

2-Ch RAID0 (NVMe-IP) Demo Instruction

Rev1.0 9-Oct-17

This document describes the instruction to run RAID0 design by NVMe-IP on FPGA development board. It is recommended to use same SSD model for both connections. The demo can write and read data with 2 NVMe PCIe SSDs as RAID0. Transfer speed from RAID0 is about two times of one SSD speed. User can control test operation through Serial console.

1 Environment Requirement

To demo 2-Ch RAID0 NVMe-IP on Xilinx board, please prepare the following hardware/software.

- 1) Supported FPGA Development board: KCU105
- 2) PC with Vivado software and Serial console software
- 3) FMC to M.2 adapter card and AB16-PCIeXOVR board + power adapter cable from AB16 delivery set
- 4) Xilinx Power adapter for Xilinx board
- 5) Two NVMe SSDs, one PCIe SSD and one M.2 SSD with PCIe to M.2 adapter card
- 6) Two Micro USB cables for programming FPGA and Serial console.

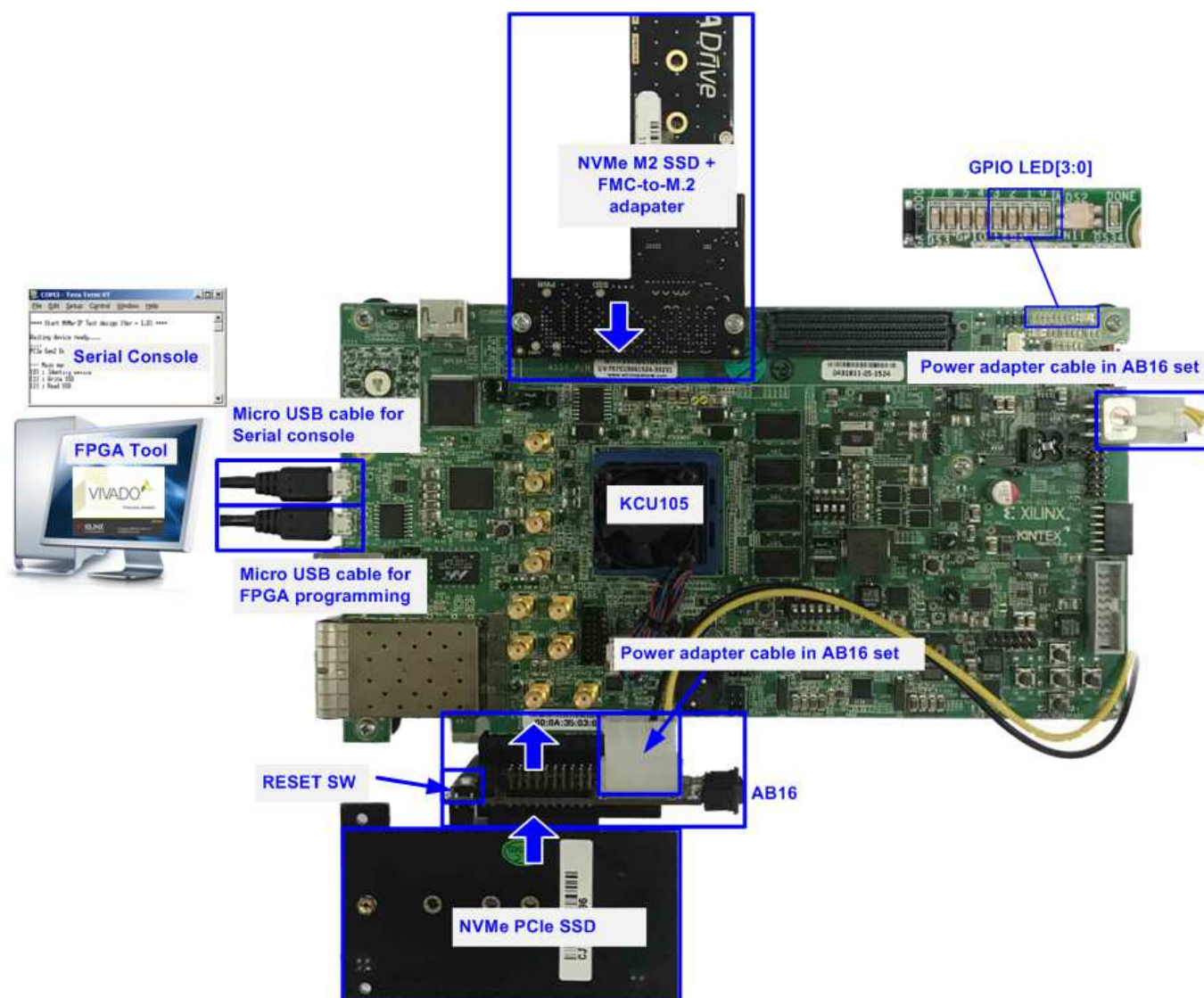


Figure 1-1 RAID0 demo environment setup on KCU105

2 Demo setup

- 1) Power off system.
- 2) Connect power adapter cable from AB16-PCIeXOVR delivery set to power connector on FPGA board, on AB16-PCIeXOVR board, and on Xilinx power adapter as shown in Figure 2-1.

