black hat EUROPE 2020
DECEMBER 9–10
BRIEFINGS

A New Hope: The One Last Chance to Save Your SSD Data

About us

- **Kwonyoup Kim, Founder, CEO of SNT Works Inc.**
- **Seungjoon Lee, Senior Researcher of SNT Works Inc.**

- **SNT Works Inc.**
  - **Specialized in Security Analysis on Embedded Systems**
    - ✓ **Reverse Engineering for Security Evaluation, Assessment**
    - ✓ **Offensive Security for Embedded Devices**
    - ✓ **Digital Forensics for Embedded Devices**
    - ✓ **Side Channel Attack & Fault Injection Attack with Reverse Engineering**
    - ✓ **Patent infringement Investigation**

SNTWORKS

Introduction

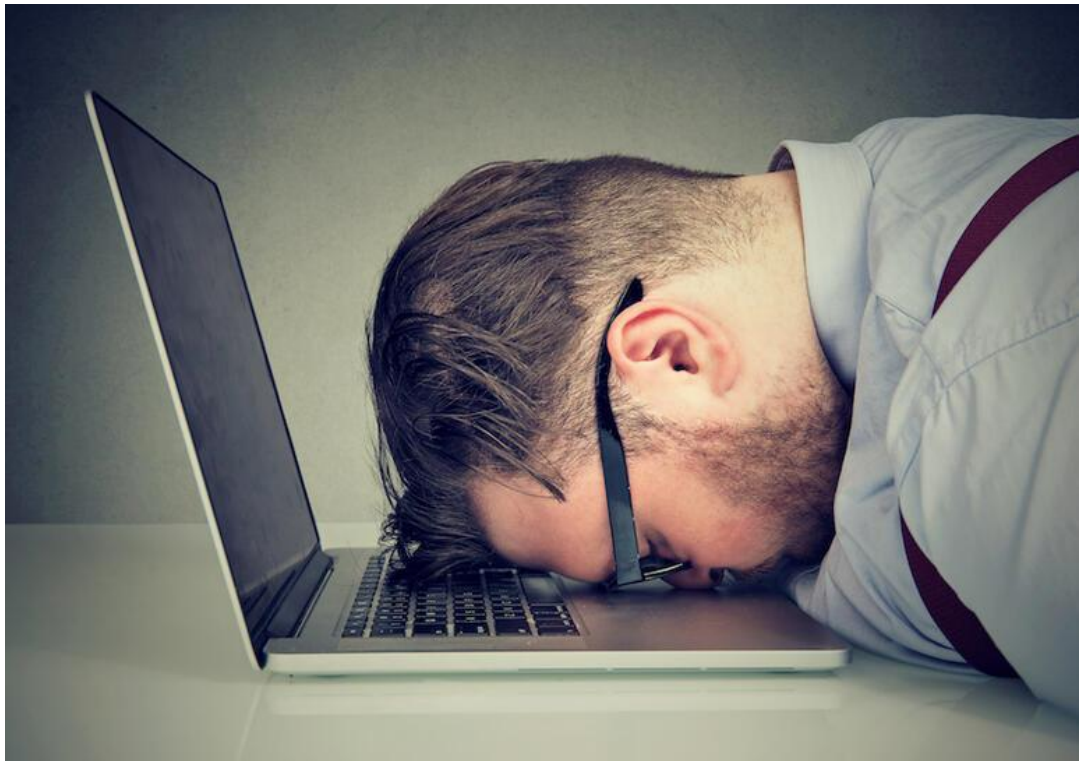
SSD Forensics : Myths and Reality

SSD Forensics : Myths and Reality

- **SSD Forensics Issues : Perfectly(Secure) Erase**
  - **Can SSD data be completely erased?**

SSD Forensics : Myths and Reality

- **SSD Forensics : Cannot Recovery Data**
  - **Can't we recover previous data from SSDs?**



SSD Forensics : Myths and Reality

- **Current status of SSD Forensics**
  - **SSD Controller information is not disclosed**
  - **Private SSD internal algorithm information**
  - **Lack of SSD Physical Data collection tools (including data reconstruction function)**
  - **Lack of SSD Physical Data recovery tools**

- **Our Challenges**
  - **Build an environment to analyze the internal operation of Commercial SSDs**
  - **Tracing the process of writing, reading and erasing SSD data**
  - **SSD physical data collection**
  - **Check and try SSD data recovery conditions**

SSD Firmware Reversing

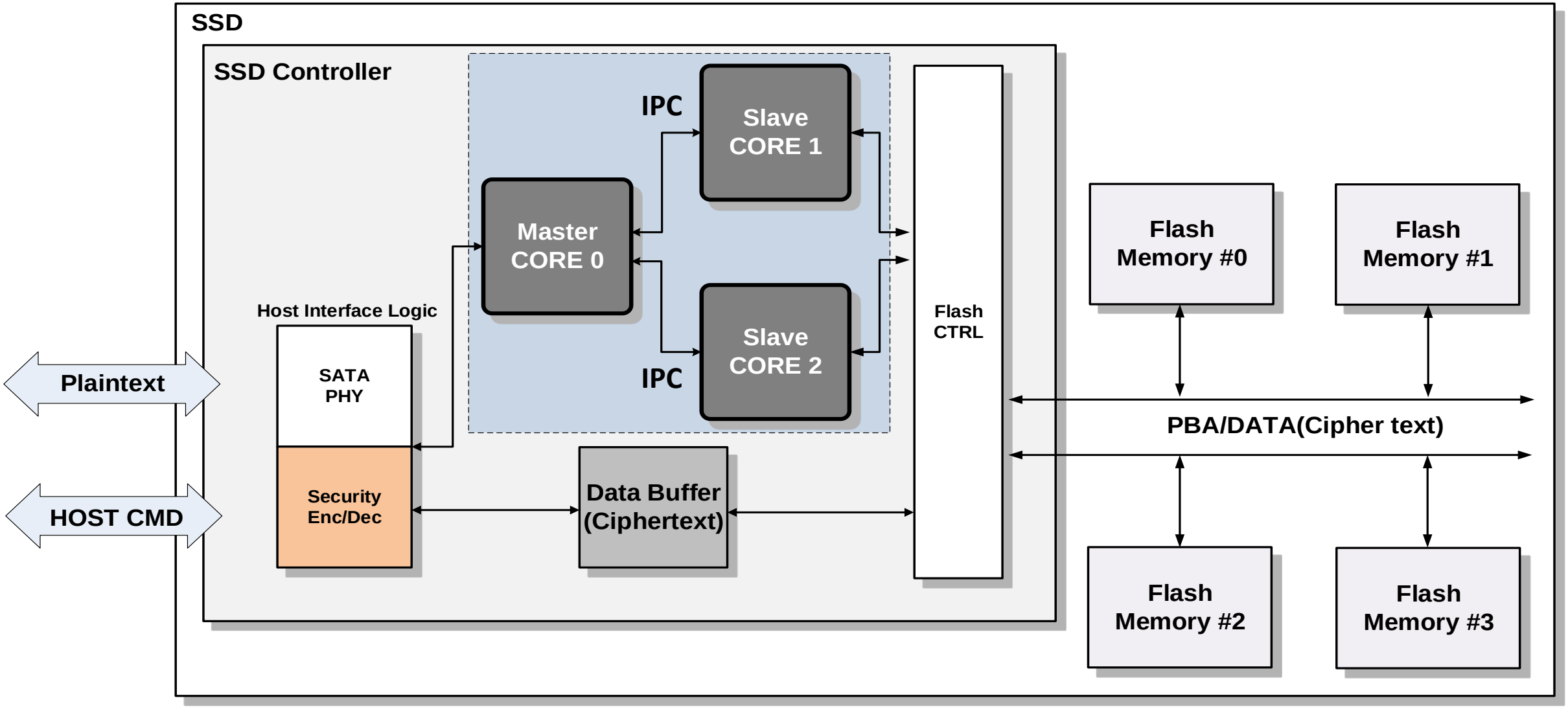
Background

Why Commercial SSDs Reversing is Hard?

- **SSD Manufacturers do not divulge information.**
- **SSD Controller manufacturers also do not divulge information.**
- **They protect their implementation details to protect their technologies.**
- **No standards**

