

# NVMe IP Core for Agilex

June 2, 2021

Product Specification

Rev1.0



## Design Gateway Co.,Ltd

E-mail: ip-sales@design-gateway.com

URL: design-gateway.com

## Features

- Access one NVMe Gen4 SSD without CPU and external memory
- Include 256 Kbyte RAM to be data buffer
- Simple user interface by dgIF typeS
- Support six commands, i.e., Identify, Shutdown, Write, Read, SMART, and Flush
- Supported NVMe device
  - Base Class Code:01h (mass storage), Sub Class code:08h (Non-volatile), Programming Interface:02h (NVMHCI)
  - MPSMIN (Memory Page Size Minimum): 0 (4Kbyte)
  - MDTS (Maximum Data Transfer Size): At least 5 (128 Kbyte) or 0 (no limitation)
  - LBA unit: 512 byte or 4096 byte
- User clock frequency: More than or equal to a half of PCIe clock frequency (250 MHz for Gen4)
- PCIe Hard IP: P-Tile Avalon-ST Intel Hard IP for PCIe IP Core (128-bit interface for Gen4 speed)
- Available reference design
  - Agilex F-Series development board with AB18-PCIeX16 adapter board
- Customized service for following features
  - Additional NVMe commands
  - RAM size modification

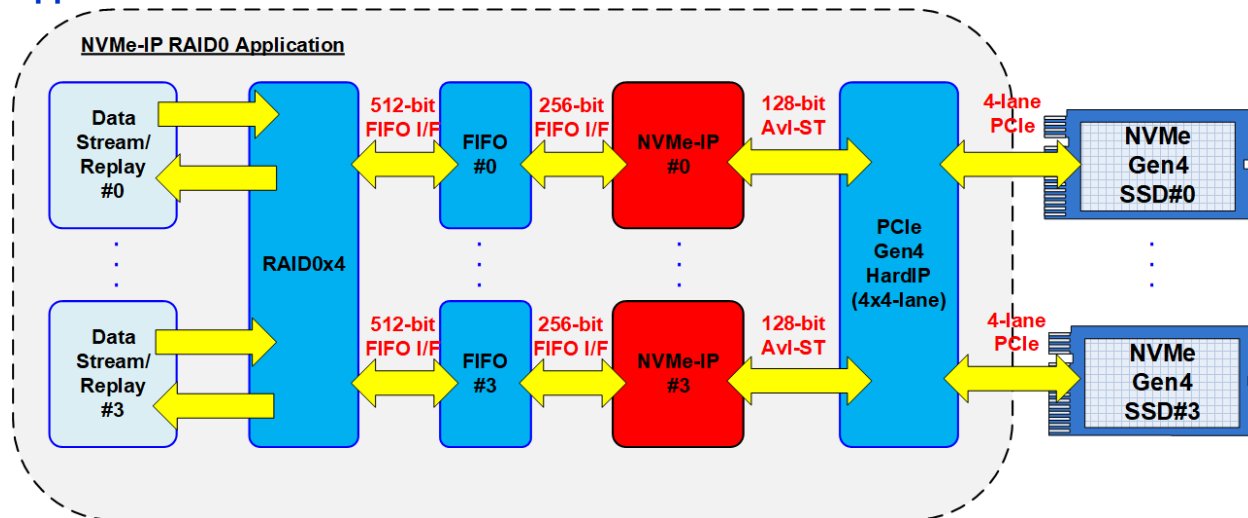
Core Facts	
Provided with Core	
Documentation	Reference Design Manual Demo Instruction Manual
Design File Formats	Encrypted hdl File
Instantiation Templates	VHDL
Reference Designs & Application Notes	QuartusII Project, See Reference Design Manual
Additional Items	Demo on Agilex F-Series development kit
Support	
Support provided by Design Gateway Co., Ltd.	

**Table 1: Example Implementation Statistics**

Family	Example Device	Fmax (MHz)	Logic utilization (ALMs)	Registers	Pin	Block Memory bit	Design Tools
Agilex F-Series	AGFB014R24A2E2VR0	450	4167	9641	-	2,205,440	QuartusII 20.4

Notes: Actual logic resource dependent on percentage of unrelated logic

## Applications



**Figure 1: NVMe IP Application**

NVMe IP Core integrated with P-Tile Avalon-ST Intel Hard IP for PCIe IP Core (PCIe Gen4 hard IP) is ideal to access NVMe Gen4 SSD without CPU and external memory such as DDR. 256 Kbyte buffer implemented by Block Memory is included in NVMe IP Core to be data buffer between user logic and NVMe Gen4 SSD. One PCIe hard IP can be configured for connecting with four NVMe-IPs and four NVMe Gen4 SSDs by using 4-lane PCIe interface. Therefore, 4-ch RAID0 system can be designed to increase transfer speed up to four times of one SSD performance, as shown in Figure 1. Besides, the total storage capacity of RAID0 system is increased to four times. Now one Gen4 SSD write/read performance is about 6000 Mbyte/sec. Using four Gen4 SSDs can achieve up to 24 Gbyte/sec.

We also provide alternative IP cores for more specific applications such as Random access, PCIe switch, and Soft IP core.

**Random Access NVMe IP Core** – To access NVMe SSD with multiple commands, individual address for each. Recommended for the application which requires access in non-continuous area, stream data recording without specified length or application that multiple users can access one NVMe SSD.

[https://dgway.com/raNVMe-IP\\_A\\_E.html](https://dgway.com/raNVMe-IP_A_E.html)

**NVMe IP Core for PCIe Switch** – For access one or multiple NVMe SSDs via PCIe switch.

[https://www.dgway.com/NVMe-IP\\_A\\_E.html](https://www.dgway.com/NVMe-IP_A_E.html)

**NVMe IP Core with PCIe Soft IP** – When the selected FPGA does not have enough PCIe hard IP for the application.

[https://www.dgway.com/NVMeG4-IP\\_A\\_E.html](https://www.dgway.com/NVMeG4-IP_A_E.html)

### General Description

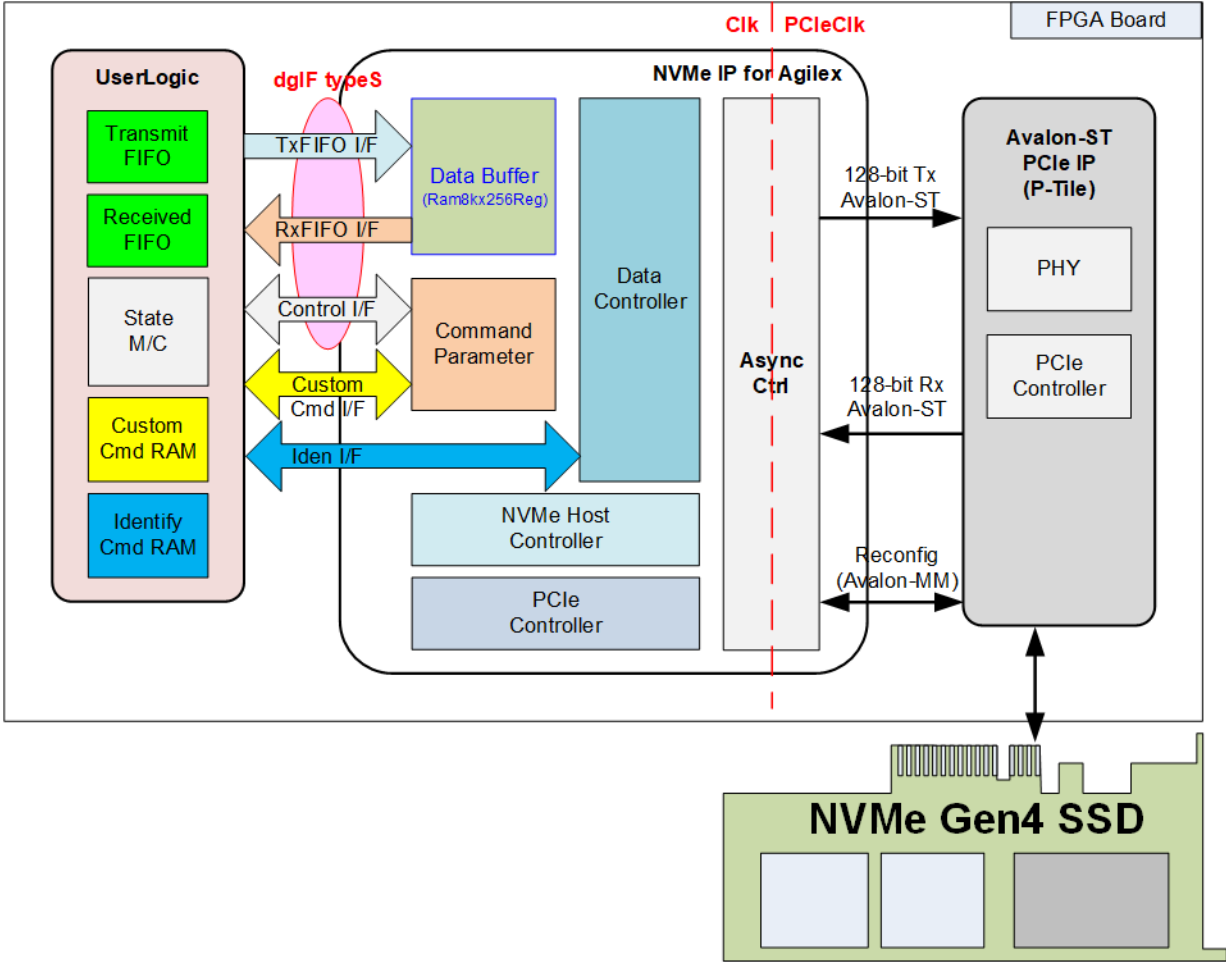


Figure 2: NVMe IP block diagram

NVMe IP implements as host controller to access NVMe SSD following NVM express standard. Physical interface of NVMe SSD is PCIe which some parts are implemented by Avalon-ST Intel Hard IP for PCIe IP Core (P-Tile).