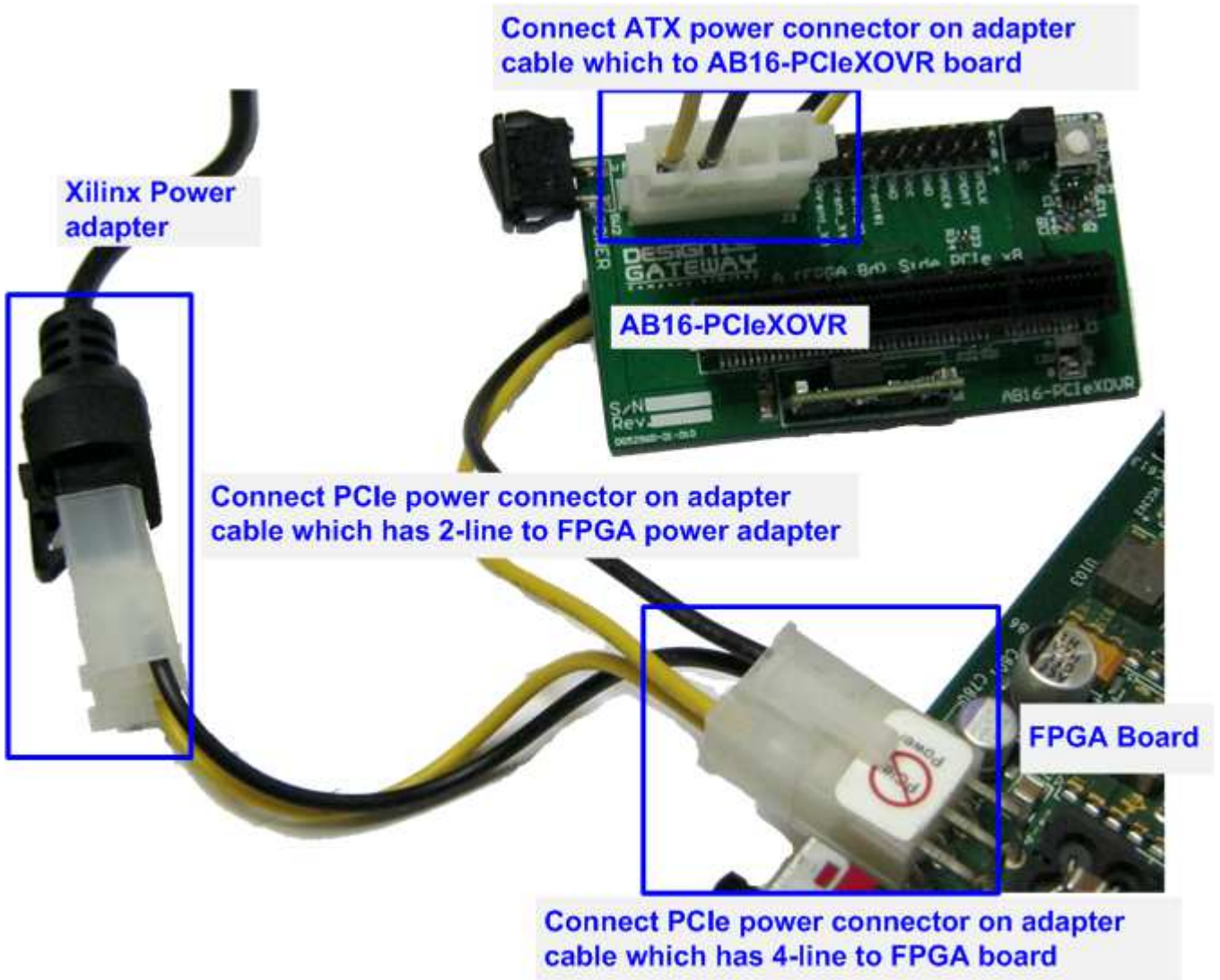


Figure 2-1 Connect power adapter cable to FPGA board, AB16 and Xilinx adapter

- 3) Connect A Side of PCIe connector on AB16-PCIeXOVR board to PCIe connector on Xilinx development board, as shown in Figure 2-2. Also, check that two mini jumpers are inserted at J5 connector on AB16.

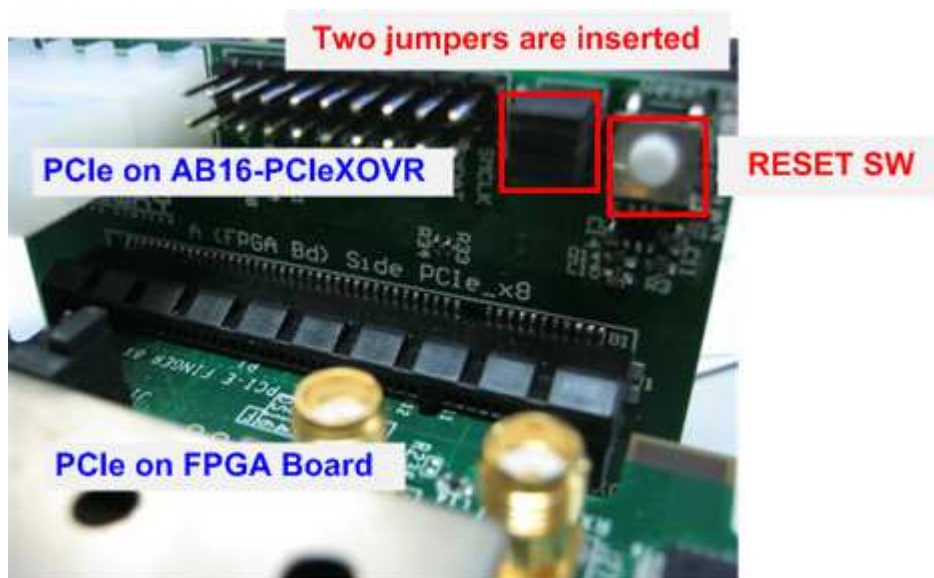


Figure 2-2 Connect PCIe connector between AB16 and FPGA board

- 4) Connect NVMe PCIe SSD to B Side of PCIe connector on AB16-PCIeXOVR board, as shown in Figure 2-3.

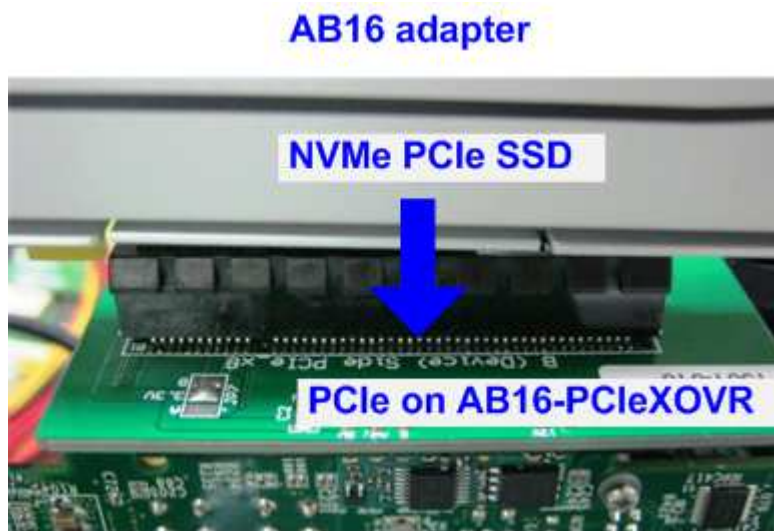


Figure 2-3 Connect NVMe PCIe SSD to AB16-PCIeXOVR board

- 5) Connect M.2 SSD to FMC-to-M.2 adapter and connect FMC-to-M.2 adapter to FMC-HPC connector.

