

# Random Access NVMe IP Core

May 27, 2022

Product Specification

Rev1.1



## Design Gateway Co.,Ltd

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

URL: www.design-gateway.com

## Features

- NVMe host controller for access one NVMe SSD without CPU and external memory
- Support up to 32 Write/Read commands for 4 Kbyte random access (Multiple mode)
- Supported command (Single mode): Identify, Shutdown, SMART, and Flush
- Not support mixed Write and Read command in the same queue (1 queue = 32 commands)
- Include 128 Kbyte RAM to be data buffer
- Simple user interface by using data stream interface
- Supported NVMe device
  - Base Class Code:01h (mass storage), Sub Class Code:08h (Non-volatile) and Programming Interface:02h (NVMHCI)
  - MPSMIN (Memory Page Size Minimum): 0 (4 Kbyte)
  - MDTS (Maximum Data Transfer Size): At least 5 (128 Kbyte) or 0 (no limitation)
  - LBA unit: 512 byte
- User clock frequency must be more than or equal to PCIe clock (250MHz for Gen3)
- Operating with Avalon-ST Hard IP for PCIe by using 4-lane PCIe Gen3 (128-bit bus interface)
- Available reference design on Arria10 SoC development board and Arria10 GX development board with AB18-PCIeX16 or AB16-PCIeXOVR adapter board
- Customized service for following features
  - Support mixed Write and Read command in the same queue
  - Additional NVMe commands
  - RAM size extension to support more than 32 commands or more than 4 Kbyte data size for one Write/Read command

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	Arria10 SoC Arria10 GX 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 <sup>1</sup>	Design Tools
Arria10 GX	10AX115S2F45I2SG	300	1974	3419	-	1,114,176	QuartusII 16.0

May 27, 2022

## Applications

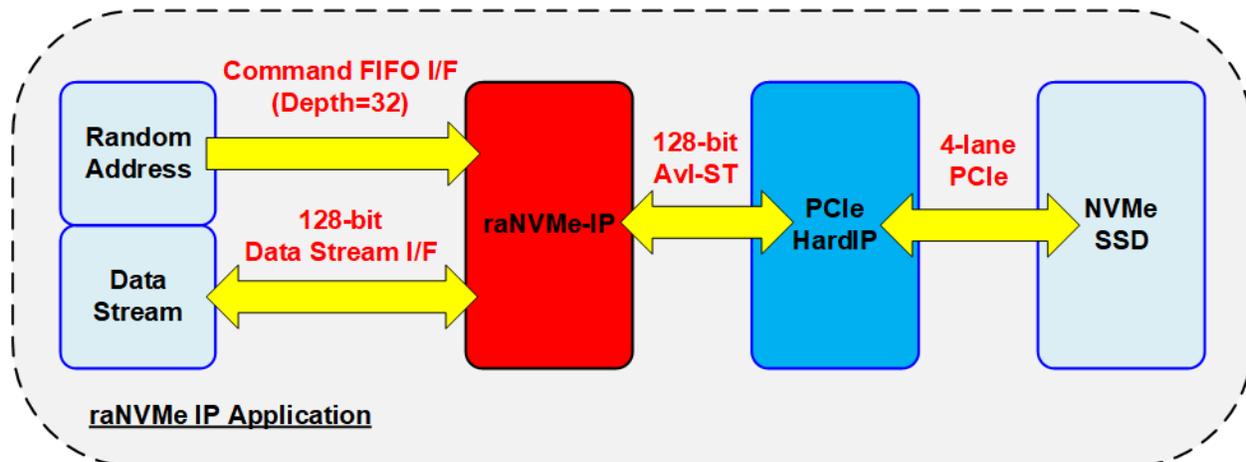


Figure 1: raNVMe IP Application

raNVMe IP Core integrated with Avalon-ST Hard IP for PCIe (PCIe hard IP) from Intel is ideal to access NVMe SSD without CPU and external memory in random access. raNVMe IP with PCIe hard IP can connect to one NVMe SSD and it is recommended for the application which needs to access NVMe SSD by using multiple addresses with less latency time. raNVMe IP supports to handle up to 32 Write commands or Read commands at the same time. Some applications of raNVMe IP are shown in Figure 2 and Figure 3.

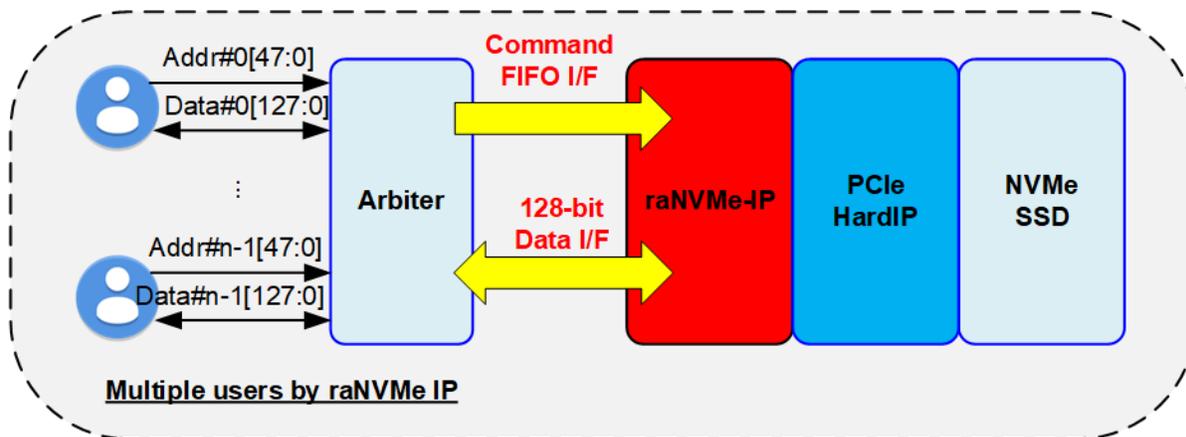
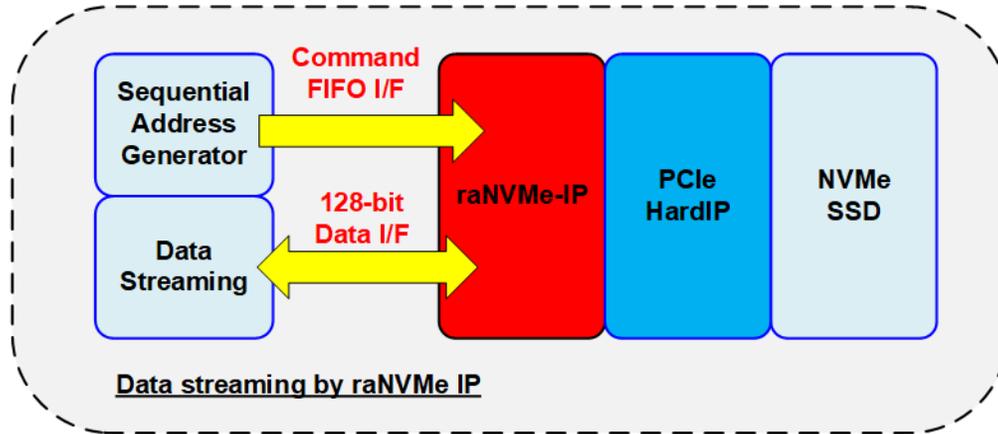


