Using MachXO3D ESB to Implement ECIES Encryption and Decryption

Reference Design

FPGA-RD-02055-1.0

August 2021

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.

© 2021 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

© 2021 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
ECC	Elliptic Curve Cryptography
ECIES	Elliptic Curve Integrated Encryption Scheme
ENC	Encryption
ESB	Embedded Security Block
KDF	Key Derivation Function
MAC	Message Authentication Code

© 2021 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

Elliptic Curve Integrated Encryption Scheme (ECIES) is an encryption standard based on asymmetric key encryption algorithm, using a public key for encryption and a private key for decryption. The public key and the private key are based on Elliptic Curve Cryptography (ECC).

ECIES provides capabilities for encryption, key exchange, and digital signature together. It is called Integrated Encryption Scheme since it is a hybrid scheme that uses a public key system to transport a session key for use by a symmetric cipher. In ECIES, a Diffie-Hellman shared secret is used to derive two symmetric keys, K1 and K2. The key K1 is used to encrypt the plaintext using a symmetric-key cipher, while the key K2 is used to authenticate the resulting cipher text.

ECIES uses the following cryptographic primitives:

- KDF is the key derivation function that is constructed from a hash function. In the Embedded Security Block (ESB) engine, ANSI X9.63 is used.
- ENC is the encryption function for a symmetric key encryption such as the AES. DEC is the decryption function. In the ESB engine, XOR is used.
- MAC is a message authentication code algorithm such as HMAC. In the ESB engine, HMAC-SHA256 is used.

Figure 1.1 shows the ECIES encryption flow.

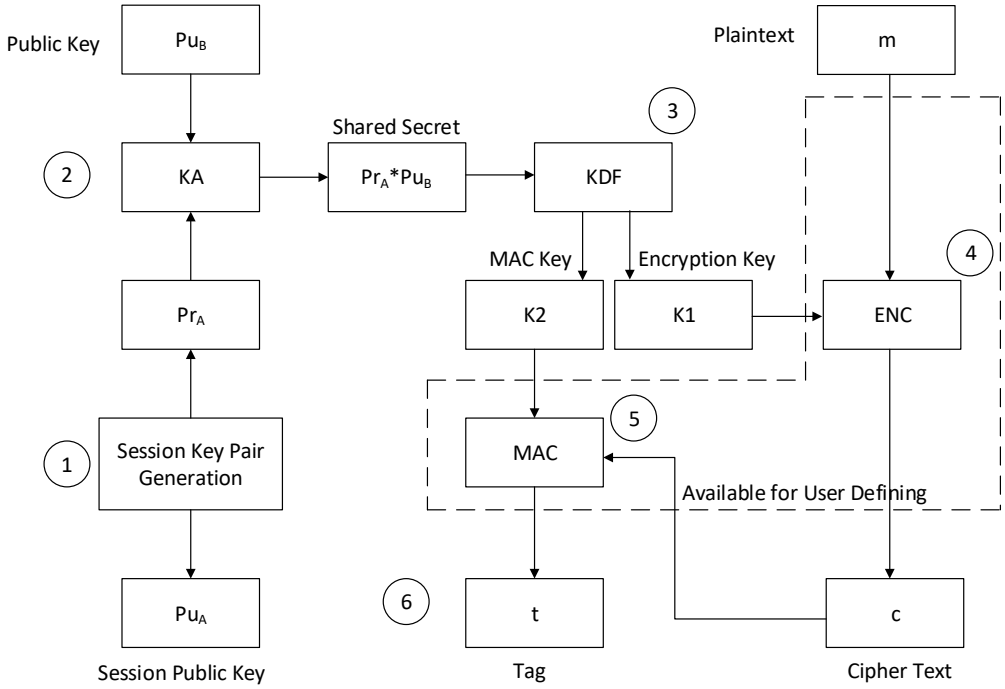

Figure 1.1. ECIES Encryption Flow

For encryption, the general flow is:

1. Alice must create a session key pair as a random secret value, PrA as the session private key and PuA as the associated session public key. The key pair should be generated exclusively for the current session.
2. Alice uses the Key Agreement function KA to create a shared secret value based on Alice's session private key Pr_A and Bob's public key Pu_B.
3. Alice must take the shared secret value as the input data for the Key Derivation Function, KDF. The output of this function is the concatenation of the symmetric encryption key, K1, and the MAC key, K2.
4. With K1 and the plaintext, m, Alice uses the symmetric encryption algorithm, ENC, to produce the cipher text, c.
5. With the cipher text c, Alice must use the selected MAC function with the MAC key K2 to produce a tag.
6. The session public key Pu_A, Tag t, and cipher text C are sent to recipient.

© 2021 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.

Figure 1.2 shows the ECIES decryption flow.

