NVMe-IP Introduction for Xilinx

Direct connection between latest PCIe Gen4 NVMe SSD and FPGA

The Very Best Solution for Data Recording Application!

Agenda

• NVMe-IP Introduction
  – Summary, Lineup, Merit
  – High Performance and Compact Size
  – Easy User Interface
  – Rich Features
  – Development Environment/Reference Design
• Optional product (exFAT-IP core)
• Application
What’s NVMe-IP

- **What’s NVMe-IP?** -> Directly connect NVMe SSD with FPGA
- **Advantage** -> No need for CPU, its F/W, External Memory
  Supports latest PCIe Gen4 protocol
- **Application** -> Best for ultra high speed data recording system
- **User Merit?** -> Can develop Storage Application in short period

**NVMe-IP Lineup**

- Multiple lineup for various functions
- PCIe Soft-IP furnished version (Gen4/Gen3) available
- raNVMe-IP suitable for random access application

<table>
<thead>
<tr>
<th>Core type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard NVMe-IP core</td>
<td>Standard core using PCIe Integrated Block in FPGA</td>
</tr>
<tr>
<td>NVMeG4-IP core</td>
<td>PCIe Soft-IP furnished, 4-Lane PCIe Gen4</td>
</tr>
<tr>
<td>NVMeG3-IP core</td>
<td>PCIe Soft-IP furnished, 4-Lane PCIe Gen3</td>
</tr>
<tr>
<td>NVMeSW-IPcore</td>
<td>Multiple SSD connection via external PCIe switch</td>
</tr>
<tr>
<td>raNVMe-IP core</td>
<td>Supports random read or write access</td>
</tr>
</tbody>
</table>
PCIe Soft-IP furnished IP core

- NVMeG4-IP core / NVMeG3-IP core
  - Can operate without PCIe Integrated Block
  - Includes data link layer and connect with transceiver by PCIe Gen4/3
  - More SSD connection regardless of PCIe Integrated Block count.

---

## PCIe Soft-IP furnished IP core (Cont’d)

<table>
<thead>
<tr>
<th>PCIe protocol</th>
<th>Product Number</th>
<th>Target device family</th>
<th>Supported transceiver</th>
<th>Evaluation env.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gen4 4Lane</td>
<td>NVMeG4-IP-VUP-GTY</td>
<td>Virtex-UltraScale+</td>
<td>GTY</td>
<td>VCU118</td>
</tr>
<tr>
<td></td>
<td>NVMeG4-IP-KUP-GTY</td>
<td>Kintex-UltraScale+</td>
<td>GTY</td>
<td>KCU116</td>
</tr>
<tr>
<td></td>
<td>NVMeG4-IP-ZUP-GTH</td>
<td>Zynq-UltraScale+</td>
<td>GTH</td>
<td>ZCU102/106</td>
</tr>
<tr>
<td>Gen3 4Lane</td>
<td>NVMeG3-IP-VUP-GTY</td>
<td>Virtex-UltraScale+</td>
<td>GTY</td>
<td>VCU118</td>
</tr>
<tr>
<td></td>
<td>NVMeG3-IP-KUP-GTY</td>
<td>Kintex-UltraScale+</td>
<td>GTY</td>
<td>KCU116</td>
</tr>
<tr>
<td></td>
<td>NVMeG3-IP-ZUP-GTH</td>
<td>Zynq-UltraScale+</td>
<td>GTH</td>
<td>ZCU102/106</td>
</tr>
<tr>
<td></td>
<td>NVMeG3-IP-KU-GTH</td>
<td>Kintex-UltraScale</td>
<td>GTH</td>
<td>KCU105</td>
</tr>
</tbody>
</table>

NVMeG4-IP/NVMeG3-IP core lineup

- Supports all UltraScale+ family and some UltraScale family
- Evaluation environment ready for all IP core products
raNVMe-IP for random access

- User can select either Write or Read operation
- Executes 32 commands at maximum concurrently with different (random) address.
- Write or read data per one command is fixed to 4KBytes.

\[
\begin{array}{c}
\text{Write Data (from User logic)}
\end{array}
\]

4KBytes data

\[
\begin{array}{c}
\text{NVMe SSD storage area}
\end{array}
\]

32 Write command concurrent operation

\[
\begin{array}{c}
\text{Read Data (to User logic)}
\end{array}
\]

