NVMe IP with PCIe Gen3 Soft IP reference design manual

Rev1.2 14-Mar-22

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). Besides, 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/.

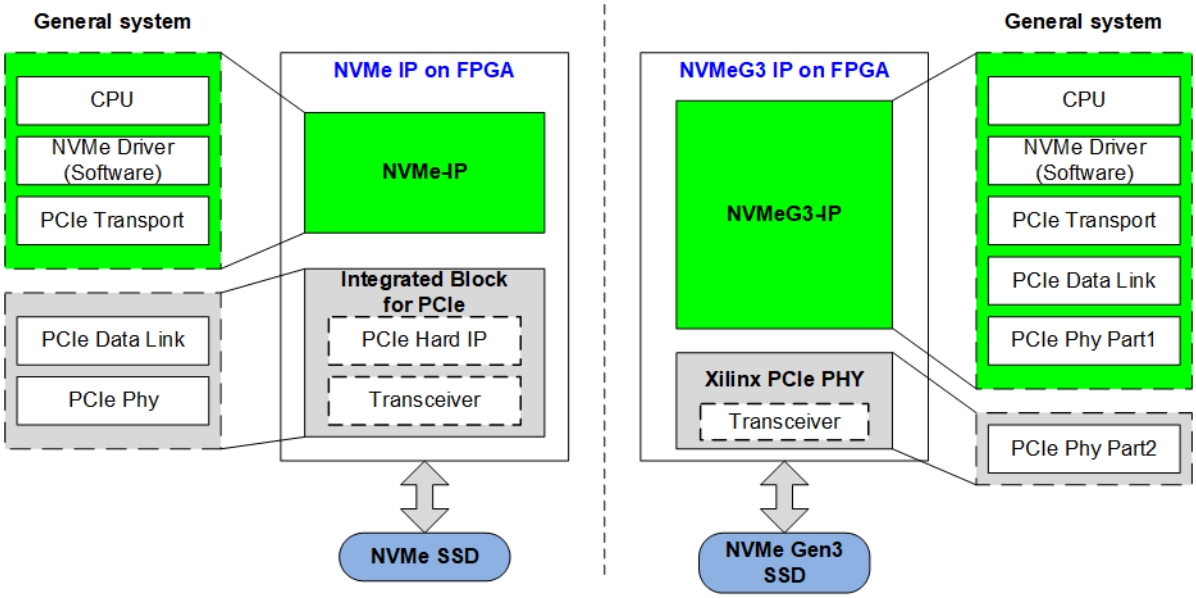

Figure 1-1 NVMe implementation

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. Besides, the maximum number of SSDs are limited by the number of PCIe hard IPs.

This document presents the solution from Design Gateway to access NVMe SSD by FPGA which does not include PCIe hard IP by using DG NVMeG3-IP. NVMeG3-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. NVMeG3-IP integrates NVMe-IP and PCIe soft IP logic and completes the host controller solution by connecting with Xilinx PCIe PHY for the physical interface with NVMe Gen3 SSD.

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. Therefore, this document describes the modification part from the NVMe-IP reference design to use NVMeG3-IP instead of NVMe-IP. NVMe-IP reference design can be downloaded from following link.
https://dgway.com/products/IP/NVMe-IP/dg_nvmeip_refdesign_en.pdf