NVMe IP supports six NVMe commands, i.e., Identify, Shutdown, Write, Read, SMART, and Flush command by using two user interface groups. First is Control interface for transferring command and the parameters. Another is Data interface for transferring data when the command must have the data transferring. Control interface and Data interface for Write/Read command use dgIF typeS format. Control interface of dgIF typeS consists of start address and transfer length with asserting the request signal while Data interface of dgIF typeS is the FIFO interface.

SMART and Flush command are Custom command which use Ctm I/F for control path and Ctm RAM I/F for data path. Identify command uses the same Control interface as Write or Read command, but uses its own data interface - Iden I/F, as shown in Figure 2.

During running initialization process or operating some commands, error signal may be asserted by NVMe IP if some abnormal conditions are found. The IP includes the error status to check more details of error condition. To recover error status, NVMe IP and SSD must be reset.

There is one limitation about clock frequency of user logic. Transmit packet to PCIe hard IP must be sent continuously until end of packet. Therefore, data must be valid every clock between start of frame and end of frame. To support this feature, user logic clock frequency must be more than or equal to a half of PCIe clock frequency. Therefore, the bandwidth on user side is higher than or equal to PCIe hard IP bandwidth. When running PCIe Gen4 speed, user logic clock must be more than or equal to 250MHz (PCIe clock frequency is set to 500MHz).

The reference design on FPGA evaluation boards are available for evaluation before purchasing.

## Functional Description

Figure 3 shows the operation flow of NVMe IP after IP reset is de-asserted. There are three phases, i.e., IP initialization, Operating command, and Inactive status.

After finishing IP initialization, the first command that user runs must be Identify command to check device status and capacity. After finishing all operations, the last command before shutdown system should be Shutdown command for safety operation.

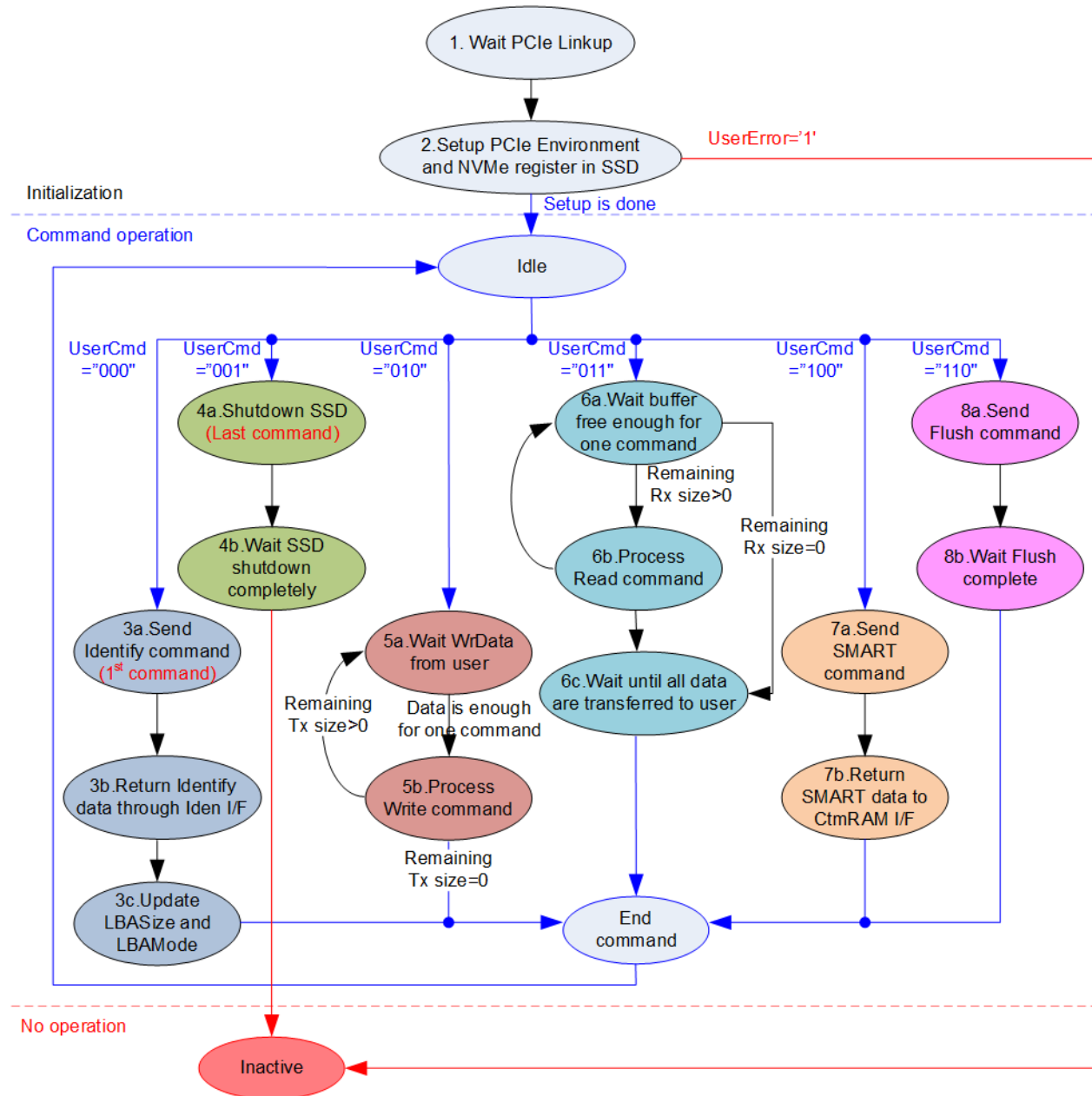


Figure 3: NVMe IP operation flow

The operation flow of NVMe IP is described as follows.

- 1) IP waits until PCIe is ready by monitoring Linkup status from PCIe IP core.
- 2) IP begins the initialization process by configuring PCIe and NVMe registers. After that, the IP enters to the Idle state to wait for a new command request from user. If some errors are detected during initialization process, the IP changes to the Inactive state with asserting UserError to '1'.
- 3) The first command from user must be Identify command (UserCmd="000") to update LBASize (disk capacity) and LBAMode (LBA unit=512 byte or 4 Kbyte).
- 4) The last command before power down the system must be Shutdown command (UserCmd="001"). This command is recommended to guarantee SSD powered down in a good sequence. Without Shutdown command, Write data in SSD cannot be guaranteed. After finishing Shutdown command, NVMe IP and SSD change to the Inactive state. The new command cannot be operated until the IP is reset.
- 5) For Write command (UserCmd="010"), the maximum data size of one command is 128 Kbyte. If total length from user is more than 128 Kbyte, the IP repeats step 5a) – 5b) automatically until total data are completely transferred.
  - a) The IP waits until Write data, sent by user, is enough for one command (transfer size of one command in NVMe IP is 128 Kbyte, except the last loop which could be less than 128 Kbyte).
  - b) The IP sends Write command to SSD and then waits until the status is returned from SSD. The IP changes to the Idle state when total data are completely transferred. Otherwise, the IP goes back to step 5a) to send the next Write command.
- 6) Similar to Write command, when running Read command (UserCmd="011") which has the transfer size more than 128 Kbyte, the IP must repeat step 6a) – 6b) many times.
  - a) If remaining transfer size is equal to zero, the IP skips to step 6c). Otherwise, the IP waits until free space of data buffer in NVMe IP is enough for one command (128 Kbyte or remaining transfer size for the last loop).
  - b) The IP sends Read command to SSD and then returns to step 6a).
  - c) IP waits until all data are completely transferred from data buffer to user logic and then changes to the Idle state. Therefore, data buffer is empty after finishing Read command.
- 7) For SMART command (UserCmd="100"), 512-byte data is returned after finishing the operation.
  - a) IP sends Get Log Page command to read SMART/Health information from the SSD.
  - b) 512-byte data is returned from the SSD. The IP forwards the data through Custom command RAM interface (CtmRamAddr=0x000 – 0x01F).
- 8) For Flush command (UserCmd="110"), there is no data transferring during the operation.
  - a) IP sends Flush command to the SSD.
  - b) IP waits until SSD returns status to complete the operation.

To design NVMe host controller, NVMe IP implements two protocols, i.e., NVMe protocol for interfacing with user and PCIe protocol for interfacing with PCIe hard IP. Figure 2 shows the hardware inside NVMe IP which can split into two groups, NVMe and PCIe. More details of each module are described as follows.

