Design Gateway Co.,Ltd
E-mail: ip-sales@design-gateway.com
URL: design-gateway.com

Core Facts

Provided with Core	
Documentation	Reference Design Manual Demo Instruction Manual
Design File Formats	Encrypted File
Instantiation Templates	VHDL
Reference Designs & Application Notes	Vivado Project, See Reference Design Manual
Additional Items	Demo on VCK190
Support	
Support Provided by Design Gateway Co., Ltd.	

Features

- NVMe host controller for access one NVMe SSD without CPU and external memory
- Support multiple Write or 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
- Configurable Command Depth: 32, 64, 128, 256 Commands
- Buffer size (depends on Command Depth): 128Kbyte (32 Commands) – 1Mbyte (256 Commands)
- 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)
  - MQES (Maximum Queue Entries Supported): more than or equal to (2 x Command Depth - 1)
  - LBA unit: 512 bytes
- User clock frequency must be more than or equal to PCIe clock (250MHz for Gen4)
- PCIe Hard IP: Integrated Block for PCI Express from Xilinx (256-bit interface of 4-lane Gen4)
- Connect to one NVMe SSD directly (without PCIe switch)
- Available reference designs: VCK190 board with AB18-PCIeX16 adapter board
- Customized service for following features
  - Additional NVMe commands

Table 1: Example Implementation Statistics (Versal)

Family	Example Device	Cmd Depth	Fmax (MHz)	CLB Regs	CLB LUTs	Slice[1]	BRAM Tile[1]	URAM	Design Tools
Versal AI Core	XCVC1902-VSVA2197-2MP-E-S	32	375	5923	3704	1032	4	4	Vivado2022.1
		256	375	5971	3759	1053	5	32	

Notes: 1) Actual logic resource dependent on percentage of unrelated logic

Applications

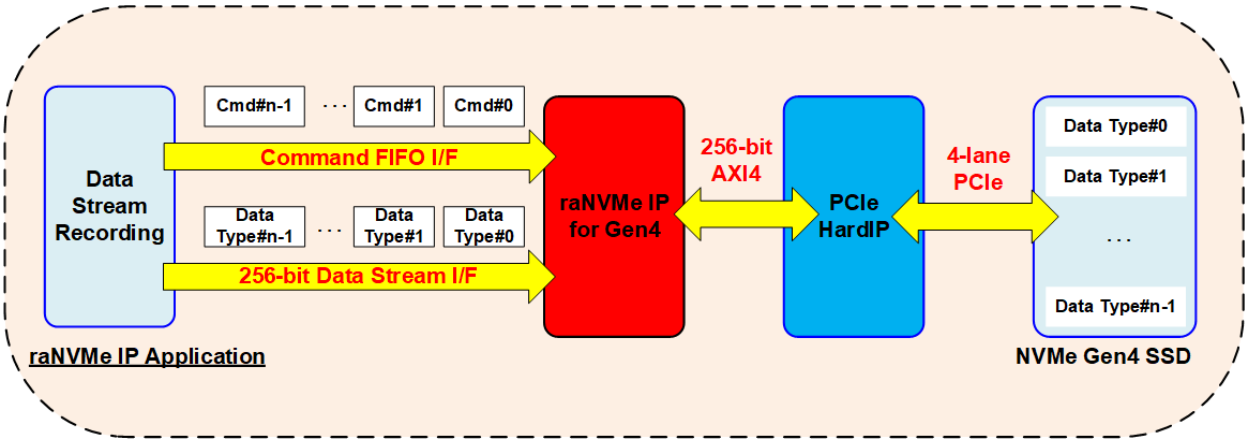

Figure 1: raNVMe IP Application

raNVMe IP Core for Gen4 integrated with Integrated Block for PCI Express (PCIe hard IP) from Xilinx is ideal to access NVMe Gen4 SSD without CPU and external memory in random access. One NVMe Gen4 SSD is connected to the system directly without PCIe switch. When the application has many data types that must be stored in the same SSD, each data type is generally stored at the different area. Therefore, the Write command for storing data stream is sent with random addressing. By using raNVMe IP for Gen4, the user logic can send up to 256 Write commands on Command FIFO I/F while 4 KB data of each Write command is transferred on 256-bit Data stream I/F without waiting the completion returned from the SSD.

On the contrary, the user can send up to 256 Read commands to read 256 data types that store at the different address by using raNVMe IP solution. It is fit to the data search system that requires to read many data from the different area of SSD.

General Description

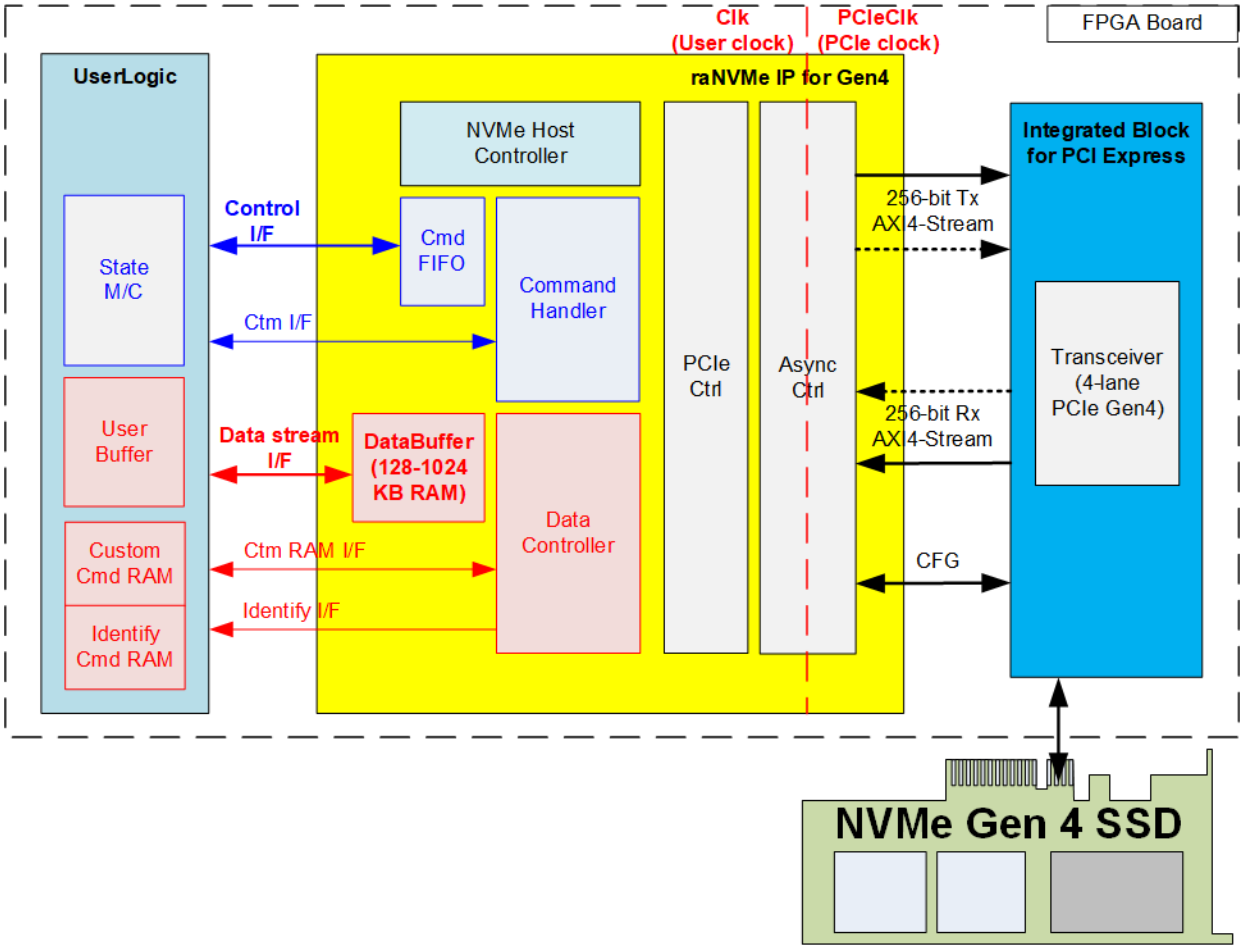

Figure 2: raNVMe IP for Gen4 block diagram

raNVMe IP for Gen4 implements the host controller to access NVMe Gen4 SSD with random access function following NVM express standard. Physical interface of NVMe SSD is PCIe which is handled by Integrated Block for PCI Express (PCIe hard IP) in the Xilinx FPGA.

raNVMe IP supports six NVMe commands, i.e., Identify, Shutdown, Write, Read, SMART, and Flush command. As shown in Figure 2, Control interface that is applied for requesting command with parameter assignments uses the FIFO interface. Therefore, it can support multiple Write/Read commands, requested by the user. While the Data interface has many interfaces for supporting the different command. The data interface of Write/Read command uses Data stream interface, 256-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 Gen4).

