

## FAT32 IP Core for NVMe

May 9, 2018

Product Specification

Rev1.1



### Design Gateway Co.,Ltd

54 BB Building 14<sup>th</sup> 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

### Features

- Simple user interface
- Operating with DG NVMe IP Core
- Implement FAT32 file system management without CPU usage
- Maximum write speed up to 2100 MB/s and read speed up to 3200 MB/s
- Support SSD capacity: 64 MB – 2 TB
- Support three commands, i.e. Format, Write file, and Read file
- Support seven file sizes, i.e. 32MB, 64MB, 128MB, 256MB, 512MB, 1024MB, 2048MB
- Reference design available on ZC706/KCU105 board with AB09-FMCRAID adapter board

### Core Facts

Provided with Core	
Documentation	Reference Design Manual Demo Instruction Manual
Design File Formats	Encrypted Netlist
Instantiation Templates	VHDL
Reference Designs & Application Notes	Vivado Project, See Reference Design Manual
Additional Items	Demo on ZC706, KCU105
Support	
Support Provided by Design Gateway Co., Ltd.	

Table 1: Example Implementation Statistics for Ultrascale device

Family	Example Device	Fmax (MHz)	CLB Regs	CLB LUTs	CLB	IOB	BRAMTile <sup>1</sup>	PLL	GTX	Design Tools
Kintex-Ultrascale	XCKU040FFVA1156-2E	344	1340	1549	306	-	2.5	-	-	Vivado2015.4

Table 2: Example Implementation Statistics for 7-Series device

Family	Example Device	Fmax (MHz)	Slice Regs	Slice LUTs	Slices	IOB	BRAMTile <sup>1</sup>	PLL	GTX	Design Tools
Zynq-7000	XC7Z045FFG900-2	300	1340	1439	512	-	2.5	-	-	Vivado2015.4

May 9, 2018

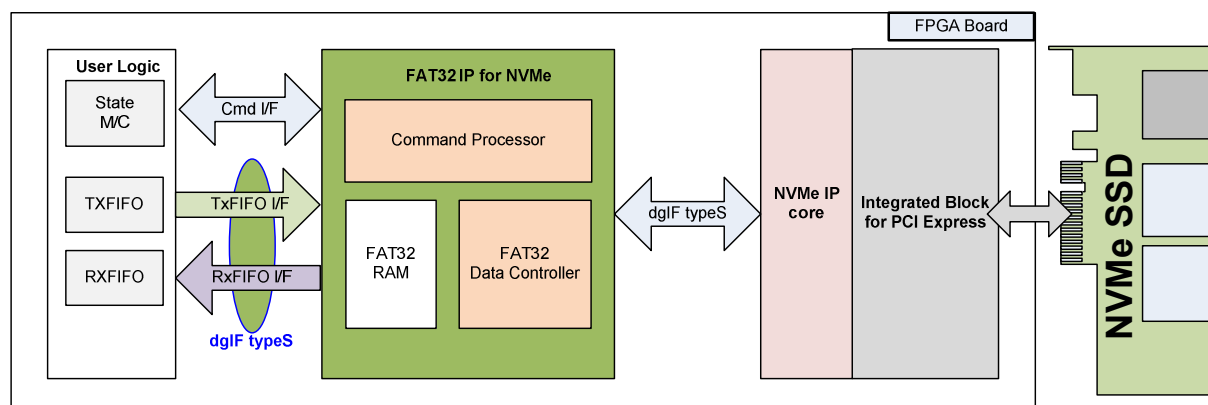


Figure 1: FAT32 IP for NVMe Block Diagram

## Applications

FAT32 IP Core for NVMe operates with DG NVMe IP core and Integrated Block for PCI Express from Xilinx. The IP core is an ideal to access NVMe SSD at ultra speed though data format is FAT32 File system; not raw data. This solution fits the application which needs to record data to NVMe SSD by using FPGA, but the data is read by other system such as PC through file system format.