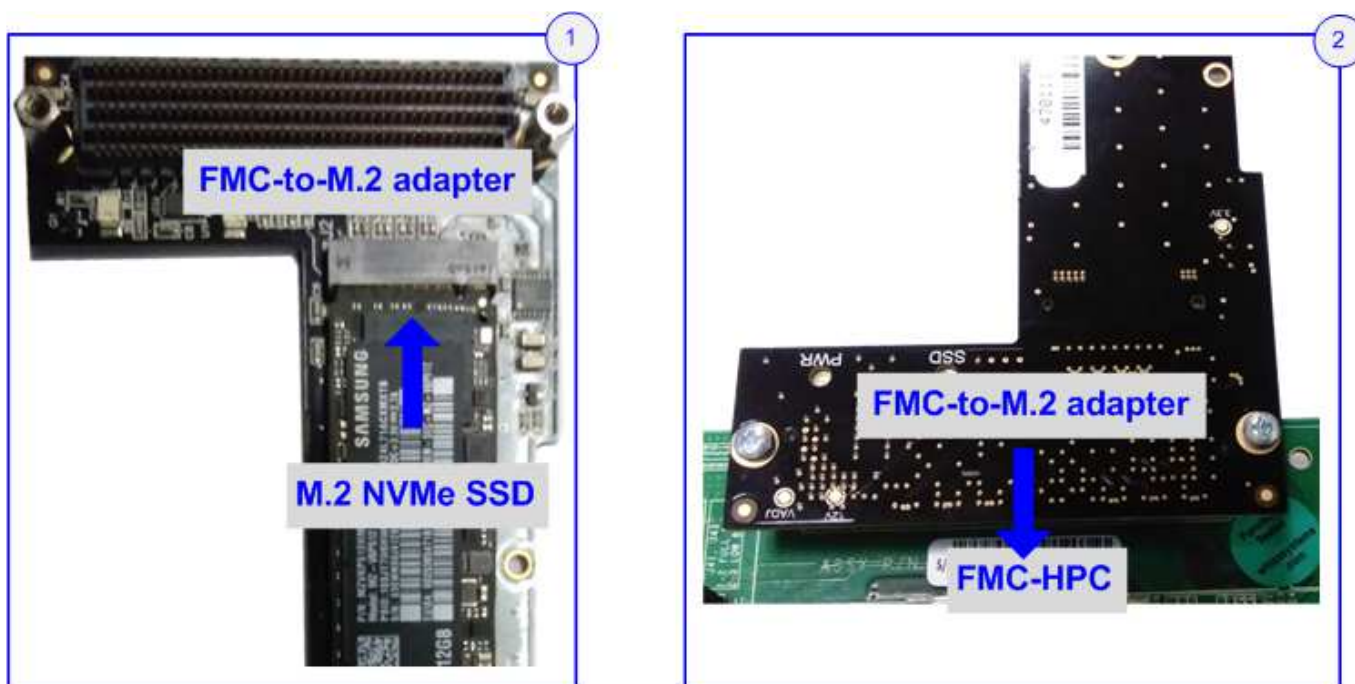


Figure 2-4 Connect M.2 SSD to FMC-to-M.2 adapter

- 6) Connect two micro USB cables for JTAG programming and Serial console.



Figure 2-5 USB Cable connection

- 7) Power on FPGA development board and AB16-PCIeXOVR board, as shown in Figure 2-6 and Figure 2-7.



Figure 2-6 Power on AB16 board

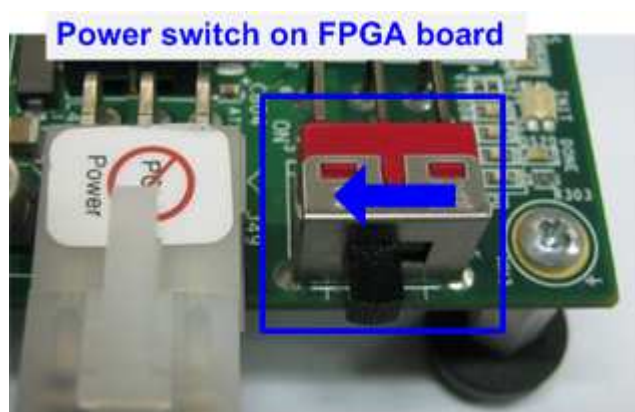


Figure 2-7 Power switch on FPGA board

3 Board setup

- 1) Open Serial console such as TeraTerm, HyperTerminal. Then, set Baud rate = 115,200 Data = 8 bit with Non-Parity Stop = 1.
- 2) For KCU105 board, use Vivado tool to program bit file by the following steps, as shown in Figure 3-1.

