NVMeTCP10G-IP Demo Instruction

Rev1.02 29-Apr-24

1 Overview

This document provides detailed instructions for running the NVMeTCP10G-IP demo, which facilitates access to an NVMe SSD within an NVMe/TCP Target. The target system setup of this demonstrates utilizes a PC running Ubuntu 20.04.1 OS. Operations performed during the demo involve writing to and reading from the NVMe SSD via a 10G Ethernet connection. These operations adhere to the NVMe/TCP sequence process, which users can manage through the FPGA console interface.

This guide is divided into two sections. Section 2 details the setup procedure for initializing the NVMe SSD to run as an NVMe/TCP target and the steps to safety remove the NVMe from the NVMe/TCP target configuration on a PC with Ubuntu OS. Section 3 explains the operations performed via the FPGA console, including detailed explanations of each test step and the presentation of test results.

2 PC Setup for NVMe/TCP

2.1 Application Installation

Prior to running the NVMeTCP10G-IP demo, it is necessary to install specific applications on the PC. These applications need to be installed only once and are not required to be reinstalled for subsequent uses.

Ethtool: This utility is used to tune network card performance. Install Ethtool by executing the following command in the terminal:

```
sudo apt install ethtool
```

NVMe Command Line Interface (NVMe-CLI): This tool is used for managing NVMe SSDs under Linux OS. Install NVMe-CLI using the command:

```
sudo apt install nvme-cli
```

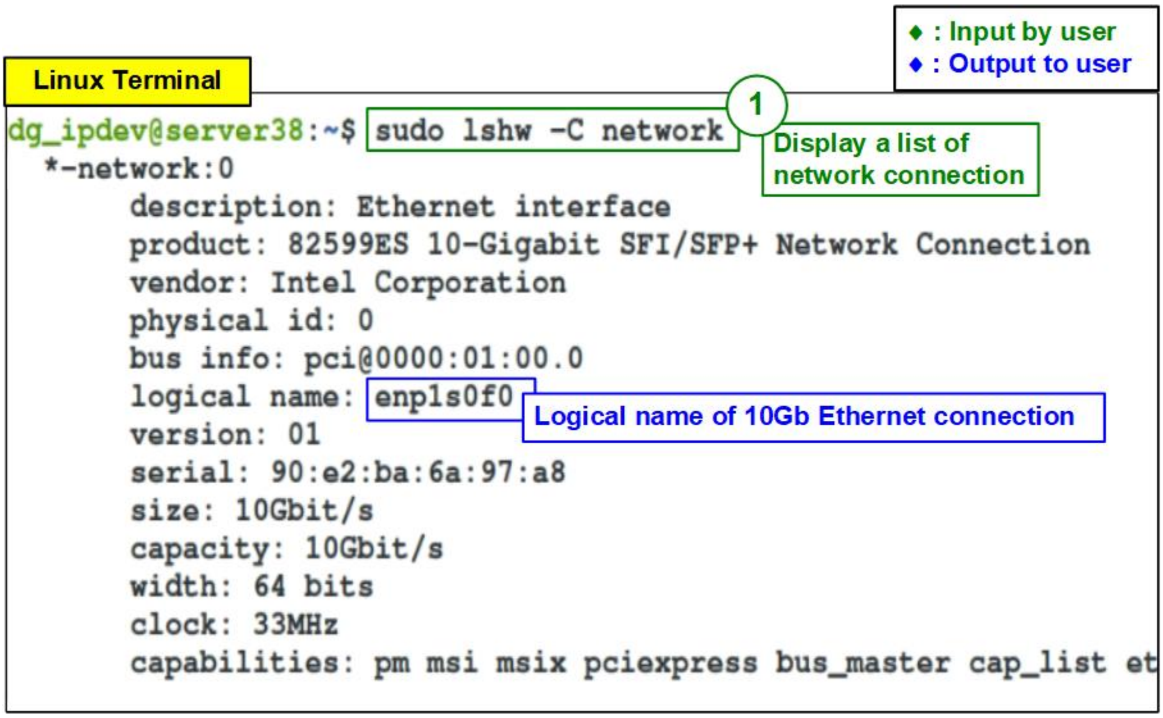
2.2 Ethernet Interface Setting on PC

To optimize the performance of the 10G Ethernet network card before running the demo, follow these steps. Open a new terminal window to begin configuring the Ethernet interface.

1) To list the logical name of 10G Ethernet port on the Linux terminal, use following command:

```
sudo lshw -C network
```

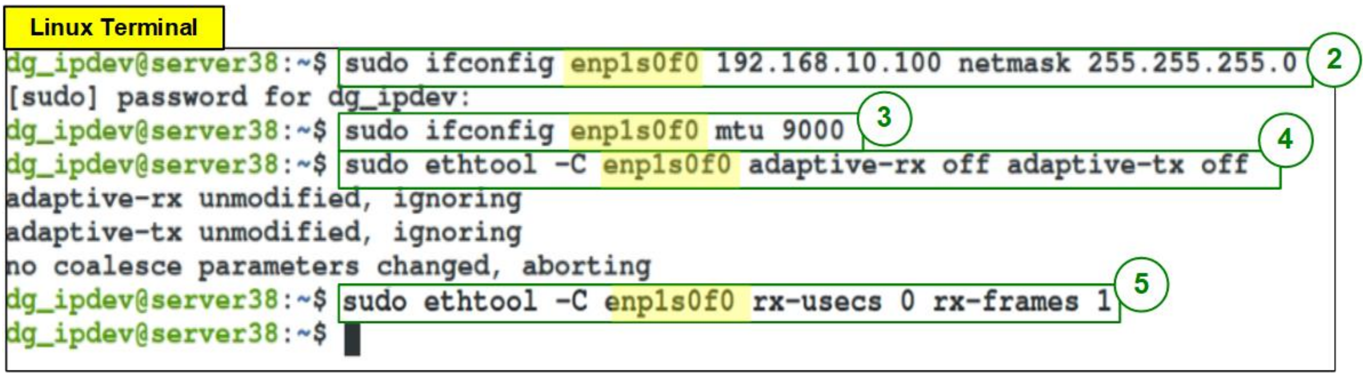
Figure 2-1 displays the output of this command. For example, “enp1s0f0” may be identified as the 10G Ethernet interface connected to the NVMe/TCP host.



♦ : Input by user
♦ : Output to user
Linux Terminal
1
Display a list of network connection
Logical name of 10Gb Ethernet connection

```
dg_ipdev@server38:~$ sudo lshw -C network
  *-network:0
       description: Ethernet interface
       product: 82599ES 10-Gigabit SFI/SFP+ Network Connection
       vendor: Intel Corporation
       physical id: 0
       bus info: pci@0000:01:00.0
       logical name: enp1s0f0
       version: 01
       serial: 90:e2:ba:6a:97:a8
       size: 10Gbit/s
       capacity: 10Gbit/s
       width: 64 bits
       clock: 33MHz
       capabilities: pm msi msix pciexpress bus_master cap_list et
```

Figure 2-1 Display logical name of 10G Ethernet port



```
dg_ipdev@server38:~$ sudo ifconfig enp1s0f0 192.168.10.100 netmask 255.255.255.0
[sudo] password for dg_ipdev:
dg_ipdev@server38:~$ sudo ifconfig enp1s0f0 mtu 9000
dg_ipdev@server38:~$ sudo ethtool -C enp1s0f0 adaptive-rx off adaptive-tx off
adaptive-rx unmodified, ignoring
adaptive-tx unmodified, ignoring
no coalesce parameters changed, aborting
dg_ipdev@server38:~$ sudo ethtool -C enp1s0f0 rx-usecs 0 rx-frames 1
dg_ipdev@server38:~$
```

Figure 2-2 IP address and Ethernet interface setting

2) Input "ifconfig <interface> <ipaddr_value> netmask <netmask_value>" to set the IP address and subnet mask for the desired Ethernet port:

```
sudo ifconfig enp1s0f0 192.168.10.100 netmask 255.255.255.0
```

The interface ('enp1s0f0') is set with an IP address of '192.168.10.100' and a subnet mask of '255.255.255.0'.

3) Input "ifconfig <interface > mtu <mtu_value>" to enable jumbo frames by setting the maximum transfer unit (MTU) to 9000 using the following command:

```
sudo ifconfig enp1s0f0 mtu 9000
```

4) Turn off Rx-Tx latency improvement algorithm to stabilize performance by "sudo ethtool -C <interface> adaptive-rx off adaptive-tx off".

