# AES256XTSSTGIP Demo Instruction

# AES256XTSSTGIP Demo Instruction

Rev1.00 25-Aug-2023

This document describes the instruction to demonstrate the operation of AES256XTSSTGIP on FPGA development boards. In the demonstration, AES256XTSSTGIP are used to encrypt and decrypt data between two memories in FPGA. User can fill memory with plain or cipher data patterns, set encryption key, tweakable key, Initialization Vector (IV) and control test operation via Nios II Command Shell.

## 1 Environment Setup

To operate AES256XTSSTGIP demo, please prepare following test environment.

1) FPGA development board
   - Agilex7 I-series development kit. or
   - Arria10 SoC Development board.
2) Test PC.
3) Micro USB cable for JTAG connection connecting between FPGA boards and Test PC.
4) Quartus programmer for programming FPGA and Nios II command shell, installed on PC.
5) SOF file named “AES256XTSSTGIP.sof” or “AES256XTSSTG4XIP.sof” (To download these files, please visit our web site at www.design-gateway.com)


Figure 1-1 AES256XTSSTGIP demo environment on Agilex7 I-series board

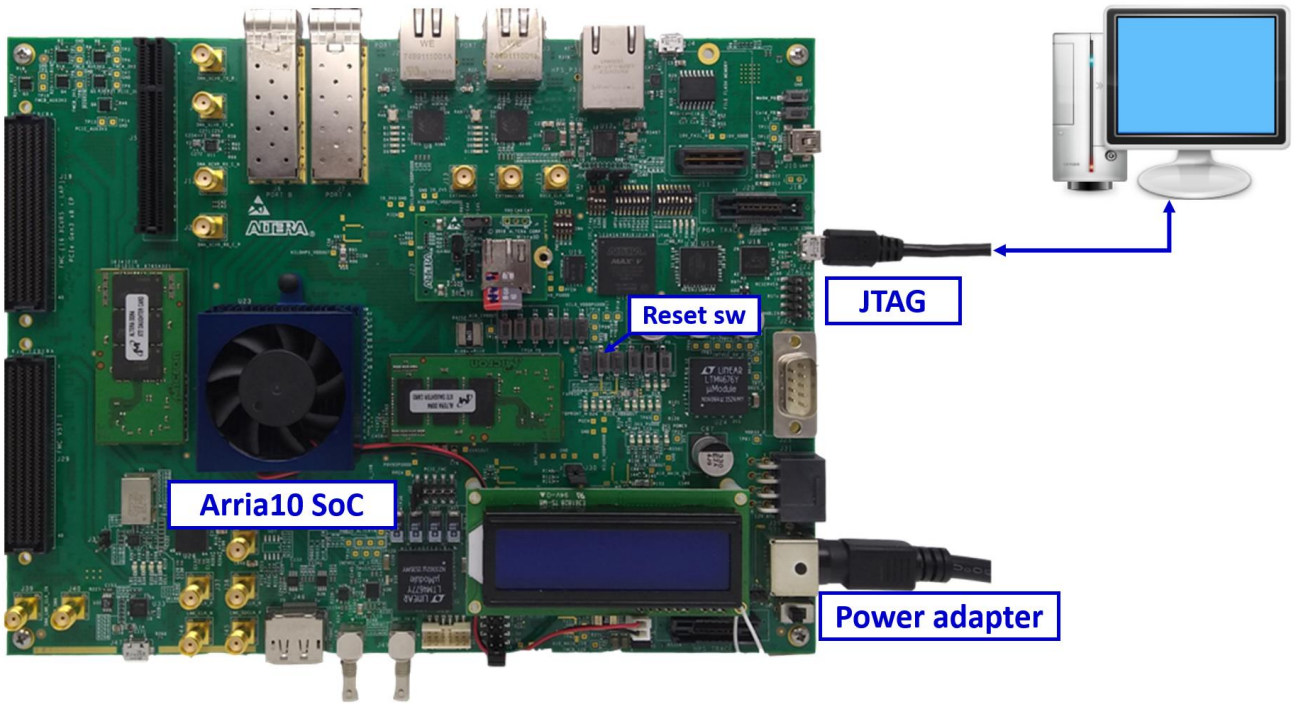



Figure 1-2 AES256XTSSTGIP demo environment on Arria10 SoC board

# 2 FPGA development board setup

1) Make sure power switch is off and connect power supply to FPGA development board.
2) Connect USB cables between FPGA board and PC via micro-USB ports.
3) Turn on power switch for FPGA board.
4) Open Quartus Programmer to program FPGA through USB-1 by following step.
    i) Click “Hardware Setup…” to select
        - AGI FPGA Development Kit [USB-1] for Agilex7 I-series
        - USB-BlasterII [USB-1] for Arria10 SoC
    ii) Click “Auto Detect” and select FPGA number.
    iii) Select FPGA device icon (Agilex or A10SoC).
    iv) Click “Change File” button, select SOF file in pop-up window and click “open” button.
    v) Check “program”.
    vi) Click “Start” button to program FPGA.
    vii) Wait until Progress status is equal to 100%.