Figure 2: Multiple user application

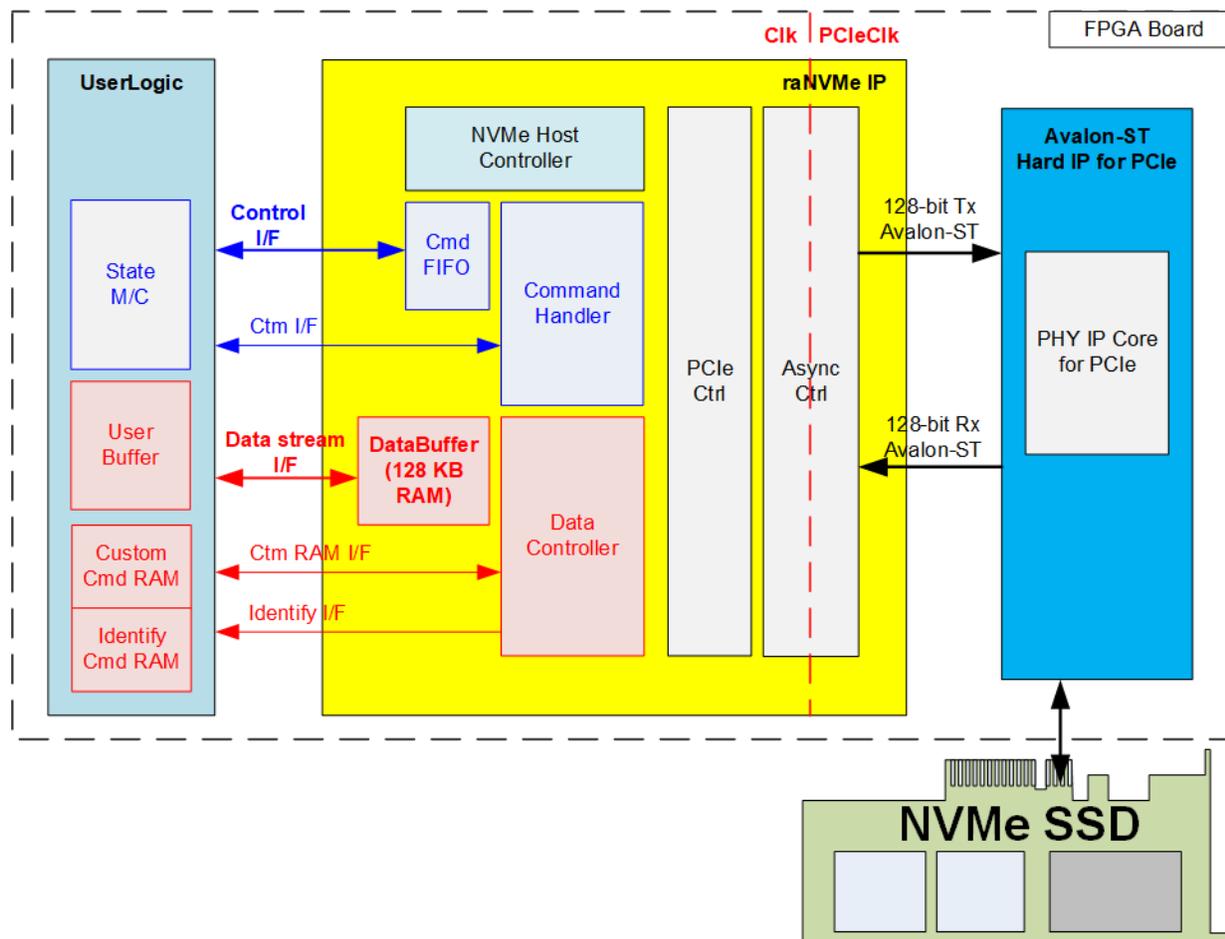
By designing Arbiter logic connecting between raNVMe-IP and multiple users, many users can transfer data at different addresses of the same NVMe SSD.



**Figure 3: Data streaming application**

The last application is data streaming system. Some applications need to record or replay the data from SSD but does not know the total transfer size. The control signals of the application are Start and Stop. One command size of raNVMe IP is fixed to 4 Kbyte and Start/Stop system is designed by sending multiple 4KB commands in sequential addressing, as shown in Figure 3.

## General Description



**Figure 4: raNVMe IP Block Diagram**

raNVMe IP implements as host controller to access NVMe SSD with random access function following NVM express standard. Physical interface of NVMe SSD is PCIe which is handled by Avalon-ST Hard IP for PCIe (PCIe hard IP) in Intel FPGA.

raNVMe IP supports six NVMe commands, i.e., Identify, Shutdown, Write, Read, SMART, and Flush command. As shown in Figure 4, Control interface for requesting command with parameter assignments uses FIFO interface to support many Write/Read command requests from the user. However, the data interface of each command uses different interface. The data interface of Write/Read command uses Data stream interface, 128-bit data bus controlled by valid/ready signal. While Data interface of SMART command and Identify command use Ctm RAM I/F and Identify I/F, respectively. Besides, SMART and Flush command require the additional control interface for parameter assignment, called Custom command interface.

raNVMe IP includes asynchronous circuit to allow the user logic running in the individual clock domain, not PCIe clock. However, clock frequency of user logic (Clk) must be more than or equal to PCIe clock (PCIeClk) frequency (250 MHz for PCIe Gen3).

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

## Functional Description

To design NVMe host controller supporting random access, raNVMe IP implements two protocols, i.e., NVMe protocol for interfacing with user and PCIe protocol for interfacing with PCIe hard IP. Figure 4 shows the hardware inside raNVMe IP that can be split to two groups – NVMe and PCIe. More details of each module are described as follows.

### NVMe

Six commands that raNVMe IP supports are arranged to two groups – Single mode group and Multiple mode group. Single mode group has four commands – Identify, SMART, Flush, and Shutdown. When user sends Single-mode command request to raNVMe-IP via Cmd FIFO interface, the command ready is de-asserted to protect the next command request from the user. The command read is re-asserted after the command is operated completely. The Multiple command group consists of Write command and Read command. The user can send up to 32 Write commands or Read commands with the address assignment to Cmd FIFO. The order of the data transferring via Data stream I/F is the same as the order of the command that sent to Cmd FIFO (Control I/F). However, all 32 commands that store to Cmd FIFO must be the same command, Write command or Read command. One Write/Read command transfers 4 Kbyte data, so 128-Kbyte buffer is integrated for supporting up to 32 Write/Read commands. The details of each submodule are described as follows.

- **NVMe Host Controller**

NVMe host controller is the main controller in raNVMe IP. The operation is split into two phases. 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.

To operate the command, NVMe host controller must handle with Command handler. When the command needs to transfer data such as Write, Read, SMART and Identify command, NVMe host controller must also handle with Data controller. The transferred order of each packet type returned from the SSD is monitored by NVMe host controller. The status value in the received packet is read to confirm that the command is finished normally or some errors are found. The error status is returned to the user when some errors are detected by NVMe host controller.

- **Cmd FIFO**

Command FIFO stores the command with the address assignment for each command from the user which is necessary for operating Write or Read command. Though Cmd FIFO can store up to 32 Write or Read commands, it has the limitation that all commands must be the same command. Cmd FIFO can accept only one command that is Single mode. To change the command, the user must wait until Cmd FIFO is empty.

- **Command Handler**

This module creates command packet and decodes the status packet returned from the SSD after finishing each command. The parameters within the command packet are set by the internal registers when the command is Identify, Shutdown, Write, or Read command. When the command is SMART or Flush command, the parameters are set via 512-bit custom command interface, called Custom submission queue entry which is defined by 16 DWORDs (1 DWORD = 32-bit).

The status of each command is returned via Completion queue entry. There are three outputs for user monitoring the status extracted from Completion queue entry, i.e., 16-bit Admin completion status when running Identify or Shutdown command, 16-bit IO completion status when running Write or Read command, or 128-bit Custom completion status defined by 4 DWORDs when running SMART or Flush command.

- **Data Buffer**

128-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, i.e., Data stream interface with 128-Kbyte buffer when running Write or Read command, Custom command RAM interface when running SMART command, or Identify interface when running Identify command.

The data controller must allocate the data buffer for transferring the data in each command. When running in Multi-mode command (Write or Read command), the Data controller must handle the order of the data transferred with the user to be the same order as the command input in Cmd FIFO.

## 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 - PCIeCtrl and AsyncCtrl. More details of each module are described as follows.

- **PCIeCtrl**

During initialization process, PCIeCtrl sets up PCIe environment of SSD via Avalon-ST interface. After that, the same interface is applied to transfer the command packet and data packet with NVMe module which are converted to be PCIe packet by PCIeCtrl. On the other hand, the received PCIe packet is decoded and converted to be NVMe packet for NVMe module by this module.

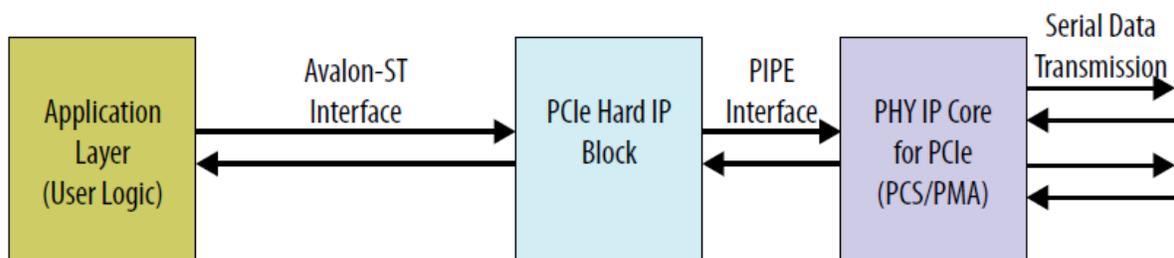
- **AsyncCtrl**

AsyncCtrl includes asynchronous registers and buffers to support clock domain crossing for control signals and data signals. Most logics in raNVMe IP run on user clock domain while PCIe hard IP runs on PCIe clock domain. Avalon-stream interface of PCIe hard IP must transfer data of each packet continuously, so the user bandwidth must be greater than or equal to PCIe bandwidth by running at higher or the same clock frequency of PCIe clock.