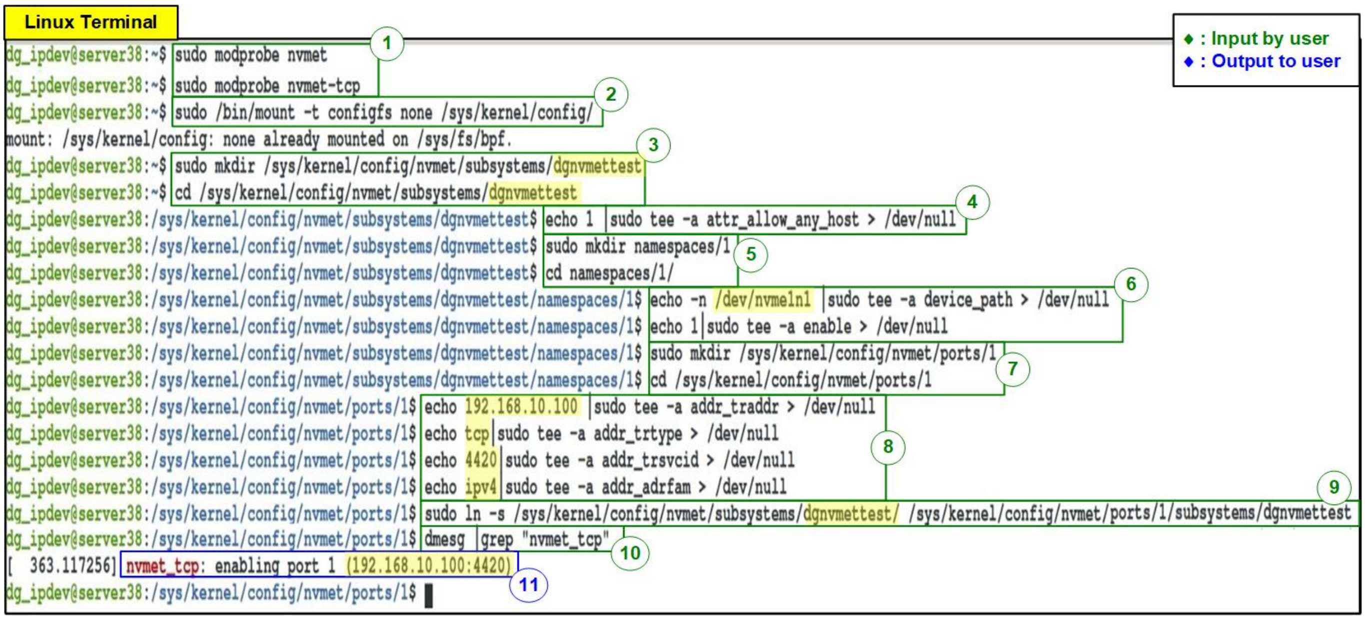
```
sudo ethtool -C enp1s0f0 adaptive-rx off adaptive-tx off
```

5) Set the highest rate of Rx interrupt to ensure the PC processes each received packet immediately by "sudo ethtool -C <interface> rx-usecs 0 rx-frames 1".

```
sudo ethtool -C enp1s0f0 rx-usecs 0 rx-frames 1
```

2.3 NVMe/TCP Target setting on PC

This section provides instructions on setting up the PC to serve as NVMe/TCP target, following the configuration of the 10G Ethernet network.



```
dg_ipdev@server38:~$ sudo modprobe nvmet
dg_ipdev@server38:~$ sudo modprobe nvmet-tcp
dg_ipdev@server38:~$ sudo /bin/mount -t configfs none /sys/kernel/config/
mount: /sys/kernel/config: none already mounted on /sys/fs/bpf.
dg_ipdev@server38:~$ sudo mkdir /sys/kernel/config/nvmet/subsystems/dgnvmettest
dg_ipdev@server38:~$ cd /sys/kernel/config/nvmet/subsystems/dgnvmettest
dg_ipdev@server38:/sys/kernel/config/nvmet/subsystems/dgnvmettest$ echo 1 |sudo tee -a attr_allow_any_host > /dev/null
dg_ipdev@server38:/sys/kernel/config/nvmet/subsystems/dgnvmettest$ sudo mkdir namespaces/1
dg_ipdev@server38:/sys/kernel/config/nvmet/subsystems/dgnvmettest$ cd namespaces/1/
dg_ipdev@server38:/sys/kernel/config/nvmet/subsystems/dgnvmettest/namespaces/1$ echo -n /dev/nvme1n1 |sudo tee -a device_path > /dev/null
dg_ipdev@server38:/sys/kernel/config/nvmet/subsystems/dgnvmettest/namespaces/1$ echo 1|sudo tee -a enable > /dev/null
dg_ipdev@server38:/sys/kernel/config/nvmet/subsystems/dgnvmettest/namespaces/1$ sudo mkdir /sys/kernel/config/nvmet/ports/1
dg_ipdev@server38:/sys/kernel/config/nvmet/subsystems/dgnvmettest/namespaces/1$ cd /sys/kernel/config/nvmet/ports/1
dg_ipdev@server38:/sys/kernel/config/nvmet/ports/1$ echo 192.168.10.100 |sudo tee -a addr_traddr > /dev/null
dg_ipdev@server38:/sys/kernel/config/nvmet/ports/1$ echo tcp|sudo tee -a addr_trtype > /dev/null
dg_ipdev@server38:/sys/kernel/config/nvmet/ports/1$ echo 4420|sudo tee -a addr_trsvcid > /dev/null
dg_ipdev@server38:/sys/kernel/config/nvmet/ports/1$ echo ipv4|sudo tee -a addr_adrfam > /dev/null
dg_ipdev@server38:/sys/kernel/config/nvmet/ports/1$ sudo ln -s /sys/kernel/config/nvmet/subsystems/dgnvmettest/ /sys/kernel/config/nvmet/ports/1/subsystems/dgnvmettest
dg_ipdev@server38:/sys/kernel/config/nvmet/ports/1$ dmesg |grep "nvmet_tcp"
[  363.117256] nvmet_tcp: enabling port 1 (192.168.10.100:4420)
dg_ipdev@server38:/sys/kernel/config/nvmet/ports/1$ █
```

Figure 2-3 Target setting

1) Load the NVMe/TCP Target module using the following command.

```
sudo modprobe nvmet
sudo modprobe nvmet-tcp
```

2) Mount the kernel user configuration filesystem to manage NVMe/TCP target settings.

```
sudo /bin/mount -t configfs none /sys/kernel/config/
```

3) Create an NVMe target subsystem and set its NVMe Qualified Name (NQN). The NQN here should match the ‘TrgNQN’ setting to the NVMeTCP10G-IP demo. Assumed that the NQN is configured by “dgnvmettest”, use following commands for changing directory to the subsystem directory.

```
sudo mkdir /sys/kernel/config/nvmet/subsystems/dgnvmettest
cd /sys/kernel/config/nvmet/subsystems/dgnvmettest
```

Note: While the ‘TrgNQN’ on the NVMeTCP10G-IP supports up to 256 characters, the demo system allows input up to 16 characters. Ensure the NQN length does not exceed this limit.

4) Set the attribute to allow any host to connect using following command. This is suitable for testing but should be restricted in production environments. The NVMeTCP10G-IP allows user to configure 'Host NQN' value.

```
echo 1 |sudo tee -a attr_allow_any_host > /dev/null
```

5) Create a namespace within the subsystem and change directory using following command.

```
sudo mkdir namespaces/1
cd namespaces/1/
```

6) Assign a local NVMe SSD to the system and enable it using following command.

```
echo -n /dev/nvme1n1 |sudo tee -a device_path > /dev/null
echo 1|sudo tee -a enable > /dev/null
```

Note: Replace 'nvme1n1' with the correct device identifier for your SSD.

7) Create an NVMe target port to export the created subsystem and change directory using following command.

```
sudo mkdir /sys/kernel/config/nvmet/ports/1
cd /sys/kernel/config/nvmet/ports/1
```

8) Configure Ethernet parameters for the NVMe target port, including

- IP address = 192.168.10.100
- Transport type = "tcp"
- Port number = 4420
- Address family = "ipv4"

```
echo 192.168.10.100 |sudo tee -a addr_traddr > /dev/null
echo tcp|sudo tee -a addr_trtype > /dev/null
echo 4420|sudo tee -a addr_trsvcid > /dev/null
echo ipv4|sudo tee -a addr_adrfam > /dev/null
```

Note: IP address value corresponds to configured value of Ethernet card port in section 2.2 (Ethernet Interface Setting on PC).

9) Create a symbolic link between the target port and the subsystem using following command.

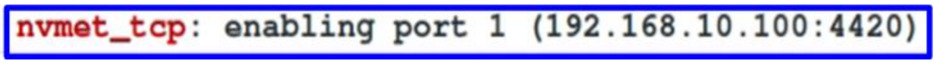
```
sudo ln -s /sys/kernel/config/nvmet/subsystems/dgnvmettest/
/sys/kernel/config/nvmet/ports/1/subsystems/dgnvmettest
```

Note: The Target NQN (dgnvmetest) must match the value configured in steps (step 3).

10) Check that the target is correctly configured by inspecting the system messages using following command.

```
dmesg |grep "nvmet_tcp"
```

11) Successful configuration messages will display the target IP address and port number.

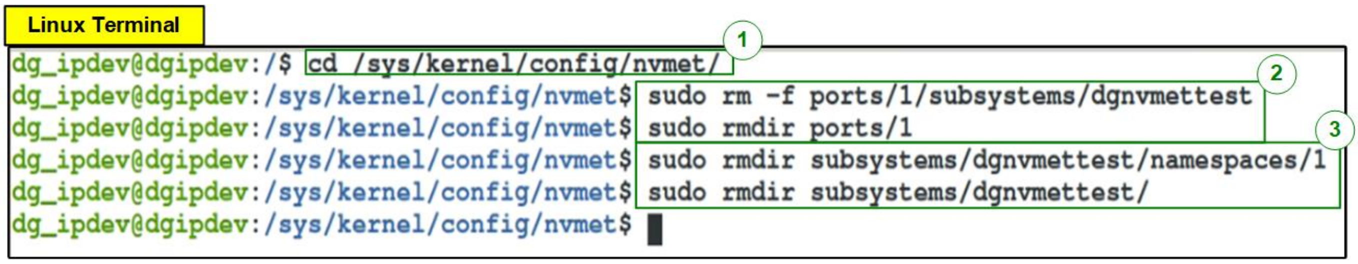


```
nvmet_tcp: enabling port 1 (192.168.10.100:4420)
```

Figure 2-4 NVMe/TCP target setup success message

2.4 Removing NVMe/TCP Target on PC

This section provides step-by-step instructions on how to dismantle the NVMe/TCP target configuration on you PC after completing tests.



```
dg_ipdev@dgipdev:/$ cd /sys/kernel/config/nvmet/
dg_ipdev@dgipdev:/sys/kernel/config/nvmet$ sudo rm -f ports/1/subsystems/dgnvmettest
dg_ipdev@dgipdev:/sys/kernel/config/nvmet$ sudo rmdir ports/1
dg_ipdev@dgipdev:/sys/kernel/config/nvmet$ sudo rmdir subsystems/dgnvmettest/namespaces/1
dg_ipdev@dgipdev:/sys/kernel/config/nvmet$ sudo rmdir subsystems/dgnvmettest/
dg_ipdev@dgipdev:/sys/kernel/config/nvmet$ 
```

Figure 2-5 Remove NVMe/TCP Target

1) Change the working directory to where NVMe/TCP configurations are stored.

```
cd sys/kernel/config/nvmet/
```

2) Delete the directory corresponding to the target port to remove its connections.

```
sudo rm -f ports/1/subsystems/dgnvmettest
sudo rmdir ports/1
```

3) Remove the subsystem linked to the NVMe SSD.

