GMI Instruction Set Reference

Shanghai Zhaoxin Semiconductor Co., Ltd.

GMI instruction set include SM2 instruction, SM3 instruction and SM4 instruction.

CPUID flag

Issue CPUID with RAX = 0xC0000001 will return extended feature flags in RDX, of which:

bit[0] - SM2 instruction is present[1]

bit[1] - SM2 instruction is enabled[2]

bit[4] - CCS(Chinese Cryptography Standard)[3] instructions is present, CCS instructions include SM3 instruction and SM4 instruction.

bit[5] - CCS instructions is enabled, CCS instructions include SM3 instruction and SM4 instruction.

Note1: Present bit is used to inform software corresponding instruction is present.

Note2: Enabled bit is used to inform software corresponding instruction is enabled for normal use.

Note3: String CCS(Chinese Cryptography Standard) is only use in this article.

GMI Instructions Prefix Affect

When executing in protected mode or compatibility mode, depending on the settings of the D flag and the operand-size and address-size prefixes, effective operand-size and address-size attributes as the following:

D Flag in Code Segment Descriptor	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**
Operand-Size Prefix 66H	N	N	Y	Y	N	N	Y	Y
Address-Size Prefix 67H	N	Y	N	Y	N	Y	N	Y
Effective Operand Size	16	16	32	32	32	32	16	16
Effective Address Size	16	32	16	32	32	16	32	16

Y：This instruction prefix is present.

N：This instruction prefix is not present

When executing in 64-bit mode, depending on the settings of the L flag and the operand-size and address-size prefixes, effective operand-size and address-size attributes as the following:

L Flag in Code Segment Descriptor	1	1	1	1	1	1	1	1
REX.W Prefix	0	0	0	0	1	1	1	1
Operand-Size Prefix 66H	N	N	Y	Y	N	N	Y	Y
Address-Size Prefix 67H	N	Y	N	Y	N	Y	N	Y
Effective Operand Size	32	32	16	16	64	64	64	64
Effective Address Size	64	32	64	32	64	32	64	32

Y：This instruction prefix is present.

N：This instruction prefix is not present

GMI Instructions Interrupt Handle

Interrupt or exception can be responded to during the SM3/SM4 instructions. The registers and memory contents are modified during SM3/SM4 instructions, these registers and memory status can be saved when interrupt or exception interrupted SM3/SM4 instructions. So it can return to the currently interrupted SM3/SM4 instructions to continue execute according to saved registers and memory status after handling the interrupt or exception.

GMI Instructions Input Data Size Explanation

SM2/SM3/SM4 instructions use CX/ECX/RCX according to actual operation mode, the maximum number of input data is limited by the maximum value that register itself can set. However, the actual number of data that can be processed in a single SM2/SM3/SM4 instruction is also constrained by the physical memory size of the chip and other factors.

GMI Instructions Input or Output Register Explanation

GMI instructions use 16-bit/32-bit/64-bit architecture registers according to actual operation mode, such as using CX/ECX/RCX, register name such as RCX hereinafter is only an example.

GMI Instructions Input Control Word Reserved Bits Explanation

Using Zhaoxin GMI instructions must follow the instructions definition in this manual. Zhaoxin is not responsible for any unexpected behavior caused by setting reserved bits in the input control word of GMI instructions.

1 SM2

The SM2 algorithm is implemented via a single SM2 instruction, identify sub-instructions which include SM2 encrypt、SM2 decrypt、SM2 signature、SM2 verify signature、SM2 key exchange and SM2 preprocess based on the control word.

Encoding: 0xF2 0x0F 0xA6 0xC0

Modes: REAL, VIRTUAL 8086, COMPAT, PROTECT, LONG

Description: The SM2 instruction is implemented in accordance with the GM/T 0003-2012 specification issued by the State Cryptography Administration.