32 Read command concurrent operation

raNVMe-IP concurrent command operation image

NVMe-IP Merit

1. High Performance and Compact size
   - Write=4288MB/s, Read=4670MB/s (measured by VCU118)
   - Support PCIe GEN4 (Operation confirmed on Ultrascale+)
   - IP-Core Size=4170CLBRegs, Memory=59BRAMTile (standard core)

2. Interface: Simple and easy connection
   - Direct connection to Xilinx Integrated Block for PCIe
   - User I/F control is parameter with pulse, data is simple FIFO
   - Use BRAM for data buffer (external DDR memory not required)

3. Rich Features: Custom command in addition to Read/Write
   - Supports SMART/FLUSH/Shutdown custom command
   - Supports both legacy 512byte and 4Kbyte Sector format

4. Environment: Full reference design project
   - Full Vivado project with real board operation in the package
Merit1: Performance (PCIe Soft-IP furnished)

- PCIe Gen4 supported core (NVMeG4-IP core) real speed
  - Unprecedented Write/Read performance!

**Performance Evaluation Result of NVMeG4-IP core**

- **Evaluation condition:**
  - FPGA Board: VCU118
  - SSD: AORUS GP-ASM2NE6100TTTD
- **Write test:** 256KB data buffer size
- **Read test:** 512KB data buffer size

<table>
<thead>
<tr>
<th>SSD Size</th>
<th>Read Performance</th>
<th>Write Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.669 GB</td>
<td>4670 MB/s</td>
<td>4288 MB/s</td>
</tr>
<tr>
<td>9.339 GB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14.518 GB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18.481 GB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23.351 GB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28.022 GB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32.693 GB</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Total = 34 (GB), Time = 735.6 ms, Transfer speed = 4670 MB/s

Merit1: Performance (Standard core)

- Automatic PCIe SSD access by pure hard-wired logic
  - Intelligent state machine for complete read/write command execution
  - Minimum over head and best performance by synchronized circuit

**Performance Evaluation Result (KCU105)**

- **Evaluation condition:**
  - SSD: Samsung MZ-V6P512BW
- **Write test:** 512KB data buffer size
- **Read test:** 1 GB data buffer size

<table>
<thead>
<tr>
<th>SSD Size</th>
<th>Read Performance</th>
<th>Write Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.669 GB</td>
<td>3252 MB/s</td>
<td>2148 MB/s</td>
</tr>
<tr>
<td>9.339 GB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14.518 GB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18.481 GB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23.351 GB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28.022 GB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32.693 GB</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Total = 34 (GB), Time = 18.0 ms, Transfer speed = 3252 MB/s
### Merit1: Compact Size (PCle Soft-IP furnished)

<table>
<thead>
<tr>
<th>PCle protocol</th>
<th>IP-Core</th>
<th>Example Device</th>
<th>Fmax (MHz)</th>
<th>CLB Regs</th>
<th>CLB LUTs</th>
<th>CLB</th>
<th>BRAM Tile</th>
<th>UR AM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gen4 4Lane</td>
<td>NVMeG4-IP-VUP-GTY</td>
<td>XCVU9P-FLGA2104-2LE</td>
<td>300</td>
<td>19215</td>
<td>21958</td>
<td>4465</td>
<td>12</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>NVMeG4-IP-KUP-GTY</td>
<td>XCKUSP-FFVB676-2E</td>
<td>300</td>
<td>19214</td>
<td>21960</td>
<td>4382</td>
<td>12</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>NVMeG4-IP-ZUP-GTH</td>
<td>XCZU7EV-FFVC1156-2E</td>
<td>300</td>
<td>19213</td>
<td>21961</td>
<td>4374</td>
<td>12</td>
<td>8</td>
</tr>
<tr>
<td>Gen3 4Lane</td>
<td>NVMeG3-IP-VUP-GTY</td>
<td>XCVU9P-FLGA2104-2LE</td>
<td>300</td>
<td>11107</td>
<td>12027</td>
<td>2546</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NVMeG3-IP-KUP-GTY</td>
<td>XCKUSP-FFVB676-2E</td>
<td>300</td>
<td>11107</td>
<td>12024</td>
<td>2461</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NVMeG3-IP-ZUP-GTH</td>
<td>XCZU7EV-FFVC1156-2E</td>
<td>300</td>
<td>11147</td>
<td>12029</td>
<td>2779</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NVMeG3-IP-KU-GTH</td>
<td>XCKU040FFVA1156-2E</td>
<td>300</td>
<td>11168</td>
<td>12047</td>
<td>2518</td>
<td>70</td>
<td></td>
</tr>
</tbody>
</table>

