Open-Channel Solid State Drives NVMe Specification

Revision 1.2
April 2016

Please write to Matias at mb@lightnvm.io for collaboration

Table of Contents

1. Introduction

NVM Express (NVMe) 1.2 and prior revisions define a register-level interface for host software to communicate with a non-volatile memory subsystem over PCI Express (i.e., NVMe over PCIe).

This specification defines a Physical Page Address Command Set extensions to the NVMe specification that enables operation with devices that expose their physical media to the host. The document contains the definition for identifying geometry and data commands to control NAND flash behind an Open-Channel Solid State Drive controller.

1.1 Definitions

1.1.1 physical media

The underlying physical non-volatile memory that is attached to the NVMe controller. Typically assumes NAND flash, but is not limited to such.

1.1.2 open-channel solid state drive

Used to refer to Solid State Drives that allow the host to understand and manage the physical mapping within a Solid State Drive.

1.1.3 media manager

A software unit that abstracts the physical media characteristics for above layers to consume.

1.1.4 target

A software unit that exposes the physical media in one or more ways. For example as a block device, file-system, key-value, or object-based interface.

1.1.5 Logical Unit Number (LUN)

Used to refer a single parallel unit within the storage media. A single SSD may contain tens or hundreds of these LUNs.

1.1.6 Physical Page Address (PPA)

Used to define an address on the physical media. The PPA follow the device PPA format; that represents the structure of the PPA.

1.2 Theory of Operation

1.2.1 Overview

An Open-Channel SSD is a device that shares responsibilities with the host to implement and maintain functionalities that a traditional SSD keeps strictly in firmware.

By moving traditional SSD responsibilities to the host, I/Os latencies can be significantly more predictable. The added responsibilities allows host software to adapt its data placement, garbage collection and I/O scheduling algorithms to a given user I/O workload. Traditional, these optimizations are applied on the device side of the storage interface and hidden behind a narrow storage interface, which prevents significant optimizations and algorithms to change during run-time.

Storage Software Stack

Figure 1 depicts the Linux host architecture for Open-channel SSDs, also known as LightNVM. It consists of three fundamental components: LightNVM compatible device drivers, media managers, and targets.

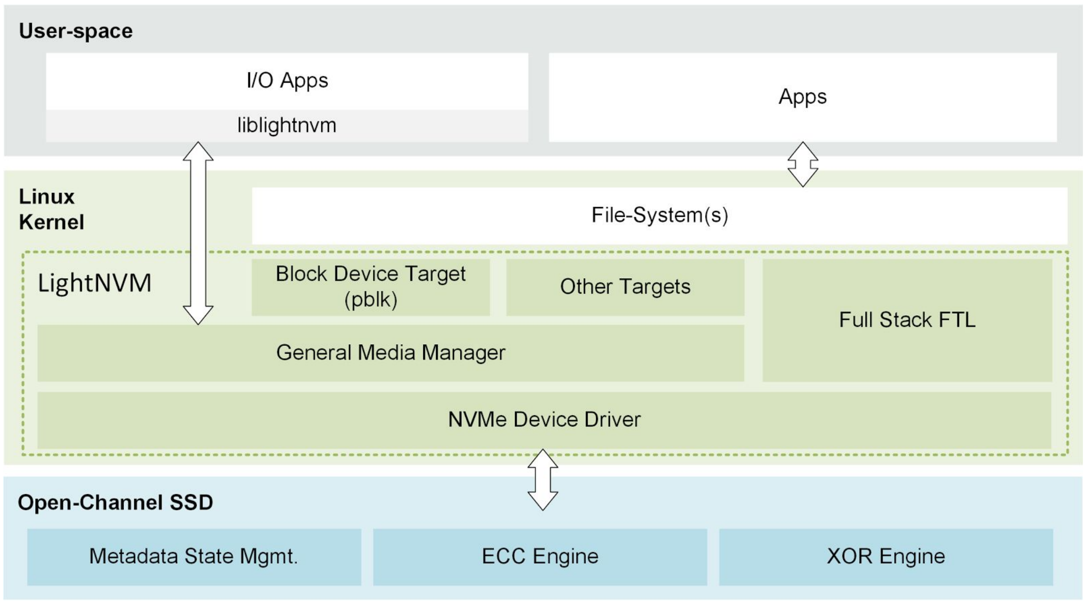

Figure 1: Overview of Open-Channel SSD host architecture as implemented in the Linux Operating System.

