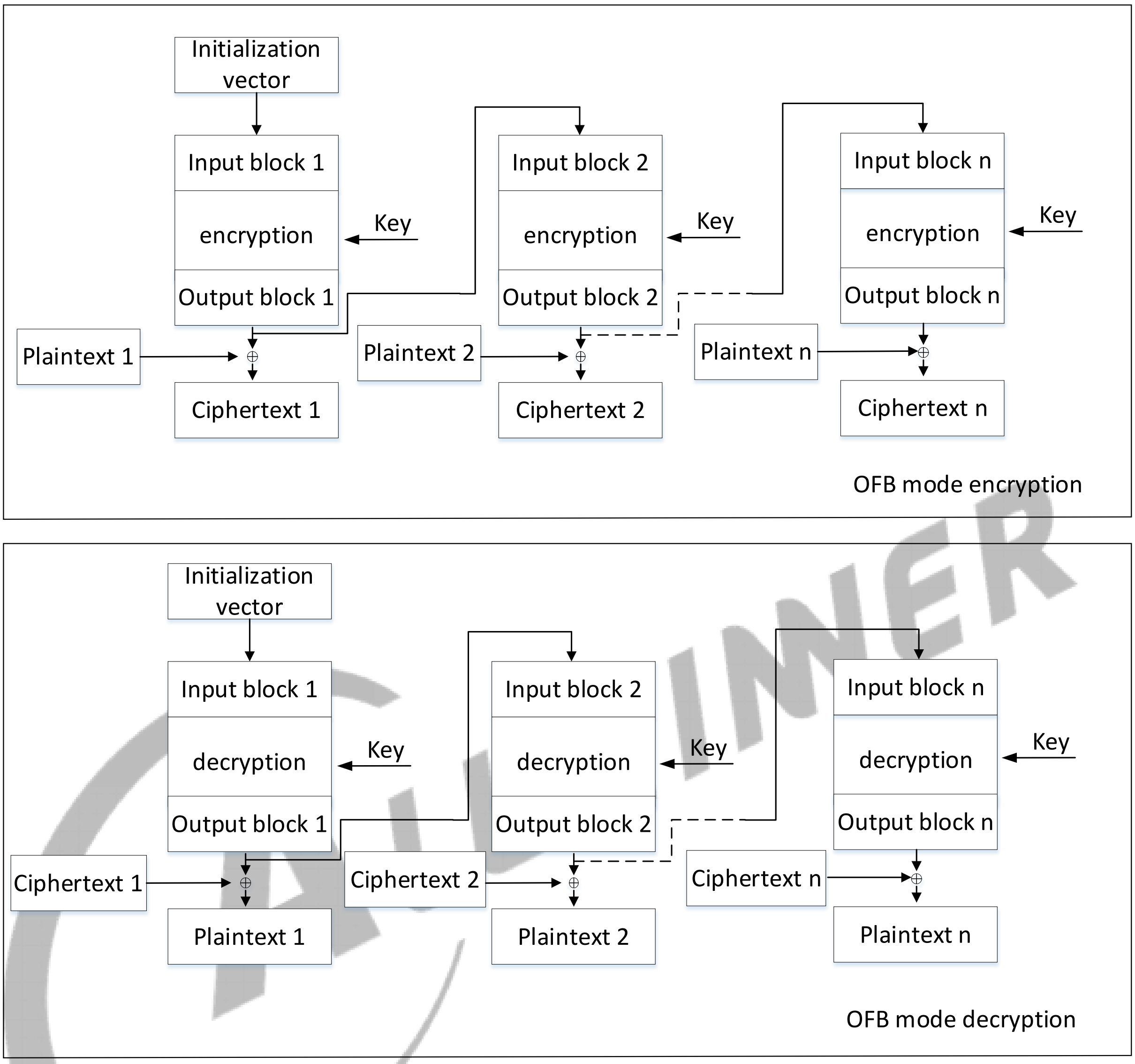
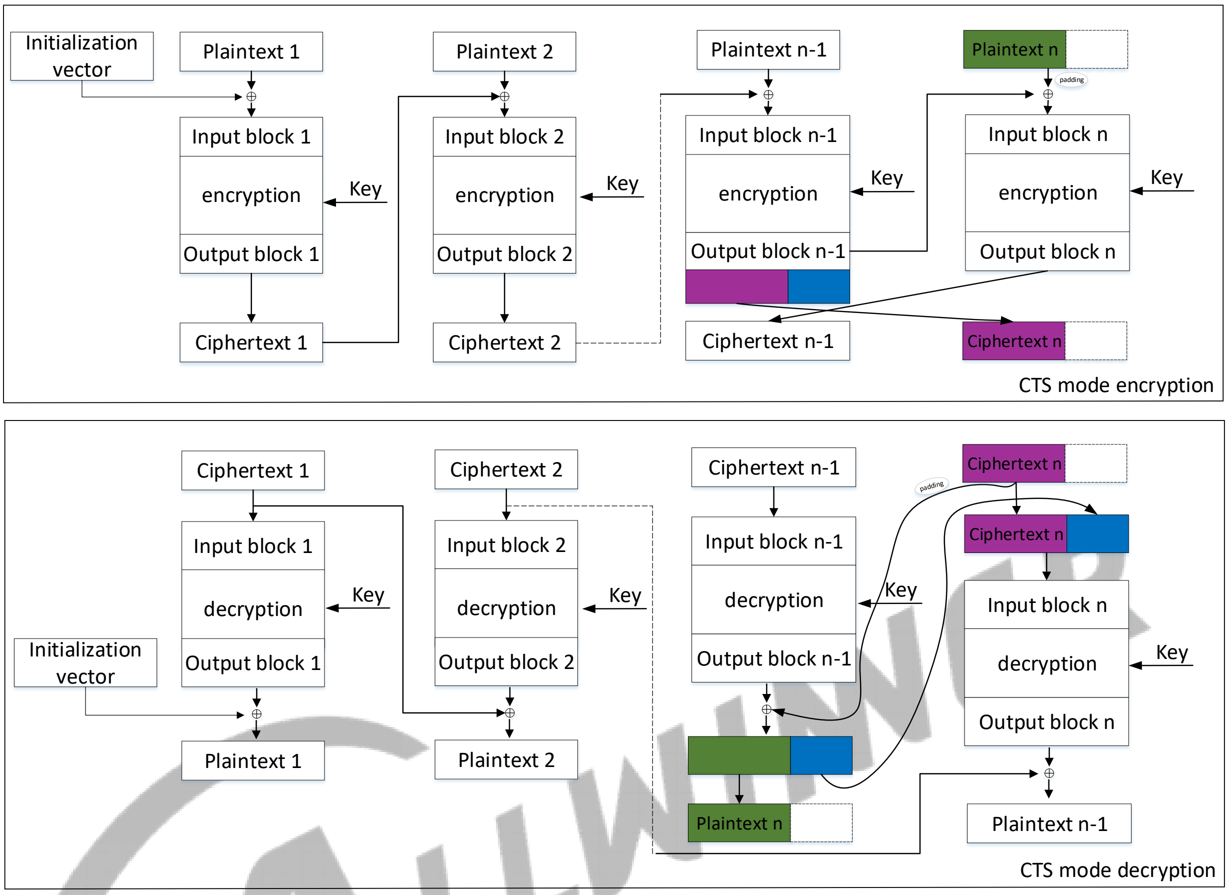
Figure 10-8 OFB Mode Encryption and Decryption



