Using MachXO3D ESB to Implement AES128/AES256 Encryption and Decryption

Reference Design

FPGA-RD-02056-0.90

May 2019

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS and with all faults, and all risk associated with such information is entirely with Buyer. Buyer shall not rely on any data and performance specifications or parameters provided herein. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. No Lattice products should be used in conjunction with mission- or safety-critical or any other application in which the failure of Lattice's product could create a situation where personal injury, death, severe property or environmental damage may occur. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

© 2019 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Contents

Figures

Tables

© 2019 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
AES	Advanced Encryption Standard
CBC	Cipher Block Chaining
CFB	Cipher Feedback
CTR	Counter
ECB	Electronic Code Book
ESB	Embedded Security Block
FIPS	Federal Information Processing Standards
HSP	High Speed Port
NIST	National Institute of Standards and Technology
OFB	Output Feedback
OSC	Oscillator

© 2019 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

1. Introduction

Lattice Semiconductor provides this Advanced Encryption Standard (AES) reference design in order to demonstrate how to use MachOX3D™ Embedded Security Block (ESB) to implement AES128 or AES256 encryption or decryption.

Advanced Encryption Standard (AES) is an encryption standard based on symmetric key algorithm, using the same key for encryption and decryption, issued by NIST in 2001. AES is one of the most widely used encryption and decryption protocols to transmit and receive data securely. The strength of AES depends on the secret key size. In the MachXO3D family, there is an option to select from two secret key sizes, 128 bits or 256 bits. The secret key size determines the execution time of the algorithm to encrypt or decrypt a particular data stream.

AES engine in the Embedded Security Block (ESB) performs AES128 or AES256 operation per FIPS197. Electronic Code Book (ECB) mode is supported with both WISHBONE interface mode and High Speed Port (HSP) interface mode. To implement the Cipher Block Chaining (CBC), Cipher Feedback (CFB), Counter (CTR), or Output Feedback (OFB) mode, you need to build block cipher mode of operation in fabric together with the ESB. These modes are only available with the WISHBONE interface mode of the ESB. Figure 1.1 shows an implementation on CFB encryption.

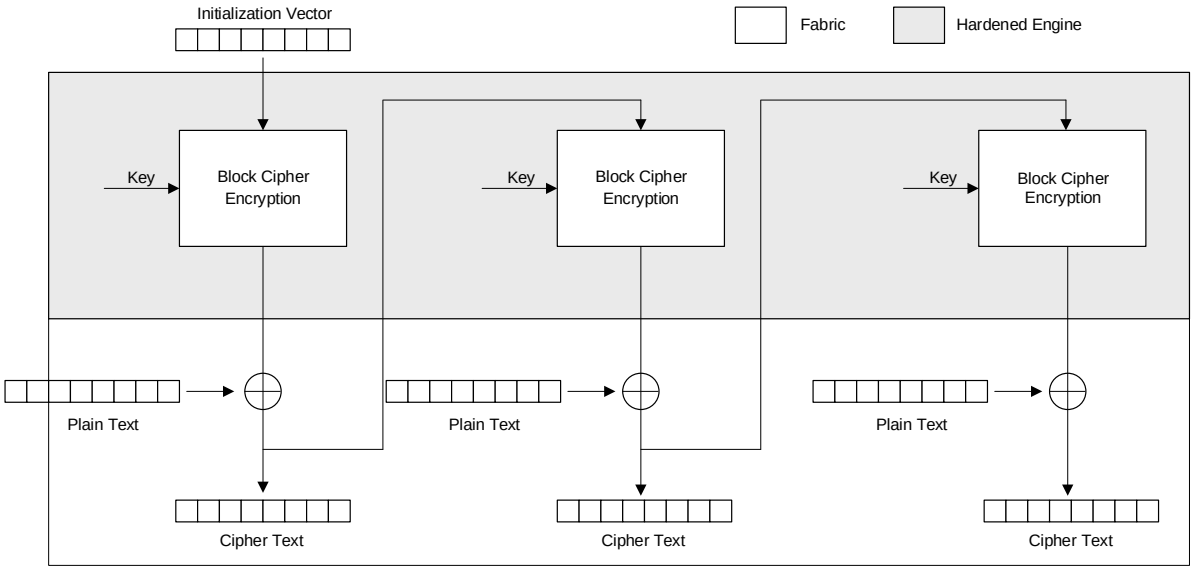

Figure 1.1.Implementation of the AES CFB Mode

In this AES reference design, only ECB mode is implemented as an example on how to use the ESB engine.

