# 8 Mbit bit Single Operating Voltage Serial Flash Memory With 104 MHz Dual- or Quad-Output SPI Bus Interface

**PRELIMINARY DATASHEET**

## FEATURES

**• Single Power Supply Operation**
- Low voltage range: 1.65 V – 1.95 V

**• Memory Organization**
- IS25WQ080: 1024K x 8 (8 Mbit)

**• Cost Effective Sector/Block Architecture**
- 8Mb : Uniform 4KByte sectors / Sixteen uniform 64KByte blocks

**• Serial Peripheral Interface (SPI) Compatible**
- Supports single-, dual- or quad-output
- Supports SPI Modes 0 and 3
- Maximum 33 MHz clock rate for normal read
- Maximum 104 MHz clock rate for fast read
- Maximum 208MHz clock rate equivalent Dual SPI
- Maximum 416MHz clock rate equivalent Quad SPI

**• Byte Program Operation**
- Typical 8 us/Byte

**• Page Program (up to 256 Bytes) Operation**
- Maximum 0.7 ms per page program

**• Sector, Block or Chip Erase Operation**
- Sector Erase (4KB)→150 ms (Max)
- Block Erase →(32K/64KB)→0.5S (Max)
- Chip Erase →6s (8Mb) (Max)

**•Deep power-down mode 1uA (Typ)**

**• Low Power Consumption**
- Max 15 mA active read current
- Max 20 mA program/erase current
- Max 50uA standby current

**• Hardware Write Protection**
- Protect and unprotect the device from write operation by Write Protect (WP#) Pin

**• Software Write Protection**
- The Block Protect (BP3, BP2, BP1, BP0) bits allow partial or entire memory to be configured as read-only

**• High Product Endurance**
- Guaranteed 100,000 program/erase cycles per single sector
- Minimum 20 years data retention

**• Industrial Standard Pin-out and Package**
- 8-pin SOIC 208mil
- 8-pin SOIC 150mil
- 8-pin VVSOP 150mil
- 8-pin WSON (5x6mm)
- Lead-free (Pb-free) package

**Additional 256-byte Security information one-time programmable (OTP) area**

## GENERAL DESCRIPTION

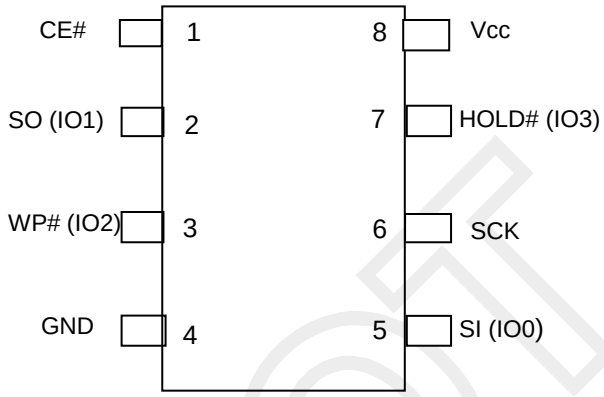
The IS25WQ080 are 8Mbit Serial Peripheral Interface (SPI) Flash memories, providing single-, dual or quad-output. The devices are designed to support a 33 MHz fclock rate in normal read mode, and 104 MHz in fast read, the fastest in the industry. The devices use a single low voltage power supply, ranging from 1.65 Volt to 1.95 Volt, to perform read, erase and program operations. The devices can be programmed in standard EPROM programmers.

The IS25WQ080 are accessed through a 4-wire SPI Interface consisting of Serial Data Input (SI), Serial Data Output (SO), Serial Clock (SCK), and Chip Enable (CE#) pins. The devices support page program mode, where 1 to 256 bytes data can be programmed into the memory in one program operation. These devices are divided into uniform 4 Kbyte sectors or uniform 64 Kbyte blocks.

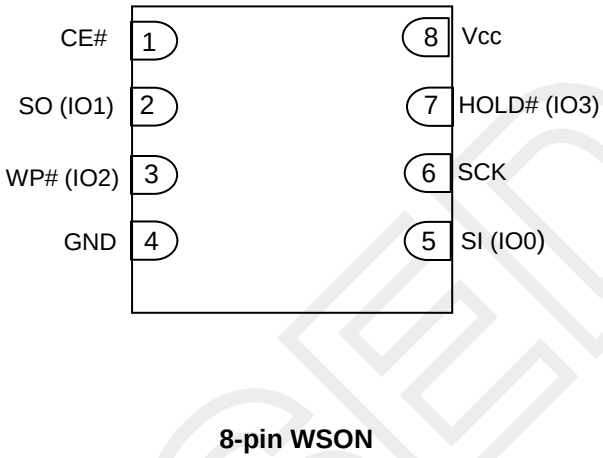
The IS25WQ080 are manufactured on pFLASH™'s advanced non-volatile technology.