## User Logic

This module could be designed by using small state machine to send the commands with assigning the parameters for each command. For Write/Read command, user can send commands with the start address to Cmd FIFO. Data interface for Write/Read command is Data stream controlled by valid/ready signal while data output of SMART command and Identify command can be mapped to simple dual port RAM with byte enable. RAM size depends on the data size transferring in each command, but data width of all commands is 128-bit. Data size of Identify command is 8-Kbyte while data size of SMART command is 512-byte.

## Avalon-ST PCIe Hard IP



**Figure 5: Architecture of Avalon-ST PCIe Hard IP**

PCIe Hard IP Block embeds the PCIe protocol stack into the Intel FPGA. The hard IP block includes the transceiver modules, physical layer, data link layer, and transaction layer. The maximum number of SSDs connecting to one FPGA device is limited by the number of PCIe Hard IP Blocks. More details of Avalon-ST PCIe Hard IP are described in following document.

Intel Arria10 and Intel Cyclone 10 GX Avalon-ST Interface for PCIe Solutions User Guide

[https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug\\_a10\\_pcie\\_avst.pdf](https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug_a10_pcie_avst.pdf)

## Core I/O Signals

Descriptions of all signal I/Os are provided in Table 2 and Table 3.

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

Signal	Dir	Description
<b>Control Interface of raNVMe IP</b>		
RstB	In	Synchronous reset signal. Active low. De-assert to '1' when Clk signal is stable.
Clk	In	System clock for running raNVMe IP. The frequency must be more than or equal to PCIeClk which is output from PCIe hard IP (250 MHz for PCIe Gen3).
raNVMCmd[2:0]	In	User command. Valid when raNVMCVld = '1'. ("000": Identify, "001": Shutdown, "010": Write SSD, "011": Read SSD, "100": SMART, "110": Flush, "101"/"111": Reserved). <i>Note: User cannot change to run other commands when the IP is busy. User must wait until the current command is finished (raNVMBusy='0').</i>
raNVMAAddr[47:0]	In	Start address of requested Write/Read command in 512-byte unit. Valid when raNVMCVld = '1'. This value must be less than LBASize which is the SSD capacity. <i>Note: raNVMAAddr[2:0] is ignored for 4-Kbyte alignment.</i>
raNVMCVld	In	Command valid. Asserted to '1' to send new command. When command is Single mode (Identify, Shutdown, SMART, and Flush), raNVMCReady is de-asserted to '0' after the user asserts raNVMeCVld. When command is Multiple mode (Write and Read), user can assert raNVMCVld for sending up to 32 same commands before raNVMCReady is de-asserted to '0'.
raNVMCReady	Out	Asserted to '1' to indicate that raNVMe IP is ready to receive command.
raNVMBusy	Out	IP busy status. Asserted to '1' when raNVMe IP is busy.
raNVMCcnt[5:0]	Out	Remaining command count stored in the Cmd FIFO for running Write/Read command. Valid from 0 to 32. This signal is ignored for Single-mode command.
raNVMCId[4:0]	Out	Command ID in the Cmd FIFO for monitoring in Write/Read command. Valid from 0 to 31 when raNVMCVld='1'. This signal is ignored for Single mode. Used to match with raNVMDId to refer the command number which is currently transferred in Data stream interface. Also, it is used to match with error ID of IOCompStatus[20:16].
LBASize[47:0]	Out	Total capacity of SSD in 512-byte unit and always aligned to 4 KB unit. This value is valid after finishing Identify command. So, Identify command must be the first command that is run to update LBASize value before running other commands. Default value after boot up is 0.
raNVMErrr	Out	Error flag. Assert to '1' when raNVMErrType is not equal to 0. The flag can be cleared by asserting RstB to '0'.
raNVMErrType[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. - NVMe 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 Interface of raNVMe IP</b>		
raNVMeErrorType[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 unsupport LBA unit (LBA unit is not equal to 512-byte)</p> <p>[31:17] – Reserved</p> <p><i>Note: Timeout period of bit[2]/[4] is set by TimeOutSet input.</i></p>
<b>Data Stream Interface of raNVMe IP</b>		
raNVMDId[4:0]	Out	Show Command ID (raNVMCId) which is currently transferred on Data stream interface when running Multi-mode commands (Write or Read command). Valid from 0 to 31 when raNVMwValid='1' or raNVMrValid='1'.
raNVMwData[127:0]	In	Write data for Write command. Valid when raNVMwValid='1'.
raNVMwValid	In	Asserted to '1' to write data to raNVMe IP for Write command. User can write data to the IP before sending Write command but the IP must be in Idle status. <i>Note: If other command which is not Write command is run and there are write data stored in the buffer, all write data in the buffer will be flushed.</i>
raNVMwReady	Out	Asserted to '1' when raNVMe IP accepts the write data (raNVMwData).
raNVMwCnt[13:0]	Out	Total data count of write data in the buffer in 128-bit unit. Valid from 0-8192 when running Write command or the IP is in Idle status. When running other commands, this signal is not valid to read.
raNVMrData[127:0]	Out	Read data from SSD in Read command. Valid when raNVMrValid = '1'.
raNVMrValid	Out	Read data valid signal. Asserted to '1' for 256 clock cycles continuously to transfer 4Kbyte data.
<b>raNVMe 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 set to 0, Timeout is disabled.
PCleLinkup	In	Asserted to '1' when LTSSM state of PCIe hard IP is in L0 State.
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[31:0]	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 [20:16] – Command ID of IO Completion Entry, referred to raNVMCId [31:21] – Reserved
NVMeCAPReg[31:0]	Out	The parameter value of the NVMe capability register when raNVMeErrorType[1] is asserted to '1'. [15:0] – MQES (Maximum Queue Entries Supported) [19:16] – DSTRD (Doorbell Stride) [20] – NVM command set flag [24:21] – MPSMIN (Memory Page Size Minimum) [31:25] – Undefined

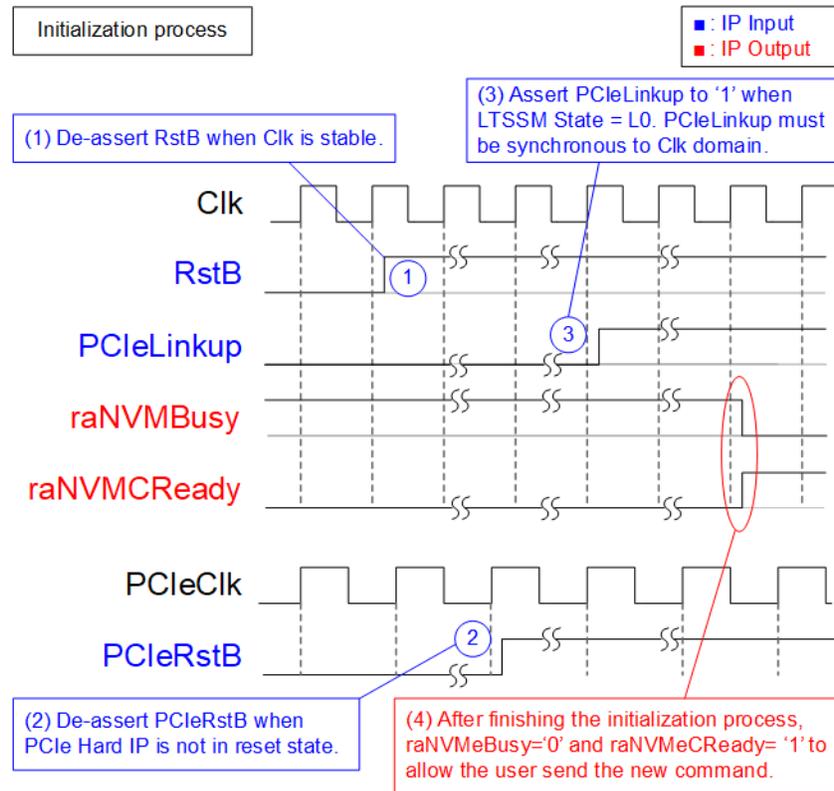
Signal	Dir	Description
<b>raNVMe IP Interface</b>		
IdeWrEn	Out	Assert to '1' for sending data output from Identify command.
IdeWrDWE[3:0]	Out	Dword (32-bit) enable of IdeWrData. Valid when IdeWrEn='1'. '1': this dword data is valid, '0': this dword data is not available. Bit[0], [1], [2], and [3] correspond to IdeWrData[31:0], [63:32], [95:64], and [127:96], respectively.
IdeWrAddr[8:0]	Out	Index of IdeWrData in 128-bit unit. Valid when IdeWrEn='1'. Bit[8]='0' (0x000-0x0FF): 4Kbyte Identify controller data. Bit[8]='1' (0x100-0x1FF): is 4Kbyte Identify namespace data.
IdeWrData[127:0]	Out	4Kbyte Identify controller data or Identify namespace data. Valid when IdeWrEn='1'.
<b>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. Valid when raNVMCValid='1' and raNVMCmd="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	Assert to '1' for sending data output from custom command such as SMART command.
CtmRamWrDWE[3:0]	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], [2], and [3] correspond to CtmRamWrData[31:0], [63:32], [95:64], and [127:96], respectively.
CtmRamAddr[8:0]	Out	Index of CtmRamWrData when SMART data is received. Valid when CtmRamWrEn='1'. (Optional) Index to request data input from CtmRamRdData for customized custom commands.
CtmRamWrData[127:0]	Out	512-byte data output from SMART command. Valid when CtmRamWrEn='1'.
CtmRamRdData[127:0]	In	(Optional) Data input for customized custom commands.

