Secure Encryption User Manual

APM32F407/417xExG

Arm® Cortex®-M4 core-based 32-bit MCU

Version: V1.0

Table of Contents

1 State Encryption

1.1 SM3

1.1.1 Introduction

SM3_H is a high-performance IP core supporting AHB interface, and is widely applied in digital signature and message authentication. SM3 is the encryption hash function standard adopted by our government. The hash function is an iterative one-way function and it can process messages to generate a condensed representation called message digest.

1.1.2 Functional description

（1） Support hardware padding and software padding.

The SM3 calculation module calculates 1 data block at a time (1 data block is 512 bits). If the message length is greater than 512 bits, the message needs to be divided into multiple data blocks for processing. According to the provisions of SM3 algorithm, one 1, several zero and the message length must be padded at the end of the message. When the message length is less than or equal to 512 bits (one data block) and the last data block, the data block needs to be padded.

If the user selects hardware padding (by configuring CTRL), the hardware will automatically complete the padding operation. If the user selects software padding, the user needs to manually complete the padding operation and write it to the DIN register

（2） Support hardware initialization and software initialization.

The initial value needs to be configured before the first data block operation. If the user selects hardware initialization, the hardware will automatically fill in the initial value of the first data block. If the user selects software initialization, it is required to manually write the initial value required by the SM3 algorithm into the IV register of the first data block.

The initial value of a data block is the calculation result of the previous data block. According to the requirements of SM3 algorithm, the initial value of the first data block is constant. If the message length is greater than 512 bits, the initial value of the second block is the calculation result of the first block.

（3） Support the big-endian mode and little-endian mode of input message, initial value and message digest. The default mode is little-endian mode.

（4） The message length only supports 8-bit multiples, and the maximum message length is 2^{64}-1 bits.

1.1.3 Register address mapping

Table 1 Register Address Mapping

Register name	Description	Offset address
REV	Version information register	0x00
CTRL	Control register	0x04
STAT	State register	0x08
DILH	Input information length high 32-bit register	0x0C
DILL	Input information length low 32-bit register	0x10
DIN0	Input information register	0x14
DIN1	Input information register	0x18
DIN2	Input information register	0x1C
DIN3	Input information register	0x20
DIN4	Input information register	0x24
DIN5	Input information register	0x28
DIN6	Input information register	0x2C
DIN7	Input information register	0x30
DIN8	Input information register	0x34
DIN9	Input information register	0x38
DIN10	Input information register	0x3C
DIN11	Input information register	0x40
DIN12	Input information register	0x44
DIN13	Input information register	0x48
DIN14	Input information register	0x4C
DIN15	Input information register	0x50
IV0	Input initial value register	0x54
IV1	Input initial value register	0x58
IV2	Input initial value register	0x5C
IV3	Input initial value register	0x60
IV4	Input initial value register	0x64
IV5	Input initial value register	0x68
IV6	Input initial value register	0x6C
IV7	Input initial value register	0x70
DOUT0	Output information digest register	0x74
DOUT1	Output information digest register	0x78

Register name	Description	Offset address
DOUT2	Output information digest register	0x7C
DOUT3	Output information digest register	0x80
DOUT4	Output information digest register	0x84
DOUT5	Output information digest register	0x88
DOUT6	Output information digest register	0x8C
DOUT7	Output information digest register	0x90

1.1.4 Register functional description

1.1.4.1 Version information register (REV)

Offset address: 0x00

Reset value: 0x00 0000

Field	Name	R/W	Description
7:0	MIN	R	Minor Revision
15:8	MID	R	Middle Revision
23:16	MAJ	R	Major Revision
31:24	Reserved		

1.1.4.2 Control register (CTRL)

Offset address: 0x04

Reset value: 0x00 0000

Field	Name	R/W	Description
0	OP_START	R/W	Start a New Operation After all other registers are written, OP_START needs to be set to 1, and this bit will be cleared automatically in the next cycle.
1	IE	R/W	Interrupt Enable Set IE bit to 1, which indicates that it is allowed to generate sensitive interrupt level. An interrupt signal will be generated when the data calculation is completed. The interrupt can be cleared by reading STAT.
2	INT_MODE	R/W	Hardware initialization Mode 0: Automatic hardware initialization mode. If the user selects hardware initialization, INI_MODE needs to be configured for the first data block. 1: Non-initialization mode. Other data blocks (except the first one) need to be configured as uninitialized.
3	PAD_MODE	R/W	Hardware Padding Mode 0: Hardware padding mode. If the user selects hardware padding, PAD_MODE needs to be configured for the last data block. 1: Non-padding mode. Other data blocks (except the last one) need to be configured as non-padding. If the user selects software padding, PAD_MODE of all data blocks needs to be configured to 1.