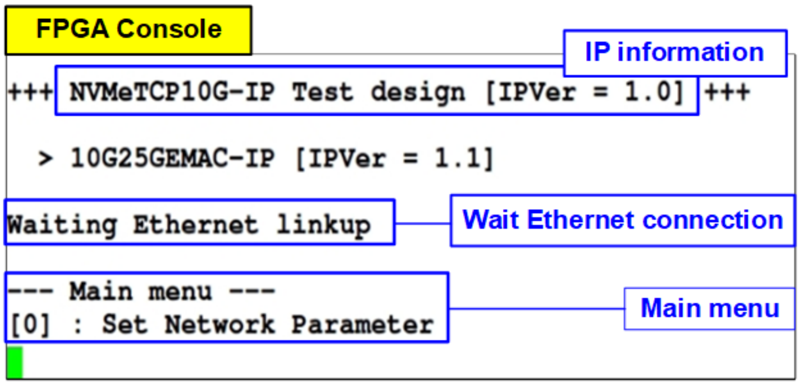
```
sudo rmdir subsystems/dgnvmettest/namespace/1
sudo rmdir subsystems/dgnvmettest/
```

Once the target subsystem is successfully removed, the NVMe SSD should be accessible by the general NVMe device driver on the PC. Use “hexdump” or other tools to inspect the SSD and verify the written data.

By following these steps, you can ensure that the NVMe/TCP target configuration is removed, and the NVMe SSD is returned to its standard operational state for regular PC usage.

3 Test operation

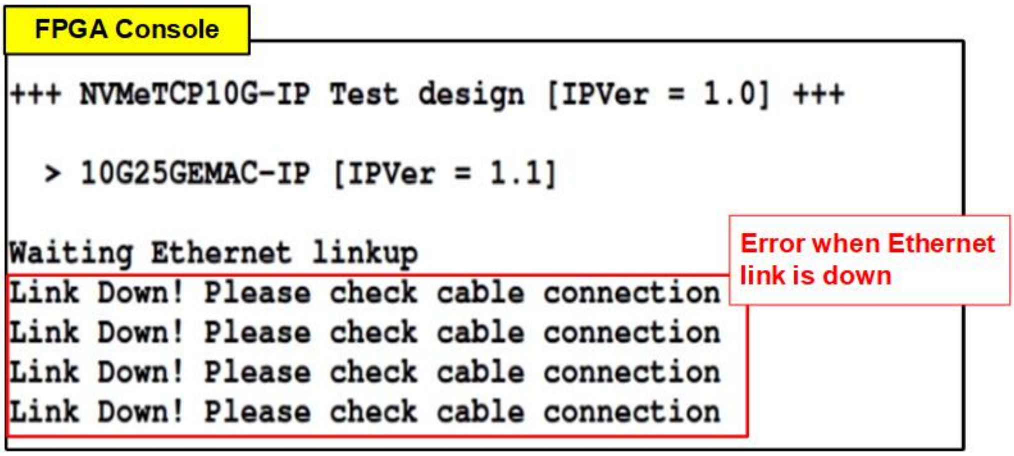
Upon the completion of both PC setup and FPGA setup, the FPGA console displays a welcome screen, indicating readiness for operations. This is confirmed by the display of the IP name, IP version number, and the status of the Ethernet connection.



```
+++ NVMeTCP10G-IP Test design [IPVer = 1.0] +++

  > 10G25GEMAC-IP [IPVer = 1.1]

Waiting Ethernet linkup

--- Main menu ---
[0] : Set Network Parameter
```

Figure 3-1 Message after system boot-up

- IP Details: The name and version of the NVMeTCP10G-IP are shown. Additionally, the IP version of DG EMAC-IP is also displayed if it is utilized.
- Ethernet Status: The connection status of the FPGA's Ethernet link is displayed. If the Ethernet link fails to establish, an error message prompting a check of the Ethernet cable appears.



```
+++ NVMeTCP10G-IP Test design [IPVer = 1.0] +++

  > 10G25GEMAC-IP [IPVer = 1.1]

Waiting Ethernet linkup
Link Down! Please check cable connection
Link Down! Please check cable connection
Link Down! Please check cable connection
Link Down! Please check cable connection
```

Figure 3-2 Error message when Ethernet connection is down

The main menu appears once the Ethernet link is established. This menu includes several options, each corresponding to different stages of the test operation. Initially, only the relevant menu items are made available to ensure the correct sequence of operations is followed. The first action after boot-up is Menu [0] for setting network parameters.

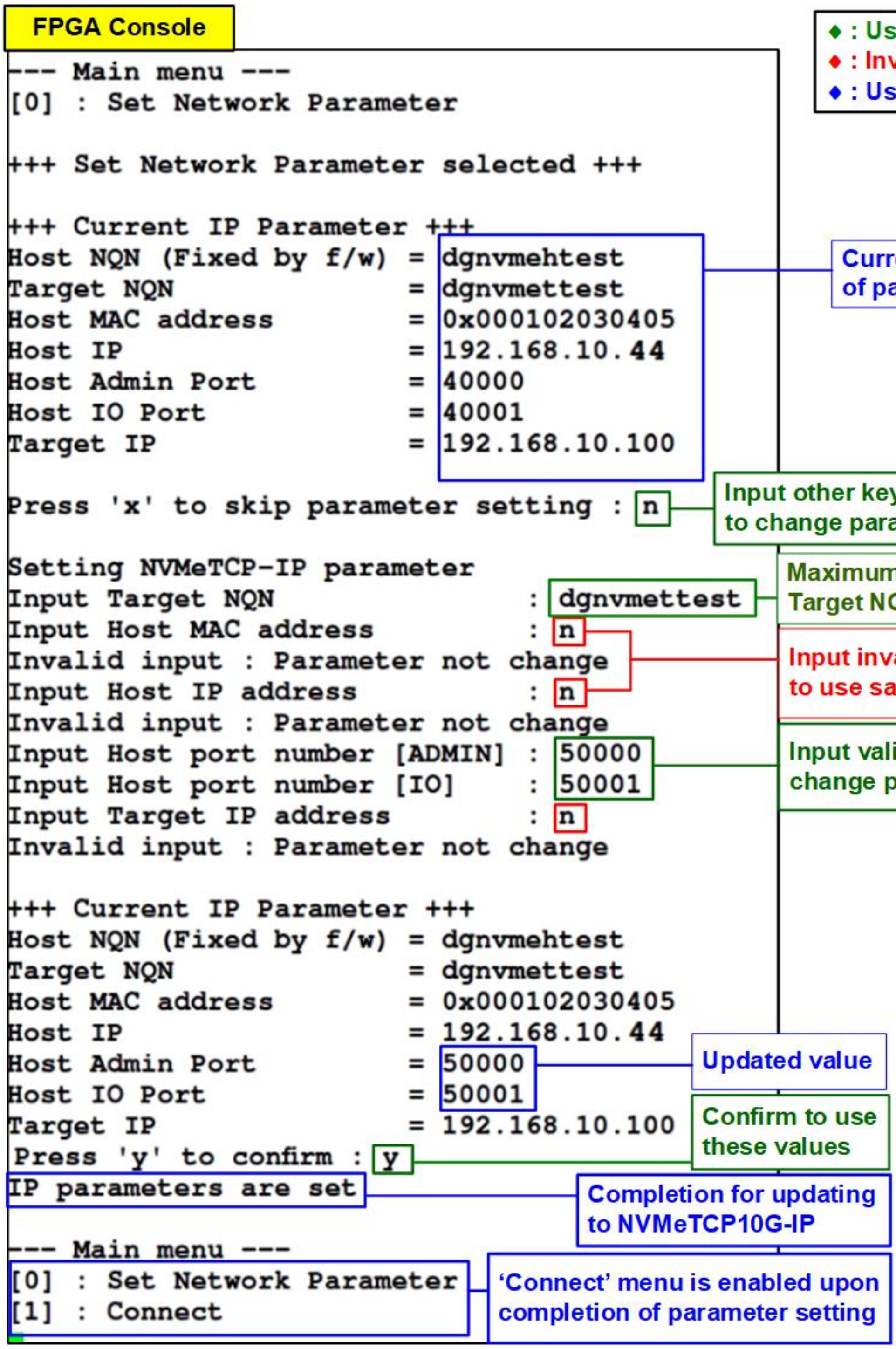
3.1 Set Network Parameter

Select ‘0’ from the main menu to configure IP parameters. This action initiates the process to set or update the network settings required for communication between the host and target. Initially, the current parameters are shown on the console. The user enters ‘x’ to continue with the existing setting, or enters any other key to begin altering the parameters. While setting each parameter, the parameter’s value is not updated if the user enters invalid value. Below are the lists of input parameters.

