NVMe IP with PCIe Gen3 Soft IP reference design manual

Rev1.0 30-Aug-19

1 NVMe

NVM Express (NVMe) defines the interface for the host controller to access solid state drive (SSD) by PCI Express. NVM Express optimizes the process to issue command and completion by using only two registers (Command issue and Command completion). Otherwise, NVMe supports parallel operation by supporting up to 64K commands within single queue. 64K command entries improve transfer performance for both sequential and random access.

In PCIe SSD market, two standards are used, i.e. AHCI and NVMe. AHCI is the older standard to provide the interface for SATA hard disk drive while NVMe is optimized for non volatile memory (SSD). The comparison between both AHCI and NVMe protocol in more details is described in "A Comparison of NVMe and AHCI" document.
https://sata-io.org/system/files/member-downloads/NVMe%20and%20AHCI_%20_long_.pdf

The example of NVMe storage device is shown in http://www.nvmexpress.org/products/.

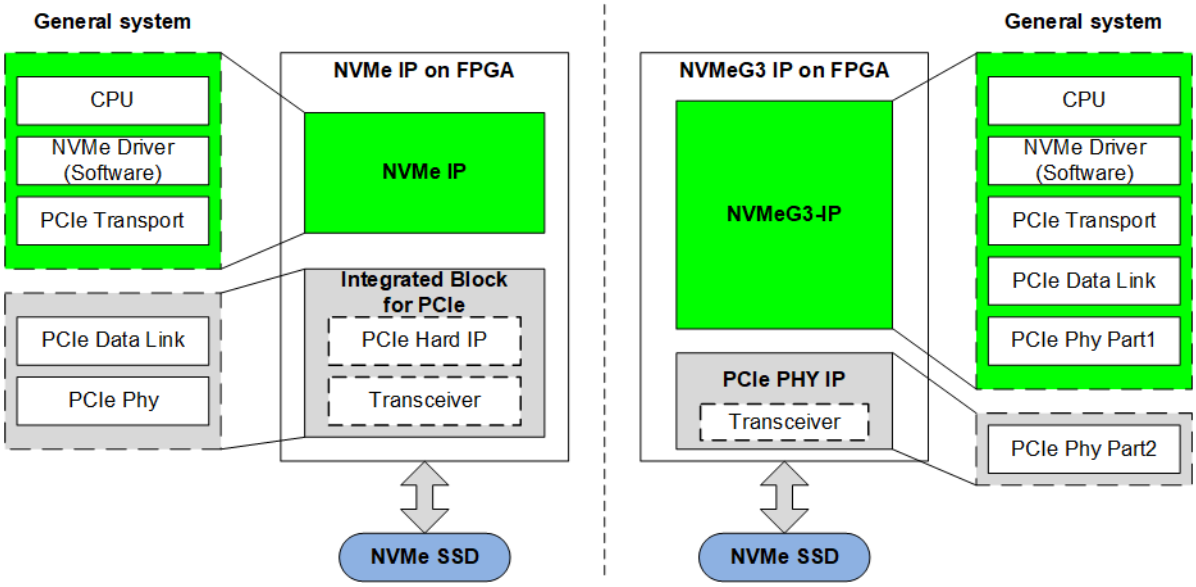

Figure 1-1 NVMe implementation

Conventionally, the NVMe host is implemented by using PC or CPU operating with PCIe controller for transferring data with NVMe SSD. NVMe protocol is designed as the driver and communicates with the PCIe controller hardware which is CPU peripherals connected through very high-speed bus. External memory is applied for transferring data between PCIe controller and SSD.

The new solution is purposed by using DG NVMe IP, as shown in Figure 1-1. Without CPU and external main memory usage, the NVMe host can be implemented within FPGA by using NVMe IP and Integrated Block for PCIe (PCIe hard IP). This solution uses less FPGA resource and achieves ultra-speed write and read performance. The limitation of this solution is the availability of PCIe hard IP which is included only some FPGA models. Otherwise, the maximum number of SSDs are limited by the number of PCIe hard IPs.

