NVMe-IP Demo Instruction

Rev3.2 20-Apr-20

This document describes the instruction to run NVMe-IP demo on FPGA development board by using the PCIe adapter board (AB18-PCIeX16 board or AB16-PCIeXOVR board). The PCIe adapter does not need for some boards which have PCIe female connector. The demo is designed to send NVMe command such as Write, Read, SMART, and Shutdown to NVMe SSD. User controls the test operation through NiosII command shell.

1 Environment Requirement

To run the demo on FPGA board, please prepare following environment.

1) IntelFPGA board:
   - ArriaV GX Starter board (PCIe Gen2)
   - Cyclone10 GX Development board (PCIe Gen2)
   - TR5-Lite Development board (PCIe Gen3)
   - Arria10 SoC Development board (PCIe Gen3)
   - Arria10 GX Development board (PCIe Gen3)
   - Alaric board (PCIe Gen3)
2) PC installing QuartusII programmer and NiosII command shell software
3) The PCIe adapter board (AB18-PCIeX16 or AB16-PCIeXOVR), provided by Design Gateway
   https://dgway.com/ABseries_E.html
4) Power adapter of FPGA board
5) ATX power supply for PCIe adapter board
6) NVMe SSD connecting to PCIe female connector on AB18/AB16 board
7) A cable for FPGA programming/NiosII command shell, connecting between FPGA board and PC.
   a) ArriaV GX Starter board: USB A-B cable
   b) TR5-Lite board: USB A-B cable and USB Blaster
   c) Cyclone10GX, Arria10 SoC, Arria10 GX, and Alaric board: micro USB cable