Figure 1.2 shows the high-level overview of the AES function implemented in the ESB of the MachXO3D device. It provides the flexibility of choosing the input data source of WISHBONE interface or the high speed port (HSP). This allows you to speed up the execution of the AES function to get data into and out of the ESB. With appropriate configuration of ESB, you can use the AES encryption/decryption function without knowing the AES algorithm in detail.

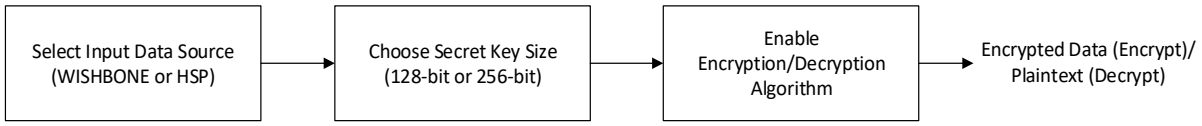

Figure 1.2. High-level Overview of AES Function in ESB

© 2019 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

2. Reference Design Overview

This AES reference design shows you how to enable the AES algorithm, and generate encrypted data through AES encryption, and decrypt the encrypted data through AES decryption.

2.1. Block Diagram

Figure 2.1 shows the block diagram of the AES reference design.

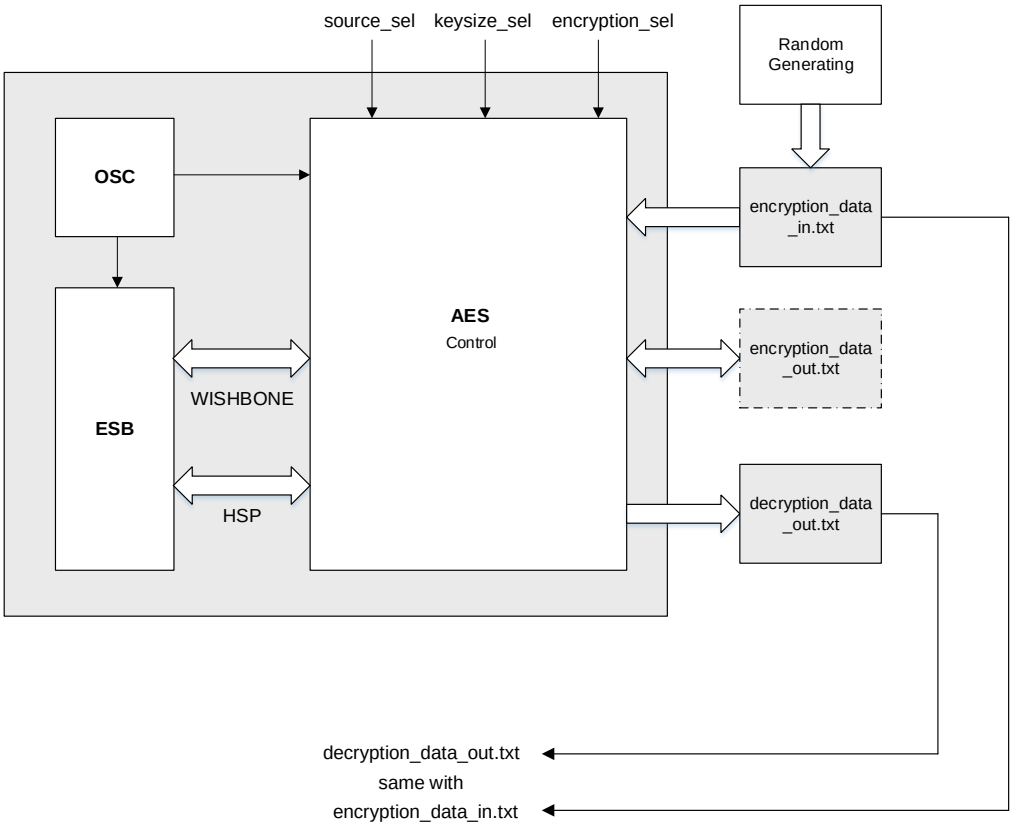

Figure 2.1. Top-level Block Diagram

2.2. Overview

This AES reference design consists of three major blocks:

- ESB
- AES Control
- Oscillator (OSC)

ESB is the Embedded Security Block. AES encryption and decryption is executed in the ESB. The user AES key and encryption/decryption input data are sent to the ESB. The encryption/decryption result is fetched from ESB.

