HCTL-IP RAID0x8 Demo Instruction

Rev1.0   8-Mar-18

This document describes the instruction to run 8-ch RAID0 by using SATA HCTL-IP on FPGA development board and AB09-FMCRAID board. The demo is designed to write/verify data with eight SATA-III devices. User controls test operation through Serial console.

1   Environment Setup

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

1) Supported FPGA Development board: ZCU102, KCU105
2) PC with Xilinx programmer software (Vivado) and Serial console software such as HyperTerminal
3) AB09-FMCRAID board, provided by Design Gateway
4) Power supply connecting to power connector on FMCRAID board to be SATA device power
5) Eight SATA-III devices connecting at CN0-CN7 of FMCRAID board
6) Xilinx Power adapter for Xilinx board
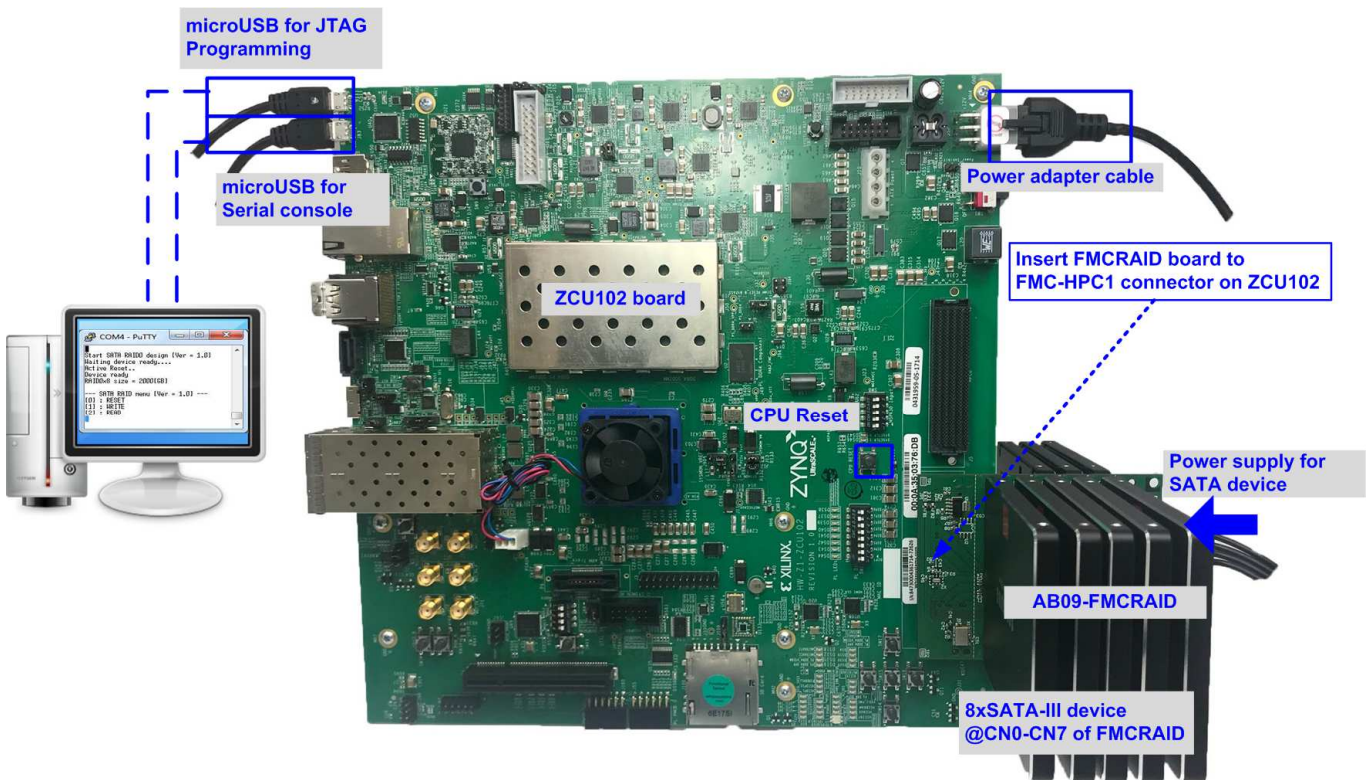7) Two micro USB cables for programming FPGA and Serial console connecting between FPGA Development board and PC
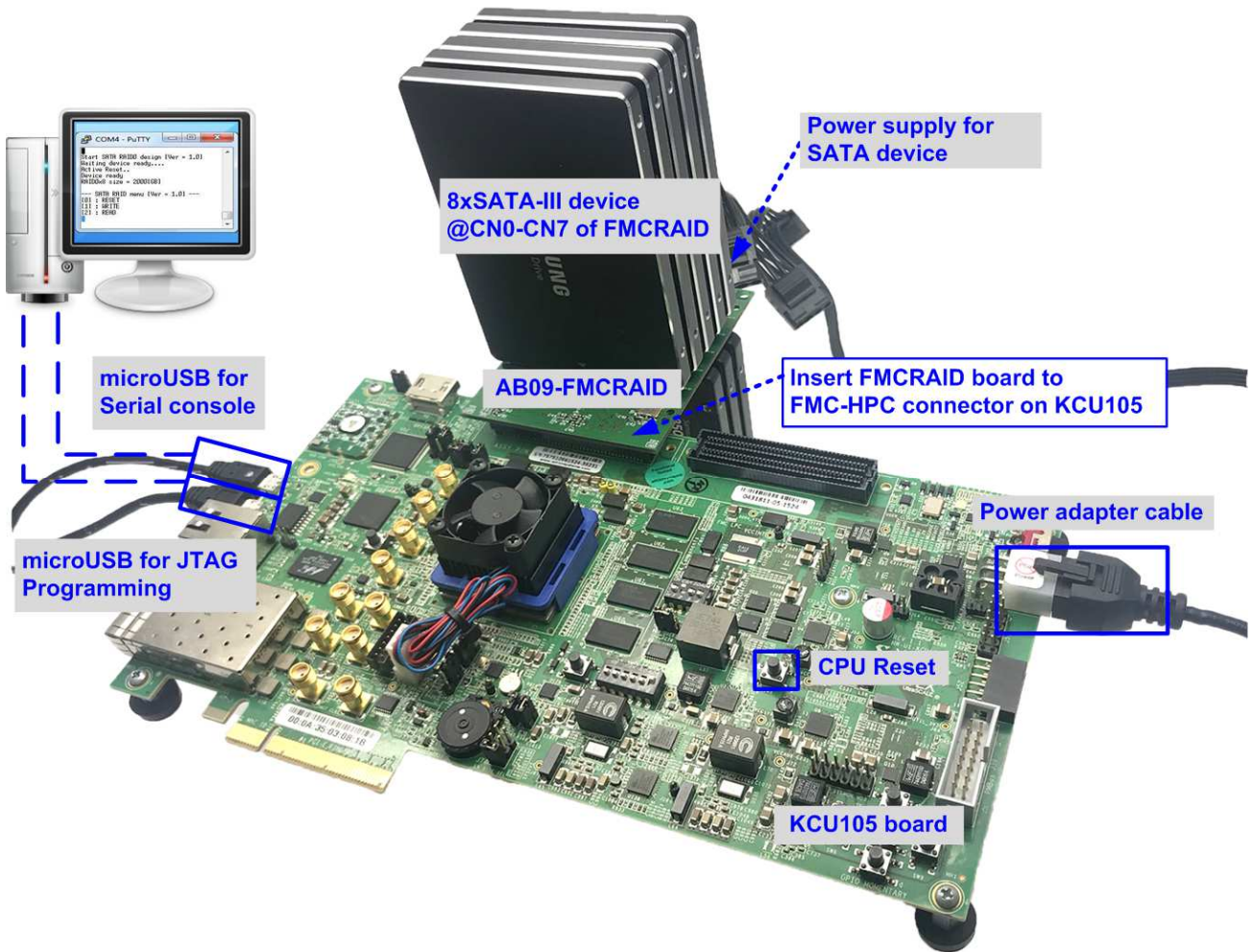

