Speedster7t Cryptographic Engine User Guide (UG104)

Speedster FPGAs

Achronix
Data Acceleration

Copyrights, Trademarks and Disclaimers

Notice of Disclaimer

Achronix Semiconductor Corporation

2903 Bunker Hill Lane
Santa Clara, CA 95054
USA

Website: www.achronix.com
E-mail : info@achronix.com

Table of Contents

Chapter 1 : Description

The cryptographic engine, ACX_AESX_GMC_K, supports data encryption/decryption and implements an AES algorithm using Rijndael encoding/decoding in compliance with the NIST Advanced Encryption Standard[1]. The encryption is suitable for a variety of applications in the public and private domain.

The Advance Encryption Standard (AES) is a symmetric block cipher chosen by the US government to protect classified information. Symmetric, also known as *secret key*, ciphers use the same key for encrypting and decrypting so the sender and receiver must both know and use the same secret key. Compared to the DES and triple DES algorithms, AES provides a higher level of security because it has a larger key size and is also faster. In addition, DES has become vulnerable to brute-force attacks.

All of the inputs are synchronous to the input clock signal. The cryptographic engine processes blocks of messages using a key. The size of the key is determined by input port `ksize.` The size of the the key based on `ksize` is determined in the table below. The number of clock cycles after which the core is ready to accept the additional or message data is dependent on input `ksize`.

Table 1 • Key Size Data Based on ksize Input

ksize	Key Size (in bits)	Clock Cycles After Key Size and Data Message can be Applied
00	128	14
01	192	16
10	256	18

The data can be input with byte resolution using the `ibyte` and `last_w` inputs. The `last_w` signal indicates the last word being input. The `ibyte` signal is ignored until `last_w` is asserted, indicating the number of valid bytes in the last word minus 1 (counted from the MSB). So `ibyte` = "0000" means that only the first byte in the incoming word is valid and `ibyte`="1111" means that all bytes are valid.

On the AC7t1400/AC7t1500, devices, the cryptographic engine is instantiated in the southeast corner of the FPGA fabric. The cryptographic engine core is pre-placed and pre-routed. Although it is implemented in the fabric, it can be considered a hard IP core because the placement and routing cannot be modified.

On the AC7t800, the cryptographic engine resides within the I/O ring.

1 https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=11510

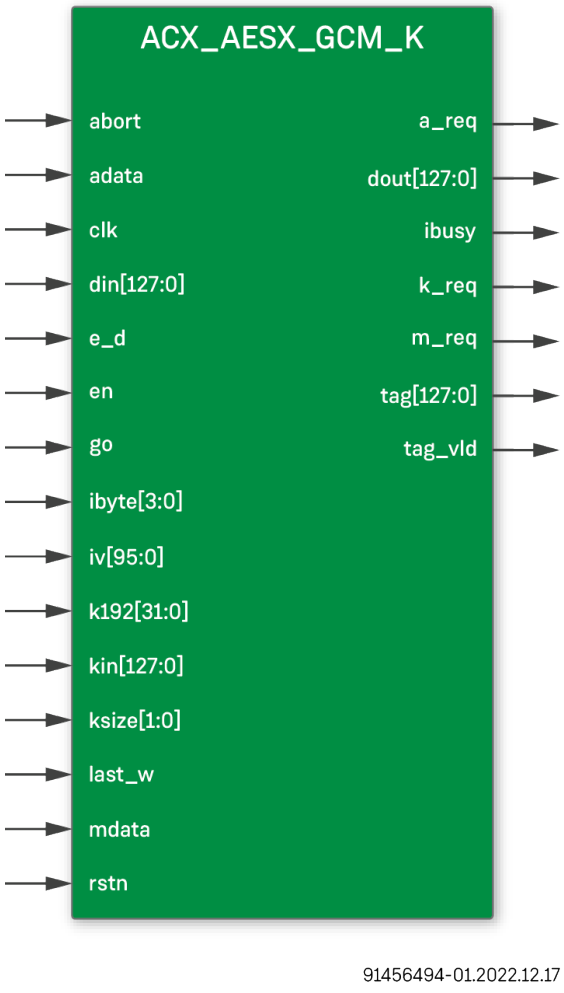

Figure 1 • ACX_AESX_GMC_K Symbol

Input and Output Ports

Table 2 • ACX_AESX_GMC_K Port Description

Name	Direction	Width	Description
rstn	Input	1	Active-low asynchronous reset.
clk	Input	1	Clock signal.
en	Input	1	Synchronous enable signal.