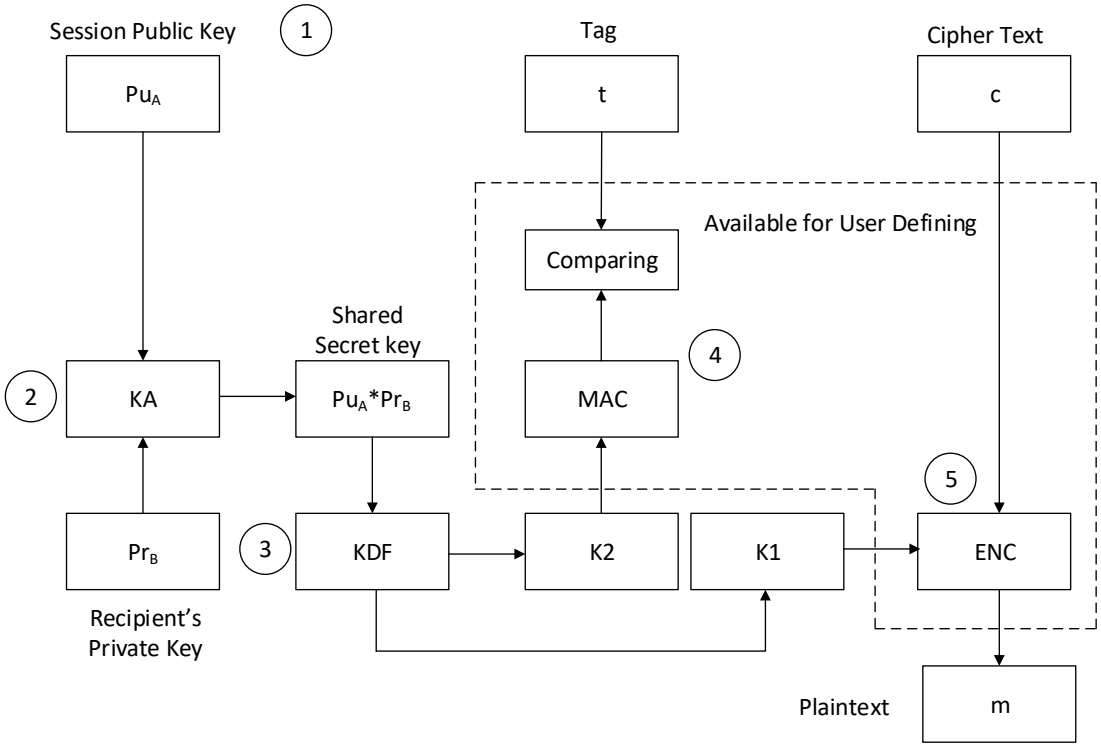

Figure 1.2. ECIES Decryption Flow

For decryption, the general flow is:

1. After receiving the cryptogram {Pu_A,t,c} from Alice, Bob must retrieve the session public key Pu_A, the tag t, and the cipher text c. Bob can deal with these elements separately.
2. Using the retrieved session public key, Pu_A, and his own private key, Pr_B, Bob produces the shared secret value $Pr_B \cdot Pu_A$. The result of this computation is the same as the product $Pr_A \cdot Pu_B$, which is the core of the Diffie-Hellman protocol.
3. Taking the shared secret value as the input to KDF, Bob produces the same encryption and MAC keys K1 and K2 with KDF.
4. With the MAC key K2 and the encrypted message c, Bob computes the element tag, and compare this with the tag he received. If the values are different, Bob must reject the cryptogram due to the failure in MAC verification.
5. If the tag value generated by Bob is correct, he can continue the process by deciphering the cipher text c using the symmetric ENC algorithm and K1. At the end of the decryption process, Bob can access the plaintext that Alice intended to send him.

The ESB engine native ECIES support can only use the XOR and HMAC-SHA256 as the underlying encryption and MAC algorithm, and the plaintext can only be up to 32 bytes. To support other user-defined encryption and MAC algorithm, such as AES, or longer plain text input, you can get K1 and K2 from the ESB engine, and implement with the combination of the ESB engine and the fabric logic.

Security is the key feature for Lattice MachXO3D™ family devices. With the embedded ESB module in MachXO3D chips, ECIES encryption and decryption functions are available with the minimum FPGA fabric resource used.

© 2021 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

2.1. Block Diagram

Figure 2.1 is the block diagram of the ECIES encryption. Figure 2.2 is the block diagram of the ECIES decryption.