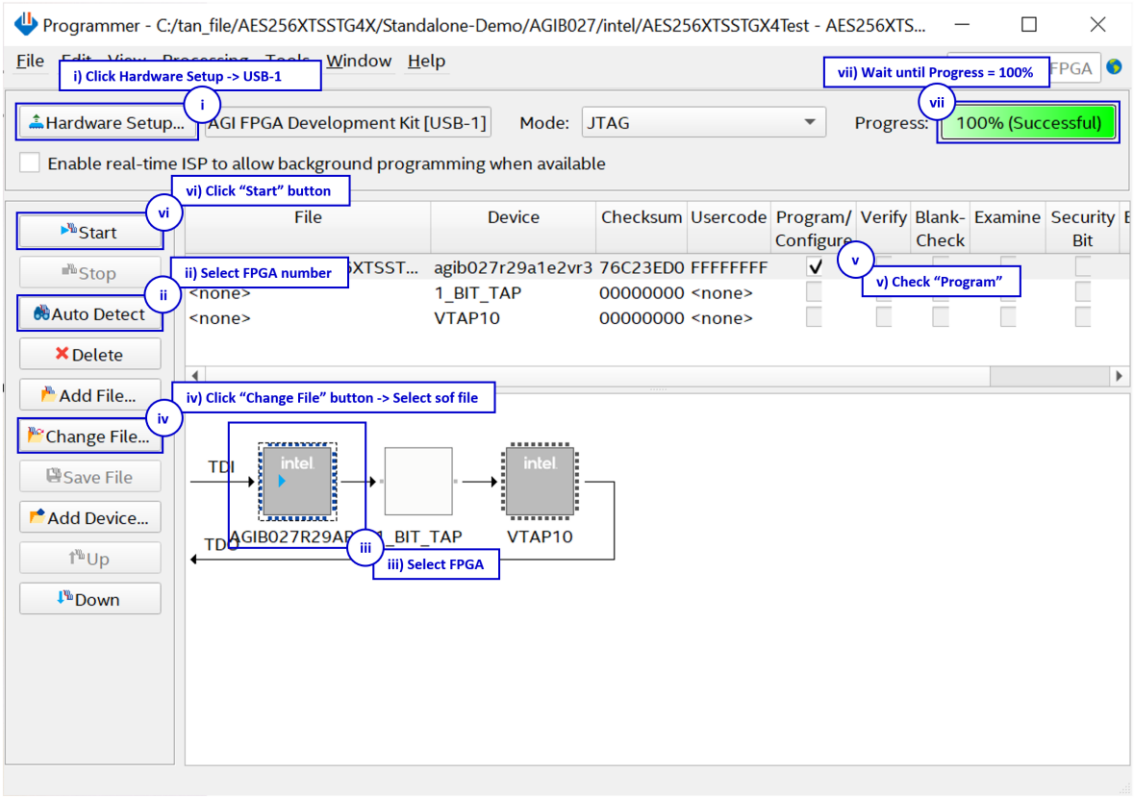



Figure 2-1 FPGA Programmer for Agilex

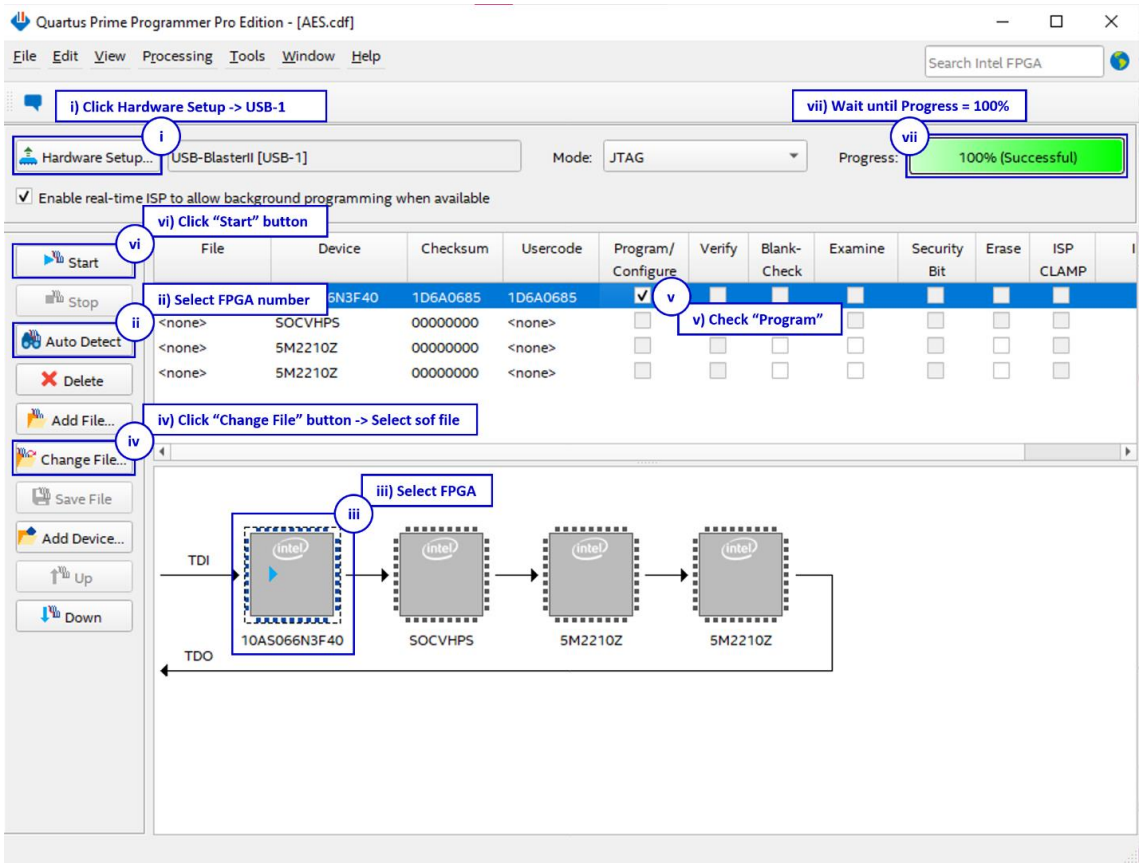



Figure 2-2 FPGA Programmer for A10SoC

For A10SoC after program SOF file complete, Quartus Prime will show popup message of Intel FPGA IP Evaluation Mode Status as shown in Figure 2-3. Please do not press cancel button.