Field	Name	R/W	Description
4	ENDIAN	R/W	Endian Mode In Word 0: Little-endian mode 1: Big-endian mode
31:5	Reserved		

1.1.4.3 State register (STAT)

Offset address: 0x08

Reset value: 0x00 0000

Field	Name	R/W	Description
0	OP_DONE	R/W	Operation is Done OP_DONE will be set to 1 when the operation is completed. Write 1 to OP_DONE, and when OP_DONE is 1, all bits of STAT will be cleared.
1	BUSY	R	The Engine is Busy When the engine is busy, BUSY bit will be set to 1. When BUSY bit is set to 1, the register configuration or start a new operation will be ignored.
31:2	Reserved		

1.1.4.4 Input information length high 32-bit register (DILH)

Offset address: 0x0C

Reset value: 0x00 0000

Field	Name	R/W	Description
31:0	DILH	W	The input message length is greater than 32 bits, and the last block must be configured with the message length

1.1.4.5 Input information length low 32-bit register (DILL)

Offset address: 0x10

Reset value: 0x00 0000

Field	Name	R/W	Description
31:0	DILL	W	The input message length is less than 32 bits, and the last block must be configured with the message length

1.1.4.6 Input message register (DIN0…1)

Offset address: 0x14-0x50

Reset value: 0x0000 0000

DIN0 (offset address: 0x14)

Field	Name	R/W	Description
31:0	DIN0	R/W	Write the first word of the message to DIN0.

DIN1 (offset address: 0x18)

Field	Name	R/W	Description
31:0	DIN1	R/W	Write the second word of the message to DIN1.

DIN2 (offset address: 0x1C)

Field	Name	R/W	Description
31:0	DIN2	R/W	Write the third word of the message to DIN2.

DIN3 (offset address: 0x20)

Field	Name	R/W	Description
31:0	DIN3	R/W	Write the fourth word of the message to DIN3.

DIN4 (offset address: 0x24)

Field	Name	R/W	Description
31:0	DIN4	R/W	Write the fifth word of the message to DIN4.

DIN5 (offset address: 0x28)

Field	Name	R/W	Description
31:0	DIN5	R/W	Write the sixth word of the message to DIN5.

DIN6 (offset address: 0x2C)

Field	Name	R/W	Description
31:0	DIN6	R/W	Write the seventh word of the message to DIN6.

DIN7 (offset address: 0x30)

Field	Name	R/W	Description
31:0	DIN7	R/W	Write the eighth word of the message to DIN7.

DIN8 (offset address: 0x34)

Field	Name	R/W	Description
31:0	DIN8	R/W	Write the ninth word of the message to DIN8.

DIN9 (offset address: 0x38)

Field	Name	R/W	Description
31:0	DIN9	R/W	Write the tenth word of the message to DIN9.

DIN10 (offset address: 0x3C)

Field	Name	R/W	Description
31:0	DIN10	R/W	Write the eleventh word of the message to DIN10.

DIN11 (offset address: 0x40)

Field	Name	R/W	Description
31:0	DIN11	R/W	Write the twelfth word of the message to DIN11.

DIN12 (offset address: 0x44)

Field	Name	R/W	Description
31:0	DIN12	R/W	Write the thirteenth word of the message to DIN12.

DIN13 (offset address: 0x48)

Field	Name	R/W	Description
31:0	DIN13	R/W	Write the fourteenth word of the message to DIN13.

DIN14 (offset address: 0x4C)

Field	Name	R/W	Description
31:0	DIN14	R/W	Write the fifteenth word of the message to DIN14.

DIN15 (offset address: 0x50)

Field	Name	R/W	Description
31:0	DIN15	R/W	Write the sixteenth word of the message to DIN15.

1.1.4.7 **Input initial value register (IV0…7)**

Offset address: 0x54-0x70

Reset value: 0x0000 0000

IV0 (offset address: 0x54)

Field	Name	R/W	Description
31:0	IV0	R/W	Input the first word of the initial value.

