AES256XTSSTGIP with NVMeIP Reference Design

Rev1.00   25-Aug-2023

1 Introduction

This document describes detailed of AES256XTSSTGIP, AES256XTSSTG2XIP and AES256XTSSTG4XIP with NVMe IP core reference design. This document will focus on AES256XTSSTGIP, which can serve as a reference for the other modules in the series. The only difference is the width of the data. In this reference design, AES256XTSSTGIP are applied to encrypt data to NVMe SSD and decrypt data from NVMe SSD via NVMeIP. This document will show detail of hardware design and detail of AES256XTSSTGIP in both encryption section and decryption section.

AES256XTSSTGIP reference design is based on NVMe-IP reference design. Users can find the detailed of NVMe-IP reference design from following link: https://dgway.com/NVMe-IP_X_E.html#docs

2 Hardware Overview

AES256XTSSTGIP reference design is based on NVMe-IP reference design. Figure 2-1 show NVMe-IP reference design block diagram. The operation of NVMe-IP start by Test PC send write or read command parameter to TestGen module. Then TestGen module start write or read pattern data from or to NVMe SSD via FIFO n-bit. NVMe-IP manage pattern data between FIFO n-bit and NVMe SSD.
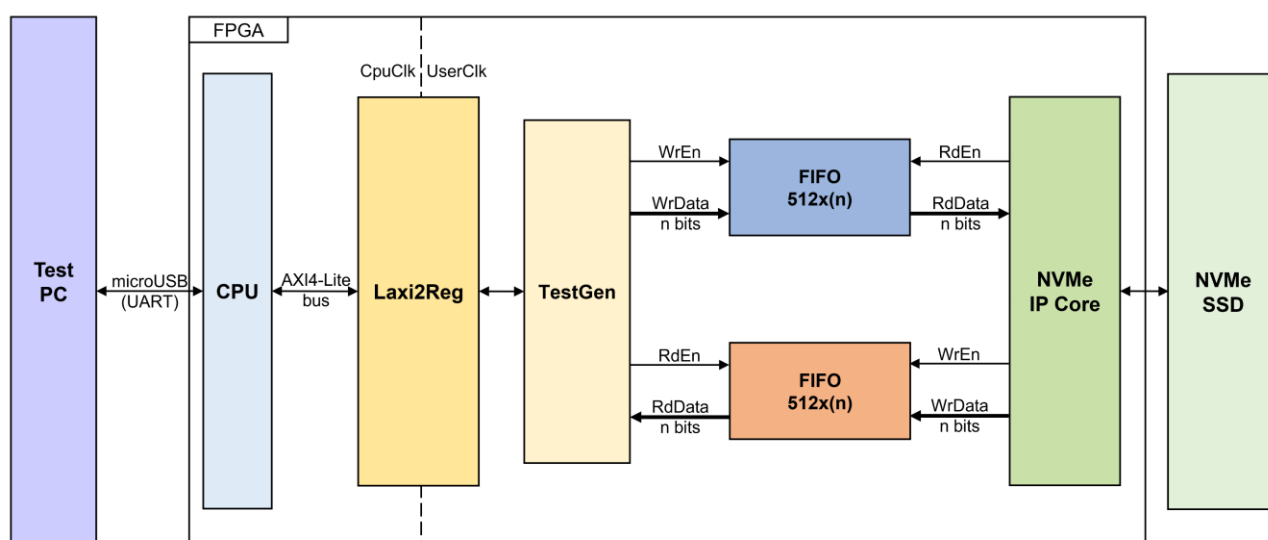

Figure 2-1 NVMe-IP reference design block diagram

Note: n are integers representing the total data path width.

- n=128 for AES256XTSSTGIP   and NVMe-IP
- n=256 for AES256XTSSTG2XIP and NVMeG4-IP
- n=512 for AES256XTSSTG4XIP and NVMeG5-IP