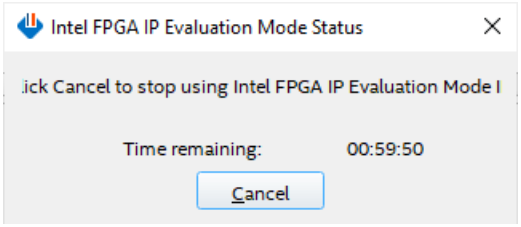



Figure 2-3 Intel FPGA IP Evaluation Mode Status

# 3 Nios II Command Shell

User can fill RAMs with plain or cipher data patterns, set encryption key, set tweakable key, IV and control test operation via Nios II Command Shell. When configuration is completed, AES256XTSSTGdemo command menu will be displayed as shown in Figure 3-1. The detailed information of each menu is described in topic 4.

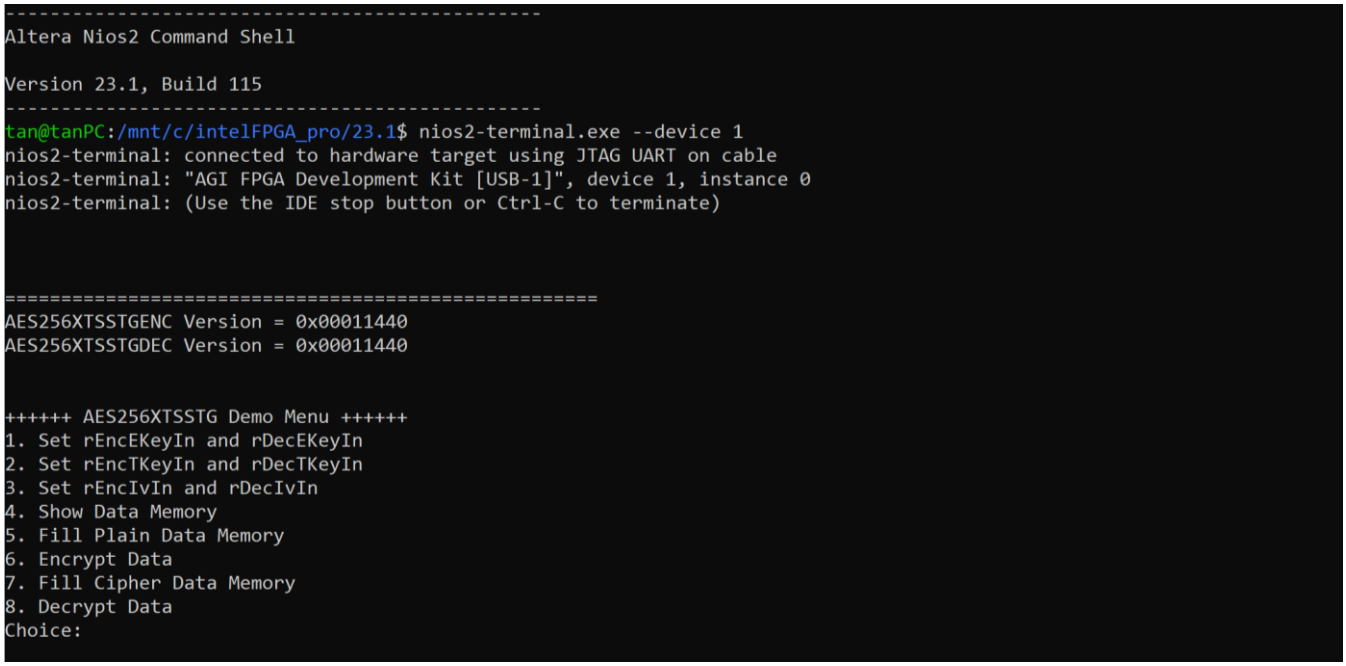



Figure 3-1 Nios II Command Shell

# 4 Command detail and testing result

## 4.1 Set encryption key

Step to set encryption key as follows

a) Select “1. Set rEncEKeyIn and rDecEKeyIn”.
b) Current rEncEKeyIn will be displayed on Nios II Command Shell as shown in Figure 4-1.
c) Set new rEncEKeyIn: User is allowed to input new key in hex format or press “enter” to skip setting new key. Then the current encryption key is printed again.
d) Current rDecEKeyIn key will be displayed on Nios II Command Shell.
e) Set new rDecEKeyIn key: User is allowed to input new key in hex format or press “enter” to use rEncEKeyIn as rDecEKeyIn. Then the current decryption key is printed again.

```
++++++ AES256XTSSTG Demo Menu ++++++
1. Set rEncEKeyIn and rDecEKeyIn
2. Set rEncTKeyIn and rDecTKeyIn
3. Set rEncIvIn and rDecIvIn
4. Show Data Memory
5. Fill Plain Data Memory
6. Encrypt Data
7. Fill Cipher Data Memory
8. Decrypt Data
Choice: 1

+++ Set rEncEKeyIn and rDecEKeyIn +++
              rEncEKeyIn= 0x0000000000000000000000000000000000000000000000000000000000000000
(enter to use rEncEKeyIn)= 0x27182818284590452353602874713526624977572470936999595749669676 27
          new rEncEKeyIn= 0x27182818284590452353602874713526624977572470936999595749669676 27

              rDecEKeyIn= 0x0000000000000000000000000000000000000000000000000000000000000000
(enter to use rEncEKeyIn)= 0x
          new rDecEKeyIn= 0x27182818284590452353602874713526624977572470936999595749669676 27
```

Figure 4-1 Set rEncEKeyIn and rDecEKeyIn example

## 4.2 Set tweakable key

Step to set tweakable key as follows

a) Select “2. Set rEncTKeyIn and rDecTKeyIn”.
b) Current rEncTKeyIn will be displayed on Nios II Command Shell as shown in Figure 4-2.
c) Set new rEncTKeyIn: User is allowed to input new key in hex format or press “enter” to skip setting new key. Then the current encryption key is printed again.
d) Current rDecTKeyIn will be displayed on Nios II Command Shell.
e) Set new rDecTKeyIn: User is allowed to input new key in hex format or press “enter” to use rEncTKeyIn as rDecTKeyIn. Then the current decryption key is printed again.

```
++++++ AES256XTSSTG Demo Menu ++++++
1. Set rEncEKeyIn and rDecEKeyIn
2. Set rEncTKeyIn and rDecTKeyIn
3. Set rEncIvIn and rDecIvIn
4. Show Data Memory
5. Fill Plain Data Memory
6. Encrypt Data
7. Fill Cipher Data Memory
8. Decrypt Data
Choice: 2

+++ Set rEncTKeyIn and rDecTKeyIn +++
               rEncTKeyIn= 0x0000000000000000000000000000000000000000000000000000000000000000
(enter to use rEncTKeyIn)= 0x3141592653589793238462643383279502884197169399375105820974944592
          new rEncTKeyIn= 0x3141592653589793238462643383279502884197169399375105820974944592

               rDecTKeyIn= 0x0000000000000000000000000000000000000000000000000000000000000000
(enter to use rEncTKeyIn)= 0x
          new rDecTKeyIn= 0x3141592653589793238462643383279502884197169399375105820974944592
```