SM2 Control Word define as the following:

Bits[5:0]: 6’b 000001: Encryption

6’b 000010: Decryption

6’b 000100: Signature

6’b 001000: Verify signature

6’b 010000: Key exchange1

6’b 010001: Key exchange2 without hash

6’b 010101: Key exchange2 with hash

6’b 010010: Key exchange3 without hash

6’b 010110: Key exchange3 with hash

6’b 100000: Preprocess1 to calculate hash value Z of user’s identification

6’b 100001: Preprocess2 to calculate hash value e of hash value Z and message M

Bit6: Output state identification. 0 means instruction executes successfully; 1 means instruction execution failed. Except SM2 verify signature.

Other bits are reserved bits.

1.1 SM2 Encrypt

- **Description**

  SM2 encryption means using specified public key to encrypt plaintext and get corresponding ciphertext. The ciphertext can only be successfully decrypted by the corresponding private key.
- **Input Registers**

  RAX Input pointer for plaintext. Assumes ES segment.

  RBX Pointer to the public key for encryption. Assumes ES segment.

  RCX Number of bytes for plaintext to be encrypted.

  RDX Control Word.

  RSI Pointer to 8K-byte scratch space. Assumes ES segment. Need to initial to 0s. The memory must be writable.

  RDI Output pointer for ciphertext. Assumes ES segment. The memory must be writable.
- **Output Registers**

  RAX Incremented by the number of bytes for plaintext.

  RCX Number of bytes for all ciphertext.

  RDX Bit6==0 means instruction executes successfully; Bit6==1 means instruction execution failed. Other bits unchanged.

  RDI Incremented by the number of bytes for ciphertext. The memory will contain ciphertext.

1.2 SM2 Decrypt

- **Description**

  SM2 decryption means using specified private key to decrypt ciphertext and get corresponding plaintext. Can only successfully decrypt the plaintext encrypted by the corresponding public key.
- **Input Registers**

  RAX Input pointer for ciphertext. Assumes ES segment;

  RBX Pointer to the private key for decryption. Assumes ES segment.

RCX Number of bytes for all ciphertext to be decrypted.

RDX Control Word.

RSI Pointer to 8K-byte scratch space. Assumes ES segment. Need to initial to 0s. The memory must be writable.

RDI Output pointer for plaintext. Assumes ES segment. The memory must be writable.

- **Output Registers**

RAX Incremented by the number of bytes of input ciphertext.

RCX Number of bytes for plaintext.

RDX Bit6==0 means instruction executes successfully; Bit6==1 means instruction execution failed. Other bits unchanged.

RDI Incremented by the number of bytes for plaintext. The memory will contain plaintext.

1.3 SM2 Signature

- **Description**

SM2 signature means using specified signer's private key to sign message and get signature. The signature can only be successfully verified by the corresponding public key.

- **Input Registers**

RAX Input pointer for hash value after preprocess2. Assumes ES segment.

RBX Pointer to the private key of signer. Assumes ES segment.

RDX Control Word.

RSI Pointer to 8K-byte scratch space. Assumes ES segment. Need to initial to 0s. The memory must be writable.

RDI Output pointer for signature. Assumes ES segment. The memory must be writable.

- **Output Registers**

RCX Number of bytes for all signature.

RDX Bit6==0 means instruction executes successfully; Bit6==1 means instruction execution failed. Other bits unchanged.

RDI Increment by the number of bytes for signature. The memory will contain signature.

1.4 SM2 Verify Signature

- **Description**

SM2 verify signature means using specified public key of signer to verify signature. Can only successfully verify the signature signed by corresponding private key.

- **Input Registers**

RAX Input pointer for hash value after preprocess2. Assumes ES segment.

RBX Pointer to the public key of signer. Assumes ES segment.

RDX Control Word.

RSI Pointer to 8K-byte scratch space; Assumes ES segment. Need to initial to 0s. The memory must be writable.

RDI Pointer to the signature. Assumes ES segment. The memory buffer of signature need to be same with the length of signature encoded by ASN.1.

- **Output Registers**

  RCX 1 means verify signature pass; 0 means verify signature fail.