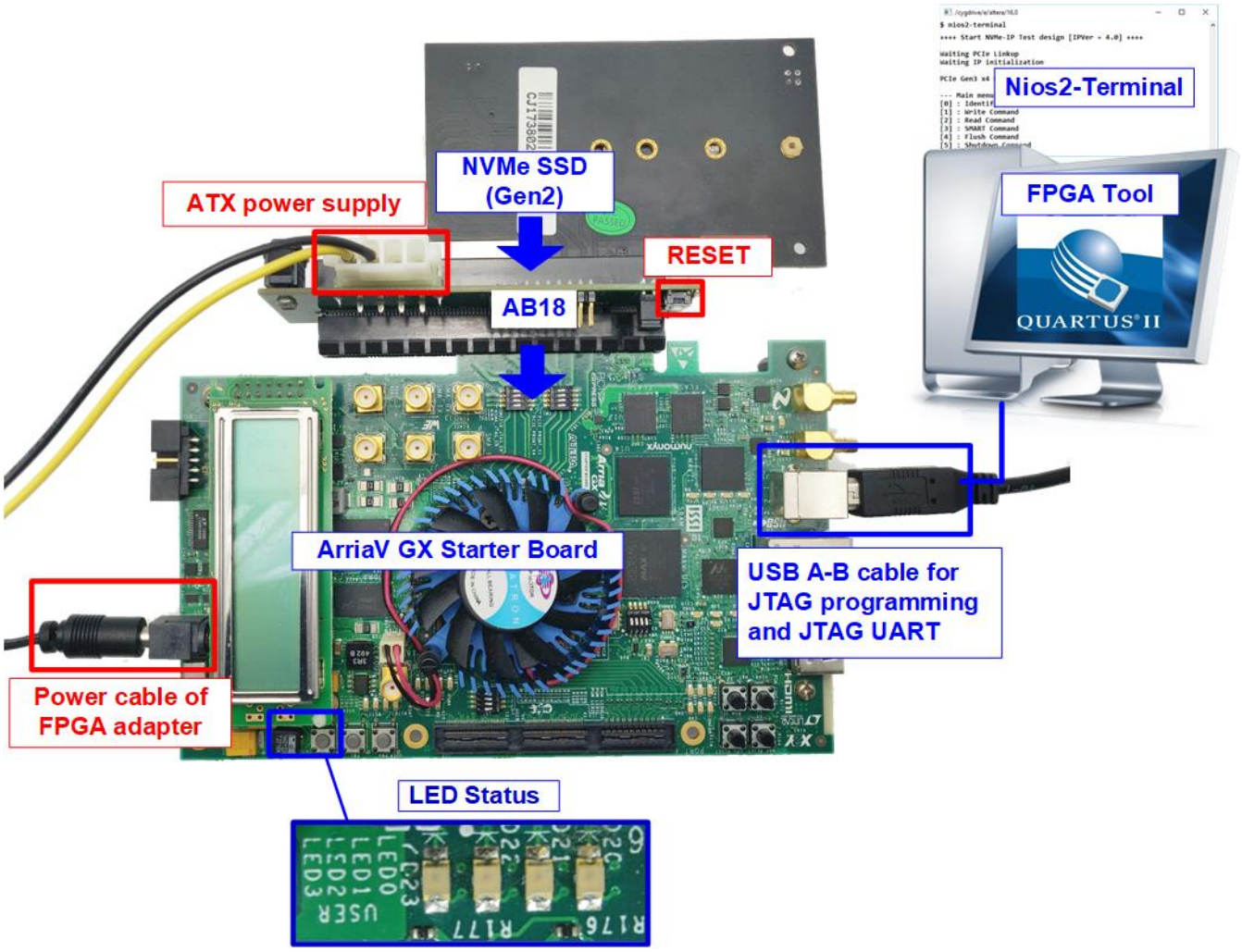

Figure 1-1 NVMe-IP Demo Environment Setup on ArriaV GX Starter Board

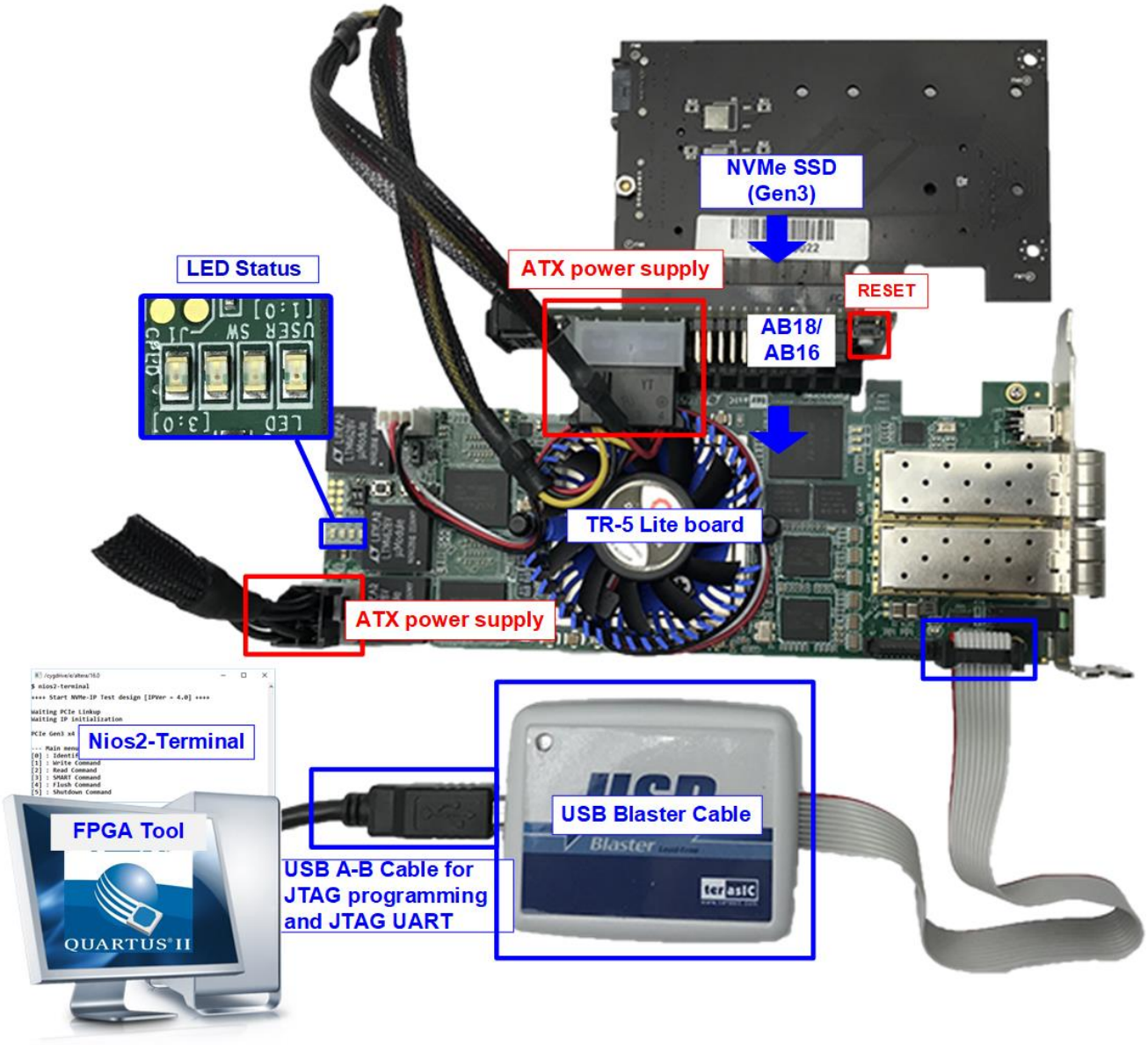

Figure 1-2 NVMe-IP Demo Environment Setup on TR-5 Lite Development Board

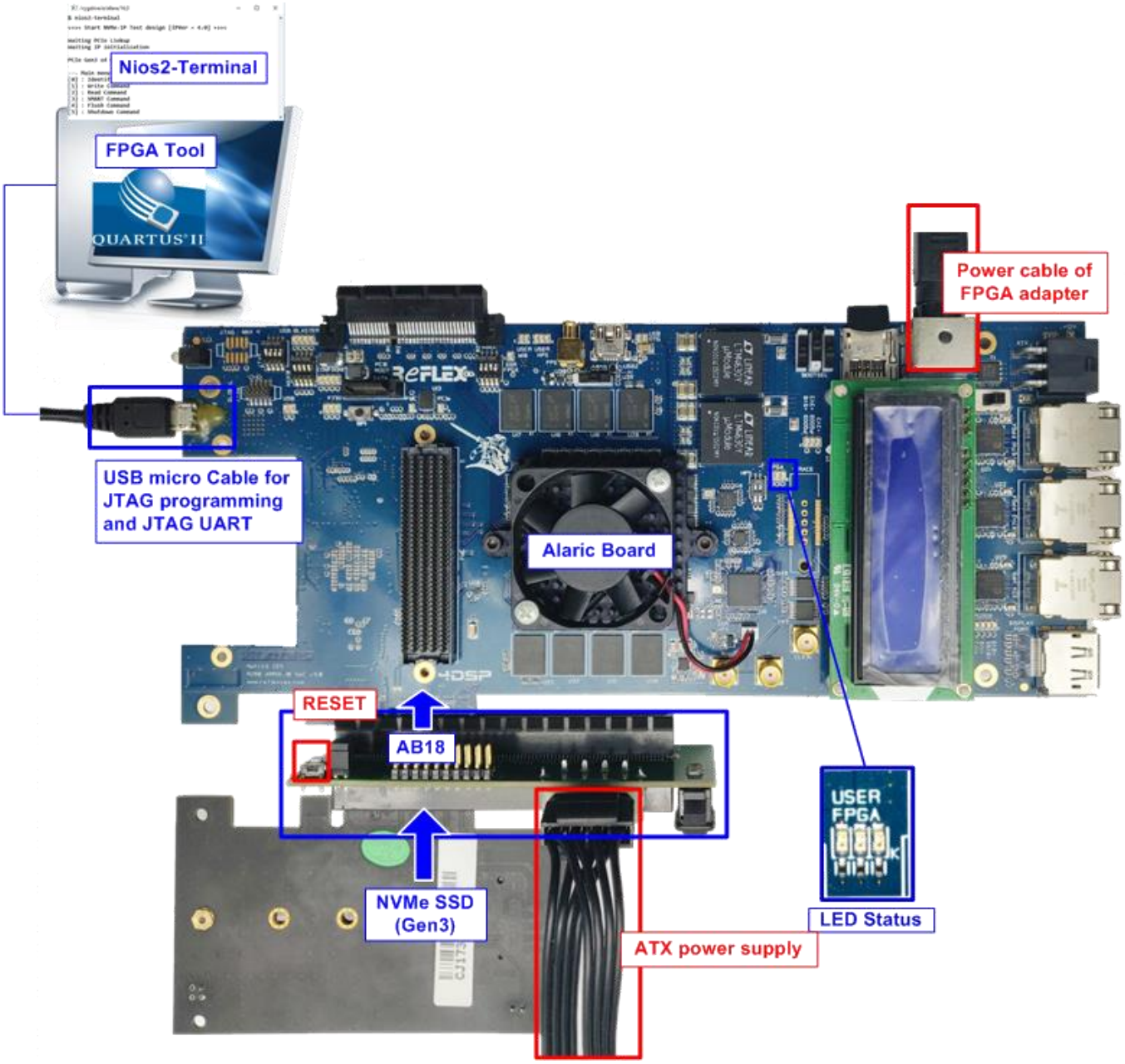

Figure 1-3 NVMe-IP Demo Environment Setup on Alaric Board

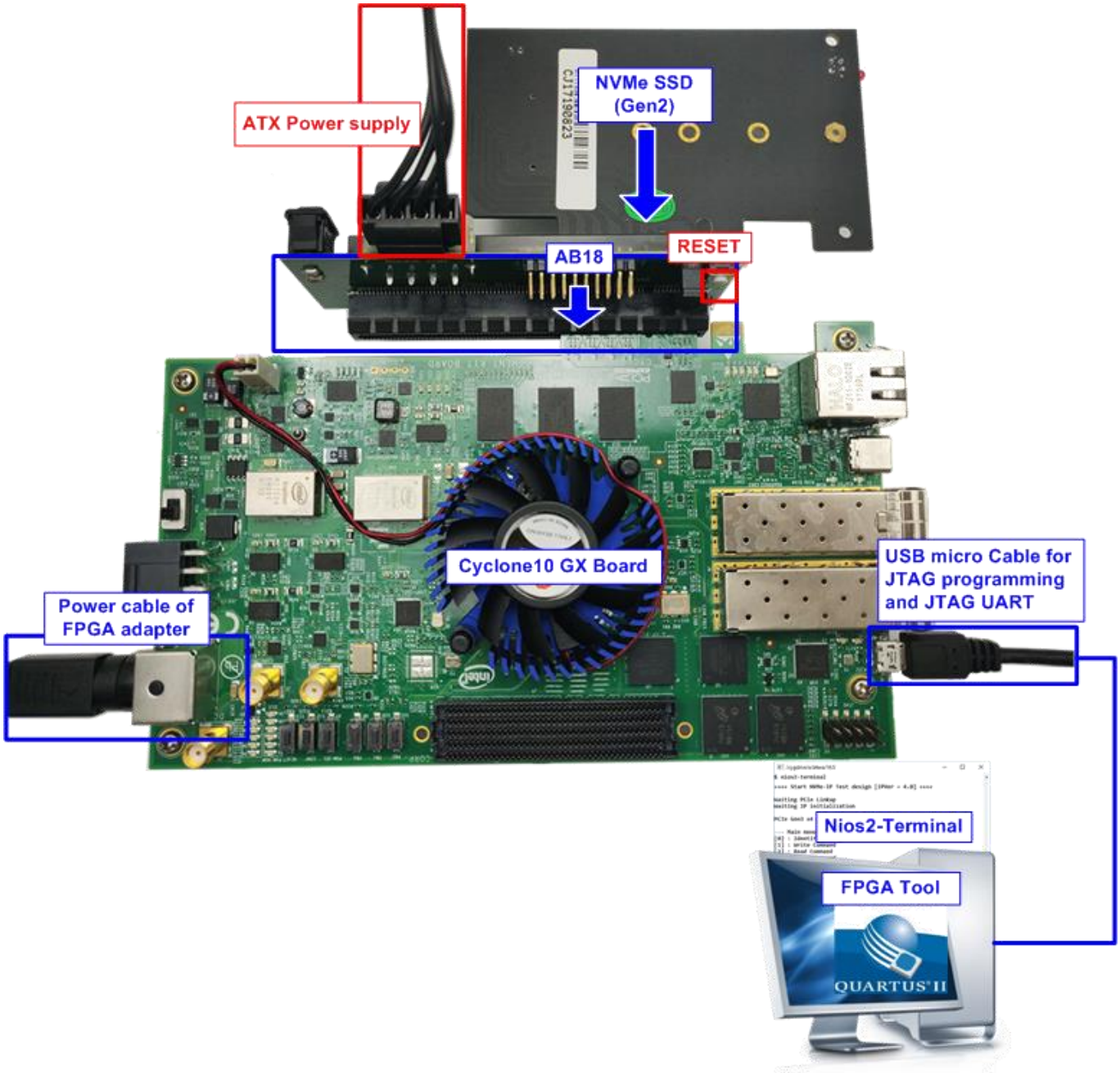

Figure 1-4 NVMe-IP Demo Environment Setup on Cyclone10 GX Development Board

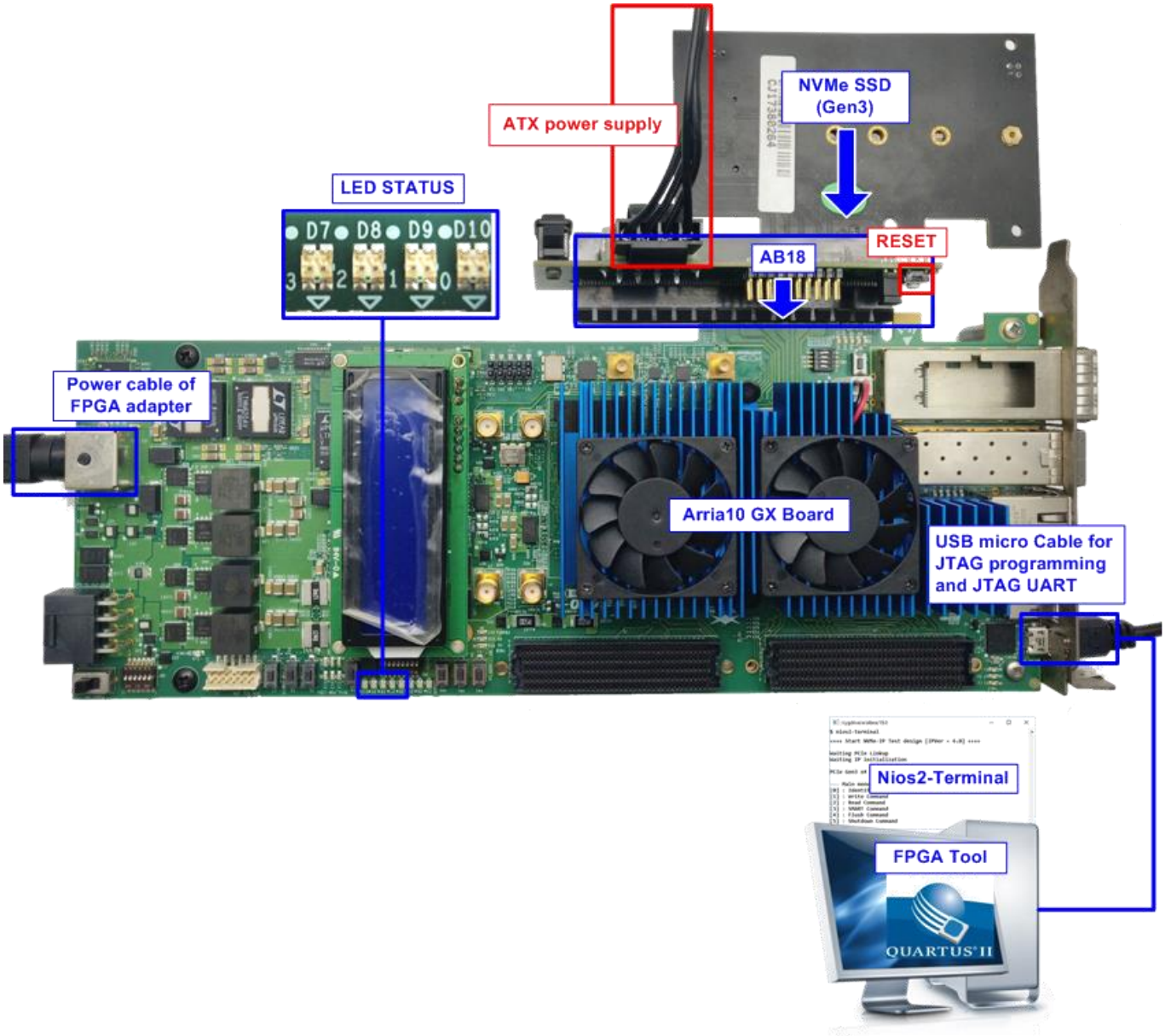

Figure 1-5 NVMe-IP Demo Environment Setup on Arria10 GX Development Board

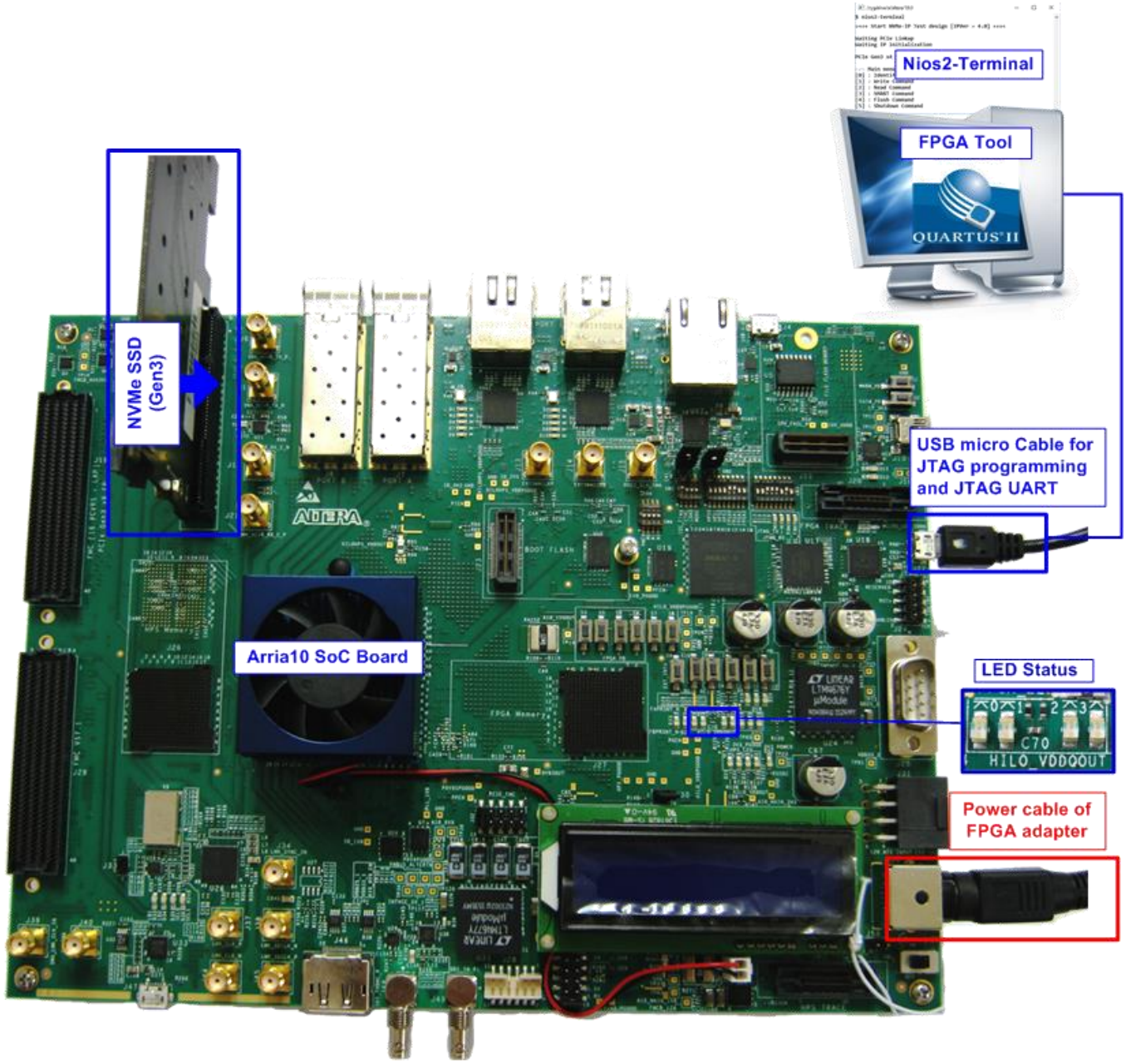

Figure 1-6 NVMe-IP Demo Environment Setup on Arria10 SoC Development Board

2 Demo setup

2.1 Board setup

2.1.1 TR-5 Lite Development board by AB18/AB16 and ATX power supply