Figure 4-2 Set rEncTKeyIn and rDecTKeyIn example

## 4.3 Set encryption/decryption IV

Step to Set encryption/decryption IV as follows

a) Select “3. Set rEncIvIn and rDecIvIn”.
b) Current rEncIvIn will be displayed on Nios II Command Shell as shown in Figure 4-3.
c) Set new rEncIvIn: User is allowed to input new IV in hex format or press “enter” to skip setting new key. Then the current encryption IV is printed again.
d) Current rDecIvIn will be displayed on Nios II Command Shell.
e) Set new rDecIvIn: User is allowed to input new IV in hex format or press “enter” to use rEncIvIn as rDecIvIn. Then the current decryption IV is printed again.

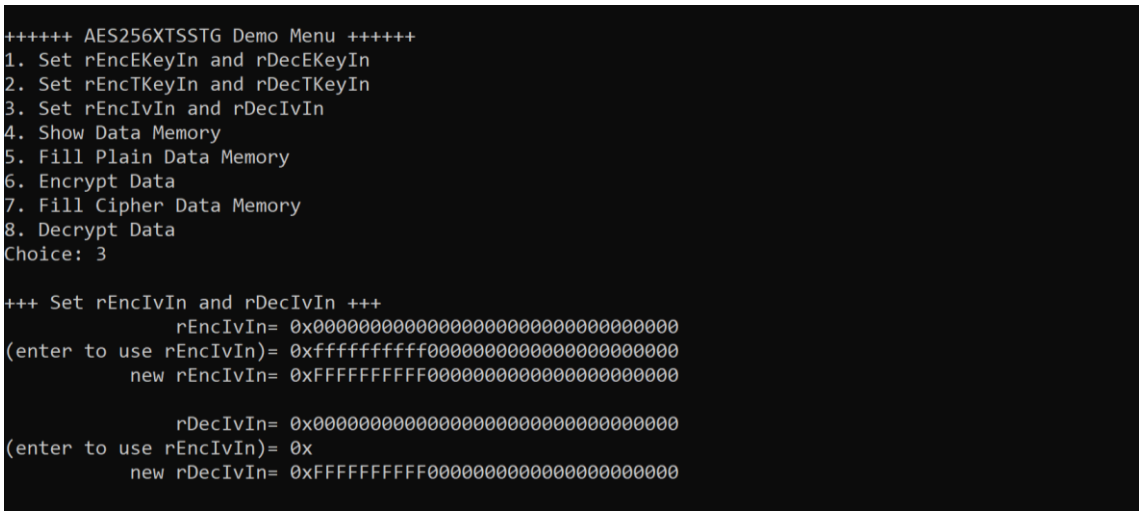



Figure 4-3 Set rEncIvIn and rDecIvIn example

## 4.4 Show Data Memory

To show data in memory, user can select “4. Show Data Memory” and input the desired number of 512-byte data to show. Both plain data and cipher data will be displayed in table-form as shown in Figure 4-4. User have the option to press “enter” to use the default value.

```
++++++ AES256XTSSTG Demo Menu ++++++
1. Set rEncEKeyIn and rDecEKeyIn
2. Set rEncTKeyIn and rDecTKeyIn
3. Set rEncIvIn and rDecIvIn
4. Show Data Memory
5. Fill Plain Data Memory
6. Encrypt Data
7. Fill Cipher Data Memory
8. Decrypt Data
Choice: 4

+++ Show Data Memory +++
Number of 512-byte Data in decimal (enter = 3): 1

                Plain Data                          Cipher Data
Addr#  .0.....3 .4.....7 .8.....B .C.....F  .0.....3 .4.....7 .8.....B .C.....F
0000:  F7607222 67070CA1 9B311167 58731F6D  3D255128 CEE78DC3 390CBA29 E4A48B50
0001:  2E21C52A 274B1231 85FF7DAB E1FEBEFF  A7EF0847 46D98983 AB55C9D9 FF8F4F17
0002:  D5688A6F 6625E5CE 0EF9DDFF C5C74AC0  E08F8B8B CC3DEDF8 72E969F1 6242FC41
0003:  6D0B0AF7 47D2C97F E9635B8B 6B3D2BE1  2F1E3E48 2DFAD4B7 8FC27477 62655B22
0004:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
0005:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
0006:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
0007:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
0008:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
0009:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
000A:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
000B:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
000C:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
000D:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
000E:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
000F:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
0010:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
0011:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
0012:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
0013:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
0014:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
0015:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
0016:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
0017:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
0018:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
0019:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
001A:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
001B:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
001C:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
001D:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
001E:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
001F:  00000000 00000000 00000000 00000000  00000000 00000000 00000000 00000000
```

Figure 4-4 Displayed Data example

## 4.5 Fill Plain Data Memory

Step to fill plain data in memory as follows

a) Select “5. Fill Plain Data Memory”.
b) Input the desired number of 512-byte data. User can press “enter” to use the default value of plain data. user can a select data pattern.
c) There are four pattern to fill memory.
    a. zero pattern
    b. 8-bit counter
    c. 16-bit counter
    d. 32-bit counter
d) Whole plain-data memory is filled with selected data pattern.

