Technical Specification (TS)

IP AES 256

Table of distribution

Elsys Design	Name – function	Contact
	Killian Michaud – SoC Design/Verif Engineer	killian.michaud@elsys-design.com +33 6 29 47 54 13

Revision History

Date	Version	Description	Author
24/10/2024	0.1	Document creation	Killian MICHAUD

Table of content

Table of figures

Scope

This document aims at providing the requirement specifications for the IP AES 256.

The **Advanced Encryption Standard (AES)** with a 256-bit key length, commonly known as **AES-256**, is a symmetric encryption algorithm used to secure data. It operates on a block size of 128 bits and uses a series of transformations, including substitution, permutation, and mixing, to encrypt and decrypt data. The process involves multiple rounds (14 for AES-256) of these transformations, making it highly secure against brute-force attacks. AES-256 is widely used in various applications, including securing sensitive data in financial transactions, communications, and data storage.

A. Mentioned documents

1. Reference documents

Index	Title	Reference	Date
[R1]	ADVANCED ENCRYPTION STANDARD (AES)	nist.fips.197.pdf	26/11/2001

B. Acronyms and abbreviations

Acronyms and abbreviations	Meaning
FPGA	Field Programmable Gate Array
AXI	Advanced eXtensible Interface
IP	Intellectual Property
FIFO	First In First Out
AES	Advanced Encryption Standard

C. Module Description

1. Interfaces block diagram

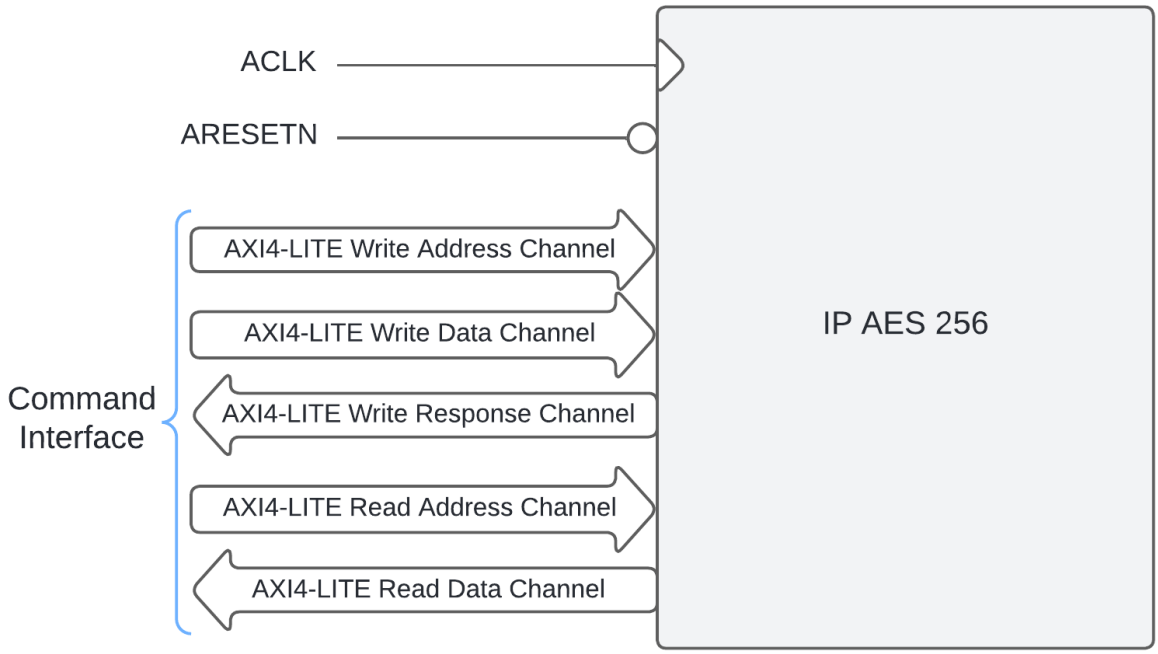


Figure 1 : Block Diagram IP AES 256

a. Interfaces description

Table 1 : IP AES 256 interfaces signals

Signal Name	Length	Type	Signal description
			Clock Reset
AXI4_Lite_Clk	1	INPUT	AXI Lite Clock
AXI4_Lite_Rst_n	1	INPUT	Asynchronous global reset active low.
			AXI4 LITE WRITE ADDRESS CHANNEL
Awaddr	32	INPUT	Write address. The write address bus gives the address of the transaction.
Awvalid	1	INPUT	Write address valid. This signal indicates that the valid write address and control information are available.

