

# Nvme Performance Test

By: NXP Semiconductor

## Contents

|            |   |    |
|------------|---|----|
| 1          | Compile SPDK.....                               | 3  |
| 1.1        | Compile on LX2160ARDB board .....               | 3  |
| 1.1.1      | Compile libuuid.....                            | 3  |
| 1.1.2      | Compile DPDK .....                              | 3  |
| 1.1.3      | Compile SPDK.....                               | 3  |
| 1.2        | Cross Compile.....                              | 4  |
| 1.2.1      | Setting environment variables .....             | 4  |
| 1.2.2      | Compile libaio .....                            | 4  |
| 1.2.3      | Compile libuuid.....                            | 4  |
| 1.2.4      | Compile openssl .....                           | 4  |
| 1.2.5      | Compile CUnit .....                             | 4  |
| 1.2.6      | Compile DPDK .....                              | 5  |
| 1.2.7      | Compile SPDK.....                               | 5  |
| 2          | Test performance with the fio tool.....         | 6  |
| 2.1        | Setting up an NVMe SSD on LX2160ARDB board..... | 6  |
| 2.2        | Performance testing.....                        | 6  |
| 3          | Test performance with SPDK perf tool .....      | 7  |
| 3.1        | Set up the spdk runtime environment .....       | 7  |
| 3.2        | Performance testing.....                        | 8  |
| 4          | Test performance with iscsi .....               | 9  |
| 4.1        | Run iscsi target.....                           | 9  |
| 4.2        | Run iscsi initiator .....                       | 11 |
| 4.2.1      | Build a proxy server .....                      | 11 |
| 4.2.2      | Install open-iscsi.....                         | 11 |
| 4.2.3      | Get iscsi kernel driver .....                   | 11 |
| 4.2.4      | Copy files to the board .....                   | 12 |
| 4.2.5      | Set u-boot .....                                | 12 |
| 4.2.6      | Start iscsi.service.....                        | 12 |
| 4.2.7      | Connect to iscsi target .....                   | 12 |
| 4.2.8      | Performance testing.....                        | 13 |
| 4.2.9      | Disconnect from target .....                    | 14 |
| Appendix A | Test instructions .....                         | 15 |

Appendix B Revision history..... 16

# 1 Compile SPDK

Before compile spdk, you need to compile its dependencies.

## 1.1 Compile on LX2160ARDB board

Make sure the file system is installed on your board.

### 1.1.1 Compile libuuid

First Get *libuuid-1.0.3.tar.gz* file, then execute the following command.

```
# tar -zxvf libuuid-1.0.3.tar.gz
# cd libuuid-1.0.3
# ./configure
# make && make install
```

### 1.1.2 Compile DPDK

Get the dpdk source code.

```
# git clone ssh://git@bitbucket.sw.nxp.com/dnstorage/dpdk.git
# cd dpdk
# git checkout -b devel origin/iNIC-devel
```

Compile dpdk.

```
# export RTE_TARGET=arm64-dpaa2-linuxapp-gcc
# export RTE_SDK=/code/dpdk
# export DESTDIR=/code/dpdk/install
# make T=arm64-dpaa2-linuxapp-gcc CONFIG_RTE_LIBRTE_PMD_KNI=y CONFIG_RTE_KNI_KMOD=n
install
```

### 1.1.3 Compile SPDK

Get spdk source code.

```
# git clone https://github.com/spdk/spdk
```

Compile spdk.

```
# cd spdk
# ./configure -with-dpdk=/code/dpdk/arm64-dpaa2-linuxapp-gcc
# make -j8 install DESTDIR=/PATH-TO-INSTALL
```

NOTE: A *PAGE\_SIZE* error has occurred. Find the file manually change *PAGE\_SIZE* to 4096. The */test* directory is used for unit testing.