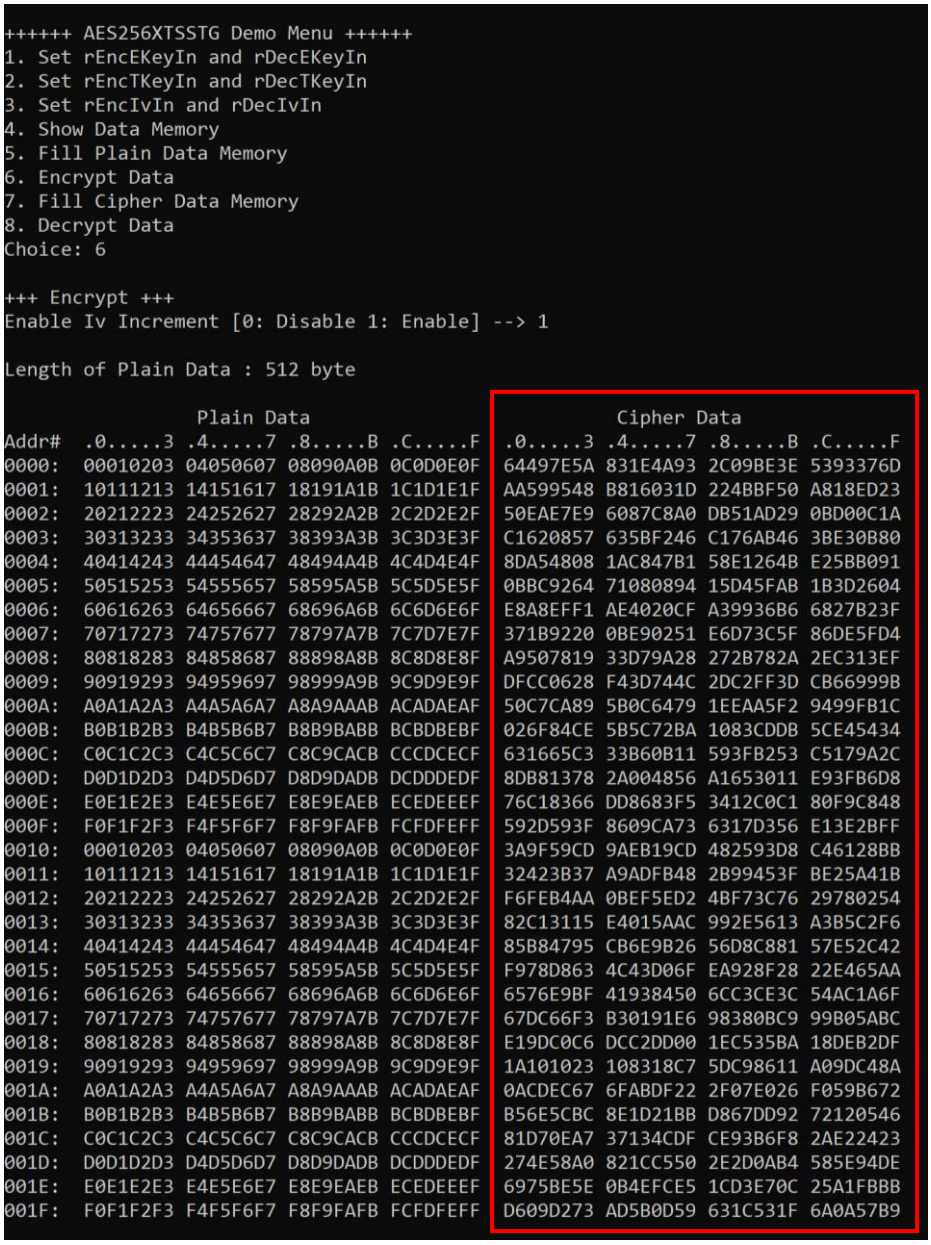
```
++++++ AES256XTSSTG Demo Menu ++++++
1. Set rEncEKeyIn and rDecEKeyIn
2. Set rEncTKeyIn and rDecTKeyIn
3. Set rEncIvIn and rDecIvIn
4. Show Data Memory
5. Fill Plain Data Memory
6. Encrypt Data
7. Fill Cipher Data Memory
8. Decrypt Data
Choice: 5

+++ Fill Plain Data Memory +++
Number of 512-byte Data in decimal (enter = 5): 1
a. zero pattern
b. 8-bit counter
c. 16-bit counter
d. 32-bit counter
Choice: b

Length of Plain Data : 512 byte

                Plain Data                                  Cipher Data
Addr#  .0.....3 .4.....7 .8.....B .C.....F  .0.....3 .4.....7 .8.....B .C.....F
0000:  00010203 04050607 08090A0B 0C0D0E0F  00000000 00000000 00000000 00000000
0001:  10111213 14151617 18191A1B 1C1D1E1F  00000000 00000000 00000000 00000000
0002:  20212223 24252627 28292A2B 2C2D2E2F  00000000 00000000 00000000 00000000
0003:  30313233 34353637 38393A3B 3C3D3E3F  00000000 00000000 00000000 00000000
0004:  40414243 44454647 48494A4B 4C4D4E4F  00000000 00000000 00000000 00000000
0005:  50515253 54555657 58595A5B 5C5D5E5F  00000000 00000000 00000000 00000000
0006:  60616263 64656667 68696A6B 6C6D6E6F  00000000 00000000 00000000 00000000
0007:  70717273 74757677 78797A7B 7C7D7E7F  00000000 00000000 00000000 00000000
0008:  80818283 84858687 88898A8B 8C8D8E8F  00000000 00000000 00000000 00000000
0009:  90919293 94959697 98999A9B 9C9D9E9F  00000000 00000000 00000000 00000000
000A:  A0A1A2A3 A4A5A6A7 A8A9AAAB ACADAEAF  00000000 00000000 00000000 00000000
000B:  B0B1B2B3 B4B5B6B7 B8B9BABB BCBDBEBF  00000000 00000000 00000000 00000000
000C:  C0C1C2C3 C4C5C6C7 C8C9CACB CCCDCECF  00000000 00000000 00000000 00000000
000D:  D0D1D2D3 D4D5D6D7 D8D9DADB DCDDDEDF  00000000 00000000 00000000 00000000
000E:  E0E1E2E3 E4E5E6E7 E8E9EAEB ECEDEEEF  00000000 00000000 00000000 00000000
000F:  F0F1F2F3 F4F5F6F7 F8F9FAFB FCFDFEFF  00000000 00000000 00000000 00000000
0010:  00010203 04050607 08090A0B 0C0D0E0F  00000000 00000000 00000000 00000000
0011:  10111213 14151617 18191A1B 1C1D1E1F  00000000 00000000 00000000 00000000
0012:  20212223 24252627 28292A2B 2C2D2E2F  00000000 00000000 00000000 00000000
0013:  30313233 34353637 38393A3B 3C3D3E3F  00000000 00000000 00000000 00000000
0014:  40414243 44454647 48494A4B 4C4D4E4F  00000000 00000000 00000000 00000000
0015:  50515253 54555657 58595A5B 5C5D5E5F  00000000 00000000 00000000 00000000
0016:  60616263 64656667 68696A6B 6C6D6E6F  00000000 00000000 00000000 00000000
0017:  70717273 74757677 78797A7B 7C7D7E7F  00000000 00000000 00000000 00000000
0018:  80818283 84858687 88898A8B 8C8D8E8F  00000000 00000000 00000000 00000000
0019:  90919293 94959697 98999A9B 9C9D9E9F  00000000 00000000 00000000 00000000
001A:  A0A1A2A3 A4A5A6A7 A8A9AAAB ACADAEAF  00000000 00000000 00000000 00000000
001B:  B0B1B2B3 B4B5B6B7 B8B9BABB BCBDBEBF  00000000 00000000 00000000 00000000
001C:  C0C1C2C3 C4C5C6C7 C8C9CACB CCCDCECF  00000000 00000000 00000000 00000000
001D:  D0D1D2D3 D4D5D6D7 D8D9DADB DCDDDEDF  00000000 00000000 00000000 00000000
001E:  E0E1E2E3 E4E5E6E7 E8E9EAEB ECEDEEEF  00000000 00000000 00000000 00000000
001F:  F0F1F2F3 F4F5F6F7 F8F9FAFB FCFDFEFF  00000000 00000000 00000000 00000000
```