IV1 (offset address: 0x58)

Field	Name	R/W	Description
31:0	IV1	R/W	Input the second word of the initial value.

IV2 (offset address: 0x5C)

Field	Name	R/W	Description
31:0	IV2	R/W	Input the third word of the initial value.

IV3 (offset address: 0x60)

Field	Name	R/W	Description
31:0	IV3	R/W	Input the fourth word of the initial value.

IV4 (offset address: 0x64)

Field	Name	R/W	Description
31:0	IV4	R/W	Input the fifth word of the initial value.

IV5 (offset address: 0x68)

Field	Name	R/W	Description
31:0	IV5	R/W	Input the sixth word of the initial value.

IV6 (offset address: 0x6C)

Field	Name	R/W	Description
31:0	IV6	R/W	Input the seventh word of the initial value.

IV7 (offset address: 0x70)

Field	Name	R/W	Description
31:0	IV7	R/W	Input the eighth word of the initial value.

1.1.4.8 **Output information digest register (DOUT0…7)**

Offset address: 0x74-0x90

Reset value: 0x0000 0000

DOUT0 (offset address: 0x74)

Field	Name	R/W	Description
31:0	DOUT0	R/W	Output the first word of information digest.

DOUT1 (offset address: 0x78)

Field	Name	R/W	Description
31:0	DOUT1	R/W	Output the second word of information digest.

DOUT2 (offset address: 0x7C)

Field	Name	R/W	Description
31:0	DOUT2	R/W	Output the third word of information digest.

DOUT3 (offset address: 0x80)

Field	Name	R/W	Description
31:0	DOUT3	R/W	Output the fourth word of information digest.

DOUT4 (offset address: 0x84)

Field	Name	R/W	Description
31:0	DOUT4	R/W	Output the fifth word of information digest.

DOUT5 (offset address: 0x88)

Field	Name	R/W	Description
31:0	DOUT5	R/W	Output the sixth word of information digest.

DOUT6 (offset address: 0x8C)

Field	Name	R/W	Description
31:0	DOUT6	R/W	Output the seventh word of information digest.

DOUT7 (offset address: 0x90)

Field	Name	R/W	Description
31:0	DOUT7	R/W	Output the eighth word of information digest.

1.2 SM4

1.2.1 Introduction

SM4_H is a high-performance IP core supporting AHB interface. It is a symmetric encryption algorithm released in January 2006, and has been widely applied in China's wireless LAN WAPI (wired authentication and privacy infrastructure). The IP core implements SM4 standard algorithm and supports encryption and decryption in ECB and CBC modes. SM4 algorithm is a symmetric block cipher of 128-bit input data and key. In order to improve the operation speed, the data path module uses 8 s boxes for encryption and decryption operation and key expansion at the same time. It only takes 32 cycles to complete a round of encryption and decryption operation.

1.2.2 Functional description

（1） Support ECB and CBC modes

In ECB mode, each data block is encrypted/decrypted with the same key. The encryption in CBC mode will take the XOR result of the current plain text and the final cipher text as the input plain text. The decryption in CBC mode will take the XOR result of the current decryption output and the final cipher text as the output plain text. If the user selects CBC mode, before each operation, the initial value must be written into the IV register according to the above requirements.

（2） Support high-speed encryption and decryption operation (32 cycles per round)

The SM4 calculation module calculates 1 data block at a time (1 data block is 128 bits). If the user needs to process multiple data blocks, the result shall be read after each processing and be written to the data block before the next calculation.

（3） Support little-endian mode

For example, SM4 algorithm standard reference value (hexadecimal value):

Plain text: 01234567 89abcdef fedcba98 76543210

DIN0：01234567 DIN1：89abcdef DIN2：fedcba98 DIN3：76543210

Key：01234567 89abcdef fedcba98 76543210

KEY0:01234567 KEY1：89abcdef KEY2：fedcba98 KEY3：76543210

Cipher text: 681edf34 d206965e 86b3e94f 536e4246

DOUT0：681edf34 DOUT1：d206965e DOUT2：86b3e94f DOUT3：536e4246

1.2.3 Register address mapping

Table 2 Register Address Mapping

Register name	Description	Offset address
REV	Version information register	0x00
CTRL	Control register	0x04
STAT	State register	0x08
KEY0	User key register	0x0C
KEY1	User key register	0x10
KEY2	User key register	0x14
KEY3	User key register	0x18
DIN0	Input data register	0x1C