This document presents the latest solution from Design Gateway. The NVMe host IP can be implemented in the low-cost FPGA or no PCIe hard IP FPGA by using DG NVMeG3 IP. The new IP implements the lower layer of PCIe protocol, i.e. Data Link Layer and some parts of Physical Layer by using the logic. This feature is known as PCIe soft IP. By integrating PCIe Soft IP and NVMe IP, NVMe G3 IP connecting with Xilinx PCIe PHY IP is the recent solution for implementing NVMe host in FPGA.

User interface and performance of DG NVMe IP and DG NVMeG3 IP are similar. The user can use the same user logic for running NVMe IP or NVMeG3 IP.

2 Hardware overview

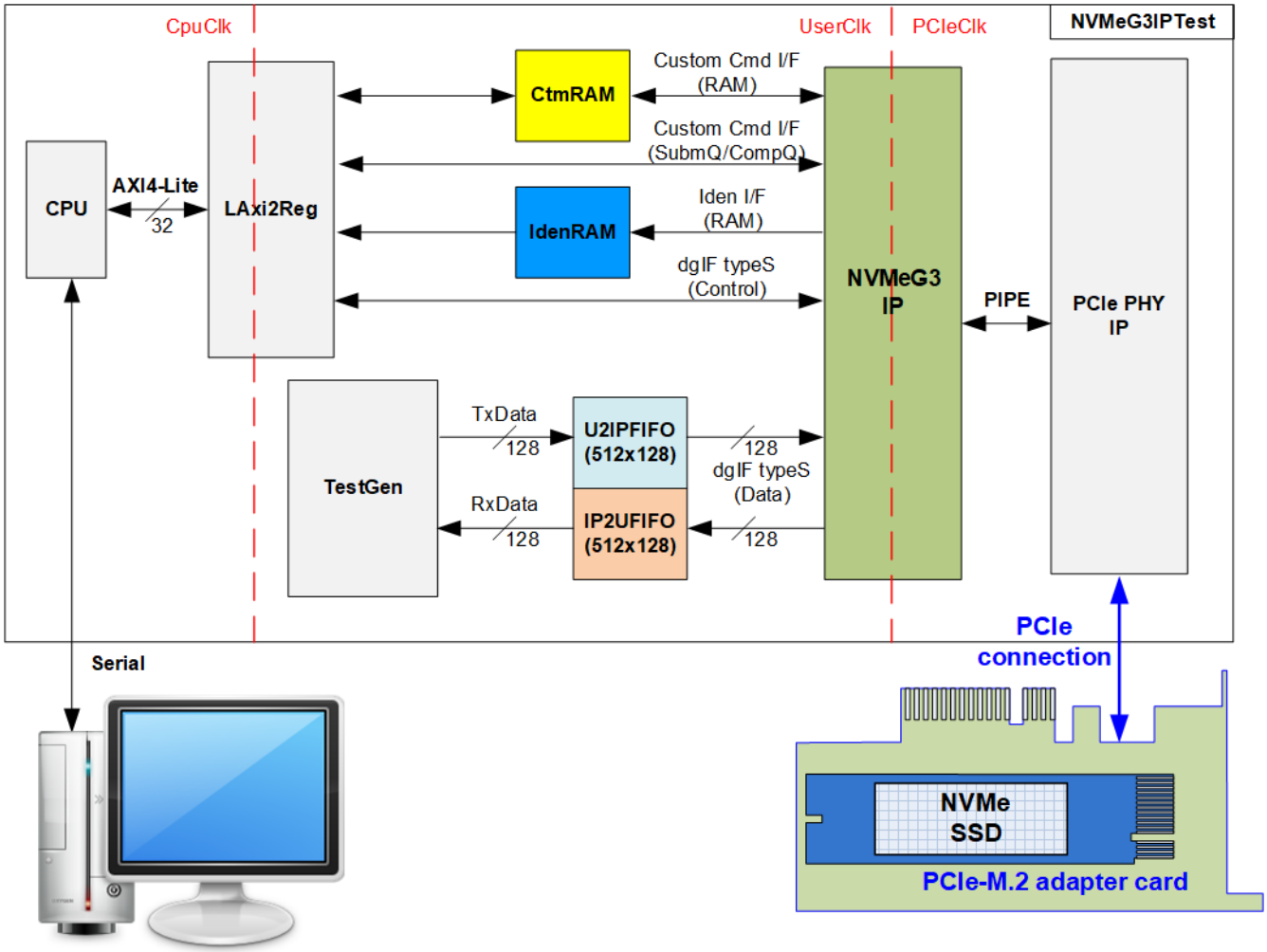

Figure 2-1 NVMeG3 IP demo hardware

Since user interface of NVMeG3 IP and NVMe IP are similar, the modules for connecting user interface such as TestGen and CPU system of NVMeG3IPTest are designed by using the same modules in NVMe IP reference design. The different module is Xilinx PCIe IP. NVMeG3 IP connects with PCIe PHY IP through PIPE instead of PCI hard IP, as shown in Figure 2-1.

This document describes only the modification point to run NVMeG3 IP reference design, based on NVMe IP reference design. The details of the user interface are described in NVMe IP reference design document which are provided on our website.
https://dgway.com/products/IP/NVMe-IP/dg_nvmeip_refdesign_en.pdf

NVMe IP reference design document shows the details of the logic design with timing diagram of TestGen, LAxi2Reg, RAM, and FIFO. All modules have same behavior as the design in NVMeG3IPTest module. However, there is the additional test pin of MAC layer which is output from NVMeG3 IP, named MACTestPin. This signal is necessary for system debugging when the problem is found from PCIe initialization process. So, UserReg within LAxi2Reg is slightly modified to read this signal by CPU. The signal is mapped to BA+0x0124 and BA+0x0128 as shown in Table 2-1.

Table 2-1 Register Map

Address Rd/Wr	Register Name (Label in the “nvmeg3iptest.c”)	Description
0x0000 – 0x00FF: Control signals of NVMeG3 IP and TestGen (Write access only)		
BA+0x0000	User Address (Low) Reg (USRADRL_REG)	[31:0]: Input to be start address as 512-byte unit (UserAddr[31:0] of dgIF typeS)