NOT RELEASED

## CONNECTION DIAGRAMS




**8-Pin SOIC/VVSOP**




**8-pin WSON**

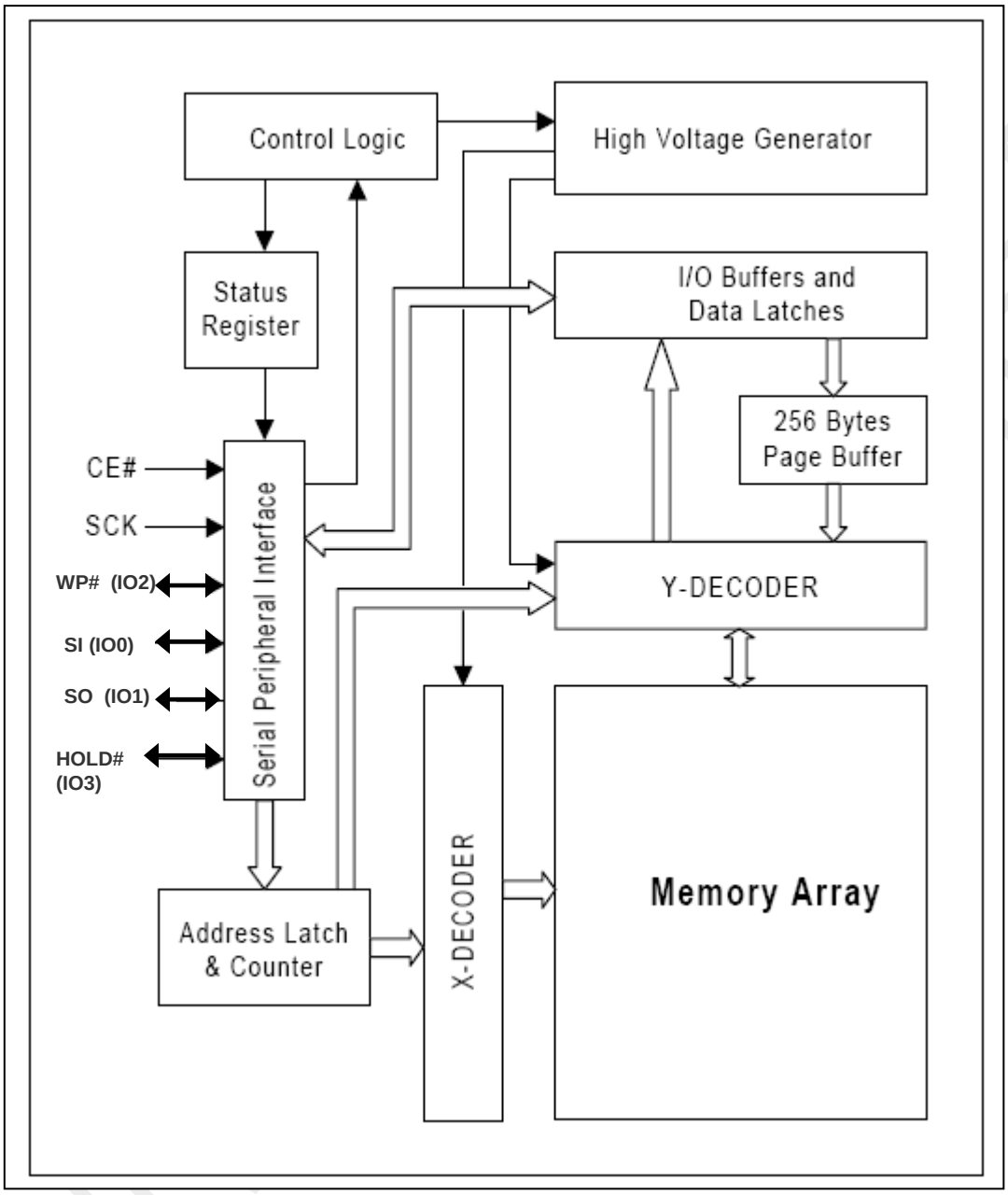
NOT RELEASED

## PIN DESCRIPTIONS

| SYMBOL | TYPE | DESCRIPTION |
|---|---|---|
| CE# | INPUT | Chip Enable: CE# low activates the devices internal circuitries for device operation. CE# high deselects the devices and switches into standby mode to reduce the power consumption. When a device is not selected, data will not be accepted via the serial input pin (SI), and the serial output pin (SO) will remain in a high impedance state. |
| SCK | INPUT | Serial Data Clock |
| SI (IO0) | INPUT/OUTPUT | Serial Data Input/Output |
| SO (IO1) | INPUT/OUTPUT | Serial Data Input/Output |
| GND | | Ground |
| Vcc | | Device Power Supply |
| WP# (IO2) | INPUT/OUTPUT | Write Protect/Serial Data Output: A hardware program/erase protection for all or part of a memory array. When the WP# pin is low, memory array write-protection depends on the setting of BP3, BP2, BP1 and BP0 bits in the Status Register. When the WP# is high, the status register are not write-protected. When the QE bit of is set “1”, the /WP pin (Hardware Write Protect) function is not available since this pin is used for IO2 |
| HOLD# (IO3) | INPUT/OUTPUT | Hold: Pause serial communication by the master device without resetting the serial sequence. When the QE bit of Status Register-2 is set for “1”, the function is Serial Data Input & Output (for 4xI/O read mode) |

NOT RELEASED

## BLOCK DIAGRAM



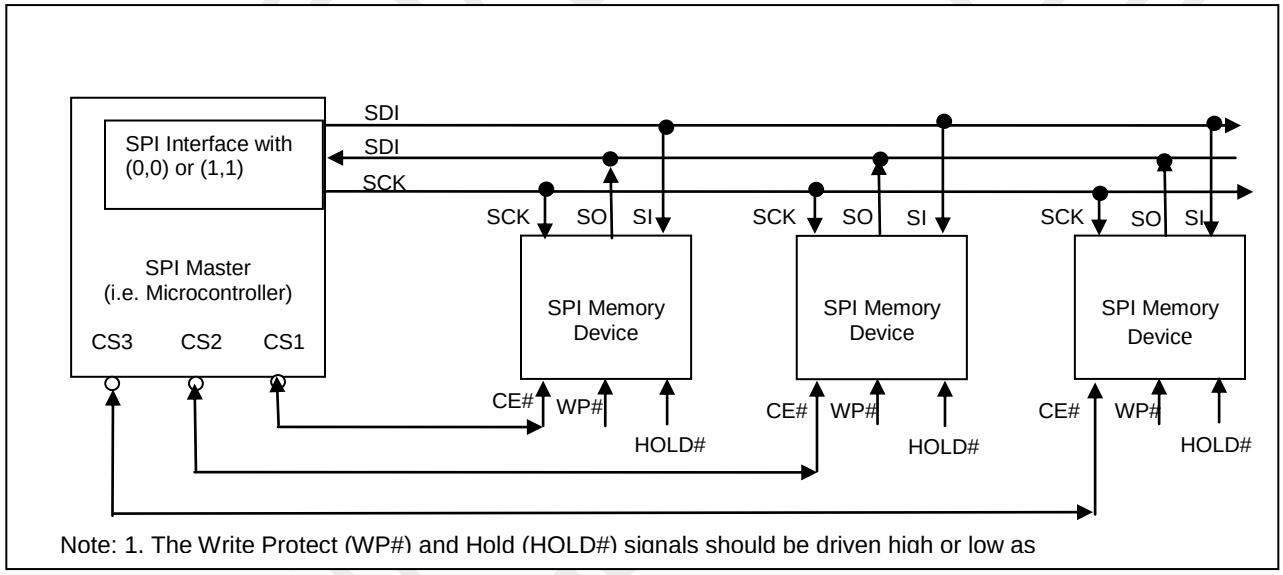
## SPI MODES DESCRIPTION

Multiple IS25WQ080 devices can be connected on the SPI serial bus and controlled by a SPI Master, i.e. microcontroller, as shown in Figure 1. The devices support either of two SPI modes:

Mode 0 (0, 0)
Mode 3 (1, 1)

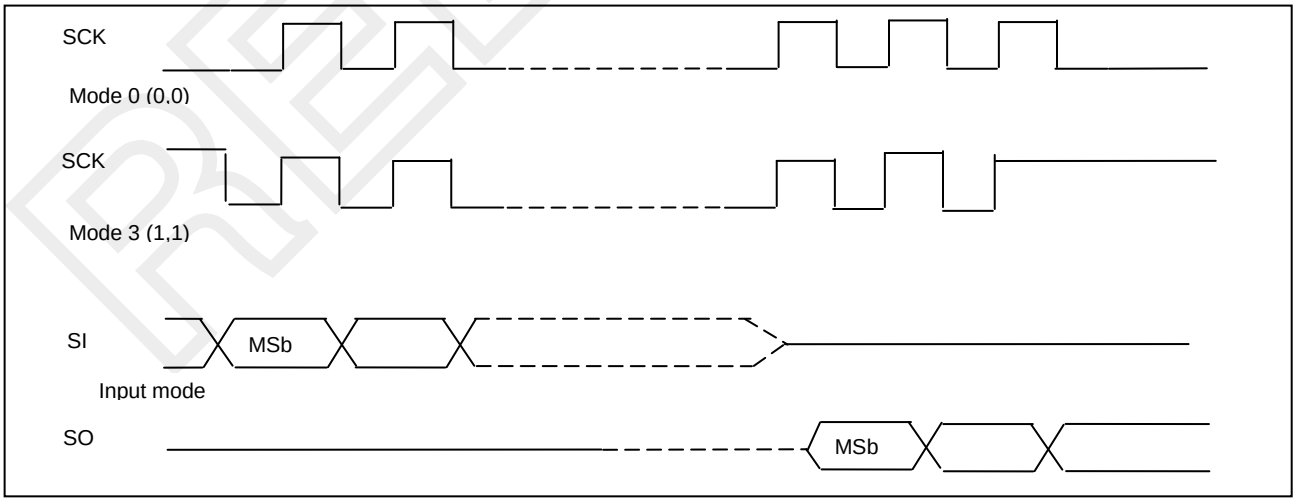
The difference between these two modes is the clock polarity when the SPI master is in Stand-by mode: the serial clock remains at "0" (SCK = 0) for Mode 0 and the clock remains at "1" (SCK = 1) for Mode 3. Please refer to Figure 2. For both modes, the input data is latched on the rising edge of Serial Clock (SCK), and the output data is available from the falling edge of SCK.

