Design Gateway Co.,Ltd

54 BB Building 14th Fl., Room No.1402 Sukhumvit 21 Rd. (Asoke), Klongtoey-Nua, Wattana, Bangkok 10110
Phone: 66(0)2-261-2277
Fax: 66(0)2-261-2290
E-mail: ip-sales@design-gateway.com
URL: www.design-gateway.com

Core Facts

Provided with Core	
Documentation	Reference Design Manual Demo Instruction Manual
Design File Formats	Encrypted Netlist
Instantiation Templates	VHDL
Reference Designs & Application Notes	Vivado Project, See Reference Design Manual
Additional Items	Demo on ZCU102,VCU118,KCU105
Support	
Support Provided by Design Gateway Co., Ltd.	

Features

- Implement application layer, transaction layer, data link layer, and some parts of physical layer to access NVMe SSD without CPU usage
- Operating with Xilinx PCIe PHY IP, configured as 4-lane PCIe Gen3 (128-bit bus interface)
- Includes 256 Kbyte RAM to be data buffer
- Simple user interface by dgIF typeS
- Supports six commands, i.e. Identify, Shutdown, Write, Read, SMART, and Flush (support additional command as optional)
- 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 must be more than or equal to PCIe clock (250MHz for Gen3)
- Available reference design:
  - KCU105 with AB18-PCIeX16/AB16-PCIeXOVR adapter board.
  - ZCU102 with AB17-M2FMC adapter board
  - VCU118 with AB18-PCIeX16 adapter board

Table 1: Example Implementation Statistics for Ultrascale/Ultrascale+ device

Family	Example Device	Fmax (MHz)	CLB Regs	CLB LUTs	CLB	IOB	BRAMTile[1]	PLL	GTX	Design Tools
Kintex-Ultrascale	XCKU040FFVA1156-2E	300	18983	16973	3455	-	70	-	-	Vivado2017.4
Zynq-Ultrascale+	XCZU7EV-FFVC1156-2E	300	18982	17109	3690	-	70	-	-	Vivado2017.4
Virtex-Ultrascale+	XCVU9P-FLGA2104-2L	300	18983	16978	3483	-	70	-	-	Vivado2017.4

