Design Gateway Co.,Ltd

54 BB Building 14th Fl., Room No.1402 Sukhumvit 21 Rd. (Asoke), Klongtoey-Nua, Wattana, Bangkok 10110
Phone: 66(0)2-261-2277
Fax: 66(0)2-261-2290
E-mail: ip-sales@design-gateway.com
URL: www.design-gateway.com

Core Facts

Provided with Core	
Documentation	Reference Design Manual Demo Instruction Manual
Design File Formats	Encrypted HDL
Instantiation Templates	VHDL
Reference Designs & Application Notes	QuartusII Project, See Reference Design Manual
Additional Items	Demo on Arria10 GX Development Kit
Support	
Support Provided by Design Gateway Co., Ltd.	

Features

- Access NVMe SSD as exFAT system without using CPU and external memory
- Simple user interface and operating with DG NVMe IP
- Achieve best write/read speed (up to 2144 MB/s for write and up to 3251 MB/s for read)
- Support disk capacity: 8 Gigabyte* – 64 Petabyte*
  *Gigabyte means 1024x1024x1024 byte while Petabyte means 1024x1024x1024x1024x1024 byte
- Support only 512 byte LBA unit
- Four user commands, i.e. Format, Write file, and Read file, and Shutdown
- Support eight file sizes, i.e. 32MB, 128MB, 512MB, 2GB, 8GB, 32GB, 128GB, and 512GB (some file sizes are not available, depending on disk capacity)
- Reference design available on Arria10 GX development board with AB16-PCIeXOVR adapter board

Table 1: Example Implementation Statistics

Family	Example Device	Fmax (MHz)	Logic utilization (ALMs)	Registers[1]	Pin	Block Memory bit	Design Tools
Arria10 GX	10AX115S2F45I1SG	300	1,827	3,346	-	131,072	QuartusII 18.0

Notes:
1) Actual logic resource dependent on percentage of unrelated logic

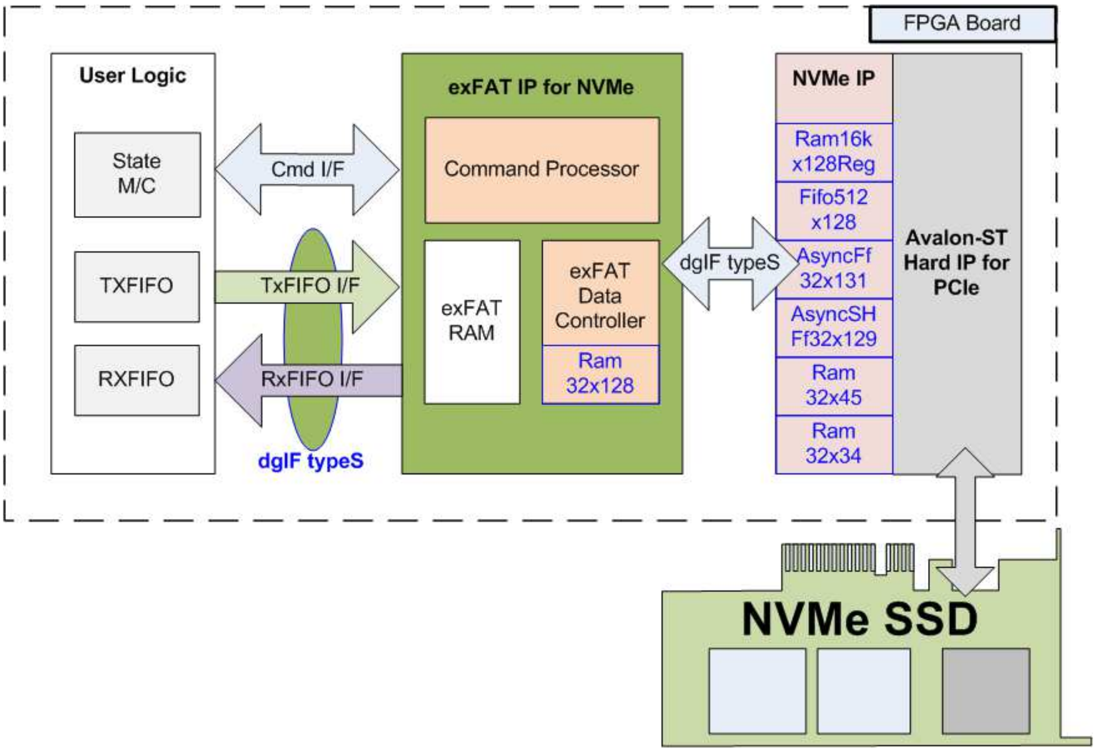

Figure 1: exFAT IP for NVMe Block Diagram

Applications

exFAT IP for NVMe must operate with DG NVMe IP and Avalon-ST Hard IP for PCIe (Hard IP in Intel FPGA device). The IP Core is an ideal to access NVMe SSD as exFAT file system with high speed performance, like raw data access. This solution fits the application which needs to record data to NVMe SSD by FPGA, but the data file is read by other system such as PC through file system format.

General Description

In general system, CPU and firmware are required to write/read data with the device as file system. So, system must include CPU and memory to store CPU firmware. Write/Read performance when access the device as file system by using CPU firmware is not good. There is overhead time which reduces the performance. Finally, most users decide to make their own file system, not using standard file system.

To solve above problem, exFAT IP for NVMe is designed to meet two requirements, i.e. using less FPGA resource (no CPU and no external memory) and writing/reading data at the highest speed like raw data access. exFAT IP for NVMe is top-up module of NVMe IP to allow user write/read data with NVMe SSD as exFAT file system instead of raw data (raw data system could be designed by using only NVMe IP). The interface of exFAT IP for NVMe is simple and almost similar to NVMe IP. Two groups of user interace are designed, i.e. control and data.

Control interface is almost similar to NVMe IP, but file parameters are used instead of physical parameters. File name is used to specify starting point to access data, instead of physical address. Numbers of file is used to specify transfer size, instead of physical length. Four commands are designed in exFAT IP to access NVMe SSD, i.e. Format, Write file, Read file, and Shutdown.

Data interface of exFAT IP is designed by using general FIFO interface, same as NVMe IP. Clock domain of exFAT IP must be same as user clock of NVMe IP because there is no asynchronous circuit to interface between exFAT IP and NVMe IP.

Otherwise, user needs to input file parameters to exFAT IP as input signals such as created date, created time, and file size. Created date and created time must be valid during sending command request to exFAT IP. File size must hold the same value after Format operation. There are eight file sizes to set, i.e. 32 MB, 128 MB, 512 MB, 2 GB, 8 GB, 32 GB, 128 GB, and 512 GB. More details of supported file size are shown in Table 2.