Device drivers: A storage device driver implements support for the Open-Channel SSD Physical Page Address command set. The command set includes (i) identify structure, used by the host to discover available features, extensions, and characteristics of an open-channel device; and (ii) and physical data commands to communicate efficiently with the storage non-volatile media.

Media Manager: The media manager abstracts the underlying physical media by hiding its constraints and access details. It is responsible for (i) name mapping between vendor specific and generic addressing format, (ii) device-specific SSD state management (if not implemented at the device-side), and (iii) recovery – to guarantee durability when manipulating the metadata associated with SSD state management. A media manager acts as a mediator between physical media, and users, abstracting away media complexities. The LightNVM general media manager implements minimal block management and lets targets or user space applications perform the actual FTL. It is possible for a media manager to implement the whole FTL and manage user-space interfaces. For example, a media manager could expose a block device similar to a traditional block storage device. Such a media manager would manage functionalities such as data placement, garbage collection, and block management.

Targets: When using the general media manager, targets implement FTL functionality (e.g., translation logic, data placement, or garbage collection). They also expose a storage interface to user space. Examples of such interfaces include block devices, and key-value stores, or object stores. Target and device (media manager) division, grouping and assignments are flexible in such a way that one device can be split and assigned to multiple targets or multiple devices can belong to one target.
Additionally, LightNVM connects to user space through liblightnvm which exposes a get/put block interface for applications to implement own application managed FTLs.

1.2.2 Hardware Architecture

Figure 2 depicts an example hardware architecture for Open-Channel SSDs. The SSD controller consists of a host interface controller, NAND controller, DRAM controller and general purpose CPUs. The dotted lines signify that when no internal firmware is implemented, the CPUs and DRAM are not strictly necessary.

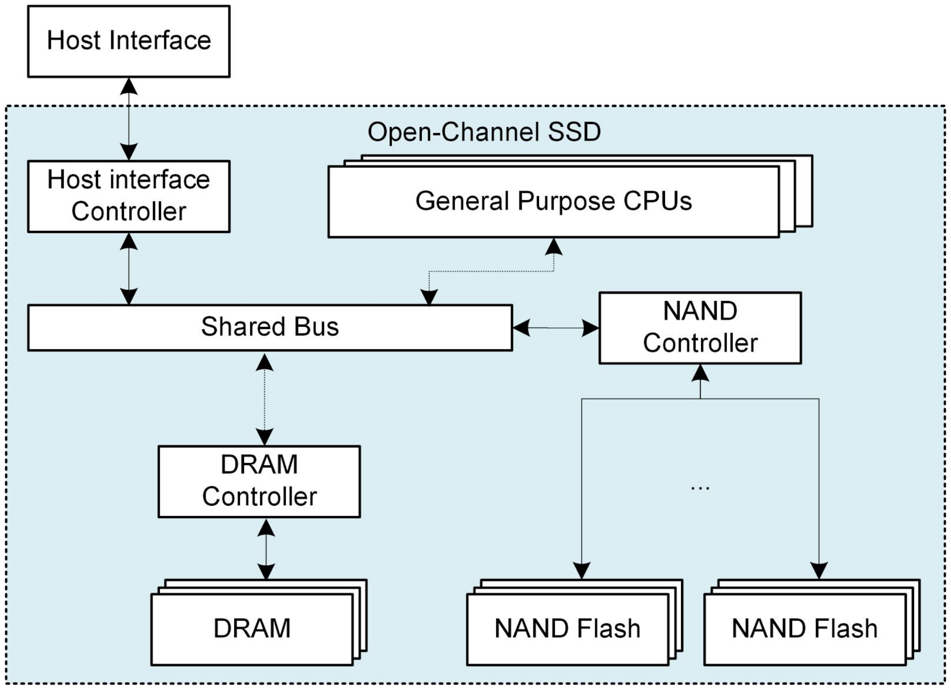

Figure 2: Overview of Open-channel SSD architecture.