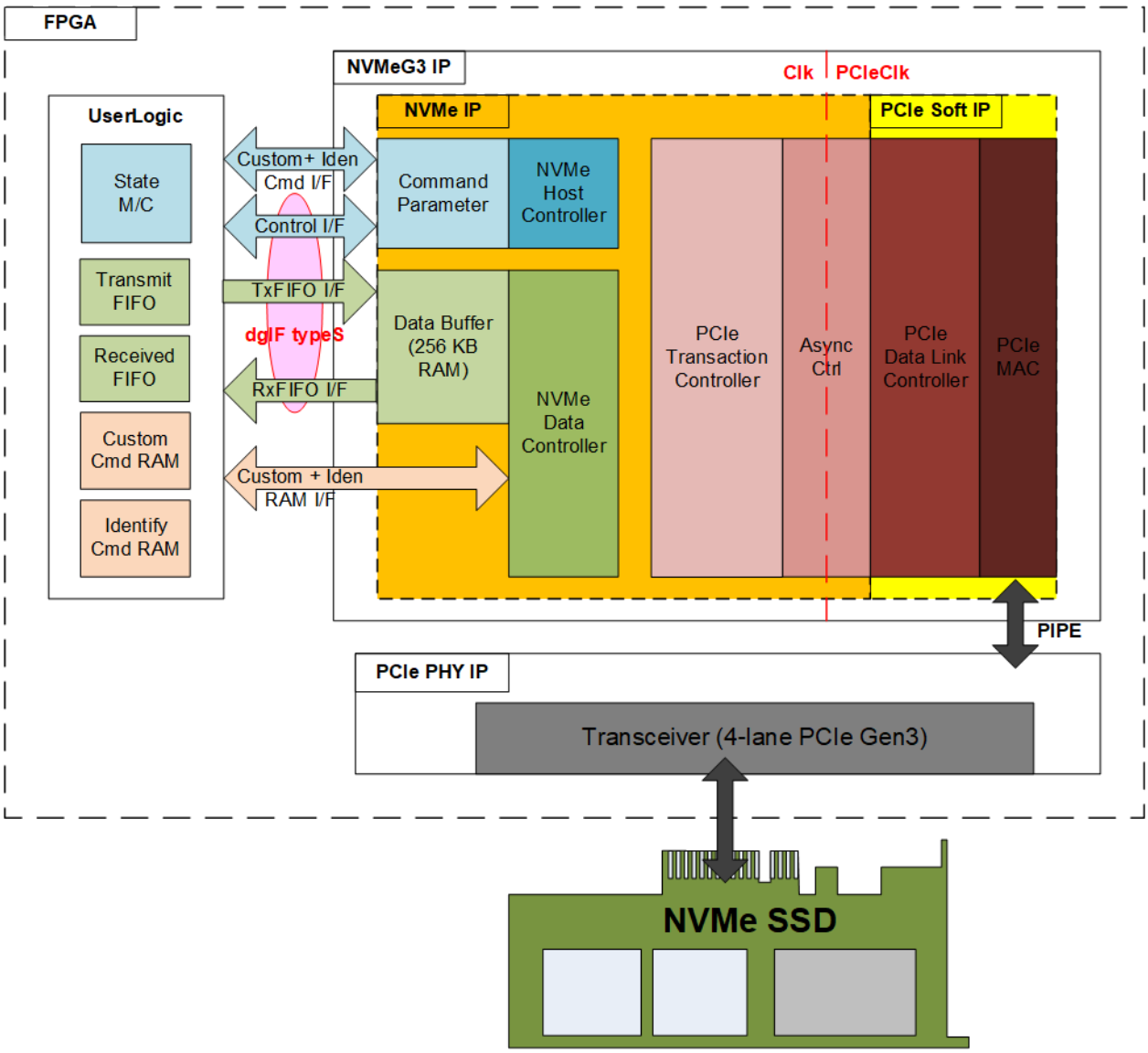

Figure 1: NVMeG3 IP Block Diagram

Applications

NVMe IP Core with PCIe Gen3 Soft IP (NVMeG3 IP) is ideal to access NVMe SSD without PCIe Hard IP, CPU, and external memory. NVMeG3 IP includes PCIe Gen3 Soft IP and 256 Kbyte memory. This solution is recommended for the application which requires very large storage with ultra high speed performance by using the low-cost FPGA which does not contain a PCIe Hard IP. When the selected FPGA integrates PCIe hard IP, it is recommended to use DG NVMe IP Core which utilizes smaller FPGA resource. Small size system is achieved by using M.2 SSD package.

General Description

Table 2: DG NVMe-IP comparison

Feature	NVMe IP	NVMeG3 IP
PCIe Interface	128-bit AXI4 Stream	128-bit PIPE
Xilinx PCIe IP	Integrated Block for PCIe (PCIe Hard IP)	PCIe PHY IP (Transceiver and equalizer)
PCIe Hard IP	Necessary	Not use
PCIe Speed	Gen3 or lower, depending on Xilinx IP	Support only 4-lane PCIe Gen3
User Interface	dgIF typeS	dgIF typeS
FPGA resource	Smaller	Larger
Maximum SSD	Depend on the number of PCIe Hard IPs	Depend on the number of transceivers
SSD Performance	Up to 3300 MB/s*	Up to 3300 MB/s*

*Note: This performance is real board test by reading data from 500GB NVMe SSD @ PCIe Gen3.

Design Gateway develops NVMeG3 IP to run as NVMe host controller for accessing NVMe SSD, like standard DG NVMe IP. The user interface and all standard features in NVMe IP have been included in NVMeG3 IP. The additional feature of NVMeG3 IP is the built-in PCIe soft IP implementing Data link layer and some parts of Physical layer of PCIe protocol by pure logic. So, NVMeG3 IP can run in FPGA which does not have PCIe Hard IP by using built-in PCIe soft IP and Xilinx PCIe PHY IP. Xilinx PCIe PHY IP is the IP core including the transceiver and the logic of equalizer.

As shown in Table 2, the main advantage of NVMeG3 IP is that PCIe hard IP is not necessary and the maximum number of SSD is not limited by the number of PCIe hard IP. The maximum number of connected SSDs is limited by the number of transceivers and the resource utilization. The disadvantage of NVMeG3 IP is the resource utilization which is larger for implementing the PCIe soft IP. Also, NVMeG3 IP supports only 4-lane PCIe Gen3 SSD.

Similar to NVMe IP, NVMeG3 IP supports six NVMe commands, i.e. Identify, Shutdown, Write, Read, SMART, and Flush command. 256 Kbyte BlockRAM is integrated in the NVMeG3 IP to be data buffer. The system does not need CPU and external memory. More details of the standard NVMe IP are described in NVMe IP datasheet which can be downloaded from our website.

https://dgway.com/products/IP/NVMe-IP/dg_nvme_ip_data_sheet_en.pdf

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

Functional Description

Figure 2 shows operation flow of NVMeG3 IP after IP reset is released.

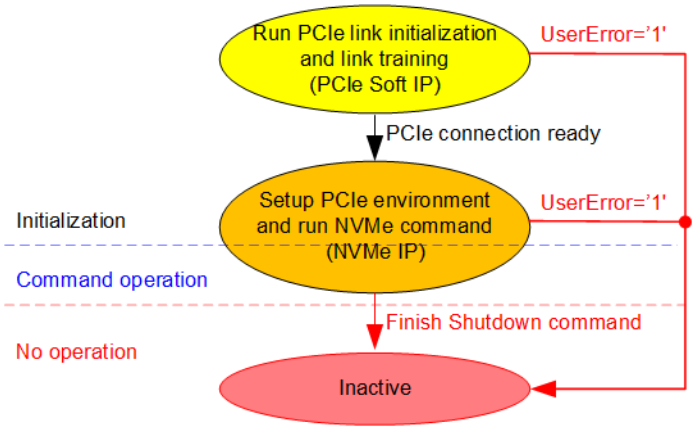

Figure 2: NVMeG3 IP Operation Flow

As shown in Figure 2, most of NVMeG3 IP operations are same as NVMe IP. The additional part is only the first step which runs PCIe link initialization and training. This process is controlled by the Physical layer to configure and initialize link and port. The process consists of several steps as follows.