## General Description

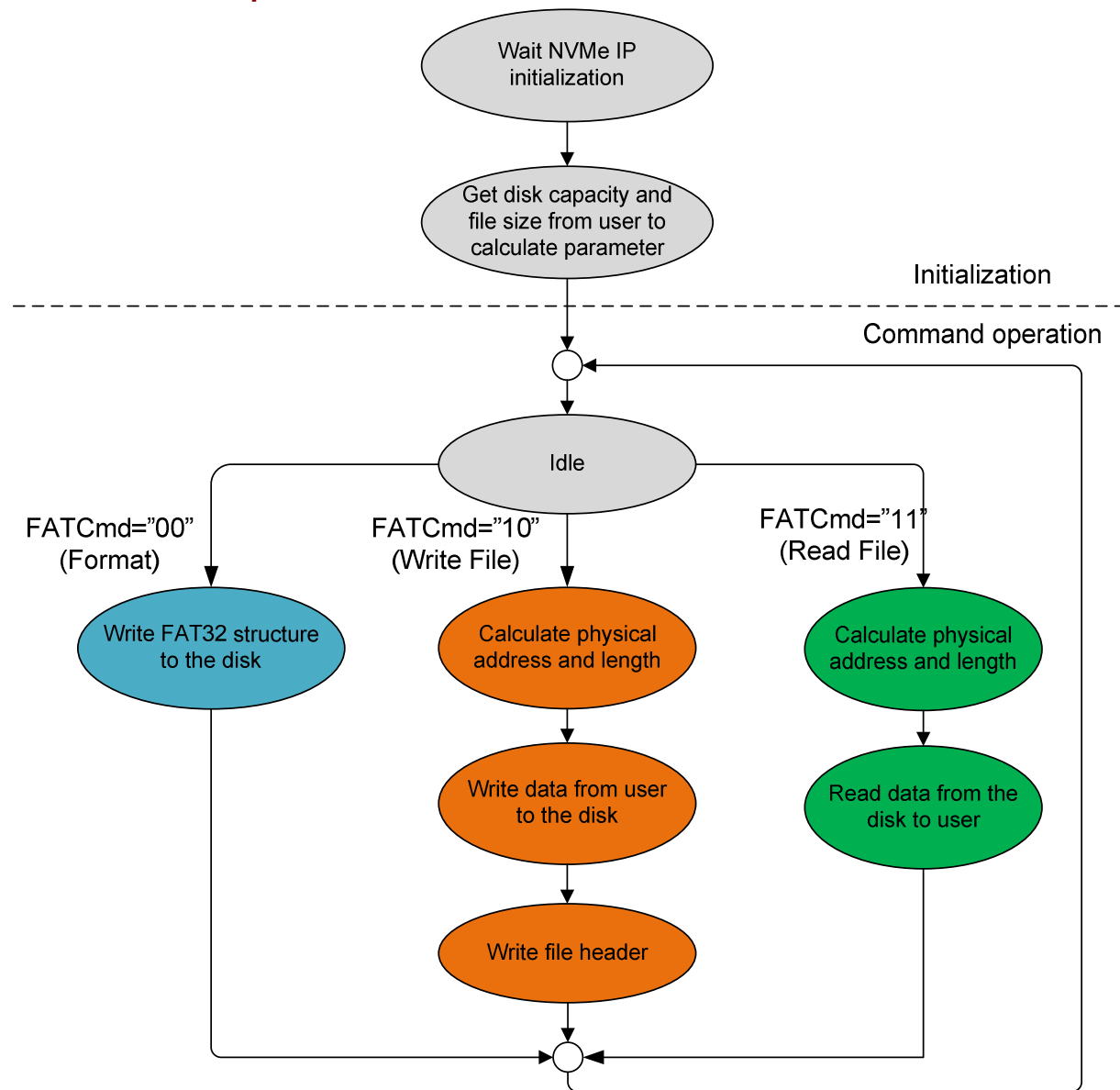
FAT32 IP Core for NVMe is designed to transfer data as FAT32 file system to NVMe SSD by interfacing between user logic and NVMe IP core. User transfers data to NVMe SSD as a file in FAT32 file system standard, instead of raw data. Data format in NVMe SSD after running NVMe IP reference design is raw data. So, it needs to use special software to read data from NVMe SSD as raw data when connecting SSD to other NVMe host. FAT32 file system is well known and generally used in many systems, so SSD could be read by standard tools on NVMe host.

FAT32 IP Core for NVMe supports three commands, i.e. Format, Write file, and Read file. FAT32 IP Core supports varieties of SSD capacity (64 MB – 2 TB) with only one partition. File size to store data is adjustable (32MB - 2GB per file). Once the SSD has stored some file, file size must not be changed. Format is required to clear old file before storing the new file size in SSD.

User interface of FAT32 IP Core for NVMe has two groups, i.e. control and data. Control interface is almost similar to dglF typeS, but physical address and length are replaced by file number and numbers of file. Data interface of FAT32 IP Core is same as dglF typeS interface.

Clock domain of FAT32 IP Core for NVMe must use same domain as NVMe IP's user clock.