Register name	Description	Offset address
DIN1	Input data register	0x20
DIN2	Input data register	0x24
DIN3	Input data register	0x28
DOUT0	Output data register	0x2C
DOUT1	Output data register	0x30
DOUT2	Output data register	0x34
DOUT3	Output data register	0x38
IV0	Initialization vector register	0x3C
IV1	Initialization vector register	0x40
IV2	Initialization vector register	0x44
IV3	Initialization vector register	0x48

1.2.4 Register functional description

1.2.4.1 Version information register (REV)

Offset address: 0x00

Field	Name	R/W	Description	Reset value
0	REV0	R	Version Record	1
1	REV1	R	Version Record	0
31:2	Reserved			0

1.2.4.2 Control register (CTRL)

Offset address: 0x04

Reset value: 0x00 0000

Field	Name	R/W	Description
0	OP_START	R/W	Start a New Operation After all other registers are written, OP_START needs to be set to 1, and this bit will be cleared automatically in the next cycle.
1	IE	R/W	Interrupt Enable If the IE bit is set to 1, the sensitive interrupt level is allowed.An interrupt signal is generated when the data calculation is complete.Clear interrupts by reading STAT.
2	ENC	R/W	Encryption/Decryption Operation 0: Decryption 1: Encryption
3	MODE	R/W	Hardware initialization Mode 0: ECB mode 1: CBC mode
31:4	Reserved		

1.2.4.3 **State register (STAT)**

Offset address: 0x08

Reset value: 0x00 0000

Field	Name	R/W	Description
0	OP_DONE	R/W	Operation is Done OP_DONE will be set to 1 when the operation is completed. Write 1 to OP_DONE, and when OP_DONE is 1, all bits of STAT will be cleared.
1	BUSY	R	The Engine is Busy When the engine is busy, BUSY bit will be set to 1. When BUSY bit is set to 1, the register configuration or start a new operation will be ignored.
31:2	Reserved		

1.2.4.4 **User key register (KEY0…3)**

Offset address: 0x0C-0x18

Reset value: 0x0000 0000

KEY0 (offset address: 0x0C)

Field	Name	R/W	Description
31:0	KEY0	W	Input the first word of the key data. The value of KEY register will not be updated after it is written. For operation of multiple data blocks, it is unnecessary to write the same key repeatedly at each time.

KEY1 (offset address: 0x10)

Field	Name	R/W	Description
31:0	KEY1	W	Input the second word of the key data.

KEY2 (offset address: 0x14)

Field	Name	R/W	Description
31:0	KEY2	W	Input the third word of the key data.

KEY3 (offset address: 0x18)

Field	Name	R/W	Description
31:0	KEY3	W	Input the fourth word of the key data.

1.2.4.5 **Input data register (DIN0…3)**

Offset address: 0x1C-0x28

Reset value: 0x0000 0000

DIN0 (offset address: 0x1C)

Field	Name	R/W	Description
31:0	DIN0	W	Input the first word of the data.

DIN1 (offset address: 0x20)

Field	Name	R/W	Description
31:0	DIN1	W	Input the second word of the data.

DIN2 (offset address: 0x24)

Field	Name	R/W	Description
31:0	DIN2	W	Input the third word of the data.

DIN3 (offset address: 0x28)

Field	Name	R/W	Description
31:0	DIN3	W	Input the fourth word of the data.

1.2.4.6 **Output data register (DOUT0…3)**

Offset address: 0x2C-0x38

Reset value: 0x0000 0000

DOUT0 (offset address: 0x2C)

Field	Name	R/W	Description
31:0	DOUT0	R	Ouput the first word of the data.

DOUT1 (offset address: 0x30)

Field	Name	R/W	Description
31:0	DOUT1	R	Output the second word of the data.

DOUT2 (offset address: 0x34)

Field	Name	R/W	Description
31:0	DOUT2	R	Output the third word of data.

DOUT3 (offset address: 0x38)

Field	Name	R/W	Description
31:0	DOUT3	R	Output the fourth word of the data.