Figure 1-1 HCTL-IP RAID0 demo setup on ZCU102

Figure 1-2 HCTL-IP RAID0 demo setup on KCU105

2 Demo setup

1) Power off system.
2) For ZCU102 board, set SW6="0000" (SW = ON) to configure PS from JTAG, as shown in Figure 2-1.

Figure 2-1 SW6 setting to configure PS from JTAG on ZCU102

3) Connect AB09-FMCRAID board to FMC-HPC(1) connector on FPGA development board.
4) Connect eight SATA-III devices to CN0-CN7 on FMCRAID board.
5) Connect power to power connector on FMCRAID board.
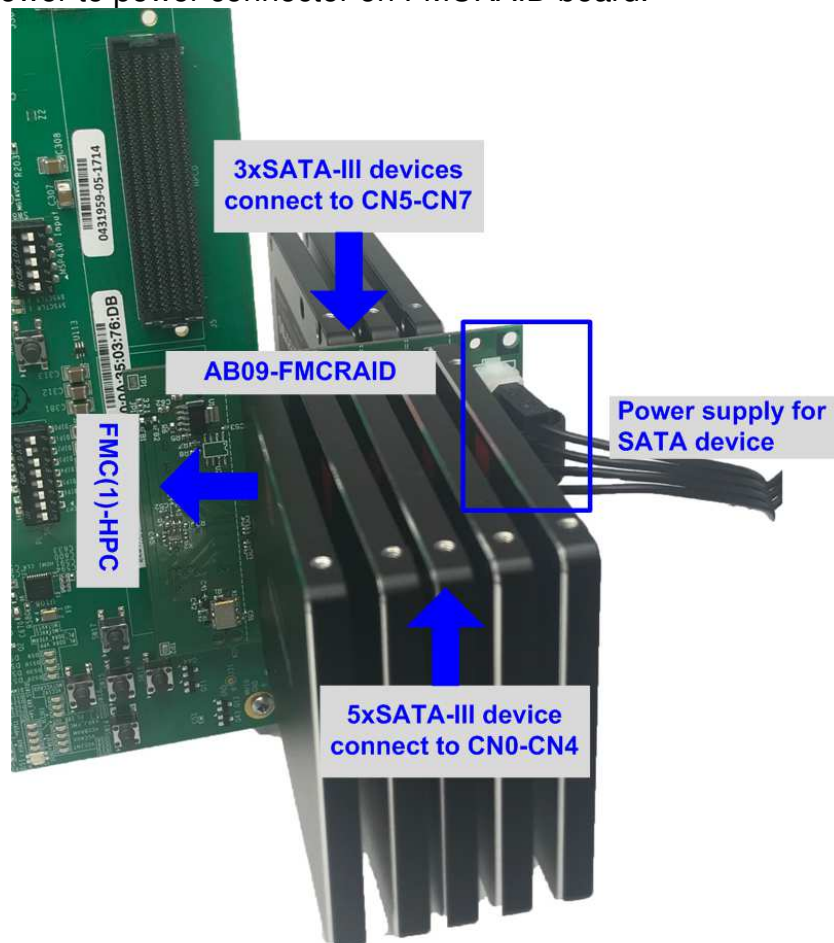

Figure 2-2 AB09-FMCRAID connection

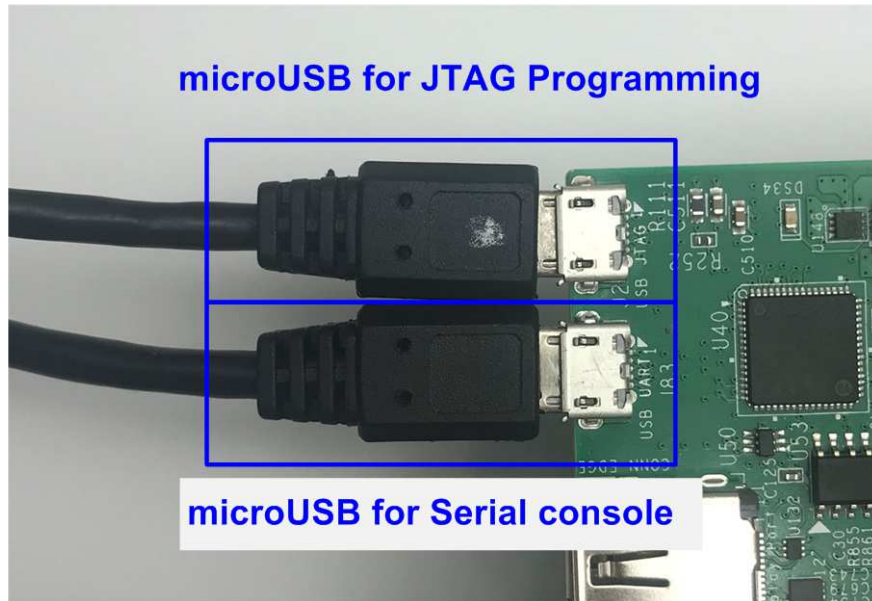6) Connect two micro USB cables from Xilinx development board to PC for JTAG programming and Serial console.