OSC is used for both the ESB clock and the system clock of the AES control module.

AES control is the master of the WISHBONE bus. It configures control registers of the ESB and monitors its status register.

This AES reference design provides AES usages, which include:

- AES encryption and decryption
- Data path support with WISHBONE or HSP
- Key size selection of AES128 and AES256

© 2019 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

3. Functional Description

In this AES reference design, the following controls are available:

- Data path
- AES key length
- Encryption or decryption selection

The AES key is always sent to the ESB through the WISHBONE interface. The plain text file and the encrypted text file can be sent to the ESB either through WISHBONE or the HSP. The reference design selects WISHBONE or HSP according to user input.

AES128 and AES256 have different AES key lengths. You can select AES128 or AES256 through user input. If you select AES128, the AES reference design requires you to provide the 128-bit AES key. If you select AES256, the AES reference design requires you to provide 256-bit AES key.

Both encryption and decryption are supported. You can select either function through user input.

Table 3.1 provides the AES reference design parameter usage.

Table 3.1. AES Reference Design Parameter

Parameter Name	Value	Function
source_sel	0	High Speed Port (HSP)
	1	WISHBONE
encryption_sel	0	Encryption
	1	Decryption
keysize_sel	0	128-bit AES key
	1	256-bit AES key

© 2019 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

4. Design Description

This AES reference design includes two major parts. One is the ESB configuration, including ESB status registers check and control registers setting. The other is the input and output data flow control.

After reset is de-asserted, you need to assert the Start signal to trigger the AES encryption/decryption session. When the get_data_ready signal is asserted, you can provide the AES key through din_valid/din bus. The AES key is provided through the din_valid/din bus. Each word is 4 bytes and you can write one word at each clock cycle. Therefore, it takes four clock cycles to write AES128 and eight cycles to write AES256. For AES128, the sequence is [127:96], [95:64], [63:32], [31:0]. And for AES256, the sequence is [255:224], [223:192], …[31:0].

After providing the AES key, in WISHBONE interface mode, wait for the next get_data_ready signal to be active to provide input data. Make sure to assert one 4-byte word of data at each clock cycle. For the 128-bit data, the sequence is [127:96], [95:64], [63:32], [31:0]. When dout_valid signal is active, you can get encryption/decryption results from the dout bus. They are provided as 4-byte word data at each clock cycle. For the 128-bit data, the sequence is [127:96], [95:64], [63:32], [31:0].

If HSP mode is used, the reference design still uses get_data_ready signal as the handshake for user-input AES Key, while using HSP_full_o and HSP_wr_i for user-input data. HSP FIFO is not the full data you can input through the pipeline.

To start a new session of AES encryption/decryption, set the new control with the three control pins, source_sel, encryption_sel, and keysize_sel, and provide one-clock cycle Active High to the reset signal to begin with.

4.1. Detailed Input/Output of the Design

Figure 4.1 shows the I/O diagram of the AES reference design.