1) Detects device connection by monitoring electrical idle signal.
2) Polling until Bit/Symbol of every lane is locked.
3) Sets the number of lanes to 4 lanes.
4) Sets PCIe speed to Gen3.
5) Adjusts the equalizer parameters.
6) Sets flow control parameters.

After finishing this step, the signal quality is in the good status and ready to transfer PCIe packet.

The next step is same process as NVMe IP. First, PCIe registers and NVMe registers are configured to finish NVMe IP initialization process. After that, the IP is ready to receive the command from user. There are six operations for six commands. When finishing Shutdown command or detecting some errors in initialization process, the IP goes to Inactive status which must be recovered by reset process. More details of NVMe IP operation flow are described in NVMe IP datasheet.

As shown in Figure 1, NVMeG3 IP consists of two modules, i.e. NVMe IP and PCIe Soft IP.

The user interface of NVMe IP has three groups. First one is dgIF typeS which is the standard storage interface of DG IP core. Data interface is 128-bit FIFO interface while control interface is simple interface by assigning address, length, and command number. Second is the control interface for custom command and Identify command. Third is the data interface of custom command and Identify command which is 128-bit RAM interface.

The user parameters are stored in Command parameter and forwarded to NVMe Host controller which is the main controller to create the command packet and decode the status packet. Data buffer is implemented by 256 KB BlockRAM to store the data from user logic. NVMe data controller is designed to create and decode data packet following NVMe protocol.

Otherwise, the NVMe IP implements PCIe transaction controller to create and decode PCIe TLP packet. AsyncCtrl is clock-crossing module to transfer packet between Clk domain (user clock) and PCIeClk domain which is output from PCIe PHY IP and fixed at 250 MHz for PCIe Gen3 speed.

The details of PCIe Soft IP which is implemented in NVMeG3 IP are described as follows.

- **PCIe Data Link Controller**

  PCIe Data Link Controller implements Data Link Layer of PCIe protocol. The function of the Data Link Layer is to ensure reliable delivery of TLPs which is the packet format transferring between PCIe Transation Controller and PCIe Data Link Controller. LCRC (Link Cyclic Redundancy Code) is added to each TLP for error checking at the receiver. Besides, the Sequence Number is appended to check the packet order. As a result, the receiver can sort the packet to be the same order as the sender. Furthermore, the receiver can detect the missing TLPs.

  After the receiver verifies LCRC and the Sequence Number, Ack DLLPs (Data Link Layer Packets) are generated to confirm good reception of TLPs. Nak DLLPs are created to indicate a transmission error. When Nak is received, the transmitter will re-send the TLPs to solve the problem.

  To re-send the TLPs, Replay buffer is included within Data Link Layer. This module is implemented by 2 Kbyte RAM. Otherwise, another 2 Kbyte RAM is included in the receiver to be the data buffer between Data Link Controller and Transaction Controller.

- **PCIe Media Access Controller (PCIe MAC)**

  PCIe MAC is designed to interface with PCIe PHY IP by PIPE. There are two purposes of this module. First is to run Link initialization and training process. Second is to control data packet by following PCIe physical specification.

  For Link initialization and training, some processes are implemented within PCIe PHY IP such as CDR for Bit lock and Block lock for Gen3 speed. LTSSM (Link Training and Status State Machine), implemented in PCIe MAC, is responsible to control Link width, Lane reversal, Polarity inversion, and Link data rate (Gen3 speed). As Gen3 operates at 8.0 GT/s which is more sensitive than Gen1 and Gen2, the additional features must be implemented in PCIe Gen3 MAC, i.e. DC balance and equalization.

  After finishing the initialization and training, data packet is transferred. To transmit the packet, PCIe MAC includes multiplexer for selecting the data types, byte striping for arranging data format in each lane, and data scrambler for reducing the noise. On the other hand, the receiver consists of the logic to run data de-scrambler, byte un-striping, and data filtering.

User Logic

User logic for running NVMe IP and NVMeG3 IP are similar. Figure 1 shows the example of user logic which is simply designed by using small state machine, two FIFOs for data transferring, and two RAMs for Identify command and custom command such as SMART command.