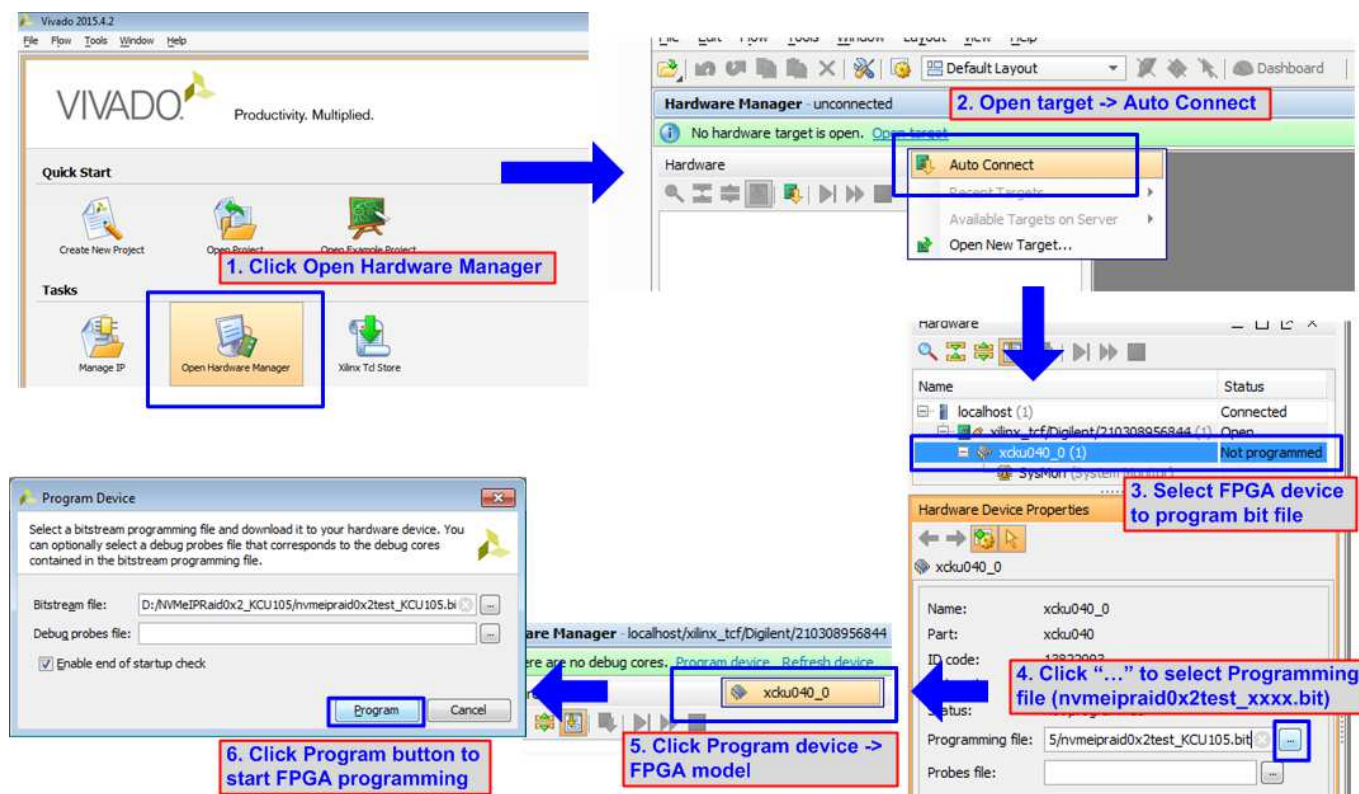


Figure 3-1 Programming steps by using Vivado

- 3) Check LED status on FPGA board. The description of LED is as follows.

Table 1 LED Definition

GPIO LED	ON	OFF
0	Normal operation	Clock is not locked
1	System is busy	Idle status
2	PCIe Error detect	Normal operation
3	Data verification fail	Normal operation

- 4) After programming completely, LED[0] and LED[1] turn ON during PCIe initialization process. Then, LED[1] turns OFF to show that PCIe completes initialization process.

Now system is ready to receive command from user. PCIe speed is displayed on the console and then Main menu is displayed to receive command from user, as shown in Figure 3-3

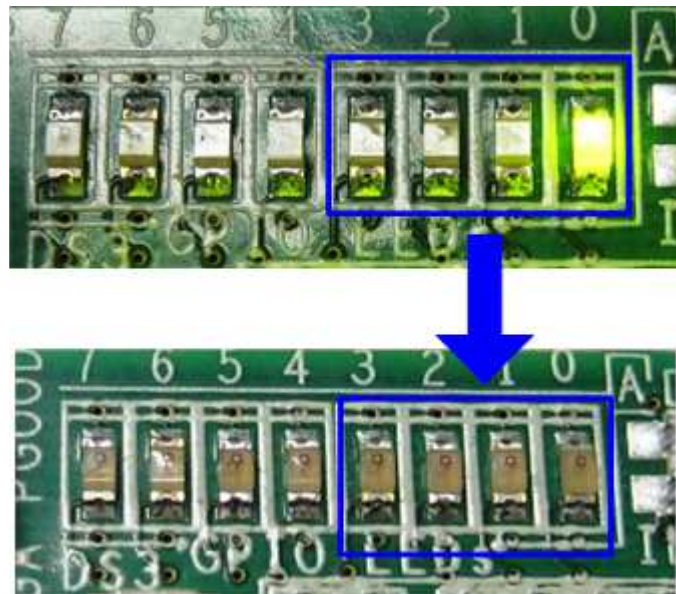


Figure 3-2 LED status after program configuration file and PCIe initialization complete

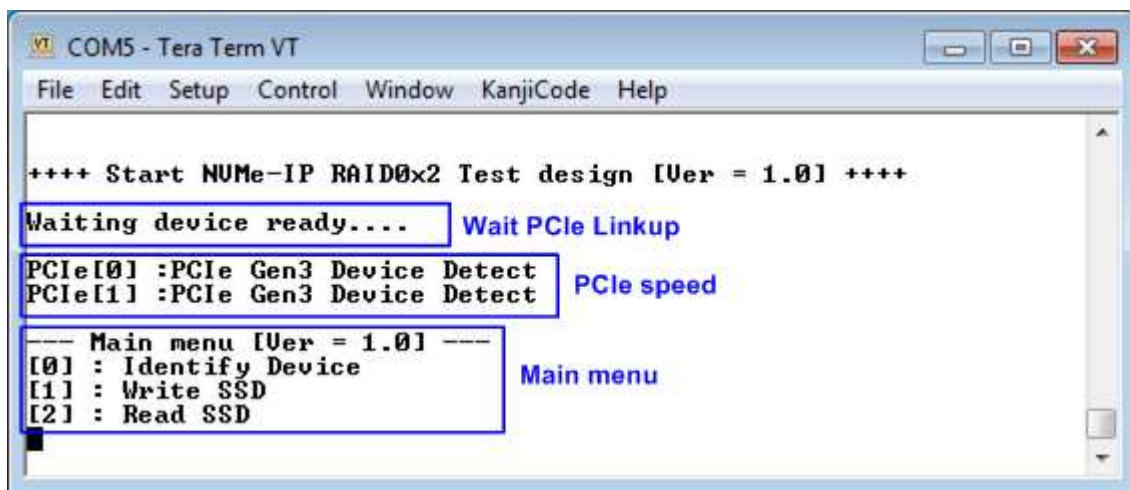


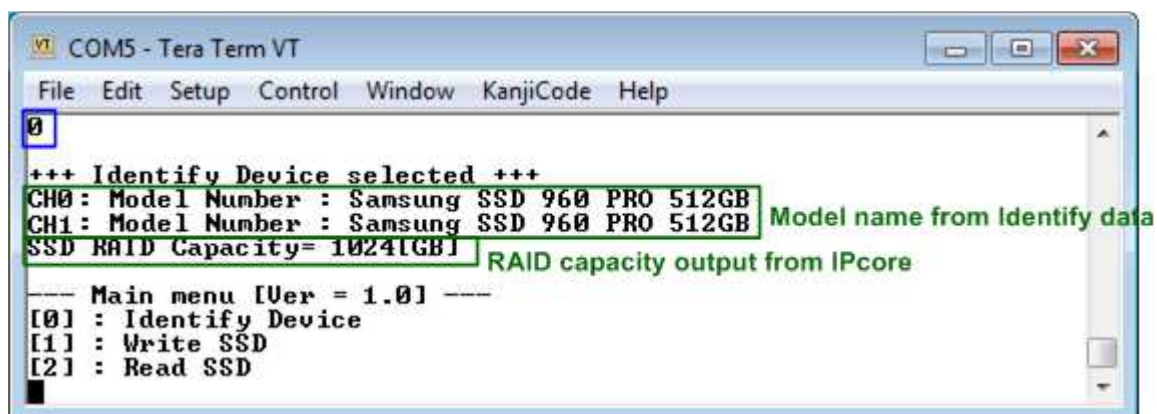
Figure 3-3 Main menu after program configuration file and PCIe initialization complete

4 Test Menu

4.1 Identify Device

Select '0' to send Identify command to RAID0. When operation is completed, RAID capacity and both SSD model names are displayed on the console.