**Figure 1. Connection Diagram among SPI Master and SPI Slaves (Memory Devices)**




Note: 1. The Write Protect (WP#) and Hold (HOLD#) signals should be driven high or low as

**Figure 2. SPI Modes Supported**



## SYSTEM CONFIGURATION

The IS25WQ080 devices are designed to interface directly with the synchronous Serial Peripheral Interface (SPI) of the Motorola MC68HCxx series of microcontrollers or any SPI interface-equipped system controllers. The devices have the superset features that can be enabled through specific software instructions:

Configurable sector size: The memory array of IS25WQ080 is divided into uniform 4 Kbyte sectors or uniform 32K/64 Kbyte blocks (a block consists of sixteen adjacent sectors).

The below Table 1 illustrates the memory map of the devices.

## BLOCK/SECTOR ADDRESSES

**Table 1. Block/Sector Addresses of IS25WQ080**

| Memory Density | | Block No.(64KB) | Block No.(32KB) | Sector No. | Sector Size (Kbytes) | Address Range |
|---|---|---|---|---|---|---|
| 16 Mbit | 8 Mbit | Block 0 | Block 0 | Sector 0 | 4 | 000000h – 000FFFh |
| | | | | : | : | 001000h – 001FFFh |
| | | | Block 1 | : | : | : |
| | | | | Sector 15 | 4 | 00F000h – 00FFFFh |
| | | Block 1 | Block 2 | Sector 16 | 4 | 010000h – 010FFFh |
| | | | | : | : | 011000h – 011FFFh |
| | | | Block 3 | : | : | : |
| | | | | Sector 31 | 4 | 01F000h – 01FFFFh |
| | | : | : | : | : | : |
| | | Block 7 | : | : | : | 070000h – 07FFFFh |
| | | Block 8 | : | Sector 128 | 4 | 080000h – 08FFFFh |
| | | : | : | : | : | : |
| | | : | : | : | : | : |
| | | Block 15 | Block30 | : | : | 0F0000h – 0FFFFFh |
| | | | Block31 | Sector 255 | 4 | |
| | | Block 16 | Block32 | Sector 256 | 4 | 100000h – 10FFFFh |
| | | | Block33 | : | : | |
| | | : | : | : | : | : |
| | | : | : | : | : | : |
| | | Block 31 | Block62 | : | 4 | 1F0000h – 1FFFFFh |
| | | | Block63 | Sector511 | | |

NOT RELEASED

## REGISTERS (CONTINUED)

### STATUS REGISTER

Refer to Tables 2 and 5 for Status Register Format and Status Register Bit Definitions.

The BP0, BP1, BP2, BP3 and SRWD are non-volatile memory cells that can be written by a Write Status Register (WRSR) instruction. The default value of the BP2, BP1, BP0, and SRWD bits were set to "0" at factory. The Status Register can be read by the Read Status Register (RDSR). Refer to Table 10 for Instruction Set.

The function of Status Register bits are described as follows:

**WIP bit**: The Write In Progress (WIP) bit is read-only, and can be used to detect the progress or completion of a program or erase operation. When the WIP bit is "0", the device is ready for a write status register, program or erase operation. When the WIP bit is "1", the device is busy.

**WEL bit**: The Write Enable Latch (WEL) bit indicates the status of the internal write enable latch. When the WEL is "0", the write enable latch is disabled, and all write operations, including write status register, write configuration register, page program, sector erase, block and chip erase operations are inhibited. When the WEL bit is "1", write operations are allowed. The WEL bit is set by a Write Enable (WREN) instruction. Each write register, program and erase instruction must be preceded by a WREN instruction. The WEL bit can be reset by a Write Disable (WRDI) instruction. It will automatically be the reset after the completion of a write instruction.

**BP3, BP2, BP1, BP0 bits**: The Block Protection (BP3, BP2, BP1 and BP0) bits are used to define the portion of the memory area to be protected. Refer to Tables 6 and 7 for the Block Write Protection bit settings. When a defined combination of BP3, BP2, BP1 and BP0 bits are set, the corresponding memory area is protected. Any program or erase operation to that area will be inhibited. Note: a Chip Erase (CHIP_ER) instruction is executed only if all the Block Protection Bits are set as "0"s.

**SRWD bit**: The Status Register Write Disable (SRWD) bits operates in conjunction with the Write Protection (WP#) signal to provide a Hardware Protection Mode. When the SRWD is set to "0", the Status Register is not write-protected. When the SRWD is set to "1" and the WP# is pulled low ($V_{IL}$), the bits of Status Register (SRWD, BP3, BP2, BP1, BP0) become read-only, and a WRSR instruction will be ignored. If the SRWD is set to "1" and WP# is pulled high ($V_{IH}$), the Status Register can be changed by a WRSR instruction.

**QE bit**: The Quad Enable (QE) is a non-volatile bit in the status register that allows Quad operation. When the QE bit is set to "0",the pin WP# and HOLD# are enable. When the QE bit is set to "1", the pin IO2 and IO3 are enable.

WARNING: The QE bit should never be set to a 1 during standard SPI or Dual SPI operation if the WP# or HOLD# pins are tied directly to the power supply or ground.

**Table 2. Status Register Format**

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | SRWD | QE | BP3 | BP2 | BP1 | BP0 | WEL | WIP |
| Default (flash bit) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- The default value of the BP3, BP2, BP1, BP0, and SRWD bits were set to "0" at factory.

## REGISTERS (CONTINUED)

### Function REGISTER

The Function Register can be read by the Read Function Register (RFR). Refer to Table 9 for Instruction Set.

The function of Function Register bits are described as follows:

**ESUS bit:** The Erase Suspend Status indicates when an Erase operation has been suspended. The ESUS bit is '1' after the host issues a suspend command during an Erase operation. Once the suspended Erase resumes, the ESUS bit is reset to '0.'

**PSUS bit:** The Program Suspend Status bit indicates when a Program operation has been suspended. The PSUS is '1' after the host issues a suspend command during the Program operation. Once the suspended Program resumes, the PSUS bit is reset to '0.'