PCI Express PHY IP

This module is the Xilinx IP core to allow for a PCIe MAC to be built as Soft IP instead of Hard IP in some Xilinx FPGA devices. The user interface is PHY Interface for PCI Express (PIPE). To operate with NVMeG3 IP, PHY IP must configure Lane width to x4 and Link speed to 8.0 GT/s. More details and supported device list of IP are described in “PG239: PCI Express PHY” document.

https://www.xilinx.com/support/documentation/ip_documentation/pcie_phy/v1_0/pg239-pcie-phy.pdf

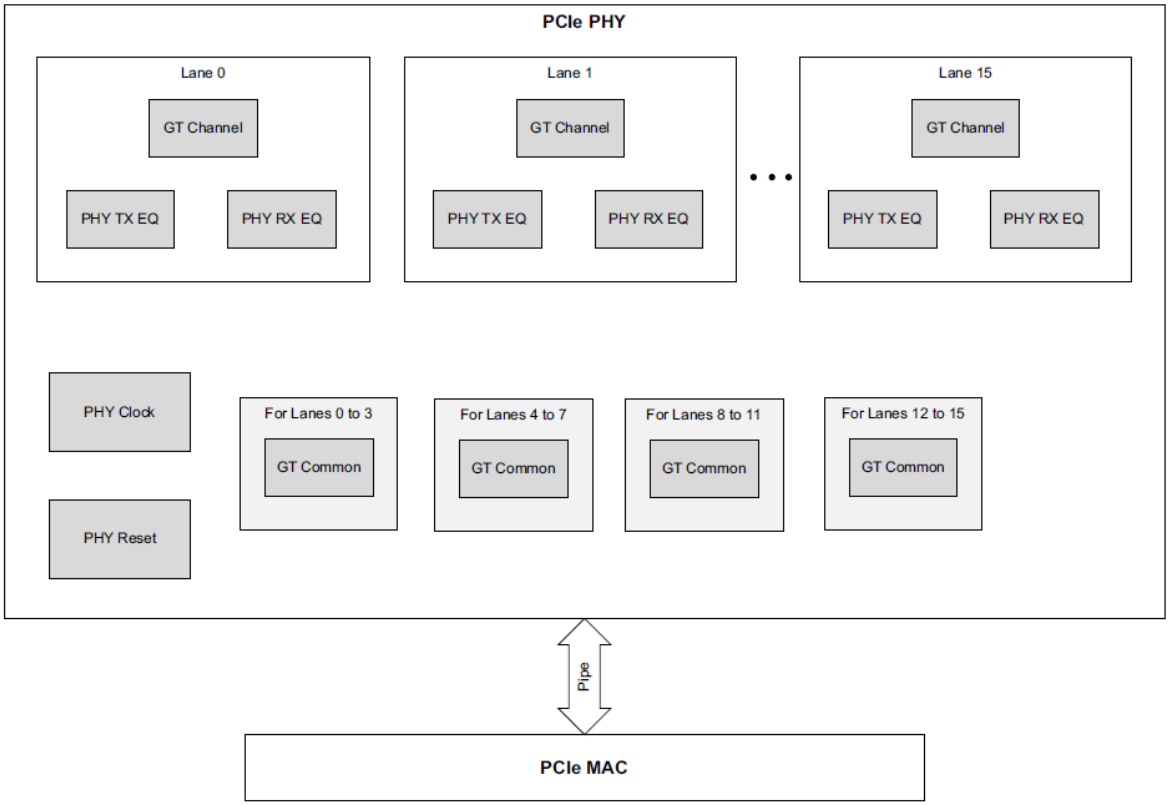

Figure 3: Block Diagram of the PCI Express PHY IP

Core I/O Signals

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

Table 3: Core I/O Signals

Signal	Dir	Description
Control I/F of dgIF typeS (Synchronous to Clk)		
RstB	In	Synchronous reset signal. Active low. Release to '1' when Clk signal is stable.
Clk	In	System clock for running NVMeG3 IP. The frequency must be more than or equal to PCIeClk which is output from PCIe PHY IP (250 MHz when operating at PCIe Gen3 speed).
UserCmd[2:0]	In	User Command ("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. When LBA unit = 4 Kbyte, UserAddr[2:0] must be always set to "000" to align 4 Kbyte. When LBA unit = 512 byte, it is recommended to set UserAddr[2:0]="000" to align 4 Kbyte (SSD page size). Write/Read performance of many 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). When LBA unit = 4 Kbyte, UserLen[2:0] must be always set to "000" to align 4 Kbyte.
UserReq	In	Asserted to '1' to send the new command request. This signal can be asserted when the IP is Idle (UserBusy='0'). Command parameter (UserCmd, UserAddr, UserLen, CtmSubmDW0-DW15) must be valid and stable when UserReq='1'. UserAddr and UserLen are the perameters for Write/Read command while CtmSubmDW0-DW15 are the parameters for SMART/Flush command.
UserBusy	Out	Asserted to '1' when IP is busy. New request must not be sent (UserReq to '1') when IP still be busy.
LBASize[47:0]	Out	Total capacity of SSD in 512-byte unit. This value is valid after finishing Identify command. So, Identify command must be run to update LBASize value before running other commands. The unit size is 512 byte. Default value after reset process is 0.
LBAMode	Out	LBA unit size of SSD. This signal is valid after finishing Identify command. So, Identify command must be run to update LBAMode value before running other commands. '0': LBA size = 512 byte, '1': LBA size = 4 Kbyte. Default value is 0.
UserError	Out	Error flag. Assert to '1' when UserErrorType is not equal to 0. The flag can be cleared 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 can be asserted by following problem. - 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. The value of each parameter can be read from NVMeCAPReg signal [2] – Error when Admin completion entry is not returned 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 returned until timeout.