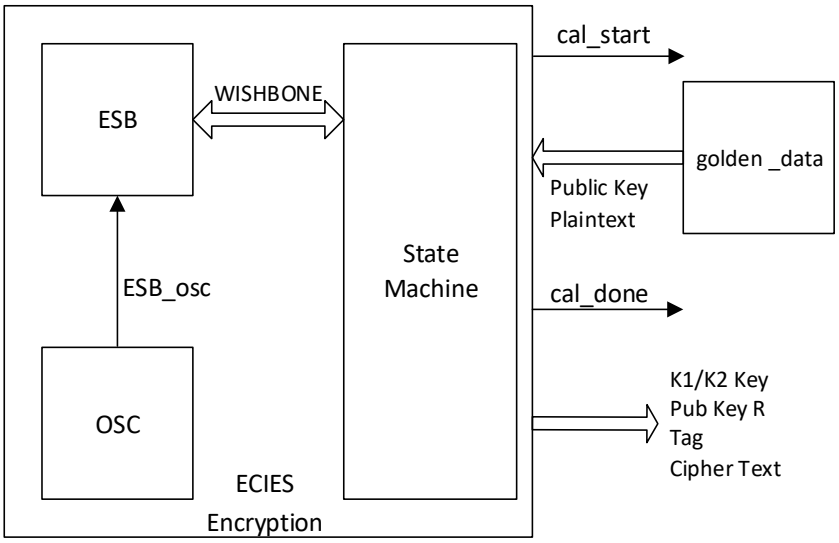

Figure 2.1. Top-Level Block Diagram for ECIES Encryption

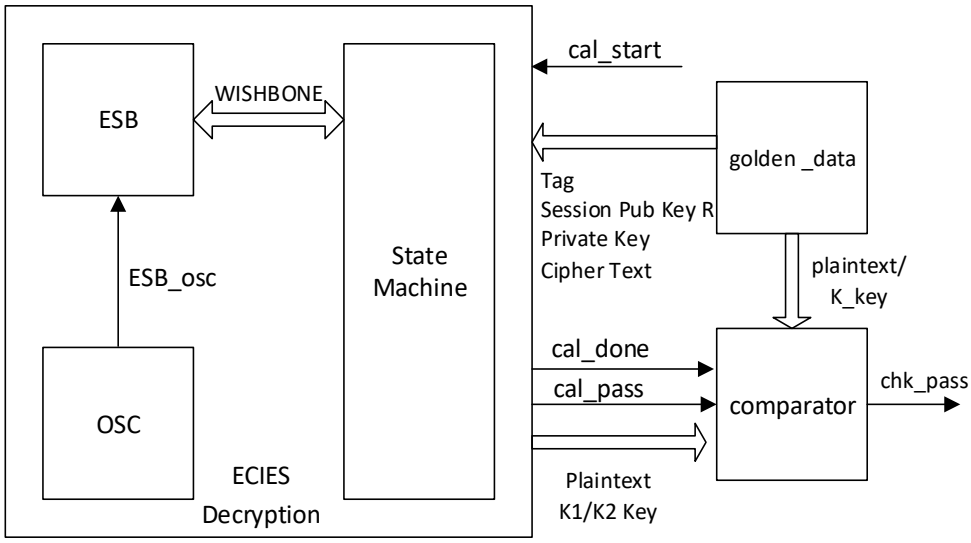

Figure 2.2. Top-Level Block Diagram for ECIES Decryption

© 2021 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.

2.2. Overview

The ECIES encryption design is used to generate the cipher text together with the session public key and tag based on user plaintext and public key input. The ECIES decryption design, on the other hand, is used to restore the plaintext from the cipher text, key, tag, and the private key. The features include support for:

- Programmable 64-byte public key (X, Y) for ECIES encryption
- 32-byte plaintext input for ECIES encryption
- Programmable 32-byte private key for ECIES decryption
- 32-byte cipher text input for ECIES decryption
- User-defined mode for encryption and MAC algorithm through the K1/K2 exporting
- User-defined plaintext and cipher text length through the fabric co-processing

© 2021 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

For ECIES encryption, with the 64-byte public key pair (X, Y) and 32-byte plain text input, after the calculation is done, the 32-byte cipher text, the 64-byte key K1/K2, the 96-byte session public key R, and the 32-byte tag are generated.

For ECIES decryption, with the 32-byte private key, the 96-byte session public key R, the 32-byte tag and the 32-byte cipher text, after the calculation is done, cal_pass indicates the decryption passes. If failed, no output data is available. If passed, the 32-byte decrypted plaintext can be read out.

3.1. Input/Output of the Design

Figure 3.1 is the I/O diagram of the ECIES encryption design. Figure 3.2 is the I/O diagram of the ECIES decryption design.

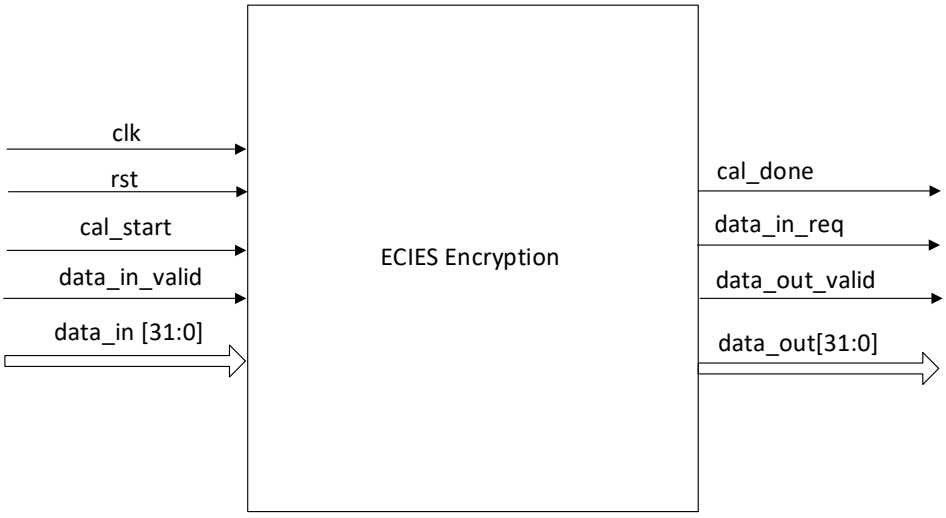

Figure 3.1. I/O Diagram for ECIES Encryption Design

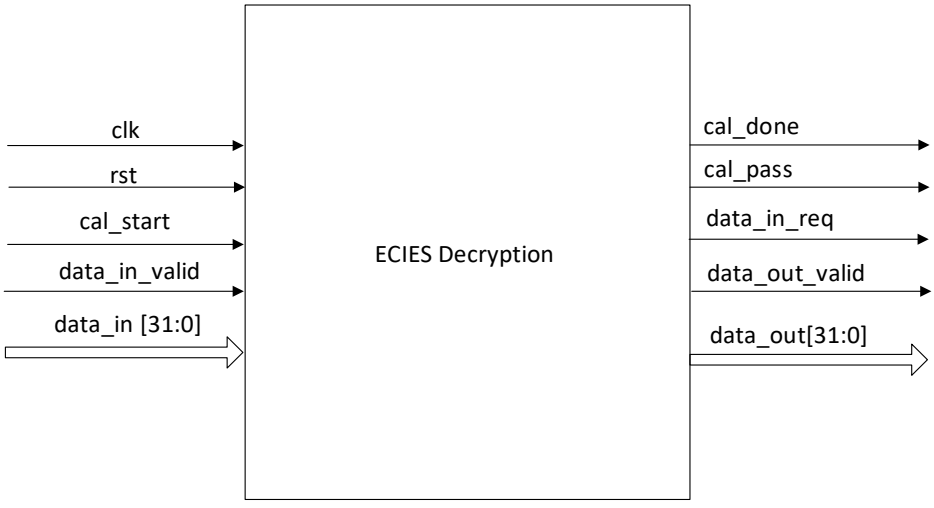

Figure 3.2. I/O Diagram for ECIES Decryption Design

© 2021 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.