1) Target NVMe Qualifed Name (NQN): Identifies the target SSD. Accepts up to 16 characters and the default is “dgnvmettest”.
2) Host MAC Address: Requires a 12-digit hexadecimal value for the host by adding prefix with “0x”. Default is 0x000102030405.
3) Host IP Address: Sets four decimal digits separated by dots. Each segment ranges from 0-255. This value is assigned to be the IP address of the host. Default is 192.168.10.1.
4) Host Port Number: Inputs the Admin and IP ports for the host with valid inputs ranging from 0-65535. Defaults are 40000 (Admin) and 40001 (IO).
5) Target IP address: Same format as the Host IP. This value is assigned to be the IP address of the target. Default is 192.168.10.100.

After all parameters are set, they are displayed for user confirmation. If the user confirms the settings by entering ‘y’, the parameters are configured on the NVMeTCP10G-IP. A confirmation message, “IP parameters are set”, is shown upon successful configuration.

The main menu reappears with offering the ‘Connect’ command as the next step, as detailed in Figure 3-3.



FPGA Console

```
--- Main menu ---
[0] : Set Network Parameter

+++ Set Network Parameter selected +++

+++ Current IP Parameter +++
Host NQN (Fixed by f/w) = dgnvmehtest
Target NQN              = dgnvmettest
Host MAC address        = 0x000102030405
Host IP                 = 192.168.10.44
Host Admin Port         = 40000
Host IO Port            = 40001
Target IP               = 192.168.10.100

Press 'x' to skip parameter setting : n

Setting NVMeTCP-IP parameter
Input Target NQN                  : dgnvmettest
Input Host MAC address            : n
Invalid input : Parameter not change
Input Host IP address             : n
Invalid input : Parameter not change
Input Host port number [ADMIN]  : 50000
Input Host port number [IO]     : 50001
Input Target IP address           : n
Invalid input : Parameter not change

+++ Current IP Parameter +++
Host NQN (Fixed by f/w) = dgnvmehtest
Target NQN              = dgnvmettest
Host MAC address        = 0x000102030405
Host IP                 = 192.168.10.44
Host Admin Port         = 50000
Host IO Port            = 50001
Target IP               = 192.168.10.100
 Press 'y' to confirm : y
IP parameters are set

--- Main menu ---
[0] : Set Network Parameter
[1] : Connect
```

- ◆ : User Input
- ◆ : Invalid Input
- ◆ : User Output

Current value of parameters

Input other keys (not ‘x’) to change parameter

Maximum chars of Target NQN is 16

Input invalid value to use same value

Input valid value to change parameter

Updated value

Confirm to use these values

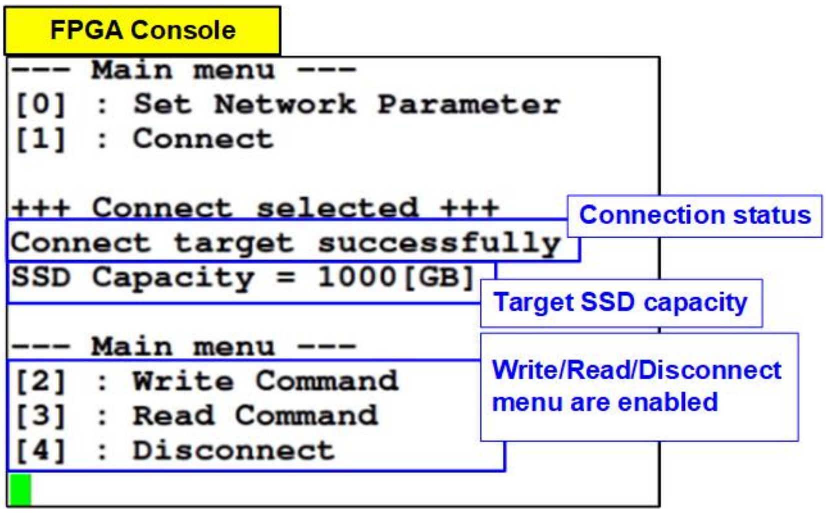
Completion for updating to NVMeTCP10G-IP

‘Connect’ menu is enabled upon completion of parameter setting

Figure 3-3 Set Network Parameter result

3.2 Connect

Select ‘1’ from the main menu to establish the connection between the host and the target.



FPGA Console

```
--- Main menu ---
[0] : Set Network Parameter
[1] : Connect

+++ Connect selected +++
Connect target successfully
SSD Capacity = 1000[GB]

--- Main menu ---
[2] : Write Command
[3] : Read Command
[4] : Disconnect
```

Connection status

Target SSD capacity

Write/Read/Disconnect menu are enabled

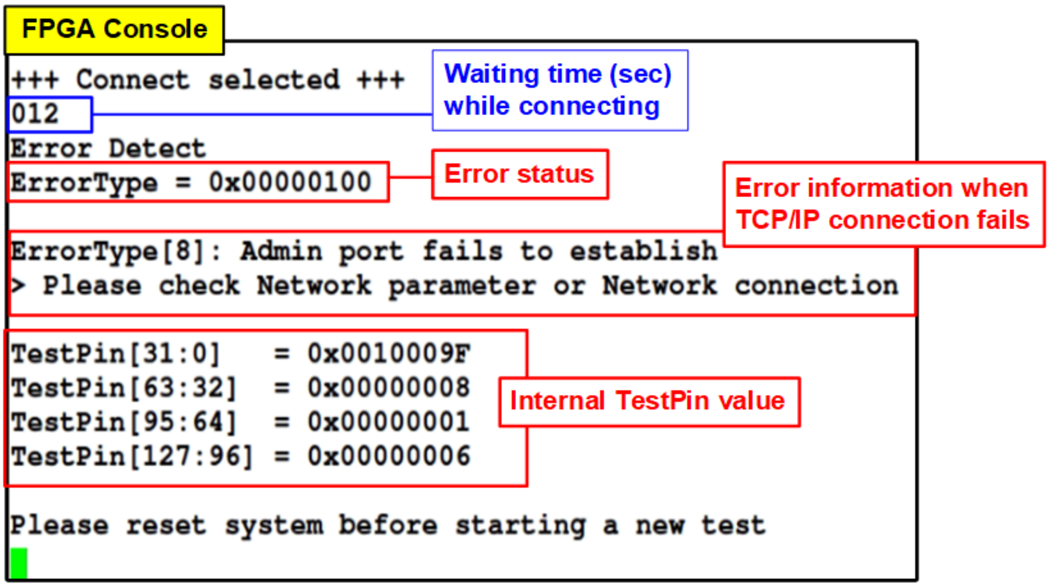
Figure 3-4 Console result when Connect succeeded

The NVMeTCP10G-IP (the host) established a TCP/IP connection using the network parameters configured in Section 3.1 (Set Network Parameter). Following successful TCP/IP setup, the NVMe/TCP connection is established using the specified Target NQN. This ensures that the host is correctly linked to the designated target NVMe SSD.

Upon successfully connecting, the console displays “Connect target successfully” along with the capacity of the target NVMe SSD. The menu then updates to include options for Write, Read, and Disconnect commands, as illustrated in Figure 3-4.

If the connecting processes takes longer than expected, the waiting time is updated and displayed every second. In cases where TCP/IP initialization fails or the host cannot connect to the target, the error status and detailed error information are shown. Additionally, the TestPin (Internal Test pin of the IP) is displayed.

If an error occurs, it is necessary to reset the system before attempting a new test. This requirement is shown in both Figure 3-5 and Figure 3-6.



```
+++ Connect selected +++
012
Error Detect
ErrorType = 0x00000100

ErrorType[8]: Admin port fails to establish
> Please check Network parameter or Network connection

TestPin[31:0]   = 0x0010009F
TestPin[63:32]  = 0x00000008
TestPin[95:64]  = 0x00000001
TestPin[127:96] = 0x00000006

Please reset system before starting a new test
```