Awready	1	OUTPUT	Write address ready. This signal indicates that the slave is ready to accept address and associated control signals.
AXI4 LITE WRITE DATA CHANNEL			
Wdata	32	INPUT	Write data. It's a 32 bits bus.
Wvalid	1	INPUT	Write valid. This signal indicates that write data and strobes are available.
Wready	1	OUTPUT	Write ready. This signal indicates that the slave can accept the write data.
AXI4 LITE WRITE RESPONSE CHANNEL			
Bresp	2	OUTPUT	Write response. This signal indicates the status of the write transaction. The allowable responses are OKAY, EXOKAY, SLVERR and DECERR.
Bvalid	1	OUTPUT	Write response valid. This signal indicates that a valid write response is available.
Bready	1	INPUT	Response ready. This signal indicates that the master can accept the response information.
AXI4 LITE READ ADDRESS CHANNEL			
Araddr	32	INPUT	Read address. The read address bus gives the address of the transaction.
Arvalid	1	INPUT	Read address valid. This signal indicates that the valid read address and control information are available.
Arready	1	OUTPUT	Read address ready. This signal indicates that the slave is ready to accept address and associated control signals.
AXI4 LITE READ DATA CHANNEL			
Rdata	32	OUTPUT	Read data. It's a 32 bit bus.
Rresp	2	OUTPUT	Read response. This signal indicates the status of the read transfer. The allowable responses are OKAY, EXOKAY, SLVERR and DECERR.
Rvalid	1	OUTPUT	Read valid. This signal indicates that the required read data is available and the transfer can complete.
Rready	1	INPUT	Read ready. This signal indicates that the master can accept the read data and response information.

2. Functional description

a. CLOCK and resets

SYS_CLK: system clock. This clock is used for the AXI Lite interface.

RESET_N: synchronous reset of the system clock. This input is subsequently used as an asynchronous reset for the various modules. Reset active in low state.

b. Encryption Algorithm

```
Cipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
begin
   byte  state[4,Nb]

   state = in

   AddRoundKey(state, w[0, Nb-1])                  // See Sec. 5.1.4

   for round = 1 step 1 to Nr-1
      SubBytes(state)                              // See Sec. 5.1.1
      ShiftRows(state)                             // See Sec. 5.1.2
      MixColumns(state)                            // See Sec. 5.1.3
      AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
   end for

   SubBytes(state)
   ShiftRows(state)
   AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])

   out = state
end
```

Figure 2 : Pseudo-code for Encryption Algorithm from document [R1]

In the Encryption Algorithm, the first step is to create all the 15 keys required to complete all the functions. This step is done with a new Algorithm, named Key Expansion (Figure 3).

In AES-256, 15 rounds is needed to complete Encryption Algorithm. The first round is a XOR operation between the first key, create with Key Expansion, and plain text.

The round 2 to 14, there a succession of methods named :

- SubBytes
- ShiftRows
- MixColumns
- AddRoundKeys

Finally the last round is a succession of methods, like the previous rounds but the MixColumns method isn't include in this round.

```
KeyExpansion(byte key[4*Nk], word w[Nb*(Nr+1)], Nk)
begin
   word  temp

   i = 0

   while (i < Nk)
      w[i] = word(key[4*i], key[4*i+1], key[4*i+2], key[4*i+3])
      i = i+1
   end while

   i = Nk

   while (i < Nb * (Nr+1)]
      temp = w[i-1]
      if (i mod Nk = 0)
         temp = SubWord(RotWord(temp)) xor Rcon[i/Nk]
      else if (Nk > 6 and i mod Nk = 4)
         temp = SubWord(temp)
      end if
      w[i] = w[i-Nk] xor temp
      i = i + 1
   end while
end

Note that Nk=4, 6, and 8 do not all have to be implemented;
they are all included in the conditional statement above for
conciseness.   Specific implementation requirements for the
Cipher Key are presented in Sec. 6.1.
```

