How to implement SSD only correspond uniquely to the Host

Considering on the strict data security, customers ask the SSD to correspond uniquely to the Host. That is to say, the SSD only can be used in the specified host. Once it is connected to other host PC, the data destruction triggered automatically. Meanwhile, the specified host must be able to use the non-customized SSD from any other company.

I. Project requirement

(1) Host interface: SATA 6.0Gbps
(2) Form factor: 2.5”
(3) Capacity: 4TB
(4) NAND Flash: MLC
(5) Operation temperature: -40°C ~ +85°C
(6) SSD must to be correspond uniquely to the host
(7) Once been connected to other host PC, the data destruction triggered automatically
(8) The host must be able to use the SSD from any other company

II. Renice solution design

(1) Renice’s solution adopts own research & development SSD Controller RS3502-IT, SATAIII 6.0Gbps interface. The single controller supports max. capacity up to 2TB. So here use 2pcs Controller to make RAID0.
(2) Adopts Xilinx FPGA as RAID0 chip, TRIM supports.
(*The RAID chip on the market normally with below two drawbacks: ① do not support extend-temp. operation; ② do not support TRIM, that caused the SSD can only run GC after SSD is full of data. During GC process, the write performance would be sharply dropped even to 0MB, which may lead to failure in critical applications.*)
(3) Self-destruction: Two method support, physical destruction and logical destroy (optional or both support).
(4) Logical destroy supports quick erase in 10s. With this method, the SSD is re-usable after initialization (all of the data have been erased); Or after logical erase, data cannot be recovered, SSD can be used after returning to manufacturer for re-planting firmware.
(5) Physical destroy: if the SSD is connected to illegal host, the NAND flash would be breakdown by high-voltage. There is no way to recovery data information.

III. Detailed of implementation

(1) The SSD only correspond uniquely to specified host

There is no change on the Pin definition of customer’s host SATA interface, to ensure the host is able to operate with other common SSDs. In Renice SSD solution there adds one FPGA chipset or SCM to achieve signal matching by setting the un-definited Pin of SATA disk and the

characteristics of the mainboard. If the signal matching failed in 3s (here the trigger time is set by cusotmers), the data destruction will be triggered.

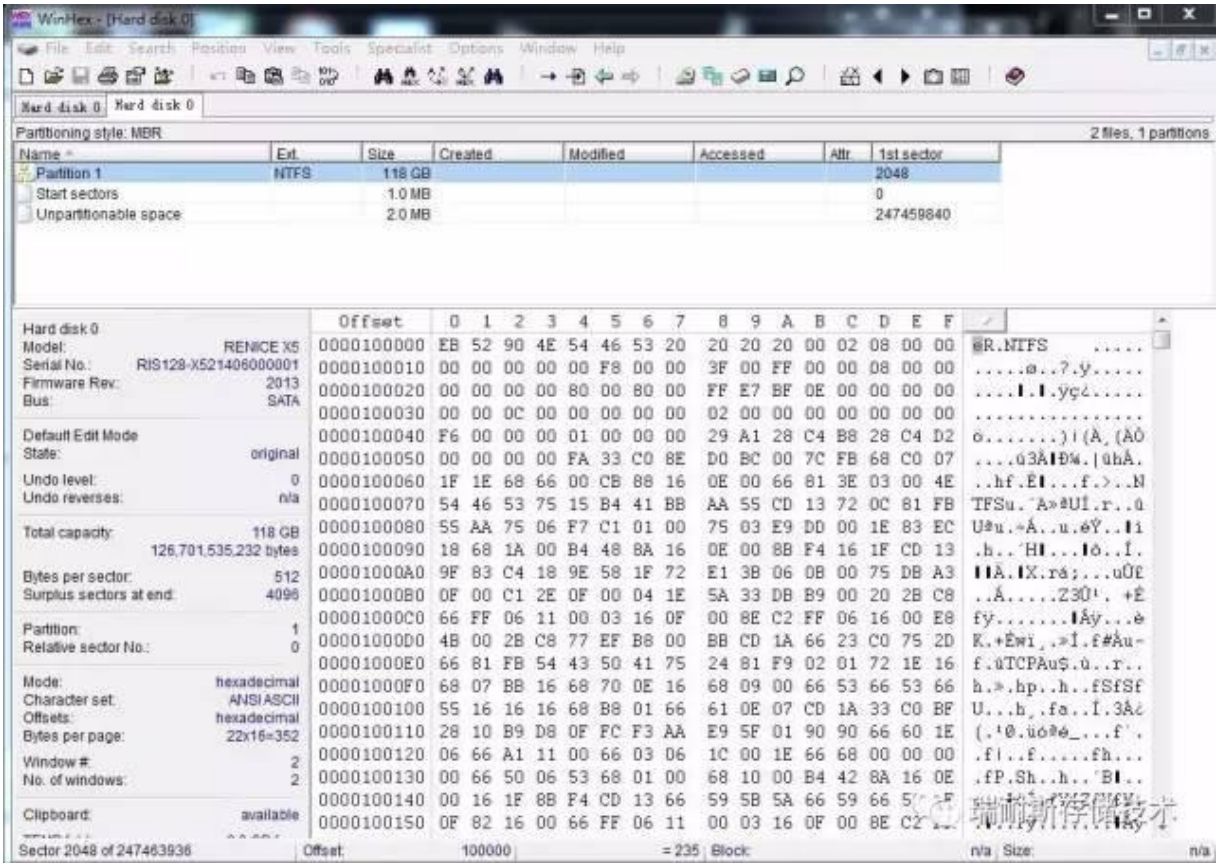
(2) Implementation and effect of the self-destruction
The SE is triggered by pulling down the voltage of GPIO signal and sending the command to Controller. The operation could be realized by HW or SW.