Open-Channel SSDs expose a series of Logical Unit Numbers (LUNs), through NAND Flash, representing a unit of parallelism on the device. Through the Open-Channel SSD interface, the host can determine the configuration and implement the necessary logic to drive the physical media.

2. Physical Page Addressing

This section defines the specification of Physical Page Addressing (PPA) command set. It consists of two parts: admin commands and physical media commands. The admin commands are used to communicate geometry of the device and the physical media commands are used to access the physical media within a device.
This specification defines a command set that enables the host to drive Open-Channel SSDs. This specification supplements NVMe specification revision 1.2.

2.1 Admin Commands

The physical page addressing command set defines three new admin commands to enable the host to enumerate the configuration of the device and administer the internal bad block table of a device.

Opcode (07)	Opcode (06:02)	Opcode (01:00)	NVMe Opcode	O/M	Command
Generic	Function	Data Transfer			
1b	110 00b	10b	E2h	M	Device Identification
1b	111 00b	01b	F1h	O	Set Bad Blocks Table
1b	111 00b	10b	F2h	O	Get Bad Blocks Table

Table 11: Opcodes for Admin Commands – PPA Command Set Specific

All commands are implemented using the regular NVMe command structure, as described in the NVMe 1.2 specification, under the section “Submission Queue Entry – Command Format”. Non-specified fields follow the standard submission queue entry fields.

2.1.1 Command Set Identification

A device is identified by its PCI IDs. Currently, the following PCI IDs are reserved and used for identification:

Device	Description
QEMU NVMe	LightNVM support is available when device matches PCI Vendor ID (0x1d1d), Device ID (0x1f1f) and the first byte (value 0x1) in the vendor specific bits of the namespace identify structure is set.
CNEX Labs	LightNVM support is available when device matches PCI Vendor ID (0x1d1d), Device ID

	(0x2807) and the first byte (value 0x1) in the vendor specific bits of the namespace identify structure is set.

2.1.2 Device Identification

For the host to issue commands to non-volatile media (e.g. NAND flash), the boundaries of the media are required. The identify device command accomplishes this by describing configurations. A configuration is a set of LUNs attached to a set of channels, with the definition of the NAND configuration (pages, block, planes, and so forth, Table 6). The first 256 bytes are global parameters and then each configuration has 960 bytes available.

Command Submission
The following NVMe command may be used to request device identification.

- **Opcode:** E2h
- **NSID:** Namespace ID
- **PRP1**: Address to 4K DMA memory region in which to store the response
- **CDW10-15:** Reserved

Command Completion
On command completion, the controller must post a completion queue entry to the Admin Completion Queue indicating the status for the command.
Status Field and Status Code follows the error codes defined in NVMe Specification Section 2.3.

The data returned follows this format:

Byte	Name	Bits	Description
0	Version	8	Version ID
1	Vendor NVM opcode command set (VNVMT)	8	Vendor opcode configuration. See Table 2 (Vendor Opcode Configuration).
2	Configuration groups (CGRPS)	8	Number of configuration groups
3	Reserved	8	
07:04	Capabilities (CAP)	32	Device capabilities and feature support. See Table Device Capabilities and Feature Support.
11:08	Device Op Mode (DOM)	32	Current device operating mode. See Table Device Operating Mode.
27:12	PPA Format (PPAF)	128	PPA Format. See Table Device PPA Format.
255:28	Reserved	1824	Reserved for future use

1215:256	Group #0 description	7680	Description of the 1st group
2175:1216	Group #1 description	7680	Description of the 2nd group
3135:2176	Group #2 description	7680	Description of the 3rd group
4095:3136	Group #3 description	7680	Description of the 4th group

Table 1: Identify Device Data Structure

The vendor code configurations allow devices to have specific vendor codes for specific operations. For example other read/write opcodes and format.

