



NVMe IP with PCIe Gen3 Soft IP reference design manual

Rev1.04 23-Feb-24

1	NVMe.....	2
2	Hardware overview.....	4
3	CPU Firmware.....	7
3.1	Test firmware (nvmeiptest.c).....	7
3.2	Function list in Test firmware.....	7
4	Example Test Result.....	8
5	Revision History.....	9

1 NVMe

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

In PCIe SSD market, two standards are commonly used: AHCI and NVMe. AHCI is the legacy standard designed for SATA hard disk drives, while NVMe, a newer standard, is specifically optimized for non-volatile memory (SSD). The detailed comparison between AHCI and NVMe protocol can be found in “A Comparison of NVMe and AHCI” document.

[https://sata-io.org/system/files/member-downloads/NVMe%20and%20AHCI_%20_long_.pdf](https://sata-io.org/system/files/member-downloads/NVMe%20and%20AHCI_%20long_.pdf)

The example of NVMe storage device can be found at <http://www.nvmeexpress.org/products/>.

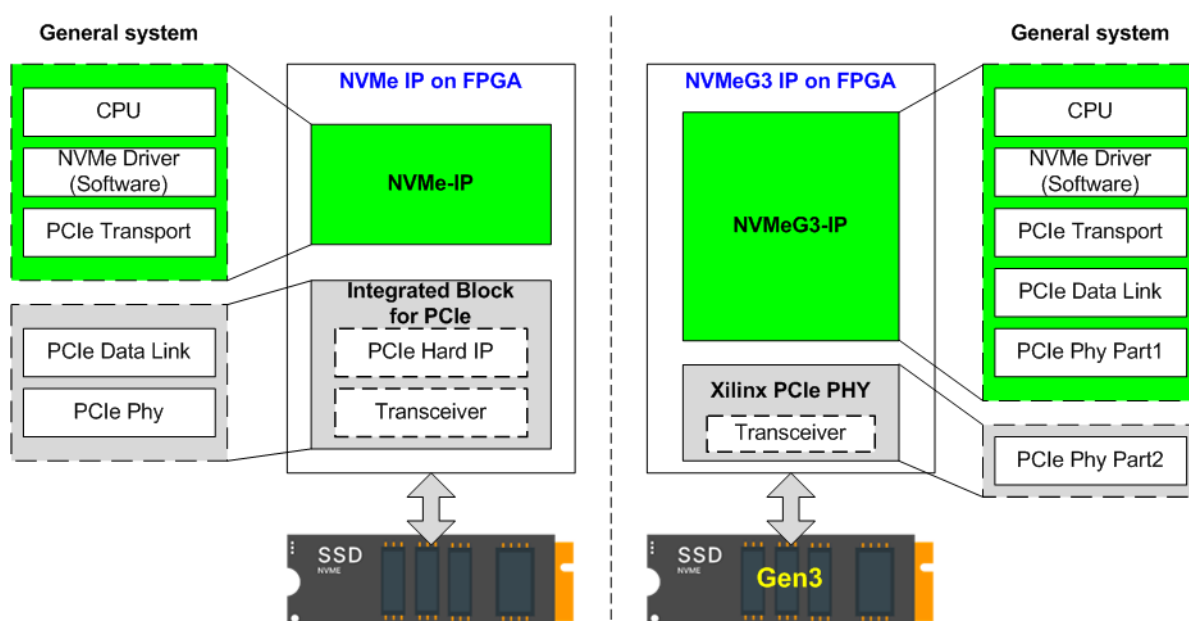


Figure 1-1 NVMe implementation

Conventionally, NVMe hosts are implemented by using PC or CPU operating with a PCIe controller for transferring data with NVMe SSD. The NVMe protocol is designed using the driver, communicating with the PCIe controller hardware, which connects to the CPU peripherals through a high-speed bus. External memory is applied for data transfer between the PCIe controller and the SSD.

A new solution is offered 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 high speed write and read performance. The limitation of this solution is the availability of PCIe hard IP which is included only specific FPGA models. Besides, the maximum number of SSDs are limited by the number of PCIe hard IPs available.