1.2.4.7 **Initialization vector register (IV0…3)**

Offset address: 0x3C-0x48

Reset value: 0x0000 0000

IV0 (offset address: 0x3C)

Field	Name	R/W	Description
31:0	IV0	W	The first word of initialization vector. IV register cannot be updated automatically. If multiple data blocks are encrypted and decrypted in CBC mode, the initial value must be written to the IV register before each calculation.

IV1 (offset address: 0x40)

Field	Name	R/W	Description
31:0	IV1	W	The second word of initialization vector.

IV2 (offset address: 0x44)

Field	Name	R/W	Description
31:0	IV2	W	The third word of initialization vector.

IV3 (offset address: 0x48)

Field	Name	R/W	Description
31:0	IV3	W	The fourth word of initialization vector.

2 BN

2.1 Introduction

The BN module is a low-cost accelerator of RSA and ECC algorithms. It realizes almost all big number operations in finite field and the point operations on elliptic curves with K not equal to 2 or 3 characteristic p defined by Weierstrass equation. RSA encryption and decryption adopts modular exponentiation operation (MEXP). The elliptic curve point multiplication (ECPM), elliptic curve point addition (ECPA) and elliptic curve point verification (ECPV) are specific to ECC cryptosystems, such as SM2. The public key is generated in RSA and the internal parameters are generated in SM2 to realize modular inversion (MINV). Rabin-Miller prime number test is conducted for the numbers in the process of prime number generation. It also supports modular addition (MADD), modular subtraction (MSUB), modular multiplication (MMUL), addition (add), subtraction (sub), and most significant bit detection (MSBD). Currently it also supports pre-composing user-defined configurable defense countermeasures.

2.2 Main characteristics

（1） Support any elliptic curve defined on GF(p), formula: $y^2=x^3+ax+b$

（2） Support elliptic curve point operations, including elliptic curve point addition (ECPA), elliptic curve point multiplication (ECPM) and elliptic curve point verification (ECPV). Check whether the point is on the elliptic curve from 192 bits to 256 bits

（3） Support Modular Exponentiation (MEXP) for RSA, from 192 bits to 2048 bits

（4） Support big number MSB detection (MSBD), addition (ADD), and subtraction (SUB), from 192 bits to 2048 bits

（5） Support big number finite field operation up to 2048 bits, including modular addition (MADD), modular subtraction (MSUB), modular multiplication (MMUL), modular inversion (MINV) and Rabin-Miller prime number test (RB)

（6） Support interrupt and poll software programming. Interrupts and exceptions can be masked

（7） Support counterattack countermeasures of SM2, such as inserting random delay before and after ADD/MADD/SUB/MSUB, opening the multiplier to generate power noise, etc. In ECPM, insert the coordinate randomization and point and scale double check when calculating the point multiplication and the attacker asserts when an error occurs

（8） Support pre-composing user-defined configurable 64-bit or 32-bit processing, or dual-engine processor ECC.

（9） The software can restructure ECC point multiplication algorithm.

（10） If the defense feature is enabled, the input data register (address offset starts from 0x40) can write only, and some additional software can configure registers to realize real random data input.

（11） Support little-endian mode

2.3 Functional description

Table 3 BN Algorithm Description

Term	Function	Description
RB	R = RB (M, sp)	Rabin-Miller probability prime number test is conducted to check whether M is a prime number. The larger the sp is, the longer the calculation time is, the more reliable the test results is. The output is stored in STAT_REG register M bit length: 32×n, n is from 6 to 64
ECPM	R(x, y) = [k] P(x, y)	Elliptic curve point multiplication. EC parameters p, a, and b should also be input. P bit length: 32×n, n is from 6 to 8
ECPA	R(x, y) = P(x, y)+S(x, y)	Elliptic curve point addition. EC parameters p, a and b should also be input. When P and S are the same point, the core will automatically perform ECPD[2] operation. P bit length: 32×n, n is from 6 to 8
ECPV	Verify whether the point P (x, y) is on the elliptic curve	Elliptic curve point verifiction. EC parameters p, a, and b should also be input. The output is stored in STAT_REG register. P bit length: 32×n, n is from 6 to 8
MEXP	$R=A^B \bmod M$	Modular exponentiation operation A<B, B<M, M is an odd number M bit length: 32×n, n is from 6 to 64
MMUL	R=(A×B) mod M	Modular multiplication operation A<B, B<M, M is an odd number M bit length: 32×n, n is from 6 to 64
MINV	$R=A^{-1} \bmod M$	Modular inversion operation A<M，gcd（M,A）=1 M bit length: 32×n, n is from 6 to 64
ADD	(cb, R) = (A+B)	Addition The length is defined by PWID: 32×n, n is from 6 to 64
SUB	(cb, R) = (A−B)	Subtraction If A<B, cb=1, R=B-A　else cb=0, R=A-B The length is defined by PWID: 32×n, n is from 6 to 64
MADD	R=(A+B) mod M	Modular Addition A<M, B<M. M bit length: 32×n, n is from 6 to 64