## NVMe

Six commands that NVMe supports can be split in two command types - Admin command and NVM command. Admin command consists of three commands - Identify, Shutdown, and SMART command while NVM command consists of three commands - Write, Read, and Flush command. After finishing operating the command, the status returned from the SSD is latched to AdmCompStatus (status returned from Admin command) or IOCompStatus (status returned from NVM command), depending on the command types.

The parameters of Write or Read command are set by Control interface of dgIF typeS while the parameters of SMART or Flush command are set by CtmSubmDWXX of Ctm interface. Data interface for Write or Read command is transferred by FIFO interface, a part of dgIF typeS. The data of Write and Read command is transferred with 256 Kbyte buffer inside the IP. The data interface of other commands has its own interface - Identify RAM for Identify command and Custom RAM for SMART command.

The details of each submodule are described as follows.

- **NVMe Host Controller**

NVMe host controller is the main controller in NVMe IP. Similar to the operation flow, the controller has two phases for operating. First is the initialization phase which is once run after the system is boot up for setting NVMe register inside the SSD. After finishing the initialization phase, the next phase is operating the command. The controller controls the order of transmitted packet and received packet for each command.

To start operating each command, the parameters of the command are latched to Command Parameter for creating the packet. After that, the packet is forwarded to AsyncCtrl to convert from NVMe packet format to be PCIe packet format. After finishing operation the command, the status packet is returned from SSD. The controller decodes the status value and decide that the operation is complete or error. When operating the command that needs to transfer the data such as Write command and Read command, the controller must handle the order of data packet that is created and decoded by Data controller.

- **Command Parameter**

This module creates Command packet sent to SSD. Also, the status packet returned from SSD is decoded by this module. However, the input/output of this module are controlled by the NVMe host controller. Typically, the command consists of 16 Dwords (1 Dword = 32-bit). When running Identify, Shutdown, Write, and Read command, all 16 Dwords are created by Command parameter, depending on the user inputs to dgIF typeS. When running SMART and Flush command, all 16 Dwords are directly loaded via CtmSubmDW0-CtmSubmDW15 of Ctm interface.

- **Data Buffer**

256 Kbyte simple dual port RAM is implemented by memory block to be data buffer. The buffer stores data transferring between UserLogic and SSD during operating Write and Read command.

- **Data Controller**

This module is operated when the command must transfer the data, i.e., Identify, SMART, Write, and Read command. There are three data interfaces for transferring with the SSD - FIFO interface with 256 Kbyte buffer when running Write or Read command, Custom command RAM interface when running SMART command, and Identify interface when running Identify command. The data packet is created and decoded by this module. Similar to Command Parameter module, Data controller input and output signals are controlled by the NVMe host controller.

## **PCIe**

The PCIe standard is the outstanding low-layer protocol for very high speed application. The NVMe standard is the protocol which is run over PCIe protocol. In the initialization process, NVMe layer is setup after finishing PCIe layer setup. Two modules are designed to support PCIe protocol - PCIe controller and AsyncCtrl. More details of each module are described as follows.

- **PCIe Controller**

During initialization process, PCIe controller sets up PCIe environment of SSD via Reconfig port which is Avalon-MM interface. After that, PCIe packet is created and decoded via 128-bit Tx/Rx Avalon-Stream. The command packet and the data packet from NVMe module are converted to be PCIe packet by PCIe controller and vice versa.

- **AsyncCtrl**

AsyncCtrl includes asynchronous registers and asynchronous buffers with asymmetric data width to support clock domain crossing and different data width. The data width on user interface is two times of the data width on PCIe interface. Therefore, user clock frequency can be equal to or more than a half value of PCIe clock frequency for balancing the data bandwidth. Most logics in NVMe IP run on user clock domain while PCIe hard IP runs on PCIe clock domain.



## User Logic

This module could be designed by using small state machine to send the commands and the parameters for each command. For example, the address and transfer size which are the parameters for Write or Read command are designed by using simple registers. While FIFO is connected for transferring data in Write and Read command. The data output of SMART command and Identify command are connected to simple dual port RAM with byte enable. The data width of FIFO and RAM are 256-bit while the memory depth can be set by different value. Data size of Identify command is 8 Kbytes while data size of SMART command is 512 bytes.

## Avalon-ST Intel Hard IP for PCIe (P-Tile)

To connect with NVMe IP, three interfaces of Avalon-ST Intel Hard IP for PCIe (P-Tile) are applied, i.e., Avalon-ST RX Port (rx\_st\_\*) for received packet, Avalon-ST Tx Port (tx\_st\_\*) for transmitted packet, and Hard IP Reconfiguration (hip\_reconfig\_\*) for PCIe configuration.

PCIe Hard IP implements Transaction layer, Data Link layer, and Physical layer of PCIe protocol. The maximum number of SSDs connecting to one FPGA device is limited by the numbers of PCIe Hard IP Block. One PCIe Hard IP (P-Tile) supports up to 16-lane PCIe Gen4 interface which can be configured to four of 4-lane PCIe Gen4 interface. Therefore, one PCIe Hard IP (P-Tile) can connect up with four NVMe IPs and four NVMe Gen4 SSD.

More details of Avalon-ST Intel Hard IP for PCIe (P-Tile) are described in following document.

[https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug\\_ptile\\_pcie\\_avst.pdf](https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug_ptile_pcie_avst.pdf)

## Core I/O Signals

Descriptions of all signal I/Os are provided in Table 2 - Table 4.

**Table 2: User logic I/O Signals (Synchronous to Clk signal)**

Signal	Dir	Description
<b>Control I/F of dglF typeS</b>		
RstB	In	Synchronous reset signal. Active low. De-asserted to '1' after Clk signal is stable.
Clk	In	System clock for running NVMe IP. The frequency of Clk must be more than or equal to a half of PCIeClk frequency, clock output from PCIe Hard IP. Therefore, when using 500 MHz PCIe hard IP for Gen4 speed, Clk frequency must be more than or equal to 250MHz. <i>Note: Some Gen4 SSDs shows the better performance when running higher Clk frequency.</i>
UserCmd[2:0]	In	User Command. Valid when UserReq='1'. ("000": Identify, "001": Shutdown, "010": Write SSD, "011": Read SSD, "100": SMART, "110": Flush, "101"/"111": Reserved)
UserAddr[47:0]	In	Start address to write/read SSD in 512-byte unit. Valid when UserReq='1'. In case LBA unit = 4 Kbyte, UserAddr[2:0] must be always set to "000" to align 4 Kbyte unit. In case LBA unit = 512 byte, it is recommended to set UserAddr[2:0]="000" to align 4 Kbyte size (SSD page size). Write/Read performance of most SSDs is reduced when start address is not aligned to page size.
UserLen[47:0]	In	Total transfer size to write/read SSD in 512-byte unit. Valid from 1 to (LBASize-UserAddr). In case LBA unit = 4 Kbyte, UserLen[2:0] must be always set to "000" to align 4 Kbyte unit. Valid when UserReq='1'.
UserReq	In	Asserted to '1' to send the new command request and de-assert to '0' after IP starts the operation by asserting UserBusy to '1'. This signal can be asserted when the IP is Idle (UserBusy='0'). Command parameters (UserCmd, UserAddr, UserLen, and CtmSubmDW0-DW15) must be valid and stable during UserReq='1'. UserAddr and UserLen are inputs for Write/Read command while CtmSubmDW0-DW15 are inputs for SMART/Flush command.
UserBusy	Out	Asserted to '1' when IP is busy. New request must not be sent (UserReq to '1') when IP is busy.
LBASize[47:0]	Out	Total capacity of SSD in 512-byte unit. Default value is 0. This value is valid after finishing Identify command.
LBAMode	Out	LBA unit size ('0': 512byte, '1': 4 Kbyte). Default value is 0. This value is valid after finishing Identify command.
UserError	Out	Error flag. Assert to '1' when UserErrorType is not equal to 0. The flag is de-asserted to '0' by asserting RstB to '0'.
UserErrorType[31:0]	Out	Error status. [0] – Error when PCIe class code is not correct. [1] – Error from CAP (Controller capabilities) register which may be caused from - MPSMIN (Memory Page Size Minimum) is not equal to 0. - NVM command set flag (bit 37 of CAP register) is not set to 1. - DSTRD (Doorbell Stride) is not 0. - MQES (Maximum Queue Entries Supported) is more than or equal to 7. More details of each register can be checked from NVMeCAPReg signal. [2] – Error when Admin completion entry is not received until timeout. [3] – Error when status register in Admin completion entry is not 0 or phase tag/command ID is invalid. Please see more details from AdmCompStatus signal. [4] – Error when IO completion entry is not received until timeout.

