PI Introduction

- based on NVMe 1.4 + 1.4 TP

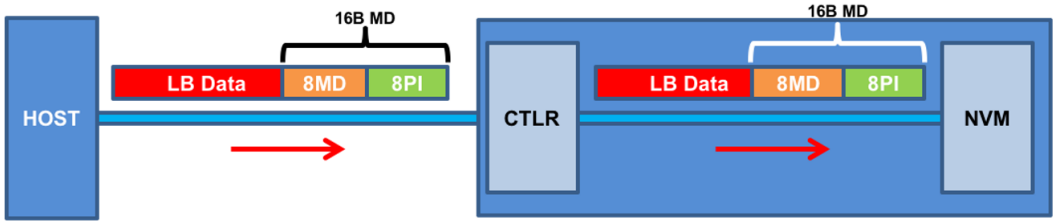
Jerry Hsu

2019.12.16

Outline

- Overview
- PI Control in Commands
- Commands with PI
  - Read / Write
  - Verify
  - Dataset Management
  - Write Zeros
  - Compare
  - Fused Operation
  - Sanitize
  - Zone Append
  - Copy
- PI Handling Attribution
- Appendix: PI using DMAC

Overview



- End-to-end Data Protection
  - application to the NVM media and back to the application itself
  - additional protection information is added to LB
  - evaluated by the controller and/or host software
- End-to-end Data Protection Type Settings (DPS)
  - PI types (1-3), PI location (first/last 8B in MD)
  - Create/Format NS
- Protection Information (PI)
  - Guard (2B)
    - CRC-16
    - based on PI location to determine if CRC covers MD (w/o PI)
  - Application Tag (2B)
    - not interpreted by the controller
    - may be used to disable checking of PI
  - Reference Tag (4B)
    - associate LB data with an address and protect against misdirected or out-of-order logical block transfer
    - may be used to disable checking of PI

PI Control in Commands

- CDW12.Protection Information Field (PRINFO)
  - Bit 03- Protection Information Action (PRACT)
    - action to take for the PI
  - Bits 02:00- Protection Information Check (PRCHK)
    - fields that need to be checked

PRACT

PRACT Value	Metadata Size	Description
1b	8 Bytes	The protection information is stripped (read) or inserted (write).
1b	> 8 Bytes	The protection information is passed (read) or replaces the first or last 8 bytes of the metadata (write).
0b	any	The protection information is passed (read and write).

PRCHK

Bit	Definition
02	If set to '1' enables protection information checking of the Guard field. If cleared to '0', the Guard field is not checked.
01	If set to '1' enables protection information checking of the Application Tag field. If cleared to '0', the Application Tag field is not checked.
00	If set to '1' enables protection information checking of the Logical Block Reference Tag field. If cleared to '0', the Logical Block Reference Tag field is not checked.

PI Control in Commands

- CDW14.(Expected) Initial Logical Block Reference Tag ((E)ILBRT)
  - For computing reference tag
  - Based on protection type

- CDW15.(Expected) Logical Block Application Tag Mask ((E)LBATM)
- CDW15.(Expected) Logical Block Application Tag ((E)LBAT)
  - For checking/inserting Application Tag

Read / Write Command with PI

- 1. Determine MD Composition & Action
  - based on CMD.PRACT + MD Size

CMD Type	CMD.PRACT	MD Size (Byte)	PI From	Metadata Composition & Action
Write	0	8	host	MD = PI; PI Check
Write	0	> 8	host	the first or last 8B of MD = PI; PI Check
Write	1	8	HIP	MD = PI; PI Appending
Write	1	> 8	HIP	the first or last 8B of MD = PI; PI Overwrite
Read	0	>= 8	Flash	PI Check and return MD to the host
Read	1	> 8	Flash	PI Check and return MD to the host
Read	1	8	Flash	PI Check and not return MD

General case but with exception
e.g PI in the deallocated data

***Inserted PI (Append/Overwrite)**:

a. **Guard field**: computed CRC16
b. **Application Tag field**: CMD.LBAT
c. **Reference Tag field**: computed reference tag (based on Protection Type)

Read / Write Command with PI

- 2. PI Check case in step 1
  - Further consider CMD.PRCHK
  - a. PRCHK bit 02 = 1: PI.Guard vs. Computed CRC16
  - b. PRCHK bit 01 = 1: PI.AT & CMD.(E)LBATM vs. CMD.(E)LBAT
  - c. PRCHK bit 00 = 1: PI.RT vs. Computed RT