Term	Function	Description
MSUB	R=(A-B) mod M	Modular Subtraction A<M, B<M. M bit length: 32×n, n is from 6 to 64
MSBD	BLEN=bit_size_of(M)	Get bit length of operand M M bit length: 32×n, n is from 6 to 64 eg. BLEN=bit_size_of(0xF)=3

Note:

(1) In ECC[1] mode, p is prime number, and in RSA mode, M is the product of two large prime numbers.

(2) ECC: Elliptic Curve Cryptography is an approach to public-key cryptography based on the algebraic structure of elliptic curve over finite fields. ECC requires smaller keys compared to non-EC cryptography to provide equivalent security. See standard GM/T0003.1-2012 for details

(3) ECPD: Elliptic Curve Point Doubling is the multiplication of points on an elliptic curve with an integer.

2.4 Register address mapping

Table 4 Register Address Mapping

Register name	Description	Offset address
REV_REG	Version information register	0x00
CTRL_REG	Control register	0x04
STAT_REG	State register	0x08
WID_REG	Operation width register	0x0C
P_REG	Modulus P/M register	0x40
A_REG	Operand A or ECC parameter A register	0x44
B_REG	Operand B or ECC parameter B register	0x48
K_REG	ECC key register	0x4C
PX_REG	ECC PX register	0x50
PY_REG	ECC PY register	0x54
SX_REG	ECC SX register	0x58
SY_REG	ECC SY register	0x5C
RX_REG	BN/ECC result register	0x60
RY_REG	ECC result RY register	0x64
RND_REG	Random number register	0x68

2.5 Register functional description

2.5.1 Version information register (REV_REG)

Offset address: 0x00

Reset value: 0x0003 0100

Field	Name	R/W	Description	Reset value
7:0	MIN	R	Minor Revision Indicate non-RTL change	0
15:8	MID	R	Middle Revision Indicate RTL change invisible to firmware	1
23:16	MAJ	R	Major Revision Indicate RTL change visible to firmware	3
31:24	Reserved			0

2.5.2 Control register (CTRL_REG)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	OP_START	R/W	Start a New Operation After all other registers are written, OP_START needs to be set to 1, and this bit will be cleared automatically in the next cycle.
3:1	SP	R/W	Security Parameter sp=0 indicates 8 when operating RB.
7:4	OP_SELECT	R/W	Calculation Select 0000：ADD 0001：MADD 0010-0011: Reserved 0100：SUB 0101：MSUB 0110-0111: Reserved 1000：RB 1001：MSBD 1010：MEXP 1011：ECPM 1100：ECPA 1101：ECPV 1110：MMUL 1111：MINV
8	IE	R/W	Interrupt Enable Set IE bit to 1 to generate an interrupt. Clear the interrupt by reading STAT_REG.
31:9	Reserved		

2.5.3 State register (STAT_REG)

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	OP_DONE	RC_W1	Operation is Done OP_DONE will be set to 1 when the operation is completed. Write 1 to OP_DONE, and when OP_DONE is 1, all bits of STAT will be cleared.
1	BUSY	R/W	The Engine is Busy When the engine is busy, BUSY bit will be set to 1. When BUSY bit is set to 1, the register configuration or start a new operation will be ignored.
2	CB	R/W	Carry out or Borrow in When the last highest word is carried out or borrowed in ADD or SUB operation, the CB will read 1.
3	TR	R/W	Test Result In RB operation, M is a prime number or when the midpoint of ECPV operation is on the elliptic curve, read 1.
4	INF	R/W	Infinity Point The result of ECPM or ECPA is infinity point.
5	EVEN	R/W	Modulo Even P/M is even at the time of MINV/MMUL/MEXP/ECPV/ECPA/ECPM.
6	ZERO	R/W	B/K/M Zero In ECPM, k=0; in MEXP, B=0; in MSBD/MMUL/MEXP/ECPV/ECPA/ECPM, M=0.
7	ATTACKED	R/W	Attacked When calculating [k]P in ECPM operation, assert whether the input point P is not on the elliptic curve or the result point is not on the elliptic curve because an error occurs in SRAM or k is modified.
23:8	BLEN	R/W	Bit length Bit length (-1) of M in MSBD operation.
31:24	Reserved		