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

Signal	Dir	Description
<b>Clock Signal</b>		
PCleRstB	In	Synchronous reset signal. Active low. De-assert to '1' when PCIe Hard IP is not in reset state.
PCleClk	In	Clock output from Avalon-ST PCIe Hard IP within NVMe IP (250 MHz for PCIe Gen3).
<b>PCIe Hard IP Rx Interface</b>		
PCleRxValid	In	Assert to '1' to indicate that PCleRxData is valid. De-assert to '0' within 2 clock cycles after PCleRxReady is de-asserted to '0'.
PCleRxEOP	In	Assert to '1' to indicate that this is the last cycle of the TLP. Valid when PCleRxValid is asserted to '1'.
PCleRxReady	Out	Assert 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.
PCleRxError	In	Assert to '1' to indicate that there is an uncorrectable error correction coding (ECC) in the internal RX buffer of Avalon-ST PCIe Hard IP.
PCleRxData[127:0]	In	Receive data bus. Valid when PCleRxValid is asserted to '1'.
<b>PCIe Hard IP Tx Interface</b>		
PCleTxValid	Out	Assert to '1' to indicate that PCleTxData is valid. De-assert to '0' within 2 clock cycles after PCleTxReady is de-asserted to '0'.
PCleTxSOP	Out	Assert to '1' to indicate first cycle of a TLP. Valid when PCleTxValid is asserted to '1'.
PCleTxEOP	Out	Assert to '1' to indicate last cycle of a TLP. Valid when PCleTxValid is asserted to '1'.
PCleTxEmpty	Out	Indicate whether the upper qword at the end of a packet (PCleTxEOP='1') contains data. '0': PCleTxData[127:0] is valid, '1': PCleTxData[63:0] is valid.
PCleTxReady	In	Assert to '1' to indicate that Avalon-ST PCIe Hard IP is ready to accept data. Data is transferred when both PCleTxValid and PCleTxReady are asserted in the same clock cycle.
PCleTxError	Out	Assert to '1' to indicate an error on transmitted TLP. This signal is always set to '0'.
PCleTxData[127:0]	Out	Data for transmission. Valid when PCleTxValid is asserted to '1'.

## Timing Diagram

### Initialization



**Figure 6: Timing diagram during initialization process**

The step 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) Assert PCIeLinkup to '1' when LTSSM state of PCIe hard IP is L0 state. Though LTSSM state is run on PCIeClk, PCIeLinkup signal must be created to synchronous on Clk domain. Therefore, asynchronous register must be applied. After that, raNVMe IP starts the initialization process.
- 4) After finishing the initialization process, raNVMe IP de-asserts raNVMeBusy to '0' and asserts raNVMeCReady to '1' for receiving the new command from user.

After finishing above sequences, raNVMe IP is ready to receive the command from user.

### Control interface (Single mode)

raNVMe IP supports two command types - Single mode (run one command at a time) and Multiple mode (supports up to 32 commands). Write command and Read command are Multiple mode while other commands are Single mode. raNVMCReady is the status signal to show if the IP is ready to receive the new command or not. The details of Control interface for sending the command are described as follows.

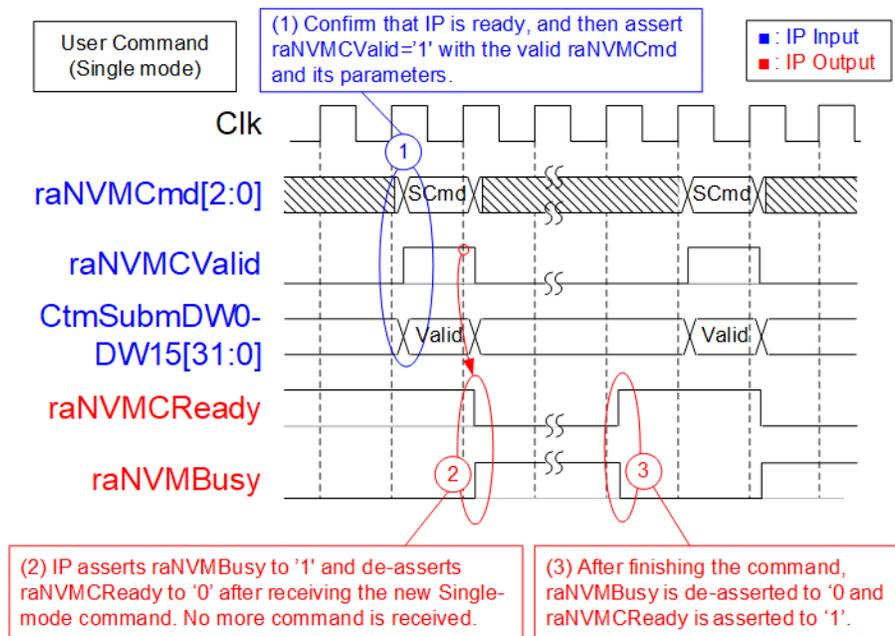


Figure 7: Single mode timing diagram

Figure 7 shows timing diagram when running Single mode, i.e., Identify, Shutdown, SMART, and Flush command.

- 1) Before sending the new command to the IP, user must check raNVMeBusy='0' and raNVMCReady='1' to confirm that raNVMe IP is Idle. After that, assert raNVMCValid to '1' for one clock cycle along with valid raNVMCcmd and the parameters which depend on the command.  
*Note: For SMART and Flush command, the parameters are CtmSubmDW0-DW15. For Identify and Shutdown command, no parameter is used.*
- 2) raNVMe IP asserts raNVMBusy to '1' after receiving the new command request and starting the command operation. Also, raNVMCReady is de-asserted to '0' when the receive command is Single mode. User must not request more commands.
- 3) After finishing command operation, raNVMBusy is de-asserted to '0' and raNVMCReady is asserted to '1' to accept the new command, except Shutdown command. If Shutdown command is run completely, raNVMBusy is de-asserted to '0' and raNVMCReady is de-asserted to '0' to wait system shut down without getting more command requests from user.

### Control interface (Multi-mode command)

For Write and Read command which are Multi-mode command, user can send many commands to raNVMe IP until Cmd FIFO is full (32 commands). However, user cannot switch the command from Write to Read and vice versa while the IP is operating (raNVMBusy='1'). To switch the command, the user must wait until all the commands are completed operated by monitoring raNVMBusy='0'.