1) Power off system.
2) Connect ATX power supply to FPGA board.

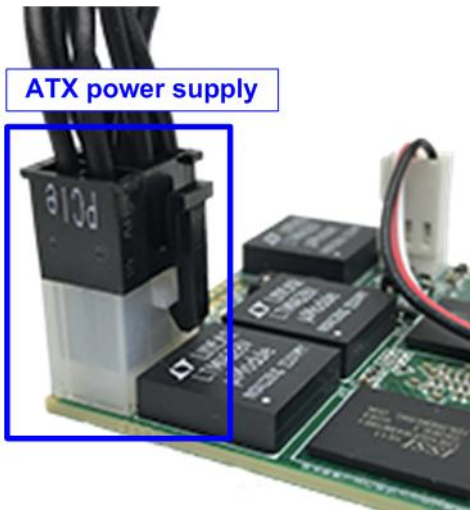

Figure 2-1 ATX power supply connected to FPGA

3) a. Connect A Side of PCIe connector on AB18/AB16-PCIeXOVR board to PCIe connector on FPGA board.
   b. Check that two mini jumpers are inserted at J5 connector on AB16.
   c. Connect NVMe SSD to B Side of PCIe connector on AB16-PCIeXOVR board
   d. Connect ATX power supply to AB18/AB16-PCIeXOVR and FPGA board as shown in Figure 2-2

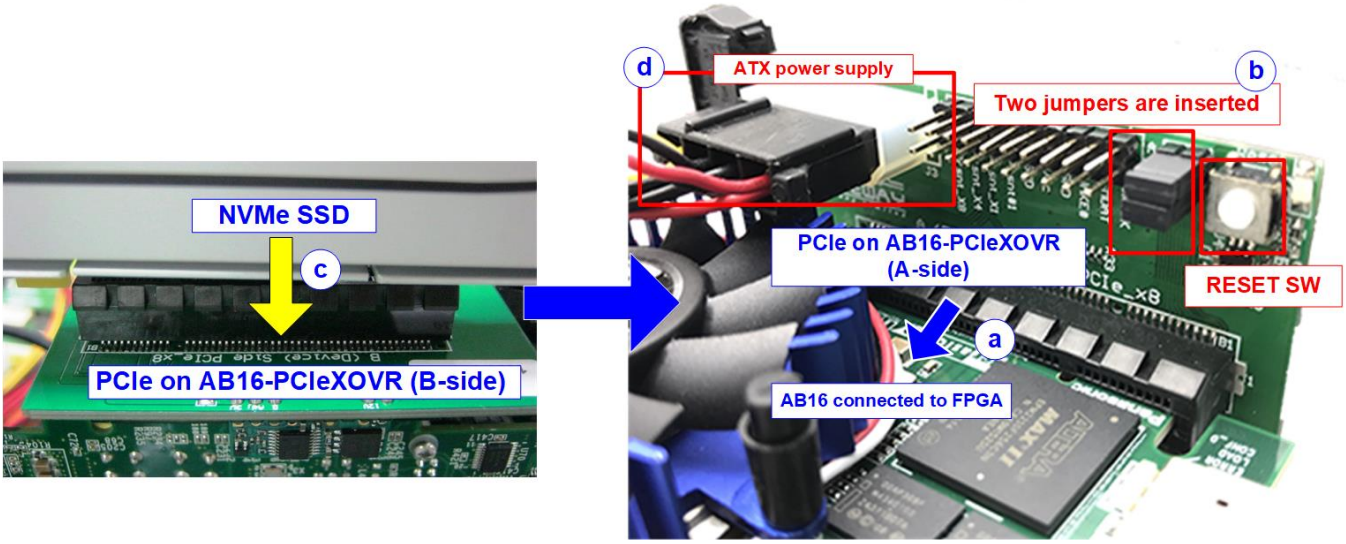

Figure 2-2 Connect PCIe connector between AB16 and FPGA board

4) a. Connect USB Blaster cable to FPGA board for JTAG programming/NiosII command shell.
   b. Connect USB A-B cable to USB Blaster, as shown in Figure 2-3.