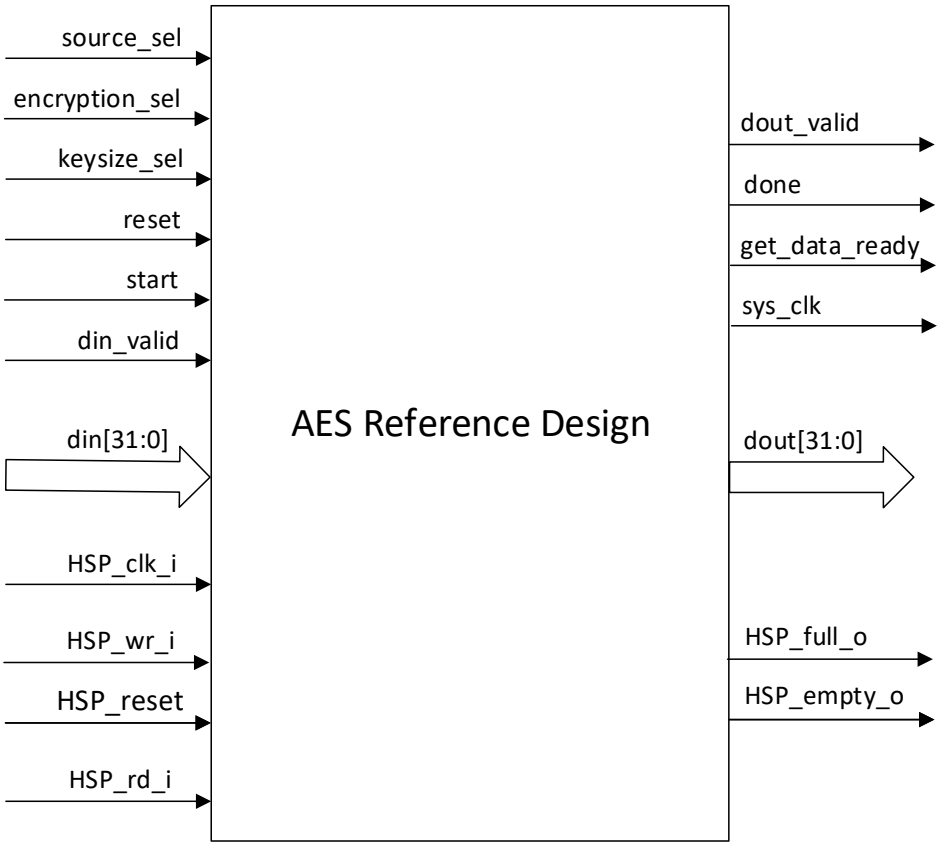

Figure 4.1. I/O Diagram for AES Reference Design

© 2019 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

4.2. Pin/Port Description of the Design

Table 4.1 lists the I/O ports of the AES reference design.

Table 4.1. Pin Descriptions

Signal	Width	Type	Description
Function Definition			
source_sel	1	Input	Input source select
encryption_sel	1	Input	AES function select
keysize_sel	1	Input	AES key size select
HSP Interface			
HSP_reset	1	Output	HSP reset
HSP_clk_i	1	Input	HSP clock
HSP_full_o	1	Output	HSP FIFO full
HSP_wr_i	1	Input	HSP write enable
HSP_empty_o	1	Output	HSP FIFO empty
HSP_rd_i	1	Input	HSP read enable
Common			
reset	1	Input	Global reset
start	1	Input	Start work
done	1	Output	Encryption done
sys_clk	1	Output	System clock
din_valid	1	Input	Input data valid
din	32	Input	Input data/AES Key
dout_valid	1	Output	Output data valid
dout	32	Output	Output data
get_data_ready	1	Output	User can input new data to WISHBONE.

© 2019 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

5. AES Configuration

5.1. ESB Registers for AES Configuration

Table 5.1 lists the registers of the AES reference design.

Table 5.1. Register Descriptions

Register Type	Register Name	Address	Read/Write	Purpose
Control Register	r0_gp0	18'h2_0020	Read	Check for busy status of ESB 0xB0: READY to get a new command 0xB2: ESB operation done
	as_src_sel	18'h2_003c	Write	Source select register [2]: 1: HSP 0: WISHBONE
	ri_ctrl1	18'h2_000c	Write	Enable ESB function: 0x0E: Enable AES 0x00: Disable AES
	ri_ctrl4	18'h2_0018	Write	AES key size selection: [0]: 0: 128 bits 1: 256 bits
AES Register	Aes_key_0	18'h2_2000	Write	AES key [255:224]; only for 256-bit key
	Aes_key_1	18'h2_2004	Write	AES key [223:192]; only for 256-bit key
	Aes_key_2	18'h2_2008	Write	AES key [191:160]; only for 256-bit key
	Aes_key_3	18'h2_200C	Write	AES key [159:128]; only for 256-bit key
	Aes_key_4	18'h2_2010	Write	AES key [127:96]
	Aes_key_5	18'h2_2014	Write	AES key [95:64]
	Aes_key_6	18'h2_2018	Write	AES key [63:32]
	Aes_key_7	18'h2_201C	Write	AES key [31:0]
	aes_in_0	18'h2_2020	Write	AES input data [127:96]
	aes_in_1	18'h2_2024	Write	AES input data [95:64]
	aes_in_2	18'h2_2028	Write	AES input data [63:32]
	aes_in_3	18'h2_202C	Write	AES input data [31:0]
	aes_res_0	18'h2_2030	Read	AES result data [127:96]
	aes_res_1	18'h2_2034	Read	AES result data [95:64]
	aes_res_2	18'h2_2038	Read	AES result data [63:32]
	aes_res_3	18'h2_203C	Read	AES result data [31:0]
	aes_con	18'h2_2040	Write	AES control register for encryption/decryption: [0]: 0: Encryption; 1: Decryption
	aes_status	18'h2_2044	Read	AES status register: [0]: Set when key expansion is done. Cleared when key expansion starts. [1]: Set when AES operation is done. Cleared at the start of every operation.