RAID0 capacity is calculated by multiplying SSD capacity @ CH0 by 2.



```

COM5 - Tera Term VT
File Edit Setup Control Window KanjiCode Help
0
+++ Identify Device selected +++
CH0: Model Number : Samsung SSD 960 PRO 512GB
CH1: Model Number : Samsung SSD 960 PRO 512GB
SSD RAID Capacity= 1024[GB]
---- Main menu [Ver = 1.0] ----
[0] : Identify Device
[1] : Write SSD
[2] : Read SSD
  
```

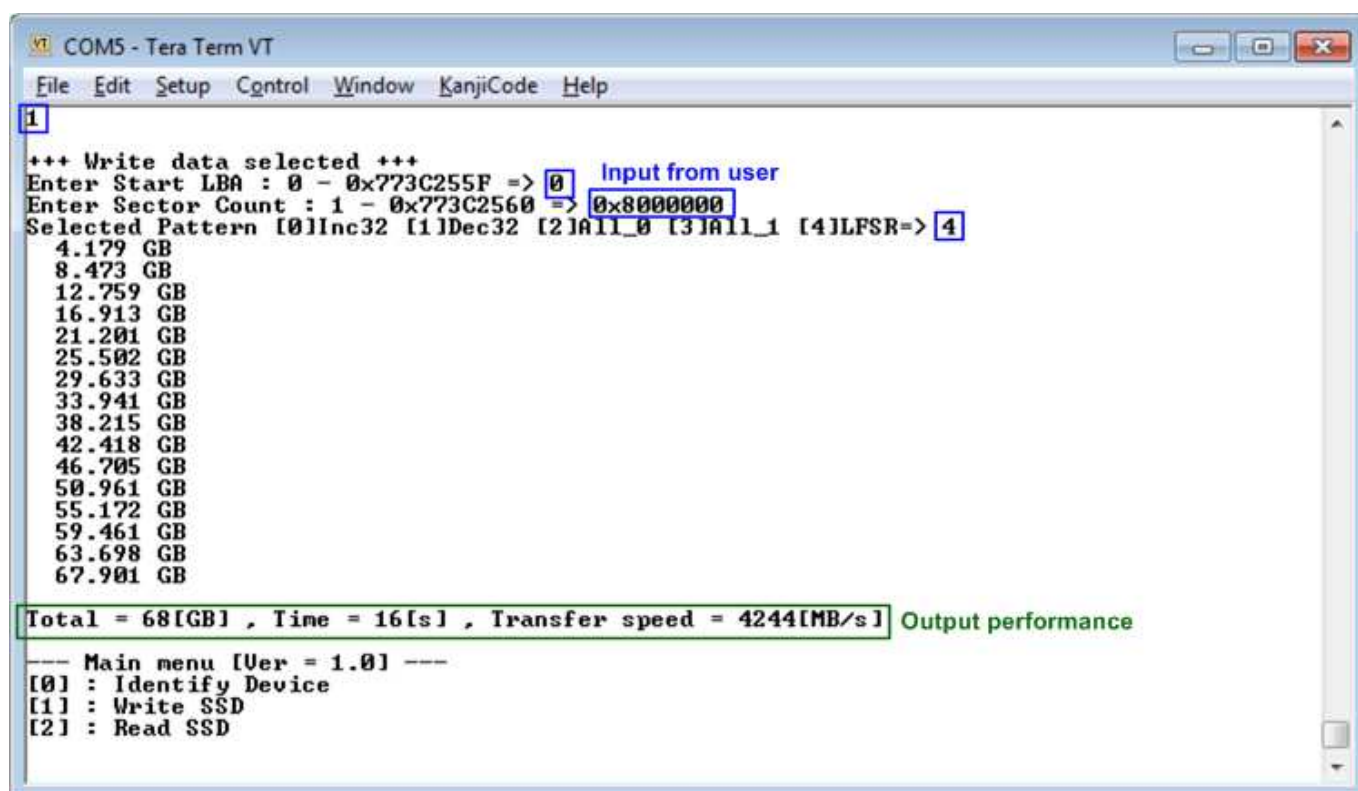
Figure 4-1 Result from Identify Device menu

4.2 Write SSD

Select '1' to send Write command to RAID0. Three inputs are required for this menu.

- 1) Start LBA: Input start address of RAID in sector unit. The input can be decimal unit or add prefix "0x" for hexadecimal unit.
- 2) Sector Count: Input total transfer size in sector unit. The input can be decimal unit or add prefix "0x" for hexadecimal unit.
- 3) Test pattern: Select test pattern of test data for writing to SSD. Five types can be used, i.e. 32-bit increment, 32-bit decrement, all 0, all 1, and LFSR.

