. Each structure consists of sub-structures. Plane consists of Banks (in general, Plane = Bank). Plane/Bank consists of Blocks. Block consists of Pages. Page consists of Data area, Spare area and ECC area. Data area consists of bytes where user data (logical sectors) are stored, in modern flash devices its size is 1024b (sometimes 512b or 2048b). Spare area consists of Block header, LBN, ECC and some optional markers.



The main window of Structure Viewer consists of image structure tree (on the left) and structure library area (on the right).



▸ Plane ▸ Block ▸ Page ▸

Data area (512)
0 - 511 | 512

Data area (512)
512 - 1023 | 512

Spare area (80)
1024 - 1103 | 80

Data area (512)
1104 - 1615 | 512

Data area (512)
1616 - 2127 | 512

Spare area (80)
2128 - 2207 | 80

Plane	18522046464	as Plane
Block	4521984	as Block
Page	8832	as Page
Data area	512	as Data area
Spare area	80	🟥
ECC	77	🟨
Spare	3	🟪

When the chip is read, such structures as Plane, Block and Page are set automatically from NAND chip configuration.

The Plane is the parent dump structure and it's size equals to NAND crystal size (depending on number of planes per crystal, in general Crystal = Plane).

The Bank structure is a child stucture of Plane. Bank structure is not set by default, because most of modern controllers don't split physical space of NAND memory by banks and LBN numeration is solid. In case if the space of NAND memory is divided by banks, each bank would have numeration beginning with LBN 0000 and Plane should include the Bank structure. Number of banks usually equals power of 2 (1,2,4,8,16).

The Block structure is a child of plane/bank and it's default size is equal to the NAND chip's physical block. As the Virtual block may consist of several physical blocks, the size of this structure must be set in accordance with the virtual block size, which can be found using the Bitmap mode (1,2,4 physical blocks).

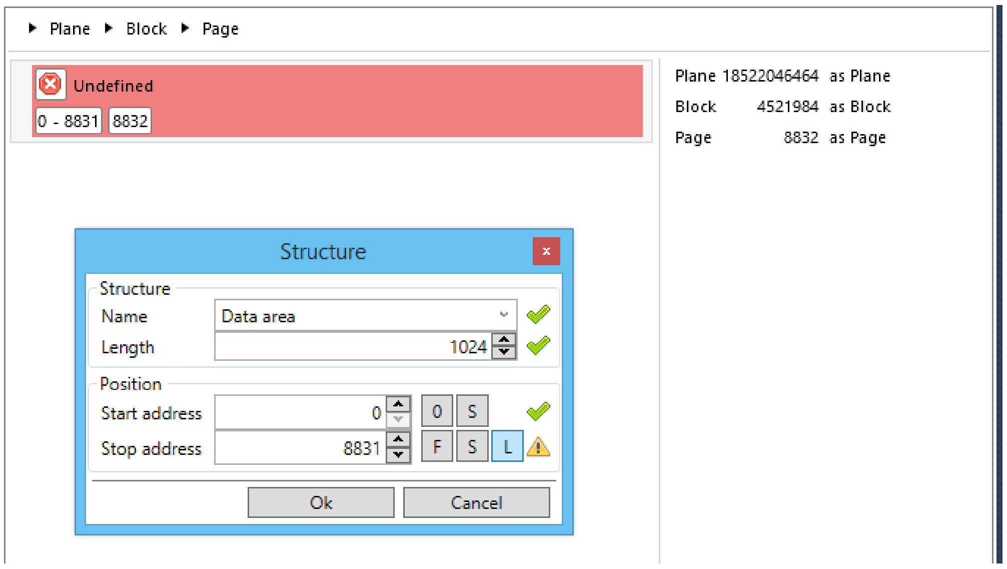
The Page structure is a child of block and it's size corresponds to physical size of page of NAND chip. This structure is defined during physical image extraction, in configuration window of NAND chip.

The Data Area structure is a child of page. The size of this structure can be found through the Bitmap mode while analysis of page structure. Usually, modern controllers use the Data area size equal to 1024 bytes (sometimes 512 or 2048 bytes). To set this structure, it must be manually "set as Data area" at structure library tab.