Table 2: Supported File size following device capacity

Device capacity	32MB(2)	128MB(2)	512MB(2)	2GB(2)	8GB(2)	32GB(2)	128GB(2)	512GB(2)
8 GB - 16 GB(1)	Yes	Yes	Yes	Yes	No	No	No	No
16 GB – 64 GB(1)	Yes	Yes	Yes	Yes	Yes	No	No	No
64 GB – 256 GB(1)	Yes	Yes	Yes	Yes	Yes	Yes	No	No
256 GB – 1 TB(1)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
1 TB – 512 TB(1)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
512 TB – 8 PB(1)	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
8 PB – 64 PB(1)	No	No	Yes	Yes	Yes	Yes	Yes	Yes

Note

(1) Upper limit shows that device capacity must be less than this value. For example, if device capacity is 512 TB, 32 MB size will not support (as described in row 512 TB – 8 PB).

(2) In the table,
MB = 1024x1024 byte.
GB = 1024x1024x1024 byte.
TB = 1024x1024x1024x1024 byte.
PB = 1024x1024x1024x1024x1024 byte.

After Format command, 128 empty directories are created in the device. Directory name is DIR00, DIR01, …, DIR0F, DIR10, … , DIR7E, and DIR7F. Two hexadecimal digits are used to refer directory number.

In Write file command operation, when FileName is equal to 0, exFAT IP creates the 1st file (0000000.BIN) to DIR00. The next new file (0000001.BIN, 0000002.BIN, …) is created to DIR00 until total files in DIR00 is equal to maximum file per directory (read from DirCap signal as shown in Table 3). After that, the next new file is stored to the next directory (DIR01). Maximum file per directory depends on device capacity.

Table 3: Maximum file per directory

Device capacity	DirCap[21:0]	Used directory
8 GB - 32 GB(1)	0x000400 (1024)	DIR00
32 GB – 128 GB(1)	0x000800 (2048)	DIR00 - DIR01
128 GB – 512	0x001000 (4096)	DIR00 – DIR03
512 GB – 2 TB(1)	0x002000 (8192)	DIR00 – DIR07
2 TB – 8 TB(1)	0x004000 (16384)	DIR00 – DIR0F
8 TB – 32 TB(1)	0x008000 (32768)	DIR00 – DIR1F
32 TB – 128 TB(1)	0x010000 (65536)	DIR00 – DIR3F
128 TB – 512 TB(1)	0x020000 (131072)	DIR00 – DIR7F
512 TB – 2 PB(1)	0x040000 (262144)	DIR00 – DIR7F
2 PB – 8 PB(1)	0x080000 (524288)	DIR00 – DIR7F
8 PB – 32 PB(1)	0x100000 (1048576)	DIR00 – DIR3F
32 PB – 64 PB(1)	0x200000 (2097152)	DIR00 – DIR3F

Note (1) The upper limit in the lower row is equal to the lower limit in the higher row. If device capacity is equal to the limit, please use the higher row. For example, if device capacity is 32 GB, please use the value in the 2nd row (32GB – 128 GB). The maximum file per directory of 32 GB device is equal to 2048.

Functional Description

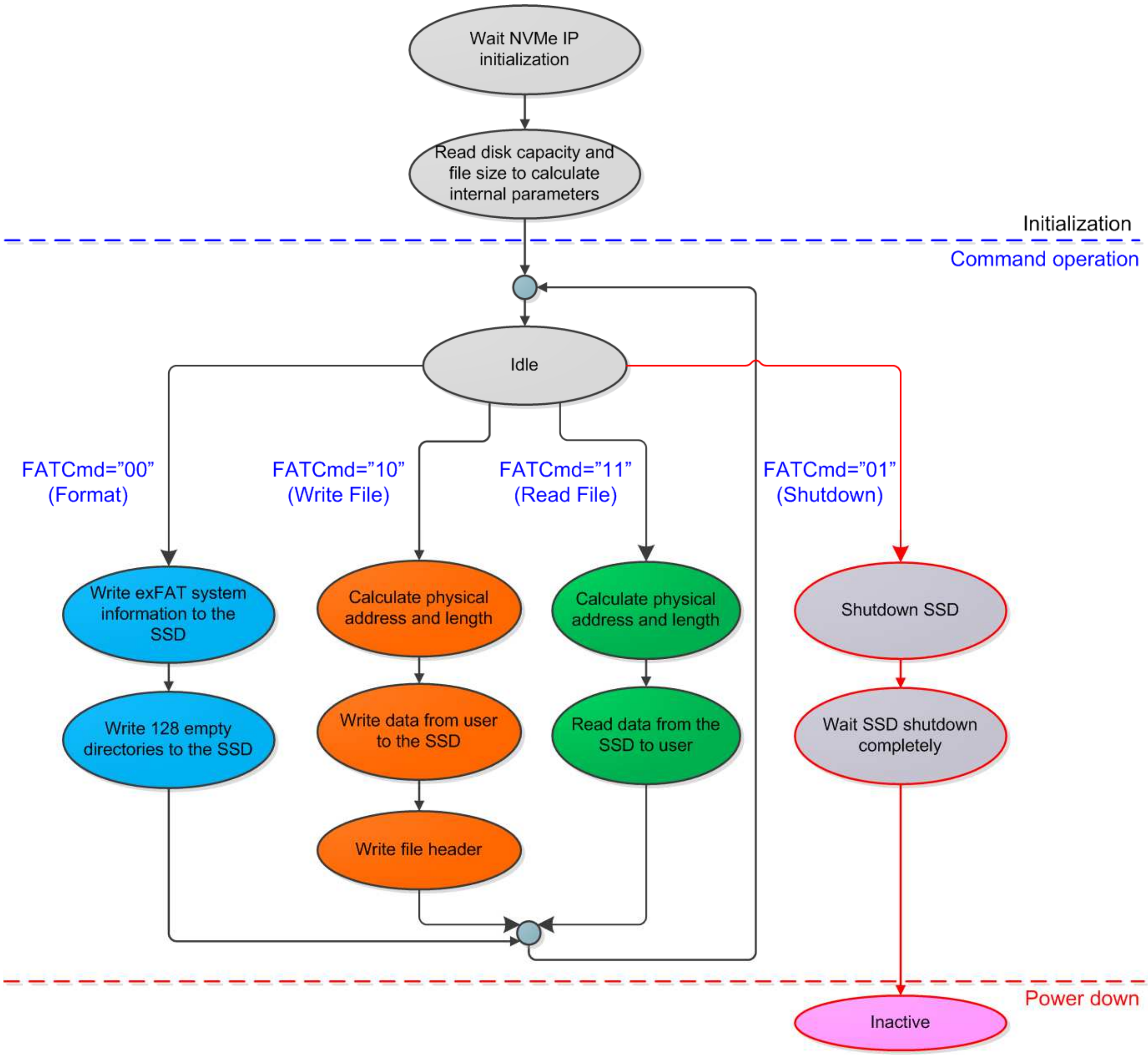

Figure 2: exFAT IP for NVMe Operation

The sequence of exFAT IP Core for NVMe operation is as follows:

1) exFAT IP waits NVMe IP to complete NVMe SSD initialization.