3 AES256XTSSTGIP reference design

AES256XTSSTGIP reference design has concept to encrypt and decrypt data from or to NVMe SSD. So AES256XTSSTGIP is applied to NVMe-IP reference design. Both FIFO n-bit are replaced by AES256XTSSTGIP. One FIFO is replaced by AES256XTSSTG encryption module (AES256XTSSTGENCTOP), another FIFO is replaced by AES256XTSSTG decryption module (AES256XTSSTGDECTOP). Figure 3-1 shows AES256XTSSTGIP was applied to NVMe-IP reference design.
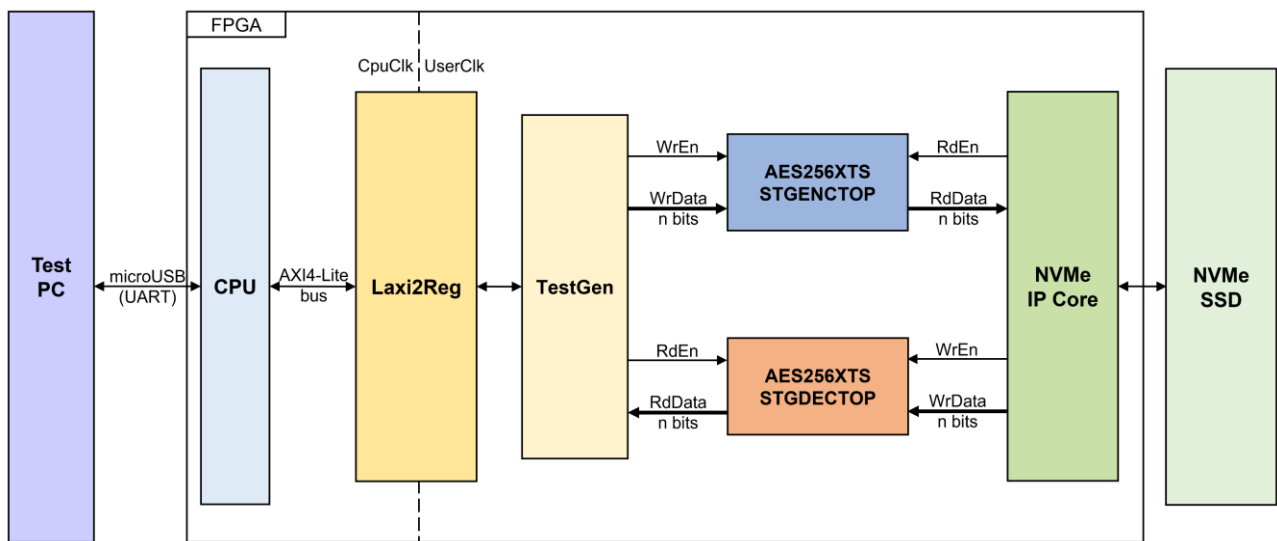

Figure 3-1 AES256XTSSTGIP reference design block diagram

In the reference design, a dip switch is included to facilitate the change of the key used for encryption and decryption. Specifically, SW6 from SW0 to SW3 is utilized for this purpose. Dip switches are employed to configure the encryption key and tweak key for both AES256XTSSTGENCTOP and AES256XTSSTGDECTOP. The position of the dip switch is shown in Figure 3-2.

The configuration of the dip switch is as follows:

- SW0 and SW1 are responsible for configuring the encryption key (EKEY).
- SW2 and SW3 are responsible for configuring the tweak key (TKEY).

The specific relationship between the dip switch settings and the encryption key and tweak key configuration as shown in Table 3-1 and Table 3-2.
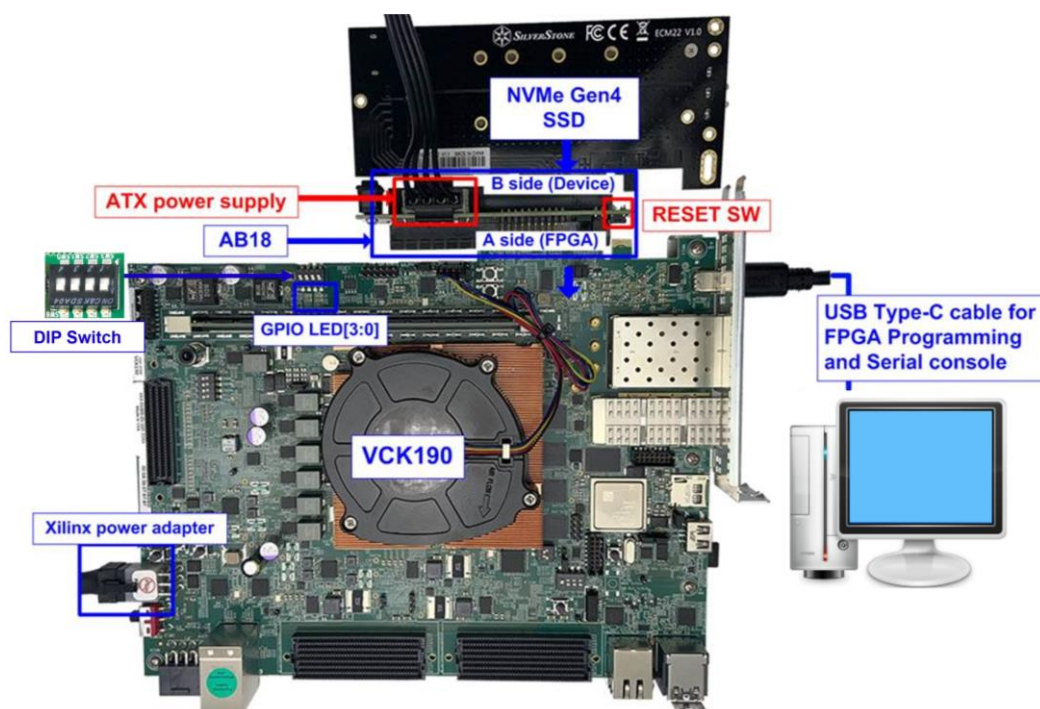
Figure 3-2 NVMe-IP demo setup on VCK190 Evaluation board