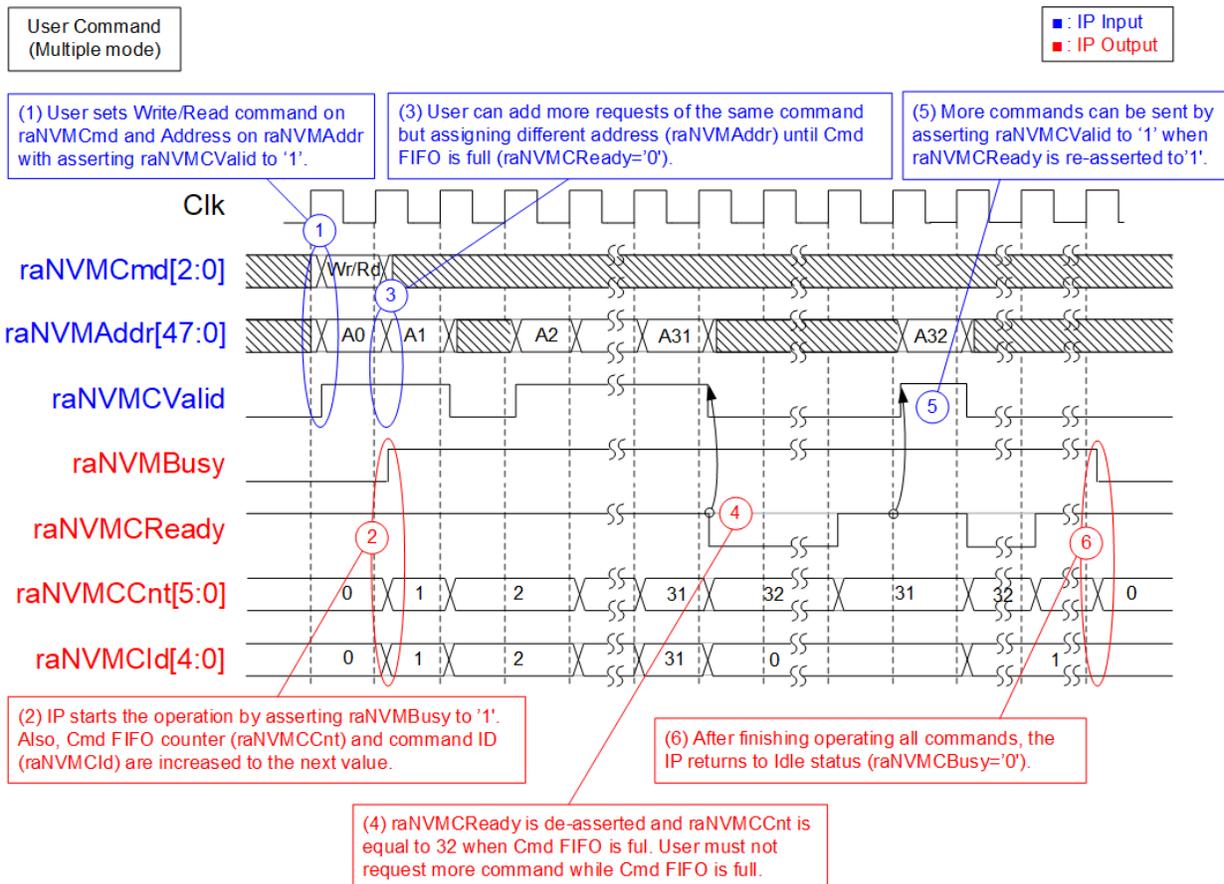


Figure 8: Multi-mode command timing diagram

- 1) User must confirm that raNVMe IP is Idle (raNVMeBusy='0') and Cmd FIFO is ready (raNVMCReady='1'). Next, send Write (raNVMCmd="010") or Read command (raNVMCmd="011") to the IP by asserting raNVMCValid to '1' for one clock cycle along with the valid raNVMAAddr.
- 2) IP starts the command operation with asserting raNVMBusy to '1'. Meanwhile, raNVMCcnt and raNVMCId are increased to the next value for indicating the number of commands in the Cmd FIFO and the next command ID, respectively.
- 3) As long as Cmd FIFO is not full (raNVMCcnt is less than 32 or raNVMCReady='1'), user can request of the same command with the new address. raNVMCmd is ignored when the new request is received but raNVMBusy is asserted to '1'. Therefore, the wrong operation will be happened if user assigns different value of raNVMCmd while raNVMeBusy='1'.
- 4) After Cmd FIFO is full (raNVMCcnt = 32), raNVMCReady is de-asserted to '0'. User shall not request any command. It needs to wait until raNVMCReady is re-asserted again.
- 5) After some commands are done, Cmd FIFO is not full. User can request the same command to the IP.
- 6) When raNVMe IP finishes all of the command request, raNVMCBusy is de-asserted to '0'. raNVMe IP returns back to the Idle state and ready to receive the new command which is different from the previous command.

### Data stream interface (Write command)

Data stream interface of raNVMe IP is applied for transferring data of Write command or Read command. The control signals of Data stream interface are valid and ready signal. Figure 9 shows the example of data stream interface when running Write command.

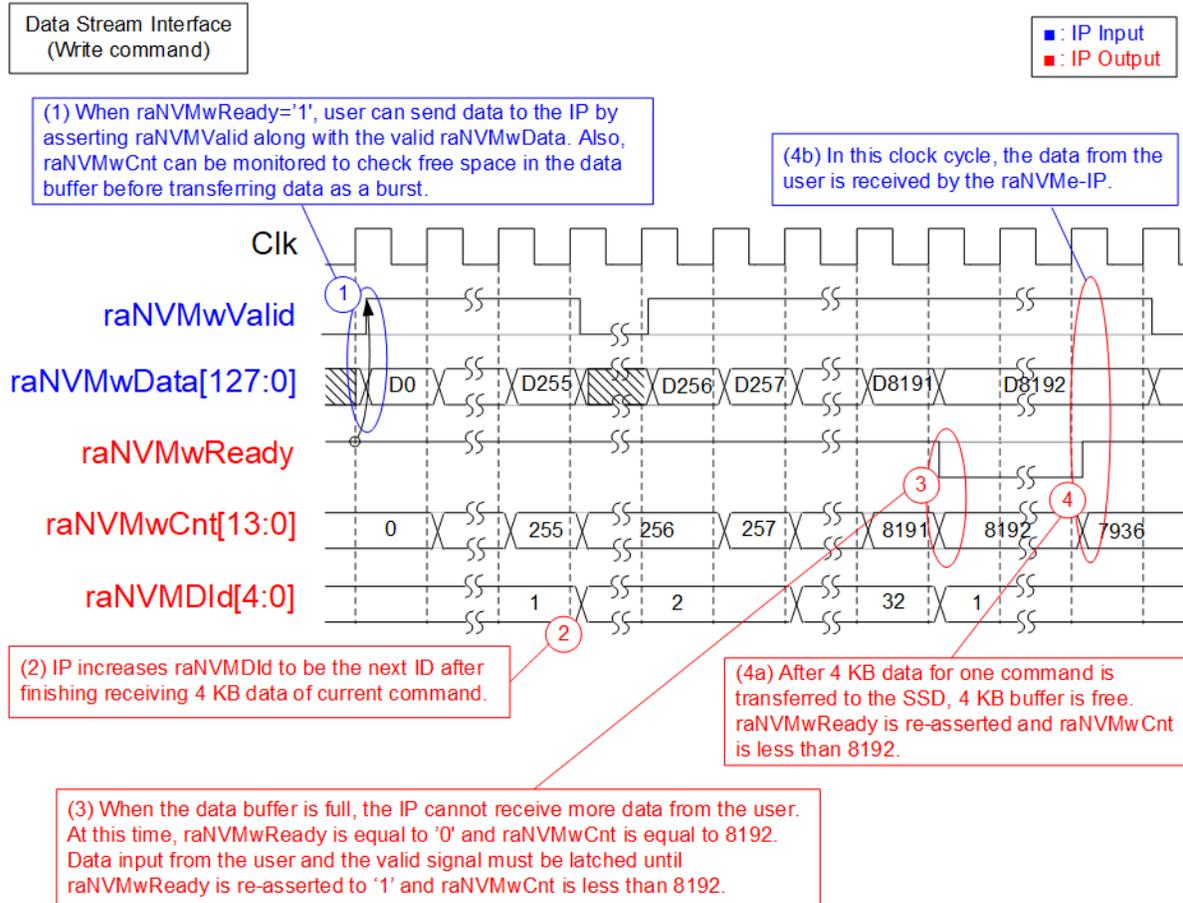


Figure 9: Data stream interface in Write command

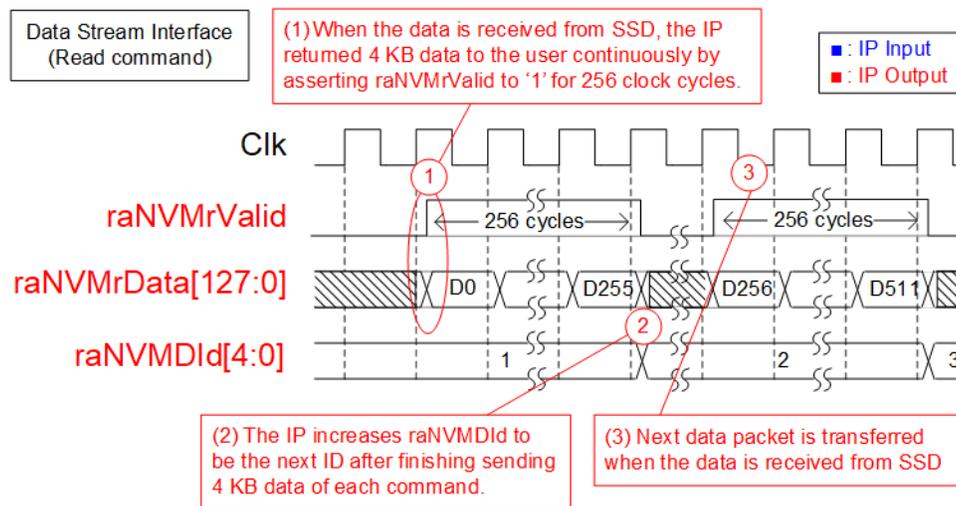
In Write command, raNVMe IP data stream interface operates independently from control interface. User can prepare the data into the data buffer before or after sending Write command request. The details of data stream interface for the Write command are as follows.