2) exFAT IP sends Identify command to NVMe IP to get disk capacity. File size (input from user) and disk capacity (input from NVMe IP) are applied to calculate system parameters inside exFAT IP. Finally, exFAT IP is ready to receive new command from user. Four commands are supported, i.e. Format, Write file, Read file, and Shutdown. Format command is the first command which must be used when connecting new NVMe SSD to the system. Shutdown is the last command which must be used before power down system.

3) a. For the new disk, Format command must be run to clean up the disk, set up file system, and create 128 empty directories. Otherwise, Format command must be applied when file size from user is changed which is effect to other system parameters. All files in one disk must have the same size because of the same FSize input.

   b. When user selects Write file command, the start file name (UserFName) and numbers of file (UserFLen) are inputs received from user. File name and numbers of file are converted to be physical address and total length to be input parameters for NVMe IP. After that, exFAT IP sends Write commands to NVMe IP. Next, data from user is forwarded to NVMe IP until complete to transfer total data. Finally, exFAT IP writes file header to link user data to file name.

   Note:
   - File header is filled to the disk as the final step. If system is powered down during writing file command, the file will be lost or corrupt.
   - There is no protection in the logic when some user inputs are out-of-range. User logic needs to send only valid input to exFAT IP.

   c. Similar to Write file command, the 1st process of Read file command is to calculate start physical address and total length to send to NVMe IP. Next, exFAT IP sends Read command to NVMe IP. After that, data is returned from NVMe IP to user until complete to transfer total data.

4) The last command which user must be sent before shutdown system is Shutdown command. This command is used to shutdown SSD in good sequence. Without Shutdown command, Write data in SSD cannot be guaranteed (some data may be stored in cache within SSD). After running Shutdown command, NVMe IP and SSD run in Inactive state.

exFAT IP for NVMe

As shown in Figure 1, there are three submodules inside the IP, i.e. Command processor, Data Controller, and exFAT RAM. To interface between user logic and NVMe IP, Command processor controls command interface while Data controller controls data interface. exFAT RAM stores internal file parameters transferring between Command processor and Data controller. More details of each submodule are described as follows.

- **Command processor**

  This module includes state machine to control the sequence to operate user command, following in Figure 2. Many calculation units are applied to convert file parameters from user to be physical parameters for NVMe IP. Most file parameters depend on file size and disk capacity.

- **exFAT RAM**

  This RAM is implemented by Block memory (M9K/M10K/M20K depending on FPGA model) to store file system data. RAM size is 1024x128-bit.

- **Data Controller**

  There are three data types to transferr with NVMe IP, i.e. raw data, file system data, and file parameters. The source and destination of raw data is FIFO interface of user logic. Most file system data are stored in exFAT RAM. File parameters are calculated and latched as internal register.

- **Ram32x128**

  This module is created by IP catalog inside Quartus tools. The buffer is two-port RAMs (one write port and one read port). RAM depth is 32 words and data size is 128 bit. “Output Registers” are not included, but “Input Registers” are included for “All write input ports” and “rdaddress port”. So, read data latency is equal to one clock cycle. To optimize resource, RAM Block Type is selected to be MLAB.

DG NVMe IP

exFATIP for NVMe is the top-up module of DG NVMe IP. So, the system needs to include DG NVMe IP. More details of NVMe IP are described in datasheet which can be downloaded from following link.

https://dgway.com/products/IP/NVMe-IP/Altera/dg_nvme_ip_datasheet_altera5_en.pdf

NVMe IP includes six buffers which are created by IP catalog inside Quartus tools. The setting of each buffer is described as follows.

- Ram16kx128Reg: Two-port RAM (one write port and one read port) which has 16k word depth. Data bus size is 128 bit. RAM is implemented by Block memory with byte enable option. “Input Registers” and “Output Registers” are included for all ports.
- Fifo512x128: Synchronous FIFO with asynchronous reset type. FIFO depth is 512 words and data width is 128 bit. FIFO is implemented by Block memory. Data counter, full, and empty signals are applied.
- AsyncFf32x131: Asynchronous FIFO (write port and read port run in different clock domain). FIFO depth is 32 words and data width is 131 bit. FIFO is implemented by MLAB. Data counter, full, and empty signals are applied.
- AsyncSHFf32x129: Asynchronous FIFO (write port and read port run in different clock domain). FIFO depth is 32 words and data width is 129 bit. Signal interface is same as AsyncFf32x131, but Rdreq option is show-ahead mode (not normal mode).
- Ram32x45 and Ram32x34: Two-port RAM (one write port and one read port). RAM depth is 32 words and data width is 45/34 bit. RAM is implemented by MLAB. “Input Registers” are included for “All write input ports” and “rdaddress port”. “Output Regsiters” are not included.

User Logic

This module could be designed by using small state machine to send command and input parameters to exFAT IP. Two FIFOs are used to be data buffer to write and read file with exFAT IP. exFAT IP reference design is designed to receive input parameters from user through JTAG UART. So, CPU with simple firmware is designed instead of using small state machine to set input parameters.

Core I/O Signals

Descriptions of all signal I/O are provided in Table 4.