Name	Direction	Width	Description
go	Input	1	Starts cryptographic operation when = 1.
abort	Input	1	Aborts current operation when = 1.
e_d	Input	1	Mode signal: • Encryption when = 0 • Decryption when = 1
kin[127:0]	Input	128	Key data input.
ksize[1:0]	Input	2	Input key size.
k192[31:0]	Input	31	Unexpanded key.
din[127:0]	Input	128	Input data: • Contains *additional* input data when adata = 1 • Contains *message* input data when mdata = 1 • adata and mdata are mutually exclusive cannot be 0 at the same time
iv[95:0]	Input	96	Initialization vector.
adata	Input	1	Additional data is input when adata = 1.
mdata	Input	1	Message data is input when mdata = 1.
ibyte[3:0]	Input	4	Indicates the number of valid bytes in the last din word – 1. Valid when last_w is asserted. (†) • ibyte = 4'h0 = 1 byte • ibyte = 4'hf = 16 bytes, (full 128-bit word)
last_w	Input	1	When = 1, last additional or message data word is input. Validates ibyte[3:0] input.
k_req	Output	1	When = 1, the unexpanded key is requested.
a_req	Output	1	When = 1, additional data is requested.

Name	Direction	Width	Description
m_req	Output	1	When = 1, message data is requested.
ibusy	Output	1	When = 1, the core is in the initialization process.
dout[127:0]	Output	128	Processed message data output.
tag[127:0]	Output	128	Authenticated tag value output.
tag_vld	Output	1	Authenticated tag value valid output.

Table Notes

† ibyte is scaled differently from many other last byte values. It is equal to (last din word – 1). If moving between other interfaces with mod signals, be aware of this scaling difference.

Chapter 2 : Operation

The ACX_AESX_GMC_K core supports both encryption and decryption according to the AES algorithm.

The rising edge on the go port triggers the beginning of a cryptographic operation using the key input as the key. The en signal must be asserted one cycle before the go signal. The mdata and adata signals must remain stable.

When the core is started, it requests the unexpanded key (one 128-bit word) by raising k_req. The application must assert the

kin

input on the same cycle that k_req is asserted (not on the following cycle). After 14 cycles from ibusy being asserted, the core is ready to accept message data or additional data. This sequeance acts as the initialization phase and is indicated by the ibusy signal being asserted.

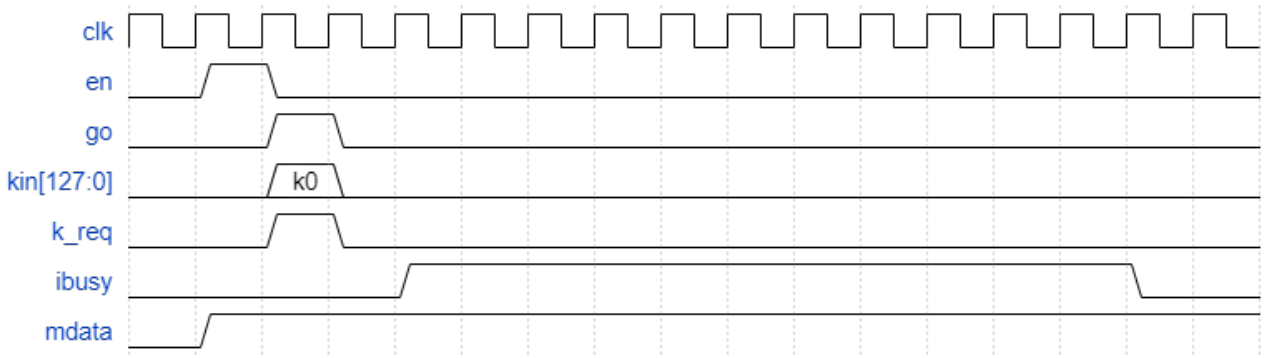

Figure 2 • Initialization Phase

After initialization, the core asserts m_req on the third clock cycle after the falling edge of ibusy if mdata is asserted, indicating that the core is now accepting message data on every clock cycle. The application must apply data to the core two cycles before the assertion of

m_req

. Therefore, it is recommended that the application count cycles from the de-assertion of ibusy to ensure that the first word of data is applied on the same cycle that m_req is asserted.