## 1.2 Cross Compile

### 1.2.1 Setting environment variables

```
# export CROSS=/usr/local/src/  
gcc-linaro-5.3.1-2016.05-x86_64_aarch64-linux-gnu/bin/aarch64-linux-gnu-  
  
# export CC=${CROSS}gcc  
  
# export AR=${CROSS}ar  
  
# export LD=${CROSS}ld
```

### 1.2.2 Compile libaio

```
# tar -zxvf libaio-0.3.111.orig.tar.gz  
  
# cd libaio-0.3.111  
  
# make && make install
```

### 1.2.3 Compile libuuid

```
# tar -zxvf libuuid-1.0.3.tar.gz  
  
# cd libuuid-1.0.3  
  
# ./configure --host=aarch64-linux-gnu  
  
# make -j8 install DESTDIR=/home/libuuid
```

### 1.2.4 Compile openssl

Get openssl source code.

```
# git clone ssh://git@bitbucket.sw.nxp.com/dnstorage/openssl.git
```

Compile openssl

```
# cd openssl  
  
# export CROSS_COMPILE=${CROSS}  
  
# /usr/bin/perl ./Configure no-asm shared linux-aarch64 --prefix=/opt/openssl/  
  
# make && make install
```

### 1.2.5 Compile CUnit

```
# tar jxvf CUnit-2.1-3.tar.bz2
# cd CUnit-2.1-3
# autoscan
# mv configure.in comfigure.ac
# aclocal
# autoheader
# libtoolize -automake -copy -debug -force
# automake
# autoconf
# ./configure --host=aarch64-linux-gnu
# make -j8 install DESTDIR=/home/CUnit
```

NOTE: if you get “Cannot find install-sh” error when executing *./configure --host=aarch64-linux-gnu*, you can execute *autoreconf -vif* and then execute *make -j8 install DESTDIR=/home/CUnit*.

### 1.2.6 Compile DPDK

```
# cd dpdk
# export RTE_TARGET=arm64-dpaa2-linuxapp-gcc
# export RTE_SDK=/home/dpdk
# export DESTDIR=/home/dpdk/install
# make T=arm64-dpaa2-linuxapp-gcc CONFIG_RTE_LIBRTE_PMD_KNI=y CONFIG_RTE_KNI_KMOD=n
install
```

### 1.2.7 Compile SPDK

```
# cd spdk
# export CFLAGS="-I/home/libuuid/usr/local/include -I/home/libaio-0.3.111/src
-I/opt/openssl/include -I/home/CUnit/usr/local/include"
# export LDFLAGS="-L/home/libuuid/usr/local/lib -L/home/libaio-0.3.111.src
-L/opt/openssl/lib -L/home/CUnit/usr/local/lib"
# export
CXX="/usr/local/src/gcc-linaro-5.3.1-2016.05-x86_64_aarch64-linux-gnu/bin/aarch64-l
inux-gnu-c++"
# ./configure --with-dpdk=/home/dpdk/arm64-dpaa2-linuxapp-gcc
```

```
# make -j8 install DESTDIR=/path/to/install
```

NOTE: A *PAGE\_SIZE* error has occurred. Find the file manually change *PAGE\_SIZE* to 4096. The */test* directory is used for unit testing.

## 2 Test performance with the fio tool

### 2.1 Setting up an NVMe SSD on LX2160ARDB board

First off is to create a partition:

```
# fdisk /dev/nvme0n1
```

Choose “n” to create a new partition, then “p” then “1” to create a new primary partition. Just use defaults for the sector numbers. Then “w” to write the data to the disk. The *lsblk* command should show something like:

```
localhost login:
Ubuntu 16.04.4 LTS localhost ttyAMA0

localhost login: root
Password:
Last login: Fri Dec 21 10:19:32 UTC 2018 on ttyAMA0
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.9.62+g4e9fcd4 aarch64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
root@localhost:~# lsblk
NAME        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
mtdblock0   31:0    0    64M 0 disk
mmcblk1boot0 179:32  0     8M 1 disk
mmcblk1     179:0    0 116.5G 0 disk
└─mmcblk1p1 179:1    0 116.5G 0 part /
mmcblk1rpb 179:96  0     4M 0 disk
mtdblock1   31:1    0    64M 0 disk
mmcblk1boot1 179:64  0     8M 1 disk
nvme0n1     259:0    0 372.6G 0 disk
└─nvme0n1p1 259:1    0 372.6G 0 part
root@localhost:~#
```