2 Hardware overview

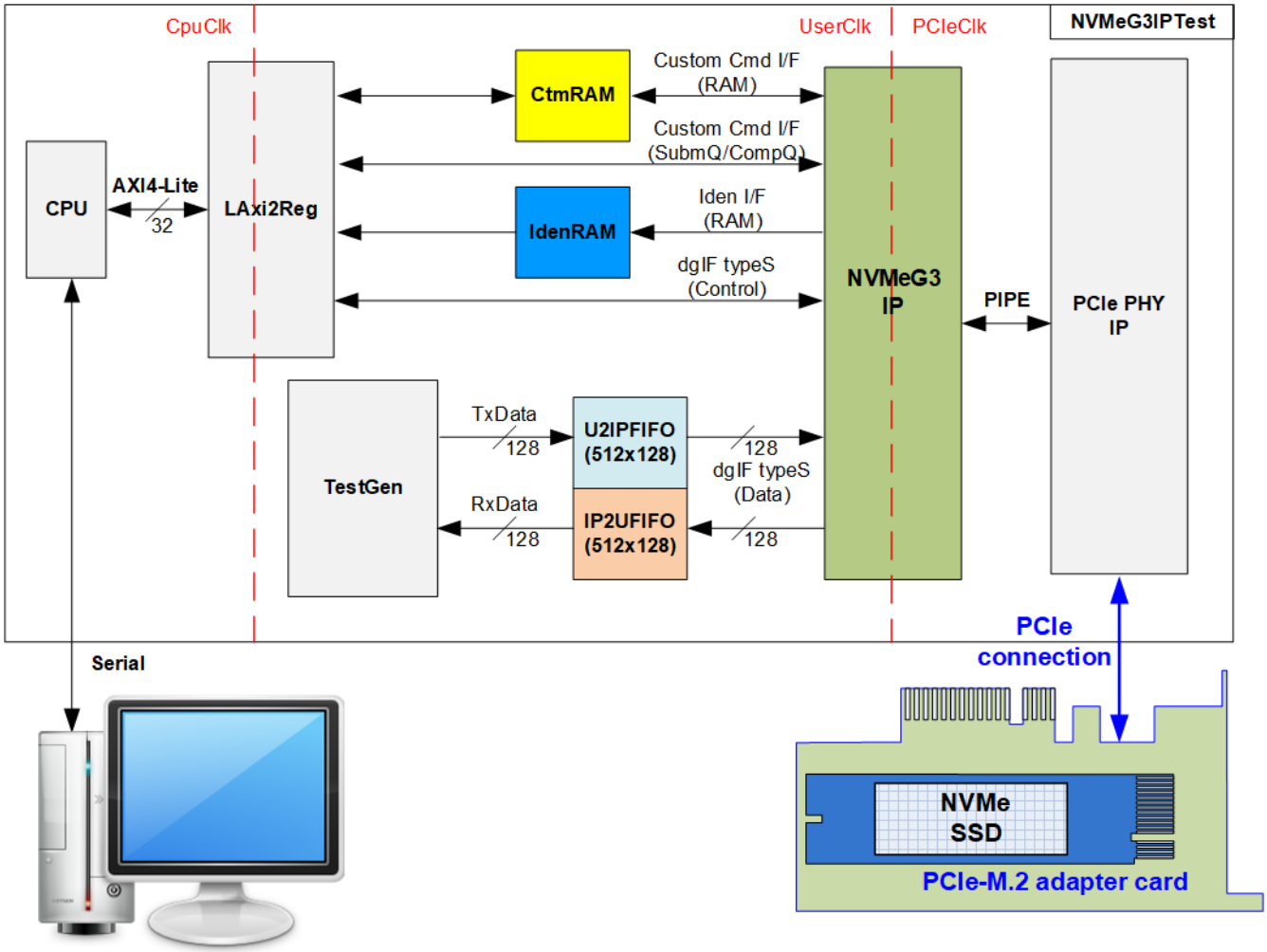

Figure 2-1 NVMeG3-IP demo hardware

User interface of NVMeG3-IP and NVMe-IP are similar, so 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 the low-level interface. NVMeG3-IP uses Xilinx PCIe PHY IP instead of PCI hard IP, as shown in Figure 2-1.

The details of the logic design and timing diagram of TestGen, LAxi2Reg, RAM, and FIFO are described in NVMe-IP reference design document. However, there is the additional test pin of MAC layer which is output from NVMeG3-IP, named MACTestPin. This signal is applied for system debugging when the problem is found from PCIe initialization process. Therefore, UserReg within LAxi2Reg is slightly modified to read this signal by CPU. The signal is mapped to BA+0x0120 and BA+0x0124 as shown in Table 2-1.