10.1.3.8 CTS Mode

The CTS mode is a confidentiality mode that accepts any plaintext input whose bit length is greater than or equal to the block size but not necessarily a multiple of the block size. Below are the diagrams for CTS encryption and decryption.

Figure 10-9 CTS Mode Encryption and Decryption



10.1.3.9 HASH Algorithm

The hash algorithms support MD5, SHA1, SHA224, SHA256, SHA384, SHA512, HMAC-SHA1, and HMAC-SHA256. All algorithms are iterative, one-way hash functions that can process a message to produce a condensed representation called a *message digest*. When a message is received, the *message digest* can be used to verify whether the data has changed, that is, to verify its integrity.

The hash algorithm of the CE supports block-aligned total length of the input data (padded by software), that is, a multiple of 64 bytes. The message length after padding by software is used as the configured data length for the hash algorithm.

10.1.3.10 RSA Algorithm

The RSA is a public key encryption/decryption algorithm implemented through the modular exponentiation operation.

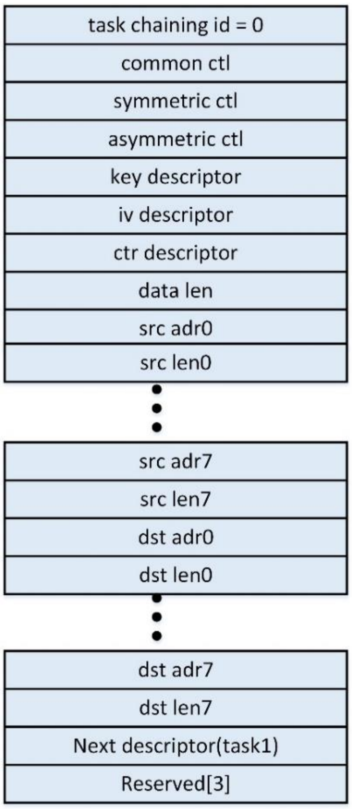
The ciphertext is obtained as follows: C = M^E mod N. The plaintext is obtained as follows: M = C^D mod N.

M indicates the plaintext, C indicates the ciphertext, (N, E) indicates the public key, and (N, D) indicates the private key.

10.1.3.11 Task Descriptor

The software makes request through task descriptor, including algorithm type, algorithm mode, key address, source/destination address and size, and so on. The structure of the task descriptor is as follows.

Figure 10-10 Structure of Task Descriptor Chaining



task chaining id = 0
common ctl
symmetric ctl
asymmetric ctl
key descriptor
iv descriptor
ctr descriptor
data len
src adr0
src len0
src adr7
src len7
dst adr0
dst len0
dst adr7
dst len7
Next descriptor(task1)
Reserved[3]

The bit definitions of the task descriptor are as follows.