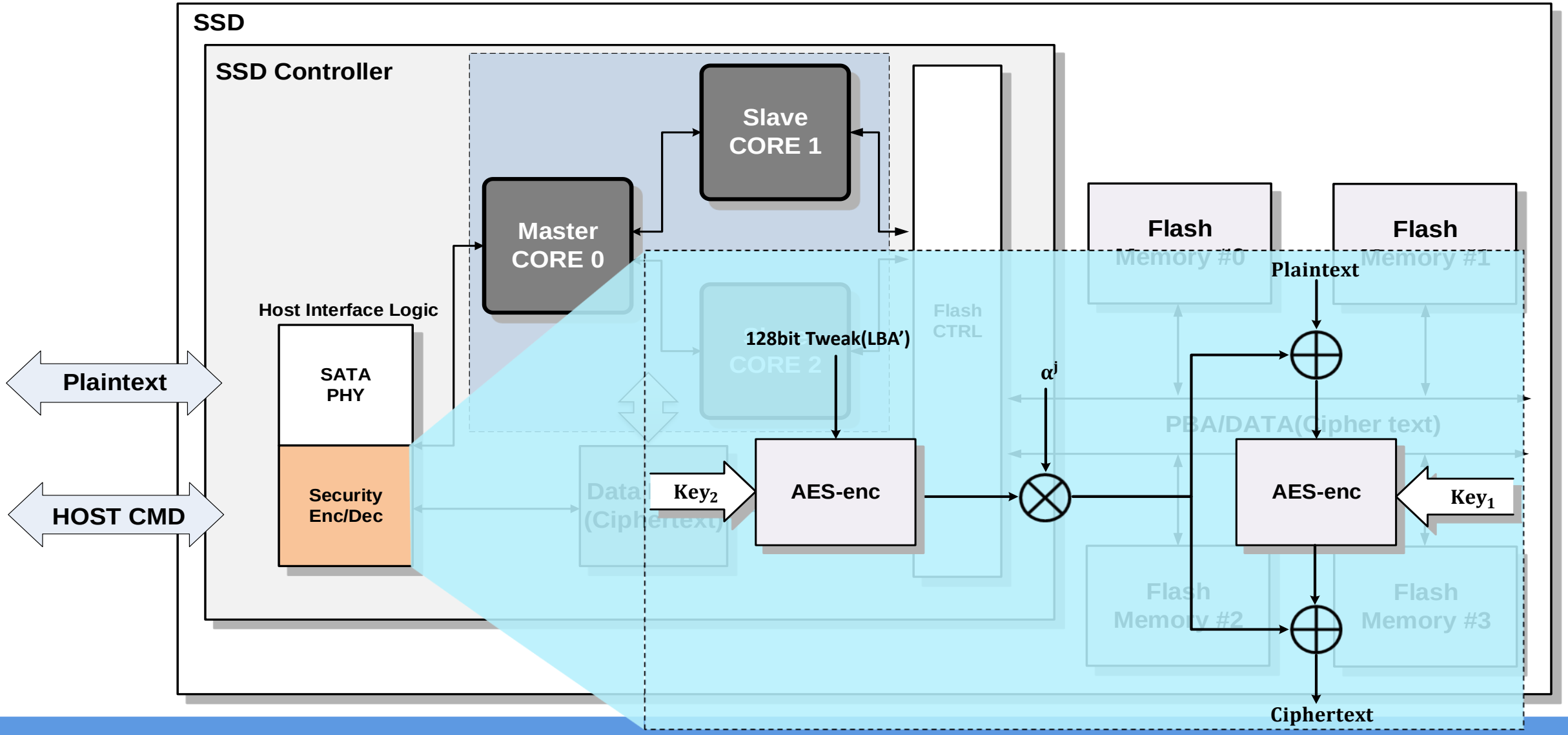
SSD Hardware Architecture

- **Multi Core based SSD Controller**

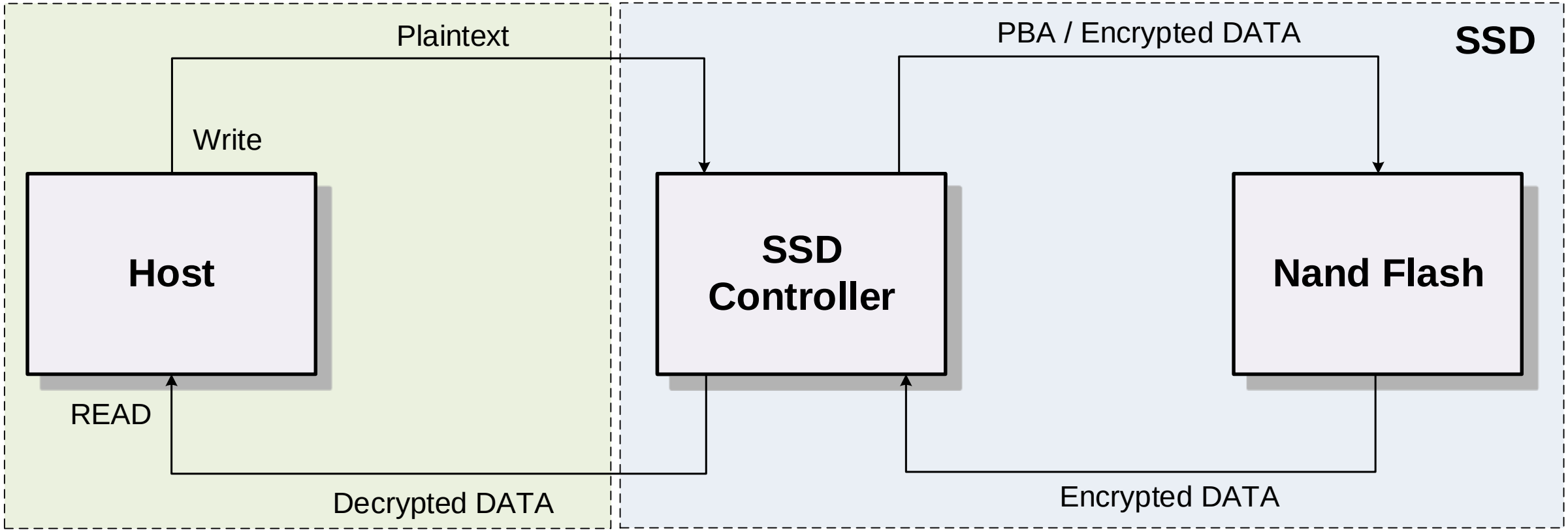


SSD Hardware Architecture

- **AES-XTS**



Security Encoding and Decoding



You can Buy Now 2020

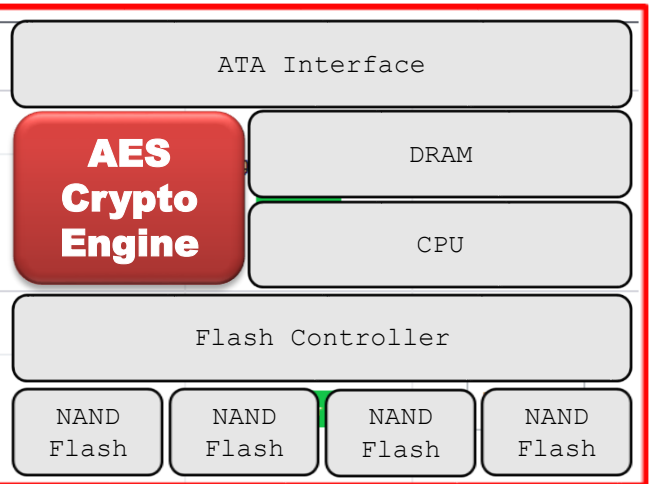
Regular shopping cart | Shopping only

shopping basket > Order/Payment > complete

- Cart items are stored for up to 30 days.
- If information such as price or options has changed, you may not be able to order.
- As for today's departure products, today's departure may change depending on the seller's setting point, so please check again when placing an order.

Security Encoding SSD

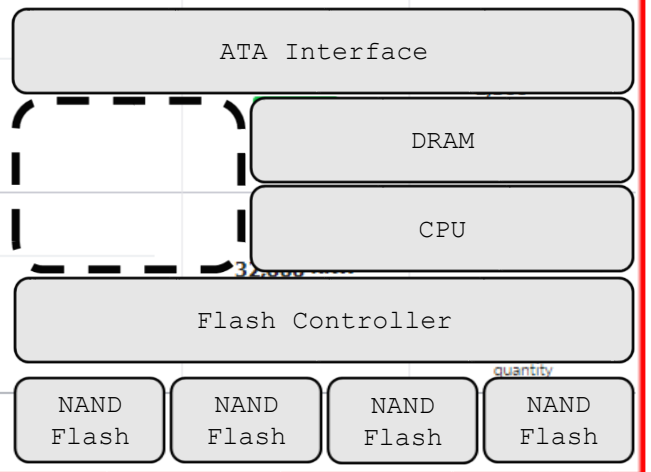
Product information	option
Reproduction System Smart Store N Pay Samsung SSD 850 PRO 128GB / MZ-7KE128B/KR Bullet Delivery 95,000 KRW 11. 12. (Thu) 90% probability of arrival	Product order quantity: 1 Add/Change order conditions
Compuzone Smart Store N Pay Micron Crucial MX500 Series 250GB TLC KRW 45,550 11. 12. (Thursday) 94% probability of arrival	Product order quantity: 1 Add/Change order conditions



Non-Security Encoding SSD

Product information	option
N Premium Shopping Mall Smart Store N Pay TeamGroup GX2 (128GB) SSD hard HOT product 22,500 won 11. 11.(Wed) 95% probability of arrival	Product order quantity: 1 Add/Change order conditions
Myssd Smart Store N Pay Review Announcement SSD 900G Blue 2.5 inch SATA 240GB Genuine Bulk Packaging Warranty 5 years 32,800 KRW ~~36,300~~ 11. 12. (Thu) 98% probability of arrival	Product order quantity: 1 Add/Change order conditions

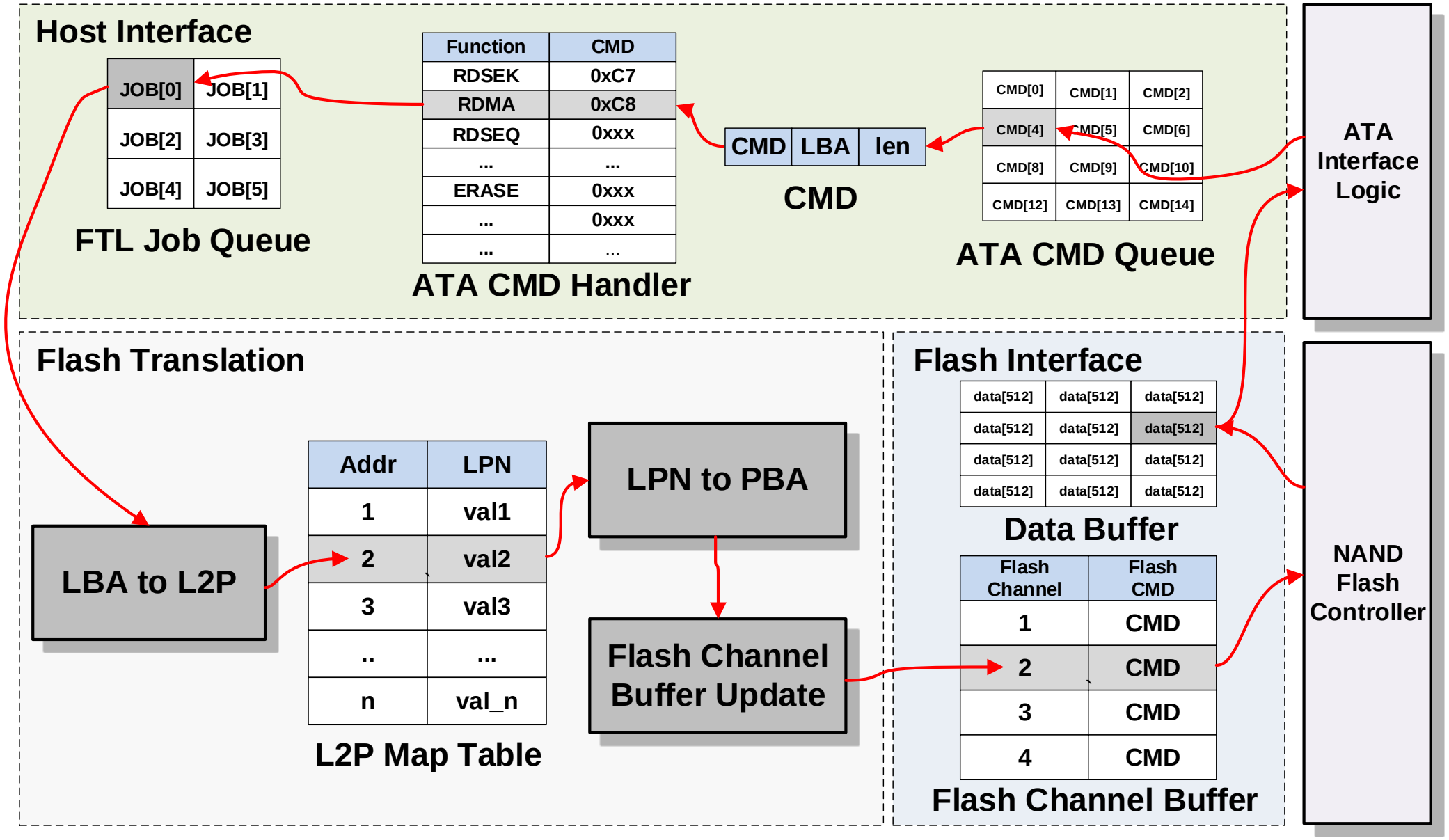
Delete selected product | Wish selected product



SSD Firmware Reversing

Access All Physical Block

SSD Read Sequence Workflow



ATA CMD Queue

- **Host Interface Layer**

ATA CMD: 0xC8(READ CMD)

Sector Size(8bit) : 0x05

LBA(24bit) : 0xE0A5A5A5

SD:20000000	00000000	00000000	00001459	00008000
SD:20000010	00000010	00000000	00000000	00004032
SD:20000020	00000000	00000410	00000F3F	00000000
SD:20000030	00000001	00000000	00000005	00000000
SD:20000040	00008027	00000C8	E0A5A5A9	00000000
SD:20000050	00000000	00000000	00000000	C708F991
SD:20000060	124F7B3F	00002000	E0A5A5A5	00000000
SD:20000070	00000005	00000000	00000000	00000A00
SD:20000080	00000000	00006739	00000000	0000011C
SD:20000090	00008EE0	00000000	00000009	808B483F
SD:200000A0	00000000	04000C00	000000B3	F9BD13F7
SD:200000B0	000513E5	000801E5	00001D81	7FA06000
SD:200000C0	0000000B	00004BFF	00000000	00000000
SD:200000D0	00010001	30002BF2	00260200	8050280D
SD:200000E0	0075320D	0D054141	B24B1030	0E390157
SD:200000F0	00000014	00000000	00000000	00000000
SD:20000100	00000000	00000000	00000000	00000000
SD:20000110	00820005	00000000	00000000	00000000
SD:20000120	3F027231	0E8E0001	03AE6260	40800C00
SD:20000130	80230001	00000000	E0A5A5A5	00000000
SD:20000140	00000000	00000000	00000000	00000000
SD:20000150	9F000000	00000000	00000000	00000000
SD:20000160	040113E5	00820000	00000000	00000000
SD:20000170	00000000	00000000	00000000	00000000
SD:20000180	00000000	000A000A	0000A5A5	00000000
SD:20000190	03020100	07060504	00000000	00000000
SD:200001A0	E0A5A5A5	C8000000	00000005	80A5A5A5
SD:200001B0	00040000	00000000	00000000	00000000

ATA CMD Handler

- **Host Interface Layer**

```
ROM:0000C50C
ROM:0000C50C
ROM:0000C50C
ROM:0000C50C                sub_C50C
ROM:0000C50C F0 5F 2D E9 PUSH            {R4-R12,LR}
ROM:0000C510 03 10 D0 E5 LDRB            R1, [R0,#3]
ROM:0000C514 8C 21 9F E5 LDR             R2, =sata_handler_table
ROM:0000C518 7C 41 9F E5 LDR             R4, =dword_1A298
ROM:0000C51C 81 11 61 E0 RSB             R1, R1, R1,LSL#3
ROM:0000C520 02 50 81 E0 ADD             R5, R1, R2
ROM:0000C524 01 10 95 E5 LDR             R1, [R5,#1]
ROM:0000C528 D4 61 C4 E1 LDRD            R6, R7, [R4,#0x14]
ROM:0000C52C 00 00 51 E3 CMP             R1, #0
ROM:0000C530 54 00 00 0A BEQ             loc_C688
```

```
ROM:0000C5C0
ROM:0000C5C0                loc_C5C0
ROM:0000C5C0 01 20 95 E5 LDR             R2, [R5,#1]
ROM:0000C5C4 B5 10 D5 E1 LDRH            R1, [R5,#5]
ROM:0000C5C8 32 FF 2F E1 BLX             R2
ROM:0000C5CC 00 00 50 E3 CMP             R0, #0
ROM:0000C5D0 01 00 A0 03 MOVEQ           R0, #1
ROM:0000C5D4 04 00 C4 05 STRBEQ          R0, [R4,#(byte_1A29C - 0x1A298)]
ROM:0000C5D8 B5 00 D5 E1 LDRH            R0, [R5,#5]
ROM:0000C5DC 01 0C 10 E3 TST             R0, #0x100
ROM:0000C5E0 14 00 00 0A BEQ             loc_C638
```