**Table 3. Function Register Format**

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | X | X | X | X | X | PSUS | ESUS | X |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 4. Function register bit definition**

| Bit | Name | Definition | Read-/Write | Non-Volatile bit |
|---|---|---|---|---|
| Bit 1 | ESUS | Erase suspend bit:<br>"0" indicates Erase is not suspend<br>"1" indicates Erase is suspend | R | No |
| Bit 2 | PSUS | Program suspend bit:<br>"0" indicates program is not suspend<br>"1" indicates program is suspend | R | **No** |

NOT RELEASED

## REGISTERS (CONTINUED)

**Table 5. Status Register Bit Definition**

| Bit | Name | Definition | Read-/Write | Non-Volatile bit |
|---|---|---|---|---|
| Bit 0 | WIP | Write In Progress Bit:<br>"0" indicates the device is ready<br>"1" indicates a write cycle is in progress and the device is busy | R | No |
| Bit 1 | WEL | Write Enable Latch:<br>"0" indicates the device is not write enabled (default)<br>"1" indicates the device is write enabled | R/W | No |
| Bit 2 | BP0 | Block Protection Bit: (See Tables 6 and 7 for details)<br>"0" indicates the specific blocks are not write-protected (default)<br>"1" indicates the specific blocks are write-protected | R/W | Yes |
| Bit 3 | BP1 | | | |
| Bit 4 | BP2 | | | |
| Bit 5 | BP3 | | | |
| Bit 6 | QE | Quad Enable bit:<br>"0" indicates the Quad output function disable (default)<br>"1" indicates the Quad output function enable | R/W | **Yes** |
| Bit 7 | SRWD | Status Register Write Disable: (See Table 10 for details)<br>"0" indicates the Status Register is not write-protected (default)<br>"1" indicates the Status Register is write-protected | R/W | Yes |

**Table 7. Block Write Protect Bits for IS25WQ080**

| Status Register Bits | | | | Protected Memory Area |
|---|---|---|---|---|
| **BP3** | **BP2** | **BP1** | **BP0** | **8 Mbit** |
| 0 | 0 | 0 | 0 | None |
| 0 | 0 | 0 | 1 | Upper sixteenth (1 block : 15$^{th}$): |
| 0 | 0 | 1 | 0 | Upper eighth (2 blocks :14$^{th}$ and 15$^{th}$): |
| 0 | 0 | 1 | 1 | Upper quarter (4 blocks :12$^{th}$ to 15$^{th}$): |
| 0 | 1 | 0 | 0 | Upper half (8 blocks :8$^{th}$ to 15$^{th}$): |
| 0 | 1 | 0 | 1 | All blocks (16blocks : 0 to 15): 000000h – 0FFFFFh |
| 0 | 1 | 1 | 0 | |
| 0 | 1 | 1 | 1 | |
| 1 | 0 | 0 | 0 | |
| 1 | 0 | 0 | 1 | |
| 1 | 0 | 1 | 0 | |
| 1 | 0 | 1 | 1 | (8blocks :0$^{th}$ to 7$^{th}$): |
| 1 | 1 | 0 | 0 | (4 blocks :0$^{th}$ to 3th): |
| 1 | 1 | 0 | 1 | (2blocks :0$^{th}$ to 1th): |
| 1 | 1 | 1 | 0 | (1 blocks :0$^{th}$ ): |
| 1 | 1 | 1 | 1 | None |

NOT RELEASED

## REGISTERS (CONTINUED)

## PROTECTION MODE

The IS25WQ080 have two types of write-protection mechanisms: hardware and software. These are used to prevent irrelevant operation in a possibly noisy environment and protect the data integrity.

**HARDWARE WRITE-PROTECTION**

The devices provide two hardware write-protection features:

a. When inputting a program, erase or write status register instruction, the number of clock pulse is checked to determine whether it is a multiple of eight before the executing. Any incomplete instruction command sequence will be ignored.

b. Write inhibit is 1.3V, all write sequence will be ignored when Vcc drop to 1.3V and lower.

c. The Write Protection (WP#) pin provides a hardware write protection method for BP3, BP2, BP1, BP0 and SRWD in the Status Register. Refer to the STATUS REGISTER description.

**SOFTWARE WRITE PROTECTION**

The IS25WQ080 also provides two software write protection features:

a. Before the execution of any program, erase or write status register instruction, the Write Enable Latch (WEL) bit must be enabled by executing a Write Enable (WREN) instruction. If the WEL bit is not enabled first, the program, erase or write register instruction will be ignored.

b. The Block Protection (BP3, BP2, BP1, BP0) bits allow part or the whole memory area to be write-protected.

**Table 8. Hardware Write Protection on Status Register**

| SRWD | WP# | Status Register |
|---|---|---|
| 0 | Low | Writable |
| 1 | Low | Protected |
| 0 | High | Writable |
| 1 | High | Writable |

## DEVICE OPERATION

The IS25WQ080 utilize an 8-bit instruction register. Refer to Table 9 Instruction Set for details of the Instructions and Instruction Codes. All instructions, addresses, and data are shifted in with the most significant bit (MSB) first on Serial Data Input (SI). The input data on SI is latched on the rising edge of Serial Clock (SCK) after Chip Enable (CE#) is driven low ($V_{IL}$). Every instruction sequence starts with a one-byte instruction code and is followed by address bytes, data bytes, or both address bytes and data bytes, depending on the type of instruction. CE# must be driven high ($V_{IH}$) after the last bit of the instruction sequence has been shifted in.

The timing for each instruction is illustrated in the following operational descriptions.

NOT RELEASED

**Table 9. Instruction Set**

| Instruction Name | Hex Code | Operation | Command Cycle | Maximum Frequency |
|---|---|---|---|---|
| RDID | Abh | Read Manufacturer and Product ID/ release deep power down mode | 4 Bytes | 104 MHz |
| JEDEC ID READ | 9Fh | Read Manufacturer and Product ID by JEDEC ID Command | 1 Byte | 104 MHz |
| RDMDID | 90h | Read Manufacturer and Device ID | 4 Bytes | 104 MHz |
| WREN | 06h | Write Enable | 1 Byte | 104 MHz |
| WRDI | 04h | Write Disable | 1 Byte | 104 MHz |
| RDSR | 05h | Read Status Register | 1 Byte | 104 MHz |
| WRSR | 01h | Write Status Register | 2 Bytes | 104 MHz |
| READ | 03h | Read Data Bytes from Memory at Normal Read Mode | 4 Bytes | 33 MHz |
| FAST_READ | 0Bh | Read Data Bytes from Memory at Fast Read Mode | 5 Bytes | 104 MHz |
| FRDO | 3Bh | Fast Read Dual Output | 5 Bytes | 104 MHz |
| FRDIO | BBh | Fast Read Dual I/O | 3 Bytes | 104MHz |
| FRQO | 6Bh | Fast Read Quad Output | 5 Bytes | 104 MHz |
| FRQIO | Ebh | Fast Read Quad I/O | 2 Bytes | 104MHz |
| MR | FFh | Mode Reset | 2 Byte | 104MHz |
| PAGE_ PROG | 02h | Page Program Data Bytes Into Memory | 4 Bytes + 256B | 104 MHz |
| SECTOR_ER | D7h/ 20h | Sector Erase | 4 Bytes | 104 MHz |
| BLOCK_ER (32KB) | 52h | Block Erase | 4 Bytes | 104 MHz |
| BLOCK_ER (64KB) | D8h | Block Erase | 4 Bytes | 104 MHz |
| CHIP_ER | C7h/ 60h | Chip Erase | 1 Byte | 104 MHz |
| Dual page program | A2h | Page Program Data Bytes Into Memory with Dual interface | | 104MHz |
| Quad page program | 32h | Page Program Data Bytes Into Memory with Quad interface | 4 Bytes + 256B | 104MHz |
| Power down | B9h | | | 104MHz |
| Program information Raw | B1h | Program 256 bytes of Security area | 4 Bytes | 104 MHz |
| Read information Raw | 4Bh | Read 256 bytes of Security area | 4 Bytes | 33 MHz |
| Program/Erase Suspend | 75h/ B0h | Suspend during the program/erase | 1 byte | 104MHz |
| Program/Erase Resume | 7Ah/ 30h | Resume program/erase | 1 byte | 104MHz |
| RDFR | 07h | Read function register | 1 byte | 104MHz |

## HOLD OPERATION

HOLD# is used in conjunction with CE# to select the IS25WQ080. When the devices are selected and a serial sequence is underway, HOLD# can be used to pause the serial communication with the master device without resetting the serial sequence.