Task ID

Bit	Read/Write	Default/Hex	Description
31:4	/	/	/

Bit	Read/Write	Default/Hex	Description
3:0	R/W	0x0	CHN Task channel ID Indicates which channel the task is running on. It supports 0 to 3.

Common Control

Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	Interrupt enable (IE) for the current task 0: disable interrupt 1: enable interrupt Represents whether an interrupt signal is generated when the task chain ends at the end of this task. When the last task in a task chain ends, the operation of the task chain will end normally; if a task fails in the middle, the task chain will be aborted abnormally. And it is determined whether to generate an interrupt signal according to the IE configuration of the current task when the current task ends or aborts. Therefore, if you want to use interrupts, it is recommended that not only the IE of the last task of each task chain is configured to 1 to generate the end interrupt of the task chain, but also the IEs of other tasks in this task chain are also configured to 1. The purpose is to generate an interrupt signal once an abnormal error occurs in these tasks and the interrupt is aborted.
30:17	/	/	/
16	R/W	0x0	IV mode IV mode for SHA1/SHA224/SHA256/SHA384/SHA512/MD5 or constants 0: use initial constants defined in FIPS-180 1: use input iv
15	R/W	0x0	Last HMAC plaintext 0: not the last HMAC plaintext package 1: the last HMAC plaintext package
14:9	/	/	/

Bit	Read/Write	Default/Hex	Description
8	R/W	0x0	OP DIR Algorithm Operation Direction 0: Encryption 1: Decryption Configure according to the requirements of encryption or decryption.
7	/	/	/
6:0	R/W	0x0	Algorithm Type 0x0: AES 0x1: DES 0x2: Triple DES (3DES) 0x10: MD5 0x11: SHA-1 0x12: SHA-224 0x13: SHA-256 0x14: SHA-384 0x15: SHA-512 0x16: HMAC-SHA1 0x17: HMAC-SHA256 0x20: RSA 0x30: TRNG 0x31: PRNG Others: reserved

Symmetric Control

Bit	Read/Write	Default/Hex	Description
31:24	/	/	/
23:20	R/W	0x0	KEY_SELECT key select for AES 0000: Select input CE_KEYx (Normal Mode) 0001: Select {SSK} 0010: Select {HUK} 0011: Select {RSSK} 0100-0111: Reserved 1000-1111: Select internal Key n (n from 0 to 7)

Bit	Read/Write	Default/Hex	Description
19:18	R/W	0x0	CFB_WIDTH AES-CFB width 00: CFB1 01: CFB8 10: CFB64 11: CFB128
17	R/W	0x0	PRNG_LD Load new 15 bits key into linear feedback shift register (LFSR) for PRNG. When the algorithm type is PRNG, it is necessary to post-process the random number output by PRNG through the linear shift operation to generate the operand. When the PRNG_LD is configured to 1, use iv_addr[14:0] as the input number for linear shift operation, and do XOR operation between the data and the random number output by PRNG to generate the post-processing result of further operation.
16	R/W	0x0	CTS_LPKG AES CTS last package flag When set to '1', it means this is the last package for AES-CTS mode (the size of the last package is larger than 128 bits).
15:12	/	/	/
11:8	R/W	0x0	ALGORITHM_MODE CE algorithm mode 0000: Electronic Code Book (ECB) mode 0001: Cipher Block Chaining (CBC) mode 0010: Counter (CTR) mode 0011: Cipher Text Stealing (CTS) mode 0100: Output Feedback (OFB) mode 0101: Cipher Feedback (CFB) mode Other: reserved
7:4	/	/	/
3:2	R/W	0x0	CTR WIDTH Counter width for CTR mode 00: 16-bit Counter 01: 32-bit Counter 10: 64-bit Counter 11: 128-bit Counter

Bit	Read/Write	Default/Hex	Description
1:0	R/W	0x0	AES KEY SIZE 00: 128-bit 01: 192-bit 10: 256-bit 11: Reserved

Asymmetric Control

Bit	Read/Write	Default/Hex	Description
31	/	/	/
30:28	R/W	0x0	RSA Pubic Modulus Width 000: 512-bit 001: 1024-bit 010: 2048-bit Other: reserved
27:19	/	/	/
18:16	R/W	0x0	RSA MODE RSA algorithm mode. For modular computation: 000: modular exponent(RSA) 001: modular div 010: modular mul 011: modular inv others: reserved
15:0	/	/	/

Key Descriptor

Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	Key Address The address of KEY that needs to be stored.

IV Descriptor

Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	IV Address The address of IV that needs to be stored.

Counter Descriptor

Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	CTR Data Output Address The address of CTR data output that needs to be stored.

Data Length

Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	Data Length Configure the data length of the corresponding segment. The data length size needs to be consistent with dst_data_length (destination data length 0 +... + destination data length 7). The data length field in the task descriptor has different meanings for different algorithms. For AES-CTS, the data length field indicates byte numbers of source data, for others indicate word numbers of source data. For PRNG, the data length should be 5 words aligned. For TRNG, it should be 8 words aligned.