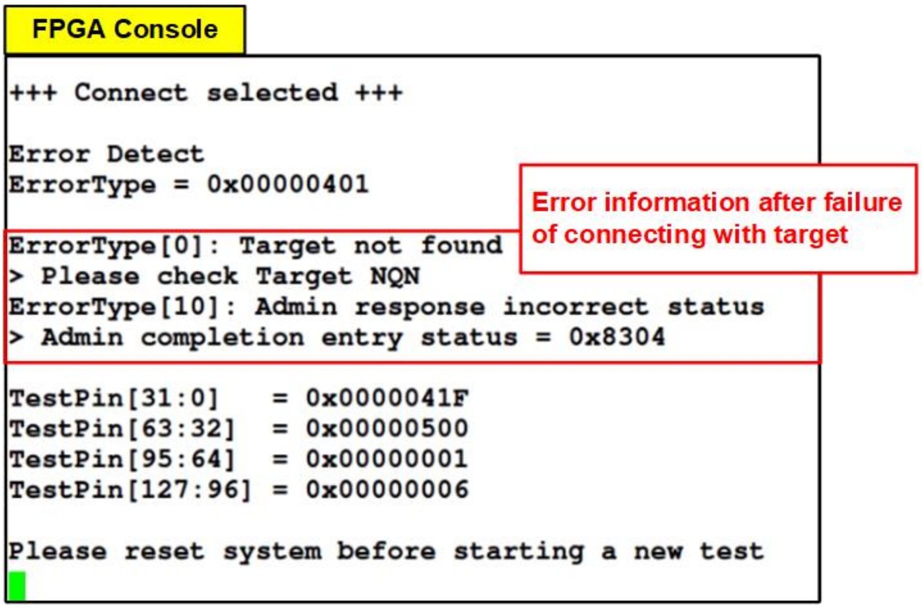
Waiting time (sec) while connecting

Error status

Error information when TCP/IP connection fails

Internal TestPin value

Figure 3-5 Console result when TCP/IP initialization fails



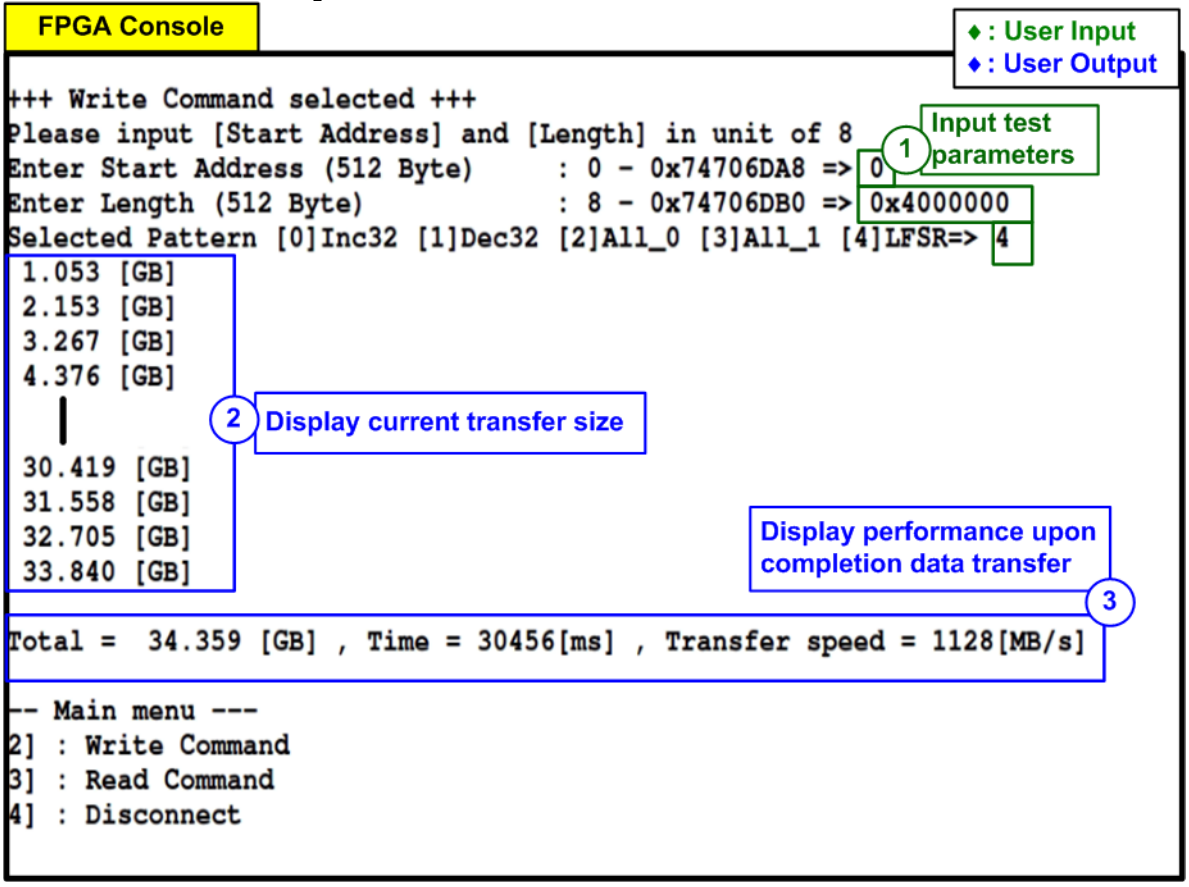
```
+++ Connect selected +++

Error Detect
ErrorType = 0x00000401

ErrorType[0]: Target not found
> Please check Target NQN
ErrorType[10]: Admin response incorrect status
> Admin completion entry status = 0x8304

TestPin[31:0]   = 0x0000041F
TestPin[63:32]  = 0x00000500
TestPin[95:64]  = 0x00000001
TestPin[127:96] = 0x00000006

Please reset system before starting a new test
```

Error information after failure of connecting with target

Figure 3-6 Console result when the host fails to connect with the target

3.3 Write Command

Select ‘2’ from the main menu to perform a write operation to the target NVMe SSD. This command tests the host’s ability to send Write commands and pattern data across Ethernet to write data on the target NVMe SSD.


FPGA Console
♦ : User Input
♦ : User Output
1
Input test parameters
2
Display current transfer size
Display performance upon completion data transfer
3

```
+++ Write Command selected +++
Please input [Start Address] and [Length] in unit of 8
Enter Start Address (512 Byte)      : 0 - 0x74706DA8 => 0
Enter Length (512 Byte)             : 8 - 0x74706DB0 => 0x4000000
Selected Pattern [0]Inc32 [1]Dec32 [2]All_0 [3]All_1 [4]LFSR=> 4
 1.053 [GB]
 2.153 [GB]
 3.267 [GB]
 4.376 [GB]
 30.419 [GB]
 31.558 [GB]
 32.705 [GB]
 33.840 [GB]
Total =  34.359 [GB] , Time = 30456[ms] , Transfer speed = 1128[MB/s]
-- Main menu ---
2] : Write Command
3] : Read Command
4] : Disconnect
```

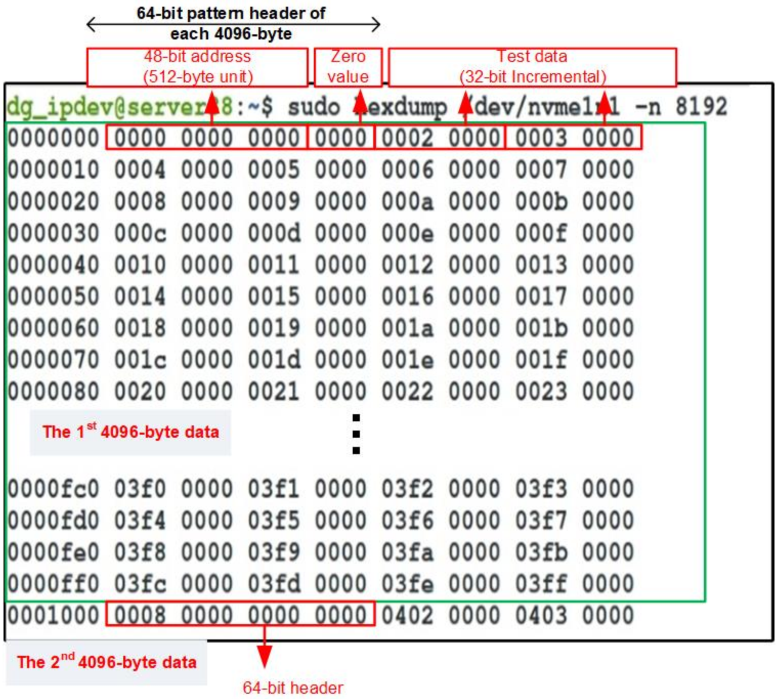
Figure 3-7 Test result when running Write command

User inputs three parameters as follows.

1) Start Address: Input the start address where the write operation begins on the target NVMe SSD, specified in 512-byte units. Enter the address in decimal form or use the “0x” prefix for hexadecimal values. Ensure the address aligns to 4 KB (8 x 512 bytes) as this is the fixed data length for one Write command.
2) Transfer Length: Input the total size to be written in 512-byte units. Like the start address, this can be entered in decimal or hexadecimal (with “0x” prefix) and must align to 4 KB.
3) Test pattern: Choose a data pattern for writing. Available options include 32-bit incremental, 32-bit decremental, all 0, all 1, and 32-bit LFSR counter.
   Note: SSDs may exhibit optimal performance with the all zeros pattern due to data compression algorithms inside the SSD controller.

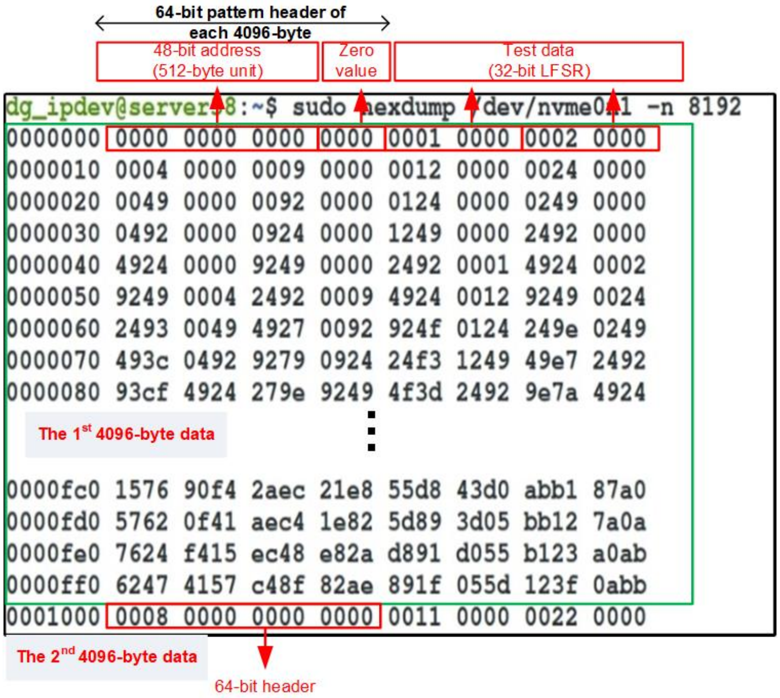
Upon validating all inputs, the write process commences. The current amount of transferred data is displayed every second on the console to confirm ongoing operation. Upon completion, the console displays the total size transferred, total time used, and test speed.