BA+0x0004	User Address (High) Reg (USRADRH_REG)	[15:0]: Input to be start address as 512-byte unit (UserAddr[47:32] of dgIF typeS)
BA+0x0008	User Length (Low) Reg (USRLENL_REG)	[31:0]: Input to be transfer length as 512-byte unit (UserLen[31:0] of dgIF typeS)
BA+0x000C	User Length (High) Reg (USRLENH_REG)	[15:0]: Input to be transfer length as 512-byte unit (UserLen[47:32] of dgIF typeS)
BA+0x0010	User Command Reg (USRCMD_REG)	[2:0]: Input to be user command (UserCmd of dgIF typeS for NVMeG3 IP) “000”: Identify, “001”: Shutdown, “010”: Write SSD, “011”: Read SSD, “100”: SMART, “110”: Flush, “101”/”111”: Reserved When this register is written, the command request is sent to NVMeG3 IP to start the operation.
BA+0x0014	Test Pattern Reg (PATTSEL_REG)	[2:0]: Select test pattern “000”-Increment, “001”-Decrement, “010”-All 0, “011”-All 1, “100”-LFSR
BA+0x0020	NVMe Timeout Reg (NVMTIMEOUT_REG)	[31:0]: Timeout value of NVMeG3 IP (TimeOutSet[31:0] of NVMeG3 IP)
0x0100 – 0x01FF: Status signals of NVMeG3 IP and TestGen (Read access only)		
BA+0x0100	User Status Reg (USRSTS_REG)	[0]: UserBusy of dgIF typeS (‘0’: Idle, ‘1’: Busy) [1]: UserError of dgIF typeS (‘0’: Normal, ‘1’: Error) [2]: Data verification fail (‘0’: Normal, ‘1’: Error)
BA+0x0104	Total disk size (Low) Reg (LBASIZEL_REG)	[31:0]: LBASize0[31:0] output from NVMeG3 IP
BA+0x0108	Total disk size (High) Reg (LBASIZEH_REG)	[15:0]: LBASize0[47:32] output from NVMeG3 IP [31]: LBAMode output from NVMeG3 IP
BA+0x010C	User Error Type Reg (USRERRTYPE_REG)	[31:0]: User error status (UserErrorType[31:0] of dgIF typeS)
BA+0x0110	PCIe Status Reg (PCISTS_REG)	[7:0]: Unused for NVMeG3 IP [15:8]: MACStatus output from NVMeG3 IP
BA+0x0114	Completion Status Reg (COMPSTS_REG)	[15:0]: Status from Admin completion (AdmCompStatus[15:0] of NVMeG3 IP) [31:16]: Status from I/O completion (IOCompStatus[15:0] of NVMeG3 IP)
BA+0x0118	NVMe CAP Reg (NVMCAP_REG)	[31:0]: NVMeCAPReg[31:0] output from NVMeG3 IP
BA+0x0120	NVMe Test pin Reg (NVMTESTPIN_REG)	[31:0]: TestPin[31:0] output from NVMeG3 IP
BA+0x0124	MAC Test pin (Low) Reg (MACTESTPINL_REG)	[31:0]: MACTestPin[31:0] output from NVMeG3 IP
BA+0x0128	MAC Test pin (High) Reg (MACTESTPINH_REG)	[31:0]: MACTestPin[63:0] output from NVMeG3 IP