Source Address 0~7

Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	Source Data Address The address of the source data that needs to be stored.

Source Data Length 0~7

Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	Source Data Length The length of the source data. Unit: byte

Destination Address 0~7

Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	Destination Data Address The address of the destination data that needs to be stored.

Destination Data Length 0~7

Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	Destination Data Length The length of the destination data. Unit: byte

Next Descriptor Address

Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	Next Task Address The address where the descriptor of the next task in a task-chain is saved. If there is the only task or the last task of a task-chain, the next task address must be 32’h0.

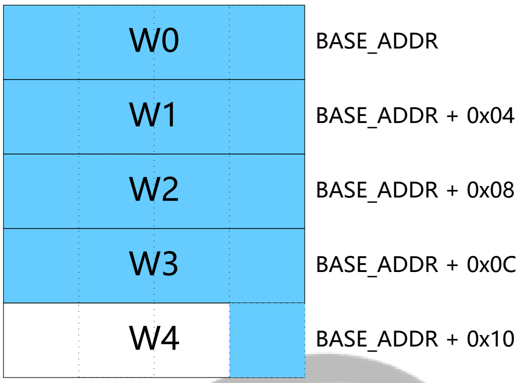
10.1.3.12 Storing Message

In the application, a message may not be stored contiguously in the memory, but divided into multiple segments. Or a piece of continuously stored messages can be artificially split into multiple pieces as needs.

Then each segment corresponds to a set of the source address and source length in the descriptor. Multiple segments correspond to groups 0-7 source address/source length in sequence.

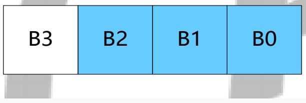
Each task supports up to 8 message segments, and the data volume of each message segment supports up to 4 GWord (AES-CTS is 1 GByte). The total amount of all segments in a task (that is a package) supports up to 4 GWord (AES-CTS is 1 GByte). If a message is divided into multiple packages, all others are required to be whole words; when the last package of AES-CTS is less than one word, 0 needs to be padded, and those less than one word are counted as one word. The following figure shows the address order structure.

Figure 10-11 Word Address of Message



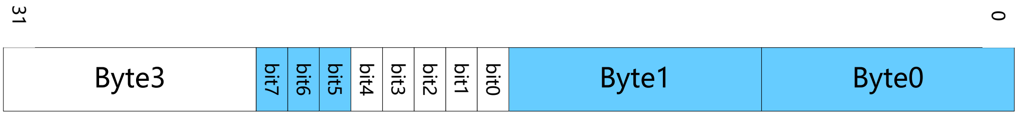
Byte order: low byte first, high byte last. When the data is less than one word, the low byte is filled first. The following figure shows the byte order structure (blue means it is filled by the message).

Figure 10-12 Byte Order



Bit order: high bit first, low bit last. When the data is less than one Byte, the high bit is filled first. The following figure shows the bit order structure.

Figure 10-13 Bit Order



10.1.3.13 Storing Key

The length of KEY must be an integer multiple of word, refer to section 10.1.3.15 "Algorithm Length Properties".

10.1.3.14 Storing IV

For different algorithms, the length of IV is different. But they are integer multiples of word. To keep the byte order of IV and HASH digest output consistent, the byte order of IV is different from that of the message. For the multi-packet operation, the first address of the digest output result of the previous HASH can be directly configured to the first address of the next IV, and the software does not need to do any processing on the digest.

The following figure shows the storage method of 32-bit IV value.

Figure 10-14 The Storage Method of 32-bit IV

IV0[31:0]	BASE_ADDR
IV1[31:0]	BASE_ADDR + 0x04
……	……
IV7[31:0]	BASE_ADDR + 0x1C

The following figure shows the storage method of 64-bit IV value.

Figure 10-15 The Storage Method of 64-bit IV

IV0[63:32]	BASE_ADDR
IV0[31:00]	BASE_ADDR + 0x04
IV1[63:32]	BASE_ADDR + 0x08
IV1[31:00]	BASE_ADDR + 0x0C
……	……
IV7[63:32]	BASE_ADDR
IV7[31:00]	BASE_ADDR + 0x3C

10.1.3.15 Algorithm Length Properties

The algorithm length has different requirements for different algorithms.