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

Functional Description

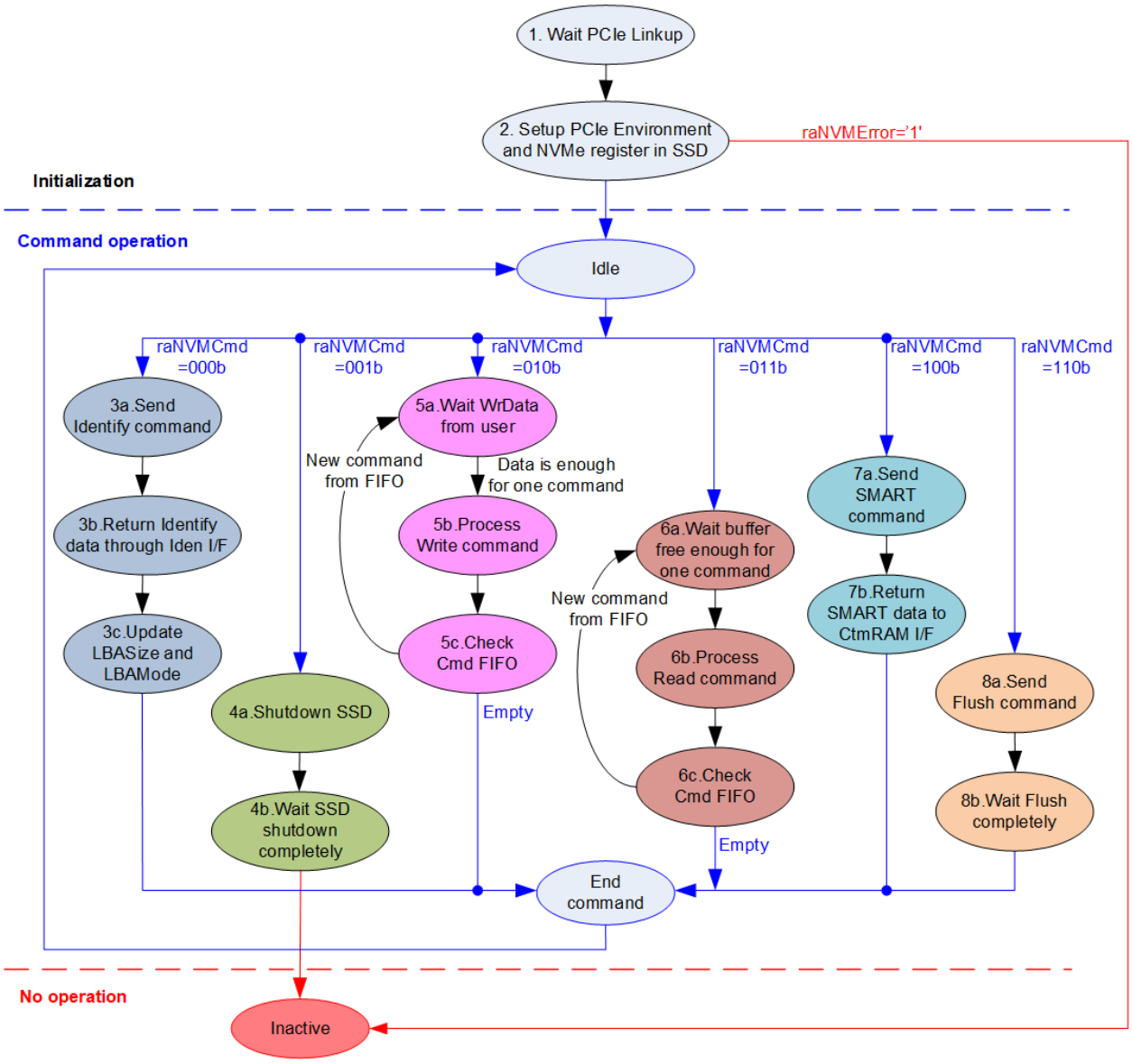

Figure 3: raNVMe IP Operation Flow

Figure 3 shows the operation of raNVMe IP is divided into three phases, i.e., Initialization, Command operation, and No operation. After power-on sequence, the IP is operated in Initialization phase to set up the hardware environment. After that, the IP enters to Command operation phase to operate the command that is requested by the user. Finally, when the IP operates Shutdown command completely, it enters to No operation phase which has no operation and wait until all hardware is reset.

The operation flow of raNVMe 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 configurating 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 enters to the Inactive state with asserting raNVMError to '1'.
3) The first command from user must be Identify command (raNVMCmd=000b) to update LBASize (disk capacity).
4) The last command before power down the system must be Shutdown command (raNVMCmd=001b). 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, raNVMe IP and SSD change to the Inactive state. The new command cannot be operated until the IP is reset.
5) For Write command (raNVMCmd=010b), while command is operated, user can request new Write commands until Cmd FIFO is full. The IP repeats step 5a) – 5c) until all Write commands are executed.
   a) The IP waits until Write data, sent by user, is enough for one command (4Kbytes).
   b) The IP sends Write command to SSD and then wait the next operation in step 5c).
   c) The IP checks whether there are new Write commands in Cmd FIFO. If a new Write command exists in the FIFO, the IP returns to step 5a). Otherwise, The IP waits until all Write command has complete by monitoring Cmd FIFO empty. After Cmd FIFO is empty, the IP returns to Idle status.
6) For Read command (raNVMCmd=011b), the data size for one command is fixed as 4 Kbytes. While command operating a Read command, user can request new Read commands until Cmd FIFO is full. The IP repeats step 6a) – 6c) until all Read commands are executed.
   a) The IP waits until free space of Data Buffer in raNVMe IP is enough for one command (4Kbytes).
   b) The IP sends Read command to SSD.
   c) After that, the IP checks whether there are new Read commands in Cmd FIFO. If a new Read command exists in the FIFO, the IP returns to step 6a) to process the next Read command. Otherwise, the IP waits until all Read data are completely transferred from Data Buffer to UserLogic by monitoring Cmd FIFO empty. After Cmd FIFO is empty, the IP returns to Idle status.
7) For SMART command (raNVMCmd=100b), 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 – 0x00F).
8) For Flush command (raNVMCmd=110b), 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 supporting random access, raNVMe IP for Gen4 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 raNVMe IP for Gen4 that can be split to two groups – NVMe and PCIe. More details of each module are described as follows.

NVMe

Six commands that are supported by raNVMe IP can be divided to two groups, i.e., Single-mode and Multiple-mode. When user sends Single-mode command request, the new command cannot be requested to raNVMe IP until the previous command is completed. For Multiple-mode command, the user can send multiple commands which are all the same to raNVMe IP until the Cmd FIFO inside raNVMe IP is full.

raNVMe IP supports four Single-mode commands – Identify, SMART, Flush, and Shutdown. While the Multiple-mode command consists of Write command and Read command. The number of Multiple-mode commands that can be stored to Cmd FIFO without waiting the command completion is configured by the Command Depth parameter. When user sends multiple commands, the order of the data block on Data stream I/F is the same as the order of the command on Cmd FIFO (Control I/F). One Mulitple-mode command requires 4 Kbyte data transferring on Data stream I/F. Since the Command Depth can be configured, the Data buffer size depends on the Command Depth value (one Command requires 4Kbyte data). For example, 128-Kbyte buffer is used to support 32 Commands and 1-Mbyte buffer is used to support 256 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 order of each packet type returned from the SSD is monitored by NVMe host controller. The status value in the received packet is decoded 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**

  The maximum number of Multiple-mode commands can be configured by Command Depth parameter (CDepth – the input parameter of raNVMe IP as shown in Table 3). It can be set by four values, i.e., 32, 64, 128, and 256. This value is applied to be the depth of Cmd FIFO inside raNVMe IP.

  Cmd FIFO is applied to store the address of Write or Read command which is Multiple-mode command. raNVMe IP determines that all addresses inside Cmd FIFO are applied for the same NVMe command which is Write or Read. If the user needs to change the NVMe command, the user must wait until the operation of all previous command is done and the Cmd FIFO is empty.

- **Command Handler**

  This module creates command packet and decodes the status packet returned from the SSD after each command is done. 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 status outputs for user monitoring 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, and 128-bit Custom completion status defined by 4 DWORDs when running SMART of Flush command.

- **Data Buffer**

  There is a Data buffer in raNVMe IP which is implemented by URAM for storing the raw data of Write command and Read command transferring between UserLogic and the SSD. The size of Data buffer is related to Command Depth parameter. One Write/Read command requires 4Kbyte buffer size. The Data buffer size of each Command Depth value is shown in Table 2.

Table 2: Data buffer size for each Command Depth value

Command Depth	Data Buffer size
32 commands	128 Kbyte
64 commands	256 Kbyte
128 commands	512 Kbyte
256 commands	1 Mbyte

- **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 Data buffer when running Write or Read command, Custom command RAM interface when running SMART command, and Identify interface when running Identify command.