**Functional Description**



**Figure 2: FAT32 IP for NVMe Operation**

The sequence of FAT32 IP Core for NVMe operation is as follows:

- 1) FAT32 IP waits NVMe IP to complete device initialization.
- 2) FAT32 IP sends Identify command to NVMe IP to get SSD capacity. File size input from user and SSD capacity are used by FAT32 IP to calculate system parameters and latch to internal register. After that, IP is ready to receive new command from user. Three commands are supported, i.e. Format, Write file, and Read file.
- 3) a. For the new SSD, Format command is required to set up file system and clean up the SSD. Also, when file size parameter is changed, Format command is required to recalculate new system parameters such as the maximum numbers of file in the SSD. In Format operation, FAT32 file structure such as Master boot record and root directory are written in the SSD by FAT32 IP.

b. When user selects Write file command, the 1<sup>st</sup> file name and numbers of file (inputs from user) are inputs to calculate SSD address and length inputs to NVMe IP. After that, total data from user are transferred to the SSD through NVMe IP. Finally, the IP writes file header to complete file operation.

Note:

1. File header is filled to the SSD in the final step. If system is powered down before the header is filled, all files will not store to SSD.
2. There is no protection if the input from user is out-of-range.

c. Similar to Write file command, the 1<sup>st</sup> process of Read file command is to calculate SSD address and length inputs to NVMe IP. After that, data from SSD is transferred to user until end of transfer.

### FAT32 IP for NVMe

As shown in Figure 1, there are three submodules inside the IP, i.e. Command processor, Data Controller, and FAT32 RAM. Command processor receives command from user logic, converts the inputs to internal parameters and sends the command to NVMe IP. Data controller is designed to transfer data between FIFOs in user logic to NVMe IP. Data sending to NVMe IP has two types, i.e. user data and file system data. Most of File system data are stored in FAT32 RAM. More details of each submodules are described as follows.

- **Command processor**

This module includes state machine to control the sequence of the logic for each user command, following in Figure 2. There are many calculation units in this block to calculate FAT32 parameters and SSD address from user command, user inputs, and SSD capacity. This block is connected to command interface of user logic and command interface of NVMe IP. Also, control signals for other internal logics are designed in this module.

- **FAT32 RAM**

This RAM is BlockRAM to store FAT32 file system data. The size is 256x128-bit.

- **FAT32 Data Controller**

This module is designed to control data flow between user logic and NVMe IP. To create file in the SSD, three data sources are transferred to NVMe IP, i.e. user data from TxFIFO in user logic, FAT32 file system data from FAT32 RAM, and FAT32 parameters from internal registers.

## **NVMe IP**

FAT32 IP for NVMe is designed to run with NVMe IP and Integrated Block for PCIe. The details of NVMe IP are described in the datasheet and can be downloaded from our website.

[http://www.dgway.com/products/IP/NVMe-IP/dg\\_nvme\\_ip\\_data\\_sheet\\_en.pdf](http://www.dgway.com/products/IP/NVMe-IP/dg_nvme_ip_data_sheet_en.pdf)

## **User Logic**

This module could be designed by using simple logic. For example, small state machine is applied to send command, file number, and file size. Two FIFOs are used to store data transferring with FAT32 IP for NVMe. One FIFO is used for Write file command and another FIFO is used for Read file command.

## Core I/O Signals

Descriptions of all signal I/O are provided in Table 3.

**Table 3: Core I/O Signals**

Signal	Dir	Description
<b>System signal</b>		
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. (At least 125 MHz for PCIe Gen2 or 250 MHz for PCIe Gen3).
FSize[2:0]	In	File size: "000" : 32MB, "001" : 64MB, "010" : 128MB, "011" : 256MB, "100" : 512MB, "101" : 1024MB, "110" : 2048MB, "111": Reserved Note: 1) 1 MB is 1024x1024 byte 2) FSize must not be changed when SSD has the file. Format command is required before changing FSize value.
FDateY[6:0]	In	Year in created date, count from 1980 (For example, FDateY=37 is year 2017). This input is used in Write file command. The value must not be changed during operation.
FDateM[3:0]	In	Month in created date. Valid from 1-12 (1=Jan, 2= Feb, ...). This input is used in Write file command. The value must not be changed during operation.
FDateD[4:0]	In	Day in created date. Valid from 1-31. This input is used in Write file command. The value must not be changed during operation.
FTimeH[4:0]	In	Hour in created time. Valid from 0-23. This input is used in Write file command. The value must not be changed during operation.
FTimeM[5:0]	In	Minute in created time. Valid from 0-59. This input is used in Write file command. The value must not be changed during operation.
FTImeS[4:0]	In	x2 sec in created time. Valid from 0-29 (1=2, 2=4, ...). This input is used in Write file command. The value must not be changed during operation.
<b>User Interface (Command)</b>		
UserCmd[1:0]	In	User Command. "00": Format command, "01" Reserved, "10": Write file, "11": Read file.
UserFName[15:0]	In	The 1 <sup>st</sup> file name to write/read file (0= FILE0000.BIN, 1=FILE0001.BIN, ...). Valid form 0 to TotalFCap - 1. This input is not used in Format command.
UserFLen[15:0]	In	Total files in the request. Valid from 1 to (TotalFCap – UserFName). This input is not used in Format command.
UserReq	In	Request the new command. Can be asserted to '1' when the IP is Idle (UserBusy='0'). Asserted with valid value on UserCmd, UserFName, and UserFLen signals.
UserBusy	Out	IP Busy status. New request will not be allowed if this signal is asserted to '1'.
TotalFCap[15:0]	Out	Maximum files to store in the SSD. This value is updated after FAT32-IP completes initialization process.
UserError	Out	Direct mapped to NVMeError signal of NVMe IP. Please see more details in NVMe IP datasheet.
UserErrorType[31:0]	Out	Direct mapped to NVMeErrorType signal of NVMe IP. Please see more details in NVMe IP datasheet.
TestPin[63:0]	Out	IP test point.