Figure 2-1: lsblk cmd show

To create a file system on it:

```
# mkfs -t ext4 /dev/nvme0n1p1
```

### 2.2 Performance testing

PCIe link status is x4 gen1.

```
CS4223: WARNING: Using EEPROM configuration...
CS4223: WARNING: Change SW2[2] for software config
FSI_MDI01:0 is connected to DPMAC5@25g-auri. Reconnecting to DPMAC6@25g-auri
PCIe0: pcie@3400000 disabled
PCIe1: pcie@3500000 disabled
PCIe2: pcie@3600000 Root Complex: x4 gen1
PCIe3: pcie@3700000 disabled
PCIe4: pcie@3800000 Root Complex: no link
PCIe5: pcie@3900000 disabled
DPMAC2@xlauri4, DPMAC3@xgmii, DPMAC4@xgmii, DPMAC5@25g-auri, DPMAC6@25g-auri, DPMAC17@rgmii-id, DPMAC18@rgmii-id
crc32+
```

Figure 2-2: PCIe link status

The commands and parameters executed are as follows:

```
# fio -ioengine=libaio -bs=${bs} -direct=1 -thread -rw=${rw} -size=10G
-filename=/dev/nvme0n1p1 -name=mytest -iodepth=32 -runtime=60
```

Test results.

| rw \ bs   | 2k                        | 4k                       | 64k                      | 1M                     | 2M                     |
|-----------|---------------------------|--------------------------|--------------------------|------------------------|------------------------|
| read      | bw=194MB/s<br>iops= 99678 | bw=365MB/s<br>iops=93542 | bw=820MB/s<br>iops=13120 | bw=820MB/s<br>iops=820 | bw=820MB/s<br>iops=410 |
| randread  | bw=179MB/s<br>iops=92158  | bw=344MB/s<br>iops=88290 | bw=819MB/s<br>iops=13119 | bw=820MB/s<br>iops=820 | bw=820MB/s<br>iops=410 |
| write     | bw=115MB/s<br>iops=59299  | bw=348MB/s<br>iops=89234 | bw=692MB/s<br>iops=11085 | bw=697MB/s<br>iops=697 | bw=697MB/s<br>iops=348 |
| randwrite | bw=169MB/s<br>iops=86844  | bw=333MB/s<br>iops=85272 | bw=692MB/s<br>iops=11082 | bw=697MB/s<br>iops=697 | bw=697MB/s<br>iops=348 |

### 3 Test performance with SPDK perf tool

#### 3.1 Set up the spdk running environment

```
# ulimit -l 65536
```

Make sure `/lib/modules/`uname -r`/` directory exists.

```
# mkdir -p /lib/modules/`uname -r`/
# cd /lib/modules/`uname -r`/
# cp ../4.9.79/* ./
```

Modify `/spdk/scripts/setup.sh` file. Add a line, use `uio_pci_generic` driver.

```
function configure_linux_pci {
    if [ -z "${DRIVER_OVERRIDE}" ]; then
        driver_name=vfio-pci
        if [ -z "$(ls /sys/kernel/iommu_groups)" ]; then
            # No IOMMU. Use uio.
            driver_name=uio_pci_generic
        fi
    else
        driver_name="${DRIVER_OVERRIDE}"
    fi
    driver_name=uio_pci_generic
}
```

Figure 3-1: setup.sh

NOTE: when execute `echo "0000:01:00.0" > /sys/bus/pci/driver/vfio-pci/bind`, There will be an error that the device does not exist. So use `uio_pci_generic` driver.

```
# ./setup.sh
```

### 3.2 Performance testing

```
# cd /spdk/examples/nvme/perf
# ./perf -h
# ./perf -q 32 -s 1024 -w ${rw} -t 60 -o ${bs} -r 'trtype:PCIe traddr:0000:01:00.0'
```