Value	Name	Description
0	Generic	Enable opcodes as found in this specification
255:1	Reserved	Reserved for future opcode configurations

Table 2: Vendor Opcode Configuration

Bit	Description
0	Bad block table management. If set to 1, · Support for Get/Set Bad Block Table commands. · Device manages a persistent bad block table If cleared, not supported.
1	Hybrid command support. If set to 1, - hybrid write/read command are supported - get translation table supported
31:2	Reserved

Table 3: Device Capabilities and Feature Support

Bits	Description
0	Hybrid mode. 0 = Normal mode (L2P MAP is in host) 1 = Hybrid mode (L2P MAP is in device)
1	Error Code Correction (ECC) mode. 0 = ECC provided by the device controller 1 = Host provides ECC
31:2	Reserved

Table 4: Device Operating Mode

The PPA format defines the structure of the device PPA. This is, describing the physical offsets of each PPA attribute. The host needs to convert its physical address format into this device side physical format. Every PPA mentioned in all command descriptions must follow this format. The global address format applies to all group description, so each bit field length must be big enough to cover maximum length among all groups.

Byte	Name	Description
0	Channel bit start	Starting bit position for channel
1	Channel bit length	Contiguous number of bits assigned to Channel information
2	LUN bit start	Starting bit position for LUN
3	LUN bit length	Contiguous number of bits assigned to LUN information
4	Plane bit start	Starting bit position for plane
5	Plane bit length	Contiguous number of bits assigned to information
6	Block bit start	Starting bit position for block
7	Block bit length	Contiguous number of bits assigned to block information
8	Page bit start	Starting bit position for plane
9	Page bit length	Contiguous number of bits assigned to information
10	Sector bit start	Starting bit position for controller sector
11	Sector bit length	Contiguous number of bits assigned to controller sector information
15:12	Reserved	Reserved for future extension

Table 5: Device PPA format

A group description consists of:

Bytes	Name	Bits	Description
0	Media Type (MTYPE)	8	Group Description Structure Type. The rest of the fields are defined by the following type:

value	Device Type
0	NAND Flash Memory
255:1	Reserved

1	Flash Media Type (FMTYPE)	8	Media type specific. NAND Flash Memory Flash Media type values are:

value	Device Type
0	Single bit Level Cell flash (SLC)
1	Two bit Level Cell flash (MLC)
2	Three bit Level Cell flash (TLC)
255:3	Reserved

3:2	Reserved	16	Reserved
4	Number of channels (NUM_CH)	8	Number of controller channels.
5	Number of LUNS (NUM_LUN)	8	Number of LUNs (flash die) per channel.
6	Number of Planes (NUM_PLN)	8	Number of flash planes per LUN.
7	Reserved	8	Reserved
9:8	Number of Blocks (NUM_BLK)	16	Number of flash blocks per plane.
11:10	Number of Pages (NUM_PG)	16	Number of flash pages per block.
13:12	Page size (FPG_SZ)	16	Number of bytes in a flash page.
15:14	Controller Sector Size (CSECS)	16	Controller defined minimum data unit protected by ECC (in bytes). For example 4096 bytes.
17:16	Sector OOB size (SOS)	16	Per-sector metadata (in bytes).
19:18	Reserved	16	
23:20	tRD Typical (TRDT)	32	Typical page read time (in ns).
27:24	tRD Max (TRDM)	32	Max page read time (in ns).
31:28	tPROG Typical (TPRT)	32	Typical page program time (in ns).
35:32	tPROG Max (TPRM)	32	Max page program time (in ns).
39:36	tBERS Typical (TBET)	32	Typical block erase time (in ns).
43:40	tBERS Max (TBEM)	32	Max block erase time (in ns)
47:44	Multi-plane Operation	32	Support for multi-plane operations. Set bit indicates

	Support (MPOS)		supported.