Signal	Dir	Description
<b>User Interface (Data)</b>		
UserFifoWrCnt[15:0]	In	Write data counter of Received FIFO. Used to check full status. If total FIFO size is less than 16-bit, please fill '1' to upper bit. UserFifoWrEn is asserted when UserFifoWrCnt[15:3] is not equal to all 1.
UserFifoWrEn	Out	Write data valid of Received FIFO.
UserFifoWrData[127:0]	Out	Write data bus of Received FIFO. Synchronous to UserFifoWrEn.
UserFifoRdCnt[15:0]	In	Read data counter of Transmit FIFO. Used to check data available size in FIFO. If total FIFO size is less than 16-bit, please fill '0' to upper bit.
UserFifoEmpty	In	FIFO empty flag of Transmit FIFO. This signal is unused for this IP.
UserFifoRdEn	Out	Read valid of Transmit FIFO.
UserFifoRdData[127:0]	In	Read data returned from Transmit FIFO. Valid in the next clock after UserFifoRdEn is asserted to '1'.
<b>NVMe IP User Interface (Connect to dgIF typeS of NVMe IP)</b> (Please see more details from NVMe IP datasheet)		
NVMeCmd[1:0]	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
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[127:0]	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. This signal is fixed to '1' because NVMe IP is not used.
NVMeFifoRdEn	In	Connect to UserFifoRdEn of NVMe IP
NVMeFifoRdData[127:0]	Out	Connect to UserFifoRdData of NVMe IP

## Timing Diagram

### Initialization

During reset operation (RstB='0'), UserBusy is asserted to '1'. UserBusy is deasserted to '0' after reset is released to '0' and FAT32 IP completes the initialization process. When UserBusy='0', the IP is ready to receive new user command and TotalFCap signal is valid as shown in Figure 3.

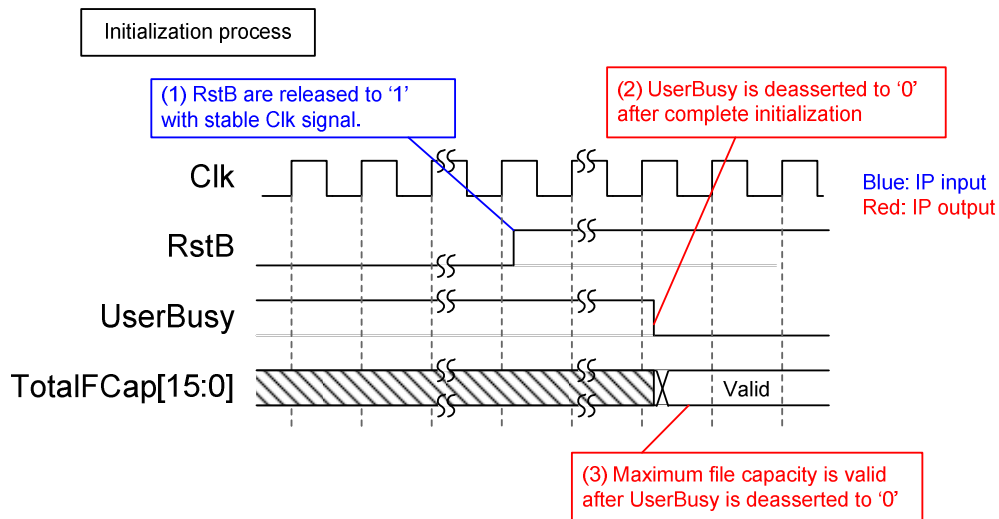


Figure 3: FAT32-IP for NVMe Initialization



## User Interface

User interface is split into two groups, i.e. command interface and data interface. Timing diagram of both groups are compatible to dgIF typeS interface. The input parameters in command interface of FAT32-IP for NVMe are file name and numbers of file instead of physical address and length as shown in Figure 4.