Figure 4-5 Displayed Data when select pattern b

## 4.6 Encrypt

Step to encrypt data as follows

a) Select “6. Encrypt” to encrypt plain data in memory.
b) Input parameter for IvIncrement.
c) When the encryption process is finished, both plain data and cipher data will be displayed in table-form as shown in Figure 4-6.



```
++++++ AES256XTSSTG Demo Menu ++++++
1. Set rEncEKeyIn and rDecEKeyIn
2. Set rEncTKeyIn and rDecTKeyIn
3. Set rEncIvIn and rDecIvIn
4. Show Data Memory
5. Fill Plain Data Memory
6. Encrypt Data
7. Fill Cipher Data Memory
8. Decrypt Data
Choice: 6

+++ Encrypt +++
Enable Iv Increment [0: Disable 1: Enable] --> 1

Length of Plain Data : 512 byte

                  Plain Data                                Cipher Data
Addr#   .0.....3 .4.....7 .8.....B .C.....F  .0.....3 .4.....7 .8.....B .C.....F
0000:   00010203 04050607 08090A0B 0C0D0E0F  64497E5A 831E4A93 2C09BE3E 5393376D
0001:   10111213 14151617 18191A1B 1C1D1E1F  AA599548 B816031D 224BBF50 A818ED23
0002:   20212223 24252627 28292A2B 2C2D2E2F  50EAE7E9 6087C8A0 DB51AD29 0BD00C1A
0003:   30313233 34353637 38393A3B 3C3D3E3F  C1620857 635BF246 C176AB46 3BE30B80
0004:   40414243 44454647 48494A4B 4C4D4E4F  8DA54808 1AC847B1 58E1264B E25BB091
0005:   50515253 54555657 58595A5B 5C5D5E5F  0BBC9264 71080894 15D45FAB 1B3D2604
0006:   60616263 64656667 68696A6B 6C6D6E6F  E8A8EFF1 AE4020CF A39936B6 6827B23F
0007:   70717273 74757677 78797A7B 7C7D7E7F  371B9220 0BE90251 E6D73C5F 86DE5FD4
0008:   80818283 84858687 88898A8B 8C8D8E8F  A9507819 33D79A28 272B782A 2EC313EF
0009:   90919293 94959697 98999A9B 9C9D9E9F  DFCC0628 F43D744C 2DC2FF3D CB66999B
000A:   A0A1A2A3 A4A5A6A7 A8A9AAAB ACADAEAF  50C7CA89 5B0C6479 1EEAA5F2 9499FB1C
000B:   B0B1B2B3 B4B5B6B7 B8B9BABB BCBDBEBF  026F84CE 5B5C72BA 1083CDDB 5CE45434
000C:   C0C1C2C3 C4C5C6C7 C8C9CACB CCCDCECF  631665C3 33B60B11 593FB253 C5179A2C
000D:   D0D1D2D3 D4D5D6D7 D8D9DADB DCDDDEDF  8DB81378 2A004856 A1653011 E93FB6D8
000E:   E0E1E2E3 E4E5E6E7 E8E9EAEB ECEDEEEF  76C18366 DD8683F5 3412C0C1 80F9C848
000F:   F0F1F2F3 F4F5F6F7 F8F9FAFB FCFDFEFF  592D593F 8609CA73 6317D356 E13E2BFF
0010:   00010203 04050607 08090A0B 0C0D0E0F  3A9F59CD 9AEB19CD 482593D8 C46128BB
0011:   10111213 14151617 18191A1B 1C1D1E1F  32423B37 A9ADFB48 2B99453F BE25A41B
0012:   20212223 24252627 28292A2B 2C2D2E2F  F6FEB4AA 0BEF5ED2 4BF73C76 29780254
0013:   30313233 34353637 38393A3B 3C3D3E3F  82C13115 E4015AAC 992E5613 A3B5C2F6
0014:   40414243 44454647 48494A4B 4C4D4E4F  85B84795 CB6E9B26 56D8C881 57E52C42
0015:   50515253 54555657 58595A5B 5C5D5E5F  F978D863 4C43D06F EA928F28 22E465AA
0016:   60616263 64656667 68696A6B 6C6D6E6F  6576E9BF 41938450 6CC3CE3C 54AC1A6F
0017:   70717273 74757677 78797A7B 7C7D7E7F  67DC66F3 B30191E6 98380BC9 99B05ABC
0018:   80818283 84858687 88898A8B 8C8D8E8F  E19DC0C6 DCC2DD00 1EC535BA 18DEB2DF
0019:   90919293 94959697 98999A9B 9C9D9E9F  1A101023 108318C7 5DC98611 A09DC48A
001A:   A0A1A2A3 A4A5A6A7 A8A9AAAB ACADAEAF  0ACDEC67 6FABDF22 2F07E026 F059B672
001B:   B0B1B2B3 B4B5B6B7 B8B9BABB BCBDBEBF  B56E5CBC 8E1D21BB D867DD92 72120546
001C:   C0C1C2C3 C4C5C6C7 C8C9CACB CCCDCECF  81D70EA7 37134CDF CE93B6F8 2AE22423
001D:   D0D1D2D3 D4D5D6D7 D8D9DADB DCDDDEDF  274E58A0 821CC550 2E2D0AB4 585E94DE
001E:   E0E1E2E3 E4E5E6E7 E8E9EAEB ECEDEEEF  6975BE5E 0B4EFCE5 1CD3E70C 25A1FBBB
001F:   F0F1F2F3 F4F5F6F7 F8F9FAFB FCFDFEFF  D609D273 AD5B0D59 631C531F 6A0A57B9
```