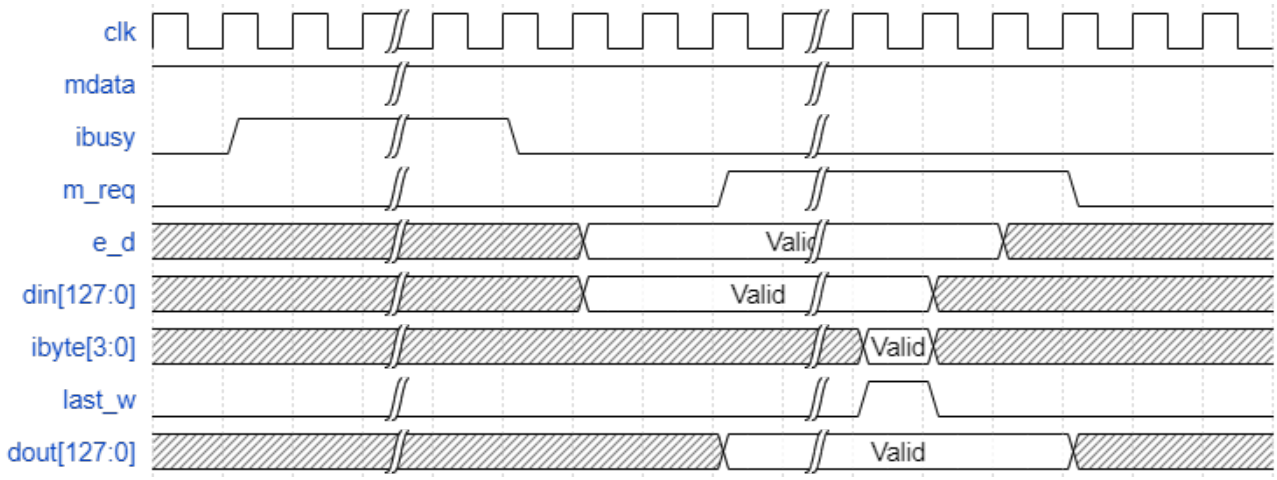

Figure 3 • Message Data Input and Output

The core asserts `a_req` on the third clock cycle after the falling edge of `ibusy` if `adata` is asserted. The type of data requested is indicated by the `a_req` and `m_req` signals for additional and message data respectively.

The output is synchronous as there are flops on the output. The encrypted or decrypted message data is the result of the AES counter operation XORed with the incoming data, as shown above.

Also, the related `dout` signal is two cycles behind `din` and so continues for two cycles after `last_w`. The `e_d` signal only needs to be valid when data is being input because decryption only affects the authentication tag calculation.