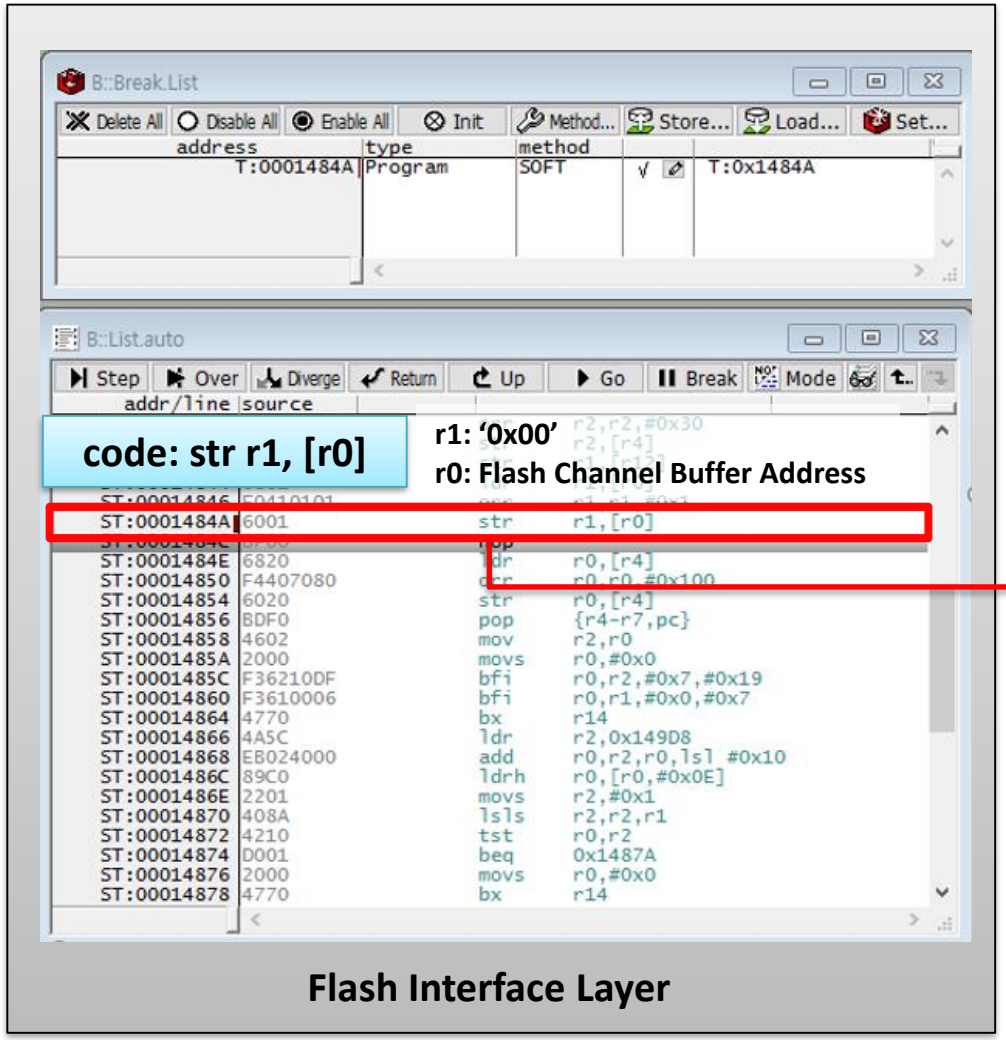
ATA CMD Handler Table

```
seg002:00808B42 00                       DCB    0
seg002:00808B43 00                       DCB    0
seg002:00808B44 C4                       DCB 0xC4
seg002:00808B45 65 0B 01 00              DCD sub_10B64+1
seg002:00808B49 07                       DCB    7
seg002:00808B4A 01                       DCB    1
seg002:00808B4B C5                       DCB 0xC5
seg002:00808B4C 78 F5 00 00              DCD sub_F578
seg002:00808B50 06                       DCB    6
seg002:00808B51 01                       DCB    1
seg002:00808B52 C6                       DCB 0xC6
seg002:00808B53 D7 08 00 80              DCD sub_800008D6+1
seg002:00808B57 00                       DCB    0
seg002:00808B58 00                       DCB    0
seg002:00808B59 00                       DCB    0
seg002:00808B5A BD CC 00 00              DCD sub_CCBC+1
seg002:00808B5E 00                       DCB    0
seg002:00808B5F 00                       DCB    0
seg002:00808B60 C8                       DCB 0xC8
seg002:00808B61 AC 03 01 00              DCD RDMA_Handler
seg002:00808B65 01                       DCB    1
seg002:00808B66 01                       DCB    1
seg002:00808B67 C9                       DCB 0xC9
seg002:00808B68 AC 03 01 00              DCD RDMA_Handler
seg002:00808B6C 01                       DCB    1
seg002:00808B6D 01                       DCB    1
seg002:00808B6E CA                       DCB 0xCA
seg002:00808B6F 78 F5 00 00              DCD sub_F578
```

: ATA Command : Handler Function

Flash Chip → Flash DATA Buffer [ ]

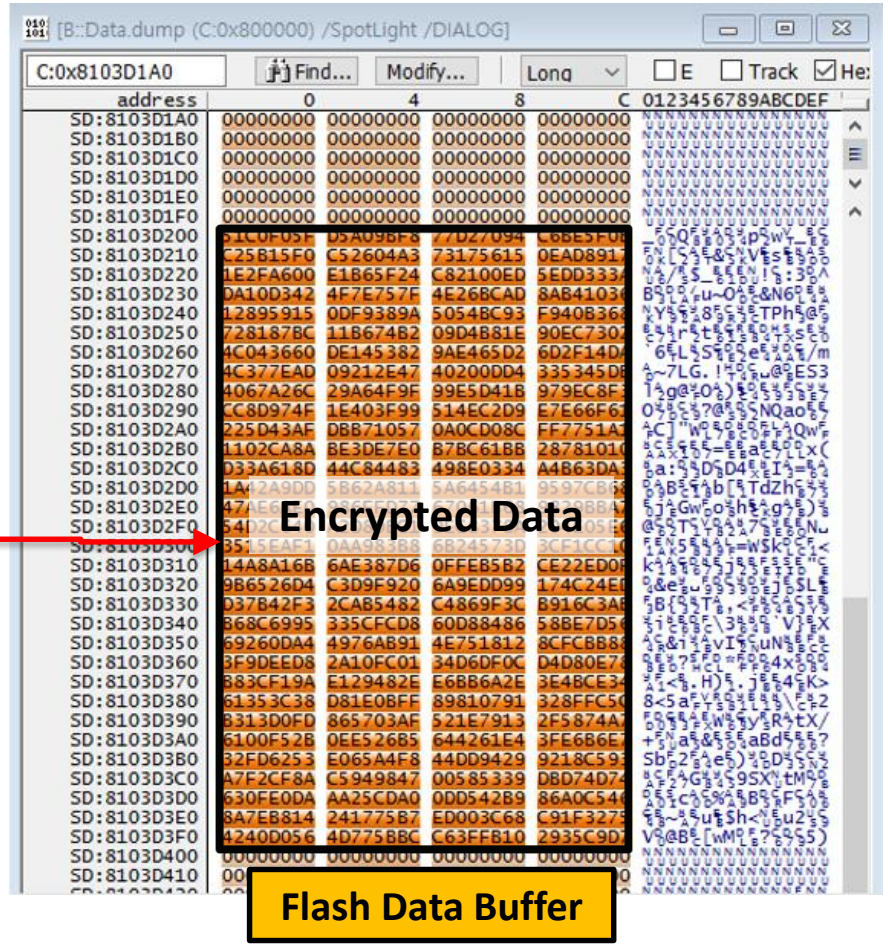
- **Flash Interface Layer → Flash Chip → Flash Interface Layer**