Figure 4-6 Nios II Command Shell after finished encryption process

## 4.7 Fill Cipher Data Memory

Step to fill Cipher data in memory as follows

a) Select “7. Fill Cipher Data Memory”.
b) Input the desired number of 512-byte data. User can press “enter” to use the default value of Cipher data. user can select data pattern.
c) There are four pattern to fill memory.
    a. zero pattern
    b. 8-bit counter
    c. 16-bit counter
    d. 32-bit counter
d) Whole cipher-data memory is filled with selected data pattern.

```
++++++ AES256XTSSTG Demo Menu ++++++
1. Set rEncEKeyIn and rDecEKeyIn
2. Set rEncTKeyIn and rDecTKeyIn
3. Set rEncIvIn and rDecIvIn
4. Show Data Memory
5. Fill Plain Data Memory
6. Encrypt Data
7. Fill Cipher Data Memory
8. Decrypt Data
Choice: 7

+++ Fill Cipher Data Memory +++
Number of 512-byte Data in decimal (enter = 5): 1
a. zero pattern
b. 8-bit counter
c. 16-bit counter
d. 32-bit counter
Choice: c

Length of Cipher Data : 512 byte

                Plain Data                                      Cipher Data
Addr#  .0.....3 .4.....7 .8.....B .C.....F  .0.....3 .4.....7 .8.....B .C.....F
0000:  00000000 00000000 00000000 00000000  00000001 00020003 00040005 00060007
0001:  00000000 00000000 00000000 00000000  00080009 000A000B 000C000D 000E000F
0002:  00000000 00000000 00000000 00000000  00100011 00120013 00140015 00160017
0003:  00000000 00000000 00000000 00000000  00180019 001A001B 001C001D 001E001F
0004:  00000000 00000000 00000000 00000000  00200021 00220023 00240025 00260027
0005:  00000000 00000000 00000000 00000000  00280029 002A002B 002C002D 002E002F
0006:  00000000 00000000 00000000 00000000  00300031 00320033 00340035 00360037
0007:  00000000 00000000 00000000 00000000  00380039 003A003B 003C003D 003E003F
0008:  00000000 00000000 00000000 00000000  00400041 00420043 00440045 00460047
0009:  00000000 00000000 00000000 00000000  00480049 004A004B 004C004D 004E004F
000A:  00000000 00000000 00000000 00000000  00500051 00520053 00540055 00560057
000B:  00000000 00000000 00000000 00000000  00580059 005A005B 005C005D 005E005F
000C:  00000000 00000000 00000000 00000000  00600061 00620063 00640065 00660067
000D:  00000000 00000000 00000000 00000000  00680069 006A006B 006C006D 006E006F
000E:  00000000 00000000 00000000 00000000  00700071 00720073 00740075 00760077
000F:  00000000 00000000 00000000 00000000  00780079 007A007B 007C007D 007E007F
0010:  00000000 00000000 00000000 00000000  00800081 00820083 00840085 00860087
0011:  00000000 00000000 00000000 00000000  00880089 008A008B 008C008D 008E008F
0012:  00000000 00000000 00000000 00000000  00900091 00920093 00940095 00960097
0013:  00000000 00000000 00000000 00000000  00980099 009A009B 009C009D 009E009F
0014:  00000000 00000000 00000000 00000000  00A000A1 00A200A3 00A400A5 00A600A7
0015:  00000000 00000000 00000000 00000000  00A800A9 00AA00AB 00AC00AD 00AE00AF
0016:  00000000 00000000 00000000 00000000  00B000B1 00B200B3 00B400B5 00B600B7
0017:  00000000 00000000 00000000 00000000  00B800B9 00BA00BB 00BC00BD 00BE00BF
0018:  00000000 00000000 00000000 00000000  00C000C1 00C200C3 00C400C5 00C600C7
0019:  00000000 00000000 00000000 00000000  00C800C9 00CA00CB 00CC00CD 00CE00CF
001A:  00000000 00000000 00000000 00000000  00D000D1 00D200D3 00D400D5 00D600D7
001B:  00000000 00000000 00000000 00000000  00D800D9 00DA00DB 00DC00DD 00DE00DF
001C:  00000000 00000000 00000000 00000000  00E000E1 00E200E3 00E400E5 00E600E7
001D:  00000000 00000000 00000000 00000000  00E800E9 00EA00EB 00EC00ED 00EE00EF
001E:  00000000 00000000 00000000 00000000  00F000F1 00F200F3 00F400F5 00F600F7
001F:  00000000 00000000 00000000 00000000  00F800F9 00FA00FB 00FC00FD 00FE00FF
```