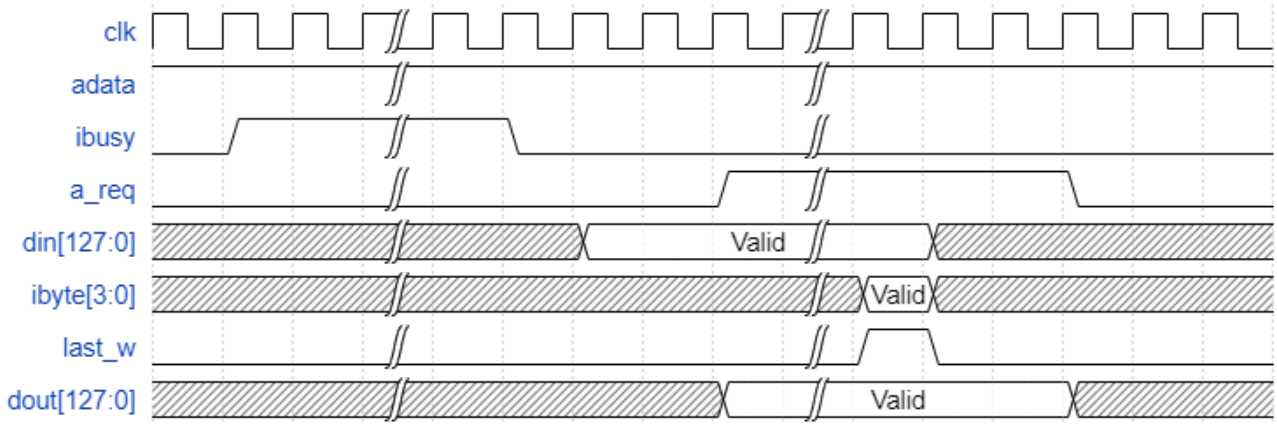

Figure 4 • Additional Data Input and Output

Chapter 3 : Usage

AC7t1400/AC7t1500

The ACX_AESX_GMC_K cryptographic engine is instantiated as a Verilog module within the AC7t1400/AC7t1500 in the southeast corner of the FPGA fabric. There can be only one instance of the ACX_AESX_GMC_K module in the user design. Even though the placement for this is in a fixed location, this module is instantiated like a soft IP block. For more information on how to make use of soft IP, consult the ***Speedster7t Soft IP User Guide*** **(UG103)**[2].

> **Caution!**
>
> This ACX_AESX_GMC_K IP is licensed from **CAST**[3] for use for evaluation purposes only. Any use this soft IP in a Speedster product requires a separate license and support agreement directly from CAST.

Implementation

Support for the cryptographic engine has been integrated into ACE via partition flow. The flow automatically takes care of importing the cryptographic engine as a partition along with the relevant EPDB file.

The cryptographic engine is placed in the southeast corner of the device as shown **below** **(see figure 4)** (highlighted in Blue). The placement is fixed and cannot be modified by the user.

2 https://achronix.com/documentation/speedster7t-soft-ip-user-guide-ug103
3 https://www.cast-inc.com/