Signal	Dir	Description
<b>Control I/F of dgIF typeS</b>		
UserErrorType[31:0]	Out	<p>[5] – Error when status register in IO completion entry is not 0 or phase tag is invalid. Please see more details from IOCompStatus signal.</p> <p>[6] – Error when Completion TLP packet size is not correct.</p> <p>[7] – Error when PCIe Hard IP detects Error correction code (ECC) error from the internal buffer.</p> <p>[8] – Error from Unsupported Request (UR) flag in Completion TLP packet.</p> <p>[9] – Error from Completer Abort (CA) flag in Completion TLP packet.</p> <p>[15:10] – Reserved</p> <p>[16]- Error from unsupported LBA unit (LBA unit is not equal to 512 byte or 4 Kbyte)</p> <p>[31:17] – Reserved</p> <p><i>Note: Timeout period of bit[2]/[4] is set from TimeOutSet input.</i></p>
<b>Data I/F of dgIF typeS</b>		
UserFifoWrCnt[15:0]	In	Write data counter of Receive FIFO. Used to check full status. When full status is detected, the returned data transmission from Read command may be paused. If the size of FIFO data count is less than 16-bit, please fill '1' to upper bit.
UserFifoWrEn	Out	Asserted to '1' to write data to Receive FIFO during running Read command.
UserFifoWrData[255:0]	Out	Write data bus of Receive FIFO. Valid when UserFifoWrEn='1'.
UserFifoRdCnt[15:0]	In	Read data counter of Transmit FIFO. Used to check data size stored in FIFO. The transmitted data packet for Write command may be paused when the counter shows empty status. If FIFO size is less than 16-bit, please fill '0' to upper bit.
UserFifoEmpty	In	The signal is unused for this IP.
UserFifoRdEn	Out	Asserted to '1' to read data from Transmit FIFO during running Write command.
UserFifoRdData[255:0]	In	Read data returned from Transmit FIFO. Valid in the next clock after UserFifoRdEn is asserted to '1'.
<b>NVMe IP Interface</b>		
IPVesion[31:0]	Out	IP version number
TestPin[31:0]	Out	Reserved to be IP Test point.
TimeOutSet[31:0]	In	Timeout value to wait completion from SSD. Time unit is equal to 1/(Clk frequency). When TimeOutSet is equal to 0, Timeout function is disabled.
PCleLinkup	In	Asserted to '1' when PCIe Hard IP is linkup.
AdmCompStatus[15:0]	Out	Status output from Admin completion entry [0] – Set to '1' when Phase tag or Command ID in Admin completion entry is invalid. [15:1] – Status field value of Admin completion entry
IOCompStatus[15:0]	Out	Status output from IO completion entry [0] – Set to '1' when Phase tag in IO completion entry is invalid. [15:1] – Status field value of IO completion entry
NVMeCAPReg[31:0]	Out	The parameter value of the NVMe capability register when UserErrorType[1] is asserted to '1'. [15:0] – MQES (Maximum Queue Entries Supported) [19:16] – DSTRD (Doorbell Stride) [20] – NVM command set flag [24:21] – MPSPMIN (Memory Page Size Minimum) [31:25] – Undefined

Signal	Dir	Description
<b>NVMe IP Interface</b>		
IdenWrEn	Out	Asserted to '1' for sending data output from Identify command.
IdenWrDWEEn[7:0]	Out	Dword (32-bit) enable of IdenWrData. Valid when IdenWrEn='1'. '1': This dword data is valid, '0': This dword data is not available. Bit[0], [1], ..., [7] corresponds to IdenWrData[31:0], [63:32], ..., [255:224] respectively.
IdenWrAddr[7:0]	Out	Index of IdenWrData in 256-bit unit. Valid when IdenWrEn='1'. 0x00-0x7F is 4Kbyte Identify controller data, 0x80-0xFF is 4Kbyte Identify namespace data.
IdenWrData[255:0]	Out	4Kbyte Identify controller data or Identify namespace data. Valid when IdenWrEn='1'.
<b>NVMe IP Custom interface</b>		
CtmSubmDW0[31:0] – CtmSubmDW15[31:0]	In	16 Dwords of Submission queue entry for SMART/Flush command. DW0: Command Dword0, DW1: Command Dword1, ..., and DW15: Command Dword15. These inputs must be valid and stable when UserReq='1' and UserCmd="100" (SMART) or "110" (Flush).
CtmCompDW0[31:0] – CtmCompDW3[31:0]	Out	4 Dwords of Completion queue entry, output from SMART/Flush command. DW0: Completion Dword0, DW1: Completion Dword1, ..., and DW3: Completion Dword3
CtmRamWrEn	Out	Asserted to '1' for sending data output from custom command such as SMART command.
CtmRamWrDWEEn[7:0]	Out	Dword (32-bit) enable of CtmRamWrData. Valid when CtmRamWrEn='1'. '1': This dword data is valid, '0': This dword data is not available. Bit[0], [1], ..., [7] corresponds to CtmRamWrData[31:0], [63:32], ..., [255:224] respectively.
CtmRamAddr[7:0]	Out	Index of CtmRamWrData when SMART data is received. Valid when CtmRamWrEn='1'. (Optional) Index to request data input through CtmRamRdData for customized custom commands.
CtmRamWrData[255:0]	Out	512-byte data output from SMART command. Valid when CtmRamWrEn='1'.
CtmRamRdData[255:0]	In	(Optional) Data input for customized custom commands.

**Table 3: Physical I/O Signals for PCIe Hard IP (Synchronous to PCIeClk signal)**

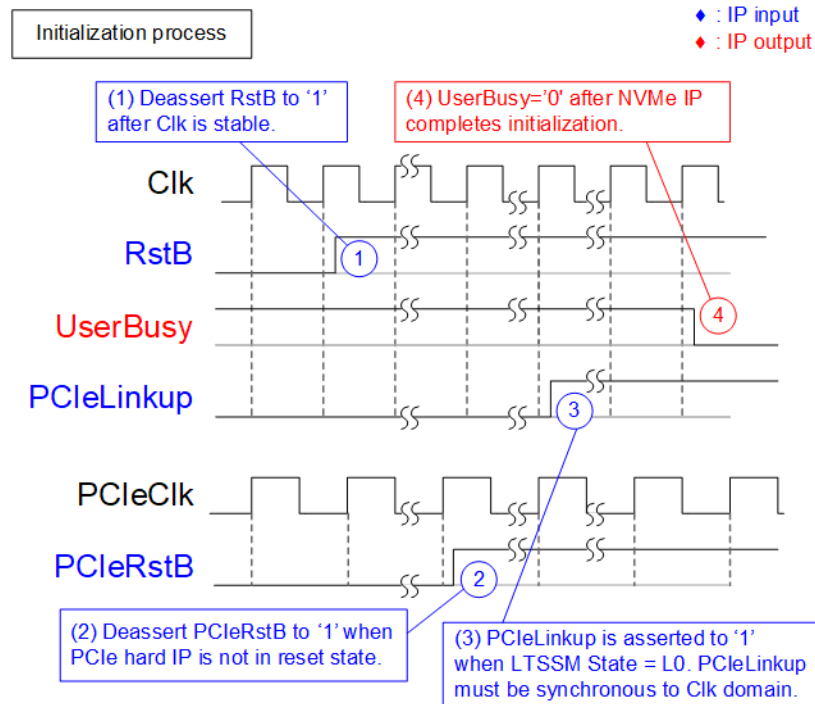
Signal	Dir	Description
<b>Clock and Reset</b>		
PCleRstB	In	Synchronous reset signal. Active low. De-asserted to '1' when PCIe Hard IP is not in reset state.
PCleClk	In	Clock output from PCIe Hard IP. It is recommended to configure as 500 MHz to achieve the best performance.
<b>PCIe Hard IP Rx Interface</b>		
PCleRxValid	In	Asserted to '1' to indicate that PCleRxData is valid. De-asserted to '0' within 27 clock cycles after PCleRxReady is de-asserted to '0'.
PCleRxSOP	In	Asserted to '1' to indicate that this is the first cycle of the TLP. Valid when PCleRxValid is asserted to '1'.
PCleRxEOP	In	Asserted to '1' to indicate that this is the last cycle of the TLP. Valid when PCleRxValid is asserted to '1'.
PCleRxReady	Out	Asserted to '1' to indicate that NVMe IP is ready to accept data. Data is transferred when both PCleRxValid and PCleRxReady are asserted in the same clock cycle.
PCleRxHd[127:0]	In	Received header data. Valid when PCleRxValid and PCleRxSOP are asserted to '1'.
PCleRxData[127:0]	In	Received data. Valid when PCleRxValid is asserted to '1'.
<b>PCIe Hard IP Tx Interface</b>		
PCleTxValid	Out	Asserted to '1' to indicate that PCleTxData is valid. When PCleTxReady is de-asserted to '0' during packet transmission, PCleTxValid must be de-asserted to '0' within 3 clock cycles. When PCleTxReady is re-asserted to '1', PCleTxValid must be re-asserted to '1' within 3 clock cycles to continue packet transmission.
PCleTxSOP	Out	Asserted to '1' to indicate first cycle of a TLP. Valid when PCleTxValid is asserted to '1'.
PCleTxEOP	Out	Asserted to '1' to indicate last cycle of a TLP. Valid when PCleTxValid is asserted to '1'.
PCleTxReady	In	Asserted to '1' to indicate that PCIe Hard IP is ready to accept data. Data is transferred when both PCleTxValid and PCleTxReady are asserted in the same clock cycle.
PCleTxHd[127:0]	In	Transmitted header data. Valid when PCleTxValid and PCleTxSOP are asserted to '1'.
PCleTxData[127:0]	Out	Received data. Valid when PCleTxValid is asserted to '1'.