Signal	Dir	Description
Control I/F of dgIF typeS (Synchronous to Clk)		
UserErrorType[31:0]	Out	[5] – Error when status register in IO completion entry is not 0 or phase tag is invalid. Please see more details from IOCompStatus signal. [6] – Error when Completion TLP packet size is not correct. [7] – Reserved [8] – Error from Unsupported Request (UR) flag in Completion TLP packet [9] – Error from Completer Abort (CA) flag in Completion TLP packet [10] – Error when Rx Buffer in Data Link controller is overflow. [15:11] – Reserved [16] - Error from unsupported LBA unit (LBA unit is not equal to 512 byte or 4 Kbyte) [31:17] – Reserved Note: Timeout period of bit[2]/[4] is set from TimeOutSet input.
Data I/F of dgIF typeS (Synchronous to Clk)		
UserFifoWrCnt[15:0]	In	Write data counter of Received FIFO. Used to check full status. When total FIFO size is less than 16 bit, please fill ‘1’ to upper bit.
UserFifoWrEn	Out	Asserted to ‘1’ to write data valid of Received FIFO when running Read command.
UserFifoWrData[127:0]	Out	Write data bus of Received FIFO. Valid at the same clock as UserFifoWrEn asserted to ’1’.
UserFifoRdCnt[15:0]	In	Read data counter of Transmit FIFO for checking total number of data stored in FIFO. When FIFO size signal 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 when running Write command.
UserFifoRdData[127:0]	In	Read data returned from Transmit FIFO. Valid in the next clock after UserFifoRdEn is asserted to ‘1’.

Signal	Dir	Description
Other User Interface (Synchronous to Clk)		
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 function is disabled.
AdmCompStatus[15:0]	Out	Status output from Admission 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] – MPSMIN (Memory Page Size Minimum) [31:25] – Undefined
IdenWrEn	Out	Asserted to '1' when IdenWrDWEn, IdenWrAddr, IdenWrData are valid on the bus.
IdenWrDWEn[3:0]	Out	Dword (32 bit) enable of IdenWrData. Valid at the same clock as IdenWrEn='1'. '1': this dword data is valid, '0': this dword data is not available. Bit[0]: IdenWrData[31:0], Bit[1]: IdenWrData[63:32], and so on.
IdenWrAddr[8:0]	Out	Index of IdenWrData in 128-bit unit. Valid at the same clock as IdenWrEn='1'. 0x000-0x0FF is 4Kbyte Identify controller data, 0x100-0x1FF is 4Kbyte Identify namespace data.
IdenWrData[127:0]	Out	4Kbyte Identify controller data or Identify namespace data. Valid at the same clock as IdenWrEn='1'.
Custom interface (Synchronous to Clk)		
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' when CtmRamWrDWEn, CtmRamAddr and CtmRamWrData are valid on the bus. This signal is asserted to'1' when running SMART command.
CtmRamWrDWEn[3:0]	Out	Dword (32 bit) enable of CtmRamWrData. Valid at the same clock as CtmRamWrEn='1'. '1': this dword data is valid, '0': this dword data is not available. Bit[0]: CtmRamWrData[31:0], Bit[1]: CtmRamWrData[63:32], and so on.
CtmRamAddr[8:0]	Out	Index of CtmRamWrData when the device returns SMART data. (Optional) Index to request data input through CtmRamRdData when running custom command.
CtmRamWrData[127:0]	Out	512-byte data output from SMART command.
CtmRamRdData[127:0]	In	(Optional) Data input when running custom command.