Address Rd/Wr	Register Name (Label in the “nvmeipg3test.c”)	Description
0x0100 – 0x01FF: Status signals of NVMeG3 IP and TestGen (Read access only)		
BA+0x0130	Expected value Word0 Reg (EXPPATW0_REG)	[31:0]: Bit[31:0] of the expected data at the 1st failure data in Read command
BA+0x0134	Expected value Word1 Reg (EXPPATW1_REG)	[31:0]: Bit[63:32] of the expected data at the 1st failure data in Read command
BA+0x0138	Expected value Word2 Reg (EXPPATW2_REG)	[31:0]: Bit[95:64] of the expected data at the 1st failure data in Read command
BA+0x013C	Expected value Word3 Reg (EXPPATW3_REG)	[31:0]: Bit[127:96] of the expected data at the 1st failure data in Read command
BA+0x0140	Read value Word0 Reg (RDPATW0_REG)	[31:0]: Bit[31:0] of the read data at the 1st failure data in Read command
BA+0x0144	Read value Word1 Reg (RDPATW1_REG)	[31:0]: Bit[63:32] of the read data at the 1st failure data in Read command
BA+0x0148	Read value Word2 Reg (RDPATW2_REG)	[31:0]: Bit[95:64] of the read data at the 1st failure data in Read command
BA+0x014C	Read value Word3 Reg (RDPATW3_REG)	[31:0]: Bit[127:96] of the read data at the 1st failure data in Read command
BA+0x0150	Data Failure Address(Low) Reg (RDFAILNOL_REG)	[31:0]: Bit[31:0] of the byte address of the 1st failure data in Read command
BA+0x0154	Data Failure Address(High) Reg (RDFAILNOH_REG)	[24:0]: Bit[56:32] of the byte address of the 1st failure data in Read command
BA+0x0158	Current test byte (Low) Reg (CURTESTSIZEL_REG)	[31:0]: Bit[31:0] of the current test data size in TestGen module
BA+0x015C	Current test byte (High) Reg (CURTESTSIZEH_REG)	[24:0]: Bit[56:32] of the current test data size of TestGen module
Other interfaces (Custom command of NVMeG3 IP, IdenRAM, and Custom RAM)		
BA+0x0200 - 0x023F Wr	Custom Submission Queue Reg (CTMSUBMQ_REG)	[31:0]: Submission queue entry of SMART and Flush command. Input to be CtmSubmDW0-DW15 of NVMeG3 IP. 0x200: DW0, 0x204: DW1, …, 0x23C: DW15
BA+0x0300 - 0x030F Rd	Custom Completion Queue Reg (CTMCOMPQ_REG)	[31:0]: CtmCompDW0-DW3 output from NVMeG3 IP. 0x300: DW0, 0x304: DW1, …, 0x30C: DW3
BA+0x0800 Rd	IP Version Reg (IPVERSION_REG)	[31:0]: IP version number (IPVersion[31:0] of NVMeG3 IP)
BA+0x2000 – 0x2FFF Rd	Identify Controller Data (IDENCTRL_REG)	4Kbyte Identify Controller Data Structure
BA+0x3000 – 0x3FFF Rd	Identify Namespace Data (IDENNAME_REG)	4Kbyte Identify Namespace Data Structure
BA+0x4000 – 0x5FFF Wr/Rd	Custom command Ram (CTMRAM_REG)	Connect to 8K byte CtmRAM interface. Used to store 512-byte data output from SMART Command.