  The data controller must allocate 4Kbyte size for transferring the data per Write/Read command. Also, the Data controller must handle the order of 4Kbyte data transferred with the user to be similar to the order of the Write/Read Command from user.

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**

  In initialization process, PCIeCtrl sets up PCIe environment of SSD via CFG interface. After that, PCIe packet is created or decoded via 256-bit Tx/Rx AXI4-Stream. The command packet and data packet from NVMe module 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. Most logics in raNVMe IP run on User clock domain while PCIe hard IP runs on PCIe clock domain. AXI4-stream interface of PCIe hard IP must transfer data of each packet continuously, so the user interface bandwidth must be greater than or equal to the PCIe interface bandwidth by setting User clock frequency at higher than or the same as PCIe clock frequency.

User Logic

This module could be designed by using small state machine to send the commands with assigning the parameters for each command. The raNVMe IP supports multiple Write/Read command, so user can send the command to Control interface (Cmd FIFO) and transfer the data by Stream interface (valid/ready signal) individually. While the data output of SMART command and Identify command can be mapped to simple dual port RAM with byte enable. All data interfaces are 256-bit interface. The returned data size of Identify command is 8-Kbyte while the returned data size of SMART command is 512-byte.

Integrated Block for PCI Express

Some UltraScale+ devices and Versal devices have Integrated Block for PCI Express (PCIe hard IP) to operate PCIe Gen4 protocol, called PCIe4C and PL PCIE4. To connect with raNVMe IP, PCIe4C or PL PCIE4 is configured to be 4-lane PCIe Gen4 by using 256-bit data interface. One NVMe IP connects to one PCIe hard IP for controlling one NVMe Gen4 SSD. Therefore, the maximum number of SSDs connecting to one FPGA device is limited by the number of PCIe hard IPs in FPGA device.

The IP wizard on Xilinx tool to generate PCIe hard IP when selecting UltraScale+ and Versal device are different. On UltraScale+ device, the user uses one IP wizard to generate the PCIe hard IP integrating with the transceiver. On Versal device, the user needs to call two IP wizards to generate PCIe hard IP and the Transceiver individually.

More details of PCIe hard IP are described in following document.

PG213: UltraScale+ Devices Integrated Block for PCI Express
https://www.xilinx.com/products/intellectual-property/pcie4-ultrascale-plus.html#documentation
PG343: Versal ACAP Integrated Block for PCI Express
https://www.xilinx.com/products/intellectual-property/pcie-versal.html#documentation

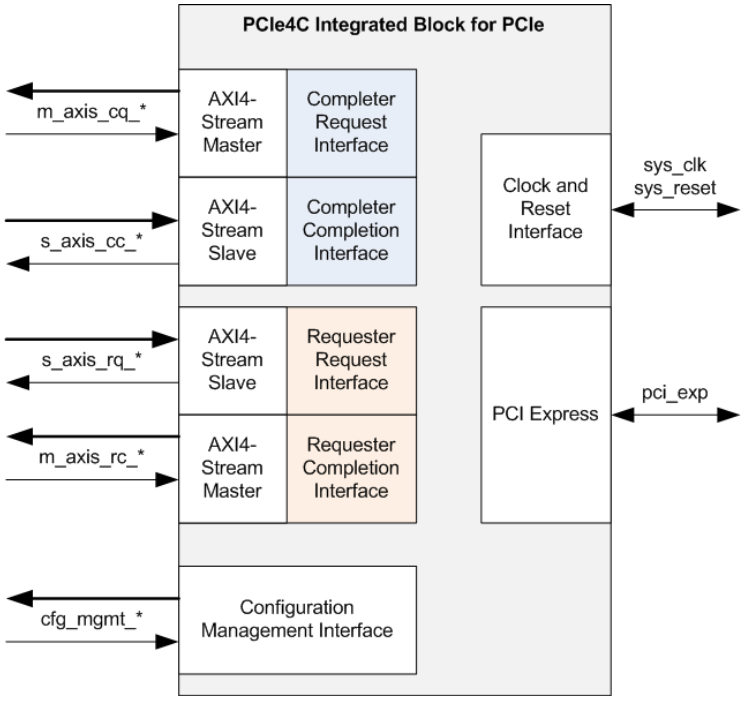

Figure 4: PCIe4C Integrated Block for PCI Express

Core I/O Signals

Descriptions of all signal I/O are provided in Table 3 and Table 4.

Table 3:Core Parameters

Name	Value	Description
CDepth	5-8	Command Queue Depth. Valid value is 5 - 8. Command Queue Depth = 2^{CDepth}. 5: 32 Commands, 6: 64 Commands, 7:128 Commands, 8:256 Commands.

Table 4: User logic I/O Signals (Synchronous to Clk signal)

Signal	Dir	Description
Control Interface of raNVMe IP for Gen4		
RstB	In	Synchronous reset signal. Active low. De-assert to '1' when Clk signal is stable.
Clk	In	User 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 Gen4).
raNVMCmd[2:0]	In	User command. Valid when raNVMCValid ='1'. (000b: Identify, 001b: Shutdown, 010b: Write SSD, 011b: Read SSD, 100b: SMART, 110b: Flush, 101b/111b: Reserved). *Note: User cannot change to run other commands when the IP is busy. User must wait until the current command is finished (raNVMBusy='0').*
raNVMAddr[47:0]	In	Start address of requested Write/Read command in 512-byte unit. Valid when raNVMCValid ='1'. This value must be less than LBASize which is the SSD capacity. *Note: raNVMAddr[2:0] is ignored for 4-Kbyte alignment.*
raNVMCValid	In	Command valid. Assert 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 raNVMCValid to accept only one command. When command is Multiple mode (Write and Read), user can assert raNVMCValid for sending the same command request but different address until 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[8:0]	Out	Remaining command count stored in the Cmd FIFO while operating Write/Read command. The value is in range 0 to Command Depth. This signal is ignored for Single-mode command. For example, when Command Depth = 32, raNVMCCnt is in range 0 to 32.
raNVMCId[7:0]	Out	Command ID in the Cmd FIFO for monitoring Write/Read command operation. The value is in range 0 to (Command Depth – 1). Valid when raNVMCValid='1'. This signal is ignored for Single-mode command. It is 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[23:16]. For example, when Command Depth = 32, raNVMCId is in range 0 to 31.
LBASize[47:0]	Out	Total capacity of SSD in 512-byte unit and always aligned to 4 KB unit. This value is valid after Identify command is done. So, Identify command must be the first command requested by the user to update LBASize value before running other commands. Default value after boot up is 0.
raNVMError	Out	Error flag. Asserted to '1' when raNVMErrorType is not equal to 0. The flag can be cleared by asserting RstB to '0'.

Signal	Dir	Description
Control Interface of raNVMe IP		
raNVMErrorType[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 less than (2 x Command Depth - 1). 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. [5] – Error when status register in IO completion entry is not equal to 0 or phase tag is invalid. Please see more details from IOCompStatus signal. [6] – Error when Completion TLP packet size is not correct. [7] – Error when PCIe hard IP detects Error correction code (ECC) error from the internal buffer. [8] – Error from Unsupported Request (UR) flag in Completion TLP packet. [9] – Error from Completer Abort (CA) flag in Completion TLP packet. [15:10] – Reserved [16] - Error from unsupport LBA unit (LBA unit is not equal to 512-byte) [31:17] – Reserved *Note: Timeout period of bit[2]/[4] is set byTimeOutSet input.*
Data Stream Interface of raNVMe IP		
raNVMDId[7:0]	Out	Show Command ID (raNVMCId) which is currently transferred on Data stream interface when running Multiple-mode commands (Write or Read command). The value is in range 0 to (Command Depth – 1), similar to raNVMCId. It is valid when raNVMwValid='1' or raNVMrValid='1'.
raNVMwData[255:0]	In	Write data for Write command. Valid when raNVMwValid='1'.
raNVMwValid	In	Assert to '1' to write data to raNVMe IP while operating Write command. Write data can be transferred before sending Write command but the IP must be in Idle status. *Note: All Write data that is stored in the buffer will be flushed, if other command which is not Write command is operating.*
raNVMwReady	Out	Asserted to '1' when raNVMe IP accepts the write data (raNVMwData).
raNVMwCnt[15:0]	Out	Total data count of write data in the buffer in 256-bit unit. The value is in range 0 to 128xCommand Depth. Valid when running Write command or the IP is in Idle status. When running other commands, this signal is not valid. Also, this signal is reset to 0 after running other commands.
raNVMrData[255:0]	Out	Read data from SSD in Read command. Valid when raNVMrValid ='1'.
raNVMrValid	Out	Read data valid signal. Asserted to '1' to transfer Read data. The user can pause data transmission by asserting raNVMrPause to '1'. After that, raNVMrValid is de-asserted to '0' within 4 clock cycles.
raNVMrPause	In	Asserted to '1' to pause Read data (raNVMrData) transmission by de-asserting raNVMrValid to '0' within 4 clock cycles.