As shown in Figure 4-2, if all inputs are valid, the operation will be started. During writing data, current transfer size is printed out to the console to show that system still be alive. Finally, test performance, total size, and total time usage are displayed on the console.



```

COM5 - Tera Term VT
File Edit Setup Control Window KanjiCode Help
1
+++ Write data selected +++
Enter Start LBA : 0 - 0x773C255F => 0 Input from user
Enter Sector Count : 1 - 0x773C2560 => 0x80000000
Selected Pattern [0]Inc32 [1]Dec32 [2]All_0 [3]All_1 [4]LFSR=> 4
4.179 GB
8.473 GB
12.759 GB
16.913 GB
21.201 GB
25.502 GB
29.633 GB
33.941 GB
38.215 GB
42.418 GB
46.705 GB
50.961 GB
55.172 GB
59.461 GB
63.698 GB
67.901 GB
Total = 68[GB] , Time = 16[s] , Transfer speed = 4244[MB/s] Output performance
--- Main menu [Ver = 1.0] ---
[0] : Identify Device
[1] : Write SSD
[2] : Read SSD
  
```

Figure 4-2 Input and result of Write SSD menu

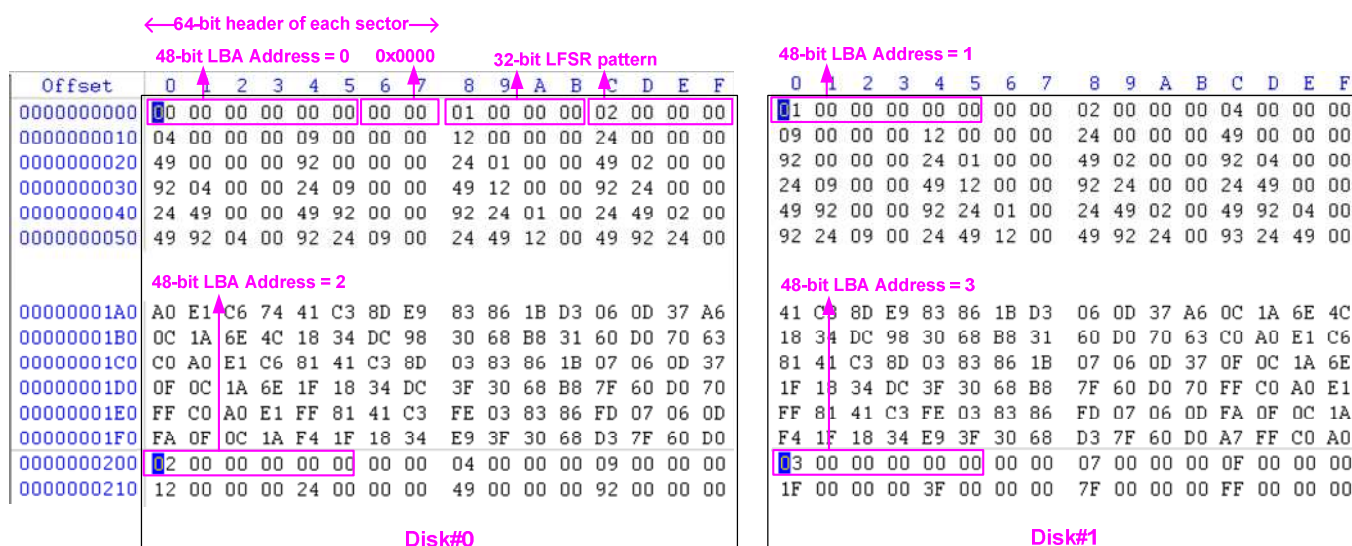


Figure 4-3 Example Test data in sector#0 and #1 of 2 devices by LFSR pattern

Test data of each sector has different 64-bit header which consists of 48-bit LBA address and 16-bit all 0 value. 48-bit LBA address is unique value for each sector. After that, the test pattern is filled following user selection such as 32-bit LFSR pattern (as shown in Figure 4-3)

Stripe size in 2-ch RAID0 demo is 1 sector (512-byte). So, LBA address in the header of 1st sector in Disk#0 and Disk#1 are equal to 0 and 1 sequentially. The address in the header of the next sector for Disk#0 is 2.

Figure 4-4 – Figure 4-6 shows error message when user input is invalid. “Invalid input” message is displayed on the console. Next, it returns to main menu to receive new command.

