# GMI Reference

Shanghai Zhaoxin Semiconductor Co., Ltd.

## CPUID flag

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

EDX bit[0] - SM2 instruction is present

EDX bit[1] - SM2 instruction is enabled

EDX bit[4] - SM3 and SM4 instructions is present

EDX bit[5] - SM3 and SM4 instructions is enabled

## 1  SM2

**Encoding:** 0xF2 0x0F 0xA6 0xC0

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

**Description:** Implement sm2 algorithm as specified by sm2 specification in 2010.12 issued by The State Encryption Administration.

**SM2 Control Word EDX 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.

### 1.1  SM2 Encrypt

- **Description**

  SM2 encryption means using specified public key to encrypt plaintext and get corresponding ciphertext. And the ciphertext can only be decrypted by the corresponding private key.

- **Input Registers**

  **EAX**  Input pointer for plaintext. Assumes ES segment.

  **EBX**  Pointer to the public key for encryption. Assumes ES segment.

**ECX**   Number of bytes for plaintext to be encrypted.

**EDX**   Control Word.

**ESI**   Pointer to 8K-byte scratch space. Assumes ES segment. Need to initial to 0s.

**EDI**   Output pointer for ciphertext. Assumes ES segment. The memory must be writable.

- **Output Registers**

**EAX**   Incremented by the number of bytes for plaintext.

**ECX**   Number of bytes for all ciphertext.

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

**EDI**   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. And the plaintext can only be encrypted by the corresponding public key.

- **Input Registers**

**EAX**   Input pointer for ciphertext. Assumes ES segment;

**EBX**   Pointer to the private key for decryption. Assumes ES segment.

**ECX**   Number of bytes for all ciphertext to be decrypted.

**EDX**   Control Word.

**ESI**   Pointer to 8K-byte scratch space. Assumes ES segment. Need to initial to 0s.

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

- **Output Registers**

**EAX**   Incremented by the number of bytes of input ciphertext.

**ECX**   Number of bytes for plaintext.

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

**EDI**    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.

- **Input Registers**

**EAX**   Input pointer for hash value after preprocess2. Assumes ES segment.

**EBX**   Pointer to the private key of signer. Assumes ES segment.

**EDX**   Control Word.

**ESI**   Pointer to 8K-byte scratch space. Assumes ES segment. Need to initial to 0s.

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

- **Output Registers**

    **ECX**    Number of bytes for all signature.

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

    **EDI**    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.

- **Input Registers**

    **EAX**    Input pointer for hash value after preprocess2. Assumes ES segment.

    **EBX**    Pointer to the public key of signer. Assumes ES segment.

    **EDX**    Control Word.

    **ESI**    Pointer to 8K-byte scratch space; Assumes ES segment. Need to initial to 0s.

    **EDI**    Pointer to the signature. Assumes ES segment.

- **Output Registers**

    **ECX**    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 sm2 key pair, include private key and public key.

- **Input Registers**

    **EDX**    Control Word.

    **ESI**    Pointer to 8K-byte scratch space. Assume ES segment. Need to initial to 0s.

    **EDI**    Output pointer for sm2 key pair for 32bytes private key and 64bytes public key. Assume ES segment. The memory must be writable.

- **Output Registers**

    **EDI**    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 user A and user B and selectively calculate hash value $S_2\&S_B$ according to control word.

- **Input Registers**

**EAX**   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;

   User B bits length of identification and identification;

**ECX**   Number of bits for shared key.

**EDX**   Control Word.

**ESI**   Pointer to 8K-byte scratch space. Assume ES segment. Need to initial to 0s.

**EDI**   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**

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

**EDI**   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 user A and user B and
   selectively calculate hash value $S_1 \& S_A$ according to control word.

- **Input Registers**

**EAX**   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;

   User B bits length of identification and identification;

**ECX**   Number of bits for key.

**EDX**   Control Word.

**ESI**   Pointer to 8K-byte scratch space. Assume ES segment. Need to initial to 0s.

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

- **Output Registers**

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

  **EDI**   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**

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

  **EBX**   Pointer to the public key. Assumes ES segment.

  **EDX**   Control Word.

  **ESI**   Pointer to 8K-byte scratch space. Assumes ES segment. Need to initial to 0s.

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

- **Output Registers**

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

  **EDI**   Incremented by the number of bytes for 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**

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

  **EBX**   Pointer to the message M to be signed. Assumes ES segment.

  **ECX**   Number of bytes for message M to be signed.

  **EDX**   Control Word.

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

- **Output Registers**

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

## 2 SM3

**Encoding:** 0xF3 0x0F 0xA6 0xE8

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

**Description:** Calculate SM3 hash algorithm as specified by GM/T 0004-2012 issued by The State Encryption Administration.

- **Input Registers**

  EAX    If EAX == 0 the SM3 instruction performs the padding for input data stream specified in SM3 specification, and ECX means byte counter of input data stream.

          If EAX == -1 the SM3 instruction does not perform the padding for input data stream, and ECX means the number of 64-byte blocks of input data stream.

  EBX    Control word. Bit5==1 means to perform SM3 hash algorithm. Other bits unused.

  ECX    Size of the input data stream:

          If EAX == 0: means in bytes

          If EAX == -1: means in 64-byte blocks

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

  EDI    Pointer to the memory for initial hash constants. Assumes ES segment. The memory must be writable.

- **Output Registers**

  EAX    If the input EAX == 0, EAX will be equal to ECX, otherwise it was 0.

  ECX    If the input EAX == 0, ECX will be unchanged, otherwise it was 0.

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

  EDI    Unchanged. The memory will contain the result of the hash calculation.

## 3 SM4

**Encoding:** 0xF3 0x0F 0xA7 0xF0

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

**Description:** Calculate SMS4 algorithm as specified by GM/T 0002-2012 issued by The State Encryption Administration.

- **Input Registers**

  EAX    Control word.

          Bit[0]: 0 means encryption; 1 means decryption.

          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.

  ECX    Number of 16-byte blocks to encrypt or decrypt.

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

  ESI    Input pointer to the memory for plaintext when encrypting or ciphertext when

decrypting. Assumes ES segment.

    **EDI**    Output pointer to the memory for ciphtertext when encrypting or plaintext when decrypting. Assumes ES segment. The memory must be writable.

- **Output Registers**

    **ECX**    0

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

    **EDI**    Incremented by the number of bytes ciphtertext when encrypting or plaintext when decrypting. The memory will contain the ciphtertext when encrypting or plaintext when decrypting.