Table 4: Core I/O Signals

Signal	Dir	Description
System signal		
RstB	In	Synchronous reset signal. Active low. Deassert to '1' after Clk signal is stable.
Clk	In	User clock. Must use the same clock as Clk input of NVMe IP which is more than or equal to PCIeClk. PCIeClk is clock output from Avalon-ST PCIe Hard IP. PCIeClk is equal to 125 MHz for PCIe Gen2 and 250 MHz for PCIe Gen3.
FSize[2:0]	In	File size: "000": 32MB, "001": 128MB, "010": 512MB, "011": 2GB, "100": 8GB, "101": 32GB, "110": 128GB, "111": 512GB Note: 1) 1 MB is 1024x1024 byte, 1 GB is 1024x1024x1024 byte 2) FSize must be fixed value after Format command. **Format command is required when FSize value is changed.**
User Interface (Command)		
UserCmd[1:0]	In	User Command. "00": Format command, "01" Shut down, "10": Write file, "11": Read file.
UserFName[26:0]	In	Start file name to write/read file (0="0000000.BIN", 1="0000001.BIN", …). Valid from 0 to TotalFCap - 1. This input is not used in Format command. In Write file command, this value must be equal 0 for empty disk or equal to (the last file name + 1) for non-empty disk (continue to write the next file from previous write).
UserFLen[26:0]	In	Total files transferring in the request. Valid from 1 to (TotalFCap – UserFName). This input is not used in Format command.
UserReq	In	Assert to '1' to send new command request with the valid value on UserCmd, UserFName, UserFLen, FDateYMD, and FTimeHMS. This signal could be asserted to '1' when the IP is Idle only (UserBusy='0').
UserBusy	Out	'0': IP is idle, '1': IP is busy.
TotalFCap[26:0]	Out	Maximum files to store in NVMe SSD. This value is valid after exFAT IP completes initialization process (UserBusy='0').
DirCap[21:0]	Out	Maximum files to store in one directory. This value is valid after exFAT IP completes initialization process (UserBusy='0').
FDateY[6:0]	In	Year in created date, count from 1980 (For example, FDateY=39 is year 2019). This input is used to be created date of directory in Format command and created date of file in Write file command.
FDateM[3:0]	In	Month in created date. Valid from 1-12 (1=Jan, 2= Feb, …). This input is used to be created date of directory in Format command and created date of file in Write file command.
FDateD[4:0]	In	Day in created date. Valid from 1-31. This input is used to be created date of directory in Format command and created date of file in Write file command.
FTimeH[4:0]	In	Hour in created time. Valid from 0-23. This input is used to be created time of directory in Format command and created time of file in Write file command.
FTimeM[5:0]	In	Minute in created time. Valid from 0-59. This input is used to be created time of directory in Format command and created time of file in Write file command.

Signal	Dir	Description
User Interface (Command)		
FTimeS[4:0]	In	x2 sec in created time. Valid from 0-29 (1=2, 2=4, …). This input is used to be created time of directory in Format command and created time of file in Write file command.
UserError	Out	Error flag. Assert to ‘1’ when UserErrorType is not equal to 0. The flag is cleared to ‘0’ by asserting RstB signal to NVMe IP and exFAT IP.
UserErrorType[31:0]	Out	[23:0] - Direct mapped to NVMeErrorType signal of NVMe IP. Please see more details in NVMe IP datasheet. [24] - Error when disk capacity is more than or equal to 64 PetaByte. [25] – Error when LBA unit of NVMe SSD is not equal to 512 byte.
IPVesion[31:0]	Out	IP version number
TesPin[63:0]	Out	Reserved to be IP test point.
User Interface (Data)		
UserFifoWrCnt[15:0]	In	Write data counter of Rx FIFO. Used to check full status. If total FIFO size is less than 16-bit, please fill ‘1’ to upper bit. UserFifoWrEn could be asserted to ‘1’ when UserFifoWrCnt[15:5] is not equal to all 1.
UserFifoWrEn	Out	Asserted to ‘1’ to write data to Rx FIFO when receive data from NVMe IP in read file command.
UserFifoWrData[127:0]	Out	Write data bus of Rx FIFO. Valid when UserFifoWrEn=’1’.
UserFifoRdCnt[15:0]	In	Read data counter of Tx FIFO. Used to check total numbers of data stored in FIFO. If FIFO size signal is less than 16-bit, please fill ‘0’ to upper bit.
UserFifoEmpty	In	This signal is unused for this IP.
UserFifoRdEn	Out	Asserted to ‘1’ to read data from Tx FIFO to NVMe IP in write file command.
UserFifoRdData[127:0]	In	Read data returned from Tx FIFO. Valid in the next clock after UserFifoRdEn is asserted to ‘1’.
NVMe IP User Interface (Connect to dgIF typeS of NVMe IP) (Please see more details from NVMe IP datasheet)		
NVMeCmd[2:0]	Out	Connect to UserCmd of NVMe IP
NVMeAddr[47:0]	Out	Connect to UserAddr of NVMe IP
NVMeLen[47:0]	Out	Connect to UserLen of NVMe IP
NVMeReq	Out	Connect to UserReq of NVMe IP
NVMeBusy	In	Connect to UserBusy of NVMe IP
NVMeLBASize[47:0]	In	Connect to LBASize of NVMe IP
NVMeLBAMode	In	Connect to LBAMode of NVMe IP core
NVMeCtmSubmDW0[31:0] - NVMeCtmSubmDW15[31:0]	out	Connect to CtmSubmDW0 – CtmSubmDW15 of NVMe IP core
NVMeError	In	Connect to UserError of NVMe IP
NVMeErrorType[31:0]	In	Connect to UserErrorType of NVMe IP
NVMeFifoWrCnt[15:0]	Out	Connect to UserFifoWrCnt of NVMe IP
NVMeFifoWrEn	In	Connect to UserFifoWrEn of NVMe IP
NVMeFifoWrData[31:0]	In	Connect to UserFifoWrData of NVMe IP
NVMeFifoRdCnt[15:0]	Out	Connect to UserFifoRdCnt of NVMe IP
NVMeFifoEmpty	Out	Connect to UserFifoEmpty of NVMe IP
NVMeFifoRdEn	In	Connect to UserFifoRdEn of NVMe IP
NVMeFifoRdData[31:0]	Out	Connect to UserFifoRdData of NVMe IP

Timing Diagram

Initialization

The sequence of the initialization process is as follows.

1) RstB is released to ‘1’ by user after Clk is stable.
2) exFAT IP starts initialization process by sending Identify command to NVMe IP.
3) UserBusy is deasserted to ‘0’ after exFAT IP receives Identify data from NVMe IP completely. Next, TotalFCap to show maximum files to store in the disk is valid to read.

After complete above sequence, exFAT IP is ready to receive the command from user.