This document presents the solution from Design Gateway to access NVMe SSD using FPGA that does not have PCIe hard IP, using DG NVMeG3-IP. NVMeG3-IP implements the lower layer of the PCIe protocol, including Data Link Layer and a part of Physical Layer using logic. This feature is known as PCIe soft IP. NVMeG3-IP integrates NVMe-IP and PCIe soft IP logic, providing a complete host controller solution by connecting to 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 focuses on the modifications required to transition from the NVMe-IP reference design to use NVMeG3-IP instead. The NVMe-IP reference design can be downloaded from following link.

https://dgway.com/products/IP/NVMe-IP/dg_nvmeip_refdesign_en/

2 Hardware overview

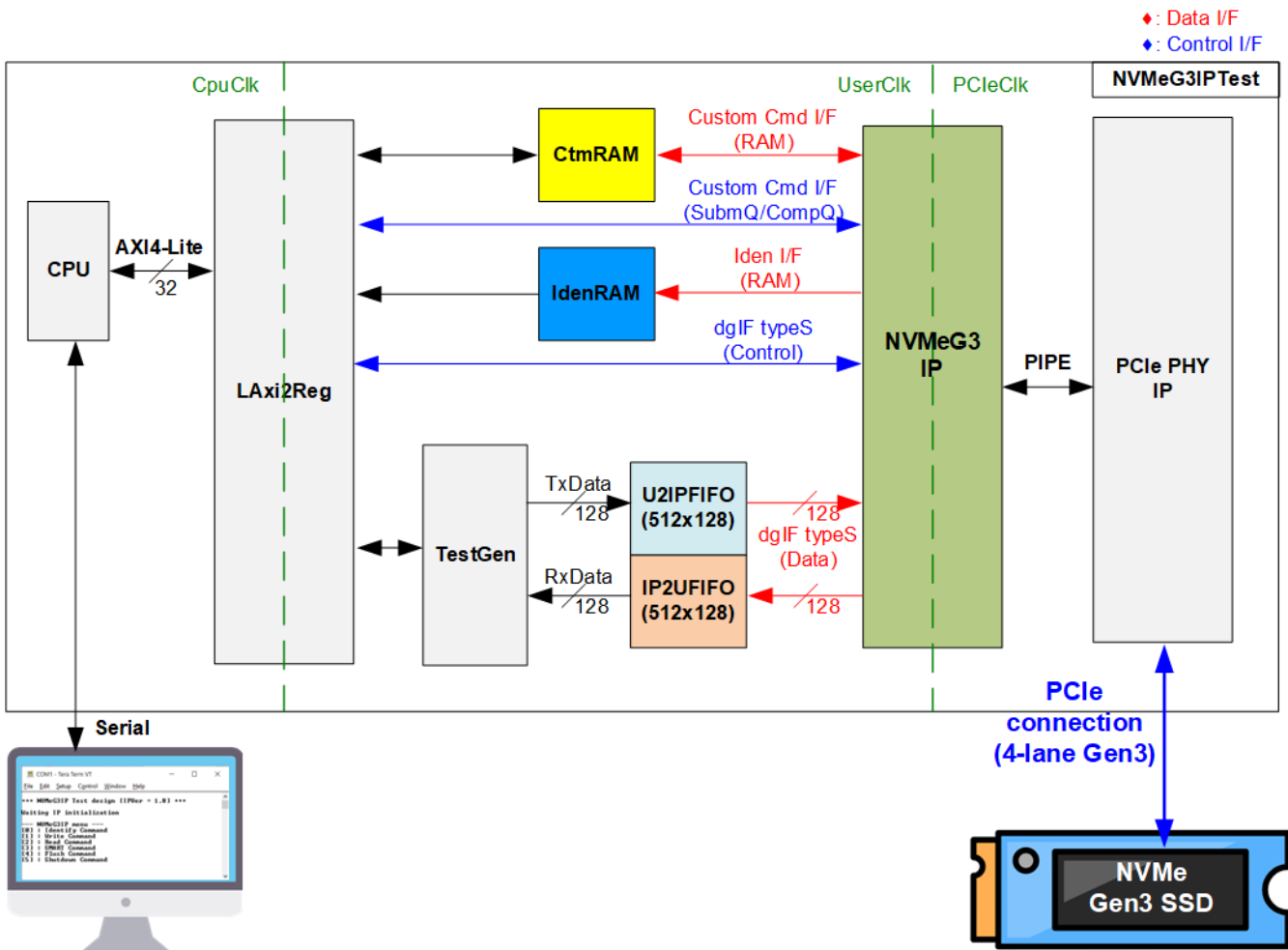


Figure 2-1 NVMeG3-IP demo hardware

User interface of NVMeG3-IP and NVMe-IP are identical. Thus, the modules for connecting user interface such as TestGen and CPU system in NVMeG3IPTest are designed using the same modules as the 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.