Table 3-1 Relationship between the dip switch and the encryption key

SW1	SW0	Encryption key Value
0	0	0x00112233445566778899AABBCCDDEEFF00112233445566778899AABBCCDDEEFF
0	1	0xFFEEDDCCBBAA99887766554433221100FFEEDDCCBBAA99887766554433221100
1	0	0x00001111222233334444555566667777888899999AAAABBBBCCCCDDDDEEEEFFFF
1	1	0xFFFFEEEEDDDDCCCCBBBBAAAA9999888877776666555544443333222211110000

Table 3-2 Relationship between the dip switch and the tweak key

| SW3	SW2	Tweak key Value
0	0	0x00112233445566778899AABBCCDDEEFF00112233445566778899AABBCCDDEEFF
0	1	0xFFEEDDCCBBAA99887766554433221100FFEEDDCCBBAA99887766554433221100
1	0	0x00001111222233334444555566667777888899999AAAABBBBCCCCDDDDEEEEFFFF
1	1	0xFFFFEEEEDDDDCCCCBBBBAAAA9999888877776666555544443333222211110000

3.1 AES256XTSSTGENCTOP
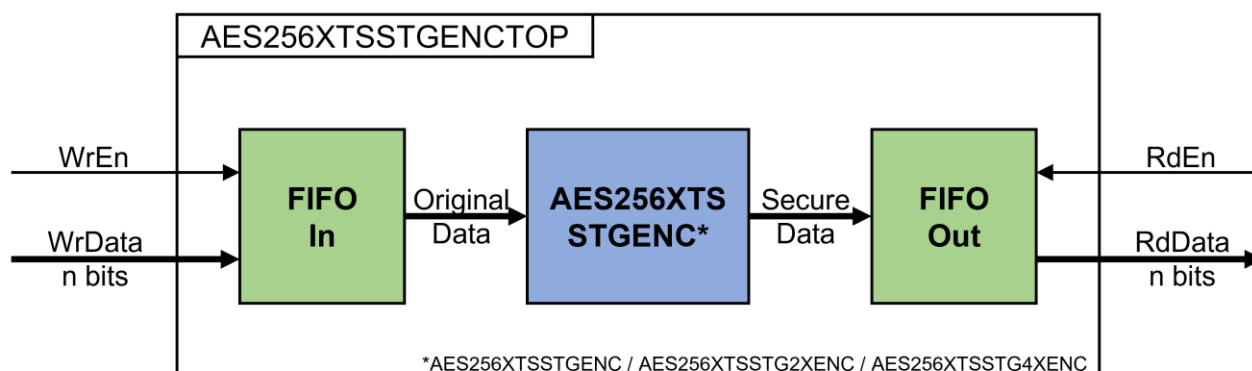
Figure 3-3 AES256XTSSTGENCTOP block diagram

The AES256XTSSTGENCTOP consists of two n-bit FIFO and one AES256XTSSTGENC, as shown in Figure 3-3.

3.1.1 parameter setting

For AES256XTSSTGENC encryption key setting, the process will start when a change in the DIPSW is detected and UserBusy='0', the rKeyInReady will be cleared, and the rKeyInValid will be active only one clock in the next cycle. Once the key generation process is complete, the wKeyInFinish will be active only one clock, and the rKeyInReady will be asserted to '1' in the next cycle. The KeyIn used in the encryption process will change according to the value of the DIPSW, as shown in Table 3-1 and Table 3-2.