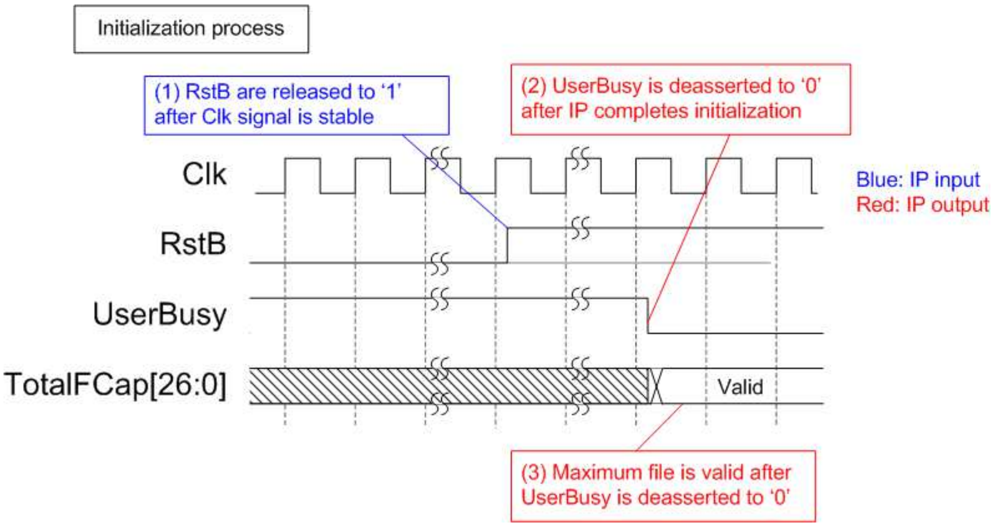

Figure 3: exFAT IP for NVMe Intialization

User Interface

User interface is split into two interfaces, i.e. command interface and data interface. Figure 4 shows timing diagram of command interface as file parameters instead of physical parameters (user interface of NVMe IP). Before sending new command to exFAT IP, UserBusy must be monitored to confirm that IP is Idle. Command parameters (UserCmd, UserFName, UserFLen, FDateYMD, and FTimeHMS) must be valid and stable during asserting UserReq='1'. After new request from user is received, UserBusy changes status from '0' to '1'. In the next clock, UserReq should be de-asserted to '0'. User logic can prepare the parameters of the next command on the bus when UserReq changes to '0'.

Note: 1) UserFName and UserFLen input are ignored in Format command.
2) FDateYMD and FTimeHMS are ignored in Read file command.

For data interface, TxFIFO (Transmit FIFO) is read in Write file command and RxFIFO (Received FIFO) is written in Read file command. Timing diagram of data interface is same as NVMe IP timing diagram, as shown in Figure 5 and Figure 6.

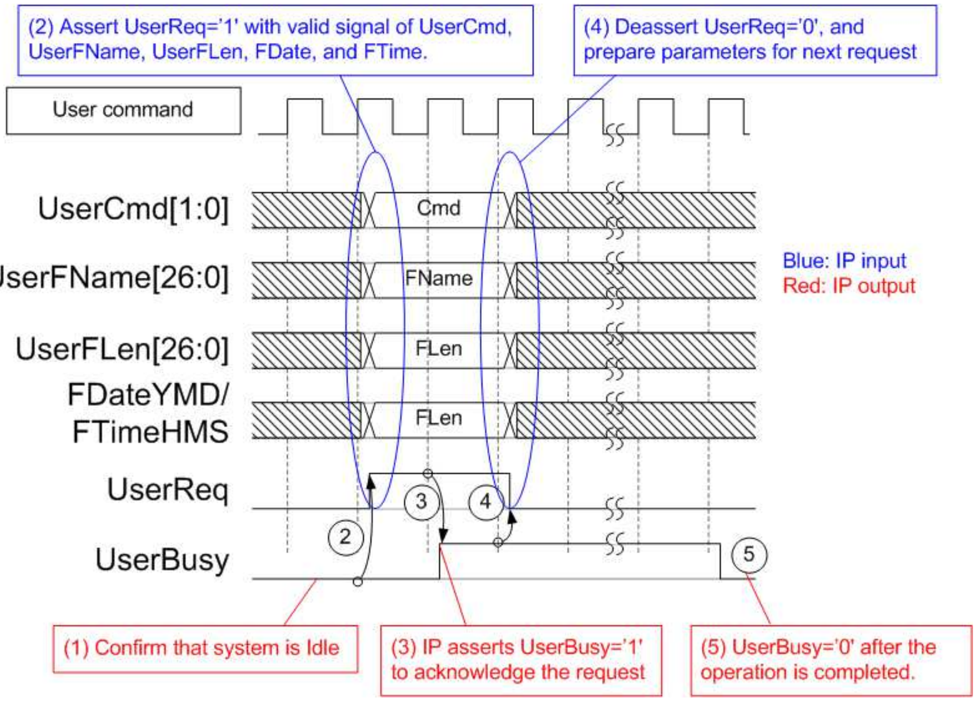

Figure 4: Command User Interface Timing diagram

For Write file command, UserFifoRdCnt is monitored by NVMe IP to check data available in Tx FIFO. When data is much enough, UserFifoRdEn is asserted to '1'. UserFifoRdData is valid in the next clock after UserFifoRdEn='1'. UserFifoRdData is forwarded from Tx FIFO to NVMe IP. Data is transferred as burst, so UserFifoRdEn is asserted to '1' for 32 clock cycles continuously (512 byte). After total data are transferred, UserFifoRdEn is de-asserted to '0'.

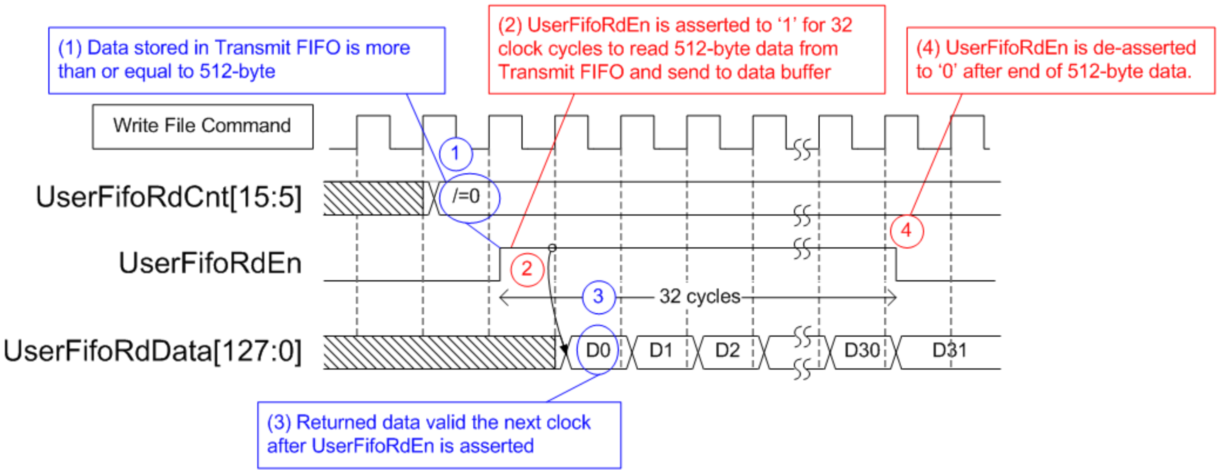

Figure 5: TX FIFO Interface for Write file command

For Read file command, UserFifoWrEn is asserted to '1' with the valid value of UserFifoWrData to send data to Rx FIFO. Before asserting UserFifoWrEn to '1', UserFifoWrCnt is monitored to check free space size of Rx FIFO. Data is transferred when free space in Rx FIFO is more than or equal to 1024 byte. Data is transferred as burst, so UserFifoWrEn is asserted to '1' for 32 clock cycles continuously (512 byte).