For details of the logic design and timing diagram of TestGen, LAXI2Reg, RAM, and FIFO, please refer to the NVMe-IP reference design document. However, an additional test pin for the MAC layer, called MACTestPin, is presented in NVMeG3-IP. This signal is applied for system debugging when the issues arise during 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 bits[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 bits[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 bits[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 bits[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) 000b: Identify, 001b: Shutdown, 010b: Write SSD, 011b: Read SSD, 100b: SMART/Secure Erase, 110b: Flush, 101b/111b: Reserved When this register is written, the command request is sent to NVMeG3-IP to start the operation.
BA+0x0014	Test Pattern Reg (PATTSEL_INTREG)	[2:0]: Select test pattern 000b-Increment, 001b-Decrement, 010b-All 0, 011b-All 1, 100b-LFSR
BA+0x0020	NVMe Timeout Reg (NVMTIMEOUT_INTREG)	[31:0]: Mapped to 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 (0b: Idle, 1b: Busy) [1]: UserError of dgIF typeS (0b: Normal, 1b: Error) [2]: Data verification fail (0b: Normal, 1b: 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]: Mapped to UserErrorType[31:0] of NVMeG3-IP to show error status
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	Register Name	Description
Rd/Wr	(Label in the "nvmeipstest.c")	
0x0100 – 0x01FF: Status signals of NVMeG3-IP and TestGen (Read access only)		
BA+0x0130 – BA+0x013F	Expected value Word0-3 Reg (EXPPATW0-W3_INTREG)	128-bit of the expected data of the 1 st failure when executing Read command. 0x0130: Bit[31:0], 0x0134[31:0]: Bit[63:32], ..., 0x013C[31:0]: Bit[127:96]
BA+0x0150 – BA+0x015F	Read value Word0-3 Reg (RDPATW0-W3_INTREG)	128-bit of the read data of the 1 st failure when executing 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 1 st failure when executing Read command
BA+0x0174	Data Failure Address(High) Reg (RDFAILNOH_INTREG)	[24:0]: Bit[56:32] of the byte address of the 1 st failure when executing 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	Custom Submission Queue Reg (CTMSUBMQ_STRUCT)	[31:0]: Submission queue entry of SMART, Secure Erase, or Flush command. Input to be CtmSubmDW0-DW15 of NVMeG3-IP. 0x200: DW0, 0x204: DW1, ..., 0x23C: DW15
BA+0x0300 – BA+0x030F	Custom Completion Queue Reg (CTMCOMPQ_STRUCT)	[31:0]: CtmCompDW0-DW3 output from NVMeG3-IP. 0x300: DW0, 0x304: DW1, ..., 0x30C: DW3
BA+0x0800	IP Version Reg (IPVERSION_INTREG)	[31:0]: Mapped to IPVersion[31:0] of NVMeG3-IP
BA+0x2000 – BA+0x2FFF	Identify Controller Data (IDENCTRL_CHARREG)	4KB Identify Controller Data structure
BA+0x3000 – BA+0x3FFF	Identify Namespace Data (IDENNAME_CHARREG)	4KB Identify Namespace Data structure
BA+0x4000 – BA+0x5FFF	Custom command RAM (CTMRAM_CHARREG)	Connect to 8KB CtmRAM interface for storing 512-byte data output from SMART Command.

3 CPU Firmware

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

3.1 Test firmware (nvmeiptest.c)

Once reset sequence is completed, the IP starts the initialization sequence by performing the following steps.