The encryption process starts when the user sends a write command to NVMe-IP, as shown in Figure 3-4. After the NVMe-IP sets UserBusy='1', rInitValid will be asserted to '1', and rIvIn is set to the value of UserAddr with the left-padded zeros. To prevent the loss of NVMe commands, if the user changes the key while sending commands. rInitValid is specifically designed to hold the command until the IP is ready for processing. the rInitStart='1' when rInitValid='1' and rKeyInReady='1'. The rInitValid will be cleared when rInitStart='1'.
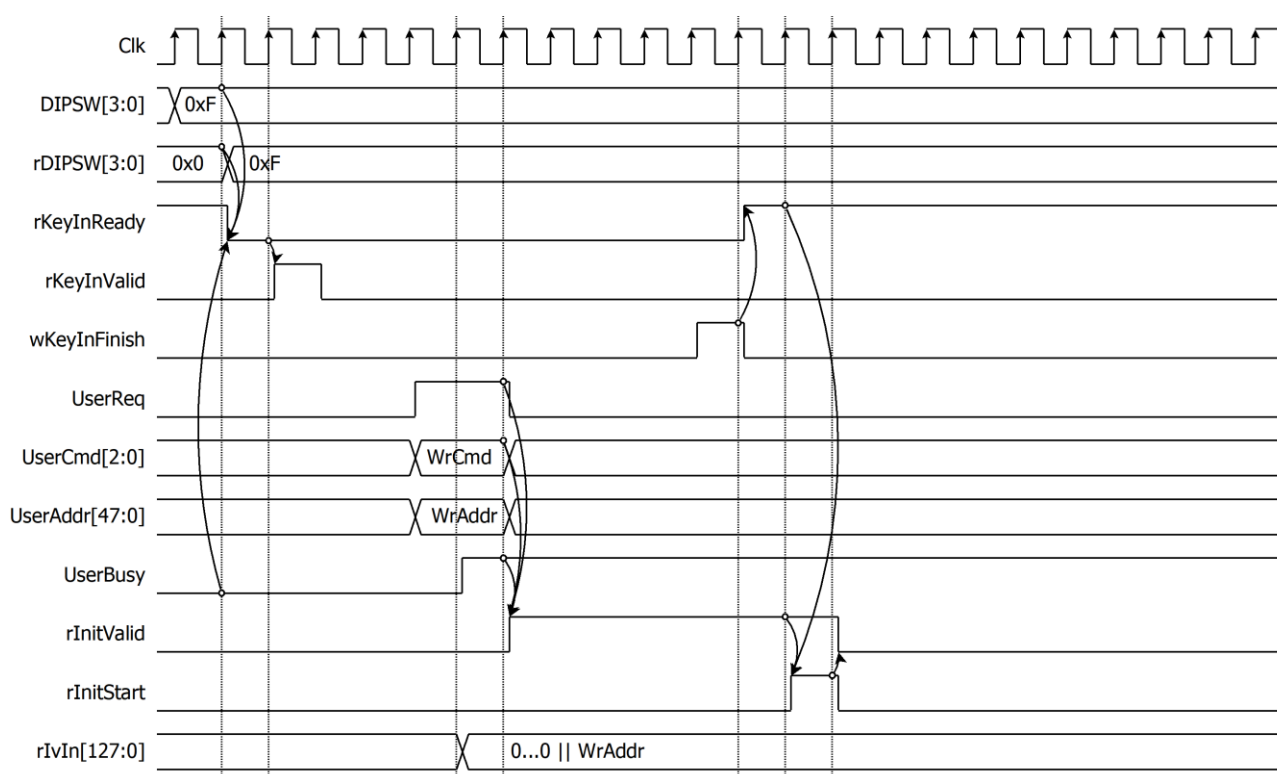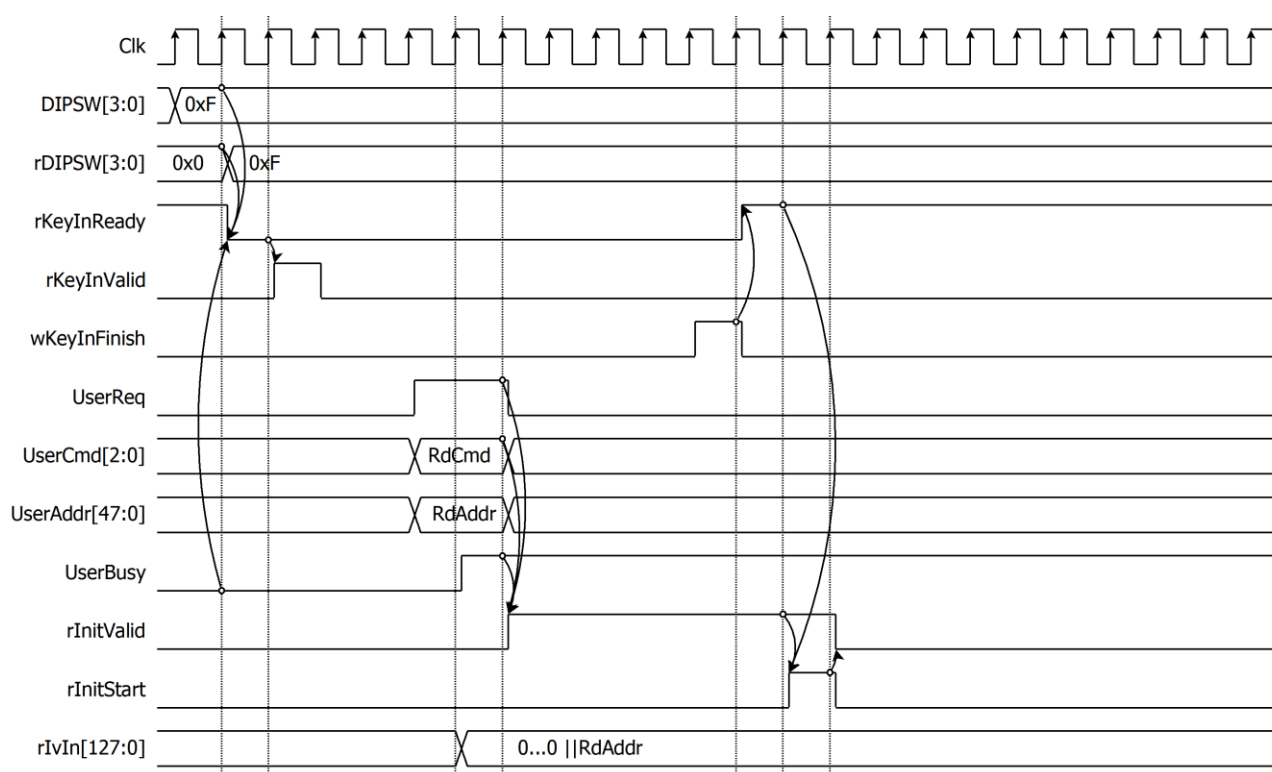
Figure 3-4 Timing diagram example of encryption parameter setting