Before sending new command to the IP, UserBusy must be monitored that IP is Idle. UserCmd, UserFName, and UserFLen must be valid during asserting UserReq='1'. UserBusy changes status from '0' to '1' after the command is operated. After that, UserReq can be de-asserted to '0' and the inputs of the next command can be prepared.

Note: UserFName and UserFLen input are ignored in Format command.

For data interface, Transmit FIFO (TXFIFO) is read in Write file command and Received FIFO (RXFIFO) 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.

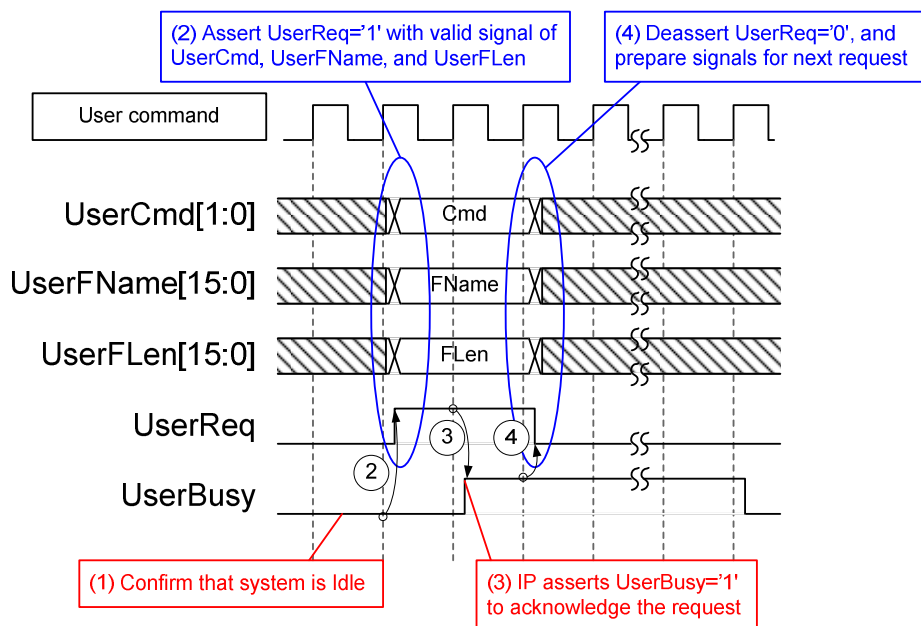


Figure 4: User Interface (Command) Timing diagram

For Write file command, UserFifoRdCnt is monitored until it indicates that data in Transmit FIFO is equal to or more than 512 bytes. After that, UserFifoRdEn is asserted for 32 clock cycles to read 512-byte data, as shown in Figure 5. Similar to general FIFO timing diagram, UserFifoRdData is valid in the next clock after UserFifoRdEn is asserted to '1'.

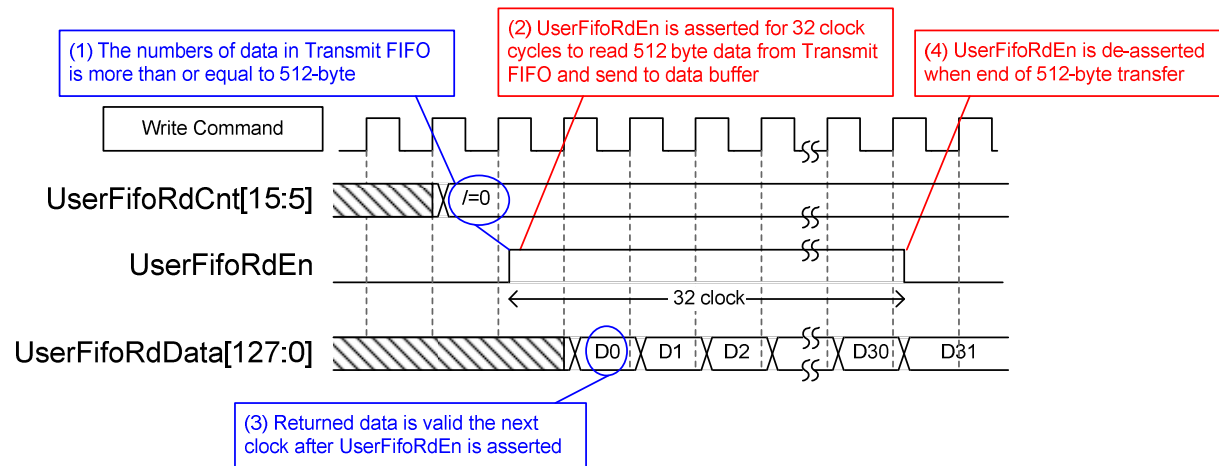


Figure 5: TX FIFO Interface for Write file command

For Read file command, UserFifoWrEn is asserted to '1' with the valid value of UserFifoWrData to send data to Received FIFO. UserFifoWrCnt is monitored to check that free space of Received FIFO. Free space must be more than or equal to 1024 byte, and then 512-byte data is written to Received FIFO.