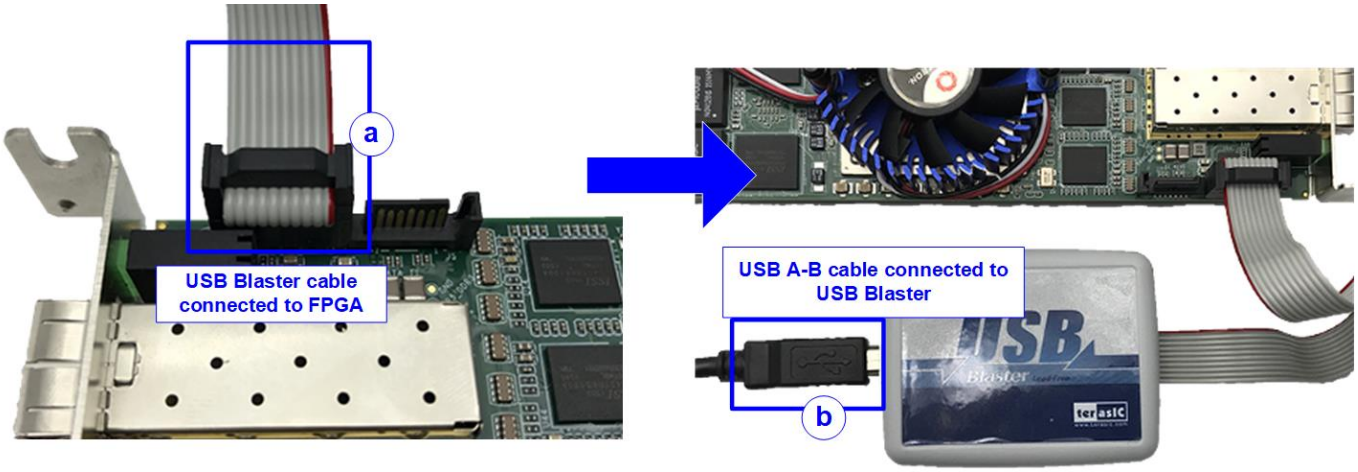

Figure 2-3 USB cable connection

5) Turn on power switch of ATX power supply and AB16-PCIeXOVR board.

Figure 2-4 Power on AB16 board

2.1.2 Arria10 GX/Cyclone10GX/Alaric board or ArriaV GX by AB18 and ATX power supply

1) Power off system.
2) a. Connect ATX power to ATX power connector on AB18-PCIeX16 board.
   b. Connect A Side of PCIe connector on AB18-PCIeX16 board to PCIe connector on FPGA board
   c. Check that two mini jumpers are inserted at J5 connector on AB18
   d. Connect NVMe SSD to B Side of PCIe connector on AB18-PCIeX16 board

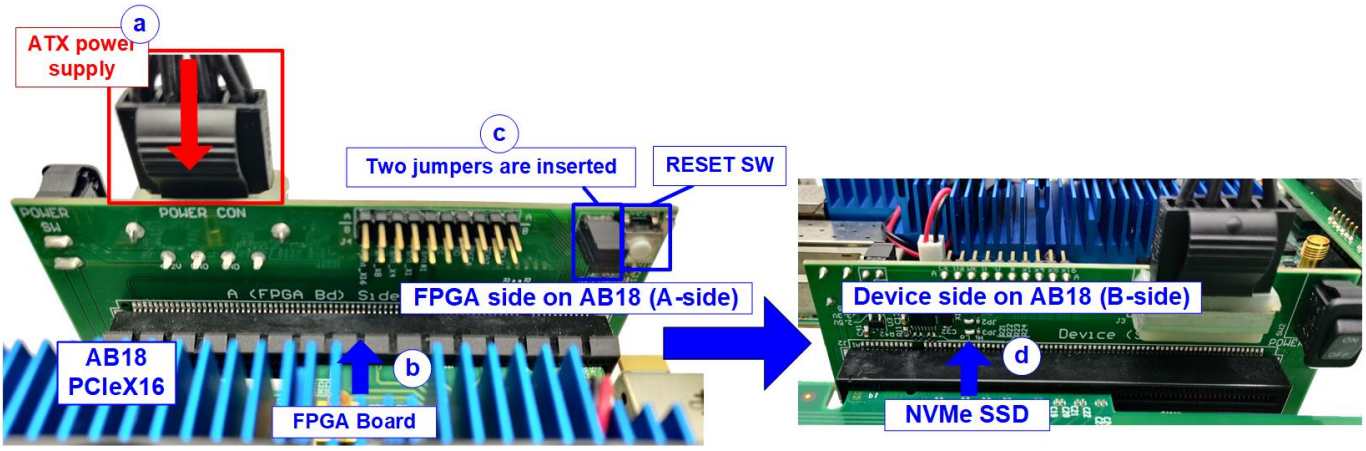

Figure 2-5 Connect PCIe connector between AB16 and FPGA board

3) For JTAG programming and NiosII command shell;
   - For Arria10 GX, Cyclone10GX and Alaric board, connect micro USB cable from FPGA board to PC
   - For ArriaV GX, connect USB A-B cable from FPGA board to PC
4) Connect FPGA power adapter to FPGA board.
5) Turn on power switch of FPGA development board, AB18-PCIeX16 board, and ATX power supply.

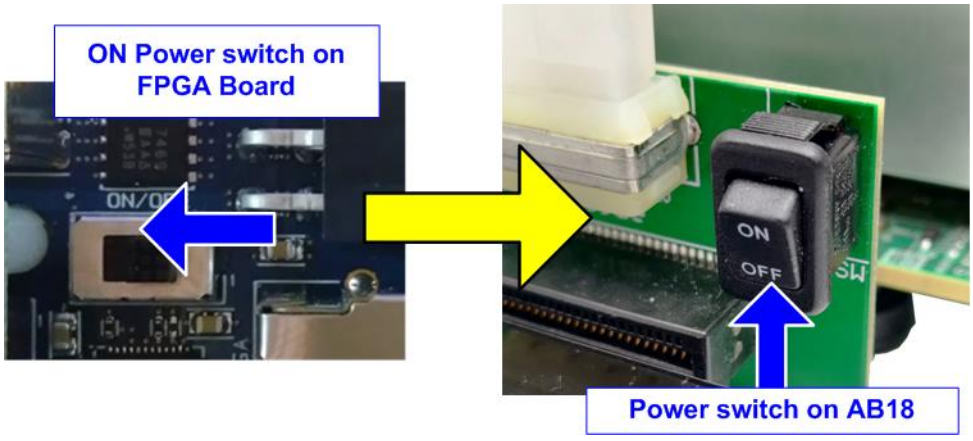

Figure 2-6 Power on FPGA board and AB16 board

2.1.3 Arria10 SoC Development board

1) Power off system.
2) Connect NVMe SSD to PCIe connector on FPGA board, as shown in Figure 2-7.
3) Connect micro USB cable from FPGA board to PC for JTAG programming/NiosII command shell.

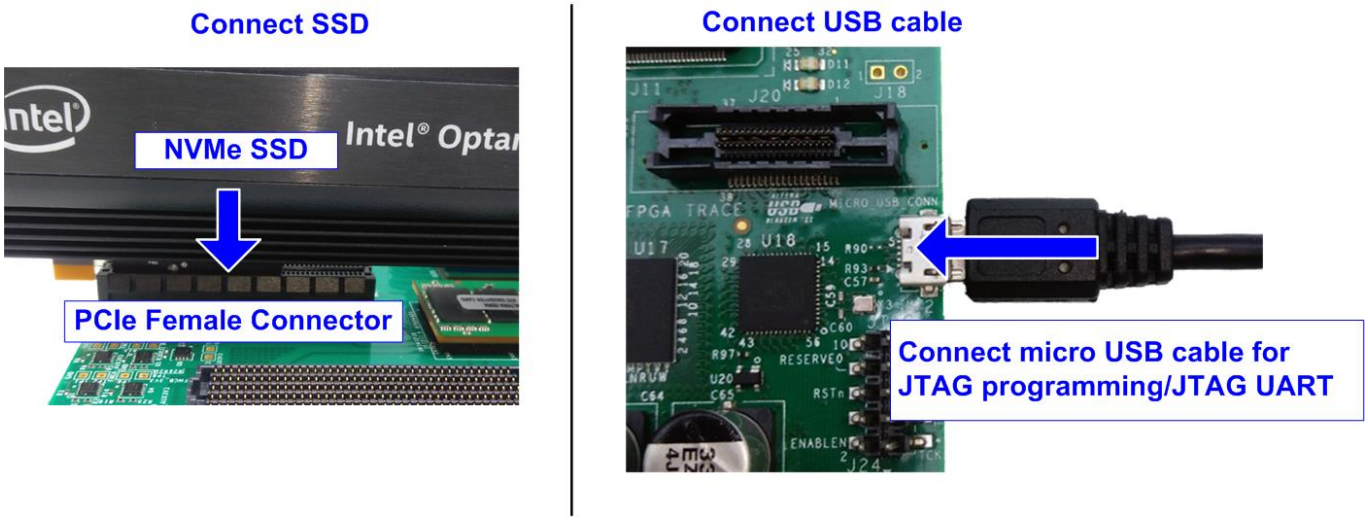

Figure 2-7 NVMe PCIe SSD connection on FPGA board

4) Set SW1[2] = OFF position to source power to the PCIe, as shown in Figure 2-8.

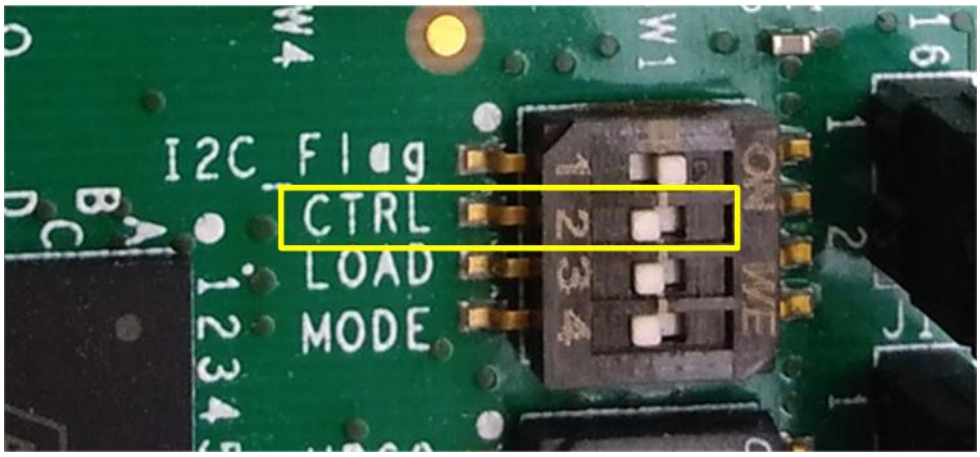

Figure 2-8 Source power to the PCIe for Arria10 SoC board

5) Connect FPGA power adapter to FPGA board and power on FPGA development board.