**Table 4: Hard IP Reconfiguration Signals (Synchronous to CfgClk signal)**

Signal	Dir	Description
<b>Clock Signal</b>		
CfgClk	In	Reconfiguration clock. The frequency range is 100-125 MHz.
<b>Reconfiguration Interface</b>		
CfgAddr[20:0]	Out	Reconfiguration address for writing or reading.
CfgWr	Out	Asserted to '1' to send Write request.
CfgWrData[7:0]	Out	Write data. Valid when CfgWrite='1'.
CfgRd	Out	Asserted to '1' to send Read request
CfgRdData[7:0]	In	Read data. Valid when CfgRdValid='1'.
CfgRdValid	In	Asserted to '1' when CfgRdData is valid.
CfgWaitRequest	In	Asserted to '1' to indicate that the IP core is not ready to respond to a request.

## Timing Diagram

### Initialization



**Figure 4: Timing diagram during initialization process**

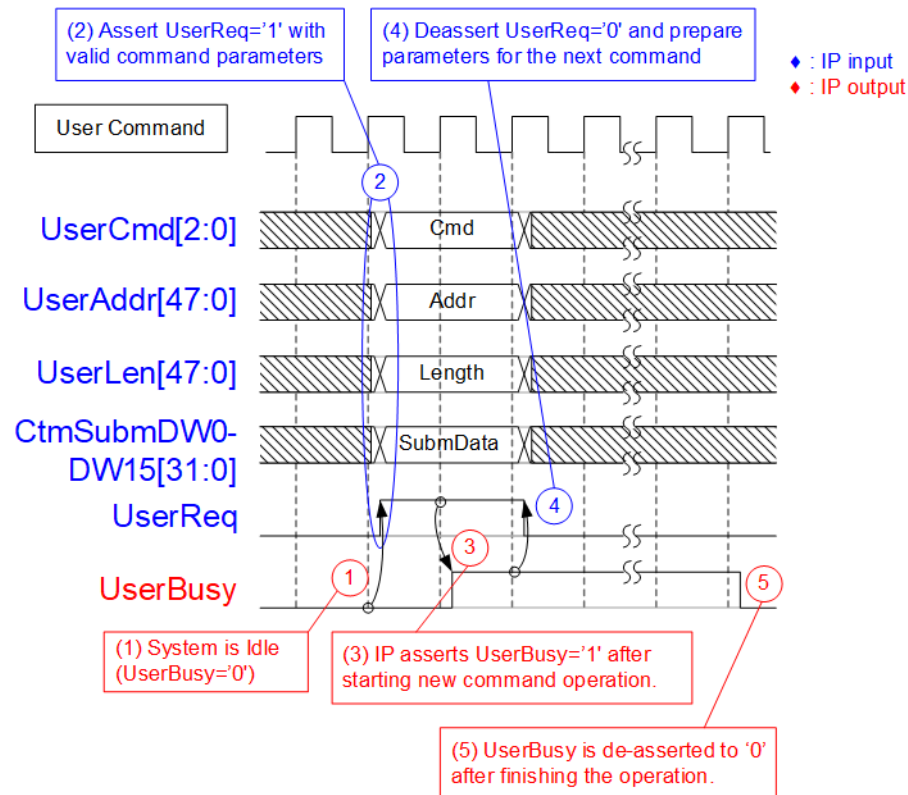
The steps of the initialization process is as follows.

- 1) Wait until Clk is stable and then de-asserts RstB to '1' to start IP initialization.
- 2) PCIe hard IP de-asserts PCIeRstB to '1' after finishing PCIe reset sequence. PCIe hard IP is ready to interface with the application layer.
- 3) After LTSSM state of PCIe hard IP is equal to L0 state, PCIe hard IP shows link up status. Linkup status output from PCIe hard IP is forwarded to asynchronous register to create PCIeLinkup on Clk domain, instead of PCIeClk domain. After that, NVMe IP starts initialization process.
- 4) UserBusy is de-asserted to '0' after NVMe IP completes initialization process.

After finishing all above steps, NVMe IP is ready to receive the command from user.

## Control interface of dgIF typeS

dgIF typeS signals are split into two groups. First group is control interface for sending command with the parameters and monitoring the status. Second group is data interface for transferring data stream in both directions.



**Figure 5: Contol interface of dgIF typeS timing diagram**

- 1) Before sending new command request to the IP, UserBusy must be equal to '0' to confirm that IP is the Idle state.
- 2) Command and the parameters such as UserCmd, UserAddr, and UserLen must be valid when asserting UserReq to '1' for sending the new command request.
- 3) IP asserts UserBusy to '1' after starting operation the new command.
- 4) After UserBusy is asserted to '1', UserReq is de-asserted to '0' to finish the current request. New parameters for the next command could be prepared on the bus. UserReq for the new command must not be asserted to '1' until the current command operation is finished.
- 5) UserBusy is de-asserted to '0' after the command operation is completed. New command request can be asserted.

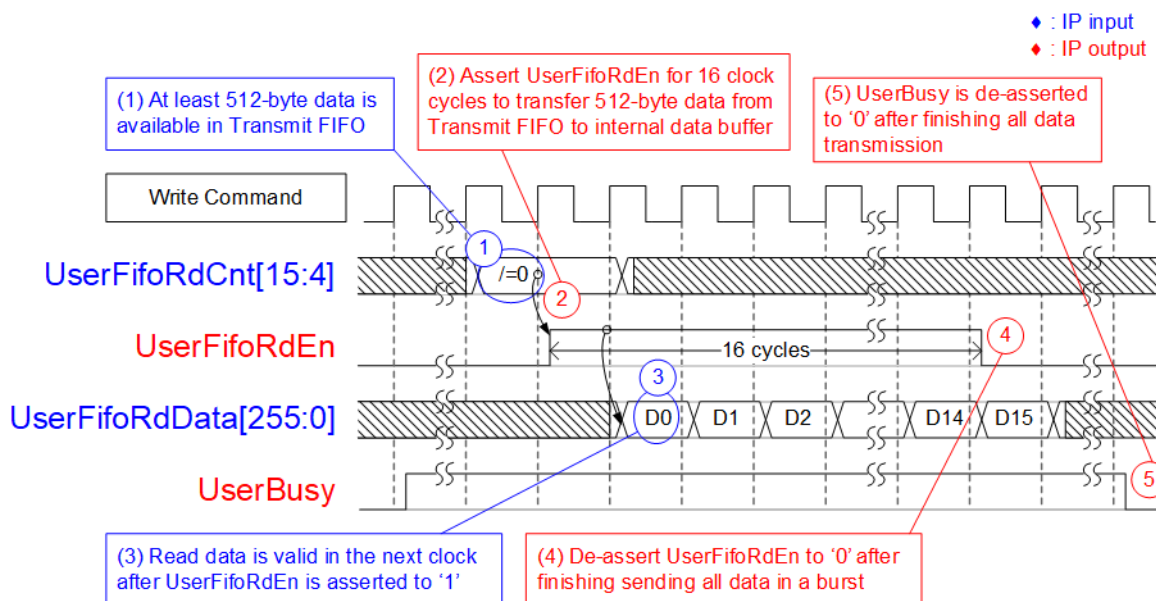
Note: The number of parameters using in each command are different, described more details as follows.

- Write and Read command: UserCmd, UserAddr, and UserLen.
- SMART and Flush command: UserCmd and CtmSubmDW0-DW15.
- Identify and Shutdown command: UserCmd.

### Data interface of dgIF types

Data interface of dgIF typeS is applied for transferring data stream when operating Write command or Read command. The interface is compatible to general FIFO interface.

In case of Write command, 16-bit FIFO read data counter (UserFifoRdCnt) which shows total number of data stored in the FIFO is monitored before starting a 512-byte (16x256-bit) burst transfer.

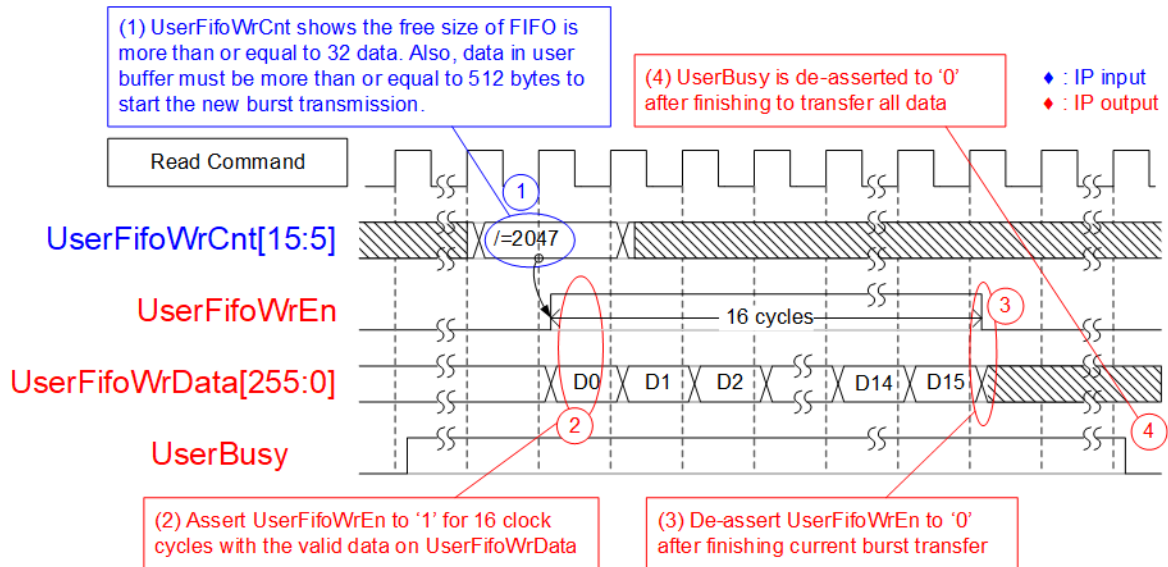


**Figure 6: Transmit FIFO interface for Write command**

In Write command, data is read from Transmit FIFO until total data are transferred completely. The details to transfer data are described as follows.