For the incremental, decremental, and LFSR patterns, each 4 KB data block includes a unique 64-bit header consisting of a 48-bit address (in 512-byte units) followed by 6 bits set to zero. The remainder of the data block is filled with the chosen test pattern. Figure 3-8 and Figure 3-9 illustrated examples using the 32-bit incremental and LFSR pattern, respectively. The unique header is omitted when using all-zero or all-one patterns.


64-bit pattern header of each 4096-byte
48-bit address (512-byte unit)
Zero value
Test data (32-bit Incremental)
The 1st 4096-byte data
The 2nd 4096-byte data
64-bit header

```
dg_ipdev@server38:~$ sudo hexdump /dev/nvme1n1 -n 8192
0000000 0000 0000 0000 0000 0002 0000 0003 0000
0000010 0004 0000 0005 0000 0006 0000 0007 0000
0000020 0008 0000 0009 0000 000a 0000 000b 0000
0000030 000c 0000 000d 0000 000e 0000 000f 0000
0000040 0010 0000 0011 0000 0012 0000 0013 0000
0000050 0014 0000 0015 0000 0016 0000 0017 0000
0000060 0018 0000 0019 0000 001a 0000 001b 0000
0000070 001c 0000 001d 0000 001e 0000 001f 0000
0000080 0020 0000 0021 0000 0022 0000 0023 0000
                            ⋮
0000fc0 03f0 0000 03f1 0000 03f2 0000 03f3 0000
0000fd0 03f4 0000 03f5 0000 03f6 0000 03f7 0000
0000fe0 03f8 0000 03f9 0000 03fa 0000 03fb 0000
0000ff0 03fc 0000 03fd 0000 03fe 0000 03ff 0000
0001000 0008 0000 0000 0000 0402 0000 0403 0000
```

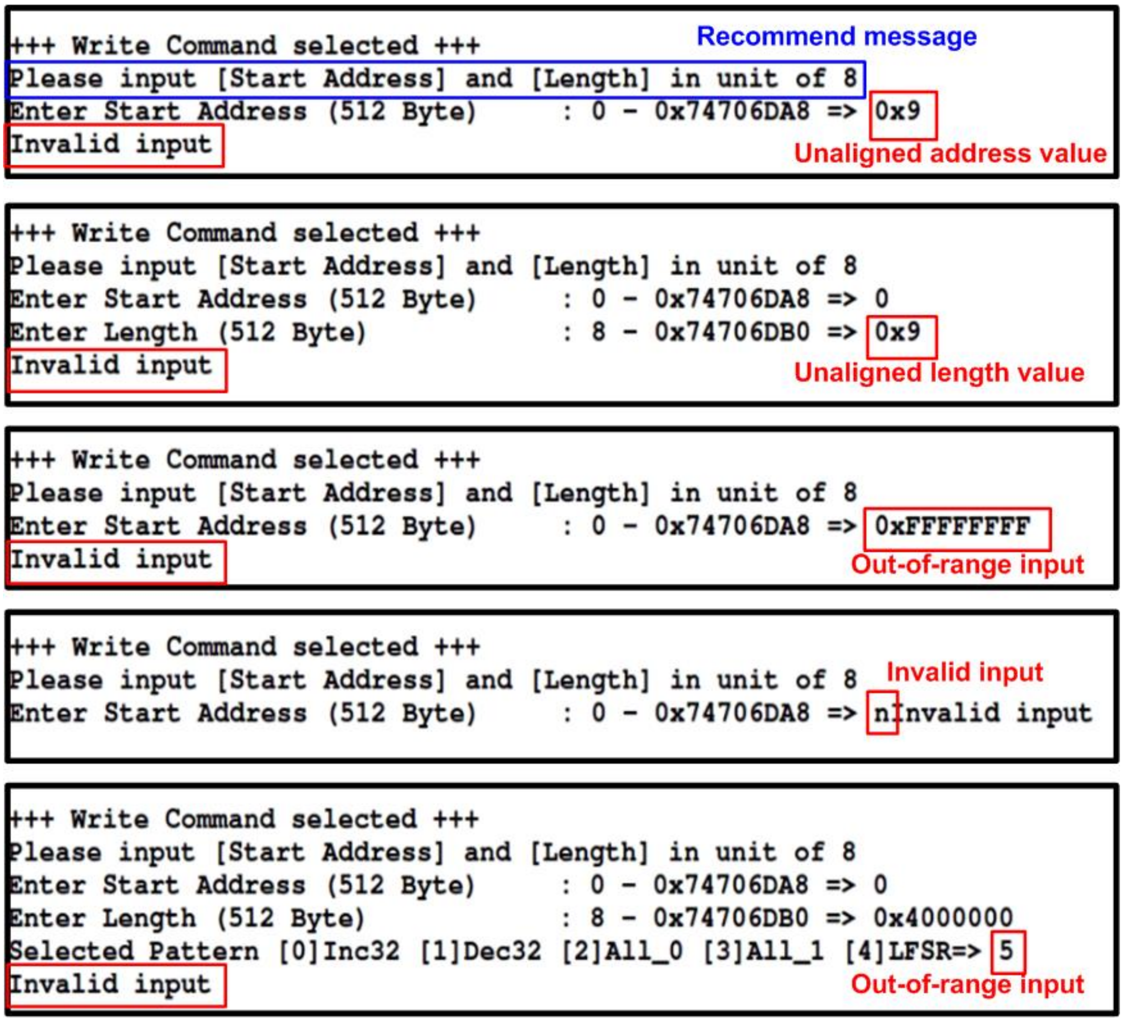
Figure 3-8 Example Test data of the 1st and 2nd 4KB data block using incremental pattern



```
dg_ipdev@server38:~$ sudo hexdump /dev/nvme0n1 -n 8192
0000000 0000 0000 0000 0000 0001 0000 0002 0000
0000010 0004 0000 0009 0000 0012 0000 0024 0000
0000020 0049 0000 0092 0000 0124 0000 0249 0000
0000030 0492 0000 0924 0000 1249 0000 2492 0000
0000040 4924 0000 9249 0000 2492 0001 4924 0002
0000050 9249 0004 2492 0009 4924 0012 9249 0024
0000060 2493 0049 4927 0092 924f 0124 249e 0249
0000070 493c 0492 9279 0924 24f3 1249 49e7 2492
0000080 93cf 4924 279e 9249 4f3d 2492 9e7a 4924
                            ⋮
0000fc0 1576 90f4 2aec 21e8 55d8 43d0 abb1 87a0
0000fd0 5762 0f41 aec4 1e82 5d89 3d05 bb12 7a0a
0000fe0 7624 f415 ec48 e82a d891 d055 b123 a0ab
0000ff0 6247 4157 c48f 82ae 891f 055d 123f 0abb
0001000 0008 0000 0000 0000 0011 0000 0022 0000
```

Figure 3-9 Example Test data of the 1st and 2nd 4 KB data block using LFSR pattern

If invalid inputs are detected, the console displays an “Invalid input” message, and the operation is cancelled, as depicted in Figure 3-10.



```
+++ Write Command selected +++
Please input [Start Address] and [Length] in unit of 8
Enter Start Address (512 Byte)      : 0 - 0x74706DA8 => 0x9
Invalid input
```

```
+++ Write Command selected +++
Please input [Start Address] and [Length] in unit of 8
Enter Start Address (512 Byte)      : 0 - 0x74706DA8 => 0
Enter Length (512 Byte)             : 8 - 0x74706DB0 => 0x9
Invalid input
```

```
+++ Write Command selected +++
Please input [Start Address] and [Length] in unit of 8
Enter Start Address (512 Byte)      : 0 - 0x74706DA8 => 0xFFFFFFFF
Invalid input
```