Table 3.1 lists the I/O ports of the ECIES encryption/decryption designs.

Table 3.1. Pin Descriptions

Ports	Width	I/O	Description
Clock and Reset			
clk	1	I	System clock
rst	1	I	Asynchronous reset. Active High.
Control and Status			
cal_start	1	I	ECIES calculation starts. The rising edge triggers the calculation process.
cal_done	1	O	ECIES calculation done. Active High.
cal_pass (only for decryption)	1	O	ECIES decryption pass. Active High.
Input Data Interface			
data_in_req	1	O	Request flag for input data. Active High.
data_in_valid	1	I	Input data valid. Active High.
data_in	32	I	Input data : • For encryption, 64-byte public key and plain text; • For decryption, 32-byte private key, 96-byte session public key R, 32-byte tag and cipher text.
Output Data Interface			
data_out_valid	1	O	Output data valid. Active High.
data_out	32	O	Output data : • For encryption, 64-byte K key, 96-byte R key, 32-byte tag and cipher text; • For decryption, none if the tag authentication fails, or decoded plain text if the tag authentication passes.

© 2021 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

4.1. ESB Registers for ECIES Encryption/Decryption

Table 4.1. Register Descriptions

Register Type	Register Name	Address	Read/Write	Description
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
	ri_ctrl1	18'h2_000c	Write	ESB function: 0x00: Clear the ESB function request 0x07: ECIES Encryption 0x08: ECIES Decryption
Memory for Encryption	Pub_key_Qx_0	18'h1_F800	Write	Public_Key_X [31:0]
	Pub_key_Qx_1	18'h1_F804	Write	Public_Key_X [63:32]
	Pub_key_Qx_2	18'h1_F808	Write	Public_Key_X [95:64]
	Pub_key_Qx_3	18'h1_F80C	Write	Public_Key_X [127:96]
	Pub_key_Qx_4	18'h1_F810	Write	Public_Key_X [159:128]
	Pub_key_Qx_5	18'h1_F814	Write	Public_Key_X [191:160]
	Pub_key_Qx_6	18'h1_F818	Write	Public_Key_X [223:192]
	Pub_key_Qx_7	18'h1_F81C	Write	Public_Key_X [255:224]
	Pub_key_Qy_0	18'h1_F820	Write	Public_Key_Y [31:0]
	Pub_key_Qy_1	18'h1_F824	Write	Public_Key_Y [63:32]
	Pub_key_Qy_2	18'h1_F828	Write	Public_Key_Y [95:64]
	Pub_key_Qy_3	18'h1_F82C	Write	Public_Key_Y [127:96]
	Pub_key_Qy_4	18'h1_F830	Write	Public_Key_Y [159:128]
	Pub_key_Qy_5	18'h1_F834	Write	Public_Key_Y [191:160]
	Pub_key_Qy_6	18'h1_F838	Write	Public_Key_Y [223:192]
	Pub_key_Qy_7	18'h1_F83C	Write	Public_Key_Y [255:224]
	P_txt_0	18'h1_F8C0	Write	Plain_Text [31:0]
	P_txt_1	18'h1_F8C4	Write	Plain_Text [63:32]
	P_txt_2	18'h1_F8C8	Write	Plain_Text [95:64]
	P_txt_3	18'h1_F8CC	Write	Plain_Text [127:96]
	P_txt_4	18'h1_F8D0	Write	Plain_Text [159:128]
	P_txt_5	18'h1_F8D4	Write	Plain_Text [191:160]
	P_txt_6	18'h1_F8D8	Write	Plain_Text [223:192]
	P_txt_7	18'h1_F8DC	Write	Plain_Text [255:224]
	Key_k1_0	18'h1_F800	Read	Key_K1 [31:0]
	Key_k1_1	18'h1_F804	Read	Key_K1 [63:32]
	Key_k1_2	18'h1_F808	Read	Key_K1 [95:64]
	Key_k1_3	18'h1_F80C	Read	Key_K1 [127:96]
	Key_k1_4	18'h1_F810	Read	Key_K1 [159:128]
	Key_k1_5	18'h1_F814	Read	Key_K1 [191:160]
	Key_k1_6	18'h1_F818	Read	Key_K1 [223:192]
	Key_k1_7	18'h1_F81C	Read	Key_K1 [255:224]
	Key_k2_0	18'h1_F820	Read	Key_K2 [31:0]
	Key_k2_1	18'h1_F824	Read	Key_K2 [63:32]
	Key_k2_2	18'h1_F828	Read	Key_K2 [95:64]
	Key_k2_3	18'h1_F82C	Read	Key_K2 [127:96]

© 2021 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.