**NVMeG4/G3-IP core resource usage example**

- Limitation point of NVMeG4/G3-IP core
  - PCIe Gen4/Gen3 only, not support other speed (Gen1/2/4)
  - 4-Lane only, not support other lane count (1/2/8/16)
  (Ask other lane count as core customization)

### Merit1: Compact Size (Standard core)

- Optimized size with minimum resource consumption
  - Implements dedicated and optimized logic for NVMe SSD control
- Block RAM for data buffer
  - Internal block memory can minimize access overhead

---

**NVMe-IP Core standalone resource usage**
Merit2: Command I/F

- Simple User I/F
  - Set Command/Address/Length
  - Issue UserReq pulse
- Full Automatic control for SSD access
  - User only can wait UserBusy negation

---

Merit2: Data I/F

- Simple 128bit FIFO for each of read and write
  - General FIFO of standard Xilinx LogiCORE library
  - Data buffer using 256KByte BRAM in NVMe-IP

---
Merit3: Rich Features

- **SMART command for SSD health condition check**
  - Can monitor internal temperature, total write size, etc.

- **FLUSH command to force cache flush operation**
  - User can adjust trade-off between performance and data evacuation

- **Safe Shutdown before SSD power down**
  - IP-core executes safe shutdown by user request

- **Supports both 512bytes and 4Kbytes sector format**
  - IP-core automatically selects sector format via Identify command

<table>
<thead>
<tr>
<th>SMART Log Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature</td>
</tr>
<tr>
<td>Total Data Read</td>
</tr>
<tr>
<td>Total Data Written</td>
</tr>
<tr>
<td>Power On Cycles</td>
</tr>
<tr>
<td>Power On Hours</td>
</tr>
<tr>
<td>Unsafe Shutdowns</td>
</tr>
</tbody>
</table>

Merit4: Environment

- **Real operation check with Xilinx evaluation board**

- **Free bit-file for evaluation before IP-core purchase**
Merit4: Development Tool#1

- PCIe Adapter board for evaluation (Part#: AB16-PCIeXOVR)
  - Connect FPGA board to PCIe socket on component side
  - Connect PCIe SSD to PCIe socket on solder side
  - SSD R/W access via adapter board from NVMe-IP in FPGA

Merit4: Development Tool#2

- FMC Adapter board for evaluation (Part#: AB17-M2FMC)
  - Two M.2 sockets on component side
  - FMC HPC connector for FPGA connection on solder side
  - High capacity power supply (max 5A for 3.3V output per one SSD)
Merit4: Reference Design

- Vivado project is attached with NVMe-IP deliverables
- Full source code (VHDL) except IP core
- Can save user system development duration
  - Confirm real board operation by original reference design.
  - Then modify a little to approach final user product.
  - Check real operation in each modification step.

Short-term development is possible without big turn back

Optional product: exFAT-IP Core Introduction

- Optional products for NVMe-IP core
  - Supports data recording with exFAT file format
- PC can directly access to recorded data as a file
  - FPGA writes data to device, reconnect with PC, then PC can read data

PC can directly read recorded data as a file
Optional product: exFAT-IP (Cont’d 1)

- **Feature description**
  - Executes drive format and data write to file by pure hardwired logic.
  - IP core automatically generates file name.
  - User logic sends file data via FIFO interface.

- **Limitation**
  - Drive must be formatted by the IP core, not by the PC.
  - Files other than those generated by the IP core cannot be written to the drive.
  - File size is determined at format execution and cannot be changed.

Optional product: exFAT-IP (Cont’d 2)

- **Reference design for real operation available**
  - Executes test file generation via serial console.
  - User can confirm file read compatibility by drive re-plug to the PC.