1.5 SM2 Key Exchange

1.5.1 SM2 Key Exchange1

- **Description**

  SM2 key exchange1 means produced a sm2 key pair, include private key and public key. This instruction also can be used for initiator user A to generate a temporary SM2 key pair.

- **Input Registers**

  RDX Control Word.

  RSI Pointer to 8K-byte scratch space. Assume ES segment. Need to initial to 0s. The memory must be writable.

  RDI Output pointer for sm2 key pair for private key and public key. Assume ES segment. The memory must be writable.

- **Output Registers**

  RDI The register is unchanged. The memory will contain sm2 key pair.

1.5.2 SM2 Key Exchange2

- **Description**

  SM2 key exchange2 means responder user B to establish a shared key of initiator user A and responder user B, and selectively calculate hash value S_2&S_B according to control word.

- **Input Registers**

  RAX Pointer to input key and identification information. Assumes ES segment. Include the following content:

  User A temporary public key;

  User B private key;

  User B public key;

  User A public key;

  User A bits length of identification and identification content;

  User B bits length of identification and identification content;

  RCX Number of bits for shared key.

  RDX Control Word.

  RSI Pointer to 8K-byte scratch space. Assume ES segment. Need to initial to 0s. The memory must be writable.

RDI Output pointer for shared key and user B temporary public key and selective hash value S_2&S_B. Assume ES segment. The memory must be writable.

- **Output Registers**

RDX Bit6==0 means instruction executes successfully; Bit6==1 means instruction execution failed. Other bits unchanged.

RDI The register is unchanged. The memory will contain shared key and user B temporary public key and selective hash value S_2&S_B.

1.5.3 SM2 Key Exchange3

- **Description**

SM2 key exchange3 means initiator user A to establish a shared key of initiator user A and responder user B and selectively calculate hash value S_1&S_A according to control word.

- **Input Registers**

RAX Pointer to input key and identification information. Assumes ES segment. Include the following:

User A temporary private key;

User A temporary public key;

User B temporary public key;

User B public key;

User A private key;

User A public key;

User A bits length of identification and identification content;

User B bits length of identification and identification content;

RCX Number of bits for shared key.

RDX Control Word.

RSI Pointer to 8K-byte scratch space. Assume ES segment. Need to initial to 0s. The memory must be writable.

RDI Output pointer for shared key and selective hash value S_1&S_A. Assume ES segment. The memory must be writable.

- **Output Registers**

RDX Bit6==0 means instruction executes successfully; Bit6==1 means instruction execution failed. Other bits unchanged.

RDI The register is unchanged. The memory will contain shared key and selective hash value S_1&S_A.

1.6 SM2 Preprocess

1.6.1 SM2 Preprocess1

- **Description**

  SM2 preprocess1 means by using signer's identification and public key to get hash value Z of user's identification.
- **Input Registers**

  RAX Input pointer for user's identification length and identification. Assumes ES segment.

  RBX Pointer to the public key. Assumes ES segment.

  RDX Control Word.

  RSI Pointer to 8K-byte scratch space. Assumes ES segment. Need to initial to 0s. The memory must be writable.

  RDI Output pointer for hash value Z. Assumes ES segment. The memory must be writable.
- **Output Registers**

  RDX Bit6==0 means instruction executes successfully; Bit6==1 means instruction execution failed. Other bits unchanged.

  RDI Incremented by the number of bytes of hash value Z. The memory will contain the result of the hash calculation.

1.6.2 SM2 Preprocess2

- **Description**

  SM2 preprocess2 means use hash value Z and message M to be signed to get hash value e.
- **Input Registers**

  RAX Input pointer for hash value Z of preprocess1 output; Assumes ES segment.

  RBX Pointer to the message M to be signed. Assumes ES segment.

  RCX Number of bytes for message M to be signed.

  RDX Control Word.

  RDI Output pointer for hash value e. Assumes ES segment. The memory must be writable.
- **Output Registers**

  RDI Incremented by the number of bytes for hash value e. The memory will contain the result of the hash calculation.