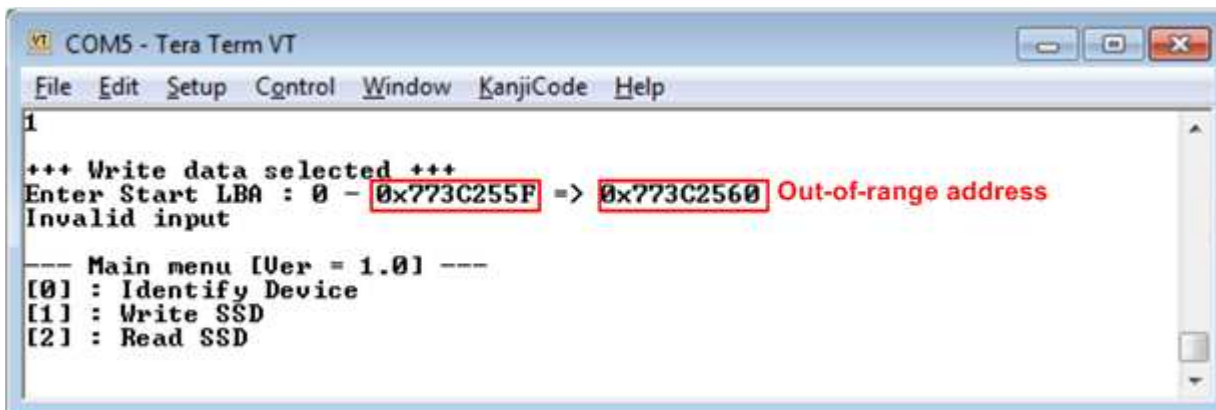


Figure 4-4 Invalid Start LBA input

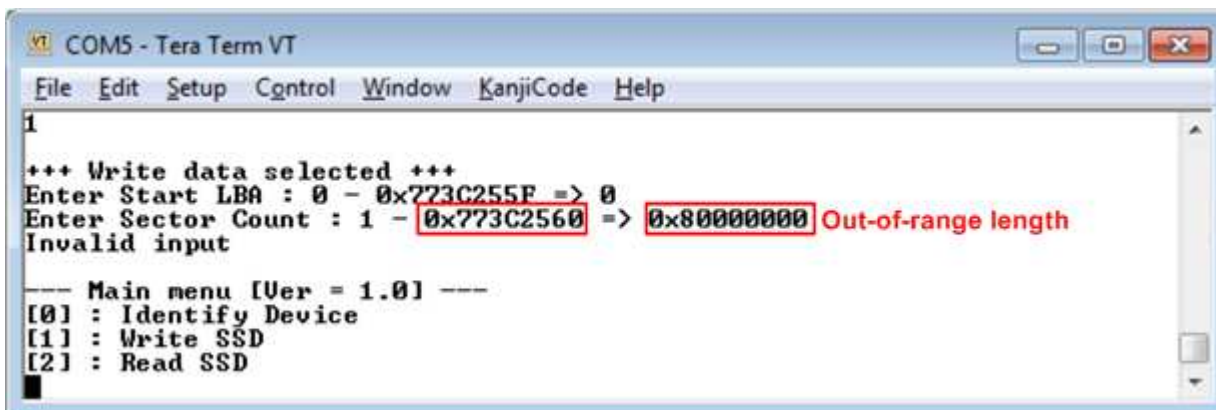


Figure 4-5 Invalid Sector count input

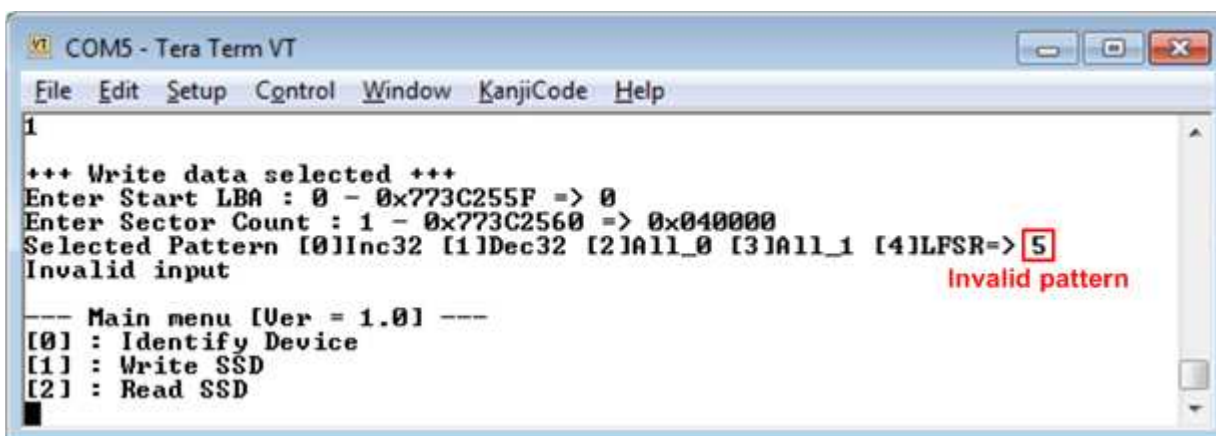


Figure 4-6 Invalid Test pattern input

4.3 Read SSD

Select '2' to send Read command to RAID0. Three inputs are required for this menu.

- 1) Start LBA: Input start address of RAID in sector unit. The input can be decimal unit or add prefix "0x" for hexadecimal unit.
- 2) Sector Count: Input total transfer size in sector unit. The input can be decimal unit or add prefix "0x" for hexadecimal unit.
- 3) Test pattern: Select test pattern to verify data from SSD. Test pattern must be matched with write test. Five types can be used, i.e. 32-bit increment, 32-bit decrement, all 0, all 1 and LFSR.

Similar to write test, if all inputs are valid, test system will read data from SSD. During reading data, current transfer size is printed out to the console to show that system still be alive. Finally, test performance, total size, and total time usage are displayed on the console. "Invalid input" will be displayed if some inputs are out-of-range.