Figure 3 : Pseudo-code for Key Expansion Algorithm from document [R1]

c. Decryption Algorithm

```
InvCipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
begin
   byte  state[4,Nb]

   state = in

   AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1]) // See Sec. 5.1.4

   for round = Nr-1 step -1 downto 1
      InvShiftRows(state)                    // See Sec. 5.3.1
      InvSubBytes(state)                     // See Sec. 5.3.2
      AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
      InvMixColumns(state)                   // See Sec. 5.3.3
   end for

   InvShiftRows(state)
   InvSubBytes(state)
   AddRoundKey(state, w[0, Nb-1])

   out = state
end
```

Figure 4 : Pseudo-code for Decryption Algorithm from document [R1]

In the Decryption Algorithm, the first step is to create all the 15 keys required to complete all the functions, the Key Expansion Algorithm is also use.

In AES-256, 15 rounds is needed to complete Encryption Algorithm. The first round is a XOR operation between the last key, create with Key Expansion, and cipher text.

The round 2 to 14, there a succession of methods named :

- InvShiftRows
- InvSubBytes
- AddRoundKeys
- InvMixColumns

Finally the last round is a succession of methods, like the previous rounds but the InvMixColumns method isn't include in this round.

D.Functional specification

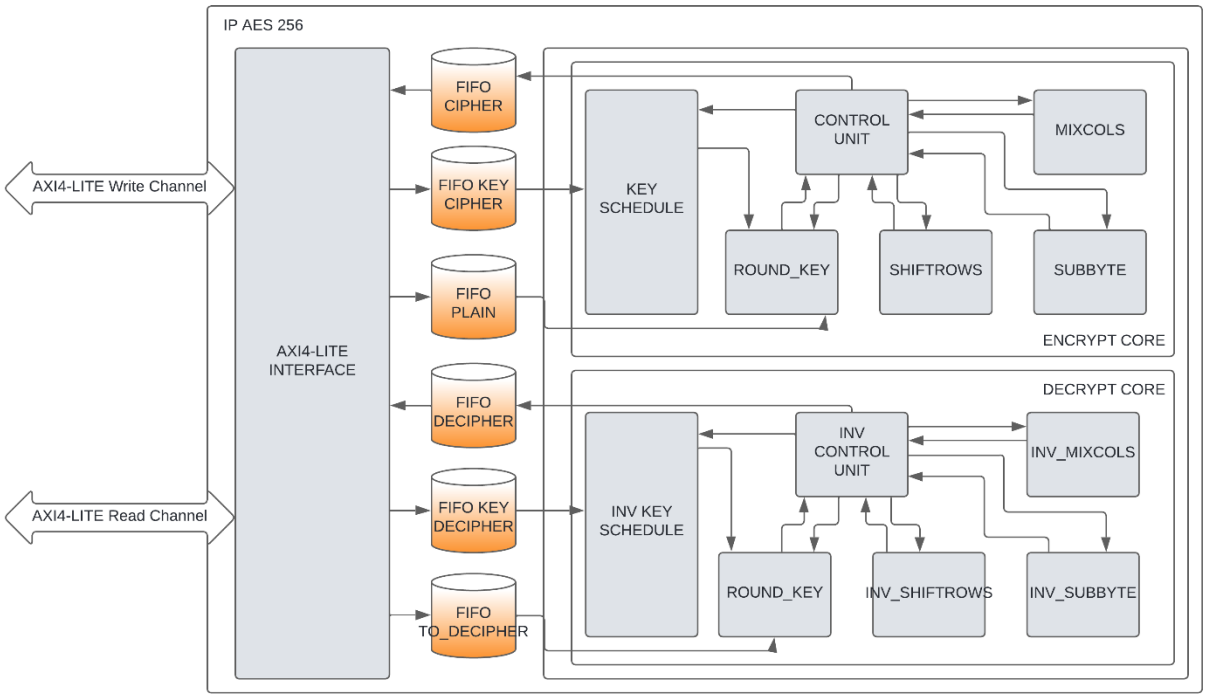


Figure 5 : Functional Block Diagram

This IP AES 256 is composed of 9 modules :