***Computed RT**

Protection Type	Computed RT	CMD.(E)ILBRT Check
Type 1	Increase from CMD.(E)ILBRT	o, check with CMD.SLBA*
Type 2	Increase from CMD.(E)ILBRT	x
Type 3	Check RT: Abort CMD or ignore ; Insert: ignore or insert CMD.(E)ILBRT for each LB	

* least significant four bytes of the LBA

Read / Write Command with PI

- 3. Disable PI Check
  - regardless of the state of the PRCHK field in the command
  - all protection information checks are disabled when
    - Type 1 or 2: PI.AT has a value of all Fh
    - Type 3: Both PI.AT and PI.RT have the value of all Fh

Verify Command with PI

- Same as Read Command

Dataset Management Command with PI

- Value return while read/verify/compare/copy after deallocate
  - value returned by subsequent reads of that LB data shall be the same
  - based on DLFEAT field in the Identify NS Data structure
- **a. LB Data + MD (exclude PI)**
  - DLFEAT[2:0] = 010b: all bytes set to FFh
  - DLFEAT[2:0] = 001b: all bytes cleared to 0h
  - DLFEAT[2:0] = 000b: Not reported
    - e.g. last written data
- **b. PI**
  - PI.Guard field
    - DLFEAT[3] = 0b: FFFFh
    - DLFEAT[3] = 1b: CRC of LB Data +MD (exclude PI)
  - PI.AT / PI.RT
    - all Fh
    - **all the PI shall not be checked**

Read Unwritten LBA

- Read Unwritten LBA
  - equal to Read Deallocated LBA
- Issue: Write non-aligned LBs + Unwritten LBs
  - Unwritten LB + MD
    - cleared to zeros
  - Unwritten PI
    - same as Dataset Management command

Write Zeros Command with PI

- Set a range of logical blocks to zero **without input data**
- Non-PI related metadata for this command
  - shall be all bytes cleared to 0h
- PI
  - CMD.PRCHK shall be 0h
    - since there is no input PI
  - CMD.PRACT = 0b: clear all PI to all 0h
    - cannot use trim under current settings
  - CMD.PRACT = 1b:
    - insert PI based on DPS, CDW15.EILBRT, CDW15.ELBATM, and CDW15.ELBAT fields

Write Zeros Command with PI

- CMD.PRACT = 1b

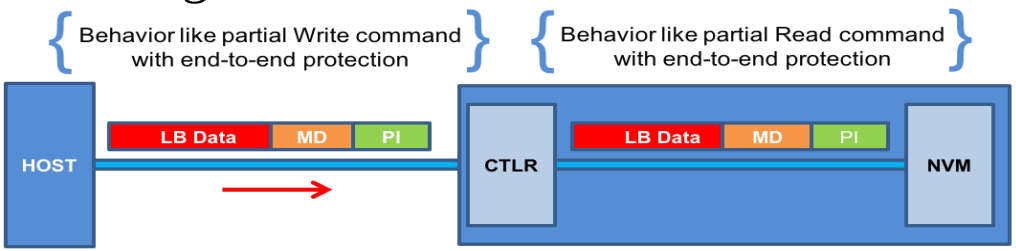
- a. CMD.DEAC = 1b & DLFEAT[2:0] = 001b (LB = 0h)
  - should deallocate that LB
  - shall return LB Data + MD with all 0h
  - shall return PI as Dataset Management cmd
- b. CMD.DEAC = 0b & DLFEAT[2:0] = 001b
  - may deallocate that LB
  - shall return LB Data + MD as 0
  - shall insert PI based on DPS, CDW15.ILBRT, CDW15.LBATM, and CDW15.LBAT fields
- **c. DLFEAT[2:0] != 001b**
  - shall not deallocate that LB
  - insert PI based on DPS, CDW15.ILBRT, CDW15.LBATM, and CDW15.LBAT fields

Summary: Trim(o): a ;Trim(x): b, c

Compare Command with PI

- Comparison performed for LB data and MD (exclude PI)
- PRACT shall be zero (PI is from the host)
- Partial Write
  - Transferred PI shall be checked since PRACT = 0
  - Check PI based on DPS, CDW15.EILBRT, CDW15.ELBATM, and CDW15.ELBAT fields to check
- Partial Read
  - Check PI based on DPS, CDW15.EILBRT, CDW15.ELBATM, and CDW15.ELBAT fields
  - compare the PI in the reading MD

Only compare Transferred PI to PI in the reading MD? (x) – may disable checking of transferred PI (PI.AT, PI.RT = Fh)



Protection Information with PRACT bit set to '0' (i.e., pass)

Fused Operation with PI

- Same as those for the individual commands
  - compare + write

Sanitize Command with PI

- Sanitize Operation Types (CMD.Sanitize Action)
  - Block Erase
    - alter user data with a low-level block erase
  - Crypto Erase
    - alter user data by changing the media encryption key
  - Overwrite
    - alter user data by writing a fixed data pattern or related patterns
    - media specific and may not be appropriate for all media types.
      - For example, if the media is NAND, multiple pass overwrite operations may have an adverse effect on media endurance.