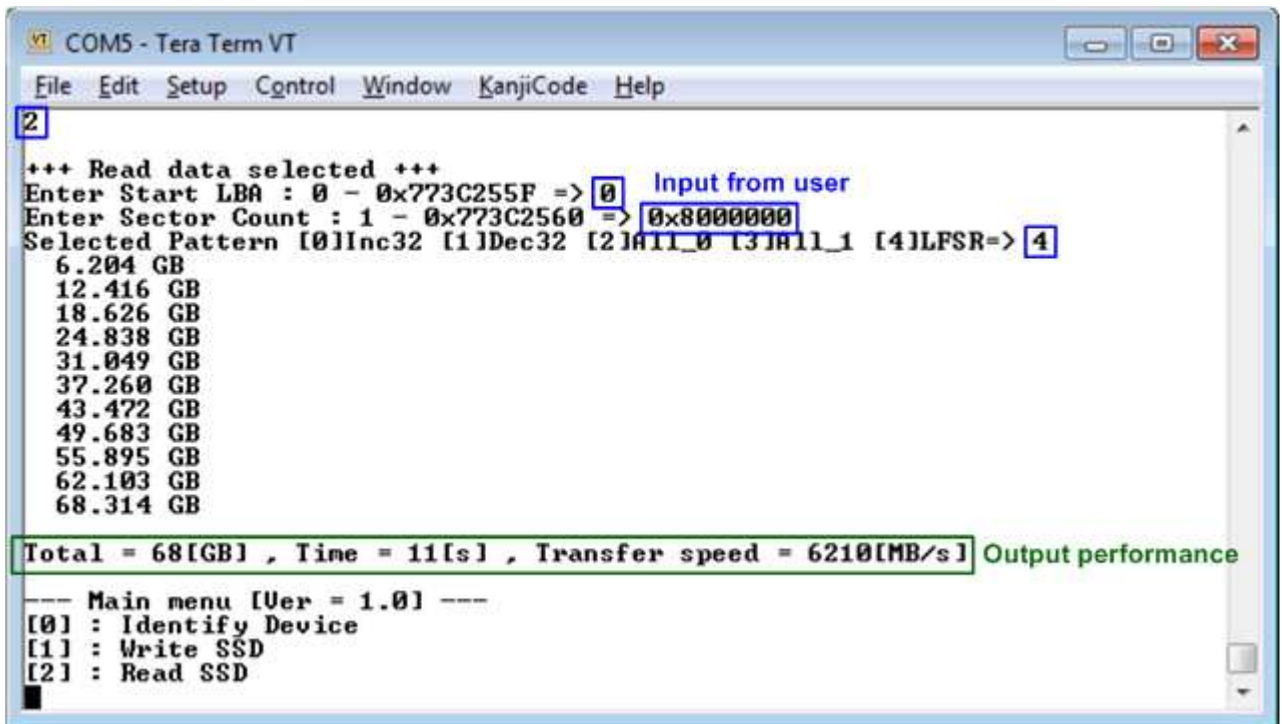


Figure 4-7 Input and result of Read SSD menu

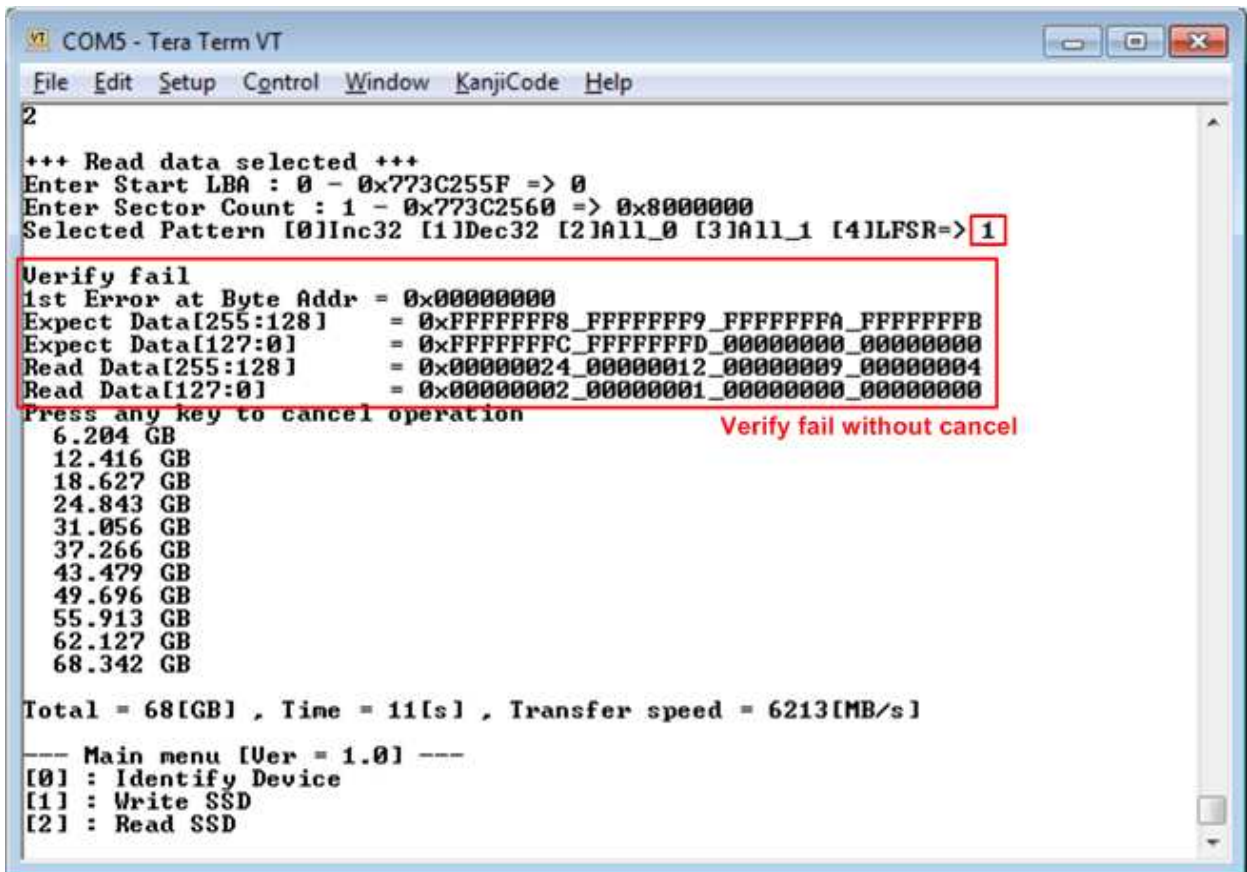


Figure 4-8 Data verification is failed, but wait until read complete

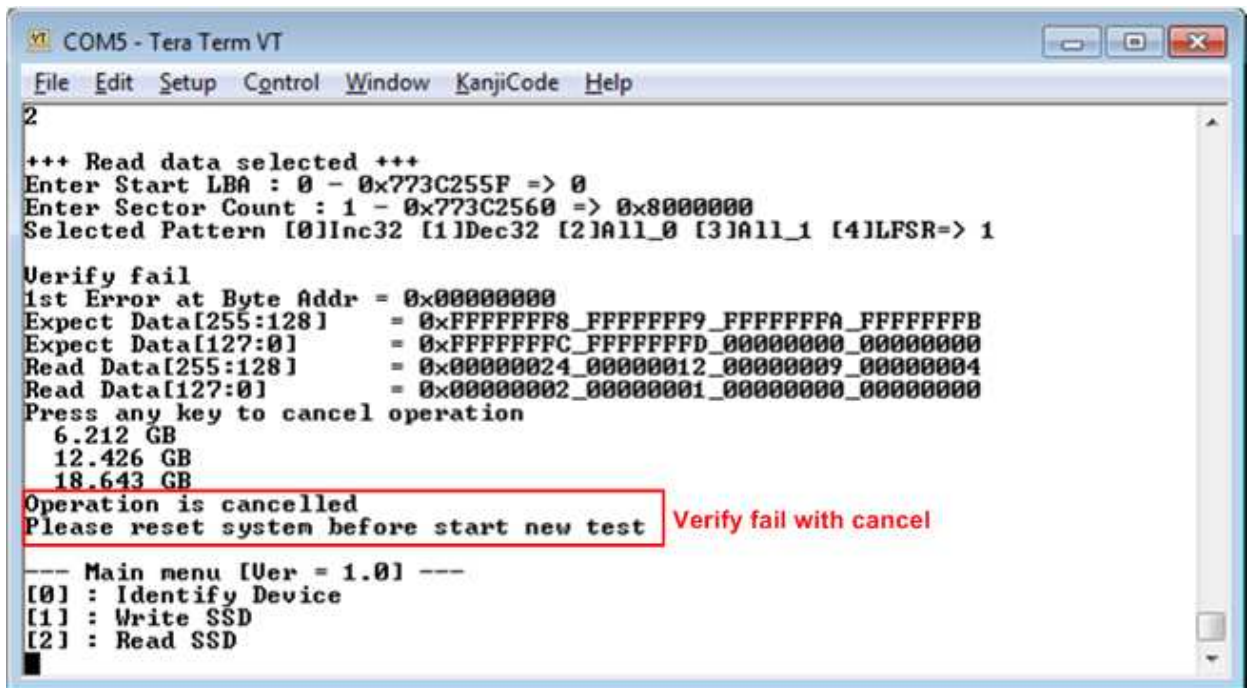


Figure 4-9 Data verification is failed, and press key to cancel operation

Figure 4-8 and Figure 4-9 show the error message when data verification is failed. “Verify fail” message is displayed with error address, expected data, and read data. User can press keyboard to cancel read operation or wait until read process complete.

5 Revision History

Revision	Date	Description
1.0	9-Oct-17	Initial version release