- 1) User must confirm that the data buffer has free space by monitoring raNVMwReady='1' or raNVMwCnt less than 8192. After that, user asserts raNVMwValid along with valid raNVMwData to transfer 128-bit data. There are two status signals to be flow control signals - raNVMwReady and raNVMwCnt. raNVMwCnt is recommended for transferring data as burst mode while raNVMwReady is applied for non-burst data transferring.

*Note: If the user writes the data to raNVMe IP and then runs other command that is not Write command, all Write data will be cleared from the buffer inside raNVMe IP.*

- 2) After finishing 4 KByte data (data size for one command), raNVMDId is increased to show the next command ID transferred on the data stream interface. raNVMDId shows which ID is currently transferred on raNVMwData.
- 3) When the data buffer is full (raNVMwCnt is equal to 8192 and raNVMwReady is de-asserted to '0'), the IP cannot accept the data from the user. Therefore, raNVMwData and raNVMwValid must hold the same value until raNVMwReady is re-asserted to '1'.
- 4) After a Write command is done, data buffer gets 4 KB free space. After that, raNVMwCnt is decreased by 4 KB and raNVMwReady is re-asserted to '1'. The data for the next command can be accepted by the IP.

### Data stream interface (Read command)



**Figure 10: Data stream interface in Read command**

Similar to Write command, the data stream interface of the IP is run independently from the control interface. There is no flow control for read data interface, so the user must be always ready to receive the data from the IP. The user logic should check the capability to store the data before sending the request of Read command to the control interface. The details of data stream interface for the Read command are as follows.

- 1) After raNVMe IP receives the 4 KB data of the current command ID from SSD, the IP returns the data from the user by asserting raNVMrValid to '1' along with valid raNVMrData continuously for 256 clock cycles.
- 2) raNVMDId is increased to indicate the next command ID that will be transferred.
- 3) The next 4 KB data is transferred after the SSD returns the data. Similar to step 1), the IP asserts raNVMrValid to '1' along with valid raNVMrData continuously.

### IdenCtrl/IdenName

It is recommended to send Identify command to the IP as the first command after system boots up. This command updates total capacity (LBASize) which is the necessary information of SSD for calculating the valid range of the address (UserAddr) when operating Write or Read command. UserAddr in Write and Read command must be less than LBASize of the SSD.

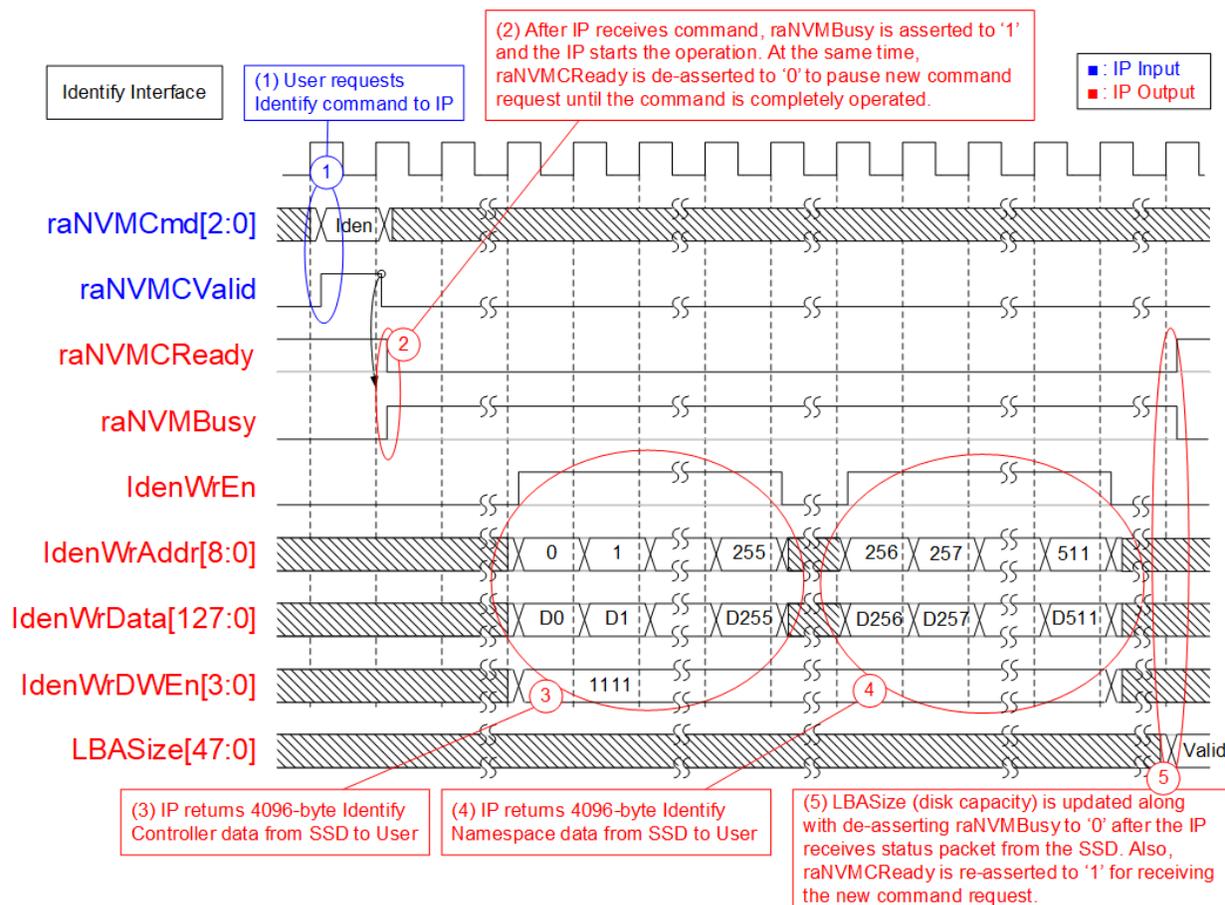
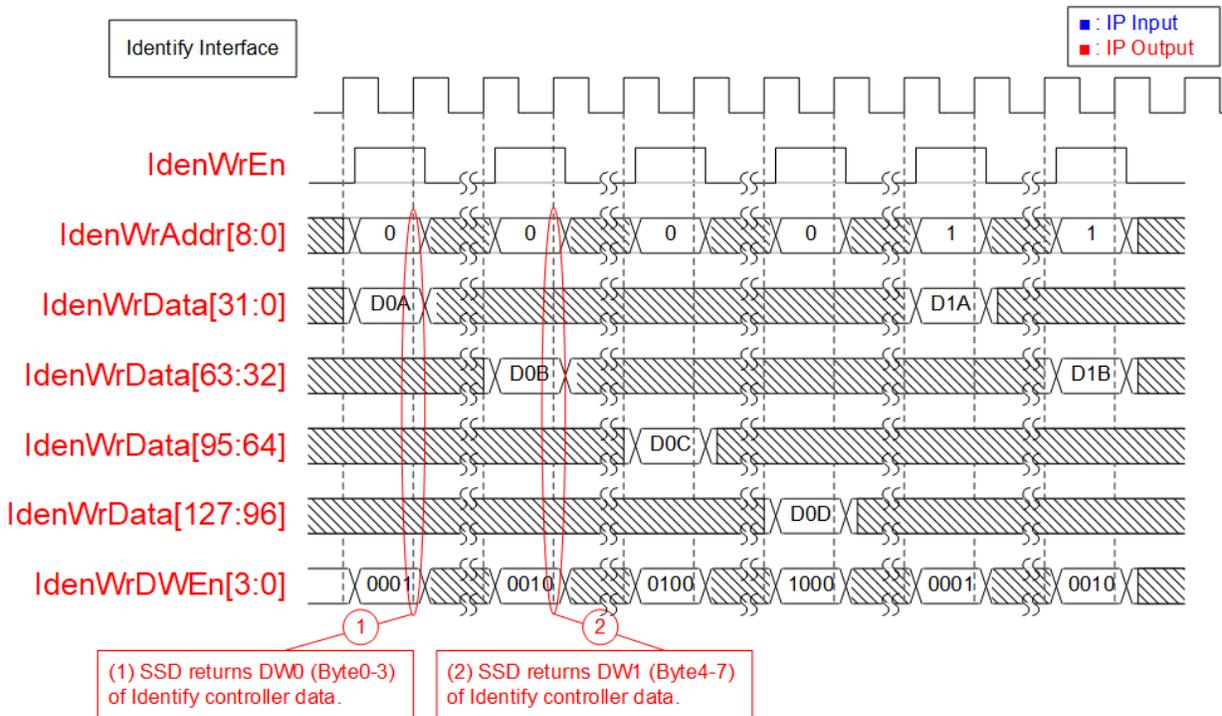


Figure 11: Identify command timing diagram

The details when running Identify command are shown as follows.