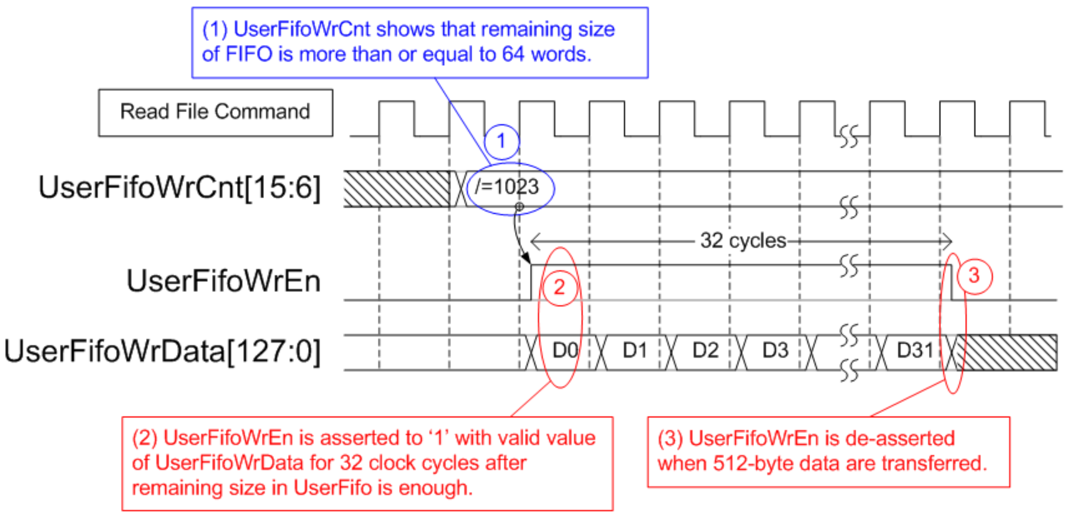

Figure 6: RX FIFO Interface for Read file command

Error

UserError is asserted to ‘1’ by error condition within NVMe IP or exFAT IP. More details of error status are defined in UserErrorType. To clear Error flag, user needs to send reset signal (RstB=’0’) to exFAT IP and NVMe IP.

After complete above sequence, exFAT IP is ready to receive the command from user.

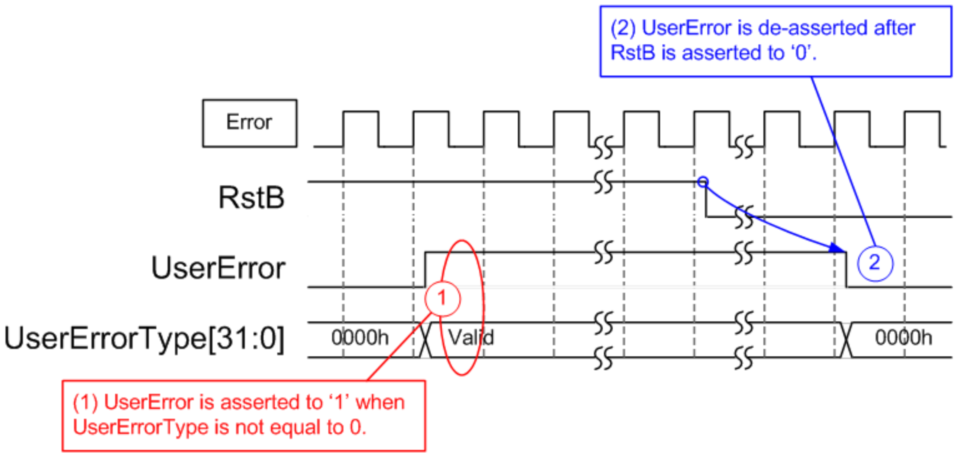

Figure 7: Error flag timing diagram

Example usage

The example sequence to use exFAT IP for NVMe to write or read file with NVMe SSD is shown as follows.

1) De-assert RstB to '1' to start system initialization. Wait until UserBusy='0'.

2) Skip to the next step when the disk does not need to format. Format command must be run when some following conditions are found.
   a) The disk is new and not formatted by exFAT IP for NVMe.
   b) User needs to delete file in the disk.
   c) User needs to change FSize (file size) parameter.

   Following is the step to operate Format command.
   a) Set value to FDateYMD and FTimeHMS to specify created date and created time of 128 empty directories (DIR00 – DIR7F) which are created by Format command operation.
   b) Assert UserReq to '1' and set UserCmd="00". Wait until UserBusy changes from '0' to '1' as acknowledge for Format command. Then, de-assert UserReq to '0' in the next clock.
   c) Wait until exFAT IP operation is completed by monitoring UserBusy='0'.

   As a result, DIR00 – DIR7F as empty directory are created in the disk.

3) Send Write file command (UserCmd="10") or Read file command (UserCmd="11") to exFAT IP for NVMe. The sequence of Write file and Read file command are as follows.

   In case of Write file command
   a) Set FDateYMD and FTimeHMS to specify created date and created time of new file.
   b) Set UserFName to be equal to the next value of the latest write file. For example, if the latest write file in the disk is 0000019.BIN, UserFName will be set to 0x000001A.
   c) Set UserFLen as valid value (less than or equal to <TotalFCap – UserFName>), and then assert UserReq='1'. Wait until UserBusy='1' and de-assert UserReq to '0' in the next clock.
   d) Send data to exFAT IP through Tx FIFO interface. Total data size is equal to UserFLen (total write files) x FSize (file size). UserFifoRdCnt[15:5] must be not equal to 0 when data is ready to read by exFAT IP.
   e) Wait until exFAT IP operation complete by monitoring UserBusy='0'.

   In case of Read file command
   a) Set UserFName and UserFLen to read the file from the disk. The sum of UserFname and UserFLen must not be more than TotalFCap.
   b) Assert UserReq='1' to send Read file command to exFAT IP. Wait until UserBusy='1' and de-assert UserReq to '0' in the next clock.
   c) Read total data from RxFIFO. When UserFifoWrCnt[15:6] is not equal to all 1 (1023), write enable of RxFIFO could be asserted to '1' to write data by exFAT IP. Total data size is equal to UserFLen (total read files) x FSize (file size).
   d) Wait until exFAT IP operation complete by monitoring UserBusy='0'.

4) Send Shutdown command (UserCmd="01") to exFAT IP before shutdown system. The sequence of Shutdown command is as follows.
   a) Assert UserReq='1' and set UserCmd="01". Wait until UserBusy changes from '0' to '1' as acknowledge for Format command. Then, de-assert UserReq to '0' in the next clock.
   b) Wait until exFAT IP operation is completed by monitoring UserBusy='0'.