Sanitize Command with PI

- Sanitize Enhancements (added in NVMe 1.4)
  - Sanitize operation with additional media modification operation
    - Making all LBA contents readable and sanitize results observable
  - NVMe 1.3
    - CMD.No Deallocate After Sanitize (1 bit)
  - Enhancement
    - Sanitize Capabilities (Identify Controller)
      - No-Deallocate Modifies Media After Sanitize (NODMMAS)
        - whether modify after Sanitize
        - not defined (compliant with 1.3 or earlier), not additionally modified, additionally modified
      - No-Deallocate Inhibited (NDI): if support CMD.No Deallocate After Sanitize
- Controller Behavior
  - CMD.No Deallocate After Sanitize + NDI to determine if no deallocate
  - NODMMAS for actions after no deallocate
  - Purpose: Making all LBA contents readable and sanitize results observable
    - CMD.No Deallocate After Sanitize = 1 & NDI = 0 & NODMMAS = 10b

Sanitize Command with PI

- Sanitize Operation- User Data Values
  - a. Deallocate (Return user data value as trim)
    - CMD.No Deallocate After Sanitize = 0b or
    - CMD.No Deallocate After Sanitize = 1b & NDI = 1b
      - & Feature: No-Deallocate Response Mode = 1b (warning and completed successfully with deallocation)
  - b. No Deallocate
    - CMD.No Deallocate After Sanitize = 1b & NDI = 0b

Figure 481: Sanitize Operations – User Data Values

Sanitize Operation	Logical Blocks	Non-PI Metadata[1]	Protection Information[2]
Block Erase	Vendor specific value	Vendor specific value	Vendor specific value
Crypto Erase	Indeterminate	Indeterminate	Indeterminate
Overwrite	Refer to Figure 480	Refer to Figure 480	Refer to Figure 480

Sanitize Command with PI

- Sanitize Operation- Overwrite Mechanism
  - CMD.Overwrite Pattern (OVRPAT): 4B pattern
  - CMD.Overwrite Pass Count:
    - # of overwrite passes = how many times the media is to be overwritten
    - value in the end: LB Data + Non-PI MD = OVRPAT; PI = all Fh
  - CMD. Overwrite Invert Pattern Between Passes (OIPBP)
    - 1: invert pattern between pass; 0: not invert

OIPBP[1]	**Overwrite Pass Count**[1]	**Overwrite Pass Number**	**Logical Block Data and Non-PI Metadata**[2]	**Protection Information**[3]
'0'	All	All	Overwrite Pattern[1]	FFFFFFFF_FFFFFFFFh
'1'	Even	First	Inversion of Overwrite Pattern[1]	00000000_00000000h
		Subsequent	Inversion of Overwrite Pattern[1] from previous pass (i.e., each bit XORed with '1')	
'1'	Odd	First	Overwrite Pattern[1]	FFFFFFFF_FFFFFFFFh
		Subsequent	Inversion of Overwrite Pattern[1] from previous pass (i.e., each bit XORed with '1')	

Zone Append Command with PI

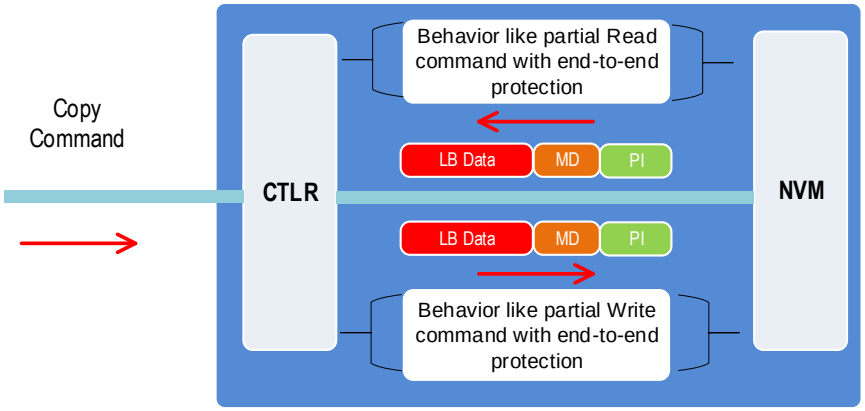
- Similar to Write Command
- 1. Determine MD Composition & Action
  - based on CMD.PRACT
- 2. PI Check case in step 1
  - Further consider CMD.PRCHK
- 3. Disable PI Check