Figure 2-3 USB cable connection

7) Power on FPGA development board and power supply for SATA device.
8) Open Serial console such as TeraTerm, HyperTerminal and set Buad rate=115,200 Data=8 bit Non-Parity Stop=1.
9) Download configuration file and CPU firmware to FPGA
   a) For ZCU102 board, open Vivado TCL shell and change current directory to ready_for_download. Then, run xxx_HSATARaid0x8.bat as shown in Figure 2-4. For example, type zcu102_HSATARaid0x8.bat.
   b) For KCU105 board, program the configuration file which includes CPU firmware by using Vivado tool, as shown in Figure 2-5

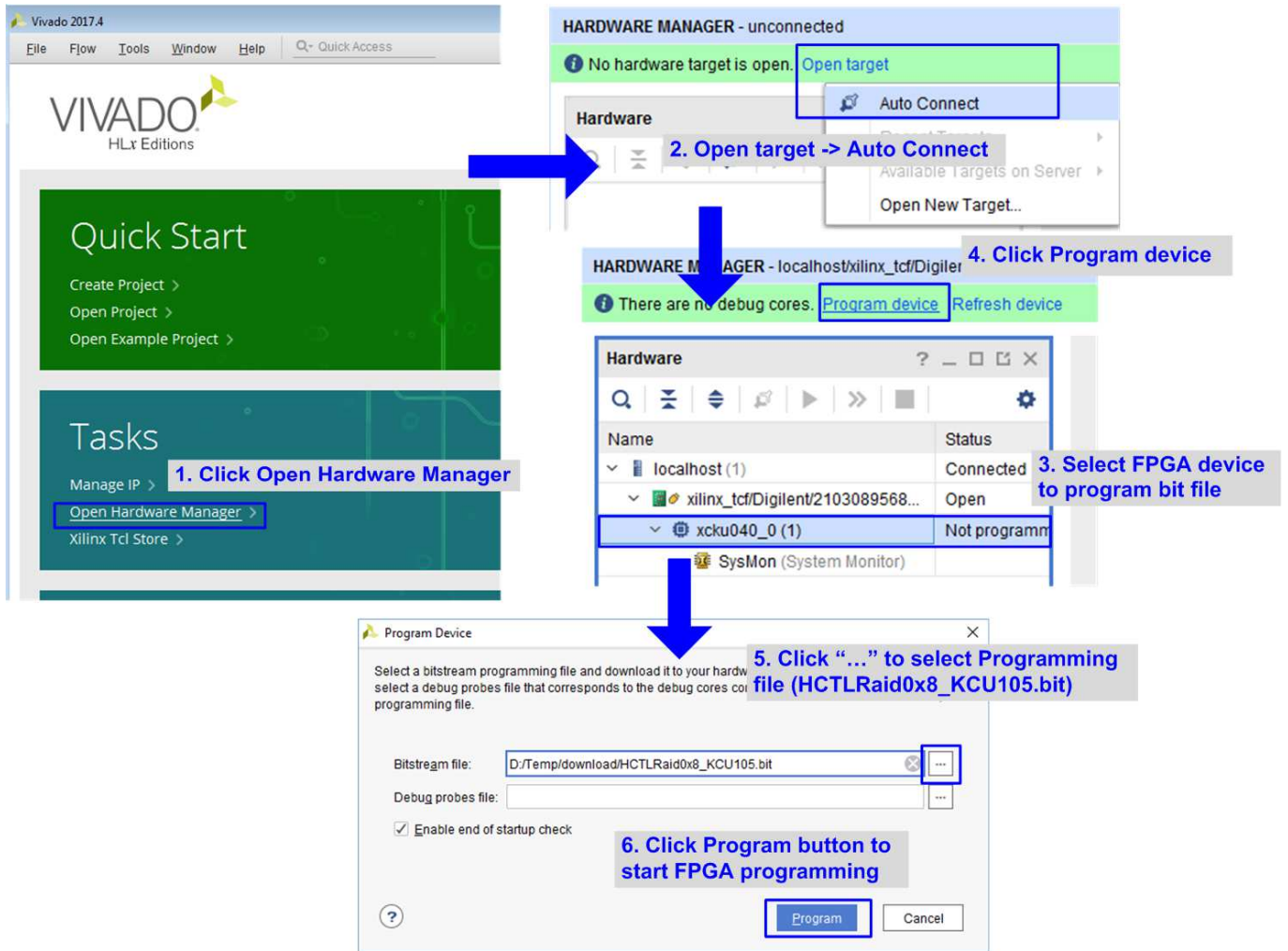Figure 2-4 Command script for download demo file to ZCU102 by Vivado tool

Figure 2-5 Program the configuration file to KCU105 by Vivado tool

10) Check LED status on Xilinx development board. The description of LED is shown in Table 2-1.

Table 2-1 LED Definition

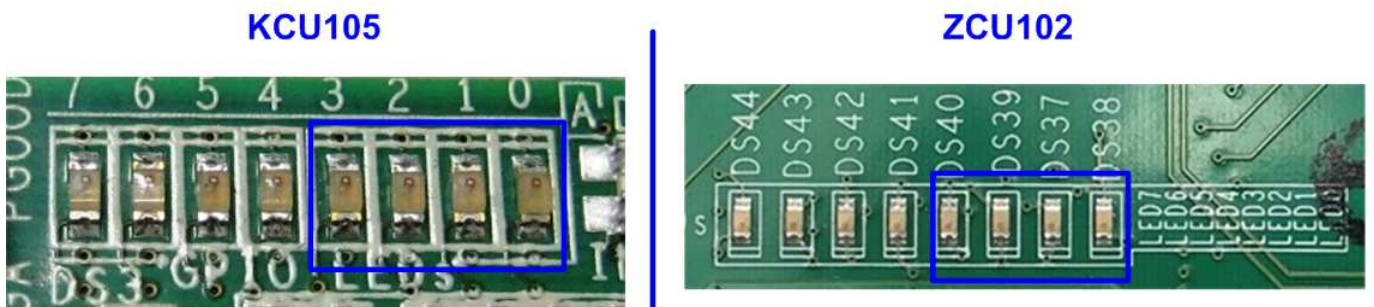| GPIO LED	ON	OFF
0	Normal operation	System is in reset condition
1	System is busy	Idle status
2	Error detect	Normal operation
3	Data verification fail	Normal operation

Figure 2-6 4-bit LED Status for user output

11) After programming completely, LED[0] and LED[1] are ON during RAID0 initialization process. Then, LED[1] changes to OFF to show that RAID0 completes initialization process and system is ready to receive command from user. After that, main menu is displayed as shown in Figure 2-8.