Register Type	Register Name	Address	Read/Write	Description
	Key_k2_4	18'h1_F830	Read	Key_K2 [159:128]
	Key_k2_5	18'h1_F834	Read	Key_K2 [191:160]
	Key_k2_6	18'h1_F838	Read	Key_K2 [223:192]
	Key_k2_7	18'h1_F83C	Read	Key_K2 [255:224]
	Key_rx_0	18'h1_F840	Read	Key_Rx [31:0]
	Key_rx_1	18'h1_F844	Read	Key_Rx [63:32]
	Key_rx_2	18'h1_F848	Read	Key_Rx [95:64]
	Key_rx_3	18'h1_F84C	Read	Key_Rx [127:96]
	Key_rx_4	18'h1_F850	Read	Key_Rx [159:128]
	Key_rx_5	18'h1_F854	Read	Key_Rx [191:160]
	Key_rx_6	18'h1_F858	Read	Key_Rx [223:192]
	Key_rx_7	18'h1_F85C	Read	Key_Rx [255:224]
	Key_ry_0	18'h1_F860	Read	Key_Ry [31:0]
	Key_ry_1	18'h1_F864	Read	Key_Ry [63:32]
	Key_ry_2	18'h1_F868	Read	Key_Ry [95:64]
	Key_ry_3	18'h1_F86C	Read	Key_Ry [127:96]
	Key_ry_4	18'h1_F870	Read	Key_Ry [159:128]
	Key_ry_5	18'h1_F874	Read	Key_Ry [191:160]
	Key_ry_6	18'h1_F878	Read	Key_Ry [223:192]
	Key_ry_7	18'h1_F87C	Read	Key_Ry [255:224]
	Key_rz_0	18'h1_F880	Read	Key_Rz [31:0]
	Key_rz_1	18'h1_F884	Read	Key_Rz [63:32]
	Key_rz_2	18'h1_F888	Read	Key_Rz [95:64]
	Key_rz_3	18'h1_F88C	Read	Key_Rz [127:96]
	Key_rz_4	18'h1_F890	Read	Key_Rz [159:128]
	Key_rz_5	18'h1_F894	Read	Key_Rz [191:160]
	Key_rz_6	18'h1_F898	Read	Key_Rz [223:192]
	Key_rz_7	18'h1_F89C	Read	Key_Rz [255:224]
	Tag_0	18'h1_F8A0	Read	Tag [31:0]
	Tag_1	18'h1_F8A4	Read	Tag [63:32]
	Tag_2	18'h1_F8A8	Read	Tag [95:64]
	Tag_3	18'h1_F8AC	Read	Tag [127:96]
	Tag_4	18'h1_F8B0	Read	Tag [159:128]
	Tag_5	18'h1_F8B4	Read	Tag [191:160]
	Tag_6	18'h1_F8B8	Read	Tag [223:192]
	Tag_7	18'h1_F8BC	Read	Tag [255:224]
	C_txt_0	18'h1_F8C0	Read	Cipher_Text [31:0]
	C_txt_1	18'h1_F8C4	Read	Cipher_Text [63:32]
	C_txt_2	18'h1_F8C8	Read	Cipher_Text [95:64]
	C_txt_3	18'h1_F8CC	Read	Cipher_Text [127:96]
	C_txt_4	18'h1_F8D0	Read	Cipher_Text [159:128]
	C_txt_5	18'h1_F8D4	Read	Cipher_Text [191:160]
	C_txt_6	18'h1_F8D8	Read	Cipher_Text [223:192]
	C_txt_7	18'h1_F8DC	Read	Cipher_Text [255:224]
Memory for Decryption	Prv_key_0	18'h1_F800	Write	Prv_Key [31:0]
	Prv_key_1	18'h1_F804	Write	Prv_Key [63:32]
	Prv_key_2	18'h1_F808	Write	Prv_Key [95:64]
	Prv_key_3	18'h1_F80C	Write	Prv_Key [127:96]

© 2021 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.

Register Type	Register Name	Address	Read/Write	Description
	Prv_key_4	18'h1_F810	Write	Prv_Key [159:128]
	Prv_key_5	18'h1_F814	Write	Prv_Key [191:160]
	Prv_key_6	18'h1_F818	Write	Prv_Key [223:192]
	Prv_key_7	18'h1_F81C	Write	Prv_Key [255:224]
	Key_rx_0	18'h1_F820	Write	Key_Rx [31:0]
	Key_rx_1	18'h1_F824	Write	Key_Rx [63:32]
	Key_rx_2	18'h1_F828	Write	Key_Rx [95:64]
	Key_rx_3	18'h1_F82C	Write	Key_Rx [127:96]
	Key_rx_4	18'h1_F830	Write	Key_Rx [159:128]
	Key_rx_5	18'h1_F834	Write	Key_Rx [191:160]
	Key_rx_6	18'h1_F838	Write	Key_Rx [223:192]
	Key_rx_7	18'h1_F83C	Write	Key_Rx [255:224]
	Key_ry_0	18'h1_F840	Write	Key_Ry [31:0]
	Key_ry_1	18'h1_F844	Write	Key_Ry [63:32]
	Key_ry_2	18'h1_F848	Write	Key_Ry [95:64]
	Key_ry_3	18'h1_F84C	Write	Key_Ry [127:96]
	Key_ry_4	18'h1_F850	Write	Key_Ry [159:128]
	Key_ry_5	18'h1_F854	Write	Key_Ry [191:160]
	Key_ry_6	18'h1_F858	Write	Key_Ry [223:192]
	Key_ry_7	18'h1_F85C	Write	Key_Ry [255:224]
	Key_rz_0	18'h1_F860	Write	Key_Rz [31:0]
	Key_rz_1	18'h1_F864	Write	Key_Rz [63:32]
	Key_rz_2	18'h1_F868	Write	Key_Rz [95:64]
	Key_rz_3	18'h1_F86C	Write	Key_Rz [127:96]
	Key_rz_4	18'h1_F870	Write	Key_Rz [159:128]
	Key_rz_5	18'h1_F874	Write	Key_Rz [191:160]
	Key_rz_6	18'h1_F878	Write	Key_Rz [223:192]
	Key_rz_7	18'h1_F87C	Write	Key_Rz [255:224]
	Tag_0	18'h1_F880	Write	Tag [31:0]
	Tag_1	18'h1_F884	Write	Tag [63:32]
	Tag_2	18'h1_F888	Write	Tag [95:64]
	Tag_3	18'h1_F88C	Write	Tag [127:96]
	Tag_4	18'h1_F890	Write	Tag [159:128]
	Tag_5	18'h1_F894	Write	Tag [191:160]
	Tag_6	18'h1_F898	Write	Tag [223:192]
	Tag_7	18'h1_F89C	Write	Tag [255:224]
	C_txt_0	18'h1_F8C0	Write	Cipher_Text [31:0]
	C_txt_1	18'h1_F8C4	Write	Cipher_Text [63:32]
	C_txt_2	18'h1_F8C8	Write	Cipher_Text [95:64]
	C_txt_3	18'h1_F8CC	Write	Cipher_Text [127:96]
	C_txt_4	18'h1_F8D0	Write	Cipher_Text [159:128]
	C_txt_5	18'h1_F8D4	Write	Cipher_Text [191:160]
	C_txt_6	18'h1_F8D8	Write	Cipher_Text [223:192]
	C_txt_7	18'h1_F8DC	Write	Cipher_Text [255:224]
	Chk_result	18'h1_F800	Read	Check_Result [31:0]
	Dec_txt_0	18'h1_F8C0	Read	Dec_Text [31:0]
	Dec_txt_1	18'h1_F8C4	Read	Dec_Text [63:32]
	Dec_txt_2	18'h1_F8C8	Read	Dec_Text [95:64]