2.2 Program setup

1) Use QuartusII Programmer to program “NVMeIPTest.sof” file, as shown in Figure 2-9.

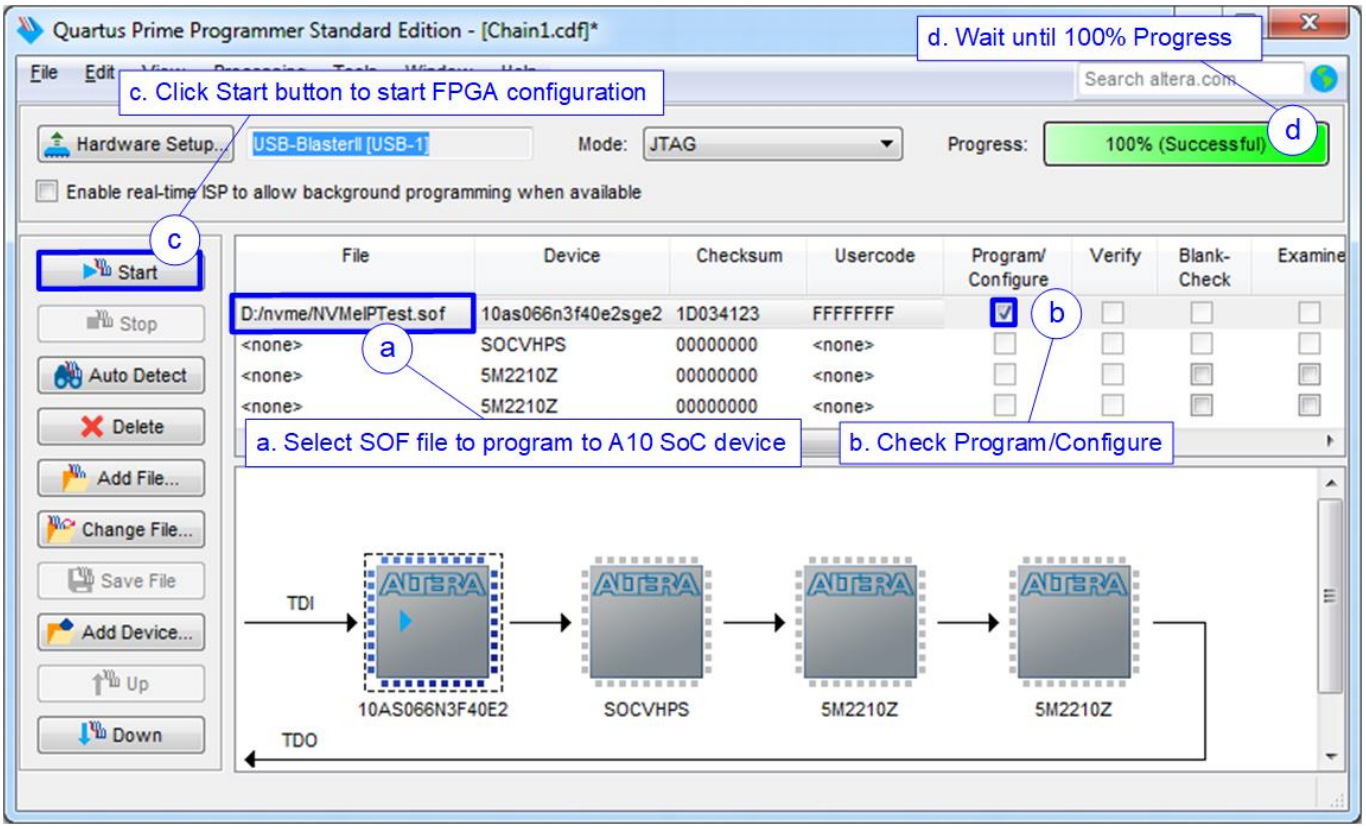

Figure 2-9 Programmed by QuartusII Programmer

2) Open NiosII Command Shell and run nios2-terminal command. Boot message is displayed.
   - "Waiting PCIe Linkup" is displayed when CPU monitors linkup signal of PCIe-IP.
   - "Waiting IP initialization" is displayed after PCIe-IP linkup and CPU waits NVMe-IP initialization process.
   - "PCIe Gen3 (or 2/1) Gen x4 (or x2/x1) Device Detect" shows PCIe speed and PCIe lane after finishing IP initialization process.
   - Finally, Main menu is displayed to receive command from user.

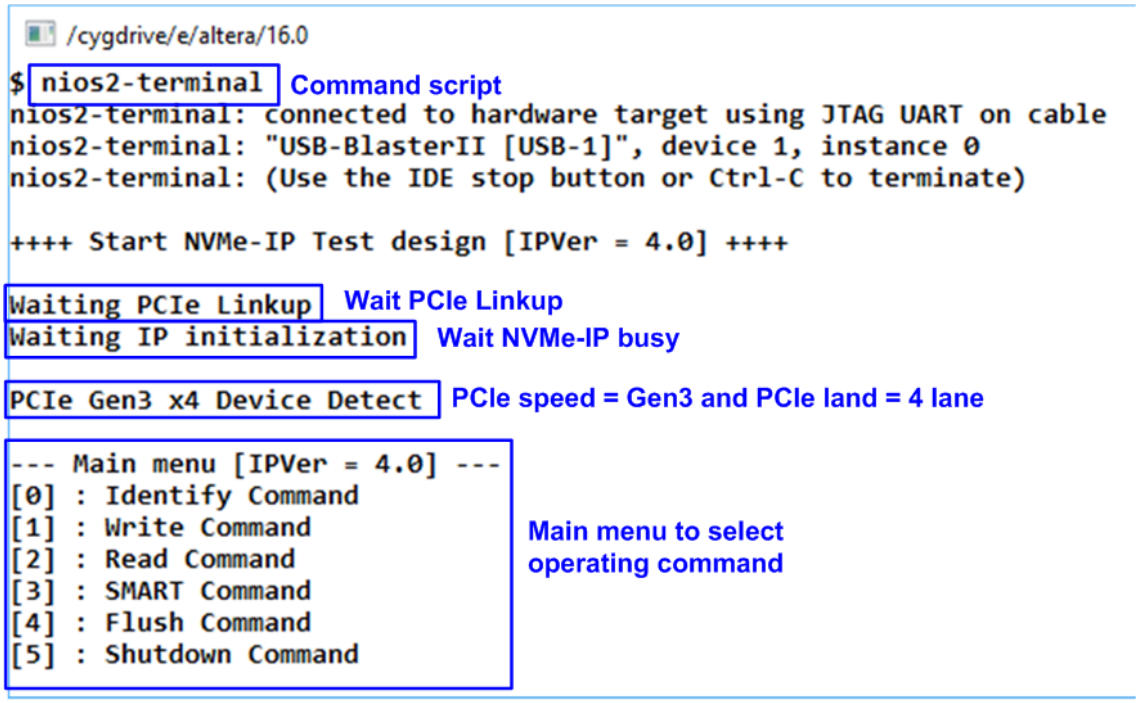

Figure 2-10 NiosII Terminal

3) Check LED status on FPGA board. The description of LED is shown as follows.
   Note: There are three LEDs on Alaric board. Therefore, LED3 is not available.

4) After programming completely, LED[0] and LED[1] are ON when PCIe initialization is processing. LED[1] is OFF after PCIe completes initialization process. After that, the system is ready to receive command from user.