KCU105   **ZCU102**
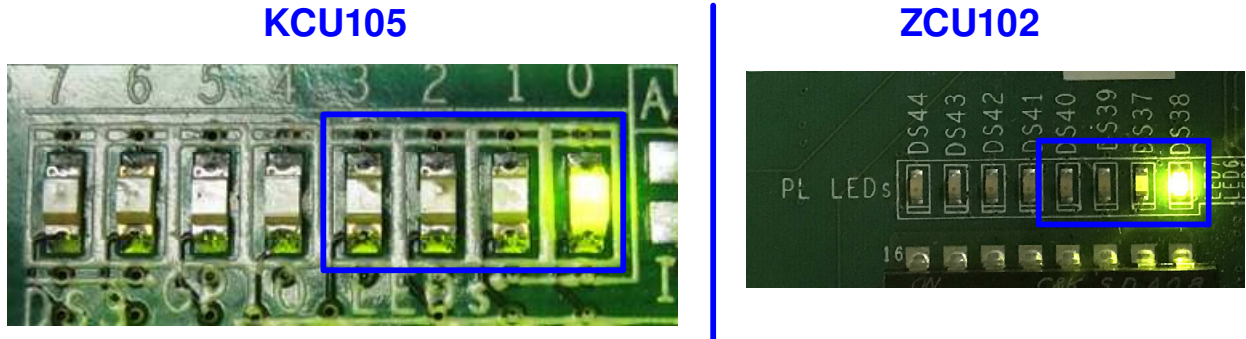
Figure 2-7 LED status after program configuration file and RAID0 initialization complete
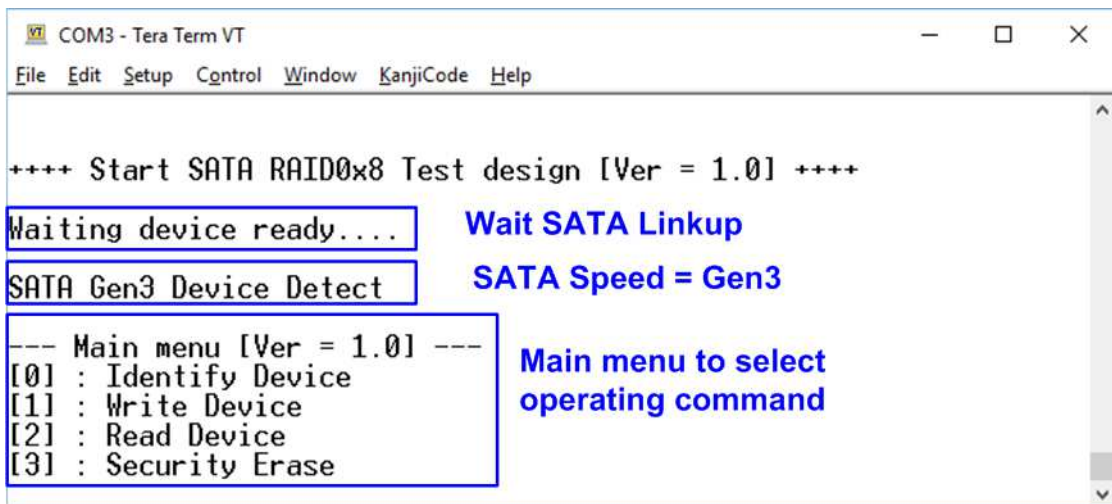

Figure 2-8 Main menu after program configuration file and RAID0 initialization complete

12) If some SATA devices are not detected, "CH[0-7] Not Detect" will be displayed as shown in Figure 2-9. CH[0]-[7] is referred to SATA channel which has found the error. Please check SATA device in error channel.
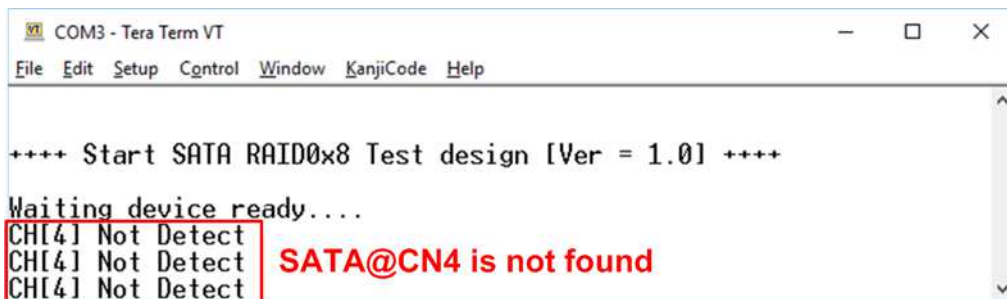
Figure 2-9 Error message when some devices are not detected
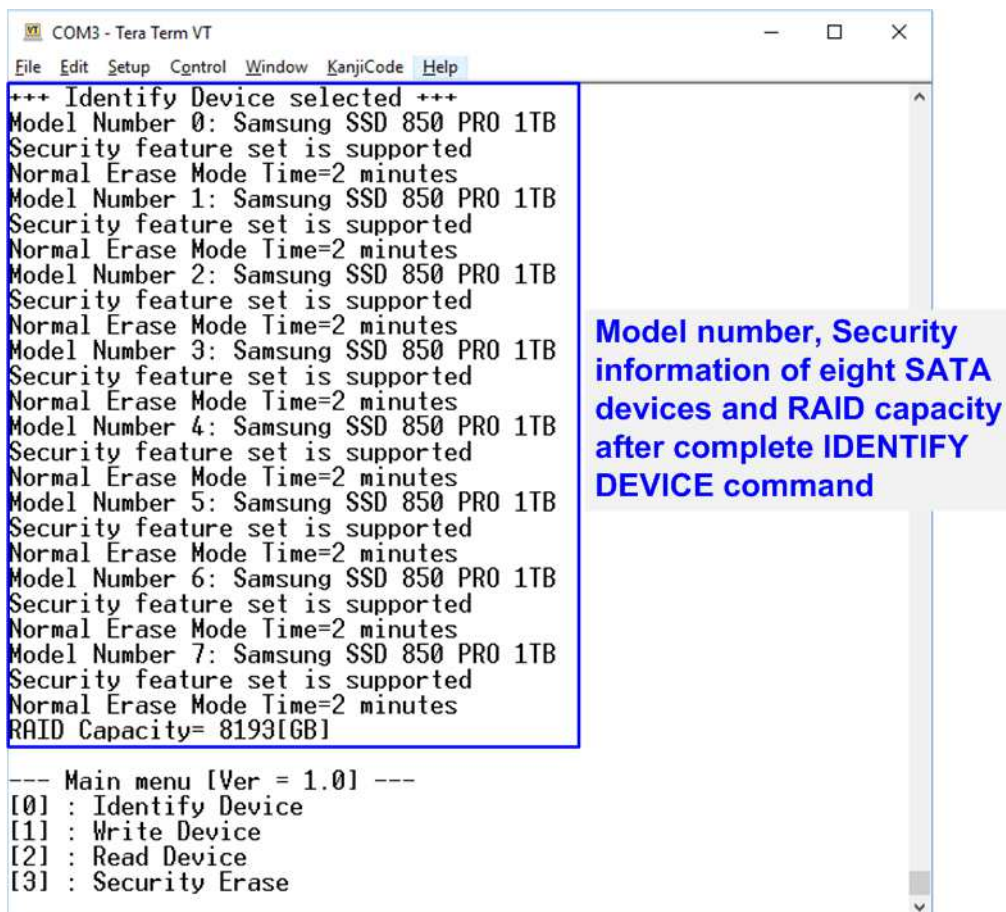
3 Test Menu