- 1) Before starting a new burst transfer, UserFifoRdCnt[15:4] is monitored. The IP waits until at least 512-byte data is available in Transmit FIFO (UserFifoRdCnt[15:4] is not equal to 0).
- 2) Assert UserFifoRdEn to '1' for 16 clock cycles to read 512 bytes data from Transmit FIFO.
- 3) UserFifoRdData is valid in the next clock cycle after asserting UserFifoRdEn to '1'. 16 data are continuously transferred.
- 4) UserFifoRdEn is de-asserted to '0' after reading the 16<sup>th</sup> data. Repeat step 1) if total transmitted data is not equal to the transfer length, set by the user.
- 5) After total data is completely transferred, UserBusy is de-asserted to '0'.





**Figure 7: Receive FIFO interface for Read command**

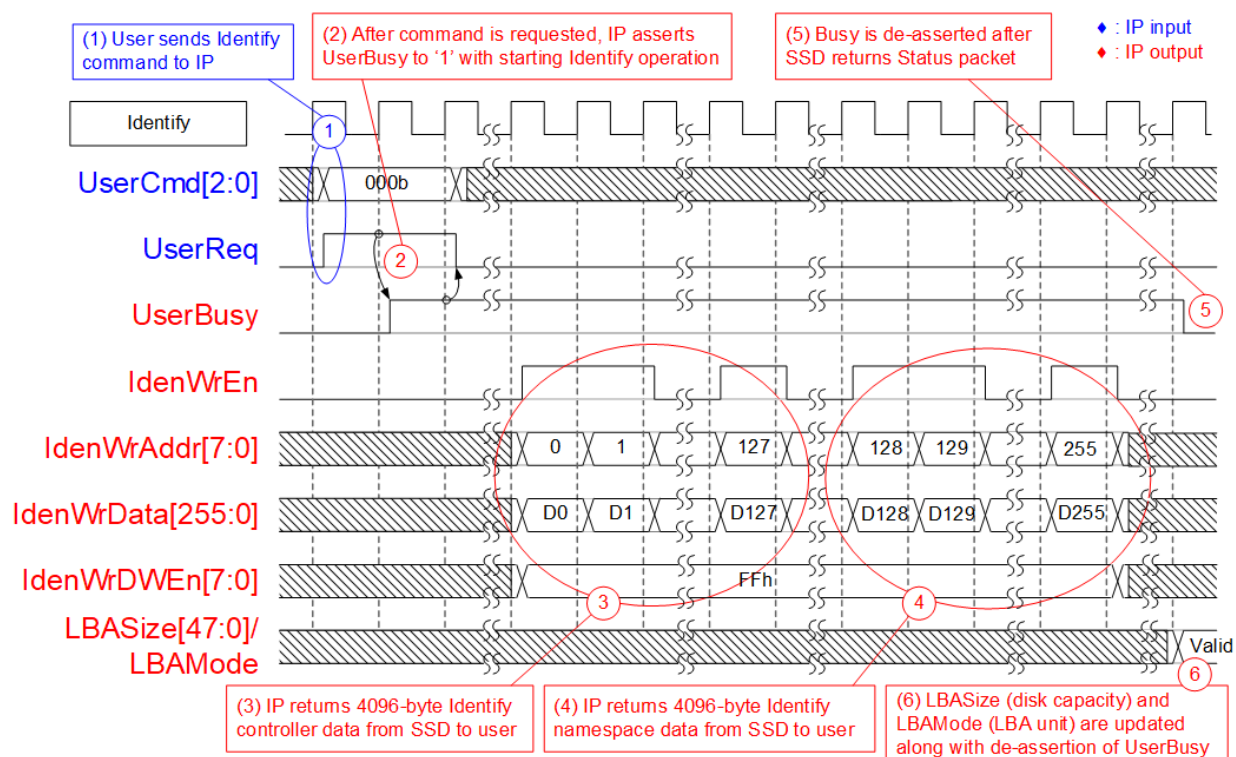
In Read command, data is sent from SSD to Receive FIFO until total data are completely transferred. The details to transfer a burst of data are as follows.

- 1) Before starting the new burst transmission, UserFifoWrCnt[15:5] is monitored. The IP waits until free space size of Received FIFO is enough (UserFifoWrCnt[15:5] is not equal to all 1 or 2047). After received data from the SSD is more than or equal to 512 bytes, the new burst transmission begins.
- 2) Assert UserFifoWrEn to '1' for 16 clock cycles to transfer 512-byte data from the data buffer to user logic.
- 3) After finishing transferring 512-byte data, UserFifoWrEn is de-asserted to '0'. When total transfer data is less than the transfer size in the command, repeat step 1) for transferring the next 512-byte data.
- 4) After total data is completely transferred, UserBusy is de-asserted to '0'.

### IdenCtrl/IdenName

It is recommended to send Identify command to the IP as the first command after system boots up. This command updates the necessary information of SSD, i.e., total capacity (LBASize) and LBA unit size (LBAMode). The SSD information is applied to be the limitation of the input parameters when operating Write and Read command, described as follows.

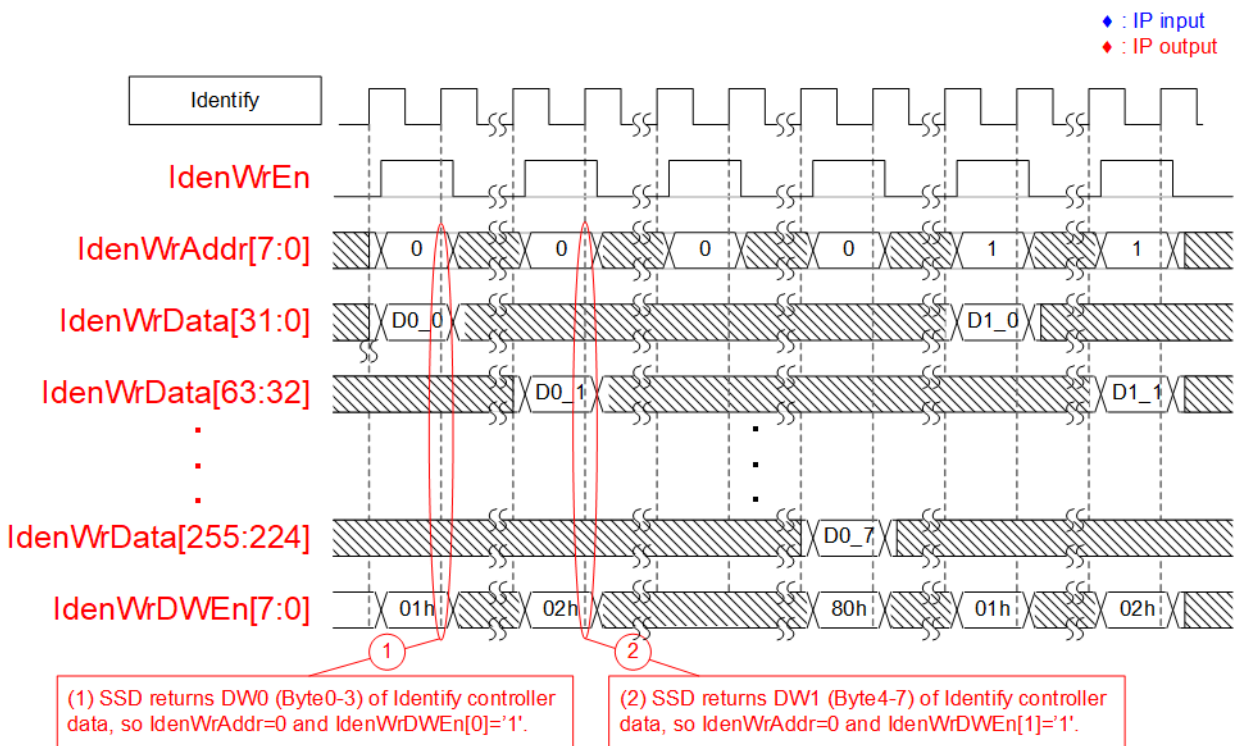
- 1) The sum of the address (UserAddr) and transfer length (UserLen), inputs of Write and Read command, must not be more than total capacity (LBASize) of the SSD.
- 2) If LBAMode of the SSD is equal to '1' (LBA unit size is 4 Kbyte), the three lower bit (bit[2:0]) of UserAddr and UserLen must be always equal to '0' to align 4 Kbyte unit.