The `./perf -q 32 -s 1024 -w randwrite -t 60 -o 2048 -r 'trtype:PCIe traddr:0000:01:00.0'` command should show something like:

```
root@localhost:/lt/spdk/new/spdk/examples/nvme/perf# ls
Makefile perf perf.c perf.d perf.o README.md
dwrite -t 60 -o 2048 -r 'trtype:PCIe traddr:0000:01:00.0'rf -q 32 -s 1024 -w ran
Starting SPDK v19.01-pre / DPDK 17.11.2 initialization...
[ DPDK EAL parameters: perf --no-shconf -c 0x1 -m 1024 --base-virtaddr=0x200000000000 --file-prefix=spdk_pid4263 ]
EAL: Detected 16 lcore(s)
EAL: Probing VFIO support...
EAL: VFIO support initialized
Initializing NVMe Controllers
EAL: PCI device 0000:01:00.0 on NUMA socket 0
EAL: probe driver: 8086:953 spdk nvme
Attaching to NVMe Controller at 0000:01:00.0
Attached to NVMe Controller at 0000:01:00.0 [8086:0953]
Associating INTEL SSDPEDMD400G4 (CVFT7384007F400BGN ) with lcore 0
Initialization complete. Launching workers.
Starting thread on core 0
=====
Device Information           : IOPS      MB/s      Average  Latency (us)
INTEL SSDPEDMD400G4 (CVFT7384007F400BGN ) from core 0: 215376.95  420.66    148.55   min      max
=====
Total                       : 215376.95  420.66    148.55   11.28    6621.12
root@localhost:/lt/spdk/new/spdk/examples/nvme/perf# █
```

Test results:

| bs \ rw   | 2048(2k)                  | 4096(4k)                  | 8192(8k)                  | 65536(64k)               | 131072(128k)            |
|-----------|---------------------------|---------------------------|---------------------------|--------------------------|-------------------------|
| read      | bw=401MB/s<br>iops=205448 | bw=716MB/s<br>iops=183361 | bw=812MB/s<br>iops=104034 | bw=820MB/s<br>iops=13128 | bw=821MB/s<br>iops=6568 |
| randread  | bw=623MB/s<br>iops=319384 | bw=802MB/s<br>iops=205464 | bw=812MB/s<br>iops=104047 | bw=820MB/s<br>iops=13128 | bw=821MB/s<br>iops=6568 |
| write     | bw=111MB/s<br>iops=57276  | bw=653MB/s<br>iops=167209 | bw=671MB/s<br>iops=85969  | bw=695MB/s<br>iops=11131 | bw=701MB/s<br>iops=5611 |
| randwrite | bw=420MB/s<br>iops=215376 | bw=654MB/s<br>iops=167460 | bw=669MB/s<br>iops=85743  | bw=693MB/s<br>iops=11091 | bw=696MB/s<br>iops=5573 |



Note: **The performance test data obtained above is not preprocessed.** Random write IOPS by Perf is over than NVMe device spec. Please refer to [Appendix-A](#) for Intel's performance test data for this product. NVMe preconditioning Please refer to [this](#).

## 4 Test performance with iscsi

We need two LX2160ARDB boards. Run spdk iscsi target on a board with nvme ssd. Run the iSCSI initiator on another board. The dpmac.5/dpmac.6 ports of the two boards are connected by SFP modules.

### 4.1 Run iscsi target

If you use dpmac.6, you can execute the following command:

```
# ls-addni dpmac.6  
  
# ls-listni  
  
# ifconfig ni2 1.1.1.1 netmask 255.255.255.0
```

Copy *iscsi.conf.in*

```
# cd spdk/app/iscsi_tgt  
  
# cp spdk/etc/spdk/iscsi.conf.in ./
```

Modify *iscsi.conf.in* configure file. Below is my configuration file, commented out other configuration with '#', commented out the configuration I did not write.

```
[Global]
[iSCSI]
NodeBase "iqn.2016-06.io.spdk"
MinConnectionsPerCore 4
Timeout 30
DiscoveryAuthMethod Auto
DefaultTime2Wait 2
DefaultTime2Retain 60
FirstBurstLength 8192
ImmediateData Yes
ErrorRecoveryLevel 0
[PortalGroup1]
Portal DA1 1.1.1.1:3260
[InitiatorGroup1]
InitiatorName ANY
Netmask 1.1.1.0/24
[Nvme]
TransportID "trtype:PCIe traddr:0000:01:00.0" Nvme0
TimeoutUsec 0
ActionOnTimeout None
HotplugEnable No
HotplugPollRate 0
[Ioat]
Enable No
Whitelist 00:04.0
Whitelist 00:04.1
[Pmem]
[TargetNode1]
TargetName disk1
TargetAlias "Data Disk1"
Mapping PortalGroup1 InitiatorGroup1
AuthMethod Auto
UseDigest Auto
LUN0 Nvme0n1
QueueDepth 128
```

Run iscsi\_tgt

```
# ./iscsi_tgt -h
# ./iscsi_tgt -c ./nvme.conf.in
```

## 4.2 Run iscsi initiator

### 4.2.1 Build a proxy server

I built a proxy server with TinyProxy on my centos system.