Table 10-1 Symmetric Algorithm Configuration Properties

Algorithm	Length Setting				Alignment	Software Padding
	Source Size	Destination Size	KEY	IV		
AES (except CTS)	< 4 GWord	< 4 GWord	AES-128: 4 Word AES-192: 6 Word AES 256: 8 Word	4 Word	Word-aligned	need
AES-CTS	< 1 GByte	< 1 GByte	AES-128: 4 word AES-192: 6 word AES 256: 8 word	4 Word	Word-aligned	need
DES	< 4 GWord	< 4 GWord	2 Word	2 Word	Word-aligned	need
TDES	< 4 GWord	< 4 GWord	6 Word	2 Word	Word-aligned	need

Table 10-2 Hash Algorithm Configuration Properties

Algorithm	Length Setting				Alignment	Software Padding
	Source Size	Destination Size	KEY	IV		
MD5	< 4 GWord	4 Word	Fixed to 0	4 Word	Word-aligned	need
SHA-1	< 4 GWord	5 Word	Fixed to 0	5 Word	Word-aligned	need
SHA-224	< 4 GWord	8 Word	Fixed to 0	8 Word	Word-aligned	need
SHA-256	< 4 GWord	8 Word	Fixed to 0	8 Word	Word-aligned	need
SHA-384	< 4 GWord	16 Word	Fixed to 0	16 Word	Word-aligned	need
SHA-512	< 4 GWord	16 Word	Fixed to 0	16 Word	Word-aligned	need
HMAC-SHA1	< 4 GWord	5 Word	16 Word	5 Word	Word-aligned	need
HMAC-SHA256	< 4 GWord	8 Word	16 Word	8 Word	Word-aligned	need

Table 10-3 RNG Configuration Properties

Algorithm	Length Setting				Alignment	Software Padding
	Source Size	Destination Size	KEY	IV		
TRNG	< 4 GWord	< 4 GWord	Fixed to 0	4 Word	Word-aligned	need
PRNG	< 4 GWord	< 4 GWord	6 Word	4 Word	Word-aligned	need

Table 10-4 Asymmetric Algorithm Configuration Properties

Algorithm	Length Setting				Alignment	Software Padding
	Source Size	Destination Size	KEY	IV		
RSA512	16 Word	16 Word	16 Word	Not use IV	Word-aligned	need
RSA1024	32 Word	32 Word	32 Word	Not use IV	Word-aligned	need
RSA2048	64 Word	64 Word	64 Word	Not use IV	Word-aligned	need

10.1.3.16 Security Operation

When the CPU issues request to the CE module, the CE module will save the secure mode of CPU. When executing this request, this state bit works as a access flag for the inner and system resources. For access to SID module through the AHB bus, only the secure mode can succeed, or else these keys will be read to 0 or cannot write. When issuing MBUS read and write requests, the CE will use send this secure mode bit to BUS, so secure requests can access secure and non-secure space, but non-secure requests only can access non-secure space.

10.1.3.17 Error Detection

The CE module includes error detection for task configuration, data computing error, and authentication invalid. When the algorithm type in task descriptor is read into the CE module, the CE will check whether this type is supported through checking algorithm type field in common control. If the type value is out of scope, the CE will issue interrupt signal and set error state. Each type has certain input and output data size. After getting a task descriptor, the input size and output size configuration will be checked to avoid size error. If the size configuration is wrong, the CE will issue interrupt signal and set error state.

10.1.3.18 Clock Requirement

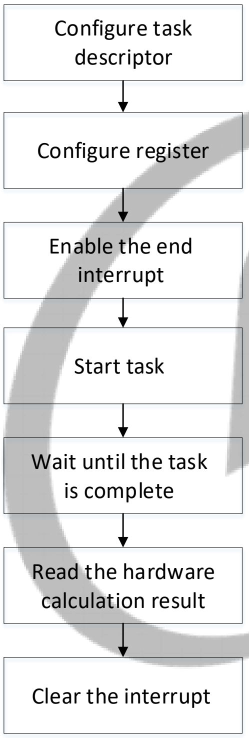
Clock Name	Description	Requirement
hclk	AHB bus clock	24 MHz – 200 MHz
mclk	MBUS clock	24 MHz – 400 MHz
ce_clk	CE work clock	24 MHz – 400 MHz

10.1.4 Programming Guidelines

10.1.4.1 Symmetrical/Asymmetrical/Hash/RNG Algorithm Operation Process

The following figure shows the process of an algorithm operation.

Figure 10-16 Task Request Process


Configure task descriptor
Configure register
Enable the end interrupt
Start task
Wait until the task is complete
Read the hardware calculation result
Clear the interrupt

Step 1 The software should configure a task descriptor in memory, including the related fields in the descriptor. The channel id corresponds to one channel in CE. According to algorithm type, the software should set the fields in common control, symmetric control, asymmetric control, then provide key/iv/ctr address and the data length of this task. The source and destination address and size are set based on the upper application. If there is another task concatenating after this task, then set its descriptor address at the next descriptor field. For more details for task descriptor, see section 10.1.4.2, section 10.1.4.3 and section 10.1.4.4.

Step 2 The software should set registers. Configure the first address of the task descriptor structure to CE Task Descriptor Address Register. Configure the source/destination address to CE Current Source Address Register/CE Current Destination Address Register.

Step 3 Enable the end interrupt of the corresponding task channel by setting CE Interrupt Control Register.

Step 4 The software reads CE Task Load Register to ensure that the bit0 is 0. If the bit0 is not read out to be 0, wait until it is 0, then configure the bit0 to be 1 to start task.

Step 5 Wait for interrupt status by reading CE Interrupt Status Register.

Step 6 Read the result from the destination address.

Step 7 Clear the interrupt.