© 2021 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.

Register Type	Register Name	Address	Read/Write	Description
	Dec_txt_3	18'h1_F8CC	Read	Dec_Text [127:96]
	Dec_txt_4	18'h1_F8D0	Read	Dec_Text [159:128]
	Dec_txt_5	18'h1_F8D4	Read	Dec_Text [191:160]
	Dec_txt_6	18'h1_F8D8	Read	Dec_Text [223:192]
	Dec_txt_7	18'h1_F8DC	Read	Dec_Text [255:224]
	Dec_key_k1_0	18'h1_F880	Read	Dec_Key_K1 [31:0]
	Dec_key_k1_1	18'h1_F884	Read	Dec_Key_K1 [63:32]
	Dec_key_k1_2	18'h1_F888	Read	Dec_Key_K1 [95:64]
	Dec_key_k1_3	18'h1_F88C	Read	Dec_Key_K1 [127:96]
	Dec_key_k1_4	18'h1_F890	Read	Dec_Key_K1 [159:128]
	Dec_key_k1_5	18'h1_F894	Read	Dec_Key_K1 [191:160]
	Dec_key_k1_6	18'h1_F898	Read	Dec_Key_K1 [223:192]
	Dec_key_k1_7	18'h1_F89C	Read	Dec_Key_K1 [255:224]

Before starting an ECIES encryption (Figure 4.1), check the status of the ESB module, and make sure that it is ready to get a new command. Then you can transfer the 64-byte public key and the 32-byte plain text to the ESB module, and set the ESB function to the ECIES encryption mode. After the calculation, Done is shown in the ESB status. The elliptic curve key pair (K1, K2), the session public key R, tag, and the cipher text are available. All the data transactions are Little Endian.

© 2021 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.

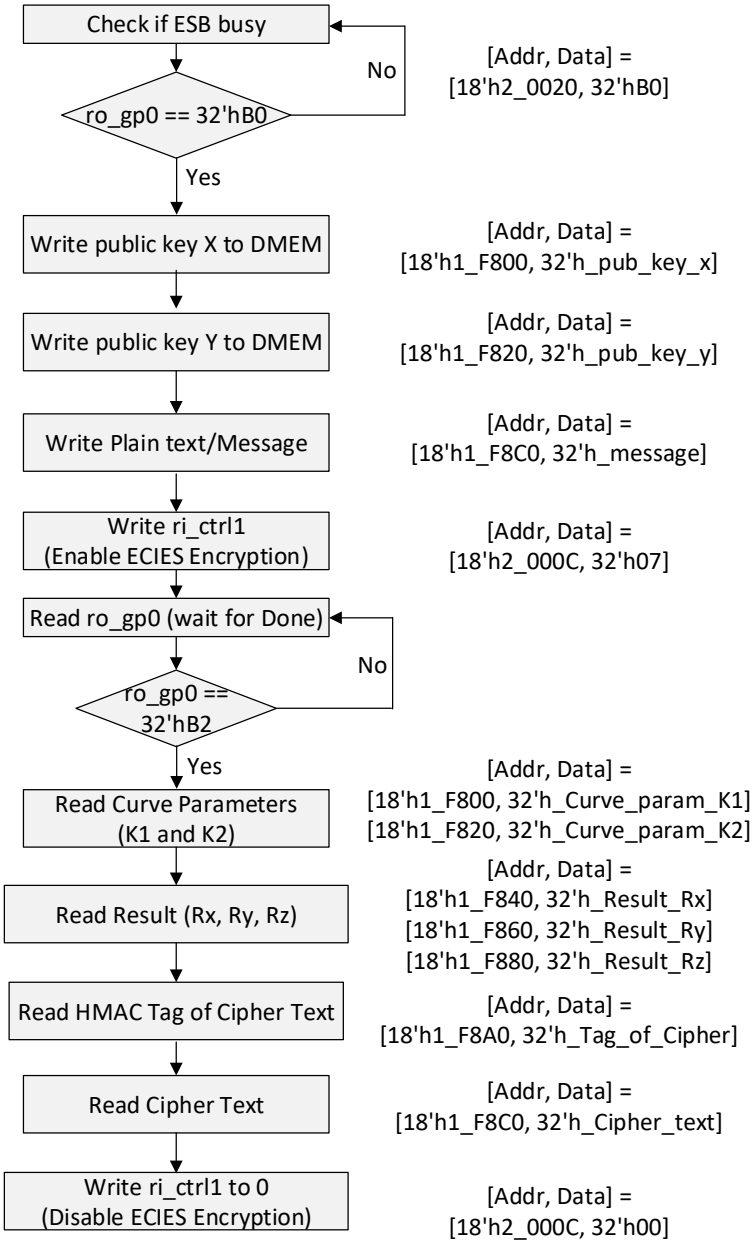

Figure 4.1. ECIES Encryption Algorithm

To decode the cipher text encrypted by ECIES (Figure 4.2), you should transfer the 32-byte private key, the 96-byte session public key R, tag, and cipher text to the ESB module. Then set the ESB function to the ECIES decryption mode. When the calculation is done, check the result of tag authentication. Only when the result passes, the decoded plaintext can be read out. The key K1 can be read out to support user-defined decryption algorithm, such as AES. All the data transactions are Little Endian.