Flash Interface Layer

Channel	CMD
0	CMD2
1	CMD1
2	CMD3
3	CMD0

Flash Channel Buffer



Flash Data Buffer [ ] → Host Interface Logic

- **Flash Interface Layer → Host Interface Layer → Host PC**

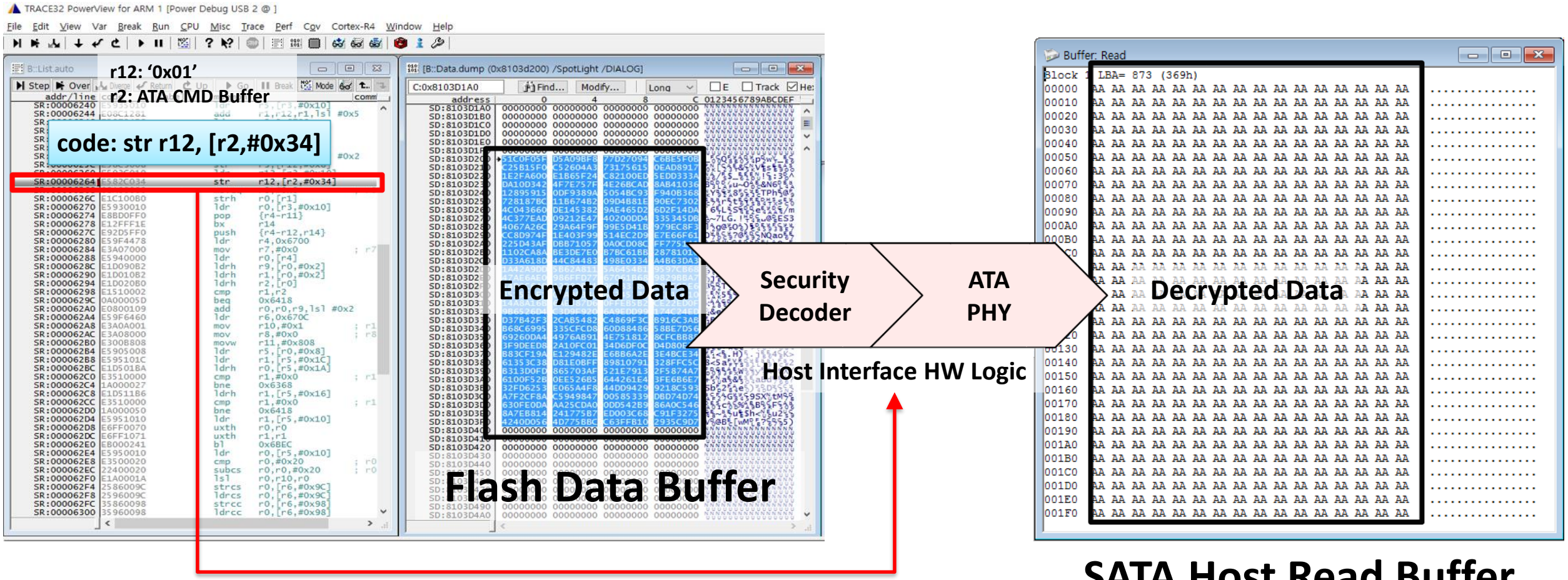
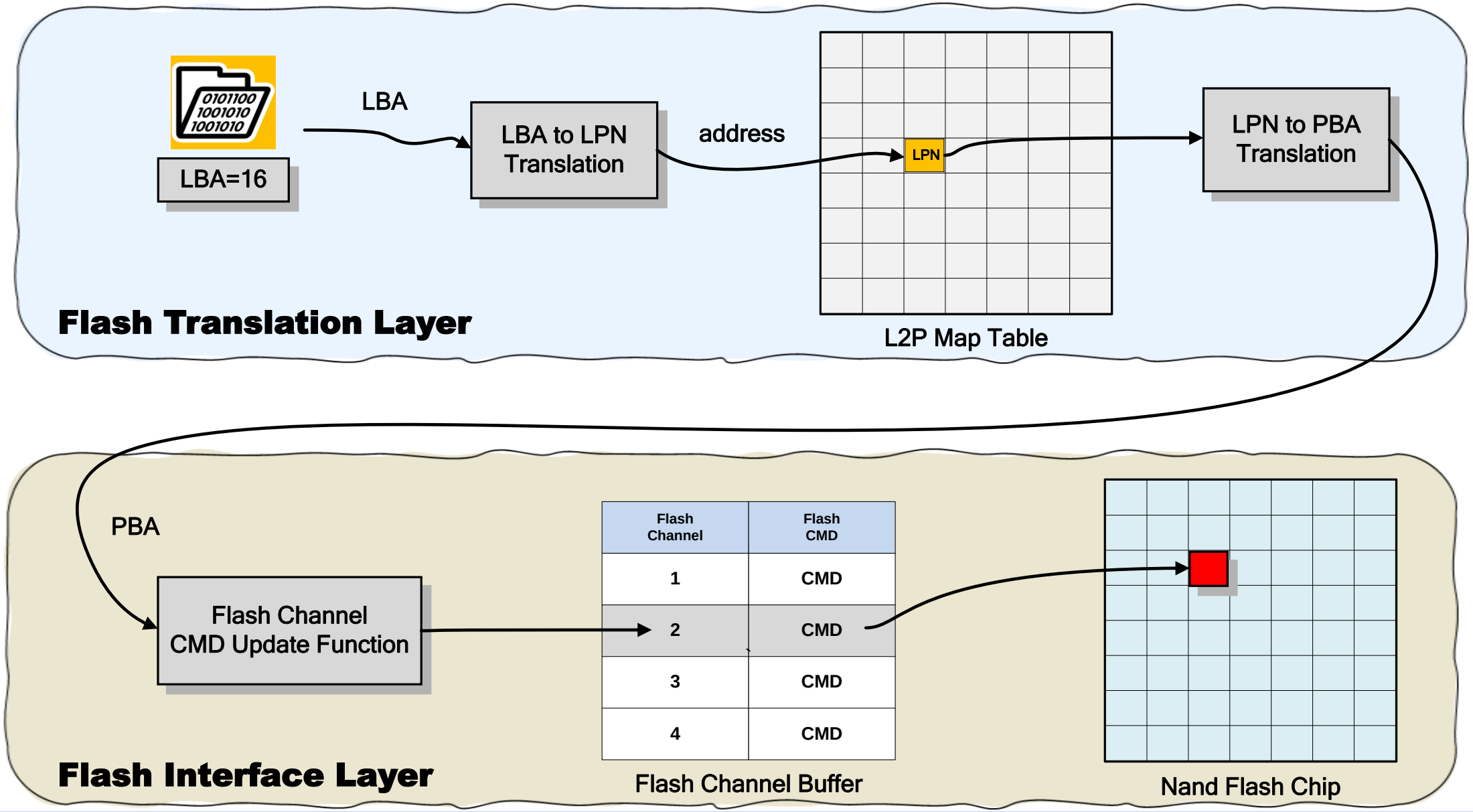
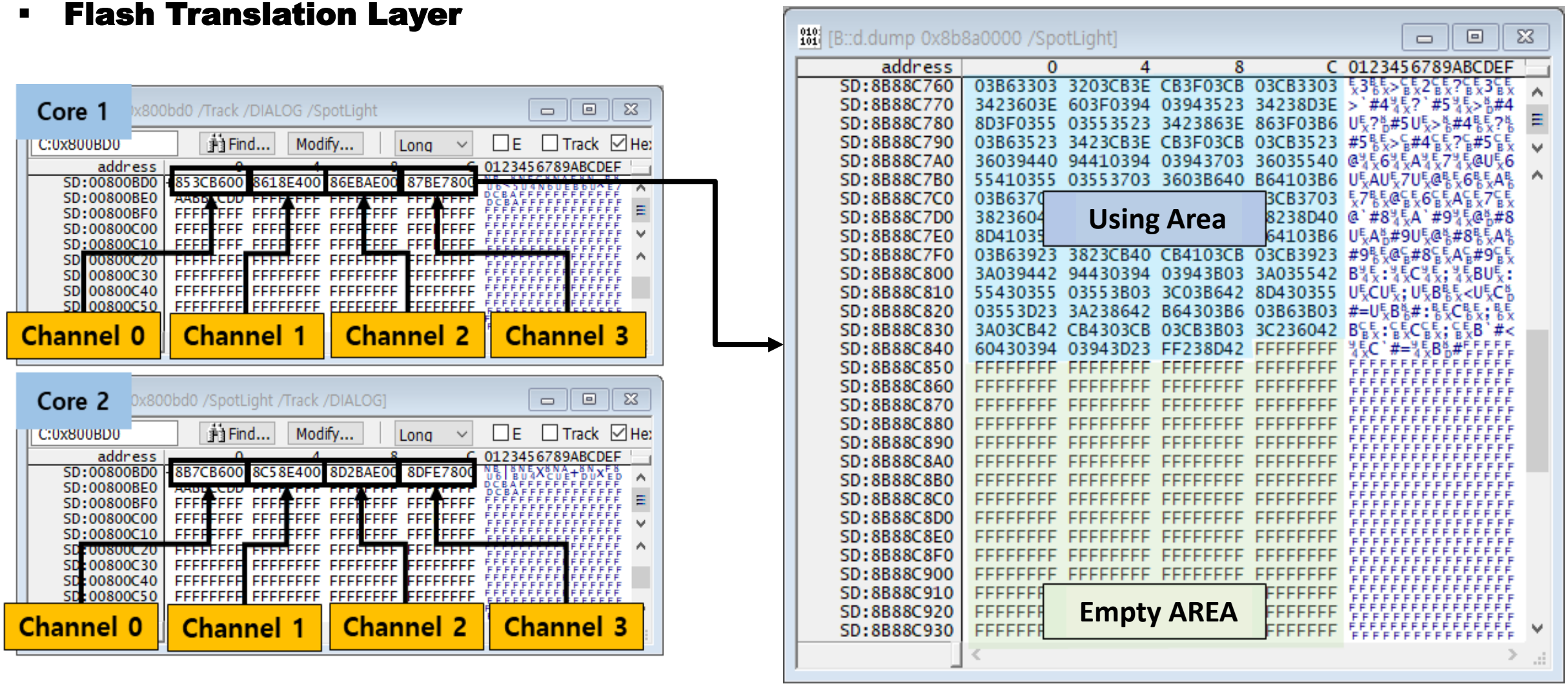


Figure Out Where PBA and Channel Buffer are



L2P Map Table

- **Flash Translation Layer**



LPN → PBA Translation

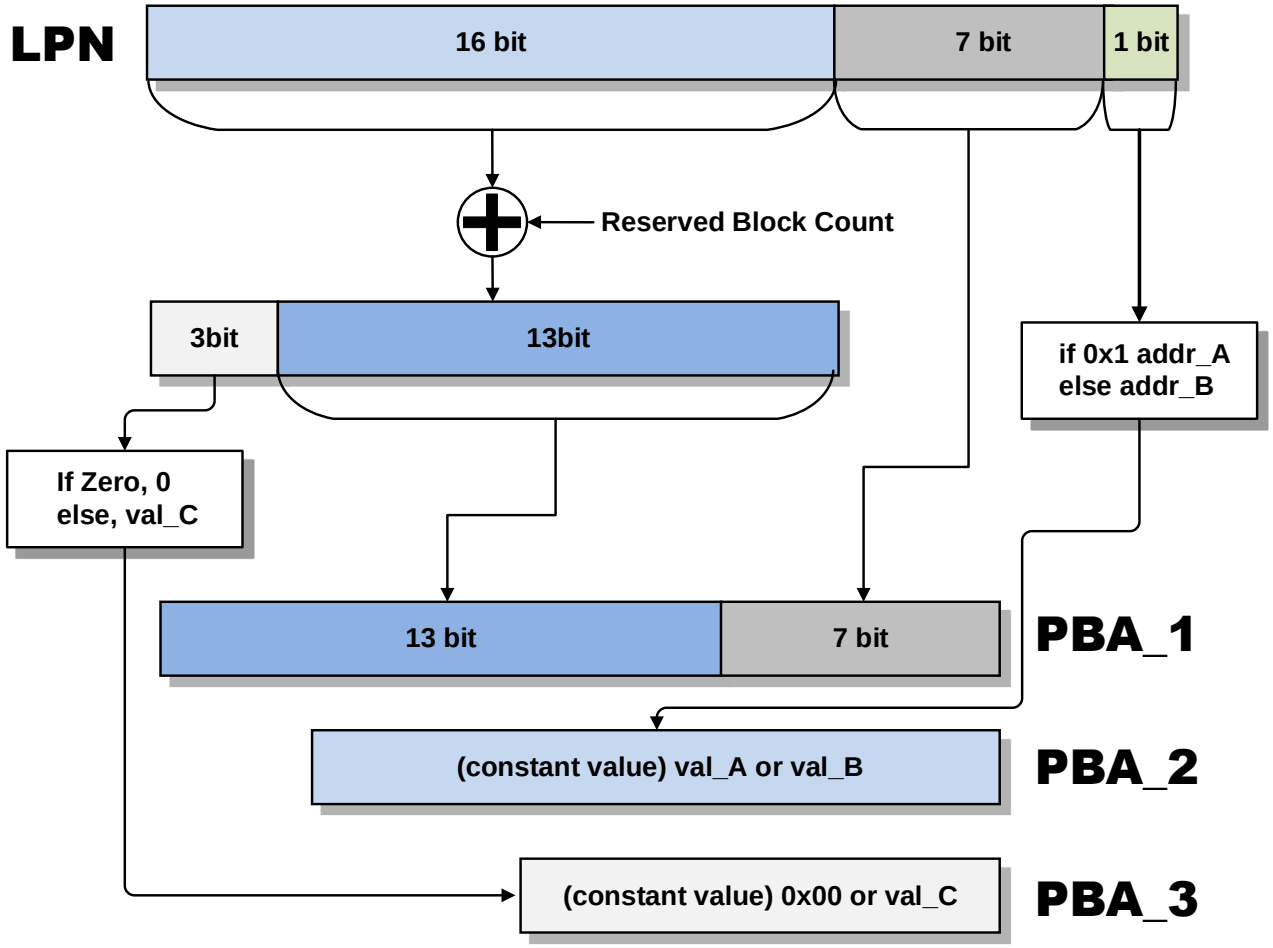
- **Flash Translation Layer**

```
ROM:000077D8
ROM:000077D8
ROM:000077D8
ROM:000077D8                PBA_calc1
ROM:000077D8 70 B5          PUSH            {R4-R6,LR}
ROM:000077DA 06 46          MOV             R6, R0
ROM:000077DC 40 F2 A8 70 MOVW               R0, #0x7A8
```

```
ROM:0000286C CB 4B       LDR            R3, =channel_count
ROM:0000286E CE 4D       LDR            R5, =dword_1C7E8
ROM:00002870 11 6A       LDR            R1, [R2,#0x20]
ROM:00002872 9C 6C       LDR            R4, [R3,#(dword_1C718 - 0x1C6D0)]
ROM:00002874 D5 F8 E0 51 LDR.W          R5, [R5,#(dword_1C9C8 - 0x1C7E8)]
ROM:00002878 DF 6C       LDR            R7, [R3,#(dword_1C71C - 0x1C6D0)]
ROM:0000287A 21 FA 04 F4 LSR.W          R4, R1, R4
ROM:0000287E 66 19       ADDS           R6, R4, R5
ROM:00002880 D3 F8 B4 50 LDR.W          R5, [R3,#(dword_1C784 - 0x1C6D0)]
ROM:00002884 5B 6C       LDR            R3, [R3,#(dword_1C714 - 0x1C6D0)]
ROM:00002886 26 FA 05 F4 LSR.W          R4, R6, R5
ROM:0000288A AF 40       LSLS           R7, R5
ROM:0000288C 7F 1E       SUBS           R7, R7, #1
ROM:0000288E 5B 1E       SUBS           R3, R3, #1
```

```
ROM:00014858
ROM:00014858
ROM:00014858
ROM:00014858                sub_14858
ROM:00014858 02 46          MOV             R2, R0
ROM:0001485A 00 20          MOVS            R0, #0
ROM:0001485C 62 F3 DF 10 BFI.W              R0, R2, #7, #0x19
ROM:00014860 61 F3 06 00 BFI.W              R0, R1, #0, #7
ROM:00014864 70 47          BX              LR
ROM:00014864                ; End of function sub_14858
ROM:00014864
```

Raw Code



Logical Expression

LPN → PBA Translation

PBA_1 = ?