- 1) Send the request of Identify command to the IP (raNVMCmd="000" and raNVMCValid='1').
- 2) The IP asserts raNVMBusy to '1' after running the command. At the same time, raNVMCReady is de-asserted to '0' to ignore the new command request from the user until the current command is completed.
- 3) 4096-byte Identify controller data is returned to user. IdenWrAddr is equal to 0-255 with asserting IdenWrEn. IdenWrData and IdenWrDWEEn are valid at the same clock as IdenWrEn='1'.
- 4) Similar to Identify controller data, 4096-byte Identify namespace data is returned but IdenWrAddr is equal to 256-511. IdenWrAddr[8] can be applied to check the data type which is Identify controller data or Identify namespace data.
- 5) After finishing the Identify command operation, raNVMBusy is de-asserted to '0' along with valid LBASize. Also, raNVMCReady is re-asserted to '1' to receive the new command from the user.



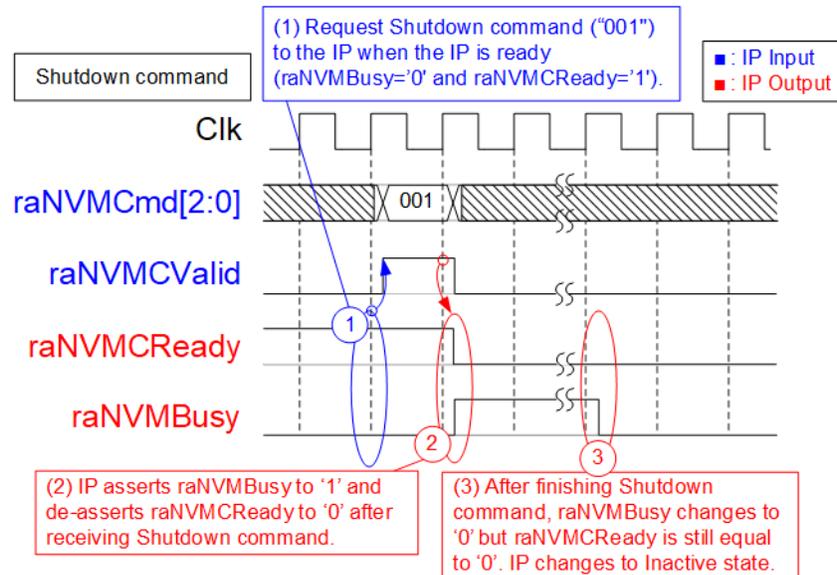
**Figure 12: IdenWrDWEn timing diagram**

IdenWrDWEn is 4-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 IdenWrDWEn is asserted to '1' in the write cycle to write 32-bit data, as shown in Figure 12. IdenWrDWEn[0], [1], [2], and [3] correspond to IdenWrData[31:0], [63:32], [95:64], and [127:96], 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, raNVMe 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 13: Shutdown command timing diagram**

The details when running Shutdown command is shown as follows.

- 1) Before sending the command request, the IP must be in the Idle state (raNVMBusy='0' and raNVMCReady='1'). To send Shutdown command, user asserts raNVMCValid to '1' with raNVMCmd = "001".
- 2) When raNVMe IP receives Shutdown command, the IP asserts raNVMBusy to '1' and de-asserts raNVMCReady to '0'.
- 3) raNVMBusy is de-asserted to '0' after the SSD is completely shut down. However, raNVMCReady is still equal to '0' and the IP does not receive any command afterwards.

## 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. After that, the SMART data is returned via CtmRAM port as shown in Figure 14.

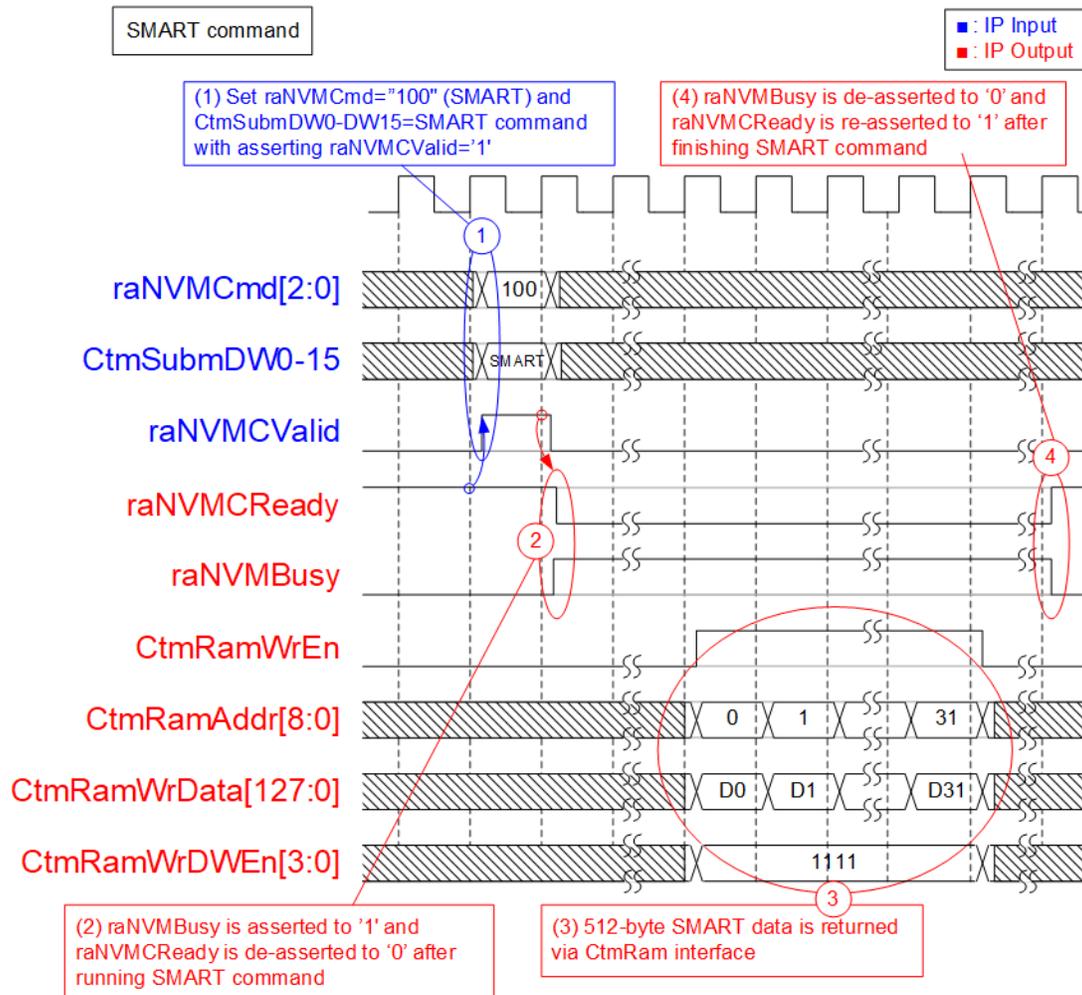


Figure 14: SMART command timing diagram

The details when running SMART command is shown as follows.

- 1) Before sending the command request, the IP must be ready to receive command (raNVMBusy='0' and raNVMCReady='1'). Then, asserts raNVMCValid with raNVMCmd="100" and valid CtmSubmDW0-DW15 which must be set 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) raNVMe IP accepts SMART command by asserting raNVMBusy to '1' and de-asserting raNVMCReady to '0'.
- 3) 512-byte SMART data is returned on CtmRamWrData signal with asserting CtmRamWrEn to '1'. CtmRamAddr is equal to 0-31 to be data index of 512-byte data. When CtmRamAddr=0, byte0-15 of SMART data is valid on CtmRamWrData. CtmRamWrDWEEn is dword enable for each 32-bit CtmRamWrData. If CtmRamWrDWEEn="1111", all 128-bit of CtmRamWrData are valid.
- 4) After SMART command operation is finished, raNVMBusy is de-asserted to '0' and raNVMCReady is asserted to '1' to receive the next command.