Table 2-1 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

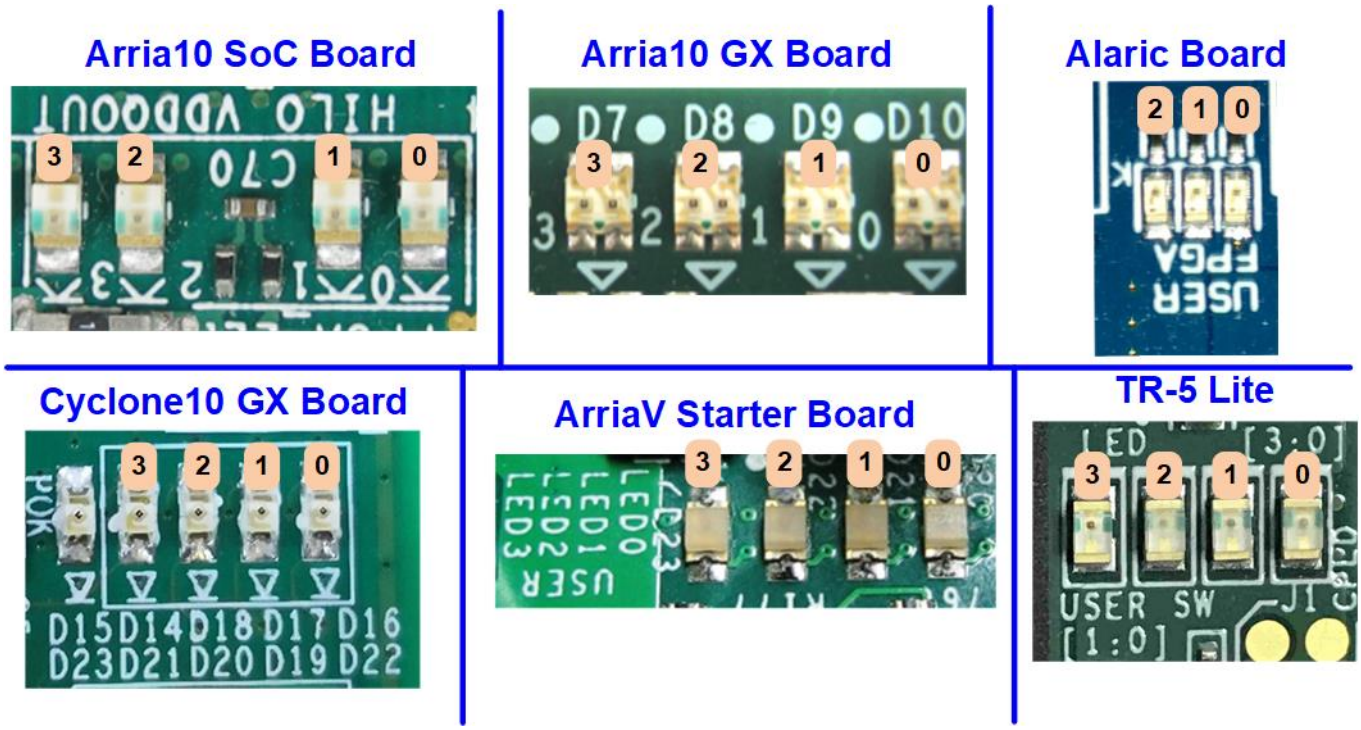

Figure 2-11 LED Status for user output

3 Test Menu

3.1 Identify Command

Select ‘0’ to send Identify command to NVMe SSD.

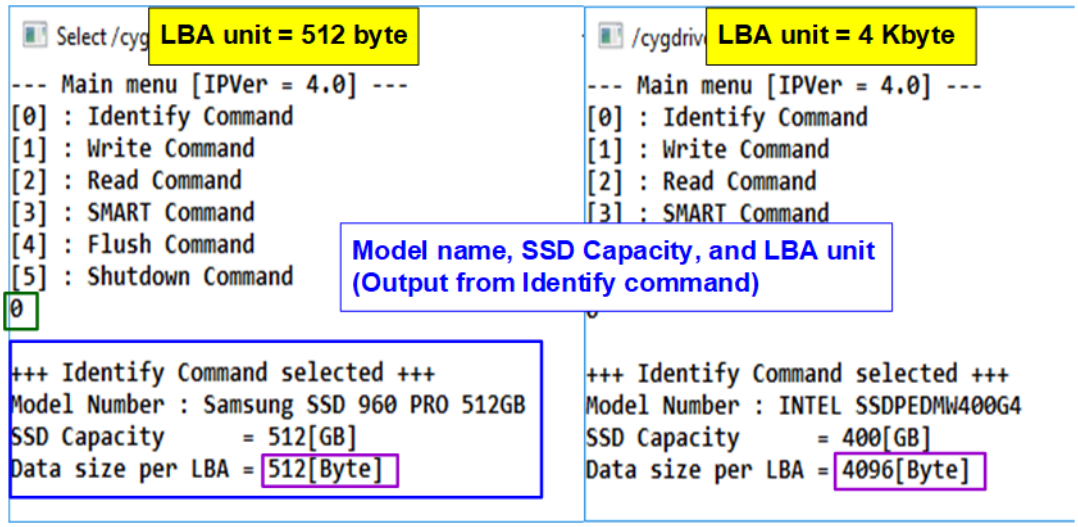

Figure 3-1 Test result when running Identify command

After finishing the operation, the SSD information output from Identify command is displayed. The console shows three values.

1) SSD model number: This value is decoded from Identify controller data.
2) SSD capacity: This value is signal output from NVMe-IP.
3) Data size per LBA: This value is signal output from NVMe-IP. Two values are supported, i.e. 512 byte and 4 Kbyte.

3.2 Write Command

Select '1' to send Write command to NVMe SSD.

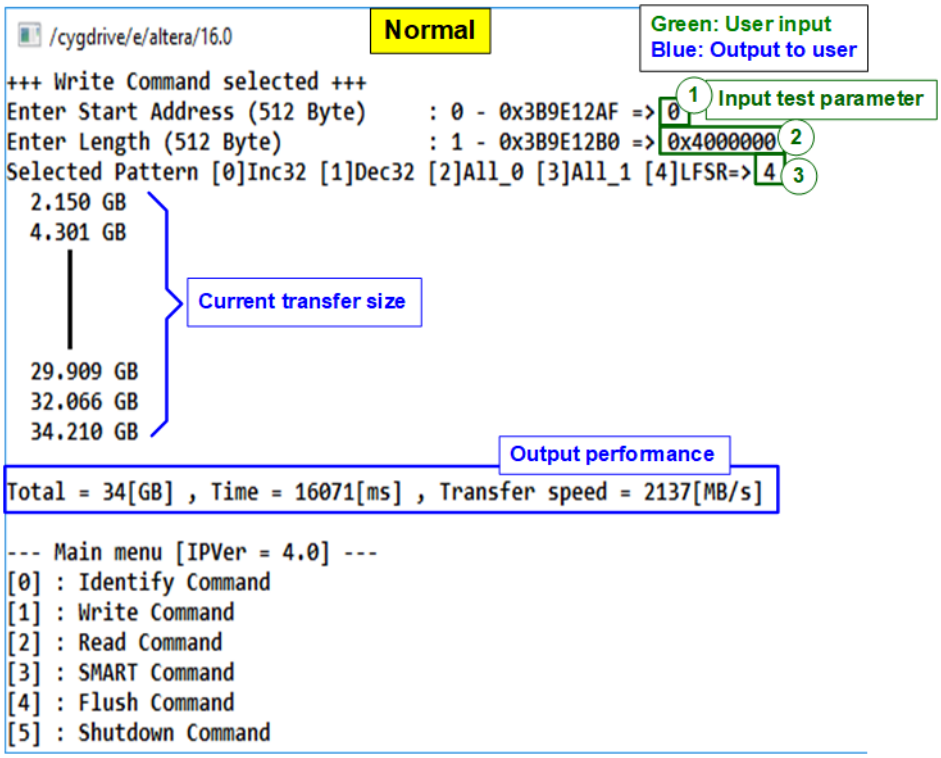

Figure 3-2 Test result when running Write command

User inputs three parameters as follows.

1) Start Address: Input start address to write SSD as 512-byte unit. The input is decimal unit when user enters only digit number. User can add "0x" to be a prefix for hexadecimal unit. When LBA unit of SSD is 4 Kbyte, this input must be aligned to 8.
2) Transfer Length: Input total transfer size as 512-byte unit. The input is decimal unit when user enters only digit number. User can add "0x" to be a prefix for hexadecimal unit. When LBA unit of SSD is 4 Kbyte, this input must be aligned to 8.
3) Test pattern: Select test data pattern for writing to SSD. There are five patterns, i.e. 32-bit incremental, 32-bit decremental, all-0, all-1, and 32-bit LFSR counter.

When all inputs are valid, the operation begins. During writing data, current transfer size is displayed on the console every second to show that system is still alive. Finally, total size, total time usage, and test speed are displayed on the console to be a test result.

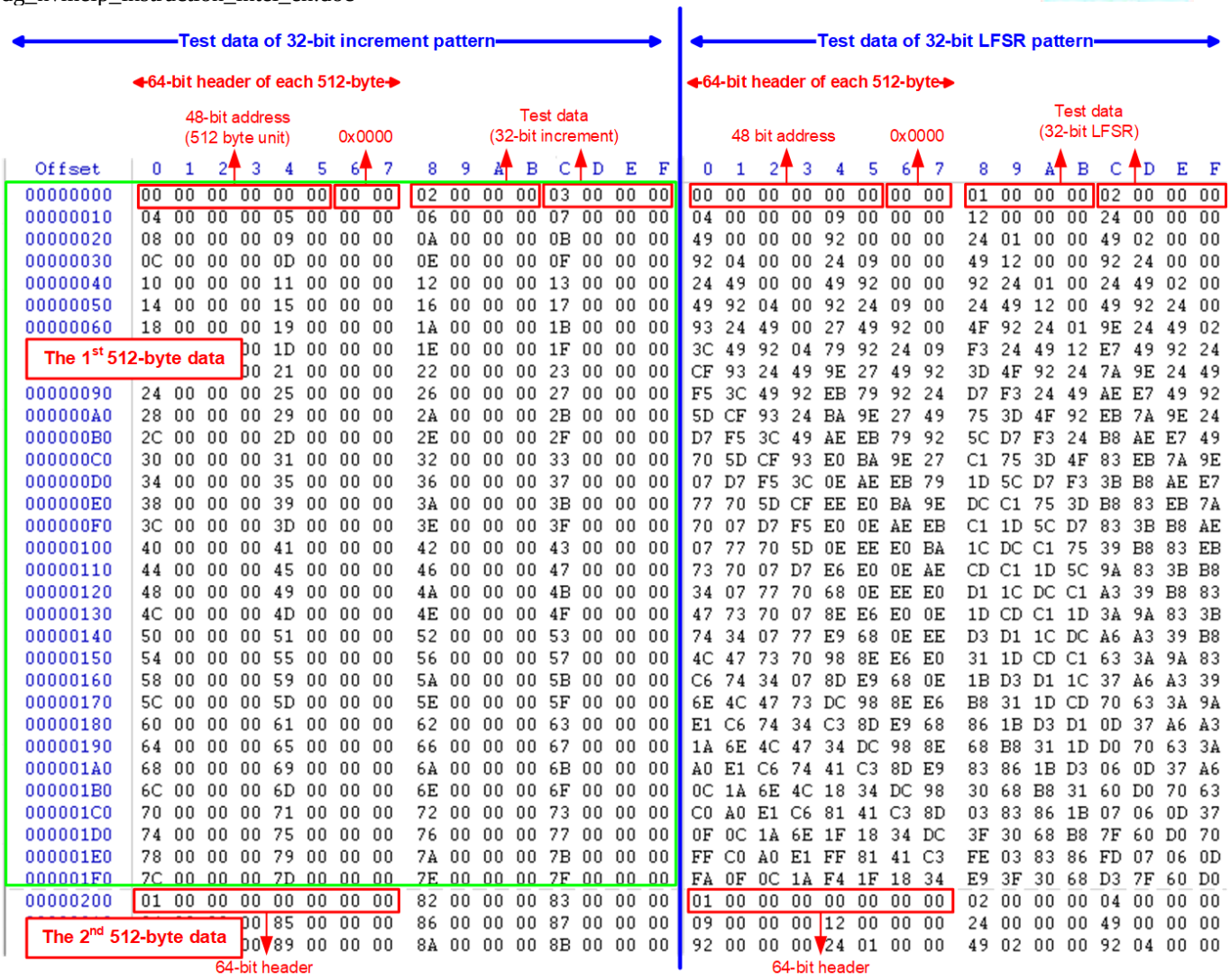

Figure 3-3 Example Test data of the 1st and 2nd 512-byte by using increment/LFSR pattern

Test data in SSD is split into 512-byte unit. For incremental, decremental, or LFSR pattern, each 512-byte data has unique 64-bit header consisting of 48-bit address (in 512-byte unit) and 16-bit zero value. The data after 64-bit header is the test pattern which is selected by user.