### 4.2.2 Install open-iscsi

```
# export http_proxy=http://USER:PASSWORD@IPADDR:PORT/
# apt-get update
# apt-get install open-iscsi
```

NOTE: Running the iscsi related command at this moment will fail due to the lack of related drivers.

### 4.2.3 Get iscsi kernel driver

Get linux kernel source code.

```
# git ssh://git@bitbucket.sw.nxp.com/dnstorage/dash-lts.git
# cd dash-lts
# git checkout -b iscsi-bsp0.4 origin/iNIC-bsp0.4
```

Add iscsi related configuration at the beginning of *./arch/arm64/configs/lSDK.config*

```
#ISCSI
CONFIG_INET=y
CONFIG_SCSI_MOD=y
CONFIG_SCSI=y
CONFIG_BLK_DEV_SD=m
CONFIG_CHR_DEV_SG=y
CONFIG_SCSI_LOGGING=y
CONFIG_SCSI_LOWLEVEL=y
CONFIG_ISCSI_TCP=m
CONFIG_SCSI_ISCSI_ATTRS=m
```

Cross-compiling the linux kernel.

```
# cd dash-lts
# source /opt/fsl-qoirq/2.5/environment-setup-aarch64-fsl-linux
# export LDFLAGS="--hash-style=gnu"
# make distclean
# make defconfig
```

```
# make lsdk.config
# make -j4 all
# make modules_install
```

#### 4.2.4 Copy files to the board

Remove the *build* and *source* soft connections under the */lib/modules/4.9.62-00346-g6ba215d-dirty* directory.

```
# cd /lib/modules/4.9.62-00346-g6ba215d-dirty
# rm -rf ./build
# rm -rf ./source
```

Copy the files under */lib/modules/4.9.62-00346-g6ba215d-dirty* to the */lib/modules/4.9.62-00346-g6ba215d-dirty* directory on the board.

Copy the **Image** file under the */dash-lts/arch/arm64/boot* directory to the board.

Copy the **fsl-lx2160a-rdb.dtb** file under the */dash-lts/arch/arm64/boot/dts/freescale* directory to the board.

#### 4.2.5 Set u-boot

Use the above compiled Image when the board starts up.

```
=> setenv ubuntubootargs "console=ttyAMA0,115200 root=/dev/mmcblk1p1 rw rootwait
earlycon=pl011,mmio32,0x21c0000 default_hugepagesz=1024m hugepagesz=1024m
hugepages=2 pci=pcie_bus_perf "
=> setenv ubuntuboot 'setenv bootargs $ubuntubootargs;fsl_mc apply dpl
0x20d00000;ext2load mmc 1:1 81000000 /Image;ext2load mmc 1:1 90000000 /
fsl-lx2160a-rdb.dtb; booti 81000000 - 90000000'
```

#### 4.2.6 Start iscsi.service

```
# depmod -a
# systemctl list-unit-files |grep iscsi
# systemctl start iscsi.service
# systemctl start iscsid.service
# systemctl start open-iscsi.service
```

#### 4.2.7 Connect to iscsi target

If you use *dpmac.6*, you can execute the following command:

```
# ls-addni dpmac.6

# ls-listni

# ifconfig ni2 1.1.1.2 netmask 255.255.255.0
```

### Discovery iscsi target

```
# iscsiadm -m discovery -t sendtargets -p 1.1.1.1
```

### Connect to target

```
# iscsiadm -m node -login
Or
# iscsiadm -m node -T iqn.2016-06.io.spdk:disk1 -l
```

The *lsblk* command should show something like:

```
root@localhost:~# iscsiadm -m discovery -t sendtargets -p 1.1.1.1
1.1.1.1:3260,1 iqn.2016-06.io.spdk:disk1
root@localhost:~# iscsiadm -m node --login
Logging in to [iface: default, target: iqn.2016-06.io.spdk:disk1, portal: 1.1.1.1,3260] (multiple)
Login to [iface: default, target: iqn.2016-06.io.spdk:disk1, portal: 1.1.1.1,3260] successful.
root@localhost:~# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
mtdblock0   31:0    0   64M  0 disk
mmcblk1boot0 179:32  0    8M  1 disk
mmcblk1     179:0    0 116.5G  0 disk
├─mmcblk1p1  179:1    0   80G  0 part /
└─mmcblk1p2  179:2    0  36.5G  0 part
mmcblk1rmpb 179:96  0    4M  0 disk
mtdblock1   31:1    0   64M  0 disk
mmcblk1boot1 179:64  0    8M  1 disk
sda         8:0     0 372.6G  0 disk
└─sda1      8:1     0 372.6G  0 part
root@localhost:~#
```