© 2021 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.

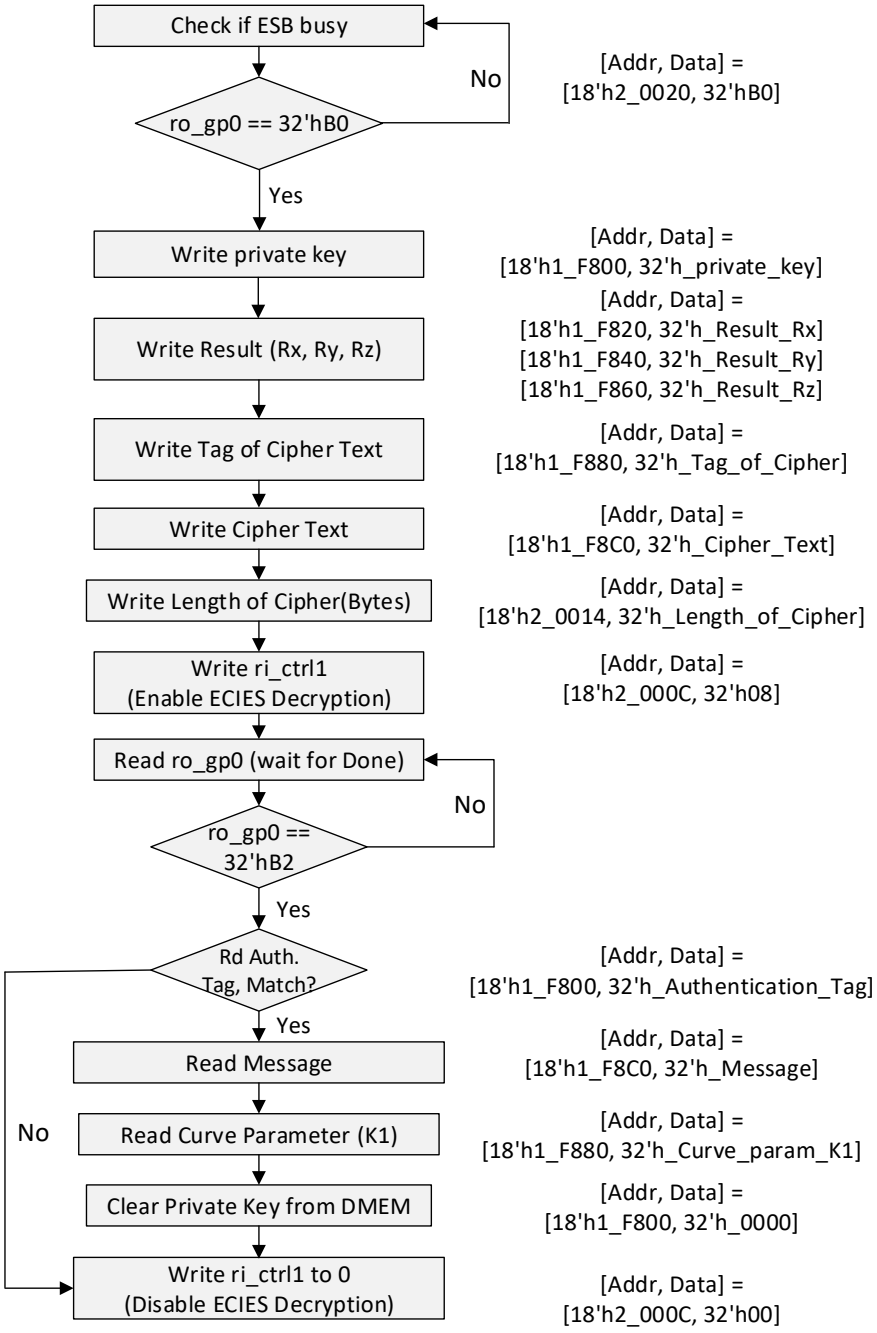

Figure 4.2. ECIES Decryption Algorithm

© 2021 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. Simulation and Verification

Active-HDL is used for the function simulation of the ECIES generation and verification of the designs.