2.1 Logical destroy
The logical destroy refers to the operation of deleting or overwriting the data information, which include the AES key, Firmware, Mapping table, etc.. The logical destroy does not damage the storage media.
In general, customers have the requirement of fast delete as follow list, each requirement is implemented by different firmware means.

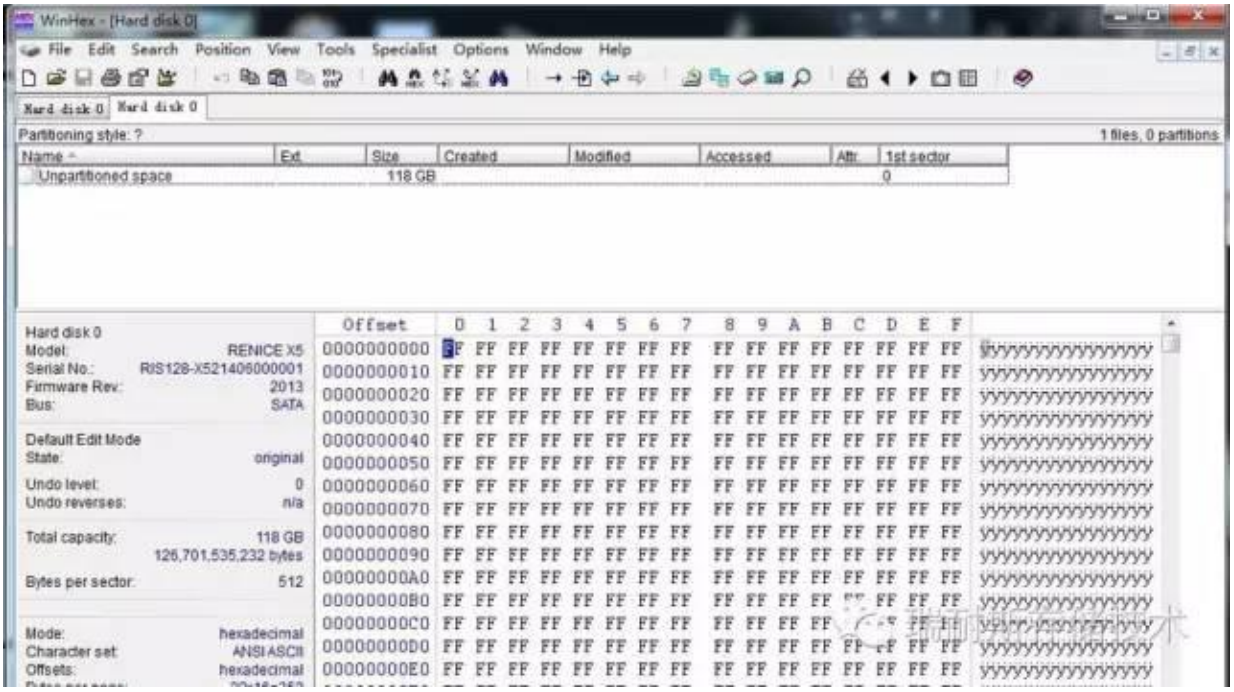
2.1.1 After the quick erase, the disk remains visible in OS and reusable after initialization, just all data are read as 0xFF by Winhex.
One problem we have to solve for this delete method: if during the delete operation, the SSD received the program command again, or the data in SDRAM have not been written into NAND flash, the program will be continued after quick erase. Then not all data are read as 0xFF with Winhex.
Renice provides two method to solve this problem: ① before SE implementation, purge the data in SDRAM firstly to avoid the data be written into NAND Flash after SE; ② enable the Write protection function when executing data destruction until power on next time. It prevents the write command received after SE executing and continued to be written into SSD.

2.12 After quick erase, the disk is invisible in OS and of course the status of the data inside cannot be checked by software.

The data before logical destroy:



The data after logical destroy:



2.2 Physical destroy

After the Physical destroy, all the NAND Flash on SSD have been breakdown completely.

2.2.1 Renice developed the solution to ensure all the NAND Flash been burned during 45s.

2.2.2 The physical destroy is to breakdown all the Die of each NAND Flash but not only the IO interface to ensure the data cannot be recovered.

2.23. The physical destroy will not be stopped until it finished after triggered. Even there is power-off during physical destroy executing, it will be continued once power-on.