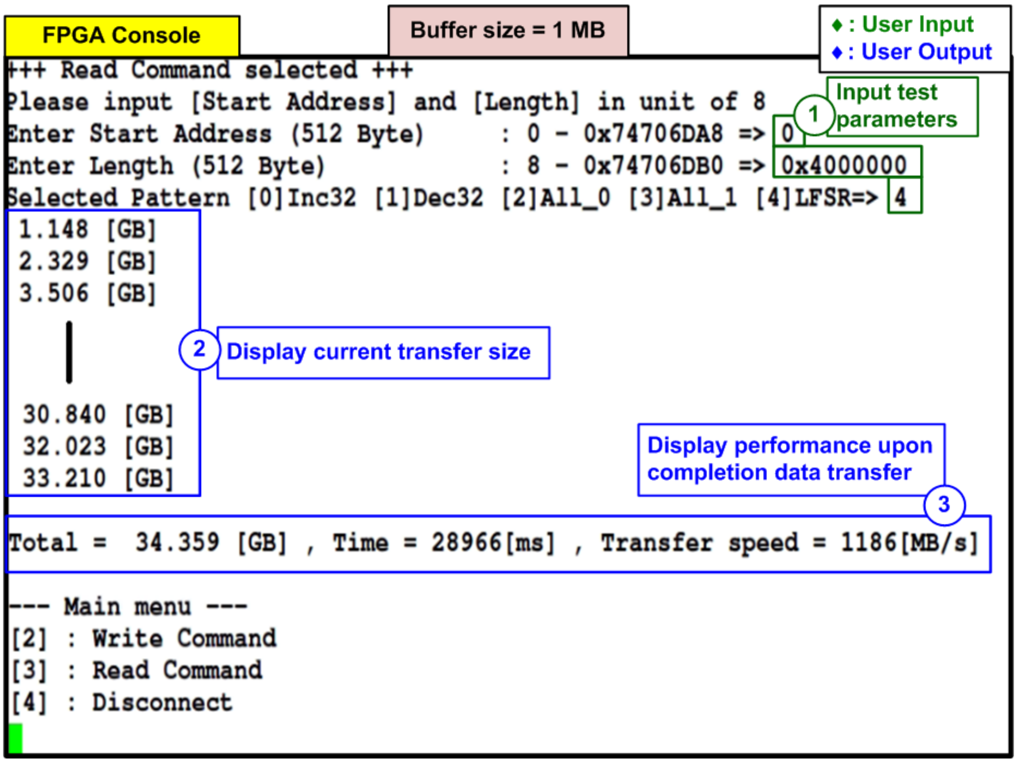
```
+++ Write Command selected +++
Please input [Start Address] and [Length] in unit of 8
Enter Start Address (512 Byte)      : 0 - 0x74706DA8 => nInvalid input
```

```
+++ Write Command selected +++
Please input [Start Address] and [Length] in unit of 8
Enter Start Address (512 Byte)      : 0 - 0x74706DA8 => 0
Enter Length (512 Byte)             : 8 - 0x74706DB0 => 0x4000000
Selected Pattern [0]Inc32 [1]Dec32 [2]All_0 [3]All_1 [4]LFSR=> 5
Invalid input
```

Figure 3-10 Error message when the input is unaligned, out-of-range, or invalid

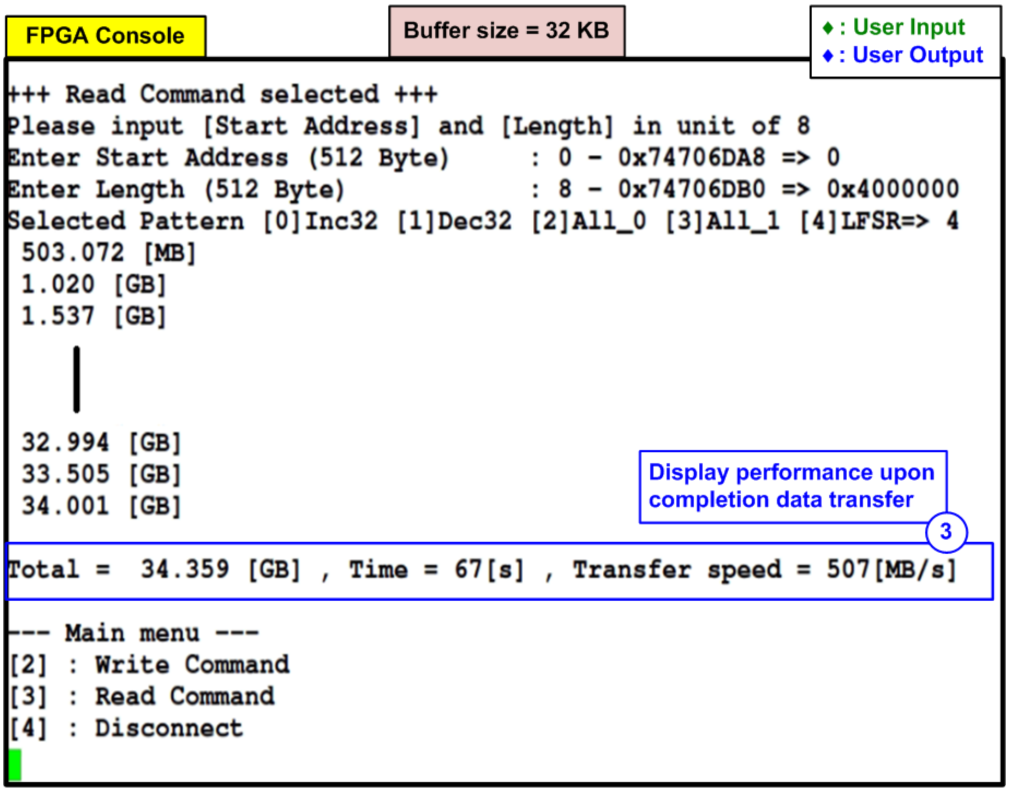
3.4 Read Command

Select ‘3’ to initiate a Read operation from the target NVMe SSD. This menu facilitates the testing of read operations by sending a Read command across Ethernet to the target. Following the command, the host awaits the data read from the target NVMe SSD.



```
+++ Read Command selected +++
Please input [Start Address] and [Length] in unit of 8
Enter Start Address (512 Byte)      : 0 - 0x74706DA8 => 0
Enter Length (512 Byte)             : 8 - 0x74706DB0 => 0x4000000
Selected Pattern [0]Inc32 [1]Dec32 [2]All_0 [3]All_1 [4]LFSR=> 4
 1.148 [GB]
 2.329 [GB]
 3.506 [GB]
    |
 30.840 [GB]
 32.023 [GB]
 33.210 [GB]
Total =  34.359 [GB] , Time = 28966[ms] , Transfer speed = 1186[MB/s]

--- Main menu ---
[2] : Write Command
[3] : Read Command
[4] : Disconnect
```

Figure 3-11 Test result when running Read command with 1-MB read buffer size



```
+++ Read Command selected +++
Please input [Start Address] and [Length] in unit of 8
Enter Start Address (512 Byte)      : 0 - 0x74706DA8 => 0
Enter Length (512 Byte)             : 8 - 0x74706DB0 => 0x4000000
Selected Pattern [0]Inc32 [1]Dec32 [2]All_0 [3]All_1 [4]LFSR=> 4
 503.072 [MB]
 1.020 [GB]
 1.537 [GB]
    |
 32.994 [GB]
 33.505 [GB]
 34.001 [GB]
Total =  34.359 [GB] , Time = 67[s] , Transfer speed = 507[MB/s]

--- Main menu ---
[2] : Write Command
[3] : Read Command
[4] : Disconnect
```

Figure 3-12 Test result when running Read command with 32-KB read buffer size

The user inputs three parameters as follows.

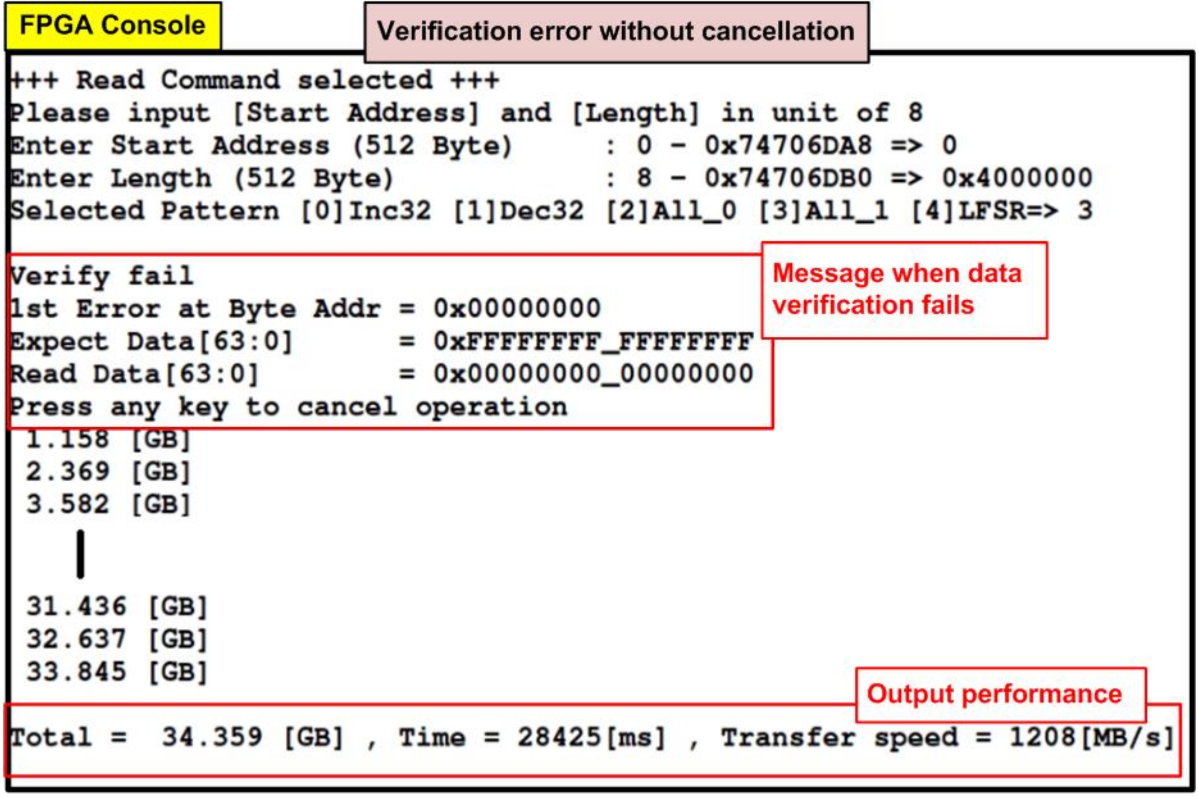
1) Start Address: Input the start address where the read operation begins on the target NVMe SSD, specified in 512-byte units. Enter the address in decimal form or use the "0x" prefix for hexadecimal values. Ensure the address aligns to 4 KB (8 x 512 bytes) as this is the fixed data length for one Read command.
2) Transfer Length: Input the total size to be read in 512-byte units. Like the start address, this can be entered in decimal or hexadecimal (with "0x" prefix) and must align to 4 KB.
3) Test pattern: Choose a data pattern to verify the data received from the target. Ensure the test pattern matches the one used during the Write command. Options include 32-bit incremental, 32-bit decremental, all 0, all 1, and 32-bit LFSR counter.