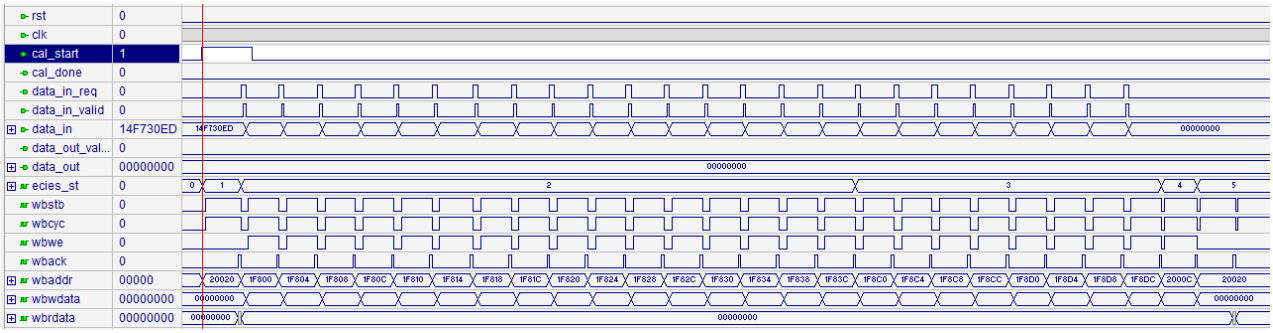

Figure 5.1. ECIES Encryption: Data Input and Mode Setting

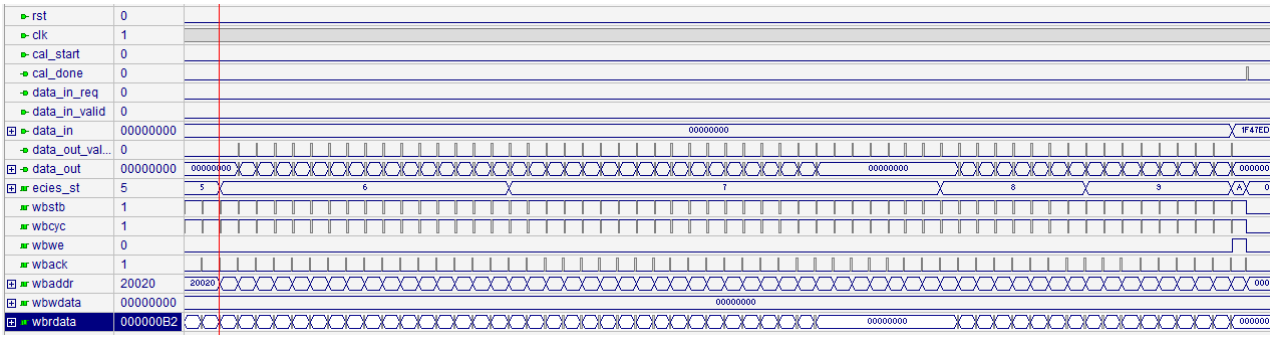

Figure 5.2. ECIES Encryption: Calculation Done and Data Output

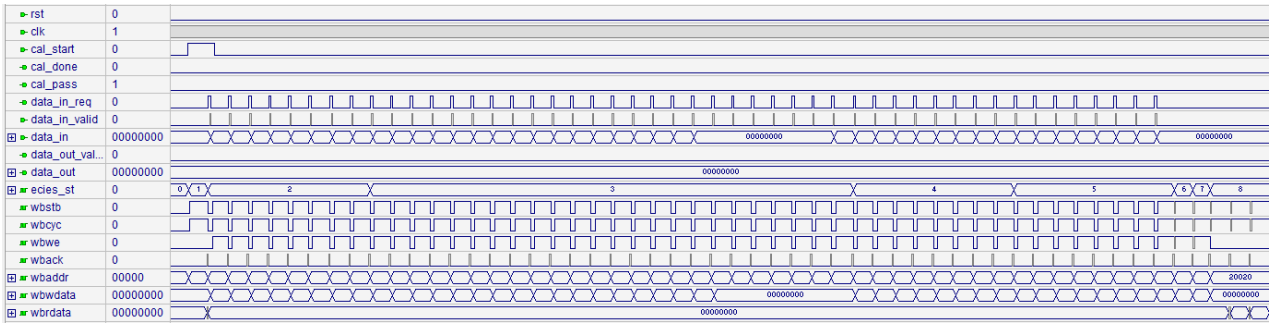

Figure 5.3. ECIES Decryption: Data Input and Mode Setting

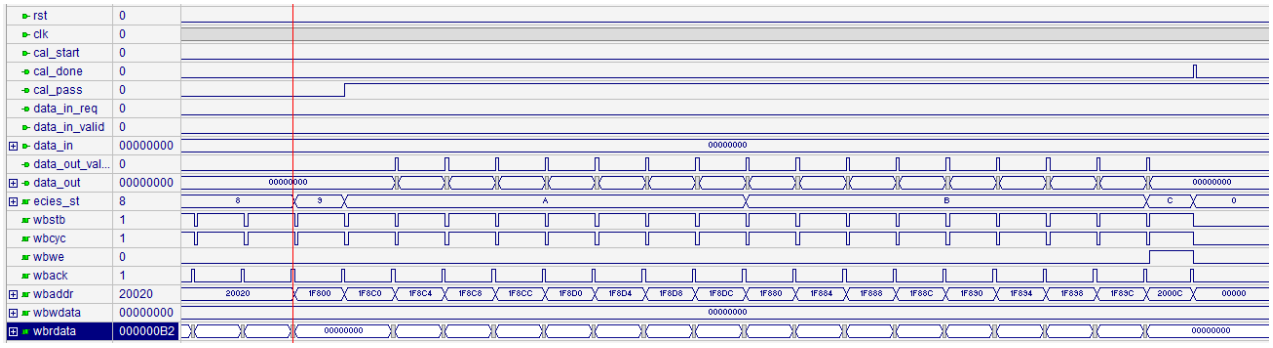

Figure 5.4. ECIES Decryption: Calculation Done and Data Output

© 2021 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. Implementation

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

Table 6.1. Performance and Resource Utilization for ECIES Encryption Design

Device Family	Language	Utilization	ESB Primitive	OSC Primitive	Fmax	Number of I/O
LCMXO3D-9400HC	Verilog HDL	185 LUTs	Yes	Yes	> 50 MHz	71

Table 6.2. Performance and Resource Utilization for ECIES Decryption Design

Device Family	Language	Utilization	ESB Primitive	OSC Primitive	Fmax	Number of I/O
LCMXO3D-9400HC	Verilog HDL	189 LUTs	Yes	Yes	> 50 MHz	72

Note: Performance and utilization characteristics are generated with LCMXO3D-9400HC and Diamond 3.11 design software.

© 2021 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)

© 2021 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.

© 2021 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 1.0, August 2021

Section	Change Summary
All	Production release.

© 2021 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