Figure 4-7 Displayed Data when select pattern c

## 4.8 Decrypt

Step to decrypt data as follows

a) Select “8. Decrypt Data” to decrypt cipher data in memory.
b) Input parameter for IvIncrement.
c) When the decryption process is finished, both plain data and cipher data will be displayed in table-form as shown in Figure 4-8.

```
++++++ AES256XTSSTG Demo Menu ++++++
1. Set rEncEKeyIn and rDecEKeyIn
2. Set rEncTKeyIn and rDecTKeyIn
3. Set rEncIvIn and rDecIvIn
4. Show Data Memory
5. Fill Plain Data Memory
6. Encrypt Data
7. Fill Cipher Data Memory
8. Decrypt Data
Choice: 8

+++ Decrypt +++
Enable Iv Increment [0: Disable 1: Enable] --> 0

Length of Cipher Data : 512 byte

                Plain Data                          Cipher Data
Addr#  .0.....3 .4.....7 .8.....B .C.....F  .0.....3 .4.....7 .8.....B .C.....F
0000:  896240B2 4B3ED455 EC9E7CC9 36361846  00000001 00020003 00040005 00060007
0001:  88753DD4 95AF78BE 0CF477F6 F2CA724D  00080009 000A000B 000C000D 000E000F
0002:  63E00AFB 129F8A10 D5901721 91F6C5C3  00100011 00120013 00140015 00160017
0003:  AC04B362 0C286815 5B6DA9AB C4A424C1  00180019 001A001B 001C001D 001E001F
0004:  9BD0405A 2DEFBE2F 566D6E41 1558F092  00200021 00220023 00240025 00260027
0005:  010D1FD6 6999136B D58C6899 49CC7FF9  00280029 002A002B 002C002D 002E002F
0006:  5C64E06B A37C84D5 6DF48CC4 5D8B316F  00300031 00320033 00340035 00360037
0007:  798ED0BC 441A7F1F 2F0E966A 2BABEE30  00380039 003A003B 003C003D 003E003F
0008:  65347823 EA972526 C1A75F00 FCB8797B  00400041 00420043 00440045 00460047
0009:  211501C8 5E7F4B90 ADCA613E E3D40419  00480049 004A004B 004C004D 004E004F
000A:  0F6C99AC F7EA1B3F 90B5F801 700F0393  00500051 00520053 00540055 00560057
000B:  0F0D2A91 7B71111F A54F78B4 116BB341  00580059 005A005B 005C005D 005E005F
000C:  2804E1EE 219FBDBF 00560D7E E98088F8  00600061 00620063 00640065 00660067
000D:  B885F1F9 45CF8949 0ECDBC13 92E82984  00680069 006A006B 006C006D 006E006F
000E:  74D0D621 EA73F168 BB278906 3C584A71  00700071 00720073 00740075 00760077
000F:  891042B4 4D29F0A7 FD498E61 E3F2C5FC  00780079 007A007B 007C007D 007E007F
0010:  B556D7CE 6DEE0869 4BAEE093 EDB25A62  00800081 00820083 00840085 00860087
0011:  FDCB7BF4 CF302454 E13D24A3 2A16DF1D  00880089 008A008B 008C008D 008E008F
0012:  A4FA4E60 151EFDAD F2765F5A 2101F97C  00900091 00920093 00940095 00960097
0013:  33EE730B 22BC9724 1A47B996 A6A19DF1  00980099 009A009B 009C009D 009E009F
0014:  04EB5F8B FECB129F B939EB86 745E7E30  00A000A1 00A200A3 00A400A5 00A600A7
0015:  4AB57E3D 255A11C3 AB94AC9A 6DAD5FCF  00A800A9 00AA00AB 00AC00AD 00AE00AF
0016:  D416A303 22AD254D 2E6689C3 9EC36671  00B000B1 00B200B3 00B400B5 00B600B7
0017:  9FADED26 EFA35260 6C7AF5BA E3B564A5  00B800B9 00BA00BB 00BC00BD 00BE00BF
0018:  7C152441 E41B9FD8 3035B207 2343FF94  00C000C1 00C200C3 00C400C5 00C600C7
0019:  A6EB90C3 EB280476 FBB58D55 2AFE0EA3  00C800C9 00CA00CB 00CC00CD 00CE00CF
001A:  D6250ADD 0359F861 085E2F0F 1852FBD0  00D000D1 00D200D3 00D400D5 00D600D7
001B:  B88DFBF5 6CF25A51 7AB82300 20106A74  00D800D9 00DA00DB 00DC00DD 00DE00DF
001C:  3610BA5B 18C589C6 810EC8CF 25150308  00E000E1 00E200E3 00E400E5 00E600E7
001D:  487A0098 7687F0CD 02287361 FEA99033  00E800E9 00EA00EB 00EC00ED 00EE00EF
001E:  470B9D26 700398D7 CE741538 B9B9C665  00F000F1 00F200F3 00F400F5 00F600F7
001F:  58DE2065 6C8E7C51 191EC9DB D79D0A07  00F800F9 00FA00FB 00FC00FD 00FE00FF
```

Figure 4-8 Nios II Command Shell after finished decryption process

# 5 Revision History

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