10.1.4.2 Configuring Task Descriptor of AES

- **Common control**: Configure Common Control[6:0] to 0x0 to select AES algorithm type.
- **Symmetric control**: According to the corresponding algorithm requirements, configure Symmetric Control to select the key size, CTR width, CTS last package flag, CFB width, and AES algorithm mode, and so on.
- **Asymmetric control**: The symmetric algorithm does not need to be configured for this field.
- **Key descriptor**: Because the storage of the key requires word alignment, ensure that this descriptor is the first address of the KEY (word address).
- **IV descriptor**: In the task that requires the IV value, configure the first address of the storage space where the IV is stored here. Because the storage of the IV requires word alignment, ensure that this descriptor is the first address of the IV (word address).
- **Data length**: Configure the data length of the corresponding segment. The data length size needs to be consistent with dst_data_length (destination data length 0 +... + destination data length 7). When the algorithm is CTS mode, the higher 30-bit of the data length is the word numbers of data volume; when the data_length[1:0] is 0, the data length is the higher 30-bit, otherwise it is increased by 1. For AES CTS, the data length indicates the byte numbers of the source data; for other algorithms, it indicates the word numbers.
- **Source address**: The first address of source data segments. Because the storage of the source data requires word alignment, ensure that this descriptor is the first address (word address).
- **Source data length**: The data volume of source data segments. The unit is word, and those less than one word are counted as one word. Note that only the last word of the entire message is allowed to be non-integer words, and the others must be integer words.
- **Destination address**: The first address of destination data segments. Because the storage of the destination data requires word alignment, ensure that this descriptor is the first address (word address).

- **Destination data length**: The data volume of destination data segments. The unit is word, and those less than one word are counted as one word. Note that only the last word of the entire message is allowed to be non-integer words, and the others must be integer words.
- **Next descriptor**: The first address of the next task descriptor. Because the storage of the descriptor requires word alignment, ensure that this descriptor is the first address (word address).
- **Reserved**: Configure to 0x0.

10.1.4.3 Configuring Task Descriptor of HASH

- **Common control**
  - **Algorithm type**: Configure Common Control[6:0] to select SHA or HMAC algorithm type.
  - **Last HMAC plaintext**: If the algorithm type is HMAC, and the task is the last package of the message or if the message has only one package, then Common Control[15] needs to set to 1.
  - **IV mode**: The Common Control[16] (IV MODE) bit is only set to 1 in the following two scenarios, except that the bit must be configured to 0. (1). When the message is split into multiple packages, the Common Control[16] bit of other packages needs to be set to 1, except that the bit of the first package needs to be cleared to 0. (2). In special applications, if you need to customize the IV value to form the initial value of a certain HASH algorithm, you need to set the Common Control[16] bit of the first (or only one) package to 1, and the first address of the storage space where the customized IV value is stored in IV address.
- **Key descriptor**: Because the storage of the key requires word alignment, ensure that this descriptor is the first address of the KEY (word address).
- **IV descriptor**: In the task that requires the IV value, configure the first address of the storage space where the IV is stored here. Because the storage of the IV requires word alignment, ensure that this descriptor is the first address of the IV (word address).
- **Data length**: Configure the data length of the corresponding segment. The data length size needs to be consistent with dst_data_length (destination data length 0 +... + destination data length 7).
- **Source address**: The first address of source data segments. Because the storage of the source data requires word alignment, ensure that this descriptor is the first address (word address).
- **Source data length**: The data volume of source data segments. The unit is word, and those less than one word are counted as one word. Note that only the last word of the entire message is allowed to be non-integer words, and the others must be integer words.
- **Destination address**: The first address of destination data segments. Because the storage of the destination data requires word alignment, ensure that this descriptor is the first address (word address).

- **Destination data length**: The data volume of destination data segments. The unit is word, and those less than one word are counted as one word. Note that only the last word of the entire message is allowed to be non-integer words, and the others must be integer words.
- **Next descriptor**: The first address of the next task descriptor. Because the storage of the descriptor requires word alignment, ensure that this descriptor is the first address (word address).
- **Reserved**: Configure to 0x0.

10.1.4.4 Configuring Task Descriptor of RSA

- **Common control**: Configure Common Control[6:0] to 0x20 to select RSA algorithm type.
- **Asymmetric control**: Configure Asymmetric Control[30:28] to select the RSA width.
- **Key descriptor**: Because the storage of the key requires word alignment, ensure that this descriptor is the first address of the KEY (word address).
- **Data length**: Configure the data length of the corresponding segment. The data length size needs to be consistent with dst_data_length (destination data length 0 +... + destination data length 7).
- **Source address**: The first address of source data segments. Because the storage of the source data requires word alignment, ensure that this descriptor is the first address (word address).
- **Source data length**: The data volume of source data segments. The unit is word, and those less than one word are counted as one word. Note that only the last word of the entire message is allowed to be non-integer words, and the others must be integer words.
- **Destination address**: The first address of destination data segments. Because the storage of the destination data requires word alignment, ensure that this descriptor is the first address (word address).
- **Destination data length**: The data volume of destination data segments. The unit is word, and those less than one word are counted as one word. Note that only the last word of the entire message is allowed to be non-integer words, and the others must be integer words.
- **Next descriptor**: The first address of the next task descriptor. Because the storage of the descriptor requires word alignment, ensure that this descriptor is the first address (word address).
- **Reserved**: Configure to 0x0.