2.5.4 Operation width register (WID_REG)

Offset address: 0x0C

Reset value: 0x0000 0000

Field	Name	R/W	Description
6:0	P_WID	R/W	Width of 32-bit word field, 6≤P_WID≤8 in ECC operation, 6≤P_WID≤64 in other operations. All data registers except K_REG are defined by P_WID.
7	Reserved		
11:8	K_WID	R/W	The ECP key width is measured by 32-bit word, 0<K_WID≤15, and in ECPM, only _REG is defined by K_WID.
31:12	Reserved		

2.5.5 Modulus P/M register (P_REG)

Offset address: 0x40

Field	Name	R/W	Description
31:0	P_REG	R/W	Write the modulus P/M to the IP from the least significant word to the most significant word. Access will respond only when BUSY is 0.

2.5.6 Operand A or ECC parameter A register (A_REG)

Offset address: 0x44

Field	Name	R/W	Description
31:0	A_REG	R/W	Write the operand A or ECC parameter A to the IP from the least significant word to the most significant word. Access will respond only when BUSY is 0.

2.5.7 Operand B or ECC parameter B register (B_REG)

Offset address: 0x48

Field	Name	R/W	Description
31:0	B_REG	R/W	Write the operand B or ECC parameter B to the IP from the least significant word to the most significant word. Access will respond only when BUSY is 0.

2.5.8 ECC key register (K_REG)

Offset address: 0x4C

Field	Name	R/W	Description
31:0	K_REG	R/W	Write the ECC key to the IP from the least significant word to the most significant word. Access will respond only when BUSY is 0.

2.5.9 ECC PX register (PX_REG)

Offset address: 0x50

Field	Name	R/W	Description
31:0	PX_REG	R/W	Write the x coordinates of the point P to the IP from the least significant word to the most significant word, and the same number of words must be written as P_REG. Access will respond only when BUSY is 0.

2.5.10 ECC PY register (PY_REG)

Offset address: 0x54

Field	Name	R/W	Description
31:0	PY_REG	R/W	Write the y coordinates of the point P to the IP from the least significant word to the most significant word, and the same number of words must be written as P_REG. Access will respond only when BUSY is 0.

2.5.11 ECC SX register (SX_REG)

Offset address: 0x58

Field	Name	R/W	Description
31:0	SX_REG	R/W	Write the x coordinates of the point S to the IP from the least significant word to the most significant word, and the same number of words must be written as P_REG. Access will respond only when BUSY is 0.

2.5.12 ECC SY register (SY_REG)

Offset address: 0x5C

Field	Name	R/W	Description
31:0	SY_REG	R/W	Write the y coordinates of the point S to the IP from the least significant word to the most significant word, and the same number of words must be written as P_REG. Access will respond only when BUSY is 0.

2.5.13 BN/ECC result register (RX_REG)

Offset address: 0x60

Field	Name	R/W	Description
31:0	RX_REG	R	Read the x coordinates of BN result R or ECC result R point from the IP from the least significant word to the most significant word. The data will take effect only when OP_DONE is 1.

2.5.14 ECC result RY register (RY_REG)

Offset address: 0x64

Field	Name	R/W	Description
31:0	RY_REG	R	Read the y coordinates of ECC result of point R from the IP from the least significant word to the most significant word. The data will take effect only when OP_DONE is 1.

2.5.15 Random number register (RND_REG)

Offset address: 0x68

Field	Name	R/W	Description
31:0	RND_REG	W	In ECPM operation, the RND length should be P_WID word or one word (32 bits) in ADD/MADD/SUB/MSUB.

3 Version History

Table 5 Document Version History

Date	Version	Change History
September, 2021	1.0	New