raNVMe IP Interface		
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.
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 [23:16] – Command ID of IO Completion Entry, referred to raNVMCId [31:24] – Reserved
NVMeCAPReg[31:0]	Out	The parameter value of the NVMe capability register when raNVMErrorType[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
IdenWrEn	Out	Asserted to '1' for sending data output from Identify command.
IdenWrDWEn[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] correspond to IdenWrData[31:0], [63:32], .. , [255:224], respectively.
IdenWrAddr[7:0]	Out	Index of IdenWrData in 256-bit unit. Valid when IdenWrEn='1'. Bit[7]='0': 0x00-0x7F is 4Kbyte Identify controller data. Bit[7]='1': 0x80-0xFF is 4Kbyte Identify namespace data.
IdenWrData[255:0]	Out	4Kbyte Identify controller data or Identify namespace data. Valid when IdenWrEn='1'.
Custom interface		
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=100b (SMART) or 110b (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.
CtmRamWrDWEn[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], [2], …, [7] correspond 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 from 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 5: Physical I/O Signals for PCIe Gen4 Hard IP (Synchronous to PCIeClk)

Signal	Dir	Description
PCIe Gen4 hard IP		
PCIeRstB	In	Synchronous reset signal. Active low. De-assert to '1' when PCIe hard IP is not in reset state.
PCIeClk	In	Clock output from PCIe hard IP (250 MHz for PCIe Gen4).
PCIeLinkup	In	Assert to '1' when PCIe hard IP is linked up.
Configuration Management Interface		
PCIeCfgDone	In	Read/Write operation complete. Assert to '1' for 1 cycle when the operation completes.
PCIeCfgRdEn	Out	Read enable. Asserted to '1' for a read operation.
PCIeCfgWrEn	Out	Write enable. Asserted to '1' for a write operation.
PCIeCfgWrData[31:0]	Out	Write data which is used to configure the Configuration and Management registers.
PCIeCfgByteEn[3:0]	Out	Byte enable for Write data, where bit[0],[1],[2], and [3] corresponds to PCIeCfgWrData[7:0], [15:8], [23:16] and [31:24] respectively.
PCIeCfgAddr[9:0]	Out	Read/Write Address.
Requester Request Interface		
PCIeMtTxData[255:0]	Out	Requester request data bus.
PCIeMtTxKeep[7:0]	Out	Bit [i] indicates that Dword [i] of PCIeMtTxData contains valid data; i=0-7.
PCIeMtTxLast	Out	Asserted this signal in the last cycle of a TLP to indicate the end of the packet.
PCIeMtTxReady[3:0]	In	Assert to accept data. Data is transferred when both PCIeMtTxValid and PCIeMtTxReady are asserted in the same cycle.
PCIeMtTxUser[61:0]	Out	Requester request user data. Valid when PCIeMtTxValid is high.
PCIeMtTxValid	Out	Asserted to drive valid data on PCIeMtTxData bus. raNVMe IP keeps the valid signal asserted during the transfer of packet.
Completer Request Interface		
PCIeMtRxData[255:0]	In	Received data from PCIe hard IP.
PCIeMtRxKeep[7:0]	In	Bit [i] indicates that Dword [i] of PCIeMtRxData contains valid data; i=0-7.
PCIeMtRxLast	In	Assert this signal in the last beat of a packet to indicate the end of the packet.
PCIeMtRxReady	Out	Indicates that raNVMe IP is ready to accept data.
PCIeMtRxUser[74:0]	In	Sideband information for the TLP being transferred. Valid when PCIeMtRxValid is high.
PCIeMtRxValid	In	Assert when PCIe hard IP drives valid data on PCIeMtRxData bus. PCIe hard IP keeps the valid signal asserted during the transfer of packet.
Completer Completion Interface		
PCIeSlTxData[255:0]	Out	Completion data from raNVMe IP.
PCIeSlTxKeep[7:0]	Out	Bit [i] indicates that Dword [i] of PCIeSlTxData contains valid data; i=0-7.
PCIeSlTxLast	Out	Asserted this signal in the last cycle of a packet to indicate the end of the packet.
PCIeSlTxReady[3:0]	In	Indicates that PCIe hard IP is ready to accept data.
PCIeSlTxUser[32:0]	Out	Sideband information for the TLP being transferred. Valid when PCIeSlTxValid is high.
PCIeSlTxValid	Out	Asserted to drive valid data on PCIeSlTxData bus. raNVMe IP keeps the valid signal asserted during the transfer of a packet.
Requester Completion Interface		
PCIeSlRxData[255:0]	In	Received data from PCIe hard IP.
PCIeSlRxKeep[7:0]	In	Bit [i] indicates that Dword [i] of PCIeSlRxData contains valid data; i=0-7.
PCIeSlRxLast	In	Asserted this signal in the last beat of a packet to indicate the end of the packet.
PCIeSlRxReady	Out	Indicates that raNVMe IP is ready to accept data.
PCIeSlRxUser[87:0]	In	Sideband information for the TLP being transferred. Valid when PCIeSlRxValid is high.
PCIeSlRxValid	In	Asserted when PCIe hard IP drives valid data on PCIeSlRxData bus. PCIe hard IP keeps the valid signal asserted during the transfer of packet.

Timing Diagram

Initialization

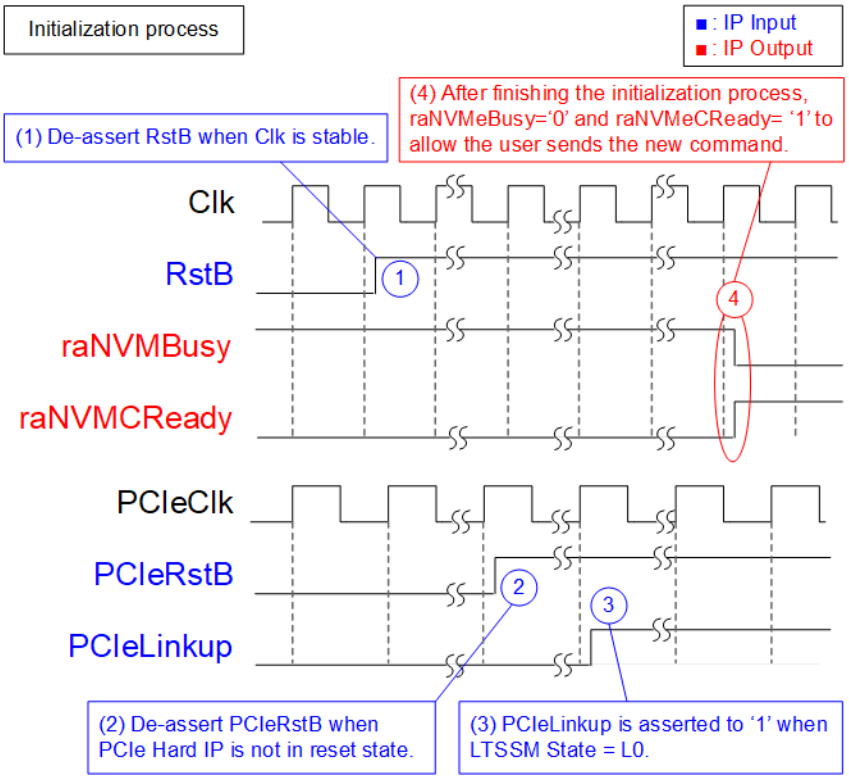

Figure 5: Timing diagram during initialization process

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 transfer data with the application layer.
3) PCIe hard IP asserts PCIeLinkup to '1' after LTSSM state of PCIe hard IP is L0 state. After that, raNVMe IP starts initialization process.
4) After finishing the initialization process, raNVMe IP de-asserts raNVMBusy to '0' and asserts raNVMCReady 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 (Maximum number of commands configured by Command Depth). 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.

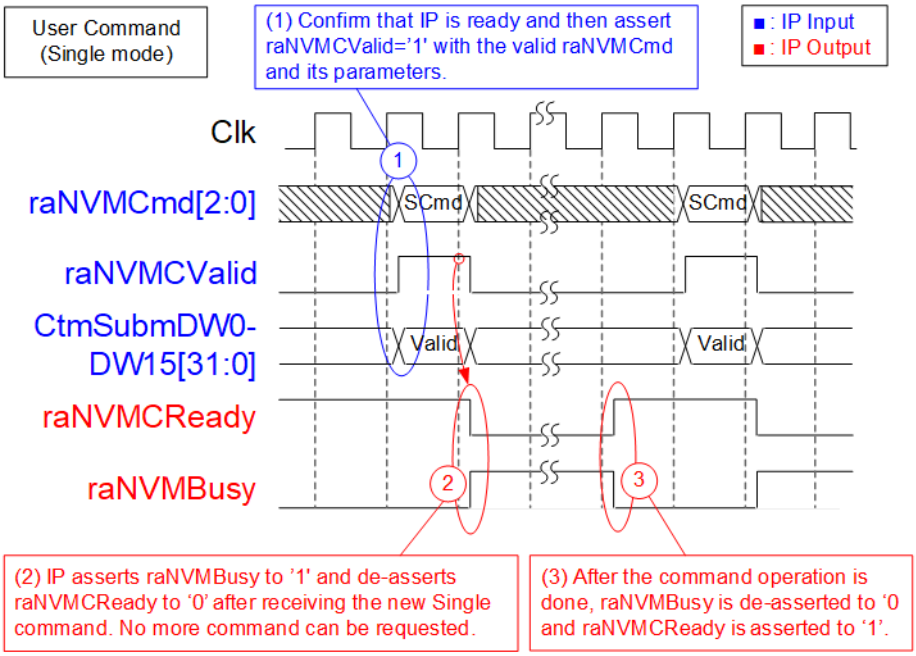

Figure 6: Single command timing diagram

Figure 6 shows timing diagram when running Single command, i.e., Identify, Shutdown, SMART, and Flush command.