3.1 Identify Device

Select '0' to send Identify device command to RAID0. When operation is completed, all information is displayed on the console, i.e.
1) SSD model number
2) Security feature set is supported or not. If some devices are not supported, user must not run menu 3 to test Security erase command.
3) Normal Erase Mode Time: This is estimation time to complete Security erase command. Minimum valid value is 2 minutes. This information is displayed when the device supports Security feature set.
4) SSD capacity which is output value from RAID0 block. The value is equal to eight times of SATA CH#0 capacity.
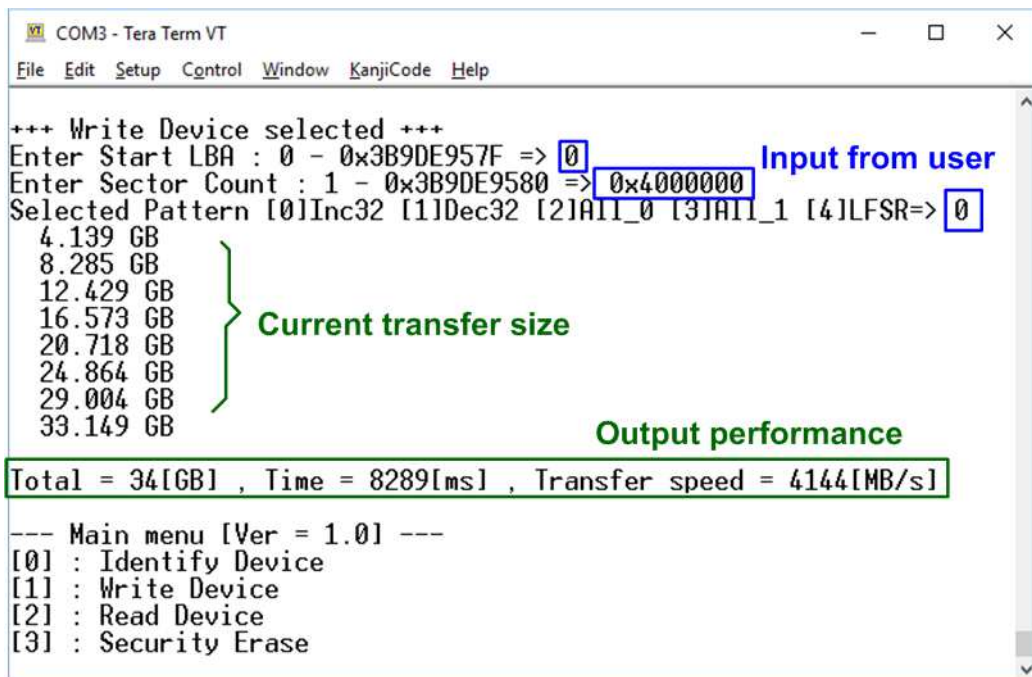
Figure 3-1 Result from Identify Device menu

3.2 Write Device

Select '1' to send Write command to RAID0. Three inputs are required for this menu.
1) Start LBA: Input start address of RAID0 in sector unit. The input is decimal unit when input only digit number. User can add "0x" to be prefix when input is hexadecimal unit.
2) Sector Count: Input total transfer size in sector unit. The input is decimal unit when input only digit number. User can add "0x" to be prefix when input is hexadecimal unit.
3) Test pattern: Select test pattern of test data for writing to RAID0. Five types can be selected, i.e. 32-bit increment, 32-bit decrement, all 0, all 1, and 32-bit LFSR counter.

As shown in Figure 3-2, if all inputs are valid, the operation will be started. During writing data, current transfer size is displayed to the console to show that system still be alive. Finally, test performance, total size, and total time usage are displayed on the console as test result.

Figure 3-2 Input and result of Write Device menu

←—64-bit header of each sector—→

48-bit LBA Address = 0   0x0000   32-bit Increment pattern   48-bit LBA Address = 1

Dev#0 (48-bit LBA Address = 0)
```
Offset     0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
0000000000 00 00 00 00 00 00 00 00 02 00 00 00 03 00 00 00
0000000010 04 00 00 00 05 00 00 00 06 00 00 00 07 00 00 00
0000000020 08 00 00 00 09 00 00 00 0A 00 00 00 0B 00 00 00
0000000030 0C 00 00 00 0D 00 00 00 0E 00 00 00 0F 00 00 00
0000000040 10 00 00 00 11 00 00 00 12 00 00 00 13 00 00 00
0000000050 14 00 00 00 15 00 00 00 16 00 00 00 17 00 00 00
                           48-bit LBA Address = 8
00000001A0 68 00 00 00 69 00 00 00 6A 00 00 00 6B 00 00 00
00000001B0 6C 00 00 00 6D 00 00 00 6E 00 00 00 6F 00 00 00
00000001C0 70 00 00 00 71 00 00 00 72 00 00 00 73 00 00 00
00000001D0 74 00 00 00 75 00 00 00 76 00 00 00 77 00 00 00
00000001E0 78 00 00 00 79 00 00 00 7A 00 00 00 7B 00 00 00
00000001F0 7C 00 00 00 7D 00 00 00 7E 00 00 00 7F 00 00 00
0000000200 08 00 00 00 00 00 00 00 02 04 00 00 03 04 00 00
0000000210 04 04 00 00 05 04 00 00 06 04 00 00 07 04 00 00
```

Dev#1 (48-bit LBA Address = 1)
```
Offset     0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
0000000000 01 00 00 00 00 00 00 00 82 00 00 00 83 00 00 00
0000000010 84 00 00 00 85 00 00 00 86 00 00 00 87 00 00 00
0000000020 88 00 00 00 89 00 00 00 8A 00 00 00 8B 00 00 00
0000000030 8C 00 00 00 8D 00 00 00 8E 00 00 00 8F 00 00 00
0000000040 90 00 00 00 91 00 00 00 92 00 00 00 93 00 00 00
0000000050 94 00 00 00 95 00 00 00 96 00 00 00 97 00 00 00
00000001A0 E8 00 00 00 E9 00 00 00 EA 00 00 00 EB 00 00 00
00000001B0 EC 00 00 00 ED 00 00 00 EE 00 00 00 EF 00 00 00
00000001C0 F0 00 00 00 F1 00 00 00 F2 00 00 00 F3 00 00 00
00000001D0 F4 00 00 00 F5 00 00 00 F6 00 00 00 F7 00 00 00
00000001E0 F8 00 00 00 F9 00 00 00 FA 00 00 00 FB 00 00 00
00000001F0 FC 00 00 00 FD 00 00 00 FE 00 00 00 FF 00 00 00
0000000200 09 00 00 00 00 00 00 00 82 04 00 00 83 04 00 00
0000000210 84 04 00 00 85 04 00 00 86 04 00 00 87 04 00 00
```