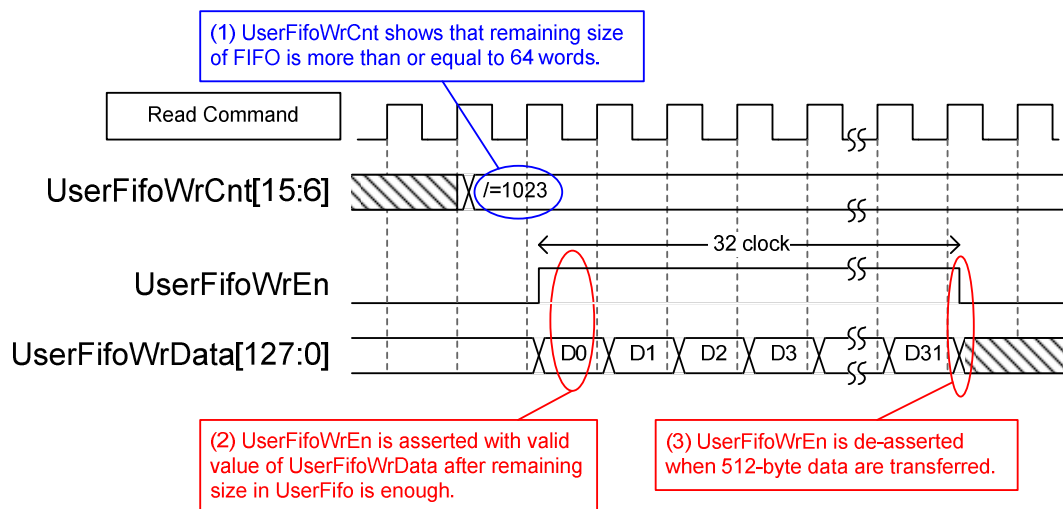


Figure 6: RX FIFO Interface for Read file command

---

## Example usage

The example sequence to use FAT32 IP for NVMe to write and read data with NVMe SSD is shown as follows.

- 1) Deassert RstB='1' to start system initialization. Wait until UserBusy='0'.
- 2) Skip to next step when the SSD does not need to format. To send Format command, assert UserReq to '1' and set UserCmd="00". UserReq is de-asserted to '0' after FAT32 IP asserts UserBusy to '1'. Finally, user waits until IP operation is completed (UserBusy='0').

Note: Format command is required by one of following conditions.

- a) The SSD is the new SSD which still not be formatted by FAT32 IP for NVMe.
  - b) User needs to delete file in the SSD.
  - c) User needs to change a file size parameter.
- 3) Send Write file command (UserCmd="10") or Read file command (UserCmd="11") to the SSD. FSize in Write file and Read file command must use same value as FSize in Format command.