   As a result, NVMe SSD changes to power down mode.

Limitation

(1) exFAT IP writes or reads file from the disk when the disk is formatted and the file is written by exFAT IP only. If not, the disk must be formatted by exFAT IP before connecting to exFAT IP. Other host system allows to access the disk as read-only mode.

(2) File size (FSize) input to exFAT IP must be fixed after Format command. All files in one disk written by exFAT IP must have the same size (set by FSize). To change FSize value, the disk must be formatted. Also, maximum files stored in one disk (TotalFCap) depends on FSize value.

(3) As shown in Table 3, one directory has limitation about number of file to store, depending on device capacity. The equation to check directory name from filename is as follows.
DIR Name = (<FileName>/DirCap) – 1
Assumed that device capacity is 200 GB, DirCap (the maximum file in one directory) is equal to 4096.
0000000.BIN – 00000FFF.BIN are stored to DIR00.
0001000.BIN – 00001FFF.BIN are stored to DIR01.
0002000.BIN – 00002FFF.BIN are stored to DIR02.
0003000.BIN – 00003FFF.BIN are stored to DIR03.
DIR04 – DIR7F are empty directories.

(4) In Write file command, UserFName value must be the next value from the latest write file name. The example of correct parameter is shown in Figure 8.
Assumed that, the 1st write file operation sends command to write 0000000.BIN and 0000001.BIN. The next write file to continue filename is 0000002.BIN. So, the next write file command must set UserFName=2.

The example to set the wrong value of UserFName is shown in Figure 9 - Figure 10.

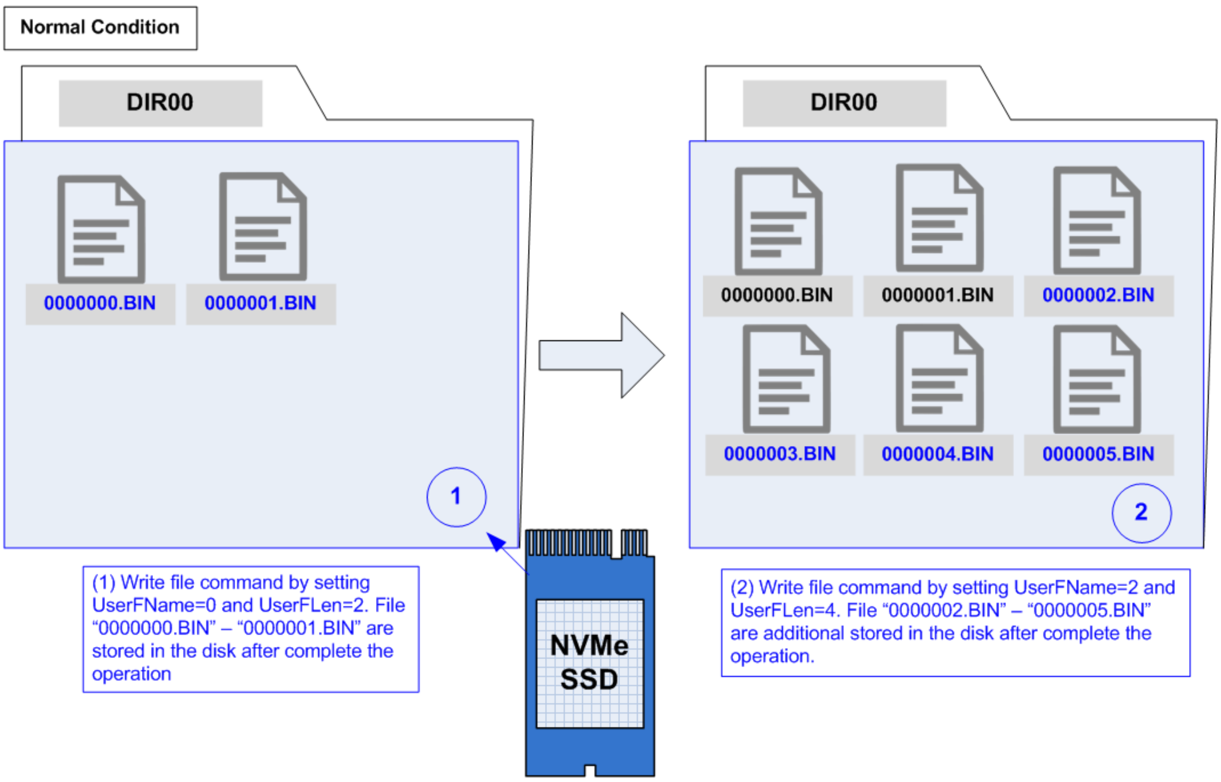

Figure 8: Write file command in normal condition

a) When UserFName is more than (the latest write file name + 1), there are three results which could be possible, i.e.
(i) Dummy file is created between the latest write file and the new file, as shown in (2a) of Figure 9. File 0000002.BIN (dummy file which consists of random data) is generated when UserFName input is equal to 3 instead of 2 which is correct value.
(ii) No new file is created, as shown in (2b) of Figure 9. If UserFName input is much skipped than correct value, no new created file will be created. For example, UserFName is equal to 8 instead of 2 which is correct value.
(iii) No dummy file is created and new file is created correctly. If UserFName is not correct but meet some special conditions, the file in directory will be found as normal operation.