The Spare Area and ECC structures are children of page and set inside the page. Their size depends on page structure.

The LBN, Header and other structures are children of Spare Area and set with the aim of pointing parameters to the Markers Table for block arrangement, filtration and sorting for further logical image creation.

For each structure any color from the palette can be set, for visual simplification during analysis. It highlights structures in Hex and Bitmap modes. Every structure has beginning, end and size. To set any structure, it's necessary to determine it's parameters in Bitmap viewer.



Plane 18522046464 as Plane
Block 4521984 as Block
Page 8832 as Page

Structure
Name: Data area
Length: 1024
Position
Start address: 0
Stop address: 8831

To assign the structure it's necessary to double-click on undefined area and add name and length of structure. Also it's necessary to set position of structure within the parent area, it's start and end addresses. The special buttons used to set up addresses:

0 - start from zero
S - start/stop on selection marker (Bitmap mode selection markers)
F - full area (to the end of parent area)
L - length (Stop address = Start address + Length)

When the structure is set, it appears in structure library, at the right tab. To describe the page it's necessary to set Data area and Spare area once and then choose it from dropdown list.

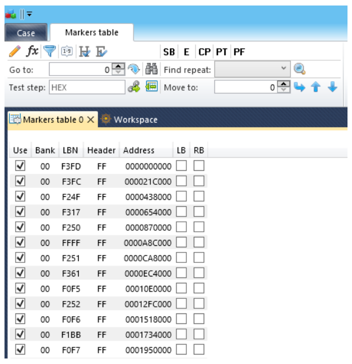
Block list functions

The group Block list functions includes the tool for block sorting and filtering for logical image creation.

Show table

List cr...

The Show Table function is used for block table displaying, analysis, filtration and sorting in accordance with LBN, Header etc.



Use	Bank	LBN	Header	Address	LB	RB
✓	00	F3FD	FF	0000000000	☐	☐
✓	00	F3FC	FF	000021C000	☐	☐
✓	00	F24F	FF	0000438000	☐	☐
✓	00	F317	FF	0000654000	☐	☐
✓	00	F250	FF	0000870000	☐	☐
✓	00	FFFF	FF	0000A8C000	☐	☐
✓	00	F251	FF	0000CA8000	☐	☐
✓	00	F361	FF	0000EC4000	☐	☐
✓	00	F0F5	FF	00010E0000	☐	☐
✓	00	F252	FF	00012FC000	☐	☐
✓	00	F0F6	FF	0001518000	☐	☐
✓	00	F1BB	FF	0001734000	☐	☐
✓	00	F0F7	FF	0001950000	☐	☐

Toolbar of block table has set of operations for block management algorithms.

Edit markers is used for appying inversion and mask to spare area markers.
Typical mask for LBN marker: not used; 0FFF; 07FF, 03FF.
Typical mask for Header: not used; F0.

Block filter is used for block filtering that won't transition into logical image (these blocks don't store user's data). The special syntax is used in block filter.

To filter by LBN range it's necessary to use syntax:
(xxxx-yyyy)
where xxxx is first block in sequence, yyyy is last block in sequence.
E.g. (1000-13FF).

To filter by Header it's necessary to use syntax:
xx/yy/zz
where xx,yy,zz are byte values that used in header to mark user's data blocks. E.g. 20/30.

The blocks can also be filtered manually, using flag ☑ near LBN.

Block sorting is used for block reordering according to their LBN in increasing order. In case when block addressing is independent for banks, blocks must be sorted by bank and by LBN inside each bank. This operation allows to set very flexible sorting rules for any controller configurations.

Find repeat is used for search of duplicated blocks that must be disabled before virtual to logical image transition. The duplicated blocks bring shifts into file system and file structure. They may be filtered by header, or manually by disabling flag ☑ near the block.

Test step is used for integrity control of the LBN sequence and analysis of places where sequence interrupts (Lost blocks). The test step should be set to 1/1 by default. The lost (missing) blocks bring shifts into file system and file structure. The dummy blocks must be added instead of lost blocks (click right button on block table).

When blocks re-ordered, the logical image can be created via Markers table element.

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/