- AXI4-Lite Interface
- FIFO CIPHER TEXT
- FIFO CIPHER KEY
- FIFO PLAIN TEXT
- FIFO DECIPHER TEXT
- FIFO DECIPHER KEY
- FIFO TO_DECIPHER TEXT
- ENCRYPT CORE
- DECRYPT CORE

1. Features

a. AXI4-Lite Interface

- Full forward and reverse direction flow control of AXI protocol-defined READY/VALID handshake.
- AXI4 memory mapped burst lengths of:
  - 1 to 256 bits for incrementing burst and
  - 1 to 16 bits for wrap burst
- For AXI4-Lite, the supported data width is 32 bits only.
- The use of Read/Write, Read-only, or Write-only interfaces.
- No support for locked transfers.
- The use of user bits is discouraged in general purpose due to interoperability concerns.
- Holding AXI ARESETN asserted for 16 cycles of the AXI clock is generally a sufficient reset pulse width.

b. FIFO CIPHER TEXT

- First input data, first output data
- 128bits length input data
- 32bits length output data

c. FIFO CIPHER KEY

- First input data, first output data
- 32bits length input data
- 256bits length output data

d. FIFO PLAIN TEXT

- First input data, first output data
- 32bits length input data
- 128bits length output data

e. FIFO DECIPHER TEXT

- First input data, first output data
- 128bits length input data

- 32bits length output data

f. FIFO DECIPHER KEY

- First input data, first output data
- 32bits length input data
- 256bits length output data

g. FIFO TO_DECIPHER TEXT

- First input data, first output data
- 32bits length input data
- 128bits length output data

h. ENCRYPT CORE

- Take a 256bits Cipher key and 128bits Plain text
- A start bit is set to enable the encrypt algorithm :
    - Follow the steps of the AES specification document :
        - Key Expansion
        - Add Round Keys
        - Sub byte
        - Shift rows
        - Mix columns
- A Control unit is there to insure the proper operation of the algorithm
- 128bits Cipher text output
- 1bit done status algorithm

i. DECRYPT CORE

- Take a 256bits Decipher Key and 128bits Cipher text
- A start bit is set to enable the decrypt algorithm :
    - Follow the steps of the AES specification document :
        - Key Expansion
        - Add Round Keys
        - Inverse Shift Rows
        - Inverse Sub byte
        - Inverse Mix columns

- A Control unit is there to insure the proper operation of the algorithm
- 128bits Plain text output
- 1bit done status algorithm

2. Mapping

Table 2 : Memory mapping IP AES 256

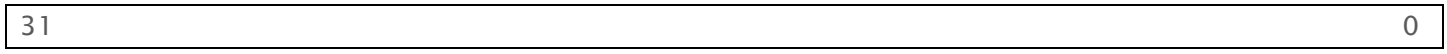
Address Offset *	Name	Access	Description
00h	STATUS	RO	Status register
04h	CONTROL	R/W	Control register
08h	VERSION	RO	Version register
0Ch	SCRPAD	R/W	Scratchpad register
10h	KEY_CIPHER	WO	Cipher key register
14h	KEY_DECIPHER	WO	Decipher key register
18h	PLAIN_WORD	WO	Plain text register
1Ch	TO_DECIPHER_WORD	WO	To decipher text register
20h	CIPHER_WORD	RO	Cipher text register
24h	DECIPHER_WORD	RO	Decipher text register

Notes: * Address offset is relative to IP AES 256 base address

3. Registers

STATUS (Status register Address – Offset 00h)

This register provides the status signals of the IP AES 256.