## 4.2.8 Performance testing

The commands and parameters executed are as follows:

```
# fio -filename=/dev/sda1 -direct=1 -iodepth=32 -thread -rw=${rw} -ioengine=libaio
-bs=${bs} -size=5G -numjobs=1 -name=mytes
```

Test results:

|          | bs | 2k                       | 4k                      | 64k                     | 1M                     | 2M                    |
|----------|----|--------------------------|-------------------------|-------------------------|------------------------|-----------------------|
| rw       |    |                          |                         |                         |                        |                       |
| read     |    | bw=49MB/s<br>iops= 25316 | bw=79MB/s<br>iops=20349 | bw=148MB/s<br>iops=2370 | bw=171MB/s<br>iops=171 | bw=171MB/s<br>iops=85 |
| randread |    | bw=49MB/s<br>iops=25208  | bw=78MB/s<br>iops=20123 | bw=152MB/s<br>iops=2442 | bw=171MB/s<br>iops=171 | bw=171MB/s<br>iops=85 |

|           |                         |                         |                        |                      |                      |
|-----------|-------------------------|-------------------------|------------------------|----------------------|----------------------|
| write     | bw=38MB/s<br>iops=19837 | bw=67MB/s<br>iops=17219 | bw=92MB/s<br>iops=1478 | bw=95MB/s<br>iops=95 | bw=95MB/s<br>iops=47 |
| randwrite | bw=40MB/s<br>iops=20627 | bw=68MB/s<br>iops=17635 | bw=93MB/s<br>iops=1489 | bw=95MB/s<br>iops=95 | bw=94MB/s<br>iops=47 |

#### 4.2.9 Disconnect from target

```
# iscsiadm -m node --logout  
Or  
# iscsiadm -m node -U all (Exit all logged in targets)
```

## Appendix A Test instructions

### 1. Nvme ssd model

The NVMe ssd model used in the test is INTEL-SSD-DC-P3700-SERIES-400GB, The following link is the performance data of intel:

[https://www.intel.com/content/dam/support/us/en/documents/ssdc/hpssd/sb/Intel\\_SSD\\_DC\\_P3700\\_Series\\_PCIE\\_Product\\_Specification-005.pdf](https://www.intel.com/content/dam/support/us/en/documents/ssdc/hpssd/sb/Intel_SSD_DC_P3700_Series_PCIE_Product_Specification-005.pdf)

[https://ark.intel.com/zh-cn/products/79625/Intel-SSD-DC-P3700-Series-400GB-2\\_5in-PCIe-3\\_0-20nm-MLC](https://ark.intel.com/zh-cn/products/79625/Intel-SSD-DC-P3700-Series-400GB-2_5in-PCIe-3_0-20nm-MLC)

### 2. PCIe link status on the LX2160ARDB board

PCIe link status is x4 gen1.

```
CS4223: WARNING: Using EEPROM configuration...
CS4223: WARNING: Change SW2[2] for software config
FSL MDIO1:0 is connected to DPMAC5@25g-aii. Reconnecting to DPMAC6@25g-aii
PCIe0: pcie@3400000 disabled
PCIe1: pcie@3500000 disabled
PCIe2: pcie@3600000 Root Complex: x4 gen1
PCIe3: pcie@3700000 disabled
PCIe4: pcie@3800000 Root Complex: no link
PCIe5: pcie@3900000 disabled
DPMAC2@xlai4, DPMAC3@xgmii, DPMAC4@xgmii, DPMAC5@25g-aii, DPMAC6@25g-aii, DPMAC17@rgmii-id, DPMAC18@rgmii-id
crc32+
```

### 3. More about SPDK

About spdk, you can get more from [here](#).

## Appendix B Revision history

| Date       | Version | Reason          |
|------------|---------|-----------------|
| 12/25/2018 | 1.0     | Initial version |