Table 2-1 Register Map

Address Rd/Wr	Register Name (Label in the “nvmeiptest.c”)	Description
0x0000 – 0x00FF: Control signals of NVMeG3-IP and TestGen (Write access only)		
BA+0x0000	User Address (Low) Reg (USRADRL_INTREG)	[31:0]: Input to be bit[31:0] of start address as 512-byte unit (UserAddr[31:0] of dgIF typeS)
BA+0x0004	User Address (High) Reg (USRADRH_INTREG)	[15:0]: Input to be bit[47:32] of start address as 512-byte unit (UserAddr[47:32] of dgIF typeS)
BA+0x0008	User Length (Low) Reg (USRLENL_INTREG)	[31:0]: Input to be bit[31:0] of transfer length as 512-byte unit (UserLen[31:0] of dgIF typeS)
BA+0x000C	User Length (High) Reg (USRLENH_INTREG)	[15:0]: Input to be bit[47:32] of transfer length as 512-byte unit (UserLen[47:32] of dgIF typeS)
BA+0x0010	User Command Reg (USRCMD_INTREG)	[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. After that, the IP starts operating the command.
BA+0x0014	Test Pattern Reg (PATTSEL_INTREG)	[2:0]: Select test pattern “000”-Increment, “001”-Decrement, “010”-All 0, “011”-All 1, “100”-LFSR
BA+0x0020	NVMe Timeout Reg (NVMTIMEOUT_INTREG)	[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_INTREG)	[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_INTREG)	[31:0]: Mapped to LBASize[31:0] of NVMeG3-IP
BA+0x0108	Total disk size (High) Reg (LBASIZEH_INTREG)	[15:0]: Mapped to LBASize[47:32] of NVMeG3-IP [31]: Mapped to LBAMode of NVMeG3-IP
BA+0x010C	User Error Type Reg (USRERRTYPE_INTREG)	[31:0]: User error status (UserErrorType[31:0] of dgIF typeS)
BA+0x0110	PCIe Status Reg (PCIESTS_INTREG)	[7:0]: Unused for NVMeG3-IP [15:8]: Mapped to MACStatus[7:0] of NVMeG3-IP
BA+0x0114	Completion Status Reg (COMPSTS_INTREG)	[15:0]: Mapped to AdmCompStatus[15:0] of NVMeG3-IP [31:16]: Mapped to IOCompStatus[15:0] of NVMeG3-IP
BA+0x0118	NVMe CAP Reg (NVMCAP_INTREG)	[31:0]: Mapped to NVMeCAPReg[31:0] of NVMeG3-IP
BA+0x011C	NVMe Test pin Reg (NVMTESTPIN_INTREG)	[31:0]: Mapped to TestPin[31:0] of NVMeG3-IP
BA+0x0120	MAC Test pin (Low) Reg (MACTESTPINL_INTREG)	[31:0]: Mapped to MACTestPin[31:0] of NVMeG3-IP
BA+0x0124	MAC Test pin (High) Reg (MACTESTPINH_INTREG)	[31:0]: Mapped to MACTestPin[63:32] of NVMeG3-IP

Address Rd/Wr	Register Name (Label in the “nvmeiptest.c”)	Description
0x0100 – 0x01FF: Status signals of NVMeG3-IP and TestGen (Read access only)		
BA+0x0130 – BA+0x013C	Expected value Word0-3 Reg (EXPPATW0-W3_INTREG)	128-bit of the expected data at the 1st failure data in Read command 0x0130: Bit[31:0], 0x0134[31:0]: Bit[63:32], …, 0x013C[31:0]: Bit[127:96]
BA+0x0150 – BA+0x016C	Read value Word0-3 Reg (RDPATW0-W3_INTREG)	128-bit of the read data at the 1st failure data in Read command 0x0150: Bit[31:0], 0x0154[31:0]: Bit[63:32], …, 0x015C[31:0]: Bit[127:96]
BA+0x0170	Data Failure Address(Low) Reg (RDFAILNOL_INTREG)	[31:0]: Bit[31:0] of the byte address of the 1st failure data in Read command
BA+0x0174	Data Failure Address(High) Reg (RDFAILNOH_INTREG)	[24:0]: Bit[56:32] of the byte address of the 1st failure data in Read command
BA+0x0178	Current test byte (Low) Reg (CURTESTSIZEL_INTREG)	[31:0]: Bit[31:0] of the current test data size in TestGen module
BA+0x017C	Current test byte (High) Reg (CURTESTSIZEH_INTREG)	[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 – BA+0x023F Wr	Custom Submission Queue Reg (CTMSUBMQ_STRUCT)	[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 – BA+0x030F Rd	Custom Completion Queue Reg (CTMCOMPQ_STRUCT)	[31:0]: CtmCompDW0-DW3 output from NVMeG3-IP. 0x300: DW0, 0x304: DW1, …, 0x30C: DW3
BA+0x0800 Rd	IP Version Reg (IPVERSION_INTREG)	[31:0]: Mapped to IPVersion[31:0] of NVMeG3-IP
BA+0x2000 – BA+0x2FFF Rd	Identify Controller Data (IDENCTRL_CHARREG)	4Kbyte Identify Controller Data Structure
BA+0x3000 – BA+0x3FFF Rd	Identify Namespace Data (IDENNAME_CHARREG)	4Kbyte Identify Namespace Data Structure
BA+0x4000 – BA+0x5FFF Wr/Rd	Custom command RAM (CTMRAM_CHARREG)	Connect to 8K byte CtmRAM interface. Used to store 512-byte data output from SMART Command.