10.1.5 Register List

Module Name	Base Address
CE_NS (Non-Security CE)	0x03040000

Module Name	Base Address
CE_S (Security CE)	0x03040800

Register Name	Offset	Description
CE_TDA	0x0000	Task Descriptor Address
CE_ICR	0x0008	Interrupt Control Register
CE_ISR	0x000C	Interrupt Status Register
CE_TLR	0x0010	Task Load Register
CE_TSR	0x0014	Task Status Register
CE_ESR	0x0018	Error Status Register
CE_CSA	0x0024	DMA Current Source Address
CE_CDA	0x0028	DMA Current Destination Address
CE_TPR	0x002C	Throughput Register

10.1.6 Register Description

10.1.6.1 0x0000 CE Task Descriptor Address Register (Default Value: 0x0000_0000)

Offset: 0x0000			Register Name: CE_TDA
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	Task Descriptor Address Configure as the first address of the descriptor structure.

10.1.6.2 0x0008 CE Interrupt Control Register (Default Value: 0x0000_0000)

Offset: 0x0008			Register Name: CE_ICR
Bit	Read/Write	Default/Hex	Description
31:4	/	/	/
3:0	R/W	0x0	Task Channel3–0 Interrupt Enable 0: Disable 1: Enable

10.1.6.3 0x000C CE Interrupt Status Register (Default Value: 0x0000_0000)

Offset: 0x000C			**Register Name: CE_ISR**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:4	/	/	/
3:0	R/W1C	0x0	Task Channel3–0 End Pending 0: Not finished 1: Finished It indicates whether task is completed. Write the corresponding channel bit of the register to clear the end flag. When the last task in the task chain ends, the operation of the task chain will end normally. If the task fails in the middle, the task chain will be aborted. The CE_ISR register will be automatically updated when it ends normally or aborts abnormally. And it is determined whether to generate an interrupt signal according to the IE configuration (bit31) of Common Control when the current task ends or aborts. If using interrupt, after receiving the interrupt, read the corresponding channel bit of CE_ISR to judge whether it ends successfully or stops failure. If not using interrupt, the CE_ISR status register can be continuously queried for the channel bit until the successful end flag is set or the failure stop flag is set. Write the corresponding channel bit of the register to clear the end flag. If it fails to stop, you can read the error code on the channel corresponding to the **CE_ESR** register.

10.1.6.4 0x0010 CE Task Load Register (Default Value: 0x0000_0000)

Offset: 0x0010			**Register Name: CE_TLR**
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/W	0x0	Task Load When set, the CE can load the descriptor of task if the task FIFO is not full.

10.1.6.5 0x0014 CE Task Status Register (Default Value: 0x0000_0000)

Offset: 0x0014			Register Name: CE_TSR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:2	/	/	/
1:0	R	0x0	Running Channel Number 00: Task channel0 01: Task channel1 10: Task channel2 11: Task channel3

10.1.6.6 0x0018 CE Error Status Register (Default Value: 0x0000_0000)

Offset: 0x0018			Register Name: CE_ESR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:12	R/W1C	0x0	Task Channel3 Error Type xxx1: Algorithm not support xx1x: Data length error x1xx: keysram access error for AES 1xxx: Reserved
11:8	R/W1C	0x0	Task Channel2 Error Type xxx1: Algorithm not support xx1x: Data length error x1xx: keysram access error for AES 1xxx: Reserved
7:4	R/W1C	0x0	Task Channel1 Error Type xxx1: Algorithm not support xx1x: Data length error x1xx: keysram access error for AES 1xxx: Reserved
3:0	R/W1C	0x0	Task Channel0 Error Type xxx1: Algorithm not support xx1x: Data length error x1xx: keysram access error for AES 1xxx: Reserved

10.1.6.7 0x0024 CE Current Source Address Register (Default Value: 0x0000_0000)

Offset: 0x0024			Register Name: CE_CSA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	CUR_SRC_ADDR Current source address

10.1.6.8 0x0028 CE Current Destination Address Register (Default Value: 0x0000_0000)

Offset: 0x0028			Register Name: CE_CDA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	CUR_DST_ADDR Current destination address

10.1.6.9 0x002C CE Throughput Register (Default Value: 0x0000_0000)

Offset: 0x002C			Register Name: CE_TPR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/WC	0x0	TP_NUM It indicates the throughput writing to this register at last time. Writing to this register will clear it to 0.

10.2 Security ID

The Security ID (SID) is used to program and read keys which include chip ID, thermal sensor, HASH code, and so on.

The SID module has the following features:

- 2 Kbits electrical fuse (eFuse)
- Backup eFuse information by using SID_SRAM
- A fuse only can program one time
- Burning the key to the SID
- Reading the key use status in the SID
- Loading the key to the CE

⚠ **CAUTION**

Before performing the burning operation, ensure that the power supply of the eFuse power pin is stable. After the burning operation is completed, cancel the power supply of the eFuse power pin.