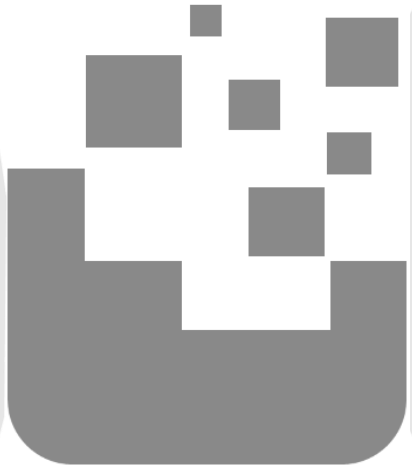
13 bit	7 bit

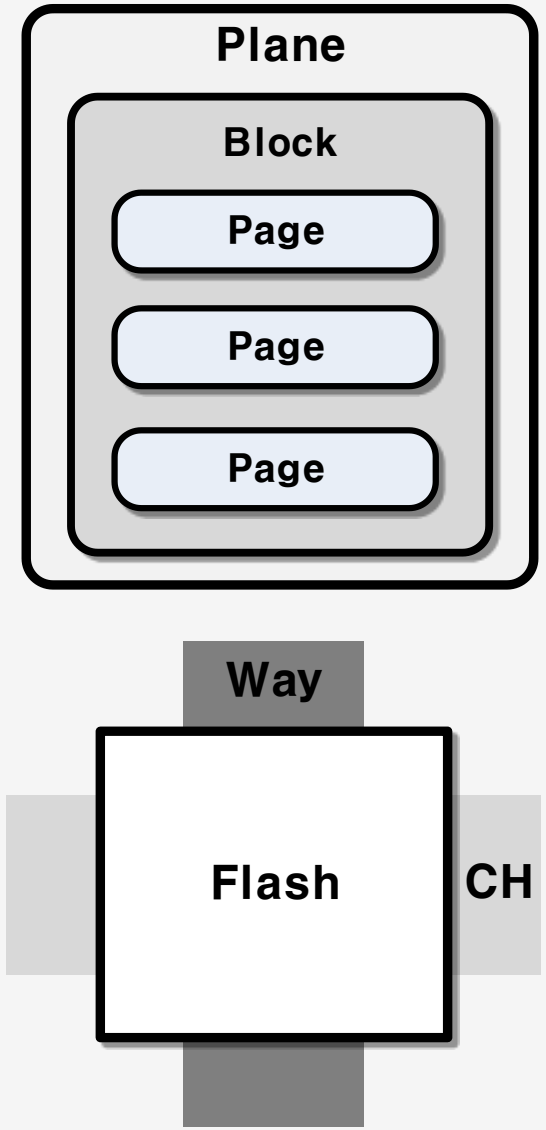
PBA_2 = ?

(constant value) addr_A or addr_B

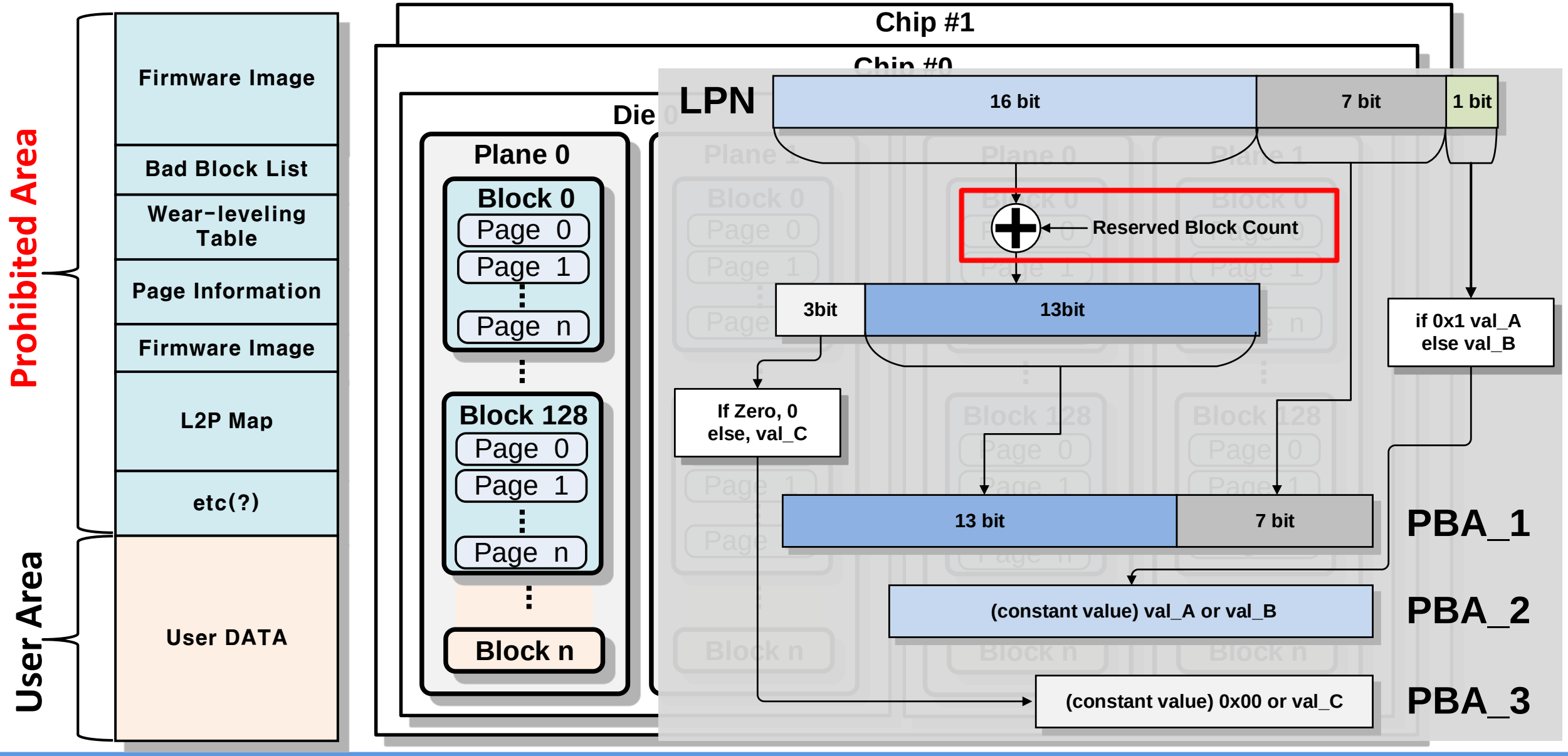
PBA_3 = ?

(constant value) 0x00 or val_C



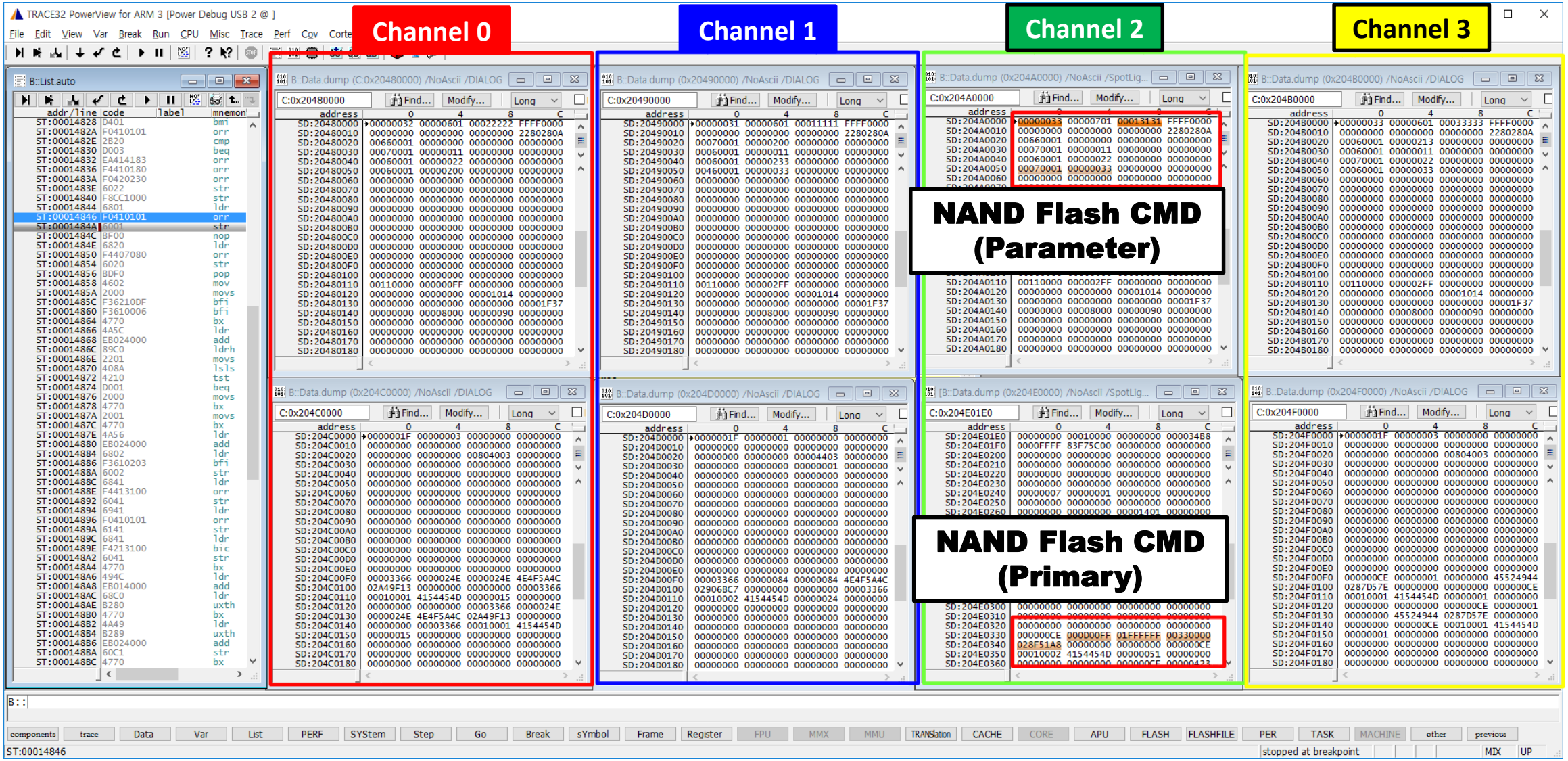


LPN → PBA Translation (Reserved Block Count)