**Figure 8: Identify command timing diagram**

The details when running Identify command are shown as follows.

- 1) Send Identify command to the IP (UserCmd="000" and UserReq='1').
- 2) The IP asserts UserBusy to '1' after running Identify command.
- 3) 4096-byte Identify controller data is returned to user. IdenWrAddr is equal to 0-127 with asserting IdenWrEn. Also, IdenWrData and IdenWrDWE are valid at the same clock as IdenWrEn='1'.
- 4) 4096-byte Identify namespace data is returned. IdenWrAddr is equal to 128-255.  
*Note: IdenWrAddr[7] can be applied to check data type that is Identify controller data or Identify namespace data.*
- 5) UserBusy is de-asserted to '0' after finishing the Identify command.
- 6) LBASize and LBAMode of the SSD are simultaneously updated.



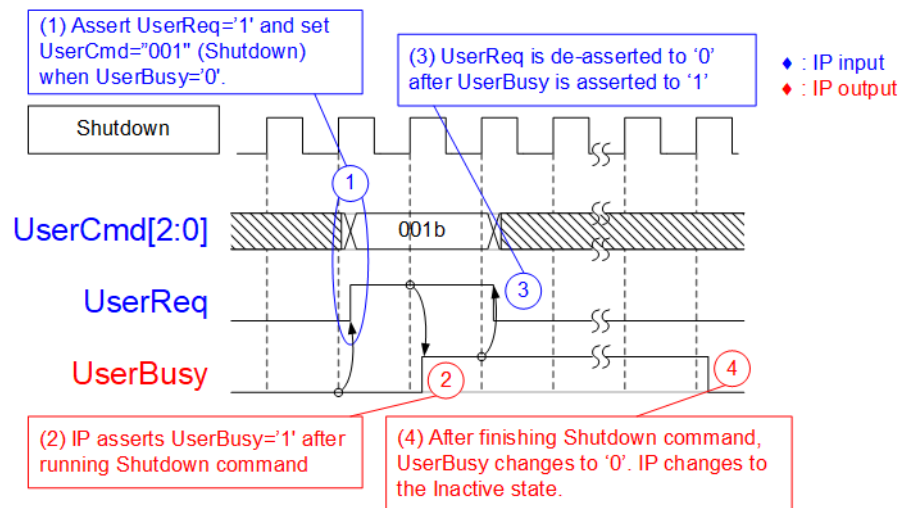
**Figure 9: IdenWrDWE n timing diagram**

IdenWrDWE n is 8-bit signal to be valid signal of 32-bit data. Some SSDs do not return 4-Kbyte Identify controller data and Identify namespace data continuously, but return only one dword (32-bit) at a time. Therefore, one bit of IdenWrDWE n is asserted to '1' in the write cycle to write 32-bit data, as shown in Figure 9. IdenWrDWE n[0], [1], ..., [7] corresponds to IdenWrData[31:0], [63:32], ..., [255:224] respectively.

## Shutdown

Shutdown command is recommended to send as the last command before the system is powered down. When Shutdown command is issued, SSD flushes the data from the internal cache to flash memory. After finishing operating Shutdown command, NVMe IP and SSD are inactive until the system is powered down.

*Note: If the SSD is powered down without Shutdown command, the total count of unsafe shutdowns (returned data of SMART command) is increased.*



**Figure 10: Shutdown command timing diagram**

The details when running Shutdown command are shown as follows.

- 1) Before sending the command request, the IP must be in the Idle state (UserBusy='0'). To send Shutdown command, user asserts UserReq to '1' with UserCmd="001".
- 2) Assert UserBusy to '1' after NVMe IP runs Shutdown command.
- 3) UserReq is de-asserted to '0' to clear the current request after UserBusy is asserted to '1'.
- 4) UserBusy is de-asserted to '0' when the SSD is shutdown completely. After that, the IP does not receive any command requested from user.

## SMART

SMART command is the command to check the SSD health. After sending SMART command, 512-byte health information is returned from the SSD. SMART command loads the parameters from CtmSubmDW0-DW15 signals on Custom command interface. User sets 16-dword data as constant value for SMART command before asserting UserReq. After that, the SMART data is returned via CtmRAM port as shown in Figure 11.

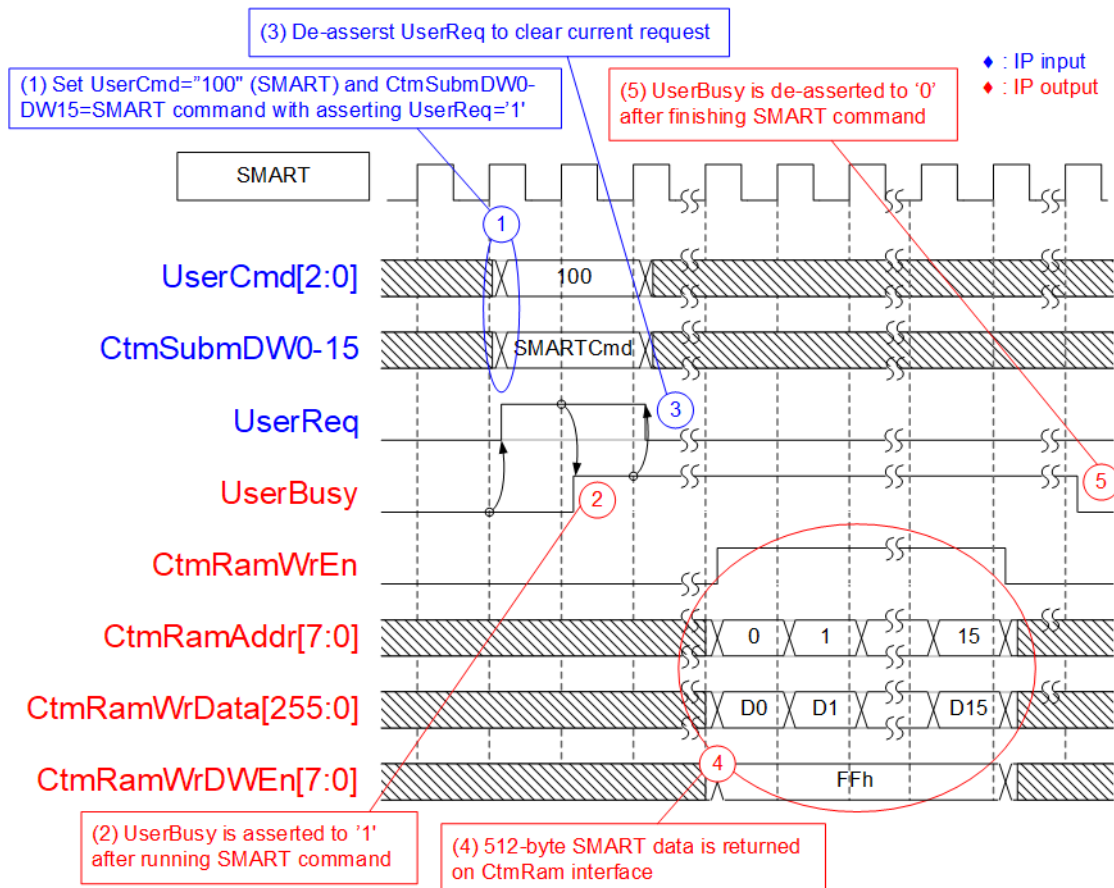


Figure 11: SMART command timing diagram

The details when running SMART command is shown as follows.

- 1) Before sending the command request, the IP must be in the Idle state (UserBusy='0'). All input parameters must be stable when UserReq is asserted to '1' for sending the request. CtmSubmDW0-DW15 is set as constant value by following value for SMART command.
 

CtmSubmDW0	= 0x0000_0002
CtmSubmDW1	= 0xFFFF_FFFF
CtmSubmDW2 – CtmSubmDW5	= 0x0000_0000
CtmSubmDW6	= 0x2000_0000
CtmSubmDW7 – CtmSubmDW9	= 0x0000_0000
CtmSubmDW10	= 0x007F_0002
CtmSubmDW11 – CtmSubmDW15	= 0x0000_0000
- 2) Assert UserBusy to '1' after NVMe IP runs SMART command.
- 3) UserReq is de-asserted to '0' to clear the current request. Next, user logic can change the input parameters for the next command request.
- 4) 512-byte SMART data is returned on CtmRamWrData signal with asserting CtmRamWrEn to '1'. CtmRamAddr is equal to 0-15 to be data index of 512-byte data. When CtmRamAddr=0, byte0-31 of SMART data is valid on CtmRamWrData. CtmRamWrDWEEn is dword enable for each 32-bit CtmRamWrData. If CtmRamWrDWEEn=FFh, all 256 bits of CtmRamWrData are valid.
- 5) UserBusy is de-asserted to '0' when finishing SMART command.