Signal	Dir	Description
PHY Clock and Reset		
PhyRstB	In	Synchronous reset signal. Active low. Release to '1' when PCIe PHY IP is not in reset state (monitored by phy_phystatus_rst signal is de-asserted to '0').
PhyClk	In	Clock output from PCIe PHY IP (250 MHz when operating at PCIe Gen3 speed).
Other PHY Interface (Synchronous to PCIeClk)		
MACTestPin[63:0]	Out	Test point of PCIe MAC.
MACStatus[7:0]	Out	Status output from PCIe MAC.
PIPE Data Interface of PCIe PHY IP (Synchronous to PCIeClk)		
PhyTxData[255:0]	Out	Parallel data output to PHY. *Note: When connecting to PHY IP on Ultrascale device, only 128-bit is used.* *Connect bit[31:0] to phy_txdata[31:0], bit[95:64] to phy_txdata[63:32],* *bit[159:128] to phy_txdata[95:64], and bit[223:192] to phy_txdata[127:96].*
PhyTxDataK[7:0]	Out	Control data to indicate whether PCIeTxData is control or data.
PhyTxDataValid[3:0]	Out	Asserted to '1' when the valid data is on PhyTxData.
PhyTxStartBlock[3:0]	Out	Asserted to '1' at the first clock of 128b block to indicate start of block. Valid when PhyTxDataValid is asserted to '1'.
PhyTxSyncHeader[7:0]	Out	Indicates whether data block is ordered set or data stream. Valid at the first clock of 128b block together with PhyTxStartBlock.
PhyRxData[255:0]	In	Data input from PHY. *Note: When connecting to PHY IP on Ultrascale device, only 128-bit is used.* *Connect bit[31:0] to phy_rxdata[31:0], bit[95:64] to phy_rxdata[63:32],* *bit[159:128] to phy_rxdata[95:64], and bit[223:192] to phy_rxdata[127:96].*
PhyRxDataK[7:0]	In	Control data to indicate whether PhyRxData is control or data.
PhyRxDataValid[3:0]	In	Asserted to '1' when the valid data is on PhyRxData.
PhyRxStartBlock[7:0]	In	Asserted to '1' at the first clock of 128b block to indicate start of block. Valid when PhyRxDataValid is asserted. *Note: When connecting to PHY IP on Ultrascale device, only 4-bit is used.* *Connect bit[0] to phy_rxstart_block[0], bit[2] to phy_rxstart_block[1],* *bit[4] to phy_rxstart_block[2], and bit[6] to phy_rxstart_block[3].*
PhyRxSyncHeader[7:0]	In	Indicates whether data block is ordered set or data stream. Valid at the first clock of 128b block together with PhyRxStartBlock.
PIPE Control and Status Signal of PCIe PHY IP (Synchronous to PCIeClk)		
PhyTxDetectRx	Out	Requests PHY IP to begin a receiver detection operation.
PhyTxElecIdle[3:0]	Out	Forces Tx to enter electrical idle state.
PhyTxCompliance[3:0]	Out	Asserted to '1' for running negative disparity.
PhyRxPolarity[3:0]	Out	Requests PHY IP to perform polarity inversion on the received data.
PhyPowerdown[1:0]	Out	Requests PHY IP to change the power state.
PhyRate[1:0]	Out	Requests PHY IP to change link rate.
PhyRxValid[3:0]	In	Indicates symbol lock and valid data when logic high.
PhyPhyStatus[3:0]	In	Used to communicate completion of several PIPE operations.
PhyRxElecIdle[3:0]	In	Indicates Rx electrical idle detected.
PhyRxStatus[11:0]	In	Rx status and error codes.

Signal	Dir	Description
Driver and Equalization Signal of PCIe PHY IP (Synchronous to PCIeClk)		
PhyTxMargin[2:0]	Out	Selects Tx voltage levels. This signal is fixed to 000b.
PhyTxSwing	Out	Conrols Tx voltage swing level. This signal is fixed to 0b.
PhyTxDeEmph	Out	Selects Tx de-emphasis. This signal is fixed to 1b.
PhyTxEqCtrl[7:0]	Out	Tx equalization control.
PhyTxEqPreset[15:0]	Out	Tx equalization preset.
PhyTxEqCoeff[23:0]	Out	Tx equalization coefficient.
PhyTxEqFS[5:0]	In	Indicates the full swing of the Tx driver. Static value based on characteristics of Tx driver.
PhyTxEqLF[5:0]	In	Indicates the low frequency of the Tx driver. Static value based on characteristics of Tx driver.
PhyTxEqNewCoeff[71:0]	In	Status of the current Tx equalization coefficient.
PhyTxEqDone[3:0]	In	Asserted to '1' when Tx equalization is done.
PhyRxEqCtrl[7:0]	Out	Rx equalization control.
PhyRxEqTxPreset[15:0]	Out	Link partner status for Tx preset.
PhyRxEqPresetSel[3:0]	In	Serves indications as Coefficient or preset.
PhyRxEqNewTxCoeff[71:0]	In	New Tx coefficient or preset to request the link partner.
PhyRxEqAdaptDone[3:0]	In	Asserted to '1' when RX equalization is successfully done. Valid when PhyRxEqDone is asserted to '1'.
PhyRxEqDone[3:0]	In	Asserted to '1' when Rx equalization is finished.
Assist signal of PCIe PHY IP (Synchronous to PCIeClk)		
AsMacInDetect	Out	Assists PHY IP to switch the receiver termination between VTT and GND.
AsCdrHoldReq	Out	Assists PHY IP to hold CDR.

Note: More details about PCIe PHY IP signal are described in "PG239: PCI Express PHY" document which can be downloaded from Xilinx website.

Timing Diagram

Initialization

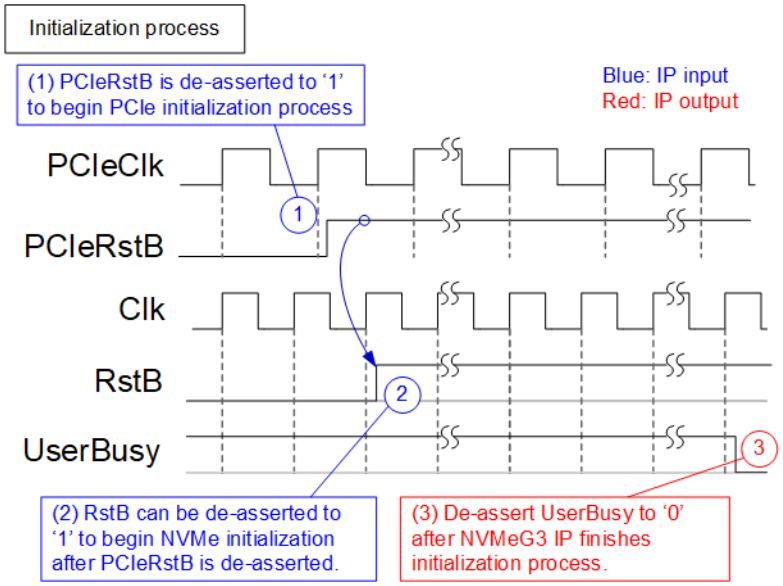

Figure 4: Timing diagram of the reset sequence

The sequences of the initialization process are as follows.