Dev#6 (48-bit LBA Address = 6)
```
Offset     0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
0000000000 06 00 00 00 00 00 00 00 02 03 00 00 03 03 00 00
0000000010 04 03 00 00 05 03 00 00 06 03 00 00 07 03 00 00
0000000020 08 03 00 00 09 03 00 00 0A 03 00 00 0B 03 00 00
0000000030 0C 03 00 00 0D 03 00 00 0E 03 00 00 0F 03 00 00
0000000040 10 03 00 00 11 03 00 00 12 03 00 00 13 03 00 00
0000000050 14 03 00 00 15 03 00 00 16 03 00 00 17 03 00 00
00000001A0 68 03 00 00 69 03 00 00 6A 03 00 00 6B 03 00 00
00000001B0 6C 03 00 00 6D 03 00 00 6E 03 00 00 6F 03 00 00
00000001C0 70 03 00 00 71 03 00 00 72 03 00 00 73 03 00 00
00000001D0 74 03 00 00 75 03 00 00 76 03 00 00 77 03 00 00
00000001E0 78 03 00 00 79 03 00 00 7A 03 00 00 7B 03 00 00
00000001F0 7C 03 00 00 7D 03 00 00 7E 03 00 00 7F 03 00 00
0000000200 0E 00 00 00 00 00 00 00 02 07 00 00 03 07 00 00
0000000210 04 07 00 00 05 07 00 00 06 07 00 00 07 07 00 00
```

Dev#7 (48-bit LBA Address = 7)
```
Offset     0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
0000000000 07 00 00 00 00 00 00 00 82 03 00 00 83 03 00 00
0000000010 84 03 00 00 85 03 00 00 86 03 00 00 87 03 00 00
0000000020 88 03 00 00 89 03 00 00 8A 03 00 00 8B 03 00 00
0000000030 8C 03 00 00 8D 03 00 00 8E 03 00 00 8F 03 00 00
0000000040 90 03 00 00 91 03 00 00 92 03 00 00 93 03 00 00
0000000050 94 03 00 00 95 03 00 00 96 03 00 00 97 03 00 00
00000001A0 E8 03 00 00 E9 03 00 00 EA 03 00 00 EB 03 00 00
00000001B0 EC 03 00 00 ED 03 00 00 EE 03 00 00 EF 03 00 00
00000001C0 F0 03 00 00 F1 03 00 00 F2 03 00 00 F3 03 00 00
00000001D0 F4 03 00 00 F5 03 00 00 F6 03 00 00 F7 03 00 00
00000001E0 F8 03 00 00 F9 03 00 00 FA 03 00 00 FB 03 00 00
00000001F0 FC 03 00 00 FD 03 00 00 FE 03 00 00 FF 03 00 00
0000000200 0F 00 00 00 00 00 00 00 82 07 00 00 83 07 00 00
0000000210 84 07 00 00 85 07 00 00 86 07 00 00 87 07 00 00
```

Figure 3-3 Example Test data in sector#0/#1 of Dev#0/#1/#6/#7 by increment 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 increment pattern (as shown in Figure 3-3).

Stripe size in 8-ch RAID0 demo is 1 sector (512-byte). So, LBA address in the header of 1st sector in Disk#0 - Disk#7 are equal to 0 - 7 sequentially. The address in the header of the next sector for Disk#0 is 8.

Figure 3-4 – Figure 3-6 show error message when user input is invalid. "Invalid input" is displayed on the console. Then, it returns to main menu to receive new command.