- 1) CPU initializes UART and Timer parameters.
- 2) CPU waits until NVMeG3-IP completes initialization process (USRSTS_INTREG[0]=0b). If any errors are detected, the process stops with displaying the corresponding error message.
- 3) CPU displays the main menu, which consists of seven menus for executing following commands: Identify, Write, Read, SMART, Flush, Secure Erase, and Shutdown.

The details of CPU firmware to operate all commands are similar to those of 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

This section presents the result achieved through the utilization of two SSD models: an 800GB Intel P5800X for maximizing write speed and a 512GB Samsung 970 Pro for maximizing read speed, as illustrated in Figure 4-1.

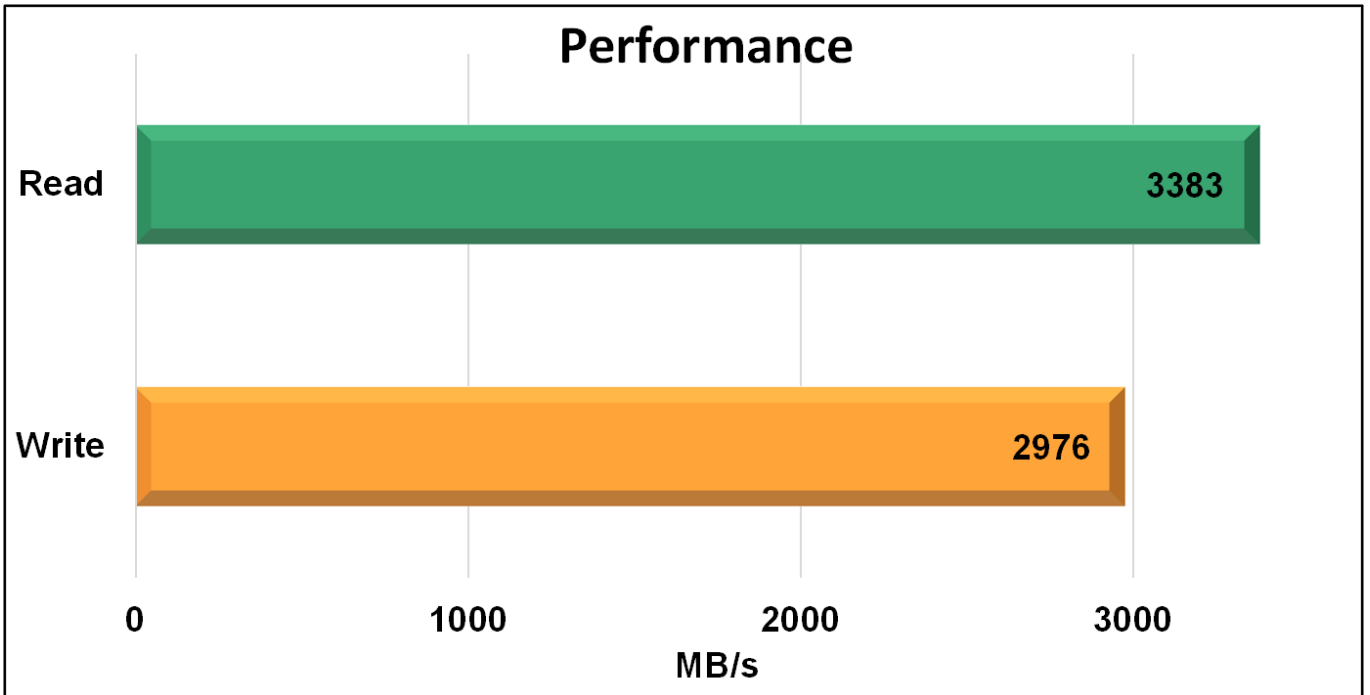


Figure 4-1 Test Performance of NVMeG3-IP demo

When utilizing PCIe Gen3 on ZCU102 board, the write performance reaches around 2900 MB/s, while the read performance reaches approximately 3300 MB/s.

5 Revision History

Revision	Date	Description
1.04	23-Feb-24	Support Secure Erase Command
1.03	22-Jun-23	Update NVMeG3-IP demo hardware figure
1.02	14-Mar-22	Update register name in the register map
1.01	4-Jun-21	Update register map
1.00	30-Aug-19	Initial Release

Copyright: 2019 Design Gateway Co.,Ltd.