1) Wait PHY IP reset completion by monitoring that phy_phystatus_rst, signal output from PHY IP, is de-asserted to '0'. At the same time, PCIeRstB is changed to '1'. After that, the link training process within NVMeG3 IP begins.
2) De-assert RstB to '0' after PCIeRstB is de-asserted. NVMe logic within NVMeG3 IP starts the operation.
3) After IP initialization processes such as link training, flow control initialization, and PCIe register and NVMe register configuration complete, UserBusy is de-asserted to '0'.

After complete above sequences, NVMeG3-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.

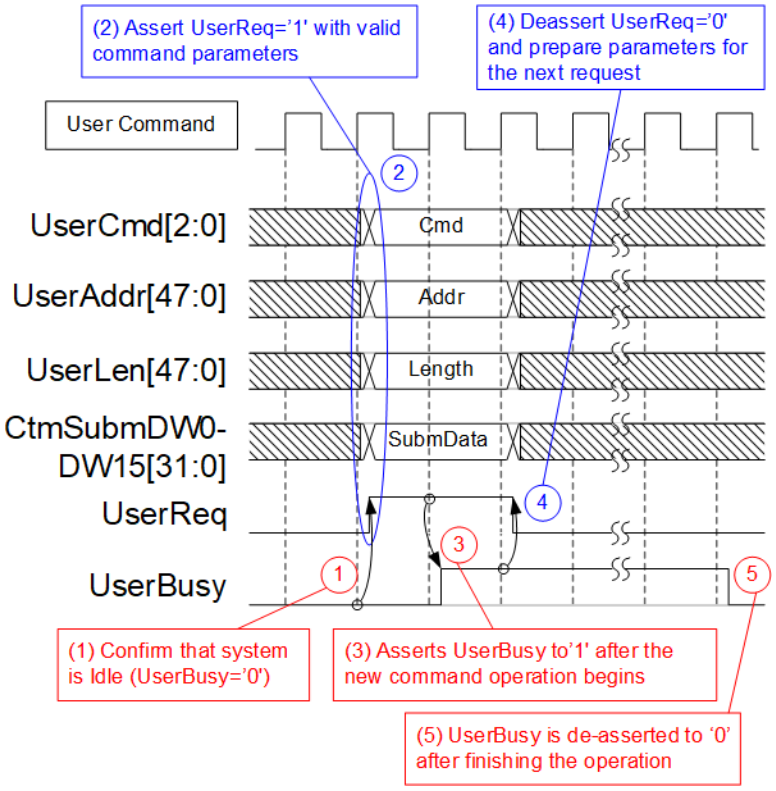

Figure 5: Control Interface of dgIF typeS Timing diagram

1) Before sending new command 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' when starting the new command operation.
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 could be sent by asserting UserReq to '1'.

Note: The number of parameters using in each command is different.

- Write and Read command: Use UserCmd, UserAddr, and UserLen.
- SMART and Flush command: Use UserCmd and CtmSubmDW0-DW15.
- Identify and Shudown command: Use only 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. 16 bit FIFO read data counter (UserFifoRdCnt) shows total data in the FIFO before transferring as a burst. The burst size is 512 byte or 32 cycles of 128 bit data.