To pause, HOLD# is brought low while the SCK signal is low. To resume serial communication, HOLD# is brought high while the SCK signal is low (SCK may still toggle during HOLD). Inputs to SI will be ignored while SO is in the high impedance state.

## DEVICE OPERATION (CONTINUED)

### RDID COMMAND (READ PRODUCT IDENTIFICATION)/ Release Power-down OPERATION

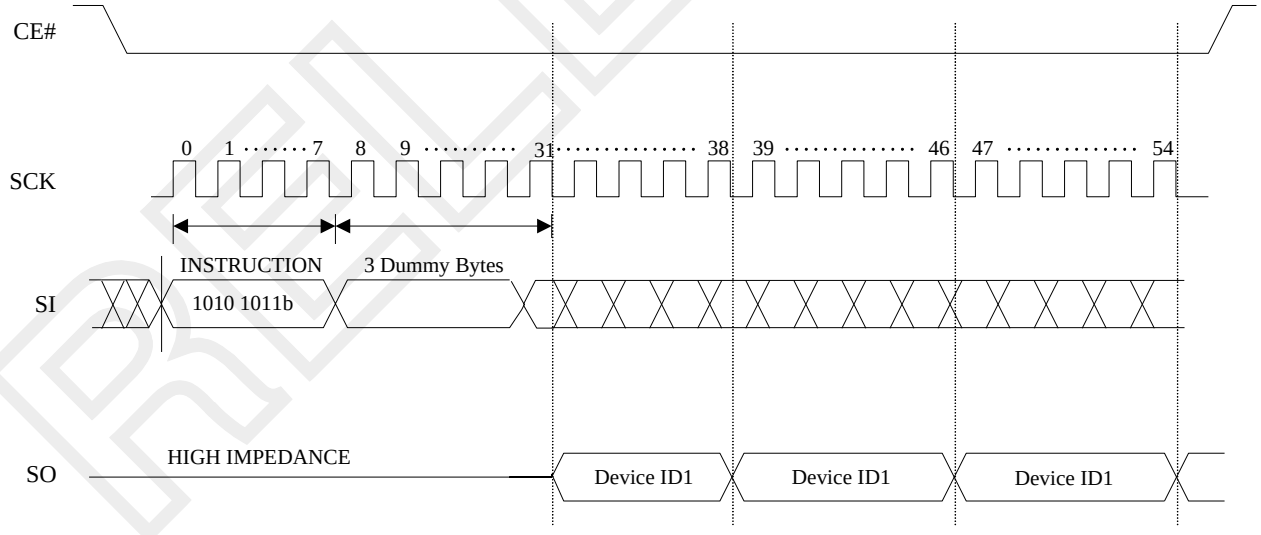
The Release from Power-down or High performance Mode / Device ID instruction is a multi-purpose instruction. The Read Product Identification (RDID) instruction is for reading out the old style of 8-bit Electronic Signature, whose values are shown as table 10 of ID Definitions. This is not same as RDID or JEDEC ID instruction. It's not recommended to use for new design. For new design, please use RDID or JEDEC ID instruction.
The RDES instruction code is followed by three dummy bytes, each bit being latched-in on SI during the rising edge of SCK. Then the Device ID is shifted out on SO with the MSB first, each bit been shifted out during the falling edge of SCK. The RDES instruction is ended by CE# goes high. The Device ID outputs repeatedly if continuously send the additional clock cycles on SCK while CE# is at low.

To release the device from the power-down state Mode, the instruction is issued by driving the CE# pin low, shifting the instruction code "Abh" and driving CE# high as shown in figure 3.
Release from power-down will take the time duration of $t_{RES1}$ before the device will resume normal operation and other instructions are accepted. The CE# pin must remain high during the $t_{RES1}$ time duration. If the Release from Power-down / RDID instruction is issued while an Erase, Program or Write cycle is in process (when WIP equals 1) the instruction is ignored and will not have any effects on the current cycle

**Table 10. Product Identification**

| Product Identification | | Data |
|---|---|---|
| Manufacturer ID | First Byte | 9Dh |
| | Second Byte | 7Fh |
| Device ID: | Device ID1 | Device ID2 |
| IS25WQ080 | 13h | 54h |

**Figure 3. Read Product Identification Sequence**



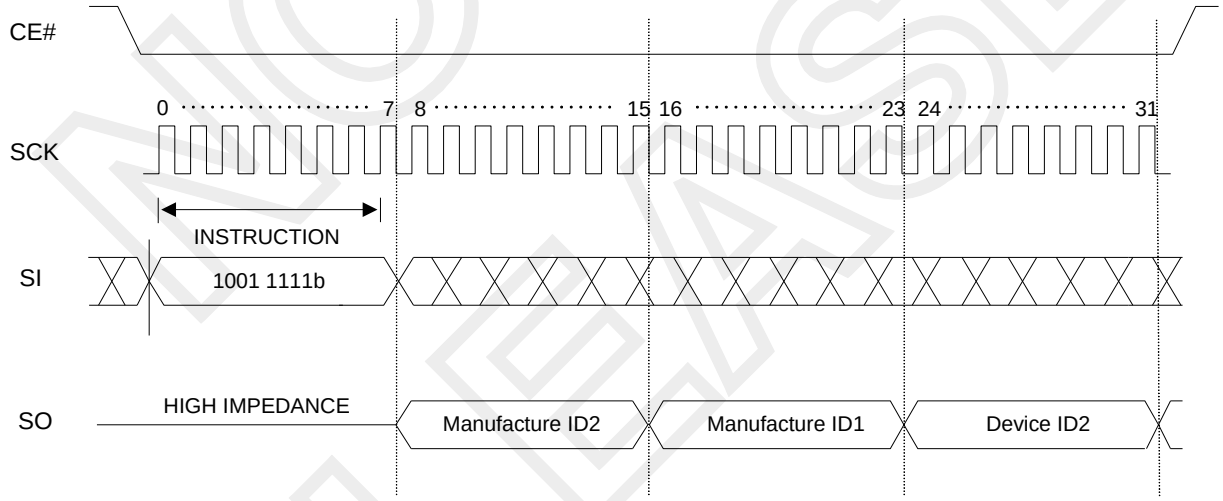

NOT RELEASED

## DEVICE OPERATION (CONTINUED)

### JEDEC ID READ COMMAND (READ PRODUCT IDENTIFICATION BY JEDEC ID) OPERATION

The JEDEC ID READ instruction allows the user to read the manufacturer and product ID of devices. Refer to Table 10 Product Identification for pFlash Manufacturer ID and Device ID. After the JEDEC ID READ command is input, the second Manufacturer ID (7Fh) is shifted out on SO with the MSB first, followed by the first Manufacturer ID (9Dh) and the Device ID2 (54h, in the case of the IS25WQ080), each bit shifted out during the falling edge of SCK. If CE# stays low after the last bit of the Device ID is shifted out, the Manufacturer ID and Device ID will loop until CE# is pulled high.