1) Before sending the new command to the IP, user must check raNVMBusy='0' and raNVMCReady='1' to confirm that raNVMe IP is Idle. After that, assert raNVMCValid to '1' for one clock cycle along with valid raNVMCmd 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 required.
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 received command is Single mode. User must not send more commands requests.
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 done, both raNVMBusy and raNVMCReady are de-asserted to '0' to wait system shut down without receiving more command requests from user.

Control interface (Multiple-mode command)

For Write or Read command which are Multiple-mode command, user can send the same commands but different Start address to raNVMe IP until Cmd FIFO is full. The 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'.

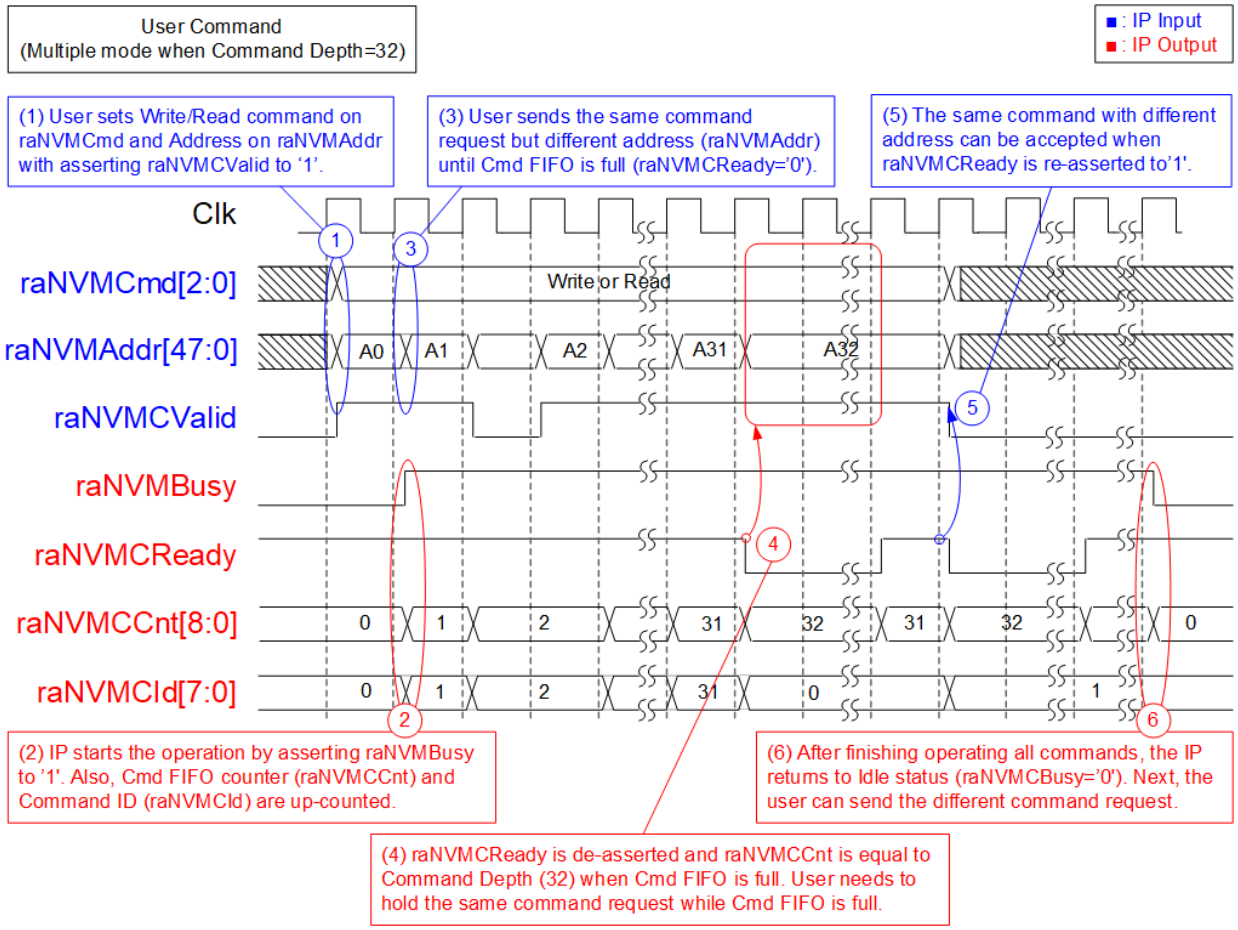

Figure 7: Multiple-mode command timing diagram

1) User must comfirm that raNVMe IP is Idle (raNVMBusy='0') and Cmd FIFO is ready (raNVMCReady='1'). Next, the user sends Write command (raNVMCmd=010b) or Read command (raNVMCmd=011b) to the IP by asserting raNVMCValid to '1' for one clock cycle along with the valid raNVMAddr.
2) IP starts the command operation with asserting raNVMBusy to '1'. Meanwhile, raNVMCCnt and raNVMCId are up-counted 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 Command Depth or raNVMCReady='1'), user can the new command request of the same command but the new address value. raNVMCmd must be the same as previous value when asserting raNVMCValid to '1' and the IP is busy (raNVMBusy='1'). The wrong operation will be happened if user sends the different command request while raNVMBusy='1'.
4) After Cmd FIFO is full (raNVMCCnt = Command Depth), raNVMCReady is de-asserted to '0'. The IP does not accept more command request from the user. All inputs for the new command request need to hold the same value until raNVMCReady is re-asserted again.
5) After some commands are completed, Cmd FIFO is not full (raNVMCCnt < Command Depth) and raNVMCReady is re-asserted. The new command request is accepted by the IP.
6) When raNVMe IP finishes all of the command request, raNVMBusy is de-asserted to '0'. raNVMe IP returns back to Idle status. Now the IP is ready to receive any commands which is different from the previous command.

Data stream interface (Write command)

Data stream interface of raNVMe is applied for transferring data while operating Write command or Read command. The control signals of Data stream interface are valid and ready signal. Figure 8 shows the example of data stream interface when running Write command.