In case of Write file command,

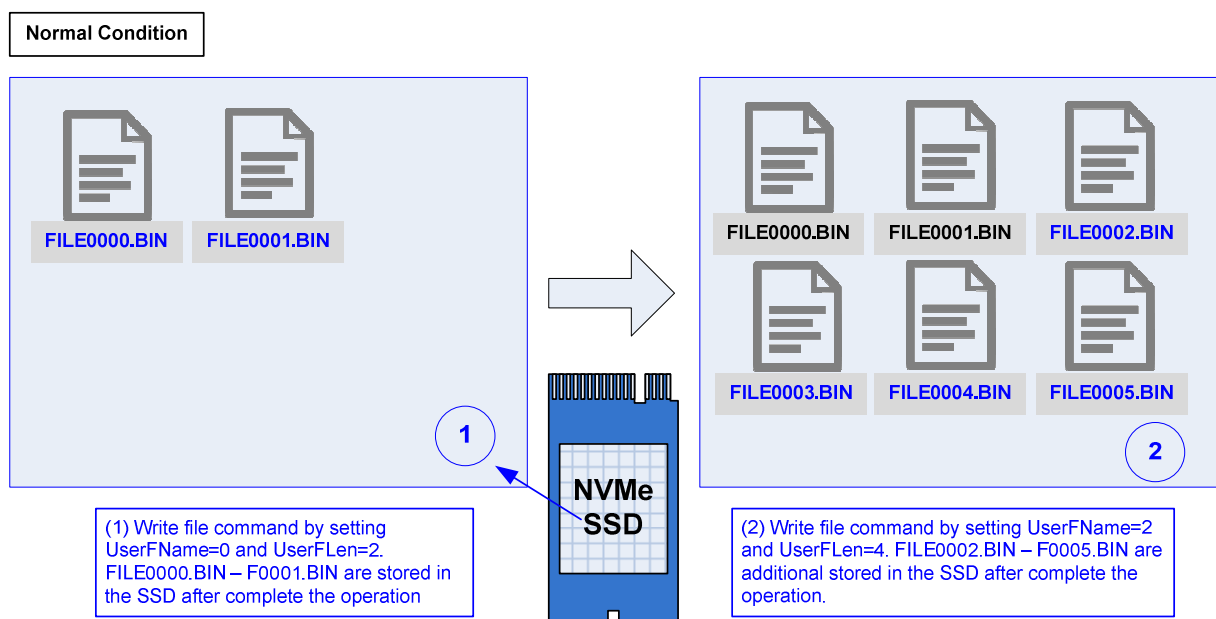
- a) Set FDateY, FDateM, FDateD, FTimeH, FTimeM, and FTimeS to set up "Created date" and "Created time" of the file. All values must not be changed during write file operation.
- b) Send Write file command with a valid UserFName and UserFLen. UserFName must be the next value from the latest file. For example, if the latest file in the SSD is FILE0010.BIN (0010 = 0x000A), UserFName will be set to 0x000B to create FILE0011.BIN as the 1<sup>st</sup> file in this command. UserFLen value must not be more than (TotalFCap – UserFName).
- c) Send total data to TXFIFO. Total data size is equal to UserFLen x FSize. The IP reads the data from TXFIFO by monitoring UserFifoRdCnt.
- d) Wait until IP operation complete (UserBusy='0').

In case of Read file command,

- a) Send Read file command with valid UserFName and UserFLen. (UserFName + UserFLen) value must not be more than total number of files in the SSD.
- b) Read total data from RXFIFO. Total data size is equal to UserFLen x FSize. The IP monitors UserFifoWrCnt[15:6] /= all 1 before sending received data to RXFIFO.
- c) Wait until IP operation complete (UserBusy='0').

**Additional information**

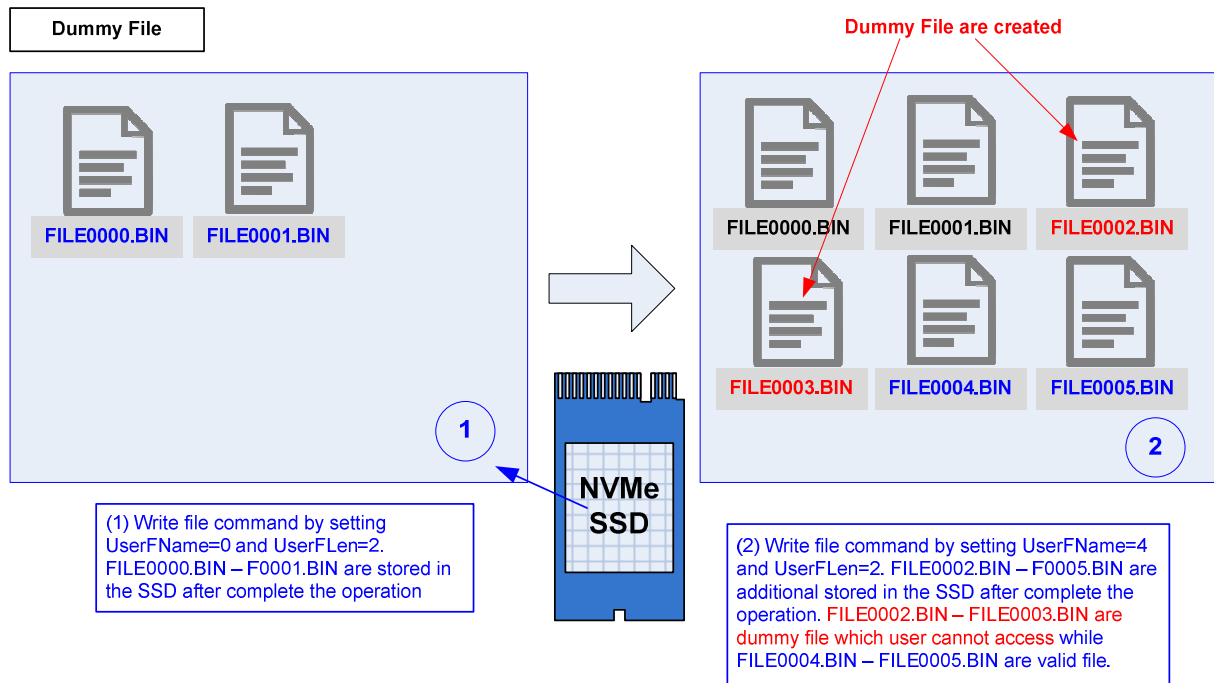
- 1) FAT32 IP for NVMe calculates FAT32 parameters using in all operations from user inputs and SSD capacity. File system in the SSD is specially designed for using in FAT32 IP for NVMe. So, the SSD must not be formatted or written by other system. Only Read file from the SSD is allowed for other system.
- 2) FSize input from user must not change when user writes or reads the file with the SSD. User needs to send Format command after user changes FSize input value. TotalFCap depends on FSize value.
- 3) For Write file command, UserFName input must be the next value from the latest file name, as shown in Figure 7.