**Figure 4. Read Product Identification by JEDEC ID READ Sequence**



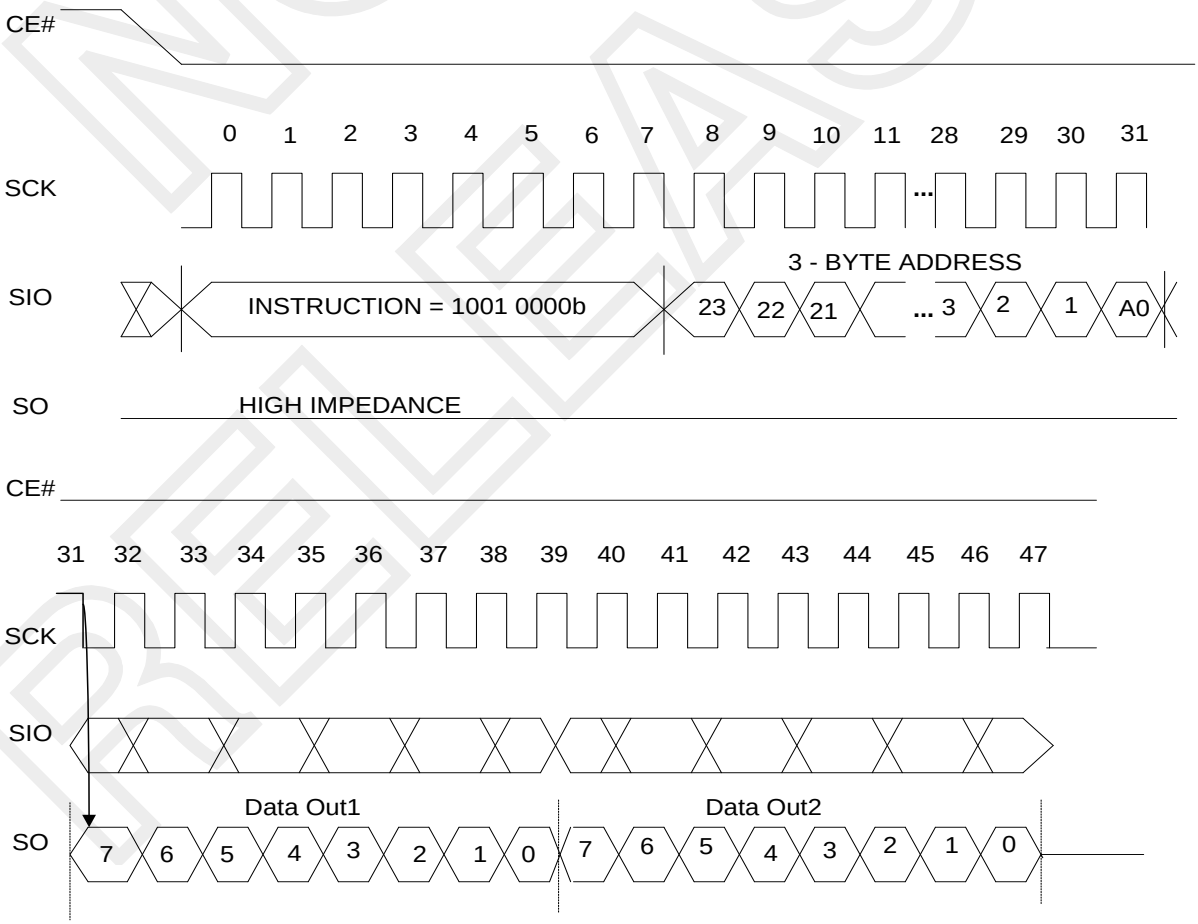

NOT RELEASED

## DEVICE OPERATION (CONTINUED)

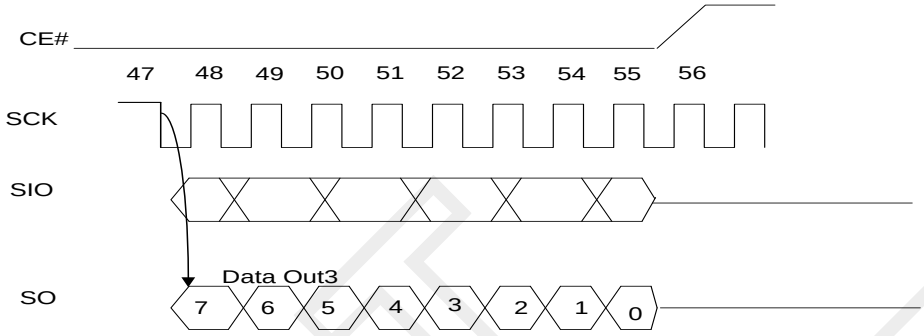
### RDMDID COMMAND (READ DEVICE MANUFACTURER AND DEVICE ID) OPERATION

The Read Product Identification (RDID) instruction allows the user to read the manufacturer and product ID of the devices. Refer to Table 10 Product Identification for pFLASH™ manufacturer ID and device ID. The RDID instruction code is followed by two dummy bytes and one byte address (A7~A0), each bit being latched-in on SI during the rising edge of SCK. If one byte address is initially set to A0 = 0, then the first manufacturer ID (9Dh) is shifted out on SO with the MSB first, the device ID1 and the second manufacturer ID (7Fh), each bit been shifted out during the falling edge of SCK. If one byte address is initially set to A0 = 1, then device ID1 will be read first, then followed by the first manufacture ID (9Dh) and then second manufacture ID (7Fh). The manufacture and device ID can be read continuously, alternating from one to the others. The instruction is completed by driving CE# high.

**Figure 5. Read Product Identification by RDMDID READ Sequence**




NOT RELEASED




**Note :**

1. ADDRESS A0 = 0, will output the 1st manufacture ID (9Dh) first -> device ID1 -> 2nd manufacture ID (7Fh)
   ADDRESS A0 = 1, will output the device ID1 -> 1st manufacture ID (9D) -> 2nd manufacture ID (7Fh)

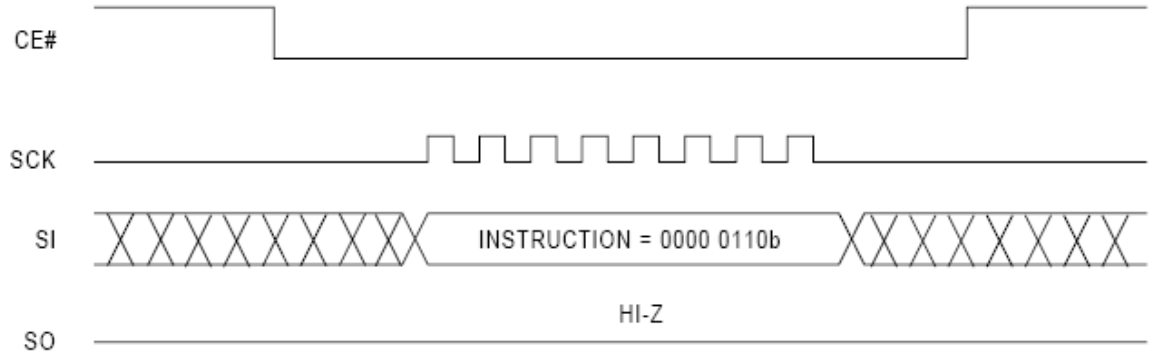
NOT RELEASED

## DEVICE OPERATION (CONTINUED)

### WRITE ENABLE OPERATION

The Write Enable (WREN) instruction is used to set the Write Enable Latch (WEL) bit. The WEL bit of the IS25WQ080 is reset to the write –protected state after power-up. The WEL bit must be write enabled before any write operation, including sector, block erase, chip erase, page program and write status register. The WEL bit will be reset to the write-protect state automatically upon completion of a write operation. The WREN instruction is required before any above operation is executed.