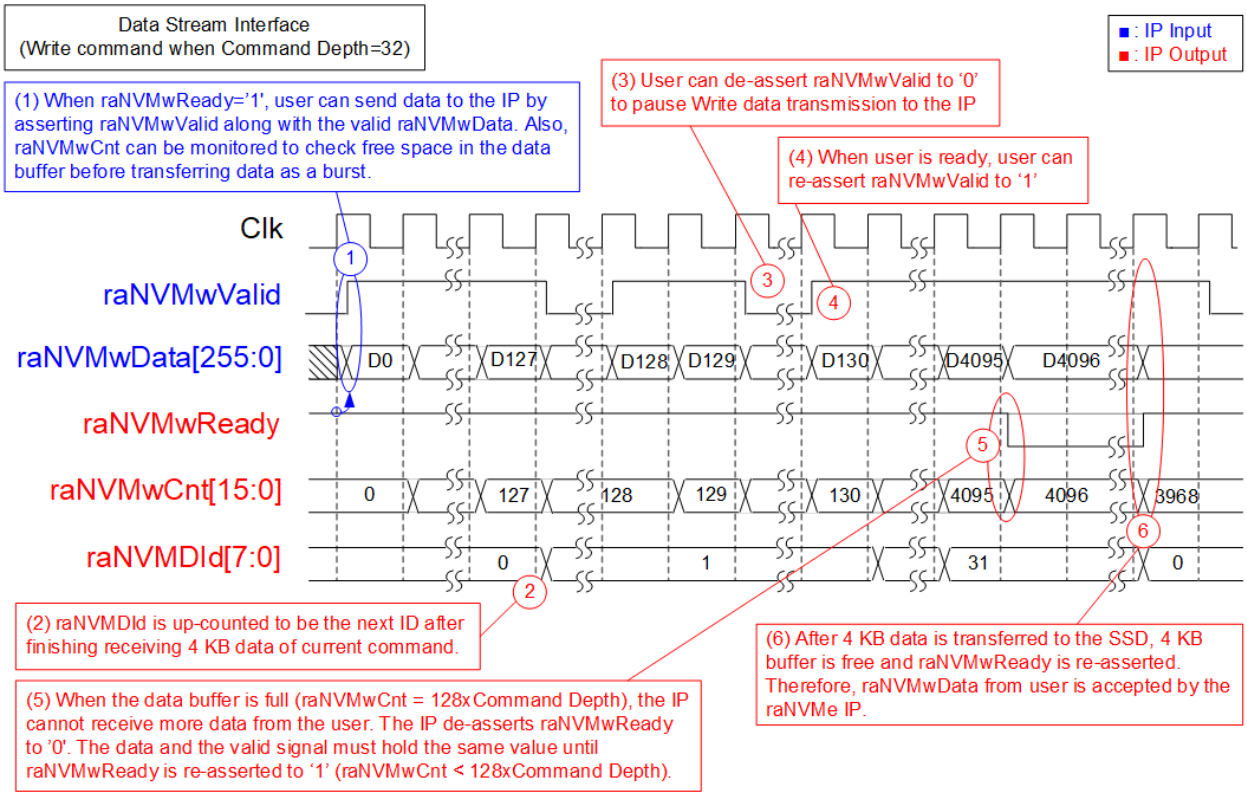

Figure 8: Data stream interface in Write command

Figure 8 shows timing diagram in Write command when Command Depth=32. raNVMe IP Data stream interface operates independently from the Control interface. User can store the data to 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 IP is ready to receive data by monitoring raNVMwReady='1' or raNVMwCnt < 128xCommand Depth (4096). After that, user asserts raNVMwValid along with valid raNVMwData to transfer 256-bit data. There are two status signals to be flow control signals for Write command - raNVMwReady and raNVMwCnt. raNVMwCnt is recommended for burst data transferring 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 up-counted to show the command ID of the next 4 Kbyte data transferred on the Data stream interface (raNVMwData).
3) When the user is not ready to send data, raNVMwValid can be de-asserted to '0' to pause transferring data.
4) When the user is ready to send data to IP, raNVMwValid can be re-asserted to '1' to transfer data.
5) When the data buffer is full (raNVMwCnt is equal to 128xCommand Depth or 4096), raNVMwReady is de-asserted to '0' and the IP cannot accept the new data from the user. Therefore, raNVMwData and raNVMwValid must hold the same value until raNVMwReady is re-asserted to '1'.
6) If the previous Write command is complete, the 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)

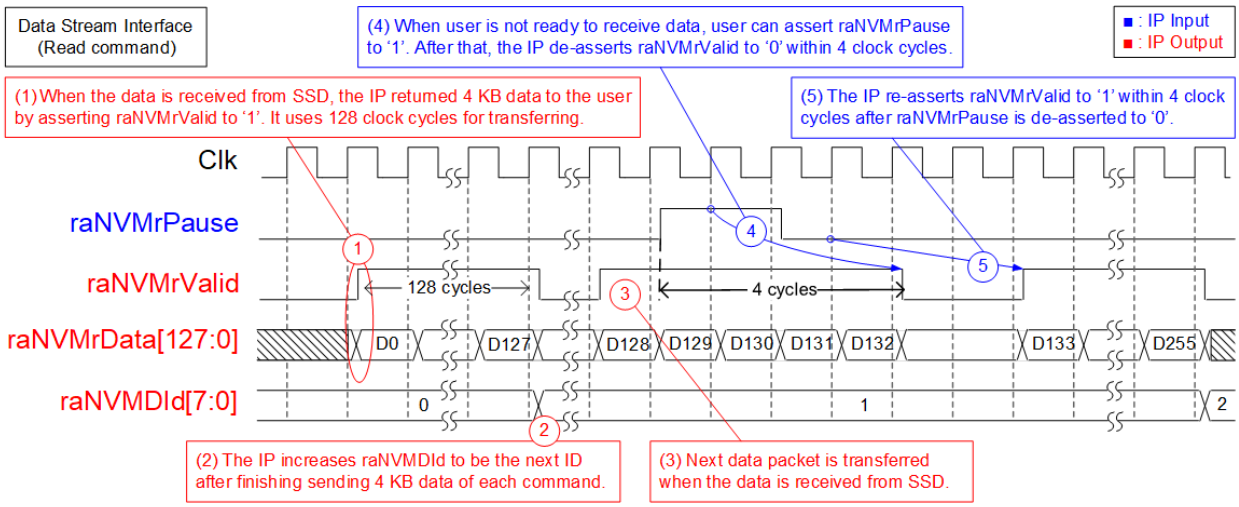

Figure 9: Data stream interface in Read command

Similar to Write command, the Data stream interface of the IP is run independently from the Control interface. The control signals of Read data stream are valid and pause signal. 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 4 KB data to the user by asserting raNVMrValid to '1' along with valid raNVMrData. To transfer 4 KB data, raNVMrValid is asserted to '1' for 128 clock cycles continuously if raNVMrPause is always de-asserted to '0'.
2) After finishing transferring each 4 KB data, raNVMDId is up-counted to show the command ID of the next 4 KB data that will be transferred.
   Note: When the Command Depth = 32, raNVMDId is in range 0 to 31.
3) The next 4 KB data is transferred after the SSD returns the 4KB data, similar to step 1).
4) When user is not ready to receive read data, user can assert raNVMrPause to '1'. After that, raNVMe IP pauses transferring data by de-asserting raNVMrValid to '0' within 4 clock cycles.
5) When user is ready to receive read data by de-asserting raNVMrPause to '0', raNVMe IP re-asserts raNVMrValid to '1' within 4 clock cycles to continue to transfer the data.

IdenCtrl/IdenName

It is recommened 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) in Write or Read command. UserAddr in Write and Read command must be less than LBASize of the SSD.

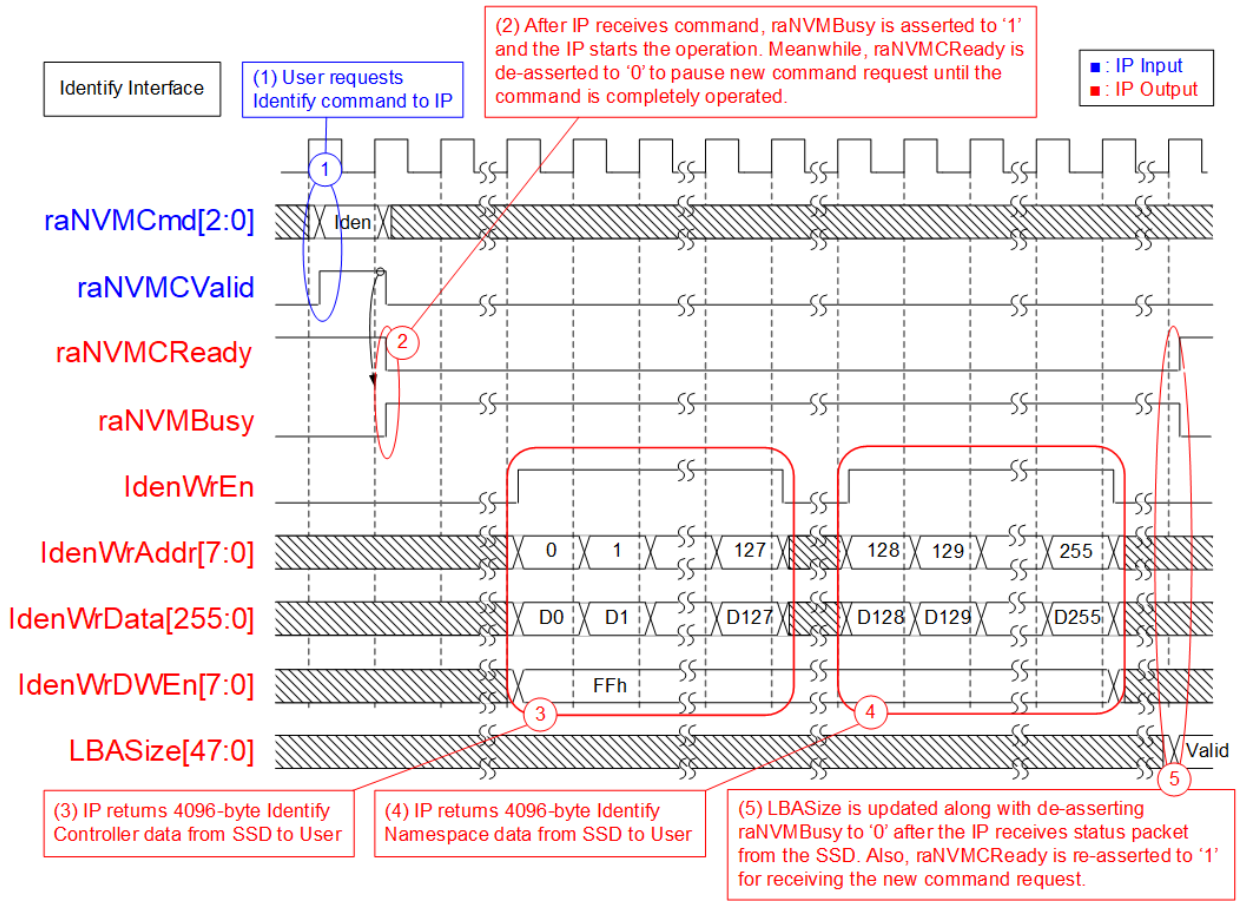

Figure 10: Identify command timing diagram

The details when running Identify command are shown as follows.