Bits	Feature Supported
0	Single plane read
1	Dual plane read
2	Quad plane read
7:3	Reserved
8	Single plane program
9	Dual plane program
10	Quad plane program
15:11	Reserved
16	Single plane erase
17	Dual plane erase
18	Quad plane erase
31:19	Reserved

51:48	Media and Controller Capabilities (MCCAP)	32	Media type specific. For NAND flash memory:

Bits	Capability Supported
0	SLC mode
1	Command suspension
2	Scramble On/Off
3	Encryption
31:4	Reserved

53:52	Channel parallelism (CPAR)	16	Number of parallel commands in-flight within a channel.
63:54	Reserved	80	
959:64	Media Type Specific	7168	NAND Flash Memory:

Bytes	Description
7:0	ID codes for READ ID command

			n:8	MLC - page pairing TLC - page pairing and program sequence
			959:n+1	TBD: Bad block marker information, etc.

Table 6: Layout of group entry returned by the identify device command.

NAND Flash MLC Page Pairing information

The MLC pairing information is packed for space optimization. The pairing information is packed into one byte for each pairing and has upper page offset from paired lower page and increment from the previous lower page.

The assumption is any next lower page from the current lower page is within 16 pages, and paired upper page is within a 15-page range. Page pairing is listed in ascending order of lower page number. The first-page is based on page number 0. So lower page number calculation is as follows:

1. The first lower page number is incremented value itself.
2. From now on, lower page number is (previous lower page number + increment value (bit 3:0)).
3. Paired upper page number is (lower page number + upper page offset (bit 7:4))

Bytes	Description
1:0	Number of page pairings (num)
(2+num):2	List of page pairing information byte in ascending order

Table 7: MLC Page Pairing Table

Bits	Description
3:0	Lower page number increment from previous lower page number. The first lower page increment is based on page 0. 0 means 16.
7:4	Paired upper page number offset from the lower page number. 0 means no upper page.

Table 8: MLC Page Pairing Information Byte Structure

TLC - Page pairing and program sequence

It will only have 16-bit information and having value of 0 indicating it has standard TLC format of page pairing and program order for now.

Byte	Description
1:0	0: Standard TLC Format Non-zero: non-standard TLC format. TBD.
*:2	Reserved for non-standard TLC format.

Table 9: NAND Flash TLC information

The following is the standard TLC format. This is for ‘n’ number of Word Line (WL).

WL#	Page Numbers	Program Order		
		First Program Cycle	Second Program Cycle	Third Program Cycle
0	0, 1, 2	1	3	6
1	3, 4, 5	2	5	9
2	6, 7, 8	4	8	12
		7	11	15
		10	14	18
:	:	:	:	:
n-4	3n-12, 3n-11, 3n-10	3n-14	3n-10	3n-6
n-3	3n-9, 3n-8, 3n-7	3n-11	3n-7	3n-3
n-2	3n-6, 3n-5, 3n-4	3n-8	3n-4	3n-1
n-1	3n-3, 3n-2, 3n-1	3n-5	3n-2	3n

Table 10: Standard TLC Page Pairing and Program Order

2.1.3 Get Block Information Command and Response

This command retrieves bad block table for a specified PPA. The PPA fields Sector, Block, Plane section is ignored. On table return, the host must allocate enough memory to contain the returning information. The number of bytes required is (header size, 32 bytes) + (number of blocks per LUN) x (number of planes per LUN).

The bad block list is factory initialized with the bad block lists of the attached physical media.

Command Submission
Returns bad block table byte array, each byte corresponding to the physical page address (PPA) in the LUN. The table size is determined by the total number of blocks in the LUN and should request exact transfer size.

The command is submitted by the following NVMe submission entry:

- **OPC**: F2h
- **NSID**: Namespace id
- **PRP1**: Address to DMA memory region with the correct size data buffer
- **CDW10-11 (PPA)**: Physical Page Address with channel and LUN of which to retrieve the bad block list.
- **CDW12-15**: Reserved

Request the bad block table for a given LUN. If the device supports Set Block Table command, the bad block table reflects the current bad block table, while else it represents the initial bad block table of a given LUN.

Command Completion
On command completion, the controller posts a completion queue entry to the Admin Completion Queue indicating the status for the command.
Status byte is set to 1h on error and 0h otherwise.