Generate test file, reconnect with PC, and can check file read compatibility
NVMe-IP Application Example 1

- Space-Saving FPGA data logging system
  - Latest FPGA+M.2 SSD

Include data logging user logic and NVMe-IP in FPGA

Record log data to M.2 SSD

FPGA (FBG484)

M.2 Connector

M.2 SSD

System area image by FBG484 FPGA and M.2 SSD (unit: mm)

NVMe-IP Application Example 2

- Recording and Analysis system on Linux
  - Mount Linux and user analysis application on SoC/MPSoC device
  - Very high-speed data recording to SSD via NVMe-IP core
  - Data read from SSD via device driver and analyze by user application

(1) Very high-speed write to SSD by NVMe-IP

(2) Read SSD data via device driver

(3) Analyze data by user application on Linux

(4) Can send analyzed data to the external

Recording and Analysis system on Linux (device driver and reference design available)
NVMe-IP Application Example 3

- **Ultra High-Speed Recorder**
  - Double write speed with multiple SSDs RAID0 configuration
  - Provide RAID0 reference design with 2 NVMe SSDs

![Diagram showing NVMe RAID system supporting 4GB/sec recording rate]

When system requires 4GB/sec write speed

Build 2ch RAID0 with 2 NVMe-IP

4GB/sec with 2 SSDs in parallel

NVMe RAID system supporting 4GB/sec recording rate

1-Dec-20 Design Gateway Page 25

NVMe-IP Application Example 4

- **Super multi-channel SSD Array by NVMeG3-IP**

![Diagram showing 30 channels M.2 SSD Array system using NVMeG4/G3-IP core]

Can build NVMe channels up to ¼ of total transceiver count

30 channels SSD connection possible by VU9P (A2577 package with 120 transceivers)

30 channels M.2 SSD Array system using NVMeG4/G3-IP core

1-Dec-20 Design Gateway Page 26
For more detail

- Detailed technical information available on the web site.
  - [https://dgway.com/NVMe-IP_X_E.html](https://dgway.com/NVMe-IP_X_E.html)

- Contact
  - Design Gateway Co., Ltd.
  - sales@design-gateway.com
  - FAX: +66-2-261-2290

---

### Revision History

<table>
<thead>
<tr>
<th>Rev.</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0E</td>
<td>10-Jun-16</td>
<td>English Version first release</td>
</tr>
<tr>
<td>1.1E</td>
<td>21-Jun-16</td>
<td>Support Kintex-Ultrascale</td>
</tr>
<tr>
<td>1.2E</td>
<td>25-Aug-16</td>
<td>Modify page17 because only one x16 DDR4 device can keep NVMe SSD performance</td>
</tr>
<tr>
<td>1.3E</td>
<td>12-Sep-16</td>
<td>Support Zynq-7000 and Kintex-7</td>
</tr>
<tr>
<td>1.4E</td>
<td>8-Nov-16</td>
<td>Support PCIe GEN3 on Virtex-7</td>
</tr>
<tr>
<td>1.5E</td>
<td>21-Dec-16</td>
<td>NVMe-IP core improvement by removing external DDR chip for data buffer</td>
</tr>
<tr>
<td>1.6E</td>
<td>6-Jun-17</td>
<td>Performance improved by internal PCIe bridge in NVMe-IP core</td>
</tr>
<tr>
<td>1.7E</td>
<td>2-Nov-17</td>
<td>Added Linux driver application and 2ch RAID0 reference design</td>
</tr>
<tr>
<td>1.8E</td>
<td>18-Jul-18</td>
<td>Added 4KB sector format, SMART/FLUSH/Shutdown command support</td>
</tr>
<tr>
<td>1.9E</td>
<td>9-Jan-19</td>
<td>Add FAT32-IP/exFAT-IP for NVMe-IP optional products</td>
</tr>
<tr>
<td>2.0E</td>
<td>24-Sep-19</td>
<td>Add new product of NVMeG3-IP that includes PCIe Soft IP core inside</td>
</tr>
<tr>
<td>2.1E</td>
<td>1-Feb-20</td>
<td>Add new product of NVMeG4-IP that includes PCIe Gen4 Soft IP core inside</td>
</tr>
<tr>
<td>2.2E</td>
<td>27-Aug-20</td>
<td>Add new product of raNVMe-IP for random access application</td>
</tr>
<tr>
<td>2.3E</td>
<td>1-Dec-20</td>
<td>Updated NVMeG4-IP/NVMeG3-IP information</td>
</tr>
</tbody>
</table>