1) Send the request of Identify command to the IP (raNVMCmd=000b and raNVMCValid='1').
2) The IP asserts raNVMBusy to '1' after running the command. Meanwhile, 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 0-127 with asserting IdenWrEn. IdenWrData and IdenWrDWEn are valid at the same clock as IdenWrEn='1'.
4) Similar to Identfy controller data, 4096-byte Identify namespace data is returned but IdenWrAddr is equal to 128-255. IdenWrAddr[7] 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.

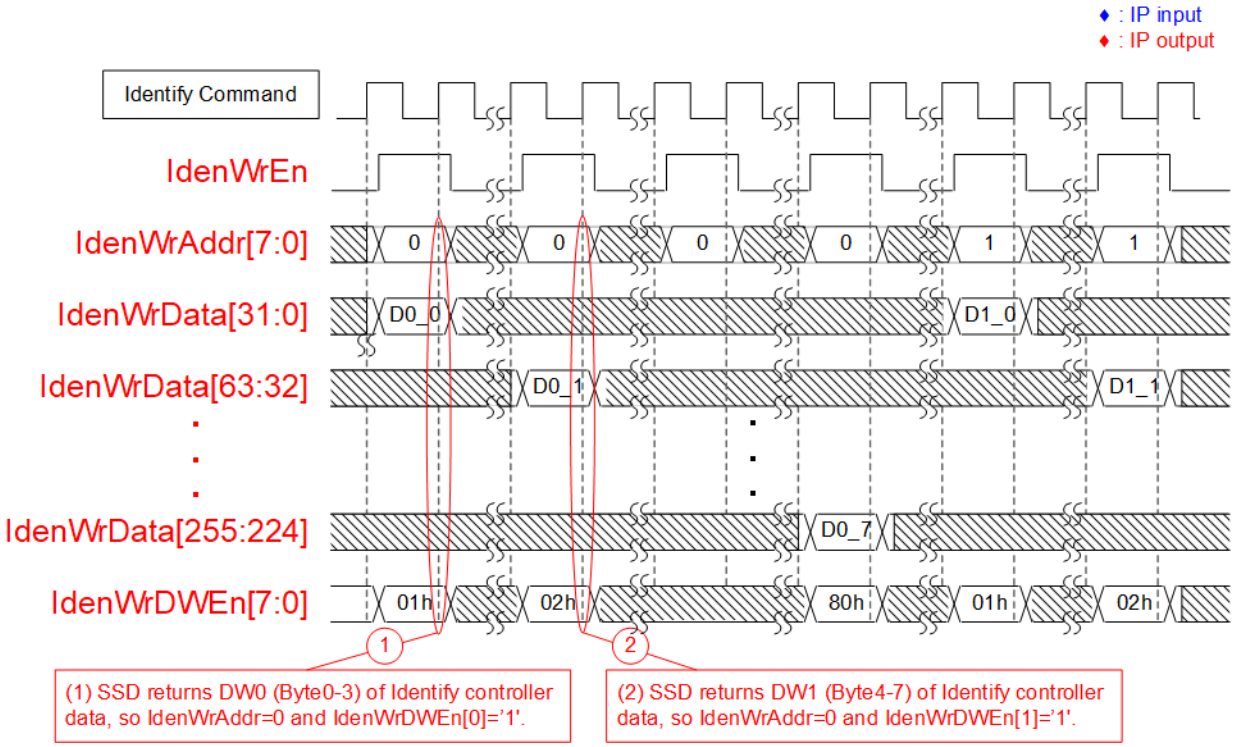

Figure 11: IdenWrDWEn timing diagram

IdenWrDWEn 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 it returns 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 11. IdenWrDWEn[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 Shutdown command operation is done, 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.

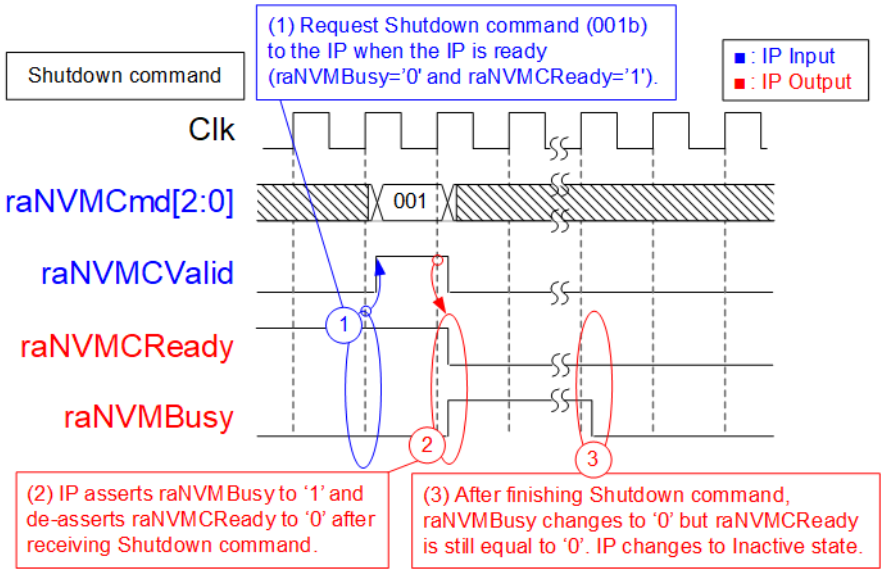

Figure 12: Shutdown command timing diagram

The details when running Shutdown command is shown as follows.

1) Before sending the command request, the IP must be Idle status (raNVMBusy='0' and raNVMCReady='1'). To send Shutdown command, user asserts raNVMValid to '1' with raNVMCmd =001b.
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 13.

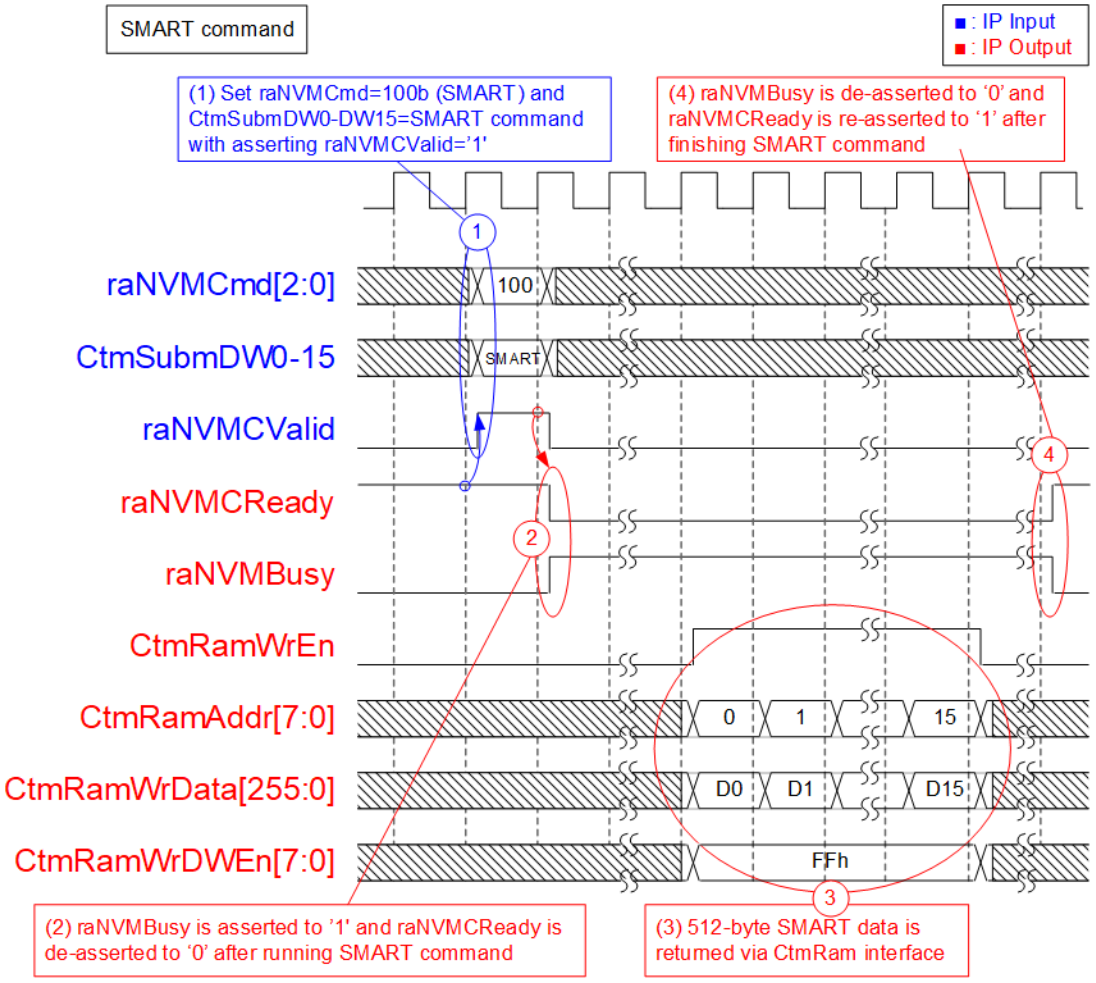

Figure 13: SMART command timing diagram

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=100b 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 0-15 to be data index of 512-byte data. When CtmRamAddr=0, byte0-31 of SMART data is valid on CtmRamWrData. CtmRamWrDWEn is Dword enable for each 32-bit CtmRamWrData. If CtmRamWrDWEn=FFh, all 256-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.