The left window of Figure 3-3 shows the example when using 32-bit incremental pattern while the right window shows the example when using 32 bit LFSR pattern. The unique header is not included when running all-0 or all-1 pattern.

When user runs Write or Read command with 4-Kbyte LBA SSD, there is the message displaying on the console to show the input limitation which must be aligned to 8, as shown in Figure 3-4. When the input does not align to 8, “Invalid input” is displayed and the operation is cancelled.

Also, Figure 3-5 shows the example when the input is out of the recommended range for each parameter. The console displays “Invalid input” and then the operation is cancelled.

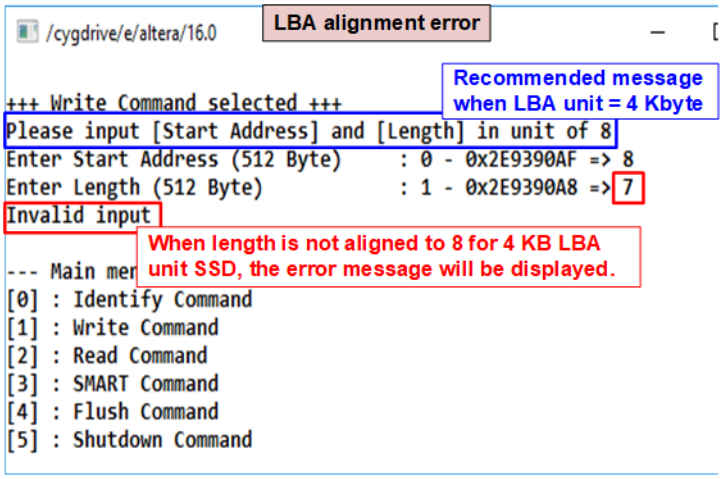

Figure 3-4 Error message when the input is unaligned for SSD with 4KB LBA unit

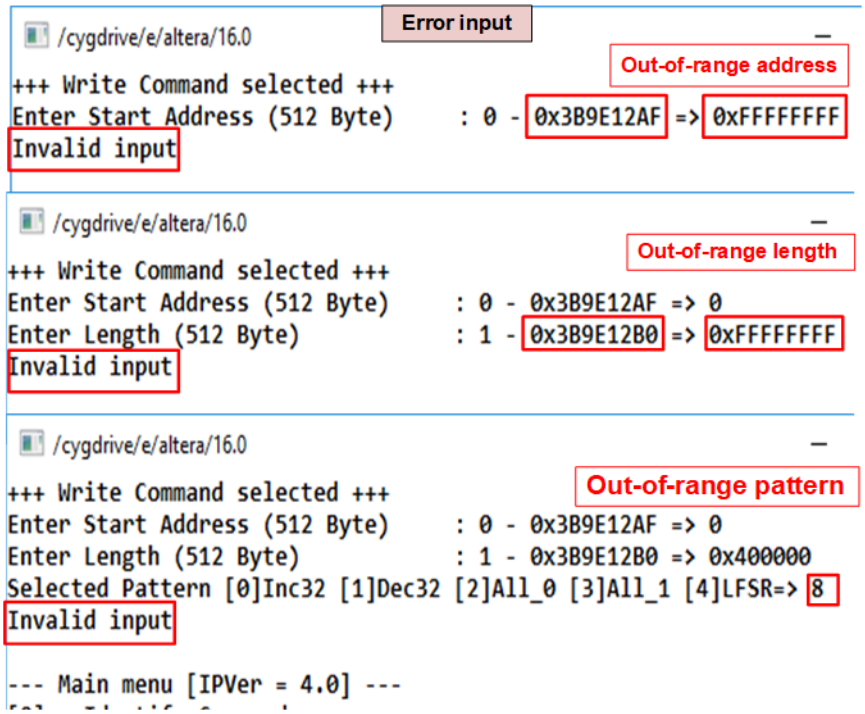

Figure 3-5 Error message from the invalid input

3.3 Read Command

Select ‘2’ to send Read command to NVMe SSD.

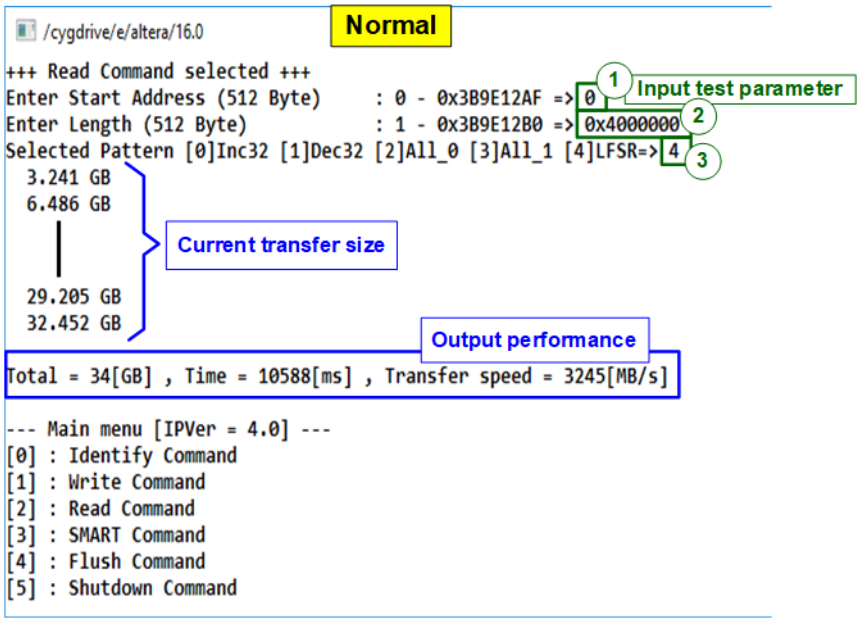

Figure 3-6 Input and result of Read Command menu

User inputs three parameters as follows.

1) Start Address: Input start address to read SSD as 512-byte unit. The input is decimal unit when user enters only digit number. User can add “0x” to be a prefix for hexadecimal unit. When LBA unit of SSD is 4 Kbyte, this input must be aligned to 8.
2) Transfer Length: Input total transfer size as 512-byte unit. The input is decimal unit when user enters only digit number. User can add “0x” to be a prefix for hexadecimal unit. If LBA unit of SSD is 4 Kbyte, this input must be aligned to 8.
3) Test pattern: Select test data pattern to verify data from SSD. Test pattern must be matched with the pattern using in Write Command menu. There are five patterns, i.e. 32-bit incremental, 32-bit decremental, all-0, al- 1, and 32-bit LFSR counter

Similar to Write command menu, test system reads data from SSD when all inputs are valid. During reading data, current transfer size is displayed on the console every second to show that system is still alive. Total size, total time usage, and test speed are displayed after finishing the operation.

“Invalid input” is displayed when some inputs are invalid or unaligned to 8 (when connecting to 4-KB LBA SSD).

Figure 3-7 shows error message when data verification is failed. “Verify fail” is displayed with the information of the 1st failure data, i.e. the error byte address, the expected value, and the read value.

User can press any key(s) to cancel read operation or wait until finishing Read command. Similar to the normal condition, the output performance is displayed on the console when the user does not enter any key(s) to stop the operation.

When cancelling the operation, the read command still runs as the background process. It is recommended to power-off/on AB18/AB16, and then press “RESET” button to restart system.