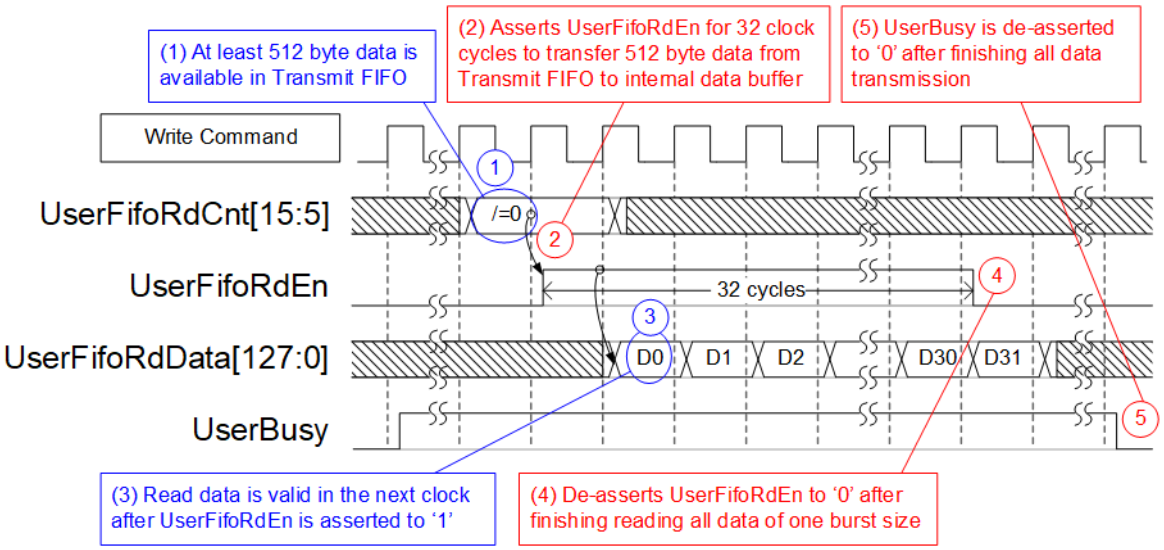

Figure 6: Transmit FIFO Interface for Write command

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

1) Before starting a new burst transfer, UserFifoRdCnt[15:5] is monitored. The IP waits until at least 512 byte data is available in Transmit FIFO (UserFifoRdCnt[15:5] is not equal to 0).
2) Asserts UserFifoRdEn to '1' for 32 clock cycles to read 512 byte data from Transmit FIFO.
3) UserFifoRdData is valid in the next clock cycle after asserting UserFifoRdEn to '1'. 32 data are read continuously.
4) UserFifoRdEn is deasserted to '0' after reading the 32th data. When the current burst transmission is not the last one, the IP goes back to step 1) for transferring the next 512 byte data.
5) After total data have already transferred, UserBusy is de-asserted to '0'. UserFifoRdEn is always de-asserted to '0' until the next Write command operates.

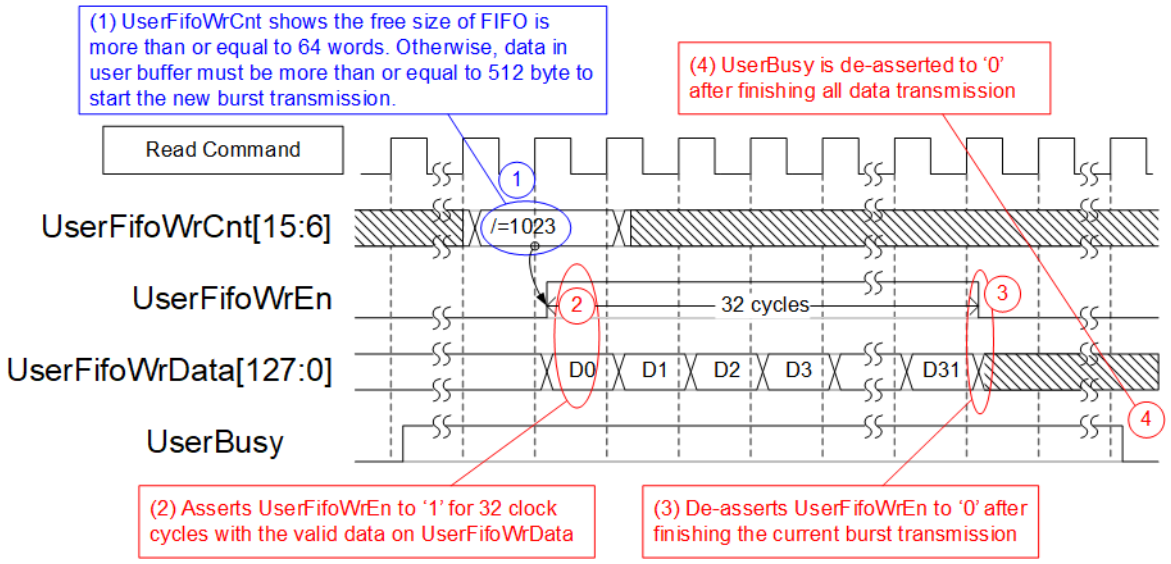

Figure 7: Received FIFO Interface for Read command

In Read command, data is sent from SSD to Received FIFO until total data are completely transmitted. The details to transfer one burst size data are as follows.

1) Before starting the new burst transmission, UserFifoWrCnt[15:5] is monitored. The IP waits until the free space of Received FIFO is much enough (UserFifoWrCnt[15:6] is not equal to all 1 or 1023). After data returned from the SSD is more than or equal to 512 byte, the new burst transmission begins.
2) Asserts UserFifoWrEn to '1' for 32 clock cycles to transfer 512 byte data from the internal buffer to user.
3) After transferring 512 byte data, UserFifoWrEn is de-asserted to '0'. When the current burst transmission is not the last one, the IP goes back to step 1) for transferring the next 512 byte data.
4) After total data have already transferred, UserBusy is de-asserted to '0'. UserFifoWrEn is always de-asserted to '0' until the next Read command operates.

The timing diagrams of user interface when running other commands are same as the timing diagram described in NVMe IP datasheet. Please see more details from NVMe IP datasheet which can be downloaded from our website.

Verification Methods

The NVMeG3-IP Core functionality was verified by simulation and also proved on real board design by using KCU105/ZCU102/VCU118 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	Aug-29-2019	Initial Release