The data buffer is filled using the following format:

Byte	Description
3:0	Table ID (=”BBLT”)
5:4	Version number
7:6	Revision number
11:8	Reserved

15:12	Total number of blocks in this LUN (Number of blocks per Plane x Number of planes)
19:16	Number of factory (flash manufacturer marked) bad blocks
23:20	Number of grown bad blocks
27:24	Number of device reserved blocks
31:28	Number of host reserved blocks
63:32	Reserved
*:64	List of bad block table entries. One byte for each physical block for bad block information. Non-zero value in bits 3-0 indicates not usable block. Bit field details:

Bit	Description
0	Factory bad block: 0 = Good block, 1 = Bad block
1	Grown bad block: 0 = Good block, 1 = Bad block
2	Device reserved block: 0 = Usable, 1 = Reserved
3	Host reserved block: 0 = Usable, 1 = Reserved
4	Media manager reserved block: 0 = Usable, 1 = Reserved
7:5	Reserved

Byte sequence will be as follows for 2 plane device example:

Byte offset	Physical block
0	Block 0 Plane 0
1	Block 0 Plane 1
2	Block 1 Plane 0
3	Block 1 Plane 1
4	Block 2 Plane 0
…	…

2.1.4 Set Block Information Command and Response

Updates the device bad block table with a list of blocks that will be set to a specific type.

Command Submission

The submission command entry defines a list of PPAs (physical blocks) to be updated. The update value is applied to each block in the PPA List. If host tries to modify non-usable block (already having non-zero value in bad block table entry), the device may reject the command or allow modification depending on implementation.

The command updates the device side bad block table. The blocks are given as a list of PPAs that should be updated.

- **OPC**: F1h
- **CDW10-11 (PPA)**: Physical Page Address or pointer to Physical Page Address List
- **CDW12:**

Bits	Description
31:24	Reserved
23:16	Update value
15:0 NPPA	Number of PPAs (zero-based value).

Command Completion
On command completion, the controller Status Code returns:
- 0h on success.
- 1h on undefined error.
- 2h on not being able to persist the bad block table update.

Set Bad Block Table Completion
On command completion, the controller posts a completion queue entry to the Admin Completion Queue indicating the status of the command.
Status byte is set to 1h on error and 0h otherwise.

2.2 Command Set Commands

The physical page addressing command set defines five media commands.

Opcode (07)	Opcode (06:02)	Opcode (01:00)	NVMe Opcode	O/M	Command
Generic	Function	Data Transfer			
1b	001 00b	00b	90h	M	Physical Block Erase
1b	001 00b	01b	91h	M	Physical Page Address Write

1b	001 00b	10b	92h	M	Physical Page Address Read
1b	001 01b	01b	95h	O	Physical Page Address Raw Write
1b	001 01b	10b	96h	O	Physical Page Address Raw Read

Table 11: Opcodes for Data Commands - PPA Command Set Specific

All commands are implemented uses the regular NVMe command structure, as described in the NVMe 1.2 specification, under the section "Submission Queue Entry – Command Format". Non-specified fields follow the standard submission queue entry fields.

The field CDW[11:10] contains either a physical page address (PPA) or a PPA list. If the number of PPA (NPPA) in the command is equal to 1, PPA (CDW[11-10]) is a physical address. If NPPA is greater than 1, then this field is a 64-bit pointer to a PPA List. A PPA List is an array of 64-bit fields, each of which contains a separate PPA.

The maximum Number of PPA (NPPA) transferred as part of a single media command is 64. This limit is because the Completion Queue Entry DW[1:0] is a 64-bit field and one bit is associated with each PPA as a pass or fail indicator of the access for that PPA.

2.2.1 Physical Erase Command and Response

This command issues the Block Erase command for the block and flash memory chip specified by the PPA field in CDW[11:10]. The page and sector portion of the PPA address is ignored.

- **OPC**: 90h
- **NSID**: Namespace ID.
- **CDW10-11 (PPA)**: Single physical page address (PPA) or pointer to a Physical Page Address List
- **CDW12:**

Bits	Description
31:16	Reserved
05:0 NPPA	Number of PPAs (zero-based value).