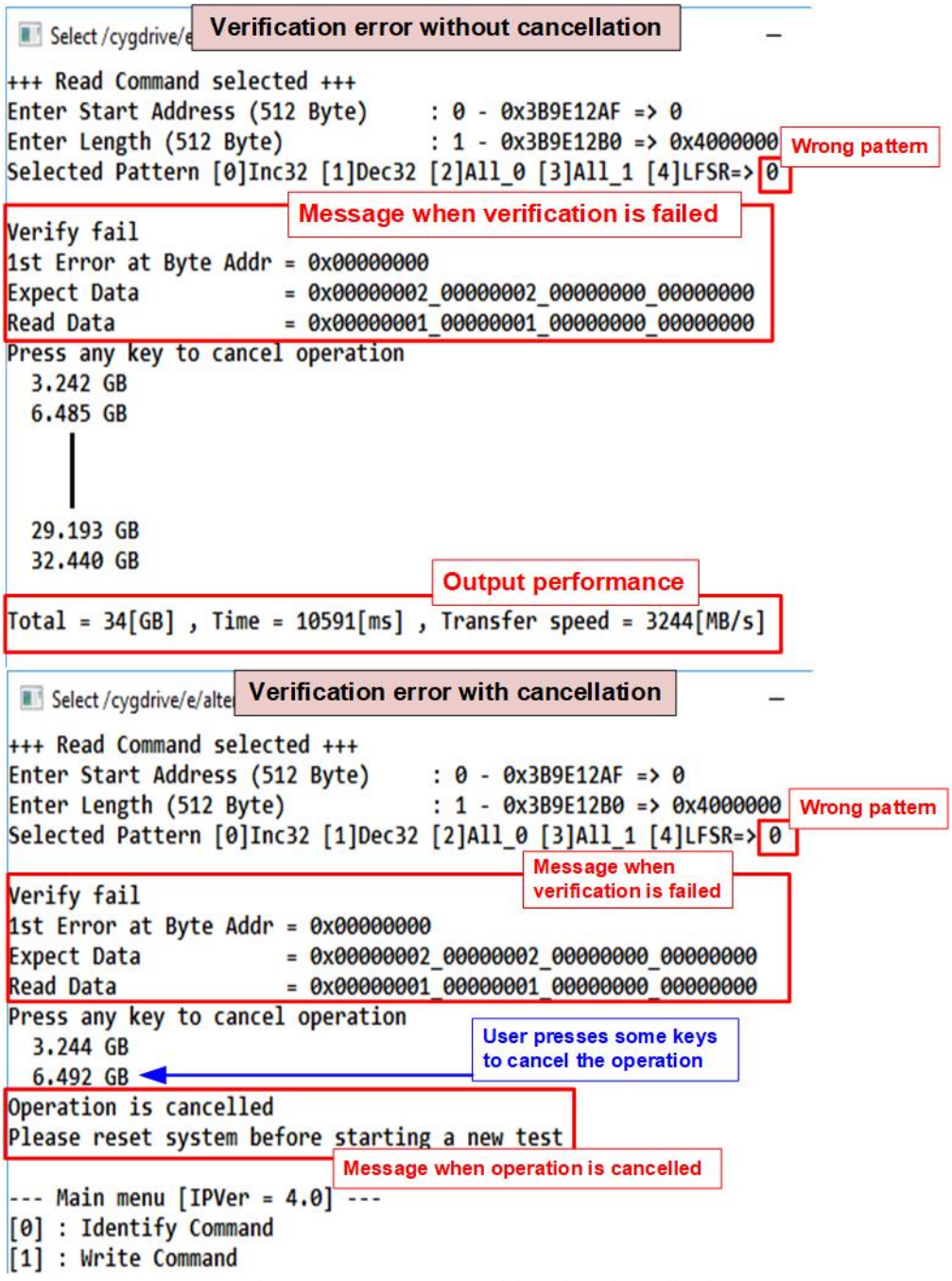

Figure 3-7 Data verification is failed

3.4 SMART Command

Select ‘3’ to send SMART command to NVMe SSD.

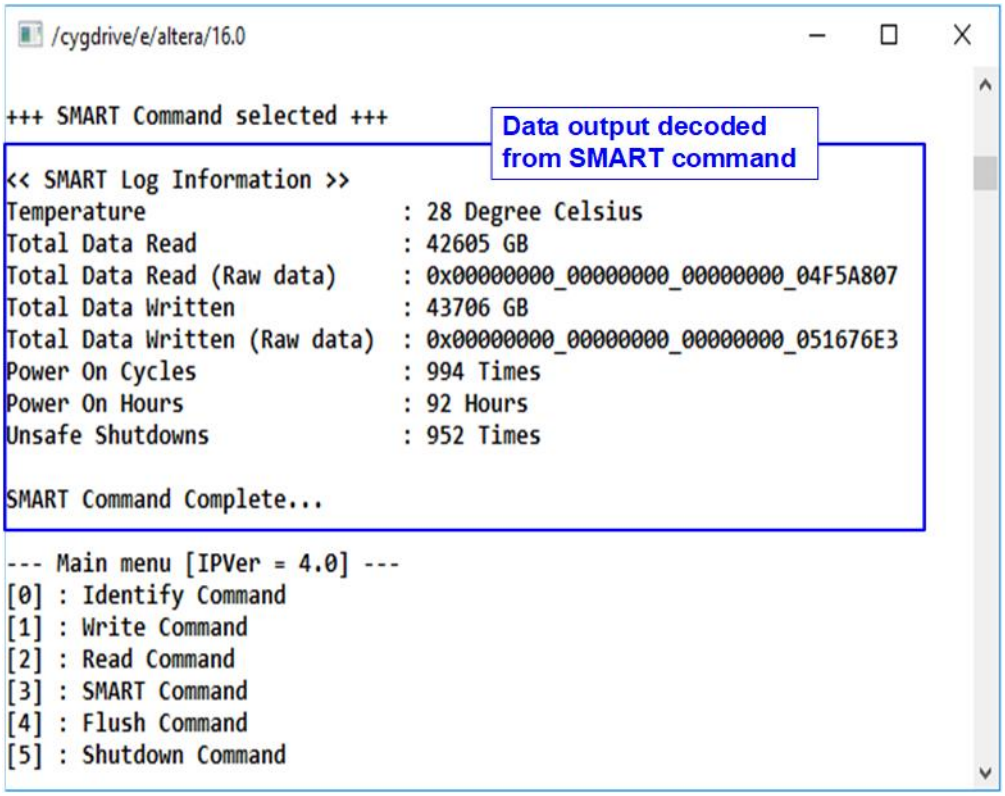

Figure 3-8 Test result when running SMART Command

When finishing the operation, SMART/Health Information (output from SMART command) is displayed as shown in Figure 3-8. The console shows six parameters, described as follows.

1) Temperature in °C unit.
2) Total Data Read decoded as GB/TB unit. Additionally, raw data without decoding is displayed in 128-bit hexadecimal unit. The unit size of raw data is 512,000 byte.
3) Total Data Written decoded as GB/TB unit. Additionally, raw data without decoding is displayed in 128-bit hexadecimal unit. The unit size of raw data is 512,000 byte.
4) Power On Cycles: Display the number of power cycles.
5) Power On Hours: Display period of time in hours to show how long the SSD has been powered on.
6) Unsafe Shutdowns: Display the number of unsafe shutdowns of SSD

3.5 Flush Command

Select ‘4’ to send Flush command to NVMe SSD.

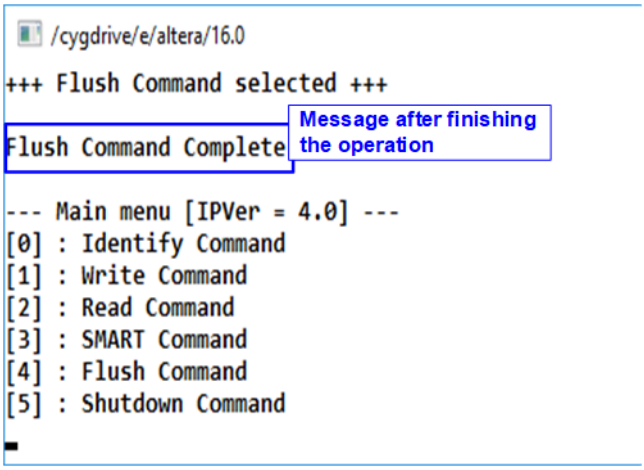

Figure 3-9 Test result when running Flush command

“Flush Command Complete” is displayed after the operation is completed.

3.6 Shutdown Command

Select ‘5’ to send Shutdown command to NVMe SSD.

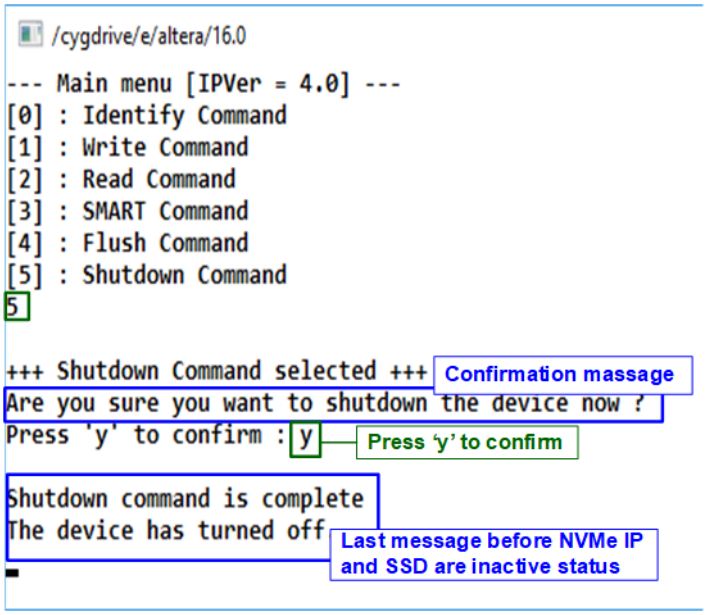

Figure 3-10 Shutdown Command with confirmation

The confirmation message is displayed on the console. User inputs ‘y’ or ‘Y’ to confirm the operation or inputs other keys to cancel the operation.

After finishing Shutdown operation, “Shutdown command is complete” is displayed on the console as the last message. Main menu is not displayed anymore. User needs to power off/on test system to start new test operation.

4 Revision History

Revision	Date	Description
1.0	9-Aug-16	Initial version release
1.1	17-Aug-16	Update message during write/read command
1.2	19-Dec-16	Update performance result of new buffer system
1.3	20-Mar-17	Support Arria10 GX Development board
1.4	9-May-17	Update SSD model and test result for new IP version
2.0	8-Jun-17	Support only 256 Kbyte buffer
2.1	31-Jul-17	Add LFSR pattern
2.2	25-Sep-17	Support Alaric board
2.3	27-Oct-17	Change PCIe connector on Alaric board
3.0	20-Jul-18	Support Shutdown, SMART, and Flush command
3.1	12-Feb-20	Support Cyclone10GX and AB18-PCIeX16 board
3.2	20-Apr-20	Remove PCIeSub board support of AB18