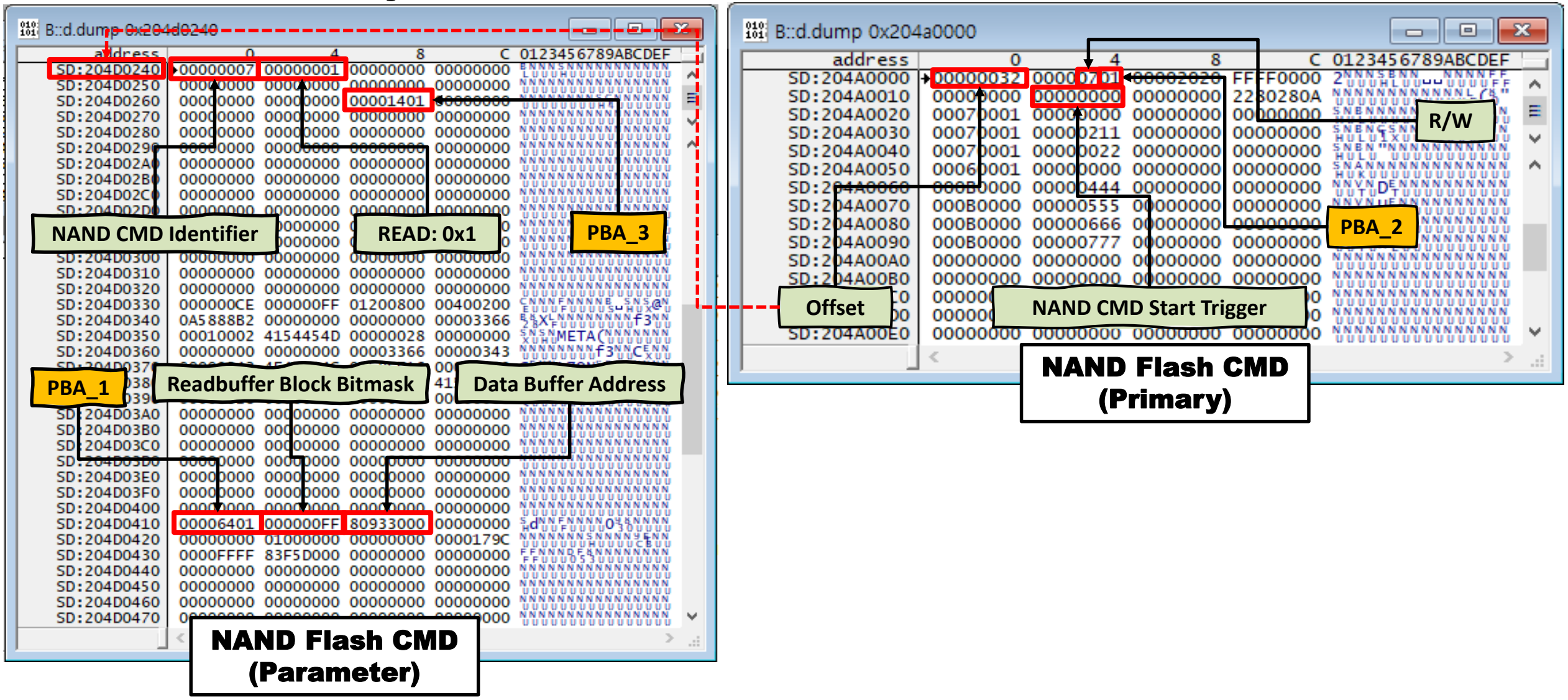
Flash Channel Buffer

- **Flash Interface Layer**

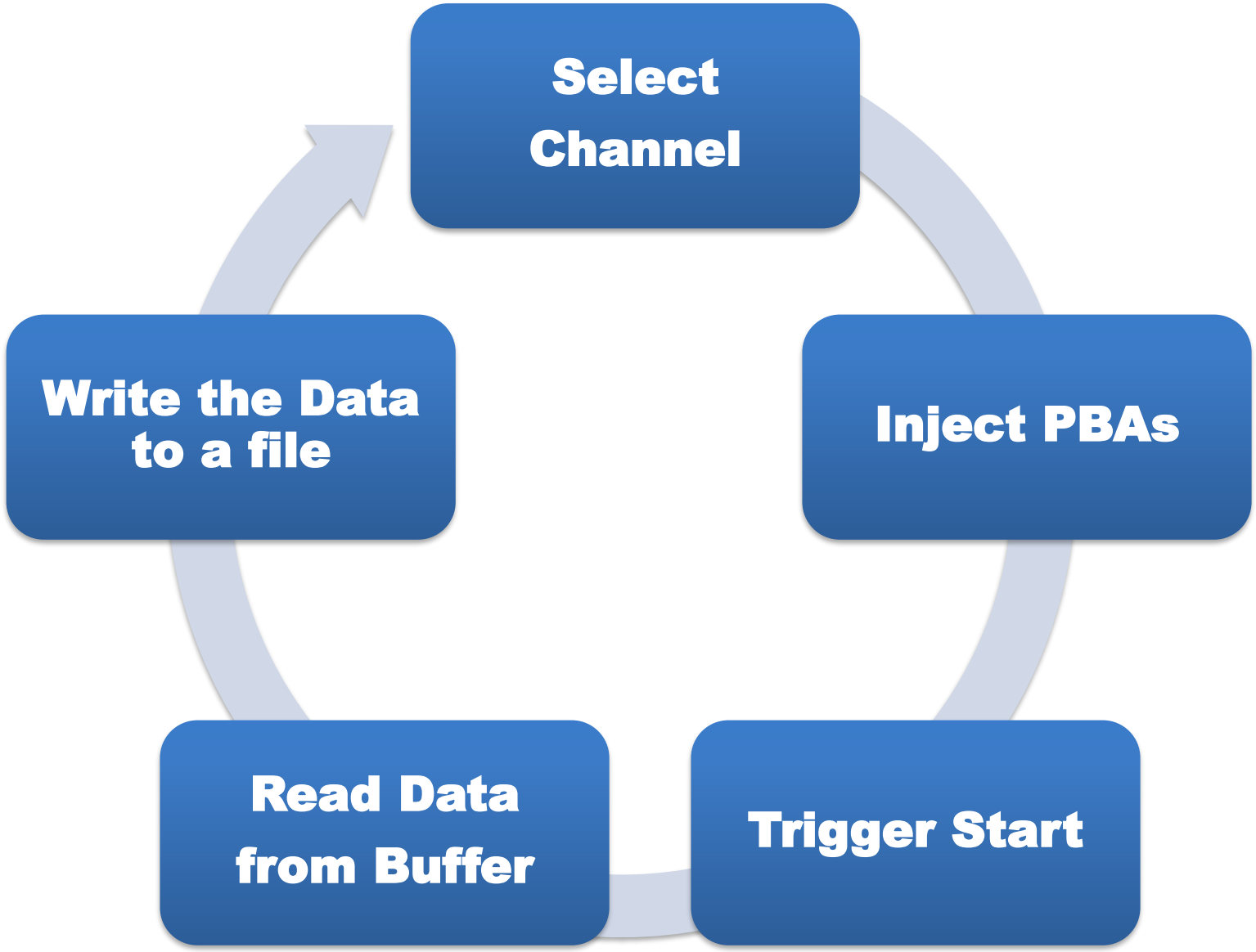


NAND Flash Command in Flash Channel Buffer

- **Flash Interface Layer**



Steps : Access All Physical Block



DEMO : Access All Physical Block

Okay, then what can we do with it?

- **Non Security Encoding-SSD**
  - **Data Restoration**
    - ✓ **Unique PBA blocks not found in the LBA**
  - **Forensics**
    - ✓ **Image based on LBA → Image based on PBA**
    - ✓ **Non-Destructive**

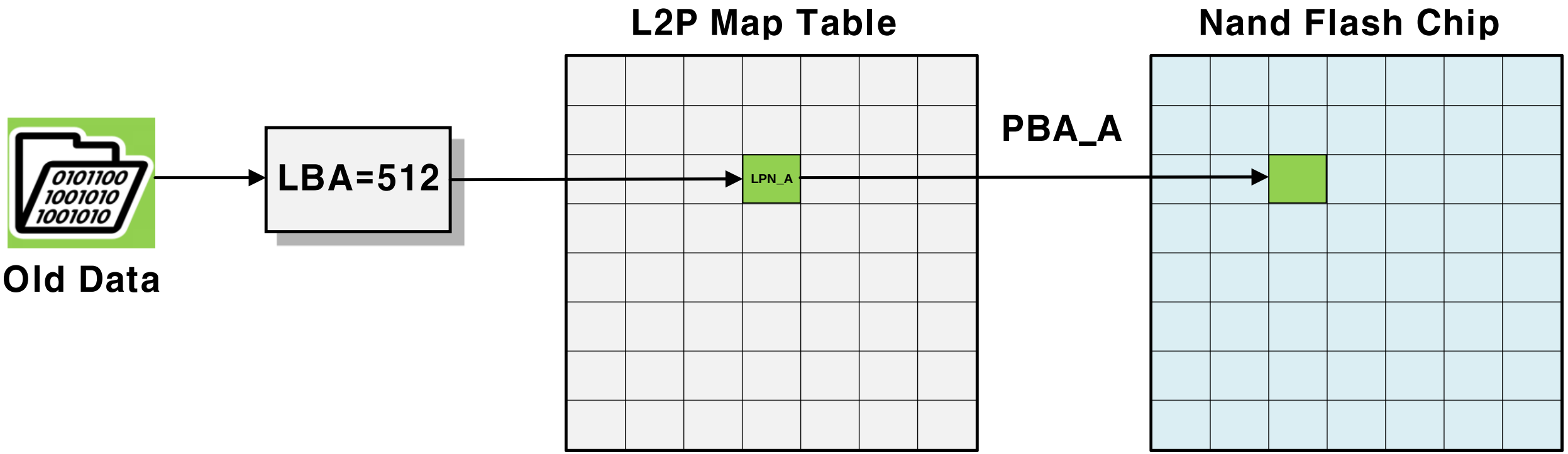
Okay, then what can we do with it?

- **Security Encoding-SSD**
  - **Secure Erase Verification**
    - ✓ **Whether SSD still leaves sensitive data or not**
  - **Access to Factory Reserved Area**
    - ✓ **You may find Critical Information in SSD**
    - ✓ **Generally, Not Encrypted**