Appendix: Glossary

The following table contains acronyms and abbreviations used in this document.

Term	Meaning
A	
ADC	Analog-to-Digital Converter
AE	Automatic Exposure
AEC	Audio Echo Cancellation
AES	Advanced Encryption Standard
AF	Automatic Focus
AGC	Automatic Gain Control
AHB	AMBA High-Speed Bus
ALC	Automatic Level Control
ANR	Active Noise Reduction
APB	Advanced Peripheral Bus
ARM	Advanced RISC Machine
AVS	Audio Video Synchronization
AWB	Automatic White Balance
B	
BROM	Boot ROM
C	
CIR	Consumer Infrared
CMOS	Complementary Metal-Oxide Semiconductor
CP15	Coprocessor 15
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CSI	Camera Serial Interface
CVBS	Composite Video Broadcast Signal
D	
DDR	Double Data Rate
DES	Data Encryption Standard
DLL	Delay-Locked Loop
DMA	Direct Memory Access
DRC	Dynamic Range Compression
DVFS	Dynamic Voltage and Frequency Scaling
E	
ECC	Error Correction Code
eFuse	Electrical Fuse, A one-time programmable memory
EHCI	Enhanced Host Controller Interface
eMMC	Embedded Multi-Media Card
ESD	Electrostatic Discharge
F	
FBGA	Fine Pitch Ball Grid Array

Term	Meaning
FEL	Fireware Exchange Launch
FIFO	First In First Out
G	
GPIO	General Purpose Input Output
I	
I2C	Inter Integrated Circuit
I2S	Inter IC Sound
ISP	Image Signal Processor
J	
JEDEC	Joint Electron Device Engineering Council
JPEG	Joint Photographic Experts Group
JTAG	Joint Test Action Group
L	
LCD	Liquid-Crystal Display
LFBGA	Low Profile Fine Pitch Ball Grid Array
LSB	Least Significant Bit
LVDS	Low Voltage Differential Signaling
M	
MAC	Media Access Control
MIC	Microphone
MIPI	Mobile Industry Processor Interface
MLC	Multi-Level Cell
MMC	Multimedia Card
MPEG	Motion Pictures Expert Group
MSB	Most Significant Bit
N	
N/A	Not Application
NMI	Non Maskable Interrupt
NTSC	National Television Standards Committee
NVM	Non Volatile Storage Medium
O	
OHCI	Open Host Controller Interface
OTP	One Time Programmable
OWA	One Wire Audio
P	
PAL	Phase Alternating Line
PCM	Pulse Code Modulation
PHY	Physical Layer Controller
PID	Packet Identifier
PLL	Phase-Locked Loop
POR	Power-On Reset
PRCM	Power Reset Clock Management
PWM	Pulse Width Modulation

Term	**Meaning**
R	
R	Read only/non-Write
RGB	Read Green Blue
RGMII	Reduced Gigabit Media Independent Interface
RMII	Reduced Media Independent Interface
ROM	Read Only Memory
RSA	Rivest-Shamir-Adleman
RTC	Real Time Clock
S	
SAR	Successive Approximation Register
SD	Secure Digital
SDIO	Secure Digital Input Output
SDK	Software Development Kit
SDRAM	Synchronous Dynamic Random Access Memory
SDXC	Secure Digital Extended Capacity
SLC	Single-Level Cell
SoC	System on Chip
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
T	
TDES	Triple Data Encryption Standard
TWI	Two Wire Interface
U	
UART	Universal Asynchronous Receiver Transmitter
UDF	Undefined
USB DRD	Universal Serial Bus Dual Role Device
UTMI	USB2.0 Transceiver Macrocell Interface

Trademarks and Permissions

Allwinner and the Allwinner logo (incomplete enumeration) are trademarks of Allwinner Technology Co.,Ltd. All other trademarks, trade names, product or service names mentioned in this document are the property of their respective owners.

Important Notice and Disclaimer

The purchased products, services and features are stipulated by the contract made between Allwinner Technology Co.,Ltd ("Allwinner") and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Please read the terms and conditions of the contract and relevant instructions carefully before using, and follow the instructions in this documentation strictly. Allwinner assumes no responsibility for the consequences of improper use (including but not limited to overvoltage, overclock, or excessive temperature).

The information in this document is provided just as a reference or typical applications, and is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents. Allwinner is not responsible for any damage (including but not limited to indirect, incidental or special loss) or any infringement of third party rights arising from the use of this document. All statements, information, and recommendations in this document do not constitute a warranty or commitment of any kind, express or implied.

No license is granted by Allwinner herein express or implied or otherwise to any patent or intellectual property of Allwinner. Third party licences may be required to implement the solution/product. Customers shall be solely responsible to obtain all appropriately required third party licences. Allwinner shall not be liable for any licence fee or royalty due in respect of any required third party licence. Allwinner shall have no warranty, indemnity or other obligations with respect to third party licences.