3 CPU Firmware

Comparing to NVMe-IP reference design, CPU Firmware is modified in PCIe initialization sequence. The steps to check PCIe link up status, a number of PCIe lanes, PCIe speed are removed in NVMeG3-IP.

3.1 Test firmware (nvmeiptest.c)

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 NVMeG3-IP completes initialization process (USRSTS_INTREG[0]='0'). If some errors are found, the process stops with displaying 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 NVMe-IP reference design.

3.2 Function list in Test firmware

The function for running NVMeG3-IP is similar to NVMe-IP reference design. Only one function is modified to read the details of test pin for debugging the problem, described as follows.

void show_pciestat(void)	
Parameters	None
Return value	None
Description	Read PCIESTS_INTREG until the read value from two read times is stable. After that, display the read value on the console. Also, debug signals (NVMTESTPIN_INTREG and MACTESTPINL/H_INTREG) are read and displayed on the console.

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.

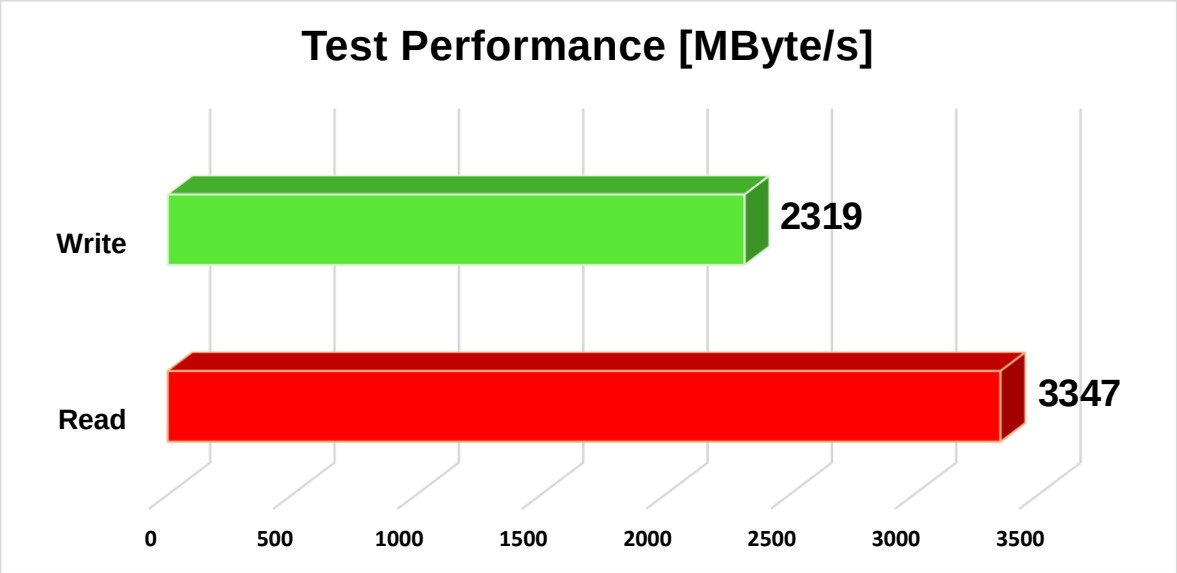

Figure 4-1 Test Performance of NVMeG3 IP demo by using Samsung 970 Pro SSD

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
1.1	4-Jun-21	Update register map
1.2	14-Mar-22	Update register name in the register map

Copyright: 2019 Design Gateway Co,Ltd.