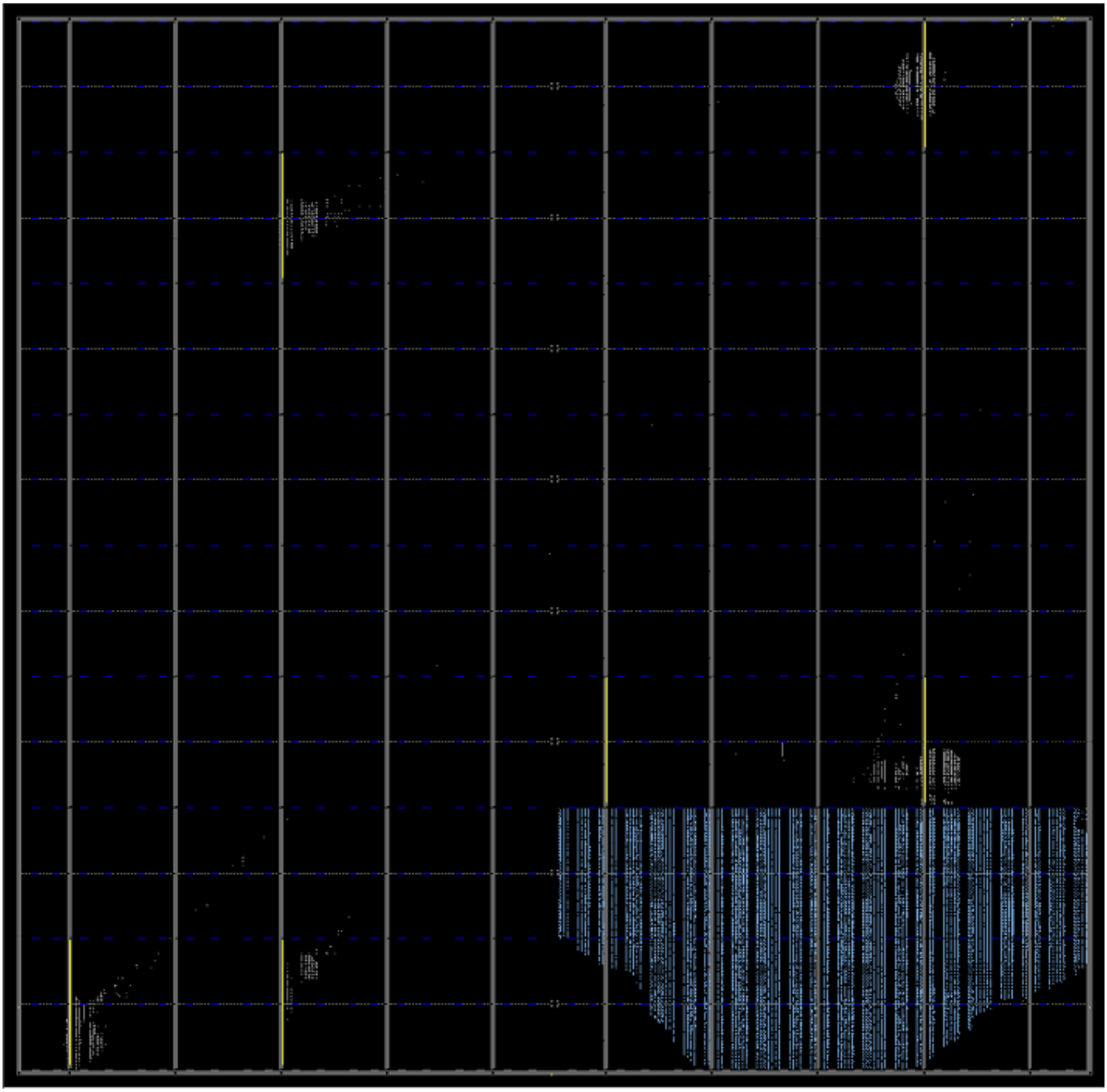

Figure 5 • Cryptographic Engine Placement

AC7t800

The cryptographic engine resides in the I/O ring on the AC7t800 as a hard IP in the southeast corner. Therefore, when generating the appropriate I/O ring files for this design, the pins will be included in the relevant I/O ring PDC and port list files that are created during I/O ring generation. Furthermore, user designs must supply a 200 MHz clock to the hard IP along with a reset source.