3.1.2 Encryption

For the encryption process, n-bit plain data, which will be input data for AES256XTSSTGENC, is stored in FIFOIn. FIFOIn has wFfRdEn as the read enable signal of the FIFO, and wFfRdEn is asserted to '1' when FIFOIn is not empty (wU2IPFfEmpty!='0') and wDataInReady='1' and FIFOOut has more than 32 words of available space (wU2IPFfDataCnt(8 downto 5) is not equal to "1111"). rDataInValid is delayed one clock cycle from wFfRdEn. When the wDataOutValid is set to '1', the FIFOOut uses as the write enable signal for the FIFO. The encrypted data is then stored from the wDataOut[n-1:0].
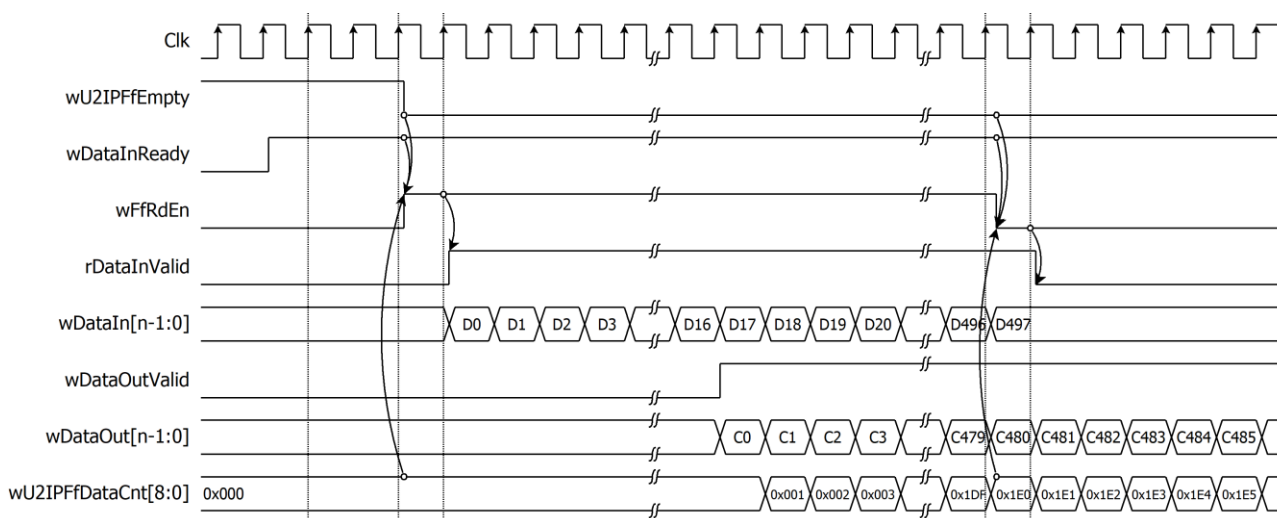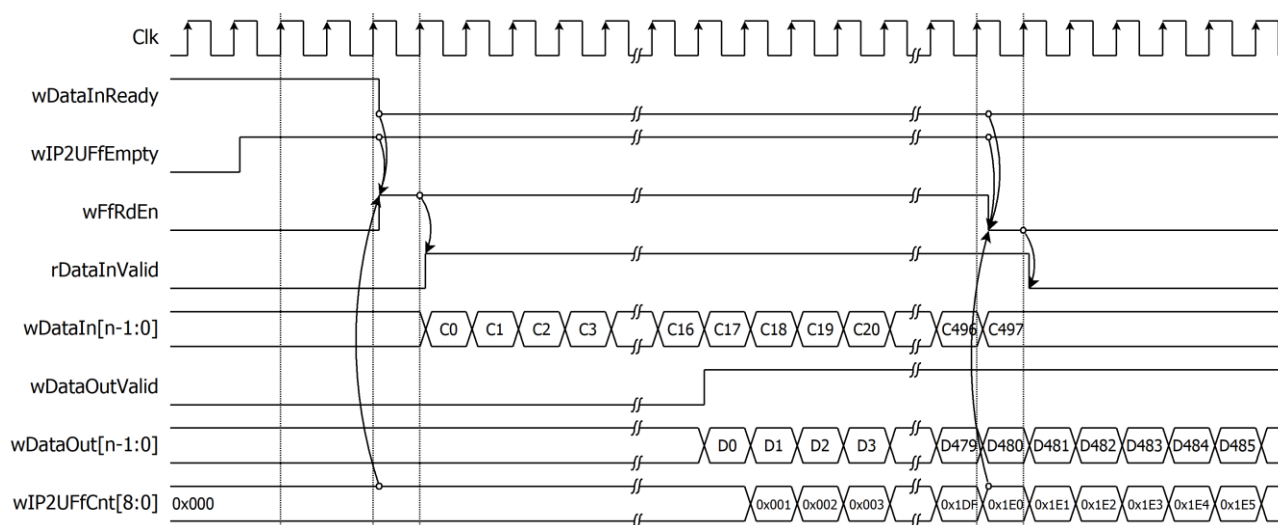

Figure 3-5 Timing diagram example of data encryption

3.2 AES256XTSSTGDECTOP
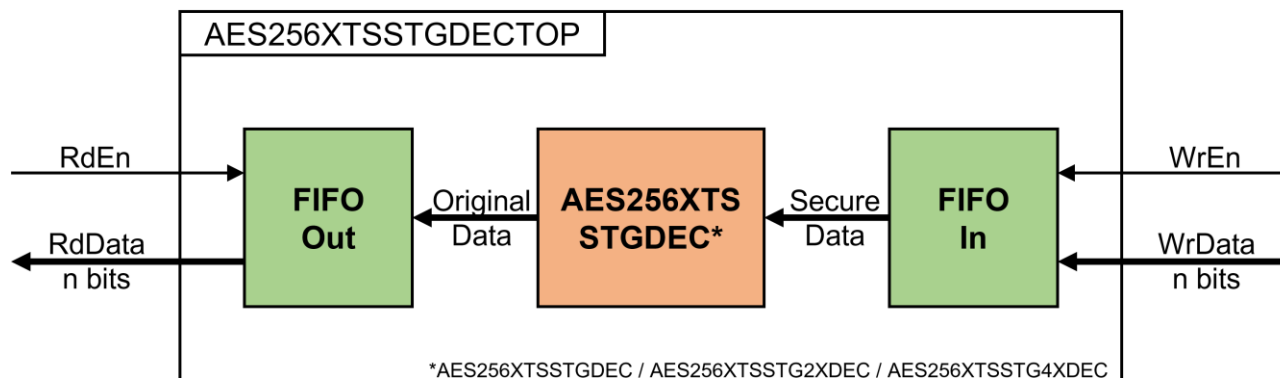
Figure 3-6 AES256XTSSTGDECTOP block diagram

The AES256XTSSTGDECTOP consists of two n-bit FIFO and one AES256XTSSTGDEC, as shown in Figure 3-6.