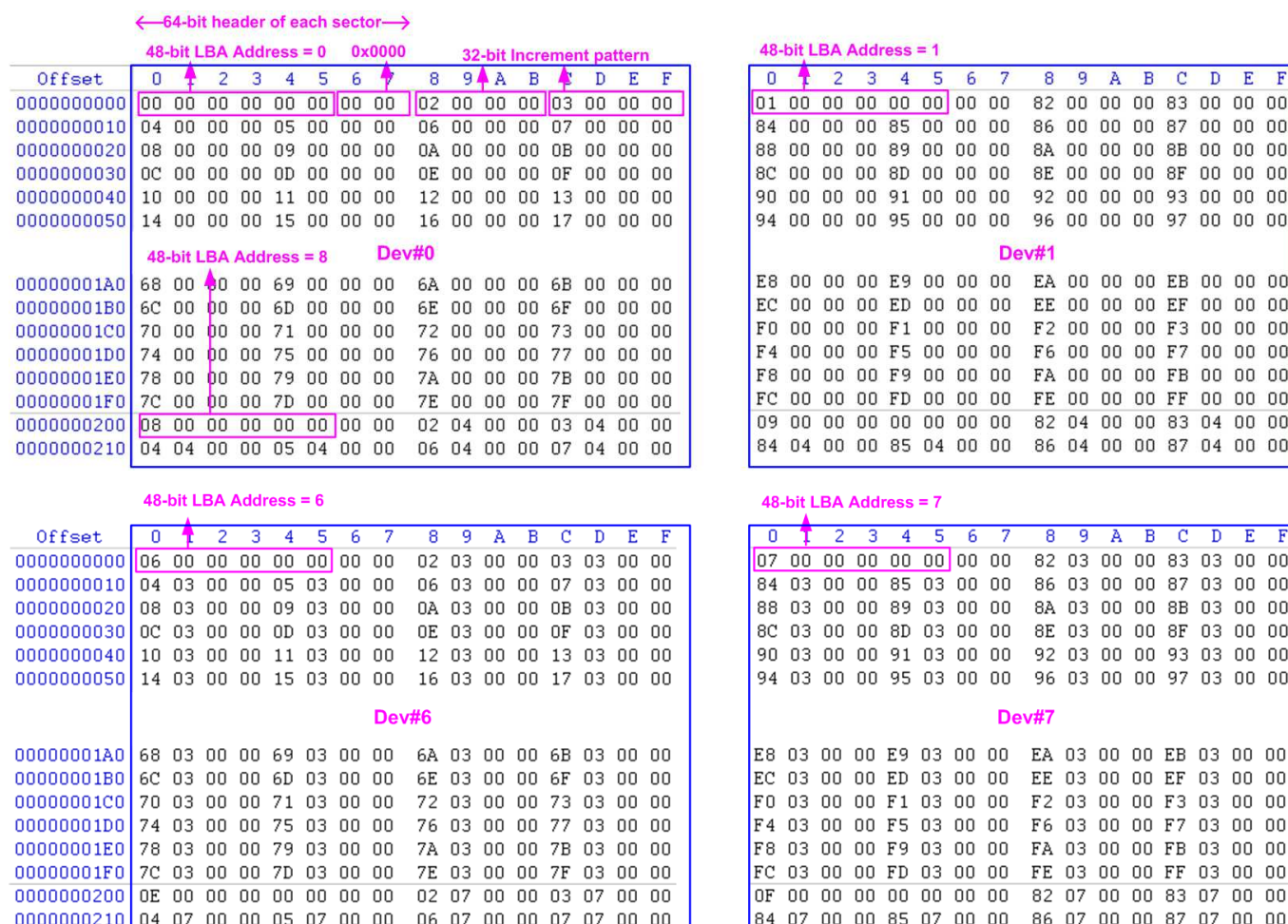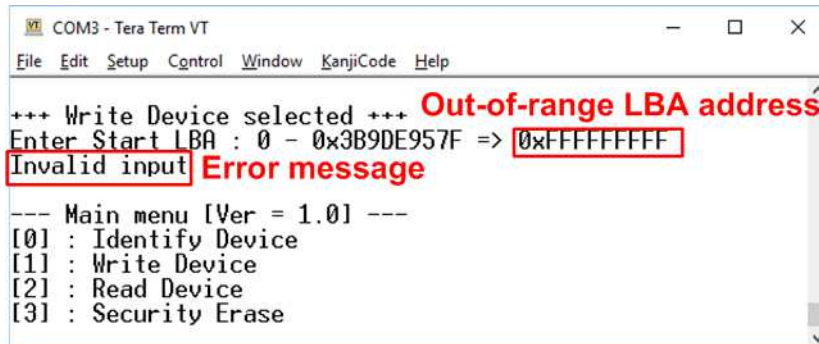Figure 3-4 Invalid Start LBA input
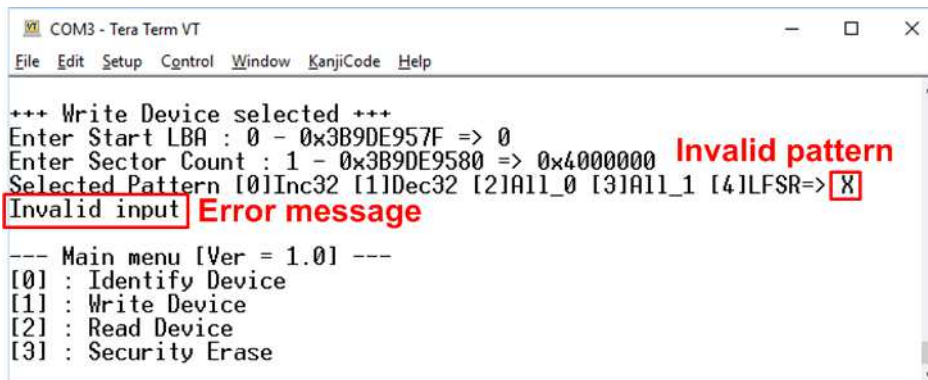

Figure 3-5 Invalid Sector count input

Figure 3-6 Invalid Test pattern input

3.3   Read Device

Select '2' to send Read command to RAID0. Three inputs are required for this menu.
1) Start LBA: Input start address of RAID0 in sector unit. The input is decimal unit when input only digit number. User can add "0x" to be prefix when input is hexadecimal unit.
2) Sector Count: Input total transfer size in sector unit. The input is decimal unit when input only digit number. User can add "0x" to be prefix when input is hexadecimal unit.
3) Test pattern: Select test pattern to verify data from RAID0. Test pattern must be matched with the test pattern which is used during write test. Five types can be selected, i.e. 32-bit increment, 32-bit decrement, all 0, all 1, and 32-bit LFSR counter.

Similar to write test if all inputs are valid, test system will read data from RAID0. Test performance, total size, and total time usage are displayed after end of transfer. "Invalid input" will be displayed if some inputs are out-of-range.

Figure 3-7 Input and result of Read Device menu

Figure 3-8 and Figure 3-9 show the error message when data verification is failed. "Verify fail" is displayed with error address, expected data, and read data. User can press any key to cancel read operation or wait until all read process complete.
If read process is completed, output performance from read process will be displayed.

In case of cancel operation, the previous command does not complete in good sequence. It is recommended to power-off/on all SATA devices and press "RESET" button to restart system.
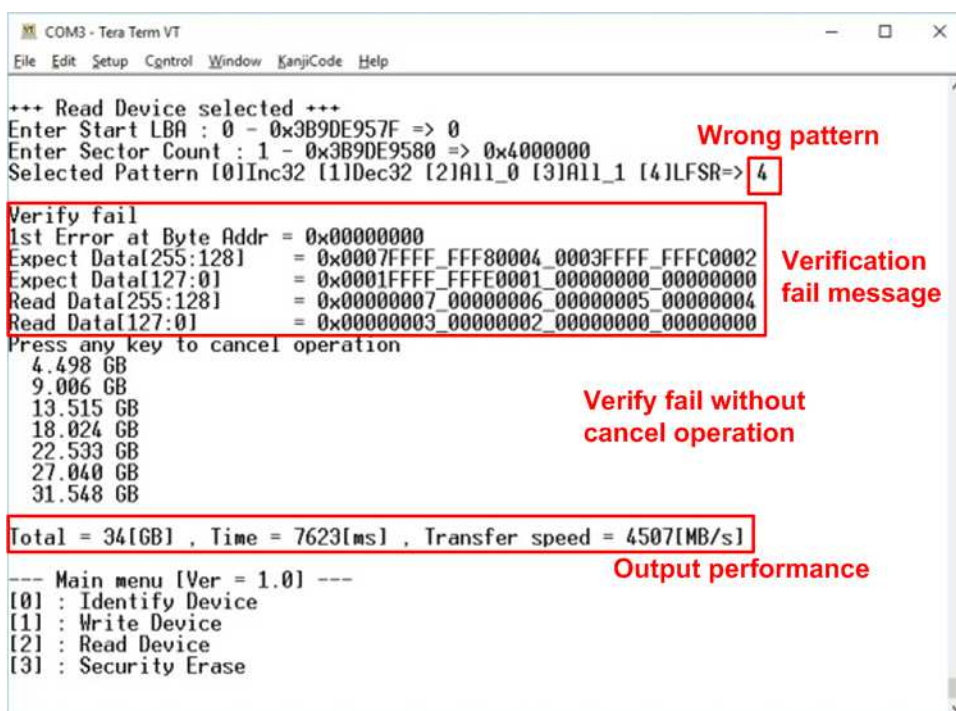