STATUS

Figure 6 : STATUS

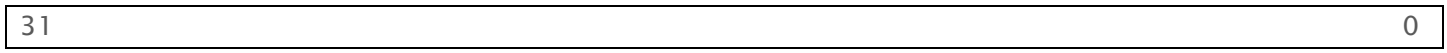
Table 3 : STATUS details

Bits	Fields Name	Default Value	Access Type	Description
0	Done_cipher	0	RO	Status of the Encrypt Algorithm
1	Done_decipher	0	RO	Status of the Decrypt Algorithm
2	Cipher_text_empty	0	RO	Flag FIFO TO_DECIPHER TEXT empty
3	Cipher_text_full	0	RO	Flag FIFO TO_DECIPHER TEXT full

4	Plain_text_empty	0	RO	Flag FIFO PLAIN TEXT empty
5	Plain_text_full	0	RO	Flag FIFO PLAIN TEXT full
6	Cipher_key_empty	0	RO	Flag FIFO CIPHER KEY empty
7	Cipher_key_full	0	RO	Flag FIFO CIPHER KEY full
8	Decipher_key_empty	0	RO	Flag FIFO DECIPHER KEY empty
9	Decipher_key_full	0	RO	Flag FIFO DECIPHER KEY full
10	Decipher_text_empty	0	RO	Flag FIFO DECIPHER TEXT empty
11	Decipher_text_full	0	RO	Flag FIFO DECIPHER TEXT full
12	Enc_text_empty	0	RO	Flag FIFO CIPHER TEXT empty
13	Enc_text_full	0	RO	Flag FIFO CIPHER TEXT full

CONTROL (Control register Address – Offset 04h)

This register provides the control signals of the IP AES 256.



CONTROL

Figure 7 : CONTROL

Table 4 : CONTROL details

Bits	Fields Name	Default Value	Access Type	Description
0	Start_cipher	0	RW	Enable a encrypt algorithm
1	Start_decipher	0	RW	Enable a encrypt algorithm
2	Flush_cipher	0	RW	Reset to zero FIFO TO_DECIPHER TEXT
3	Flush_plain	0	RW	Reset to zero FIFO PLAIN TEXT
4	Flush_cipher_key	0	RW	Reset to zero FIFO CIPHER KEY
5	Flush_decipher_key	0	RW	Reset to zero FIFO DECIPHER KEY
6	Flush_enc	0	RW	Reset to zero FIFO CIPHER TEXT
7	Flush_dec	0	RW	Reset to zero FIFO DECIPHER TEXT
31 to 8	Reserved	0	RW	N/A

VERSION (Version register Address – Offset 08h)

This register provides the version of the IP AES 256.

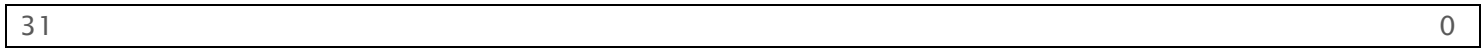


Figure 8 : VERSION

Table 5 : VERSION details

Bits	Fields Name	Default Value	Access Type	Description
31 to 0	Version	0	RO	Number of version of the IP

SCRATCHPAD (Scratchpad register Address – Offset 0Ch)

This register provides a scratchpad area for the IP AES 256.

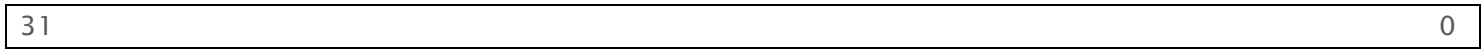


Figure 9 : SCRPAD

Table 6 : SCRPAD details

Bits	Fields Name	Default Value	Access Type	Description
31 to 0	SCRPAD	0	R/W	Scratchpad data

KEY_CIPHER (KEY_CIPHER register Address – Offset 10h)

This register provides the cipher key to the FIFO CIPHER KEY of the IP AES 256.