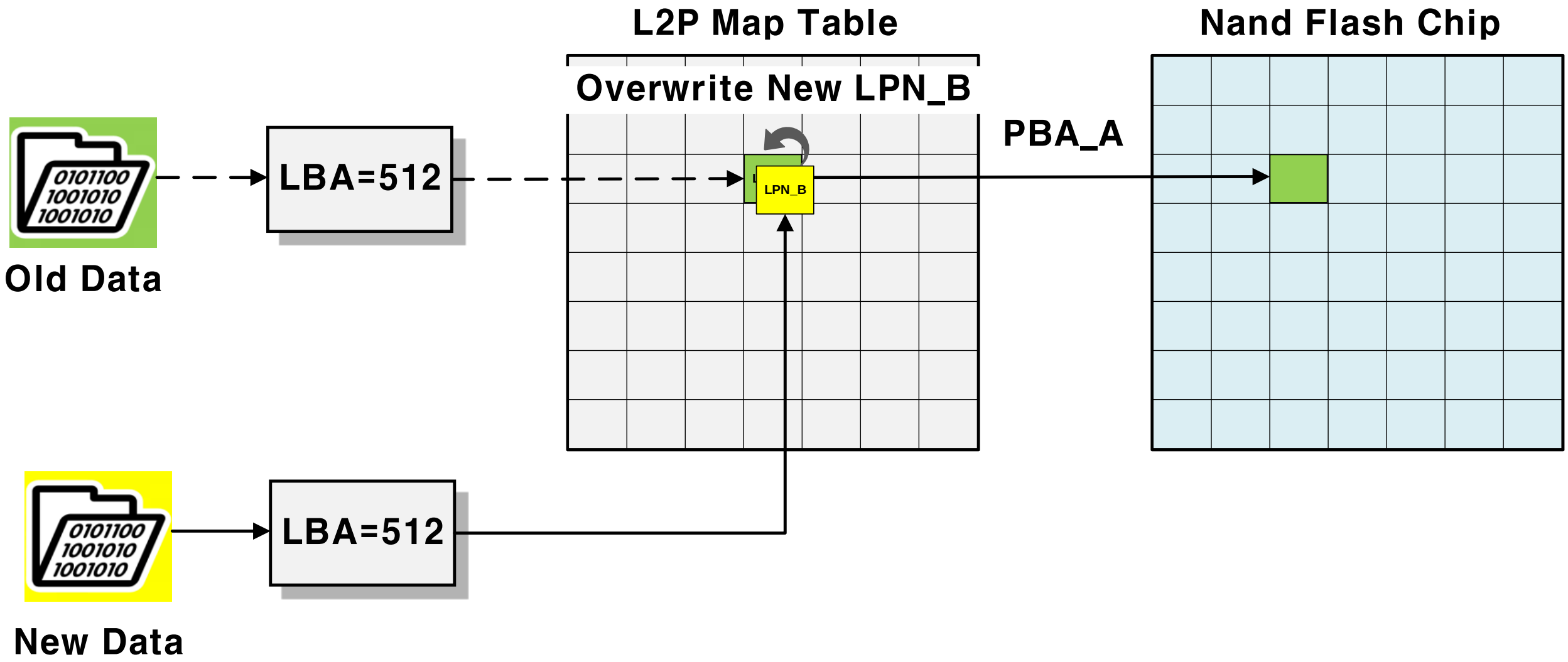
SSD Firmware Reversing

Recover Overwritten Data

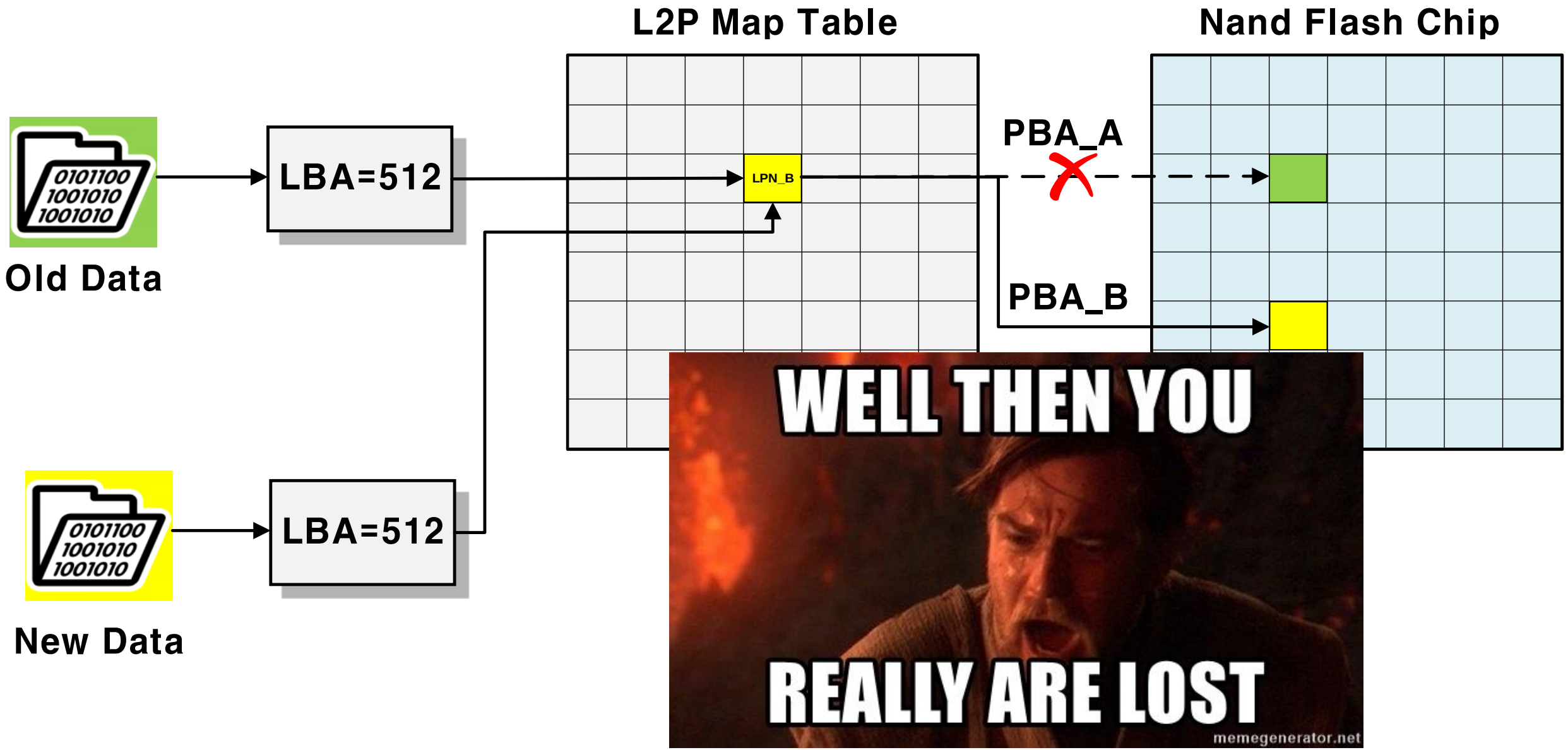
Overwrite to Same LBA



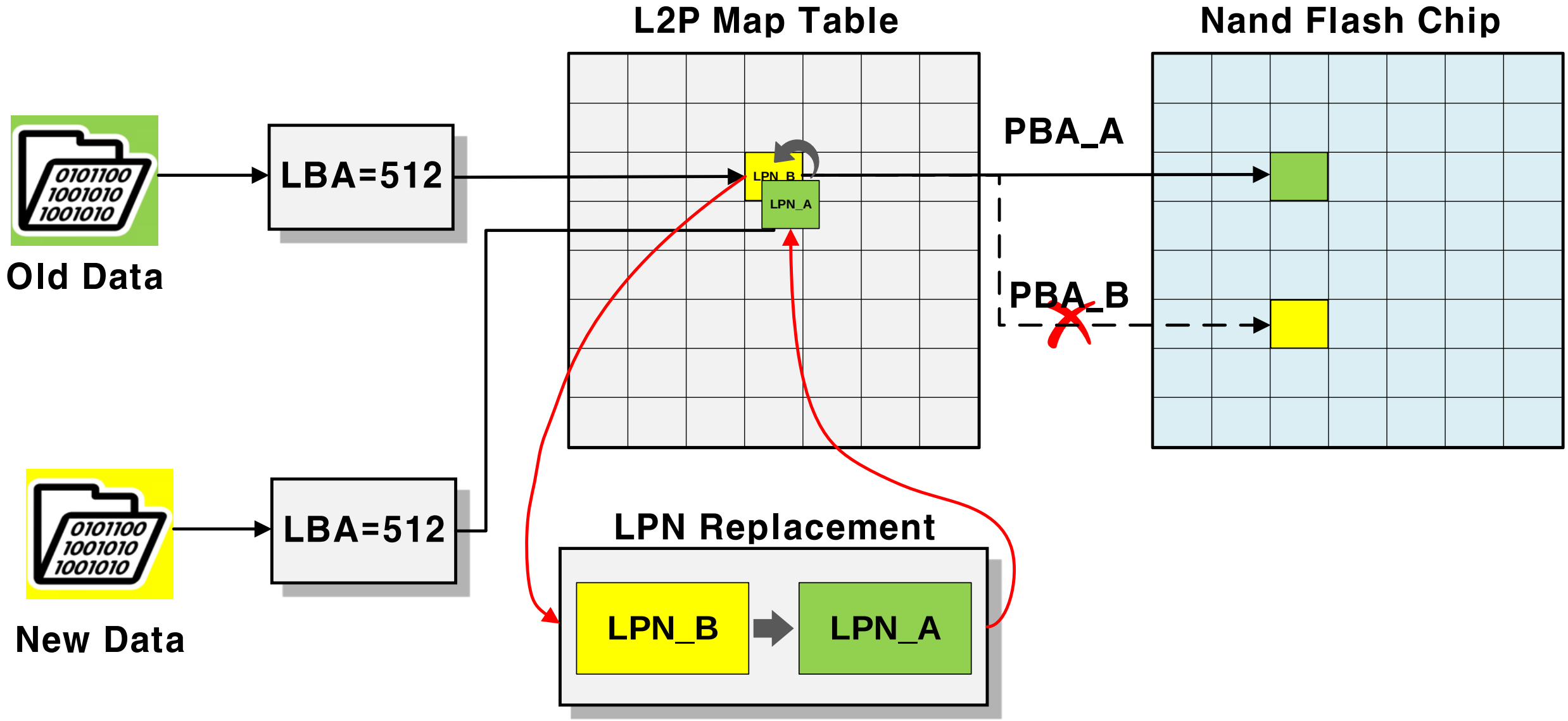
Overwrite to Same LBA



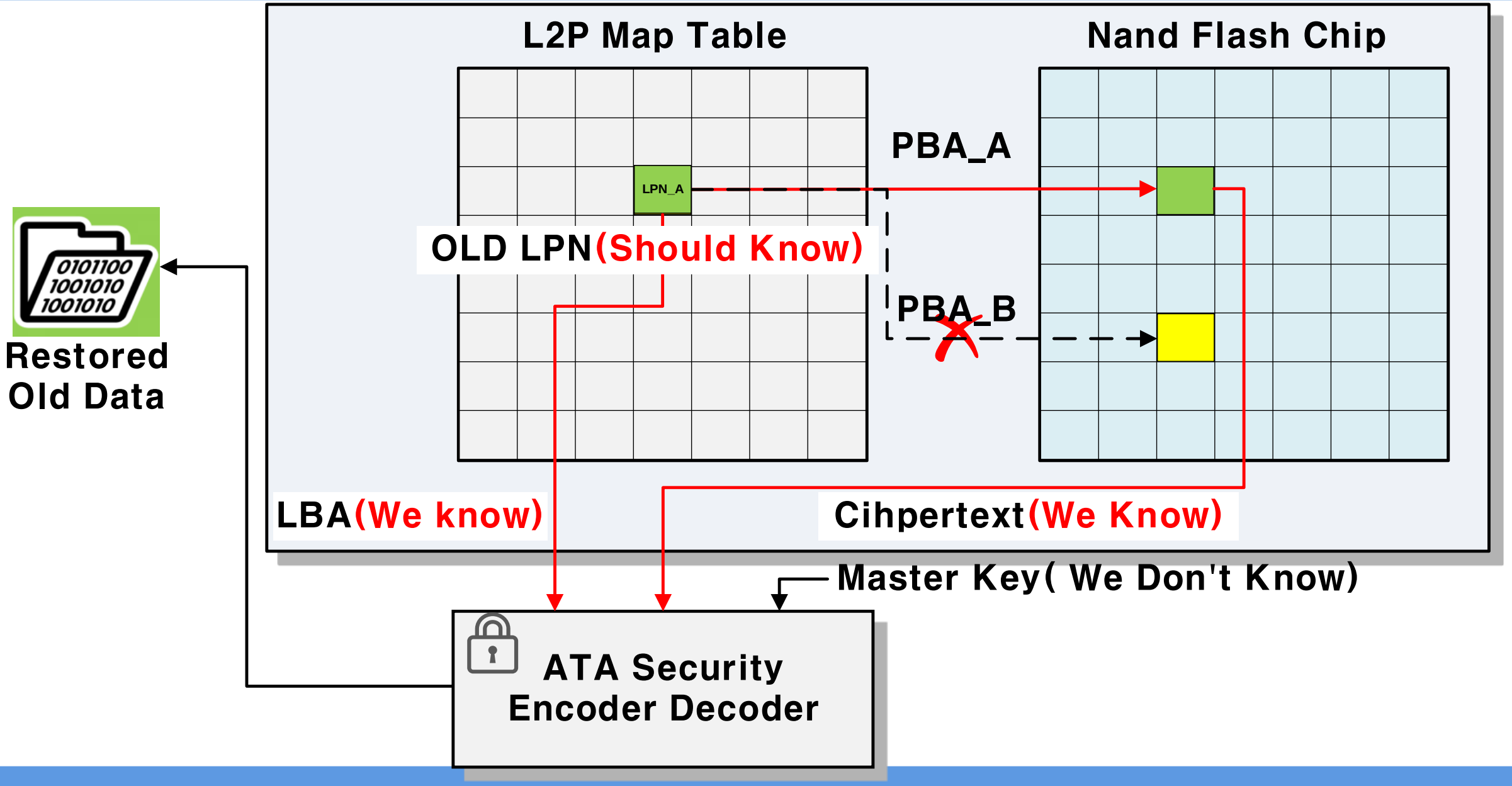
Overwrite to Same LBA



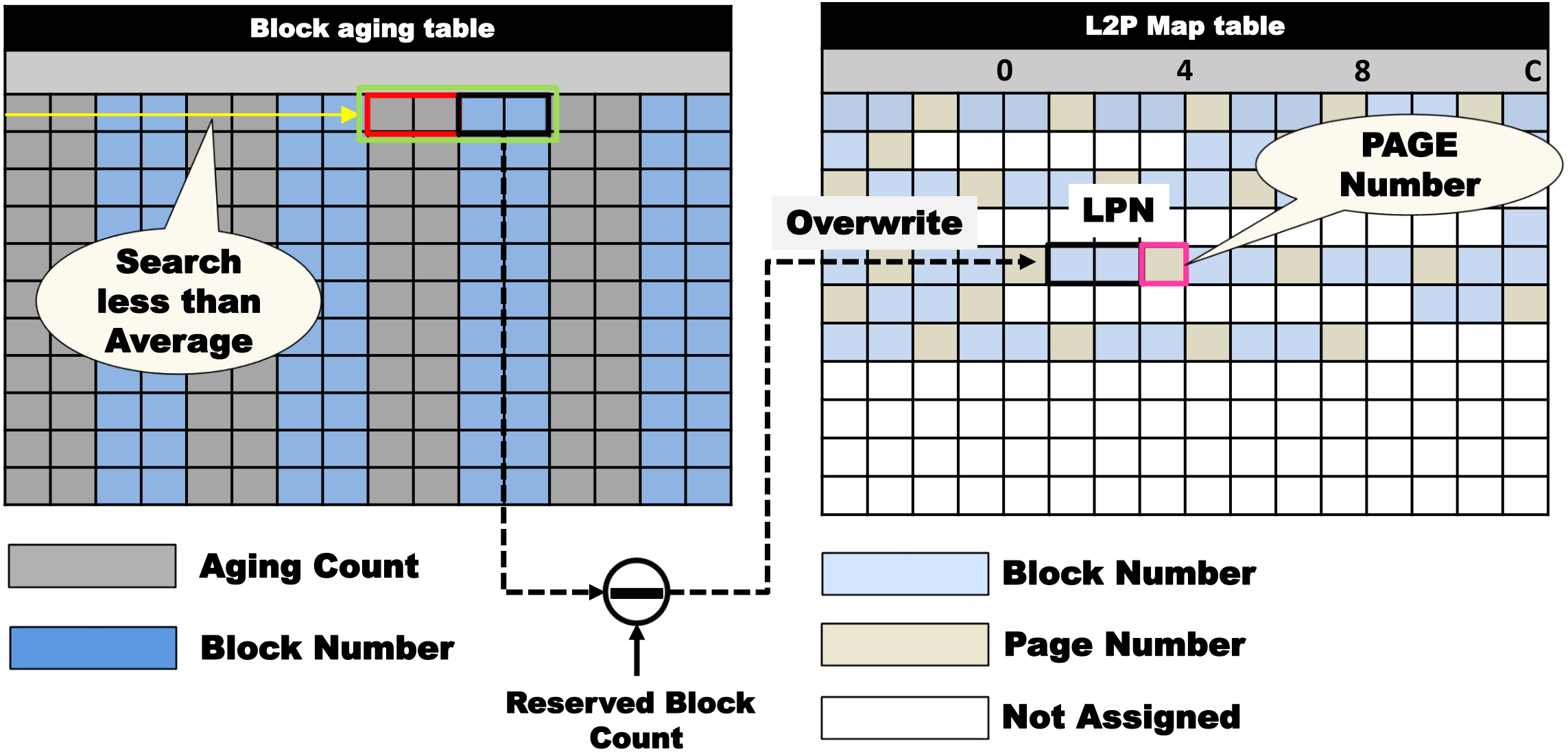
IS It Still Alive?