***Computed RT**

Protection Type	Computed RT	CMD.ILBRT Check
Type 1	Increase from 1st actual LBA	o, check with CMD.ZSLBA
Type 2	Increase from CMD.(E)ILBRT	x
Type 3	Insert: ignore or insert CMD.ILBRT for each LB	

Copy Command with PI

- Copy data from one or more source logical block ranges to a single consecutive destination logical block range
  - Partial Read (NVM to Controller)
    - Check PI based on CMD.PRINFOR
    - EILBRTs, EILBATMs, EILBATs from Source Range Entries (DPTR)
  - Partial Write (Controller to NVM)
    - based on PRINFOW
    - CMD.ILBRT, CMD.LBATM and CMD.LBAT



Protection Information with PRACT bit cleared to '0' (i.e., pass)

Protection shall be overlapped

Copy Command with PI

- Protection Information Processing

Guard Field may not be checked since it has checked in the Partial Read

R.PRACT	W.PRACT	Action
0	0	R: PI Check; W: PI Check and not update PI
1	1	R: PI Check; W: PI Insert
0	1	Abort command
1	0	Abort command

PI Handling Attribution

Command	PI Insert	PI Check
Write	HIP	HIP
Read	*	HIP
Verify	*	HIP
Write Zero	FW	N/A
Compare	N/A	W: HIP; R: FW
Sanitize	FW	N/A
Dataset Management (Trim)	FW	N/A
Fused Operation	C: N/A; W: HIP	C: HIP/FW; W: HIP
Zone Append	HIP	HIP
Copy	W: FW	W:FW; R: FW

*FW: CoP1 or R5

Appendix: PI using DMAC

- Suppose the PI value returned after Deallocated:
  - a. DLFEAT[3] = 0, i.e. PI.Guard returns FFFFh
  - b. PI.AT, PI.RT = all Fh

Command	Expected PI Value	DMAC Function
Trim	PI = all Fh	SV
Write Zero	a. CMD.PRACT = 0b: clear PI to 0h b. CMD.PRACT = 1b: b.1 CMD.DEAC = 1b & DLFEAT[2:0] = 001b: PI = all Fh b.2 CMD.DEAC = 0b & DLFEAT[2:0] = 001b: PI = insert PI b.3 DLFEAT[2:0] != 001b: PI = insert PI	a. SV b.1 SV b.2 Note b.3 Note
Compare	a. Partial Write: check PI (may be checked by HIP) b. Partial Read: check PI	a. Note b. Note
Sanitize	a. Deallocate: PI = all Fh b. No Deallocate b.1 Block Erase- PI: Vendor Specific b.2 Crypto Erase-PI: Indeterminate b.3 Overwrite-PI: all 0h or Fh based on odd or even pass	a. SV b.1 SV b.2 SV b.3 SV
Fused Operation	as compare + write	as compare + write
Copy	a. Partial Read: check PI b. Partial Write: check PI/insert PI	a. Note b. Note

Appendix: PI using DMAC

- Note:
  - insert/check PI[1]

[1] 另一種做法是: 用來check 的PI全部算完後，可與MD裡的PI用DMAC的CMP function做compare (compare data size最小8B)。若有compare error，可由cmp_err_offset得知是哪個field錯。
[2] HIP目前做法- check: ignore; insert: insert CMD.(E)ILBRT for each LB，看FW是否要一致。

Field	Value	DMAC Function
Guard	CRC16 based on DPS	insert: CRCPI + SV check: CRCPI + SRH
AT	1. insert: CMD.(E)LBAT 2. check: a. PI.AT & CMD.(E)LBATM b. CMD.(E)LBAT	1. SV 2. SRH
RT	1. Protection Type 1: a. check CMD.(E)ILBRT with CMD.SLBA b. increase from CMD.(E)ILBRT 2. Protection Type 2: - increase from CMD.(E)ILBRT 3. Protection Type 3[2]: - check: ignore or abort cmd - insert: ignore or insert CMD.(E)ILBRT for each LB	1.a SRH 1.b SV or INC 2. SV or INC 3. insert: SV