Once all inputs are validated, the read process begins. Current transfer data amounts are displayed every second on the console to indicate ongoing operation. Upon completion, the console displays the total size transferred, total time used, and test speed.

According to the NVMeTCP10G-IP specification, read performance varies with the read buffer size (illustrated in Figure 3-11 and Figure 3-12). Larger read buffer sizes may enhance read performance.

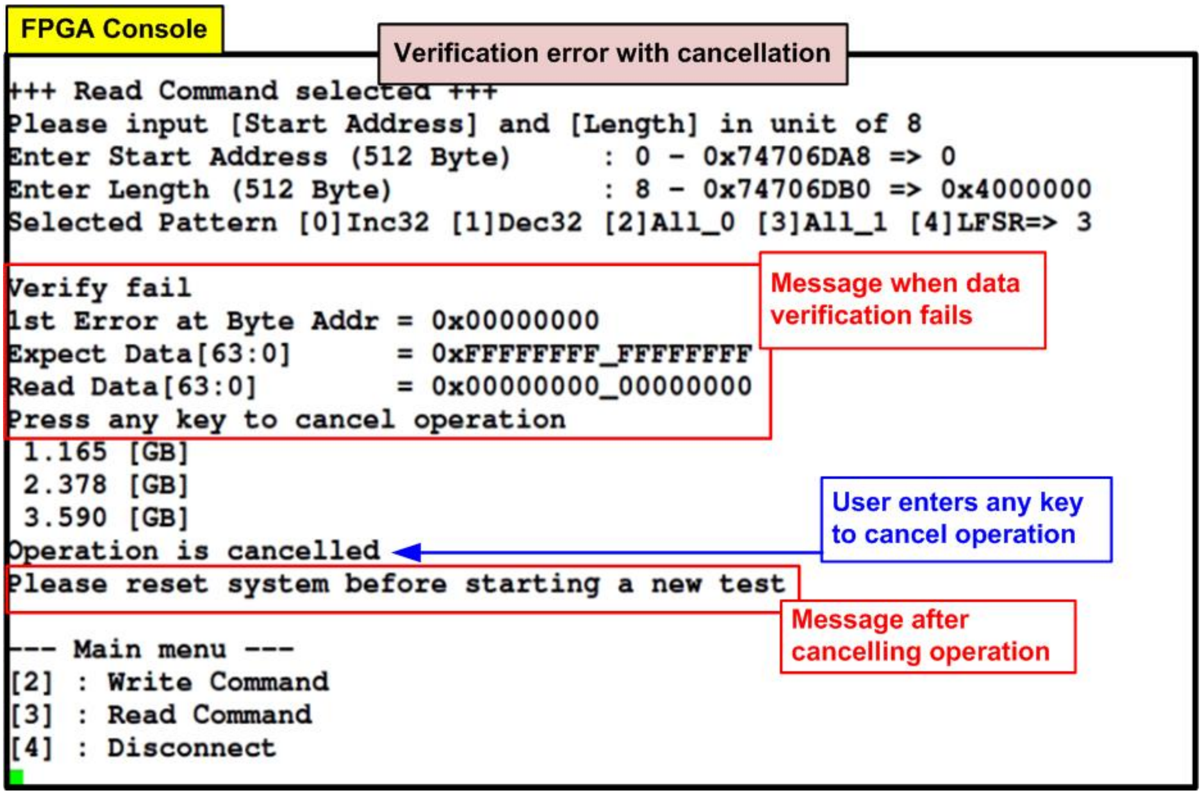
If data verification fails, an error message is displayed, detailing the first failure data including error byte address, expected value, and the read value (shown in Figure 3-13 and Figure 3-14). If verification fails, users can press any key to cancel the operation.

Pressing any key during the read operation will cancel it, although the command may still run as a background process. This can result in an incomplete sequence, and it is recommended to reset the FPGA board afterward. Additionally, the port connection on the target may not close properly; thus, changing the Admin/IO port number before reconnecting is advised.


FPGA Console
Verification error without cancellation
Message when data verification fails
Output performance

```
+++ Read Command selected +++
Please input [Start Address] and [Length] in unit of 8
Enter Start Address (512 Byte)      : 0 - 0x74706DA8 => 0
Enter Length (512 Byte)             : 8 - 0x74706DB0 => 0x4000000
Selected Pattern [0]Inc32 [1]Dec32 [2]All_0 [3]All_1 [4]LFSR=> 3

Verify fail
1st Error at Byte Addr = 0x00000000
Expect Data[63:0]      = 0xFFFFFFFF_FFFFFFFF
Read Data[63:0]        = 0x00000000_00000000
Press any key to cancel operation
 1.158 [GB]
 2.369 [GB]
 3.582 [GB]
    |

 31.436 [GB]
 32.637 [GB]
 33.845 [GB]

Total =  34.359 [GB] , Time = 28425[ms] , Transfer speed = 1208[MB/s]
```

Figure 3-13 Test result when data verification fails without cancellation

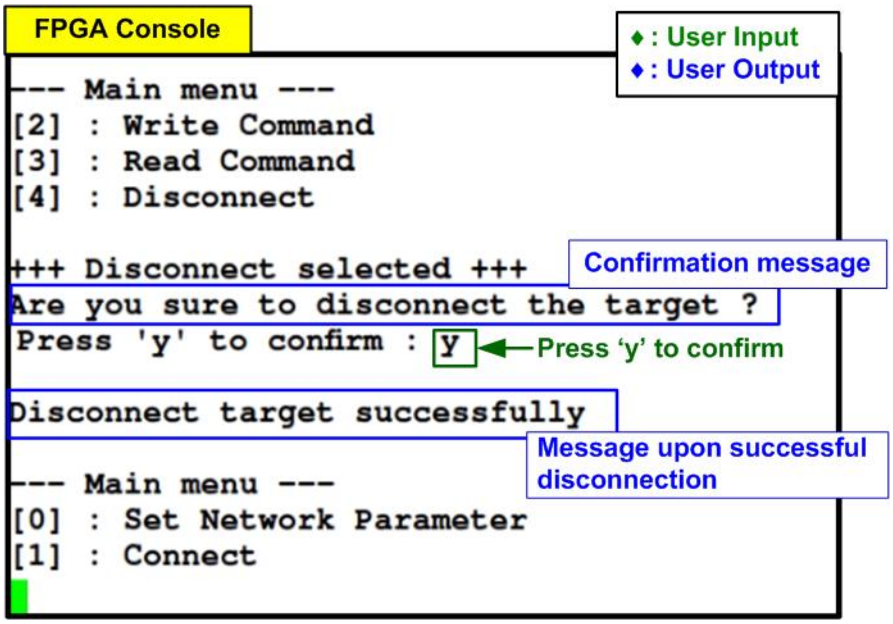

FPGA Console
Verification error with cancellation
Message when data verification fails
User enters any key to cancel operation
Message after cancelling operation

```
+++ Read Command selected +++
Please input [Start Address] and [Length] in unit of 8
Enter Start Address (512 Byte)      : 0 - 0x74706DA8 => 0
Enter Length (512 Byte)             : 8 - 0x74706DB0 => 0x4000000
Selected Pattern [0]Inc32 [1]Dec32 [2]All_0 [3]All_1 [4]LFSR=> 3

Verify fail
1st Error at Byte Addr = 0x00000000
Expect Data[63:0]      = 0xFFFFFFFF_FFFFFFFF
Read Data[63:0]        = 0x00000000_00000000
Press any key to cancel operation
 1.165 [GB]
 2.378 [GB]
 3.590 [GB]
Operation is cancelled
Please reset system before starting a new test

--- Main menu ---
[2] : Write Command
[3] : Read Command
[4] : Disconnect
```

Figure 3-14 Test result when data verification fails with cancellation

3.5 Disconnect Command

Select ‘4’ to disconnect the host from the target. This menu terminates the connection between the NVMeTCP10G-IP (host) and the target, previously established in Section 3.2 (Connect Command). Upon selecting Disconnect, a confirmation prompt appears on the console. Respond with ‘y’ to proceed with the disconnection or press any other key to cancel the operation.



FPGA Console

♦ : User Input
♦ : User Output

```
--- Main menu ---
[2] : Write Command
[3] : Read Command
[4] : Disconnect

+++ Disconnect selected +++
Are you sure to disconnect the target ?
Press 'y' to confirm : y

Disconnect target successfully

--- Main menu ---
[0] : Set Network Parameter
[1] : Connect
```

Confirmation message

Press ‘y’ to confirm

Message upon successful disconnection

Figure 3-15 Console result when running Disconnect command

Following a successful disconnection, the message “Disconnect target successful” will be displayed on the console. This indicates that the TCP/IP and NVMe/TCP connections have been successfully terminated.

After disconnecting, the user has the option to

- Reset network parameters via the ‘Set Network Parameter’ menu for adjustments or setting up a new target.
- Re-establish the connection with the same target using the ‘Connect’ command.

4 Revision History

Revision	Date	Description
1.02	29-Apr-24	Add table of content
1.01	25-Mar-22	Add topic 2.4 NVMe/TCP target removing on TestPC
1.00	3-Nov-21	Initial version release