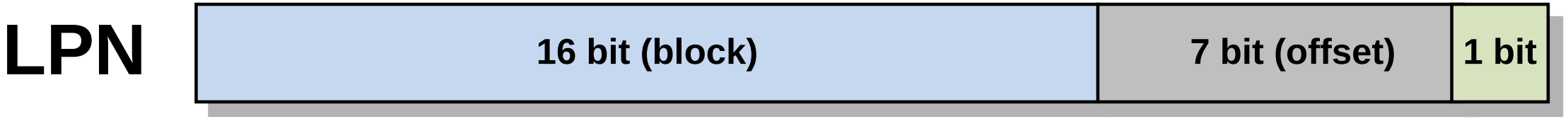
Now, What we Know & Don't know & Should know



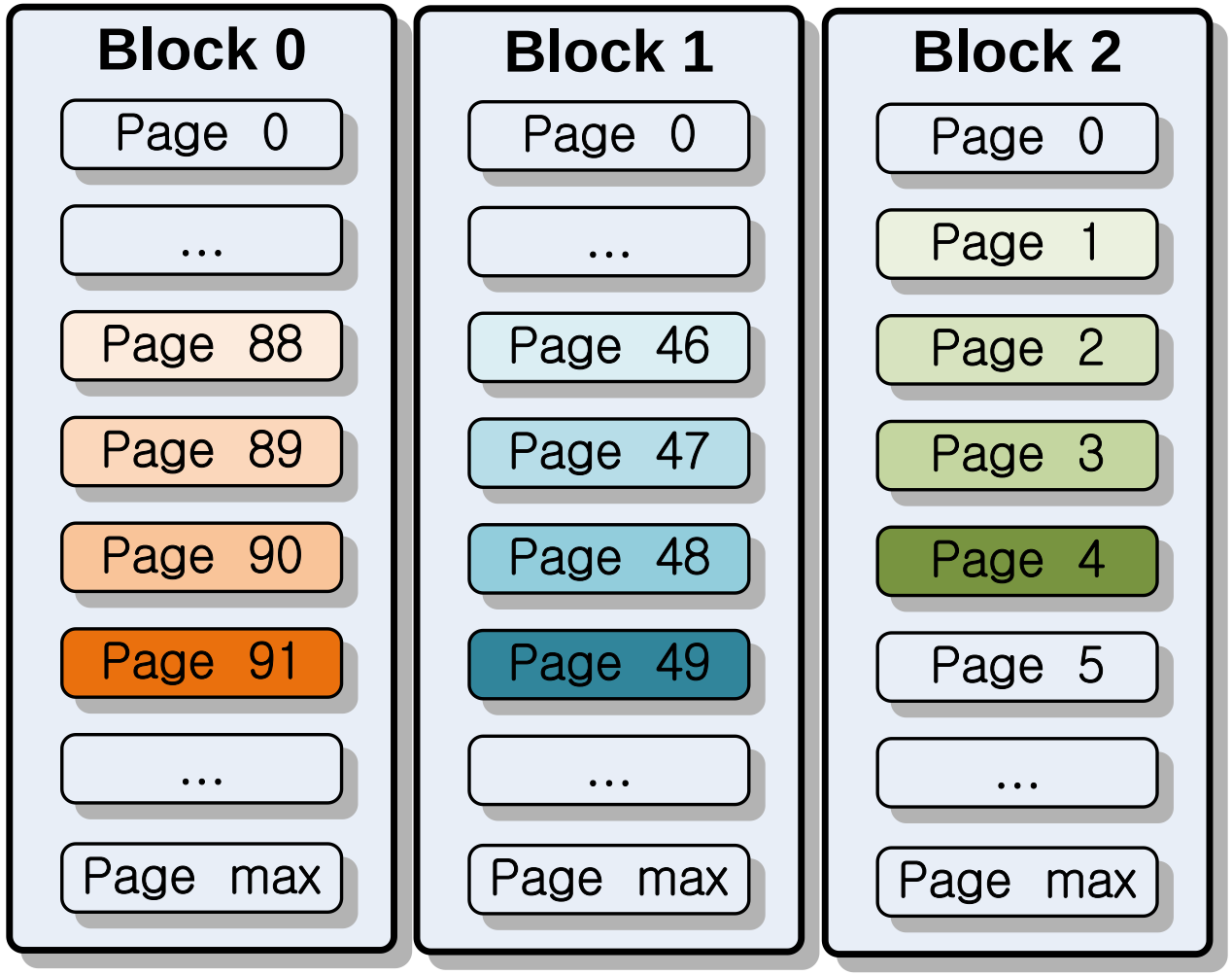
Block aging table and Data Overwriting



Mapping Technique and Data Overwriting

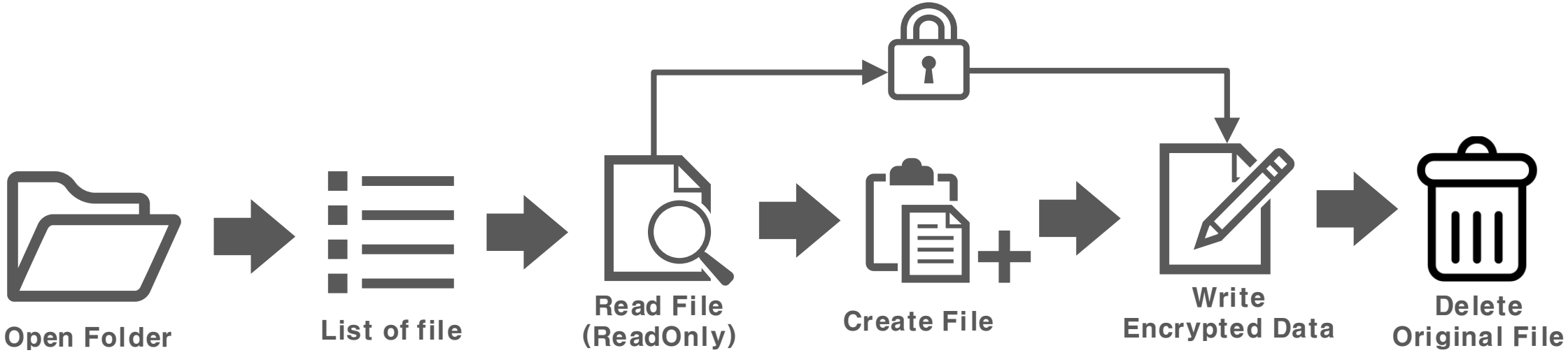


	Overwrite cnt=0	Overwrite cnt=1	Overwrite cnt=2	Overwrite cnt=3
LBA	LPN (Block, Offset)	LPN (Block, Offset)	LPN (Block, Offset)	LPN (Block, Offset)
43	(2,3)	(2,4)	(2,5)	(2,6)
72	(0,88)	(0,89)	(0,90)	(0,91)
84	(1,46)	(1,47)	(1,48)	(1,49)



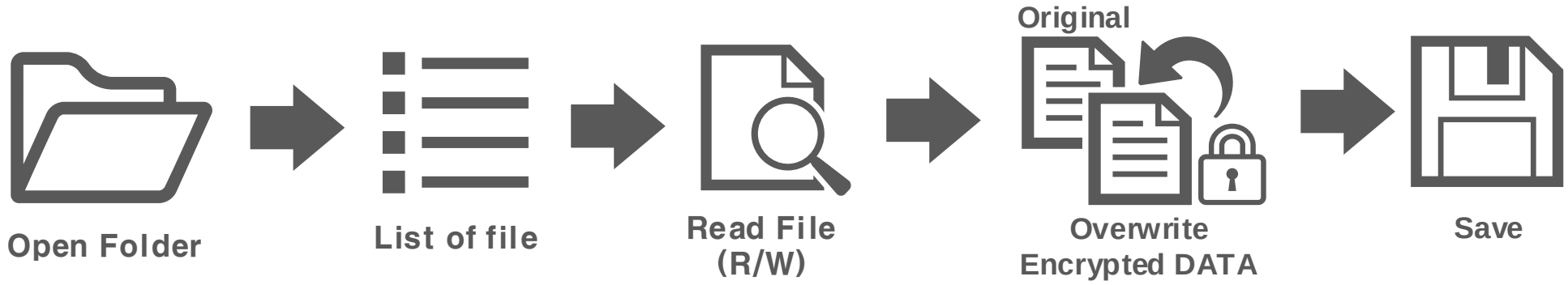
Mapping Technique and Data Overwriting

Type of Ransomware : Overwrite or Copy



Copy And Delete Type

Overwrite Type(In this time)



Steps : Overwritten Data Restoration

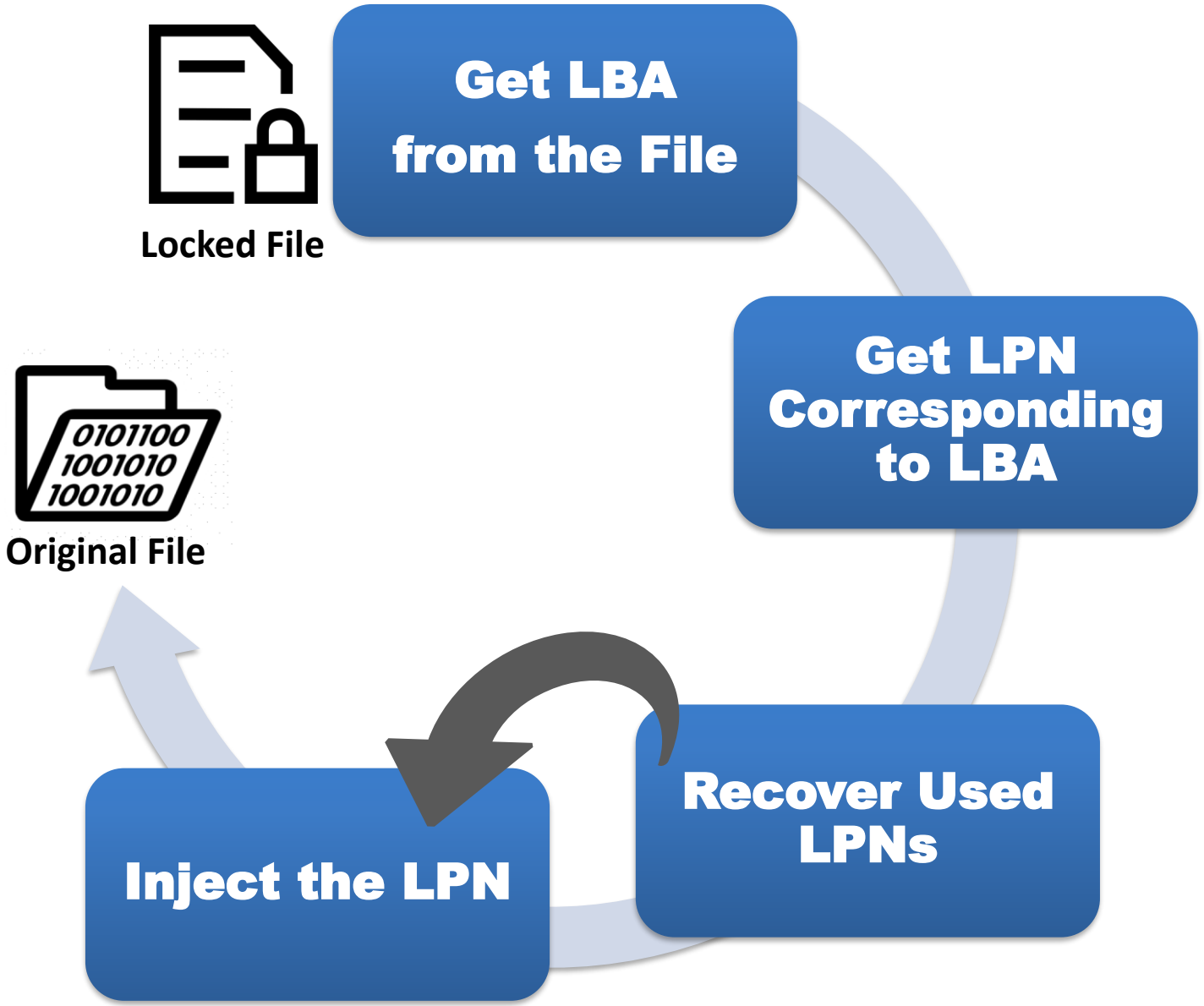
Locked File

Get LBA from the File

Get LPN Corresponding to LBA

Recover Used LPNs

Steps : Overwritten Data Restoration



DEMO : Overwritten Data Restoration

What is Next?

- **Support for various SSDs.**
- **We need to Enhance our tool.**
- **Focus on Forensic tool for their needs.**
- **AES-XTS key extraction using side-channel attack.**
- **A study about Manage to keep Stale data Securely and Efficiently.**

black hat EUROPE 2020

DECEMBER 9–10

BRIEFINGS

Thank you!

If you have any question, please send me email

Kwonyoup Kim
CEO/founder
kkyoup@sntworks.kr

Seungjoon Lee
Senior Researcher
sj.lee@sntworks.kr

Twitter
@SNTWORKS1

Youtube Channel
SNTWORKS Inc.

SNTWORKS