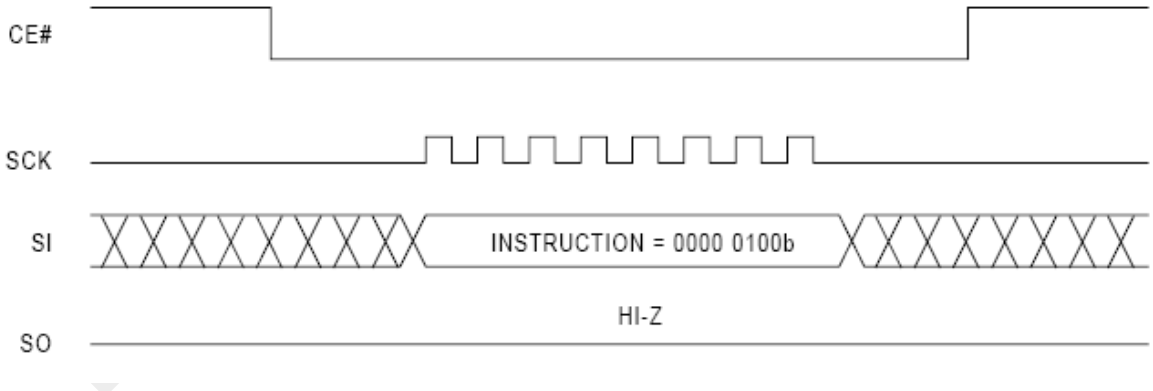
**Figure 6. Write Enable Sequence**




### WRDI COMMAND (WRITE DISABLE) OPERATION

The Write Disable (WRDI) instruction resets the WEL bit and disables all write instructions. The WRDI instruction is not required after the execution of a write instruction, since the WEL bit is automatically reset.

**Figure 7. Write Disable Sequence**

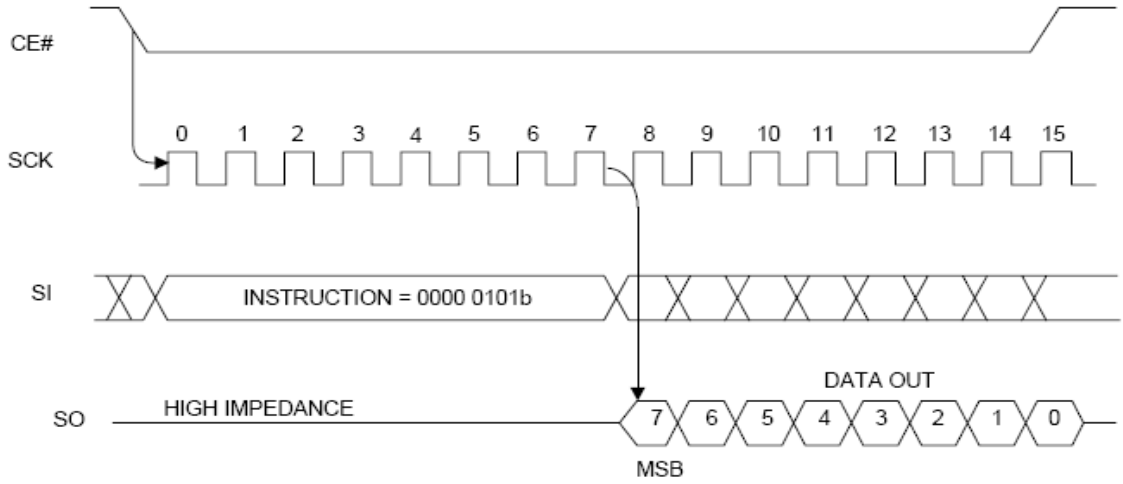


## DEVICE OPERATION (CONTINUED)

### RDSR COMMAND (READ STATUS REGISTER) OPERATION

The Read Status Register (RDSR) instruction provides access to the Status Register. During the execution of a program, erase or write status register operation, all other instructions will be ignored except the RDSR instruction, which can be used to check the progress or completion of an operation by reading the WIP bit of Status Register.

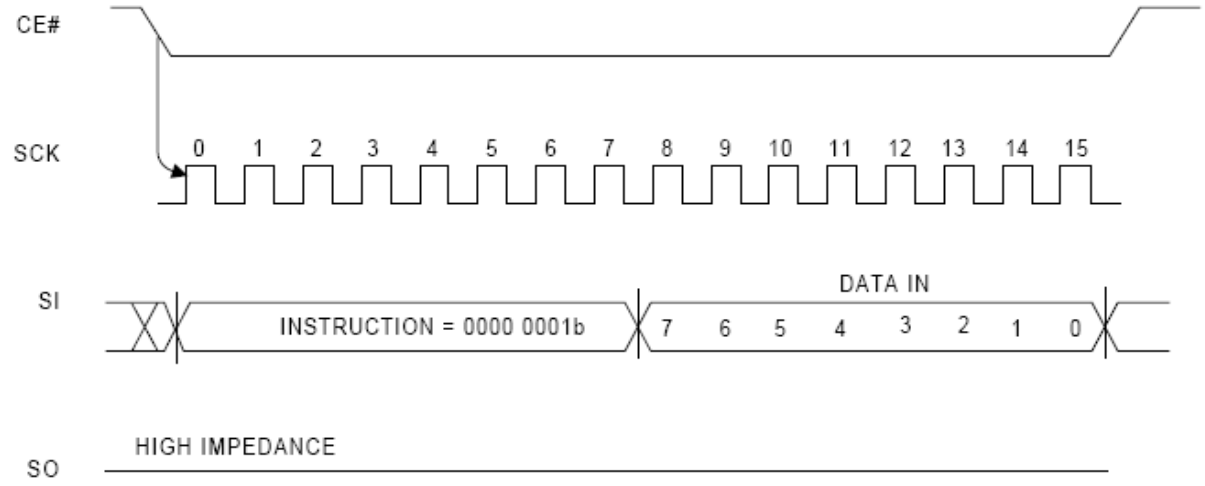
**Figure 8. Read Status Register Sequence**




### WRSR COMMAND (WRITE STATUS REGISTER) OPERATION

The Write Status Register (WRSR) instruction allows the user to enable or disable the block protection and status register write protection features by writing “0”s or “1”s into the non-volatile BP3, BP2, BP1, BP0 and SRWD bits.

**Figure 9. Write Status Register Sequence**



## DEVICE OPERATION (CONTINUED)

### READ COMMAND (READ DATA) OPERATION

The Read Data (READ) instruction is used to read memory data of a IS25WQ080 under normal mode running up to 33 MHz.

The READ instruction code is transmitted via the SI line, followed by three address bytes (A23 – A0) of the first memory location to be read. A total of 24 address bits are shifted in, but only $A_{MS}$ (most significant address) – A0 are decoded. The remaining bits (A23 – $A_{MS}$) are ignored. The first byte addressed can be at any memory location. Upon completion, any data on the SI will be ignored. Refer to Table 11 for the related Address Key.