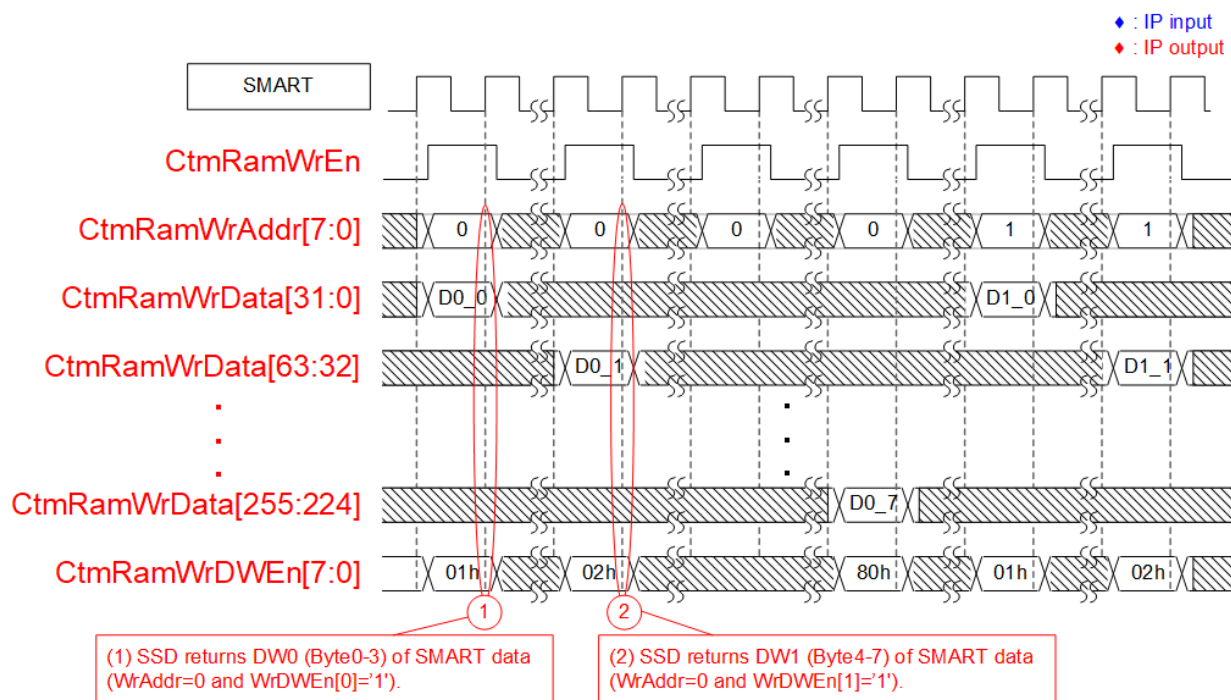
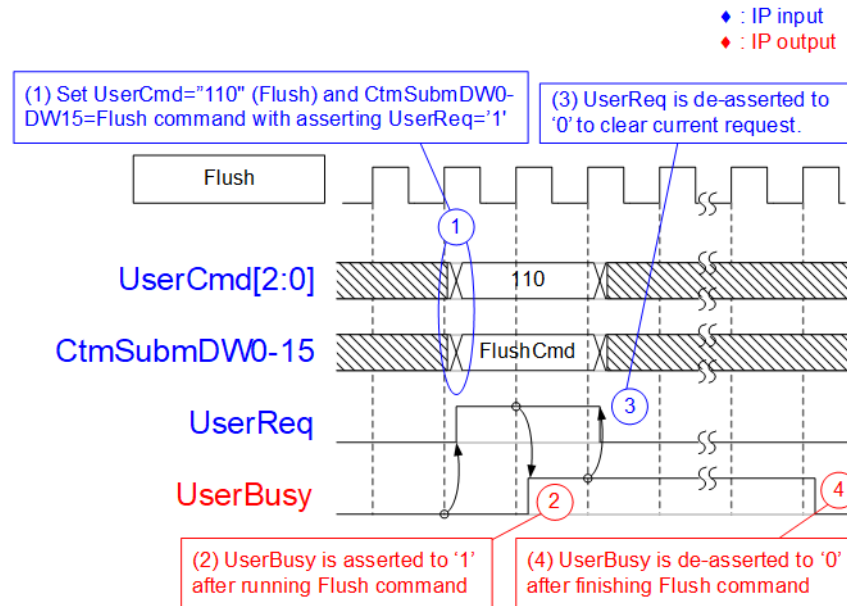


Figure 12: CtmRamWrDWEEn timing diagram

Similar to Identify command, some SSDs do not return 512-byte data continuously but return only one dword (32-bit) at a time. Therefore, one bit of CtmRamWrDWEEn is asserted to '1' in the write cycle to be the valid signal of 32-bit CtmRamWrData. CtmRamWrDWEEn[0], [1], ..., [7] corresponds to CtmRamWrData[31:0], [63:32], ..., [255:224] respectively.

## Flush

Most SSDs accelerate write performance by storing write data to cache before flushing to the flash memory by the SSD controller. If power is down unexpectedly, the data in the cache may be lost and not stored to the flash memory. Flush command is the command to force the SSD controller to flush data from the cache. After sending Flush command, all data in previous Write command can be guaranteed.



**Figure 13: Flush command timing diagram**

The details for running Flush command are shown as follows.

- 1) Before sending the command request, the IP must be in the Idle state (UserBusy='0'). All input parameters must be stable when UserReq is asserted to '1' for sending the request. CtmSubmDW0-DW15 is set as constant value by following value for Flush command.
 

CtmSubmDW0	= 0x0000_0000
CtmSubmDW1	= 0x0000_0001
CtmSubmDW2 – CtmSubmDW15	= 0x0000_0000
- 2) Assert UserBusy to '1' after NVMe IP runs Flush command.
- 3) UserReq is de-asserted to '0' to clear the current request. Next, user logic can change the input parameters for the next command request.
- 4) UserBusy is de-asserted to '0' when finishing Flush command.

Error

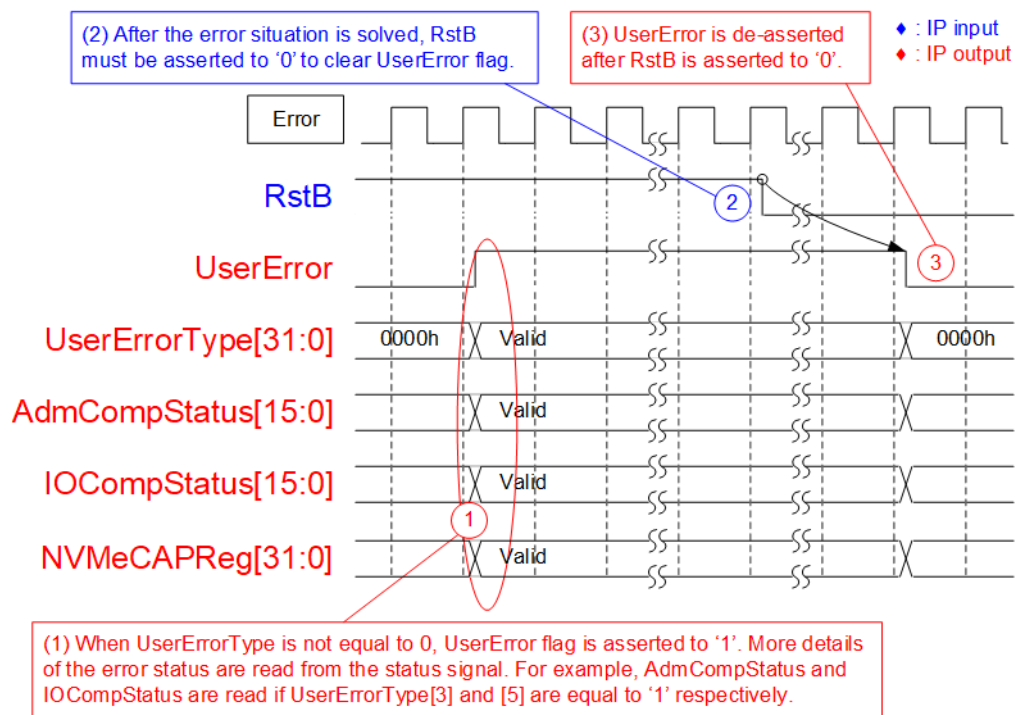


Figure 14: Error flag timing diagram

When the error is found during running initialization process or operating some commands, UserError flag is asserted to '1'. UserErrorType is read to check the error type. NVMeCAPReg, AdmCompStatus, and IOCompStatus are also valid for monitoring error details after UserError is asserted to '1'.

When the error is found during running initialization process, it is recommended to read NVMeCAPReg to check capability of NVMe SSD. When the error is found during operating the command, it is recommended to read AdmCompStatus and IOCompStatus. When bit[3] of UserErrorType is asserted, read AdmCompStatus to check more details. When bit[5] of UserErrorType is asserted, read IOCompStatus to check more details.

UserError flag is cleared by RstB signal only. After the failure is solved, RstB is asserted to '0' to clear the error flag.



## Verification Methods

The NVMe IP Core functionality was verified by simulation and also proved on real board design by using Agilex F-series FPGA Development board.

## Recommended Design Experience

Experience design engineers with a knowledge of QuartusII Tools should easily integrate this IP into their design.

## Ordering Information

This product is available directly from Design Gateway Co., Ltd. Please contact Design Gateway Co., Ltd. For pricing and additional information about this product using the contact information on the front page of this datasheet.

## Revision History

Revision	Date	Description
1.0	2-Jun-21	New release