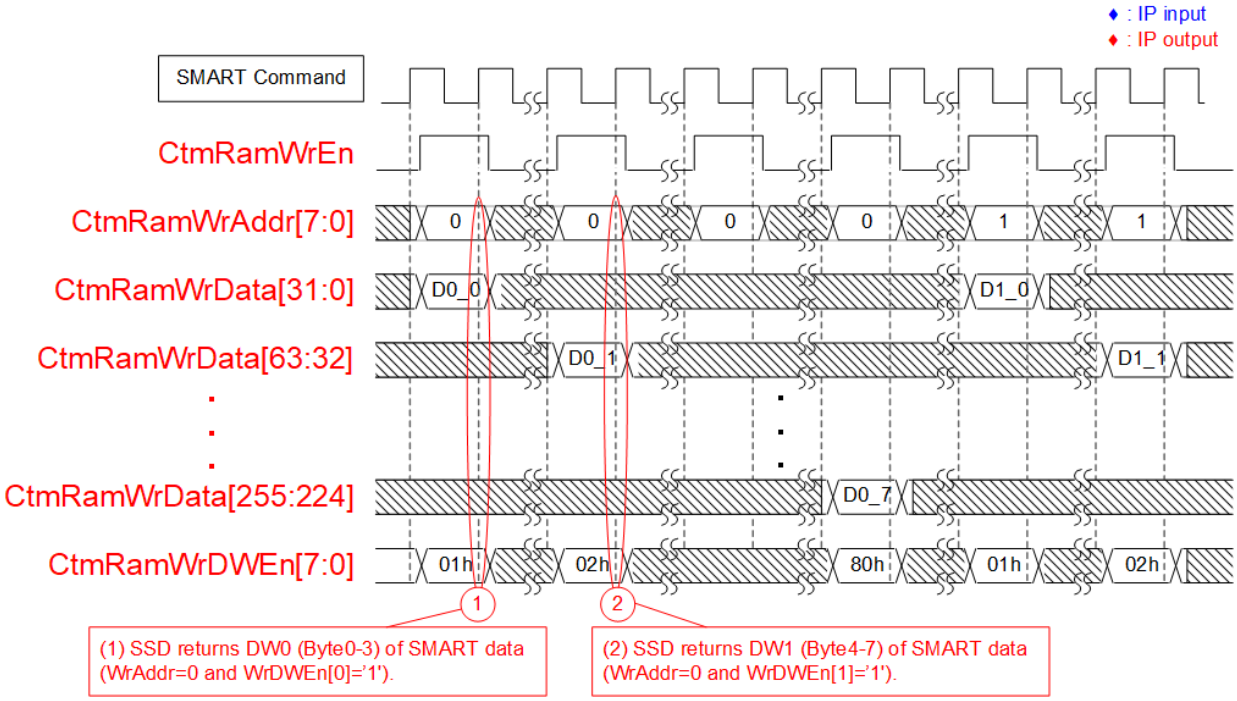

Figure 14: CtmRamWrDWEn timing diagram

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 CtmRamWrDWEn is asserted to '1' in the write cycle to be the valid signal of 32-bit CtmRamWrData. CtmRamWrDWEn[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.

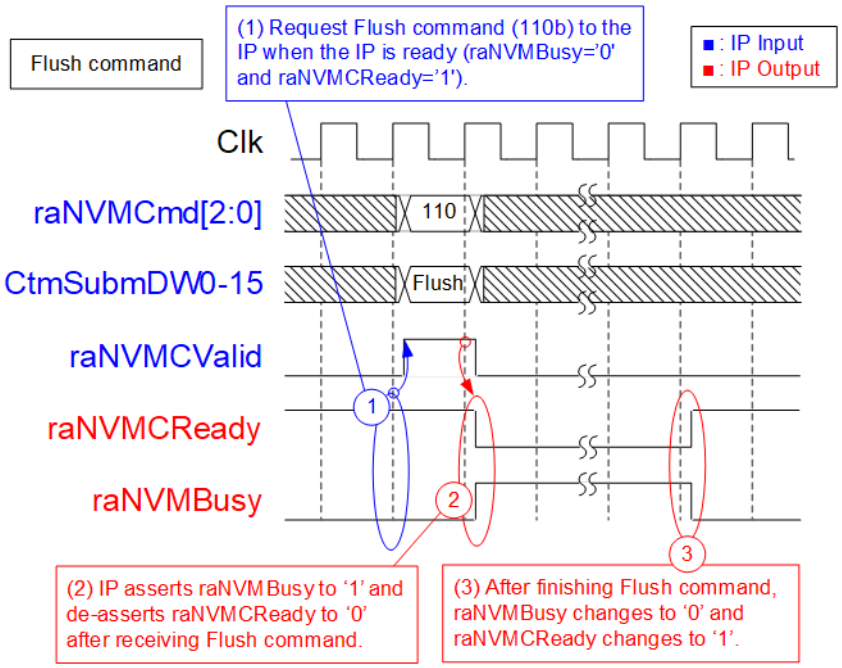

Figure 15: Flush command timing diagram

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 along with raNVMCmd=110b 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

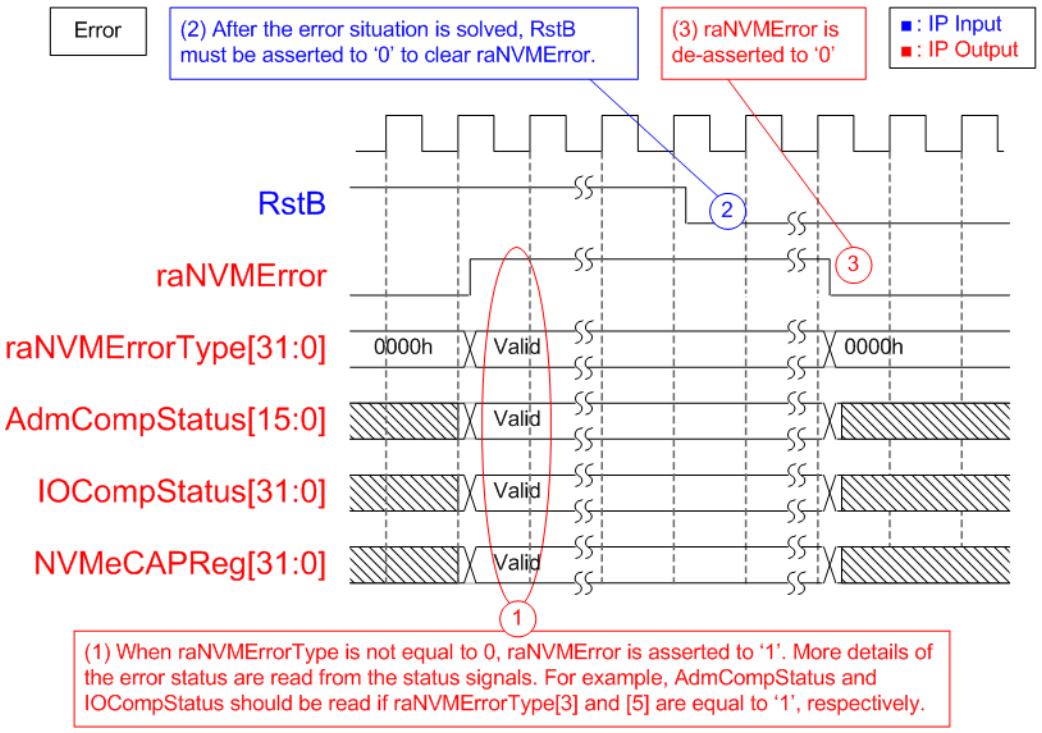

Figure 16: Error flag timing diagram

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

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

The raNVMError 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 raNVMe IP Core for Gen4 functionality was verified by simulation and also proved on real board design by using VCK190 evaluation board.

Recommended Design Experience

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

Ordering Information

This product is available directly from Design Gateway Co., Ltd. Please contact Design Gatway 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	11-Jan-2023	Initial Release