3.2.1 parameter setting

For AES256XTSSTGDEC encryption key setting, the process will start when a change in the DIPSW is detected and UserBusy='0', the rKeyInReady will be cleared, and the rKeyInValid will be active only one clock in the next cycle. Once the key generation process is complete, the wKeyInFinish will be active only one clock, and the rKeyInReady will be asserted to '1' in the next cycle. The KeyIn used in the encryption process will change according to the value of the DIPSW, as shown in Table 3-1 and Table 3-2.

The decryption process starts when the user sends a read command to NVMe-IP, as shown in Figure 3-7. After the NVMe-IP sets UserBusy='1', rInitValid will be asserted to '1', and rIvIn is set to the value of UserAddr with the left-padded zeros. To prevent the loss of NVMe commands, if user change the key while sending commands. rInitValid is specifically designed to hold the command until the IP ready for processing. the rInitStart='1' when rInitValid='1' and rKeyInReady='1'. The rInitValid will be cleared when rInitStart='1'.

Figure 3-7 Timing diagram example of decryption parameter setting

3.2.2 Decryption

For the decryption process, n-bit cipher data, which will be input data for AES256XTSSTGDEC, is stored in FIFOIn. FIFOIn has wFfRdEn as the read enable signal of the FIFO, and wFfRdEn is asserted to '1' when FIFOIn is not empty (wIP2UFfEmpty!='0') and wDataInReady='1' and FIFOOut has more than 32 words of available space (wIP2UFfCnt(8 downto 5) is not equal to "1111"). rDataInValid is delayed one clock cycle from wFfRdEn. When the wDataOutValid is set to '1', the FIFOOut uses as the write enable signal for the FIFO. The decrypted data is then stored from the wDataOut[n-1:0].

Figure 3-8 Timing diagram example of data decryption

3.3 Auto IV Increment mode

In the reference design example, Full Drive Encryption (FDE) is employed to encrypt all data on the SSD. The Auto IV Increment mode is employed for the purpose of ensuring continuous data encryption. As shown in Figure 3-9, the data is encrypted using UserAddr (or Logical Block Address - LBA) as the initialization vector (IV). For example, if the user intends to write data and the starting UserAddr/LBA is 0x0000_0000_0123, the first 512 bytes of data will be written to the address 0x0000_0000_0123. Subsequently, the next 512 bytes of data will be written to the address 0x0000_0000_0124, and this pattern will continue until the end of the data.
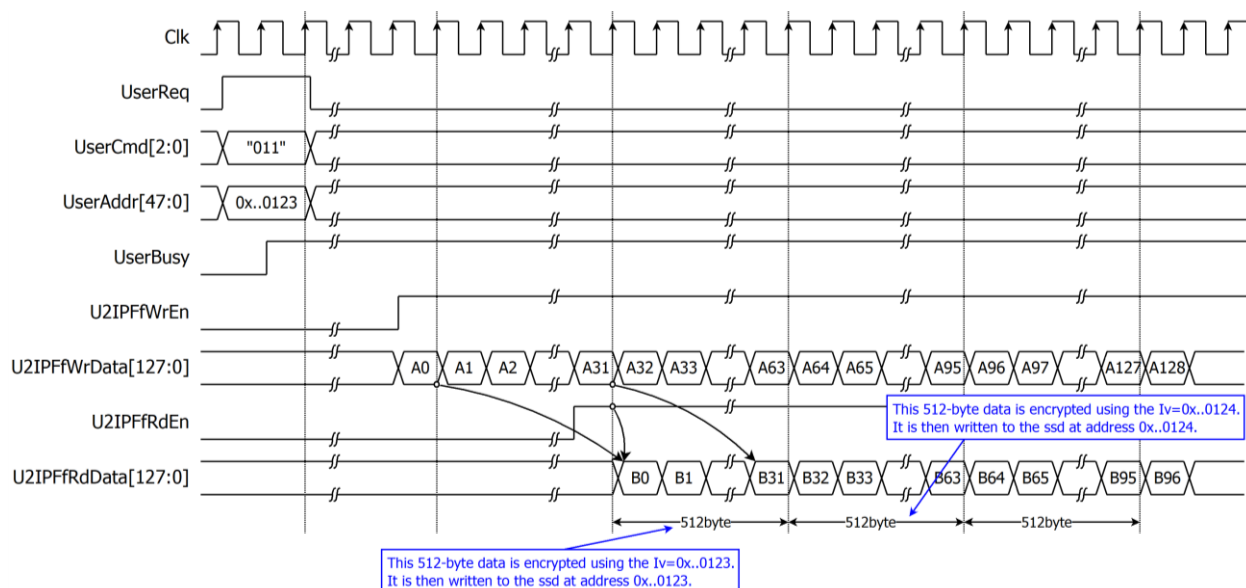
Figure 3-9 AES256XTSSTGENCTOP operation timing diagram for AES256XTSSTGIP