3 CPU Firmware

CPU Firmware on NVMeG3 IP reference design is slightly modified from NVMe IP reference design in PCIe initialization sequence. The step to check PCIe link up signal is removed in NVMeG3 IP. After reset sequence is finished, the IP starts the initialization sequence. To initialize the system, CPU runs the following step.

1) CPU initializes UART and Timer parameters.
2) CPU waits until IP completes PCIe and NVMe initialization process by monitoring IP busy flag (USRSTS_REG[0]=’0’). When some errors are found, the process stops and displays the error message.
3) CPU displays the main menu. There are six menus for running six commands, i.e. Identify, Write, Read, SMART, Flush, and Shutdown.

The details of CPU firmware to operate all commands are similar to the sequence described in NVMe IP reference design. Please see more details from NVMe IP reference design document.

4 Example Test Result

The example test result when running demo system by using 512 GB Samsung 970 Pro is shown in Figure 4-1.

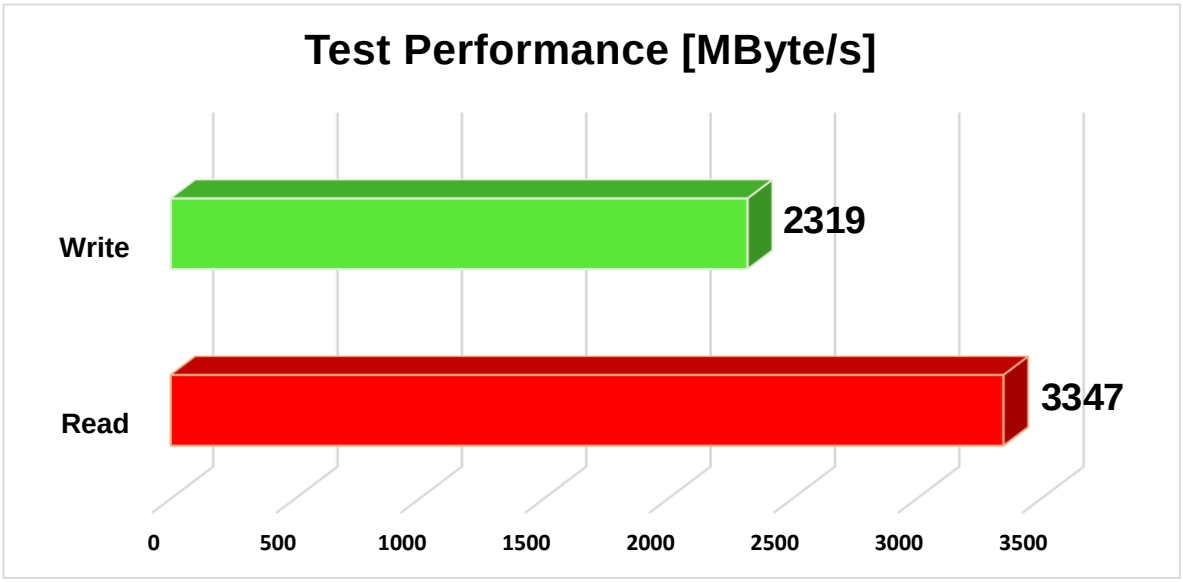

Figure 4-1 Test Performance of NVMeG3 IP demo by using Samsung 970 Pro SSD

By using PCIe Gen3 on ZCU102 board, write performance is about 2300 Mbyte/sec and read performance is about 3350 Mbyte/sec.

5 Revision History

Revision	Date	Description
1.0	30-Aug-19	Initial Release