© 2019 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

5.2. AES Configuration Flowchart

Figure 5.1 shows the configuration flow for enabling the AES function.

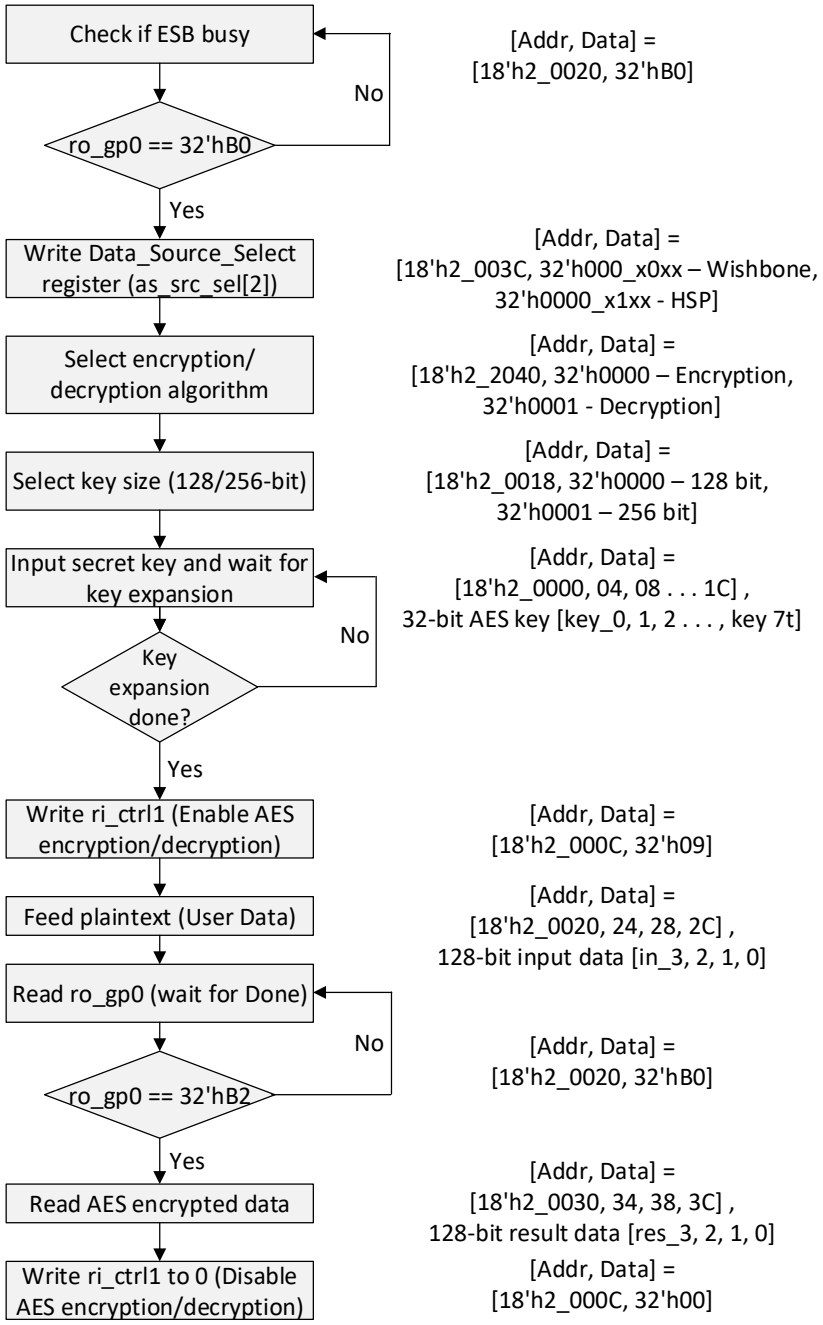

Figure 5.1. Flow Chart of AES Algorithm

© 2019 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

6. Simulation and Verification

This section explains some examples of WISHBONE and HSP simulation and verification results.