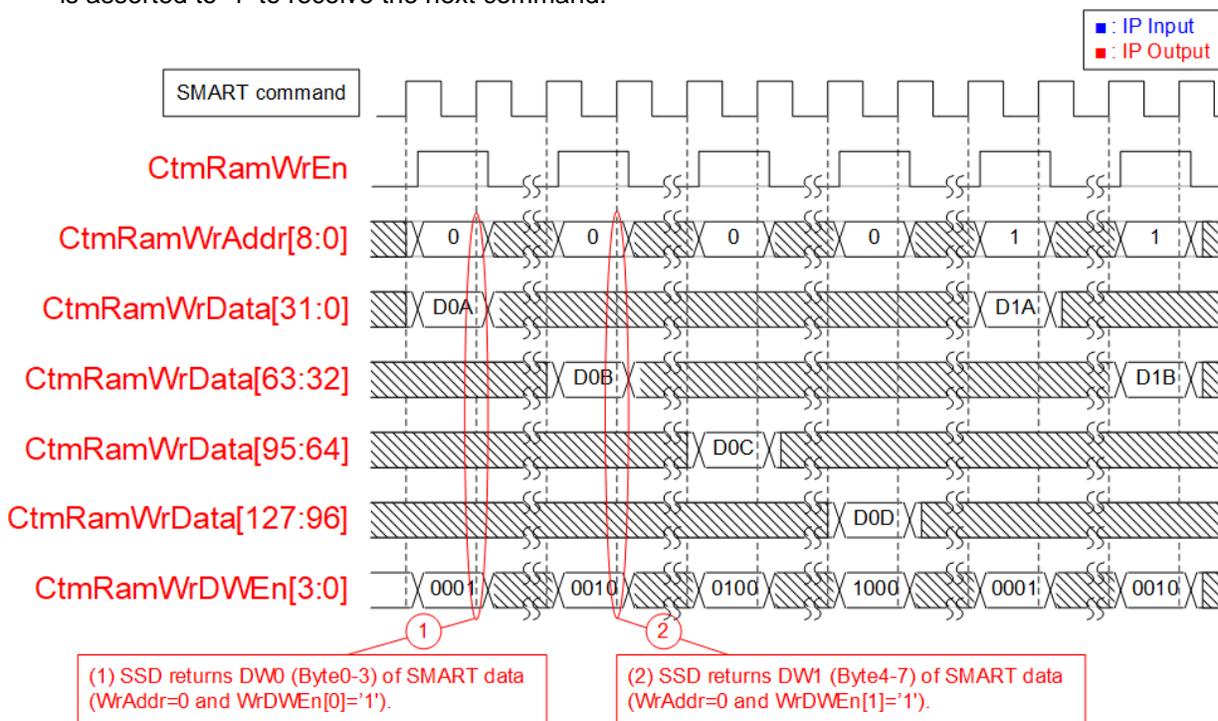
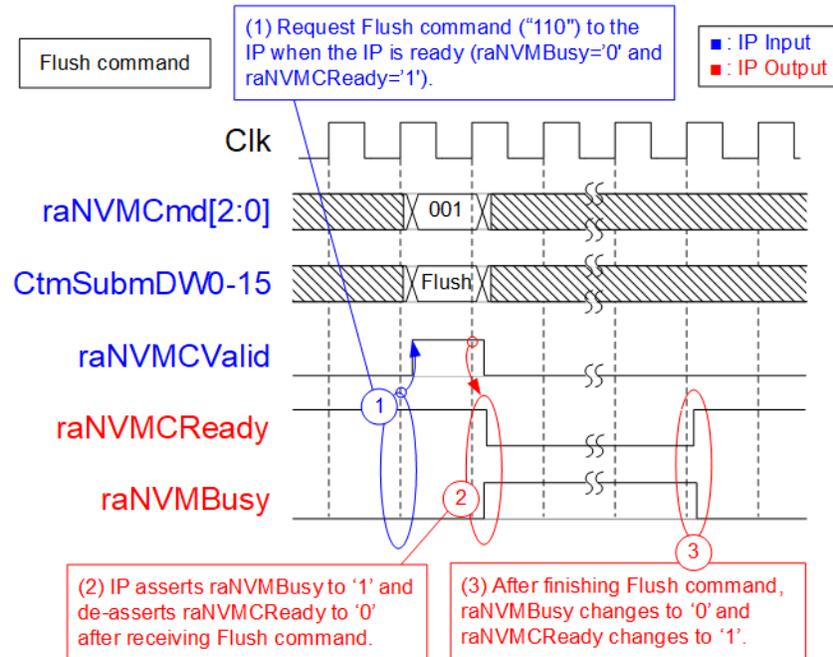


Figure 15: CtmRamWrDWEEn timing diagram

Similar to Identify command, some SSDs do not return 512-byte data continuously but returns 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], [2], and [3] correspond to CtmRamWrData[31:0], [63:32], [95:64], and [127:96], 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 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.

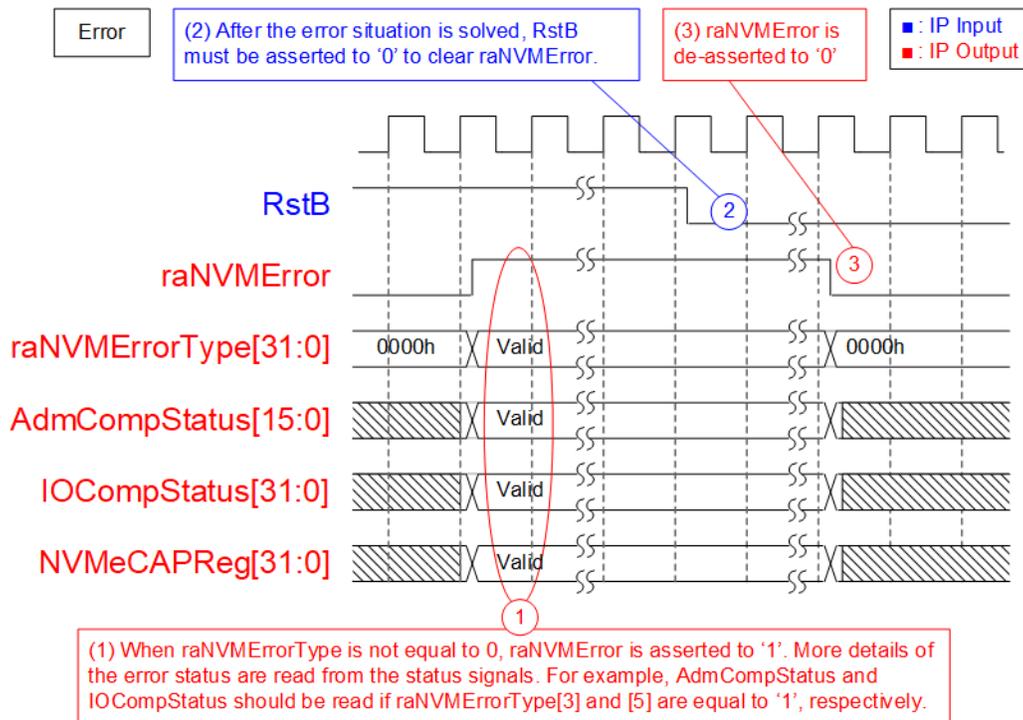


**Figure 16: Flush command timing diagram**

The details for running Flush command are shown as follows.

- 1) Before sending the command request, the IP must be ready to receive command ( $raNVMBusy=0$  and  $raNVMCReady=1$ ). Then, asserts  $raNVMCValid$  with  $raNVMCmd="100"$  and valid  $CtmSubmDW0-DW15$  which must be set by following value for Flush command.  
 $CtmSubmDW0 = 0x0000\_0000$   
 $CtmSubmDW1 = 0x0000\_0001$   
 $CtmSubmDW2 - CtmSubmDW15 = 0x0000\_0000$
- 2) raNVMe IP accepts Flush command by asserting  $raNVMBusy$  to '1' and de-asserting  $raNVMCReady$  to '0'.
- 3) After finishing Flush command operation,  $raNVMBusy$  is de-asserted to '0' and  $raNVMCReady$  is asserted to '1' to receive the next command.

Error



**Figure 17: Error flag timing diagram**

When the error is found during running initialization process or operating some commands, raNVMeError flag is asserted to '1'. raNVMeErrorType is read to check the error type. NVMeCAPReg, AdmCompStatus, and IOCompStatus are also valid for monitoring error details after raNVMeError 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 raNVMeErrorType is asserted, read AdmCompStatus to check more details. When bit[5] of raNVMeErrorType is asserted, read IOCompStatus to check more details.

The raNVMeError can be 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 Arria10 GX and Arria10 SoC 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
1.1	27-May-2022	Support Arria10SoC board
1.0	25-Aug-2020	Initial release