Figure 3-8 Data verification is failed but wait until read complete

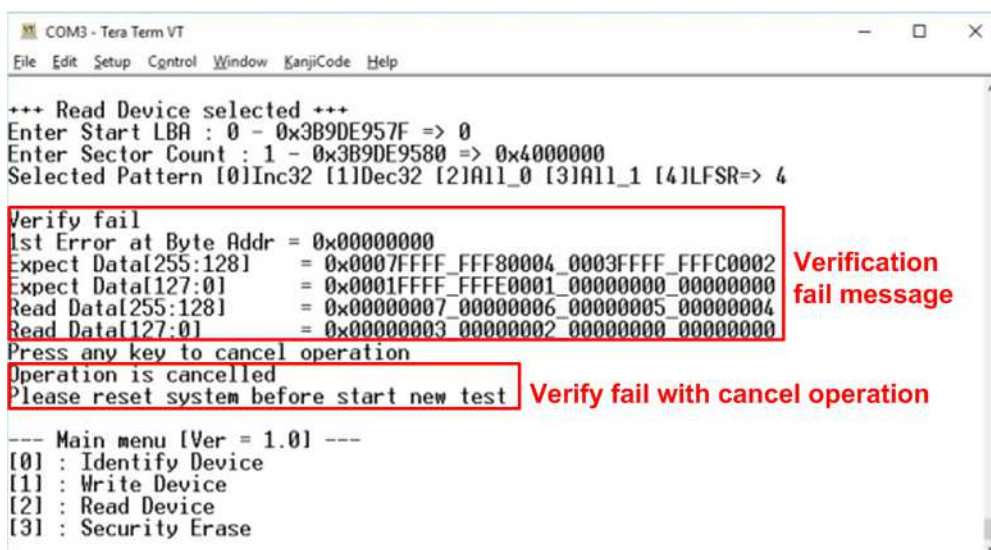Figure 3-9 Data verification is failed and press any key to cancel operation

3.4 Security Erase

Select '3' to send Security Erase command to RAID0. Please confirm that all SATA devices support Security Erase feature by using Identify device menu. The estimation of operation time to run security erase is also displayed in Identify device menu.

After selecting the menu, warning message is displayed on the console. User input 'y' or 'Y' to continue security erase operation or input other keys to cancel operation.

Number 0-9 is displayed on the console every second to show that system still be alive. Finally, total time usage is displayed as shown in Figure 3-10.

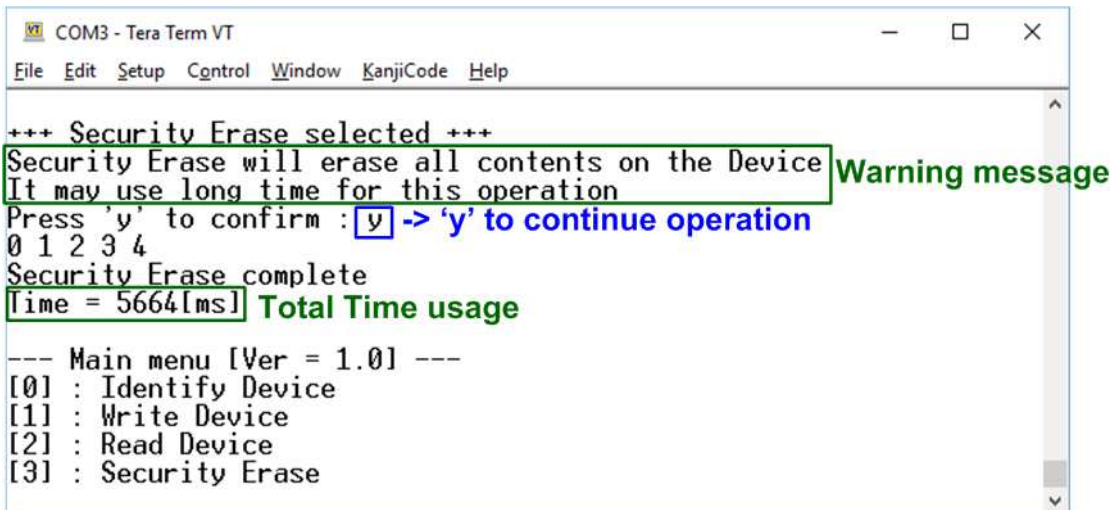Figure 3-11 shows the example when user inputs other keys to cancel the command.

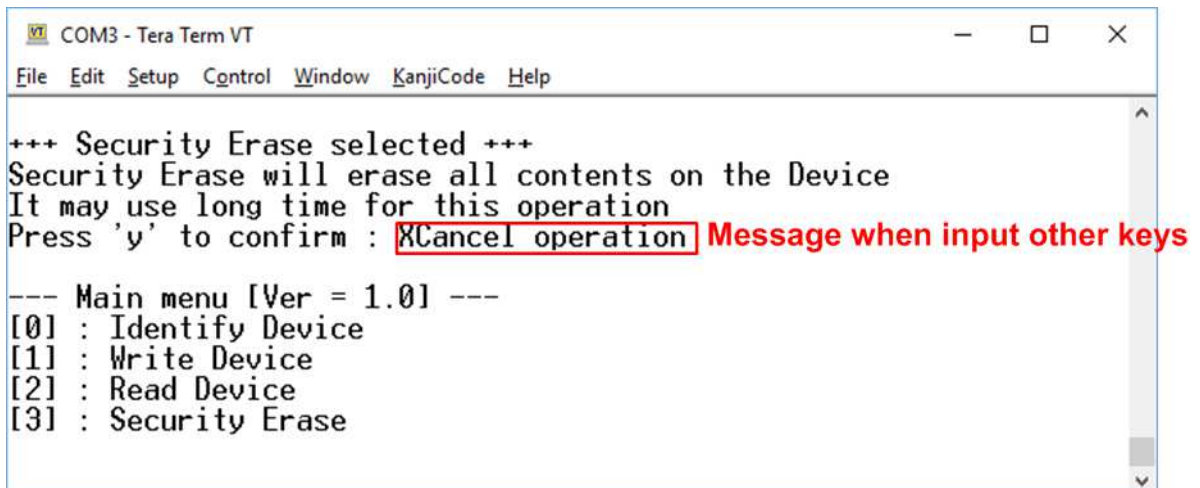Figure 3-10 Result from Security Erase command

Figure 3-11 Cancel Security Erase command

4   Revision History

Revision	Date	Description
1.0	8-Mar-18	Initial version release