As shown in Figure 6.1 and Figure 6.2, when the get_data_ready signal goes high the first time, you can send the 128-bit or the 256-bit AES key. After sending the AES key, you can send the 128-bit plaintext when the get_data_ready signal goes high again.

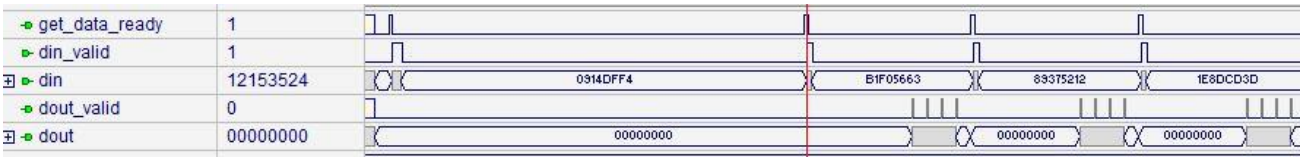

Figure 6.1. Waveform of WISHBONE Data Flow Simulation

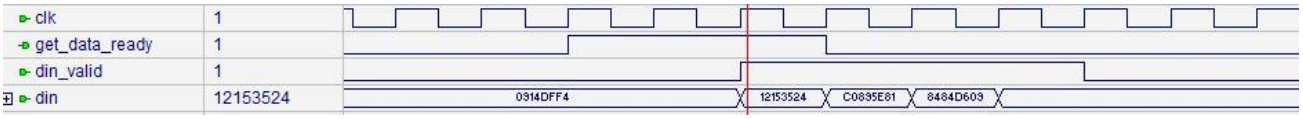

Figure 6.2. Waveform of 128-bit AES Key or Plain Text Data Input Simulation

As shown in Figure 6.3, when you input plain text data through HSP, this reference design still uses the din_valid signal as AES key write enable. The plaintext write enable signal is changed to the HSP_wr_i signal.

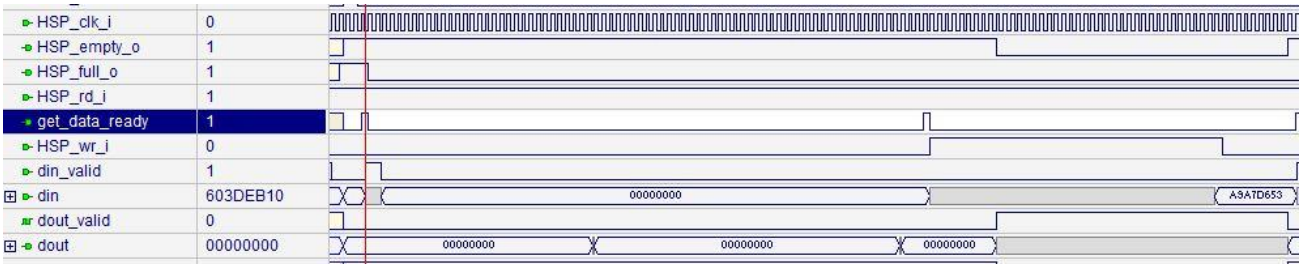

Figure 6.3. Waveform of HSP Flow Simulation

The data flow starts from random number generation. These random numbers are stored to encryption_data_in.txt. You can use these random numbers as plain text performing the AES encryption. After the encryption is done, the encryption result is stored to encryption_data_out.txt. You can use the encryption result as plain text performing the AES decryption. After the decryption is done, the decryption result is stored to decryption_data_out.txt. The contents of encryption_data_in.txt should be the same as those of decryption_data_out.txt.

© 2019 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

7. Implementation

This reference design is implemented in Verilog HDL using Lattice Diamond® software. The synthesis tool is set to Synplify Pro®. When using this design in a different device, density, speed, or grade, performance and utilization may vary.

Table 7.1 shows the device performance and resource utilization for this reference design.

Table 7.1. Performance and Resource Utilization

Device Family	Language	Utilization	Operating Frequency	ESB Primitive	OSC Primitive	Number of I/O
LCMXO3D-9400HC	Verilog HDL	407 LUTs	>50 MHz	Yes	Yes	80

© 2019 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Reference

MachXO3D Embedded Security Block (FPGA-TN-02091)

© 2019 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

© 2019 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Revision History

Revision 0.90, May 2019

Section	Change Summary
All	First preliminary release.

© 2019 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

www.latticesemi.com