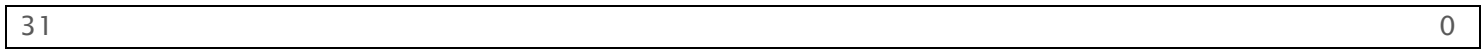


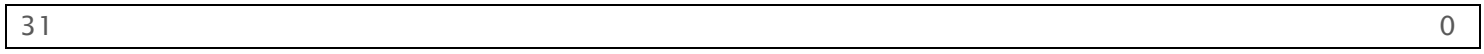
Figure 10 : KEY_CIPHER

Table 7 : KEY_CIPHER details

Bits	Fields Name	Default Value	Access Type	Description
31 to 0	KEY_CIPHER	0	WO	Data transmitted to the ENCRYPT CORE module

KEY_DECIPHER (KEY_DECIPHER register Address – Offset 14h)

This register provides the cipher key to the FIFO CIPHER KEY of the IP AES 256.



KEY_DECIPHER

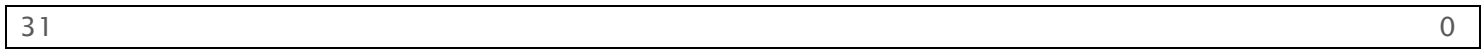
Figure 11 : KEY_DECIPHER

Table 8 : KEY_DECIPHER details

Bits	Fields Name	Default Value	Access Type	Description
31 to 0	KEY_DECIPHER	0	WO	Data transmitted to the DECRYPT CORE module

PLAIN_WORD (Plain Text register Address – Offset 18h)

This register provides the cipher key to the FIFO CIPHER KEY of the IP AES 256.



PLAIN_WORD

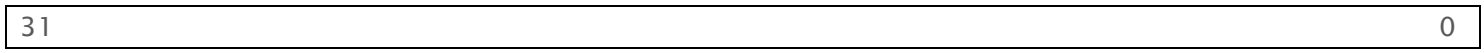
Figure 12 : PLAIN_WORD

Table 9 : PLAIN_WORD details

Bits	Fields Name	Default Value	Access Type	Description
31 to 0	PLAIN_WORD	0	WO	Data transmitted to the ENCRYPT CORE module

TO_DECIPHER_WORD (To decipher text register Address – Offset 1Ch)

This register provides the cipher key to the FIFO CIPHER KEY of the IP AES 256.



TO_DECIPHER_WORD

Figure 13 : TO_DECIPHER_WORD

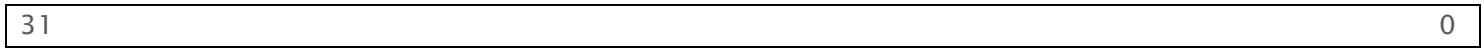
Table 10 : TO_DECIPHER_WORD details

Bits	Fields Name	Default Value	Access Type	Description

31 to 0	TO_DECIPHER_WORD	0	WO	Data transmitted to the DECRYPT CORE module

CIPHER_WORD (Cipher text register Address – Offset 20h)

This register provides the cipher key to the FIFO CIPHER KEY of the IP AES 256.



CIPHER_WORD

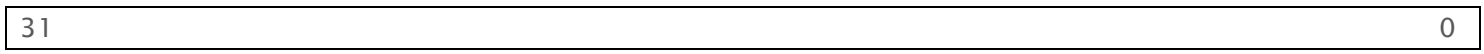
Figure 14 : CIPHER_WORD

Table 11 : CIPHER_WORD details

Bits	Fields Name	Default Value	Access Type	Description
31 to 0	CIPHER_WORD	0	RO	Data from the ENCRYPT CORE module

DECIPHER_WORD (Decipher text register Address – Offset 24h)

This register provides the cipher key to the FIFO CIPHER KEY of the IP AES 256.