In case the user intends to read data and the starting UserAddr/LBA is 0x0000_0000_0124, the first 512 bytes of data will be read from the address 0x0000_0000_0124. Subsequently, the next 512 bytes of data will be read from the address 0x0000_0000_0125, and this pattern will continue until the end of the data.
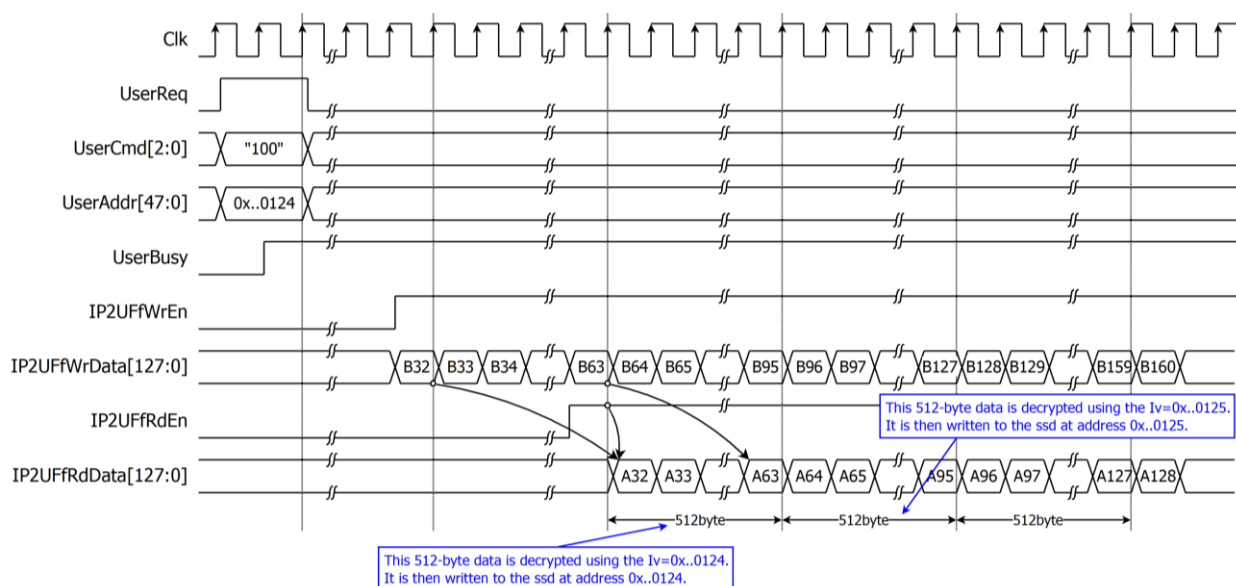
Figure 3-10 AES256XTSSTGDECTOP operation timing diagram for AES256XTSSTGIP

4 Example Test Result

The objective is to compare the read and write performance of an 800GB Intel® Optane™ P5800X storage device utilizing AES256XTSSTG2XIP with NVMeG4IP, as opposed to using NVMeG4IP alone. The AES256XTSSTG2XIP is fully compatible with NVMeG4IP. The tests were conducted on a VCK190 Evaluation Board with PCIe Gen4, the data read speeds of around 6,000 Mbyte/sec and write speeds of around 7,400 Mbyte/sec for both setups.

(a) AES256XTSSTGIP + NVMeG4IP   (b) Only NVMeG4IP

Figure 4-1 Input and Test result when running Write command

(a) AES256XTSSTGIP + NVMeG4IP   (b) Only NVMeG4IP

Figure 4-2 Input and Test result when running Read command

5 Revision History

Revision	Date	Description
1.00	25-Aug-2023	Initial version release