**Figure 7: Write file command in normal condition**

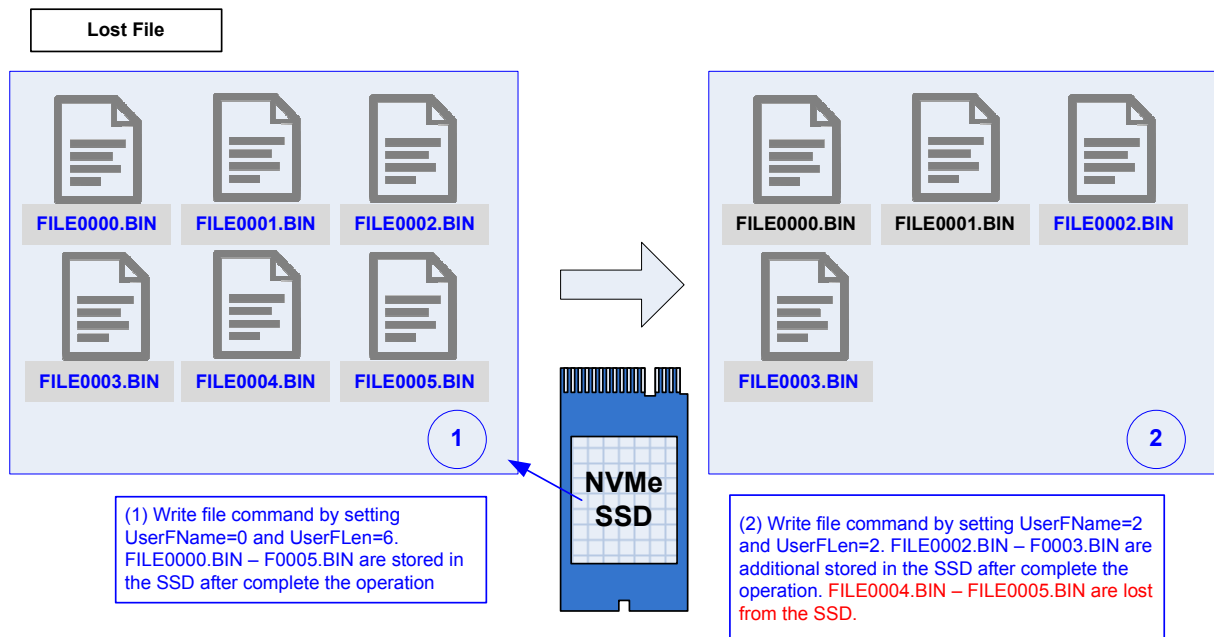
If UserFName is not equal to the next value, the new file will be created with unusual status. The example of the unusual status is as follows.

- a) When UserFName is more than the next value of the latest file, the dummy file which cannot be accessed is created between the latest file and the 1<sup>st</sup> new file, as shown in Figure 8.



**Figure 8: Write file command but UserFName is more than normal value**

- b) When UesrFName is less than the next value of the latest file, the file after the 1<sup>st</sup> new file is corrupt, as shown in Figure 9.



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

## Verification Methods

The FAT32 IP Core for NVMe functionality was verified by simulation and also proved on real board design by using ZC706/KCU105 board.

## Recommended Design Experience

Experience design engineers with a knowledge of Vivado Tools should easily integrate this IP into their design.

## Ordering Information

This product is available directly from Design Gateway Co., Ltd. Please contact Design Gateway 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	Nov-8-2017	Initial Release
1.1	May-9-2018	Correct information