DECIPHER_WORD

Figure 15 : DECIPHER_WORD

Table 12 : DECIPHER_WORD details

Bits	Fields Name	Default Value	Access Type	Description
31 to 0	DECIPHER_WORD	0	RO	Data from the DECRYPT CORE module

4. Timings

a. Encrypt text timing

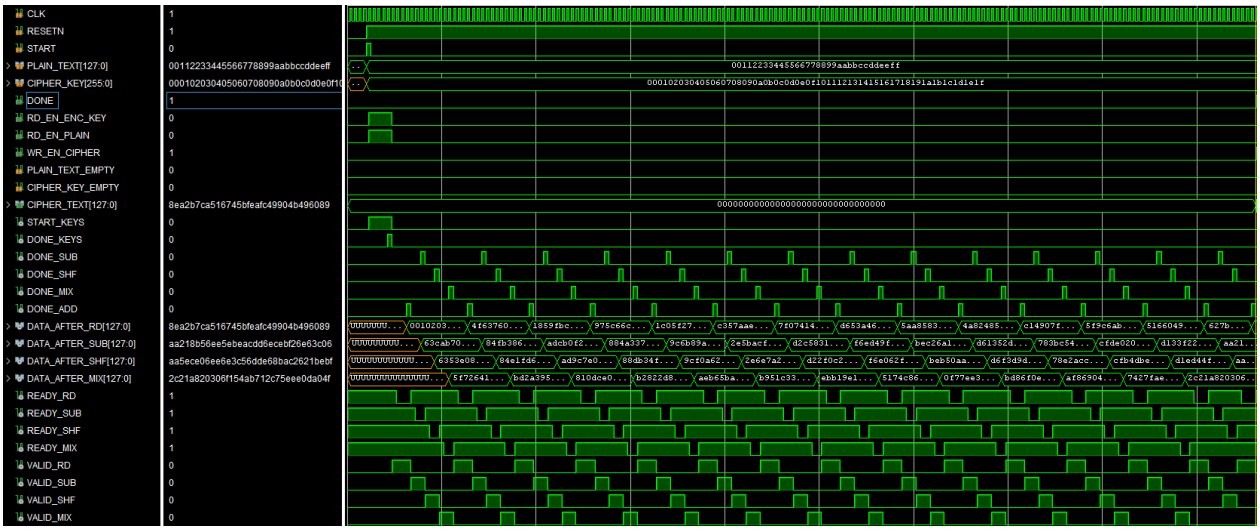


Figure 16 : Encrypt text timing signals

b. Decrypt text timing

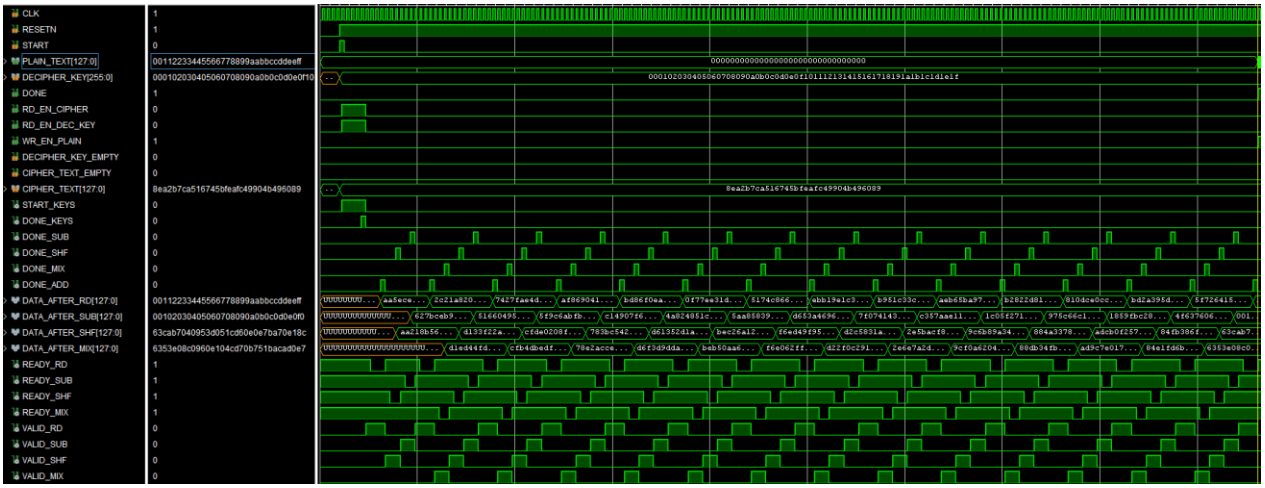


Figure 17 : Decrypt text timing signals