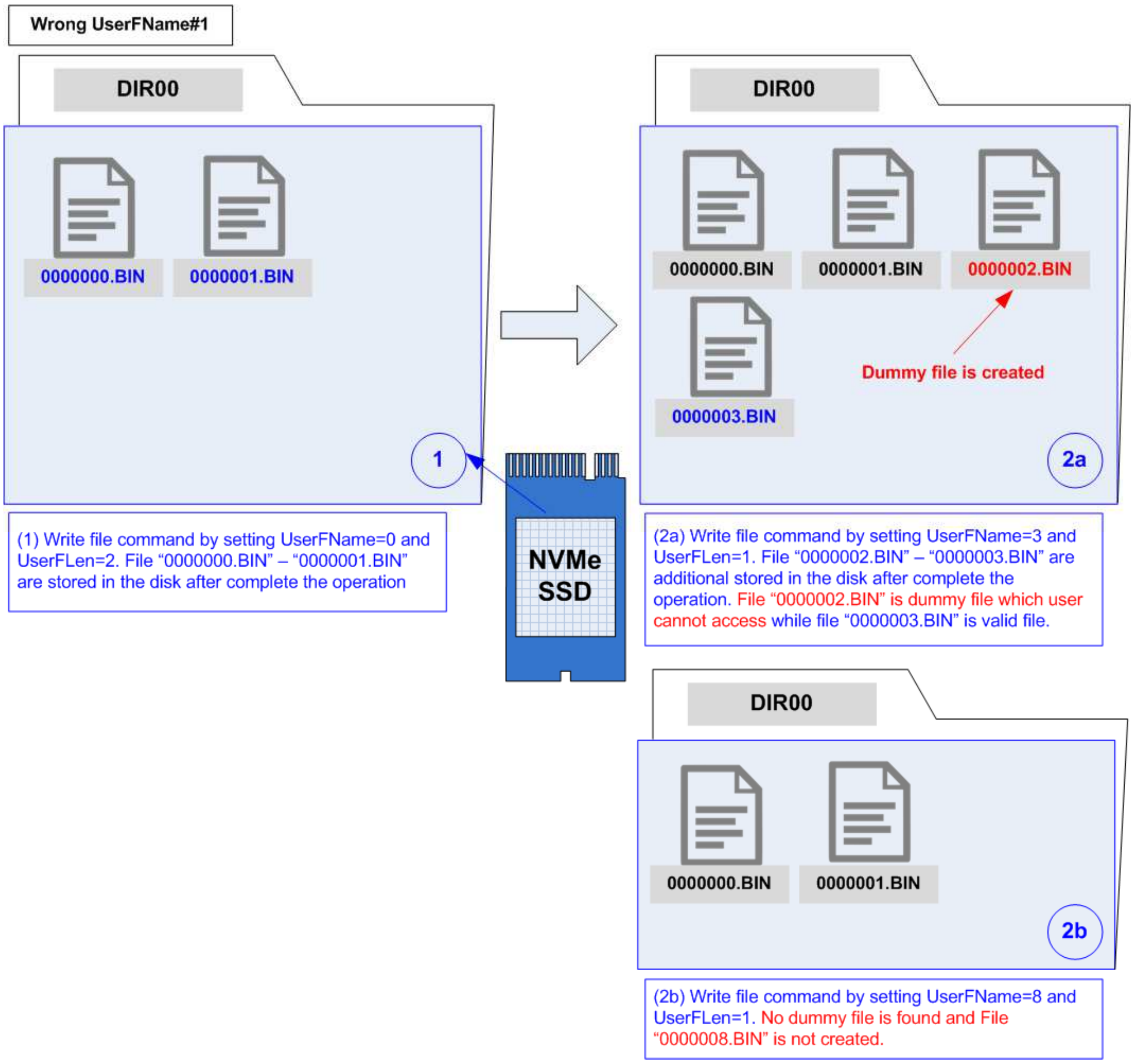

Figure 9: Write file command but UserFName is more than normal value

b) When UserFName is less than (the latest write file name + 1), some old files which are more than UserFName may be deleted, as shown in Figure 10.
Assumed that, the 1st write file operation sends command to write 0000000.BIN – 0000003.BIN. In the next write file command, UserFName is equal to 2 and UserFLen is equal to 1. As a result, 0000002.BIN is created by the new data and 0000003.BIN is lost.

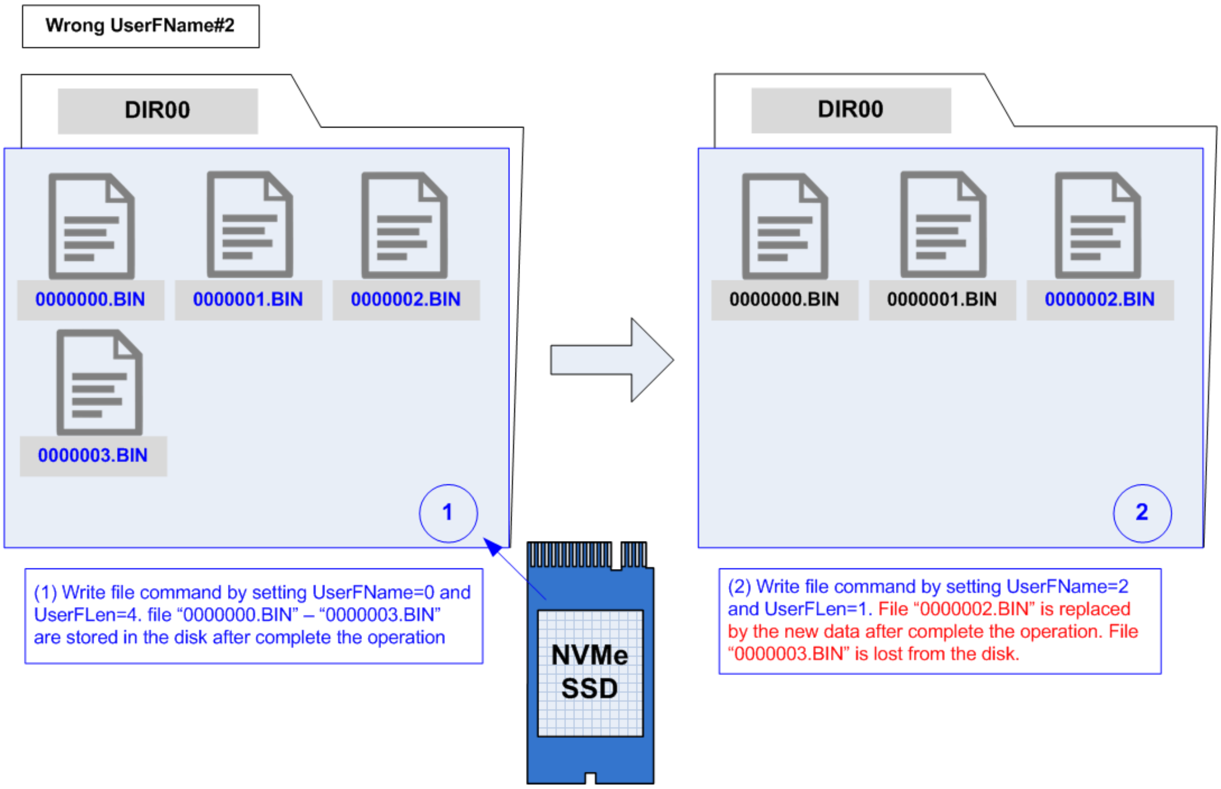

Figure 10: Write file command but UserFName is less than normal value

Verification Methods

The exFAT IP Core for NVMe functionality was verified by simulation and also proved on real board design by using Arria10 GX development board.

Recommended Design Experience

Experience design engineers with a knowledge of Quartus tools should easily integrate this IP into their design.

Ordering Information

This product is available directly from Design Gateway Co., Ltd. Please contact Design Gatway Co., Ltd. for pricing and additional information about this product using the contact information on the front page of this datasheet.

Revision History

Revision	Date	Description
1.0	Jan-22-2019	New release