- **CDW13:**

Bits	Description
31:0	Reserved

Physical Block Erase Completion

When the command completes, the controller posts a completion queue entry to the User Completion Queue indicating the status for the command.

Status Field and Status Code follows the error codes defined in Section 2.3.

2.2.2 Physical Write Command and Response

The Write PPA command writes data and metadata to the PPAs indicated.

The NVMe submission entry is as following:

- **OPC**: 91h
- **CDW10-11 (PPA)**: Single physical page address (PPA) or pointer to a Physical Page Address List
- **CDW4-5**: If metadata pointer is given, this is stored within the OOB area of the flash page.
- **CDW12:**

Bits	Description
31 LR	Limited retry If set the controller should apply limit retry efforts. If cleared, the controller should apply all available means to complete the operation.
30 FUA	Force Unit Access For a write, this field specifies that the data shall be written to non-volatile media before indicating command completion. There is no implied order with respect to other commands.
29:26	Reserved
25	Scramble Disable When this bit is set, the scramble function will be disabled.
24	SLC Mode Enable When this bit is set, it indicates that the Flash memory block accessed should be done in SLC mode. Otherwise, will be in default MLC mode or TLC mode.
23	Suspend Enable For a Read PPA command, when this bit is set, it enables Flash memory suspend or Erase suspend function for read to get lower latency access.
22:18	Reserved
17:16	Flash memory plane operation mode 0 – Will erase/write/read the flash memory using single plane command. 1 – Will erase/write/read the flash memory using dual plane command. 2 – Will erase/write/read the flash memory using quad plane command. 3 – Reserved
15:6	Reserved
05:0	Number of PPAs (zero-based value).

NPPA	

Physical Page Write Address Completion
Status Field and Status Code follows the error codes defined in Section 2.3.

When using the Write PPA command, Completion Entry Dwords 0 and 1 are used to return status from the flash memory. The total number of PPA Data Units in one Write PPA command is limited to 64 because there are only 64 bits of return status possible.

2.2.3 Physical Read Command and Response

Physical Page Address Read follows the same command specification as Physical Page Address Write.

2.2.4 Physical Write Raw Command and Response

This command writes data plus additional metadata to flash memory without built-in hardware calculation of ECC bits. Instead, the ECC bits to write are pointed to by the Metadata Pointer (CDW[5:4]). The command reuses the Write PPA format with changes to CDW12.
Write PPA Raw provides high-level software direct access to the flash memory for flash memory characterization and diagnostic purposes.

- **OPC**: 95h
- **CDW4-5**: If metadata pointer is given, this is stored within the OOB area of the flash page.
- **CDW10-11 (PPA)**: Single physical page address (PPA) or pointer to a Physical Page
- **CDW12:**

Bits	Description
31 LR	Limited retry If set the controller should apply limit retry efforts. If cleared, the controller should apply all available means to complete the operation.
30 FUA	Force Unit Access For a write, this field specifies that the data shall be written to non-volatile media before indicating command completion. There is no implied order with respect to other commands.
29:26	Reserved
25	Scramble Disable When this bit is set, the scramble function will be disabled.
24	SLC Mode Enable When this bit is set, it indicates that the Flash memory block accessed should

	be done in SLC mode. Otherwise, will be in default MLC mode or TLC mode.
23	Suspend Enable For a Read PPA command, when this bit is set, it enables Flash memory suspend or Erase suspend function for read to get lower latency access.
22:18	Reserved
17:16	Flash memory plane operation mode 0 – Will erase/write/read the flash memory using single plane command. 1 – Will erase/write/read the flash memory using dual plane command. 2 – Will erase/write/read the flash memory using quad plane command. 3 – Reserved
15:6	Reserved
05:0 NPPA	Number of PPAs (zero-based value).

Raw Physical Page Address Write Completion
Follows the standard specification. If the PBA is 0, an LBA_RANGE error should be returned.

2.2.5 Physical Read Raw Command and Response

Raw Physical Page Address Read follows the same command specification as Raw Physical Page Address Write.

2.3 Command Completion Error Codes

When a command completes, the command is returned using a completion entry. This entry contains a Status Code (SC) and a Status Field (SF). These two fields are used to communicate all types of errors.