2 CCS instructions

2.1 SM3

The SM3 algorithm is implemented via a single SM3 instruction.

Encoding: 0xF3 0x0F 0xA6 0xE8

Modes: REAL, VIRTUAL 8086, COMPAT, PROTECT, LONG

Description: The SM3 instruction is implemented in accordance with the GM/T 0004-2012 specification issued by the State Cryptography Administration.

- **Input Registers**

  RAX If RAX == 0 the SM3 instruction performs the padding for input data stream specified in SM3 specification, and RCX means byte counter of input data stream. If RAX == -1 the SM3 instruction does not perform the padding for input data stream, and RCX means the number of 64-byte blocks of input data stream.

  RBX Control word. Bit5==1 means to perform SM3 hash algorithm. Other bits are reserved bits.

  RCX Size of the input data stream:
  If RAX == 0: means in bytes. If RCX == 0, means to perform NOP operation.
  If RAX == -1: means in 64-byte blocks. If RCX == 0, means to perform NOP operation.

  RSI Pointer to the memory for input data stream. Assumes ES segment.

  RDI Pointer to the memory for initial hash constants. Assumes ES segment. The memory must be writable. The constants should be stored as a sequence of big-endian 32-byte integers.

- **Output Registers**

  RAX If the input RAX == 0, RAX will be equal to RCX, if the input RAX == -1, RAX will be unchanged.

  RBX Unchanged.

  RCX If the input RAX== 0, RCX will be unchanged, if the input RAX== -1, RCX will be 0.

  RSI Incremented by the number of bytes input data stream to perform the hash calculation.

  RDI Unchanged. The memory will contain the result of the hash calculation. The result stored as a sequence of big-endian 32-byte integers.

Note: For the case of input RAX is 0, if interrupt happened when performing SM3, the value in RAX will be modified to indicate the number of bytes already processed before handle the interrupt.

2.2 SM4

The SM4 algorithm is implemented via a single SM4 instruction.

Encoding: 0xF3 0x0F 0xA7 0xF0

Modes: REAL, VIRTUAL 8086, COMPAT, PROTECT, LONG

Description: The SM4 instruction is implemented in accordance with the GM/T 0002-2012 specification issued by the State Cryptography Administration.

- **Input Registers**

  RAX Control word.

  Bit[0]: 0 means encryption; 1 means decryption. Ignore this bit0 for counter and output feedback mode, the decryption behavior is the same with encryption.

  Bit[5]: 1 means to perform SM4 algorithm.

  Bit[10:6]: 5’b00001 means using electronic code book mode.
  5’b00010 means using cipher block chaining mode.
  5’b00100 means using cipher feedback mode.

5’b01000 means using output feedback mode.

5’b10000 means using counter mode.

Bit[11]: 1 means support message authentication code(MAC); 0 means don’t support MAC. MAC is only valid in CBC and CFB mode.

Other bits are reserved bits.

RCX Number of 16-byte blocks to encrypt or decrypt. If RCX == 0, means to perform NOP operation.

RBX Pointer to the memory for encryption or decryption key. Assumes ES segment.

RDX Pointer to the memory for initialization vector. Assumes ES segment. The memory must be writable. The high-order 16 bits of the initialization vector is in big-endian as counter, this counter will be incremented by one for every block processed.

RSI Input pointer to the memory for plaintext when encrypting or ciphertext when decrypting. Assumes ES segment.

RDI Output pointer to the memory for ciphertext when encrypting or plaintext when decrypting. Assumes ES segment. The memory must be writable.

- **Output Registers**

RAX Unchanged.

RBX Unchanged.

RCX 0

RSI Incremented by the number of bytes for plaintext when encrypting or ciphertext when decrypting.

RDI Incremented by the number of bytes ciphertext when encrypting or plaintext when decrypting except when the control word specifies MAC. The memory will contain the ciphertext when encrypting or plaintext when decrypting, or the MAC.

Note：For a key loaded from memory for decryption, the equivalent inverse cipher extended key must be provided in the inverse order.