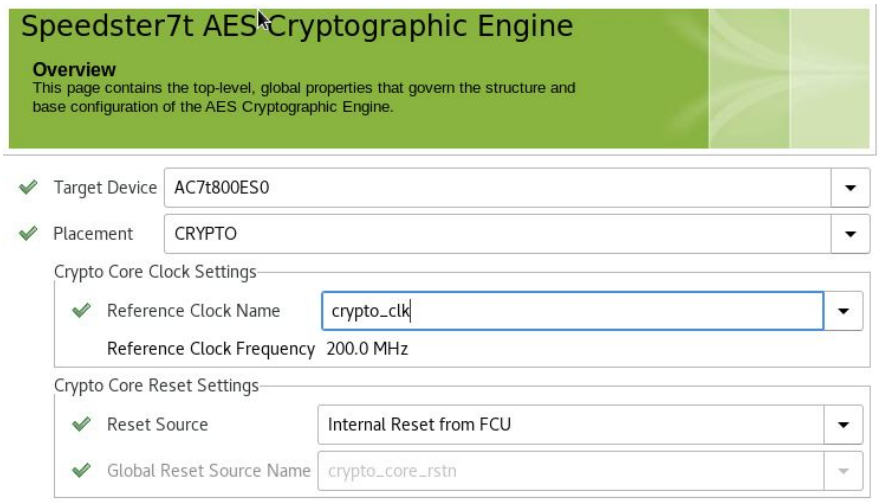

Figure 6 • Cryptographic Engine as Hard IP

Chapter 4 : Templates

Verilog Functional Core as Generated By ACE

The following code block lists the ports of the Verilog module generated by ACE for the cryptographic engine on AC7t1400/AC7t1500. This Verilog module gets generated like all other soft IP modules as described in the ***peedster7t Soft IP User Guide*** **(UG103)**[4]. The name of the module can be specified by the user. The Verilog code below makes use of the default name set by ACE.

```
module aes_25g_cryptographic_engine_1
  (
   input abort,    // Aborts current operation when = 1.
   input adata,    // Additional data is input when adata = 1.
   input [127:0] din,      // data input; Contains Additional input data when adata = 1.
Contains Message input data when mdata = 1. adata and mdata are mutually exclusive
cannot be 0 at the same time.
   input e_d,      // 1:encryption, 0:decryption
   input en,       // Synchronous enable signal.
   input go,       // Starts cryptographic operation when = 1.
   input [3:0] ibyte,    // Indicates the number of valid bytes in the last din word-1.
Valid when last_w is asserted.
   input [95:0] iv,       // initialization vector
   input [31:0] k192,     // unexpanded key
   input [127:0] kin,      // key data input
   input [1:0] ksize,    // input key size
   input last_w,   // When = 1, last Additional or Message data word is input. Validates
ibyte[3:0] input.
   input mdata,    // Message data is input when mdata = 1.
   output a_req,    // When = 1, Additional data is requested.
   output [127:0] dout,     // Processed data out.
   output ibusy,    // When = 1, the core is in the initialization process.
   output k_req,    // When = 1, the unexpanded key is requested.
   output m_req,    // When = 1, Message data is requested.
   output [127:0] tag,      // Authenticated tag value output.
   output tag_vld,  // Authenticated tag value valid output.

   // Reset and Clock

   input clk,      // Clock input for 25G Cryptographic Engine block
   input rstn      // Neg-edge reset input for 25G Cryptographic Engine block
  );
```

4 https://achronix.com/documentation/speedster7t-soft-ip-user-guide-ug103

I/O Ring Ports

The following ports are generated by ACE during I/O ring generation as seen in the `.pdc` file created during this process.

```
// Ports for aes_cryptographic_engine_1
// Clocks and Resets
input wire        aes_cryptographic_engine_1_clk,
// Crypto Interface
input wire        aes_cryptographic_engine_1_crypto_a_req,
input wire        aes_cryptographic_engine_1_crypto_data_message_req,
input wire  [127:0] aes_cryptographic_engine_1_crypto_dout,
input wire        aes_cryptographic_engine_1_crypto_ibusy,
input wire        aes_cryptographic_engine_1_crypto_k_req,
input wire        aes_cryptographic_engine_1_crypto_m_req,
input wire  [127:0] aes_cryptographic_engine_1_crypto_tag,
input wire        aes_cryptographic_engine_1_crypto_tag_vld,
input wire        aes_cryptographic_engine_1_rstn,
output wire       aes_cryptographic_engine_1_crypto_abort,
output wire       aes_cryptographic_engine_1_crypto_adata,
output wire [127:0] aes_cryptographic_engine_1_crypto_din,
output wire       aes_cryptographic_engine_1_crypto_e_d,
output wire       aes_cryptographic_engine_1_crypto_en,
output wire       aes_cryptographic_engine_1_crypto_go,
output wire [3:0] aes_cryptographic_engine_1_crypto_ibyte,
output wire [95:0] aes_cryptographic_engine_1_crypto_iv,
output wire [31:0] aes_cryptographic_engine_1_crypto_k192,
output wire [127:0] aes_cryptographic_engine_1_crypto_kin,
output wire [1:0] aes_cryptographic_engine_1_crypto_ksize,
output wire       aes_cryptographic_engine_1_crypto_last_w,
output wire       aes_cryptographic_engine_1_crypto_mdata
```

Chapter 5 : Revision History

Version	Date	Description
1.0	17 Sep 2021	Initial release.
2.0	12 Feb 2025	▪ Removed references to AC7t1550 as device has been deprecated ▪ Added information for AC7t800 and AC7t1400 support.