The standard Status Codes and Status Field is defined in the NVMe 1.2 specification Section 4.6. Additional opcodes are defined in the 0x80h-BFh opcode range.

2.3.1 Status Field

The Status Field defines the status of the command indicated in the CQ entry. A value of 0h for the Status Field indicates a successful command completion, with no fatal or nonfatal error conditions.

2.3.2 Error Codes

The following error codes follow the already defined error codes in Section 4.6.1 of the NVMe specification.

Error Code	Description
80h	Write Fault The write data could not be committed to the media. This may be due to a lack of available spare locations that is reported as an asynchronous event.
81h	Unrecovered Read Error The read data could not be recovered from the media.

2.4 Device Translation Map SSD Commands

An Open-Channel SSD may implement translation mapping on the device, as well as on the host. The device-side translation map is not mandatory for Open-Channel SSDs. Early versions of this specification held these commands but is not necessary for future operation. Please note that the Hybrid Page Address Command is fold into the Physical Page Write, with the LBA communicated as well.

2.4.1 Get Logical to Physical Translation Table (EAh)

Request a range of the logical to physical translation table, identified by a starting block address (SLBA) and a 16 bit value (NPPA) denoting the number of blocks whose entries are requested.

- **OPC**: EAh

- **NSID**: Namespace id
- **PRP1**: Address to DMA memory region in which to store the response.
- **CDW10-11**: The starting logical block address (SLBA) of the request.
- **CDW12 :** The number of logical blocks (64b L2P entries) requested.
- **CDW13-15**: Reserved

Note: Each table entry is 4B in size, indexing a 4K data region.

Get Logical to Physical Translation Table Completion

When the command is completed, the controller posts a completion queue entry to the Admin Completion Queue indicating the status for the command.

Status byte is set to 1h on error and 0h otherwise.

2.4.2 Hybrid Page Address Write (81h)

Commands that are the unit size of the namespace are issued using the standard write and read opcodes with the following modifications:

The logical address is mapped to the SLBA field.

- **OPC**: 81h
- **PRP1**: Address to DMA memory region with data
- **SLBA**: Physical Page Address or pointer to Physical Page Address List
- **CDW4-5**: If metadata pointer is given, this is stored within the OOB area of the flash page.
- **CDW10-11:**
    - If NPPA equals zero: Physical Page Address.
    - If NPPA is larger than zero: Pointer to a continuous physical buffer with logical to physical address pairs. Both logical and physical address is zero-indexed.
- **CDW12:**

Bits	Description
31 LR	Limited retry If set the controller should apply limit retry efforts. If cleared, the controller should apply all available means to complete the operation.
30 FUA	Force Unit Access For a write, this field specifies that the data shall be written to non-volatile media before indicating command completion. There is no implied order with respect to other commands.
29:26	Reserved
25	Scramble Disable When this bit is set, the scramble function will be disabled.

24	SLC Mode Enable When this bit is set, it indicates that the Flash memory block accessed should be done in SLC mode. Otherwise will be in default MLC mode or TLC mode.
23	Suspend Enable For a Read PPA command, when this bit is set, it enables Flash memory suspend or Erase suspend function for read to get lower latency access.
22:18	Reserved
17:16	Flash memory plane operation mode 0 – Will erase/write/read the flash memory using single plane command. 1 – Will erase/write/read the flash memory using dual plane command. 2 – Will erase/write/read the flash memory using quad plane command. 3 – Reserved.
15:6	Reserved
05:0 NPPA	Number of PPAs (zero-based value).

- **CDW14-15 (SLBA)**: Start Logical Block Address.

Hybrid Page Address Write Completion

The maximum number of PPA data units transferred as part of a single command is 64. This limit is due to that we use 1 bit in the Completion queue DW[1:0] that write physical page address command uses to indicate each write PPA data unit pass or fail.

2.4.3 Hybrid Page Address Read (02h)

The hybrid read follows the same specification as the standard NVMe Read LBA 02h command.

The device controller should use its internal mapping table to lookup the physical pages to be read and return them to the host.