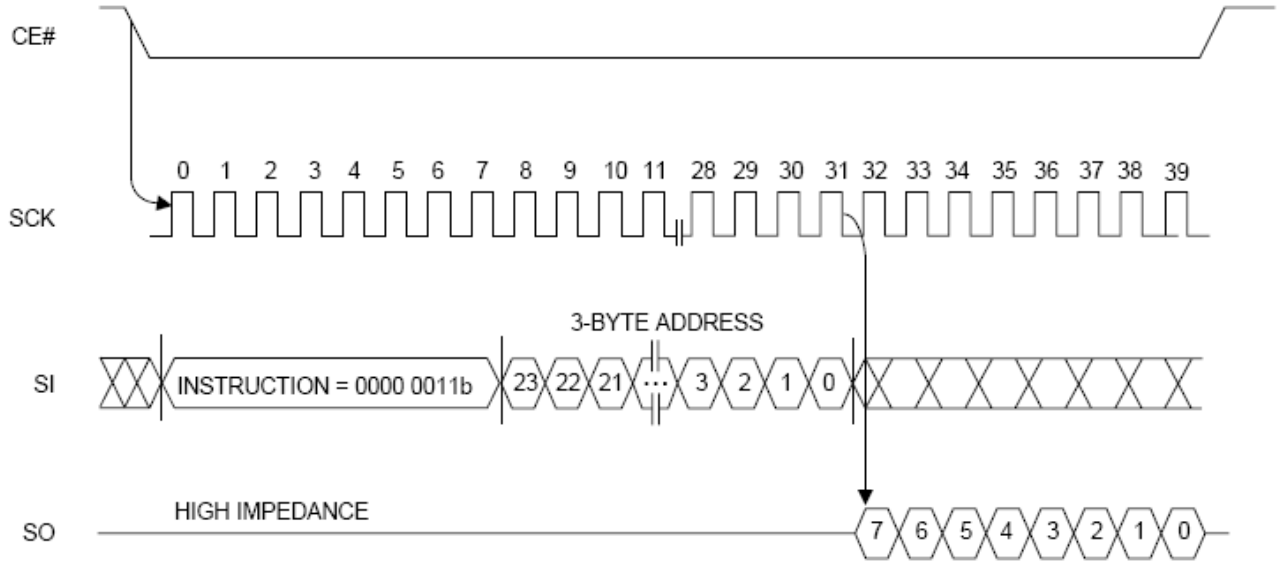
The first byte data (D7 – D0) addressed is then shifted out on the SO line, MSb first. A single byte of data, or up to the whole memory array, can be read out in one READ instruction. The address is automatically incremented after each byte of data is shifted out. The read operation can be terminated at any time by driving CE# high ($V_{IH}$) after the data comes out. When the highest address of the devices is reached, the address counter will roll over to the 000000h address, allowing the entire memory to be read in one continuous READ instruction.

If a Read Data instruction is issued while an Erase, Program or Write cycle is in process (WIP=1) the instruction is ignored and will not have any effects on the current cycle

**Table 11. Address Key**

| Address | IS25WQ080 |
|---|---|
| $A_N$ ($A_{MS}$ – $A_{0)}$ | A19 – A0 |
| Don't Care Bits | A23 – A20 |

**Figure 10. Read Data Sequence**




NOT RECOMMENDED FOR NEW DESIGN

## DEVICE OPERATION (CONTINUED)

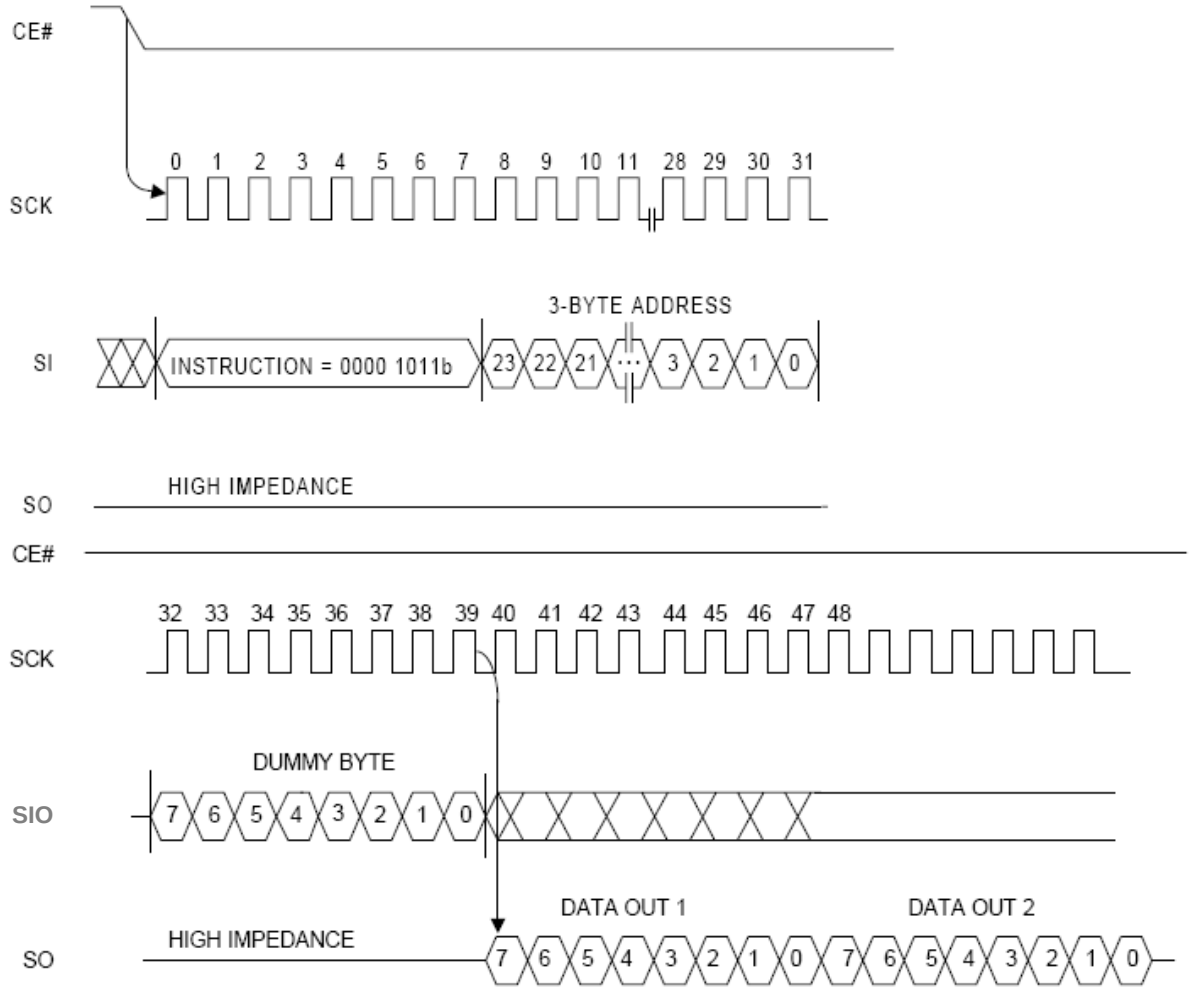
### FAST_READ COMMAND (FAST READ DATA) OPERATION

The FAST_READ instruction is used to read memory data at up to a 104 MHz clock.

The FAST_READ instruction code is followed by three address bytes (A23 – A0) and a dummy byte (8 clocks), transmitted via the SI line, with each bit latched-in during the rising edge of SCK. Then the first data byte addressed is shifted out on the SO line, with each bit shifted out at a maximum frequency $f_{CT}$, during the falling edge of SCK.

The first byte addressed can be at any memory location. The address is automatically incremented after each byte of data is shifted out. When the highest address is reached, the address counter will roll over to the 000000h address, allowing the entire memory to be read with a single FAST_READ instruction. The FAST_READ instruction is terminated by driving CE# high ($V_{IH}$). If a Fast Read Data instruction is issued while an Erase, Program or Write cycle is in process (WIP=1) the instruction is ignored and will not have any effects on the current cycle

**Figure 11. Fast Read Data Sequence**



## DEVICE OPERATION (CONTINUED)

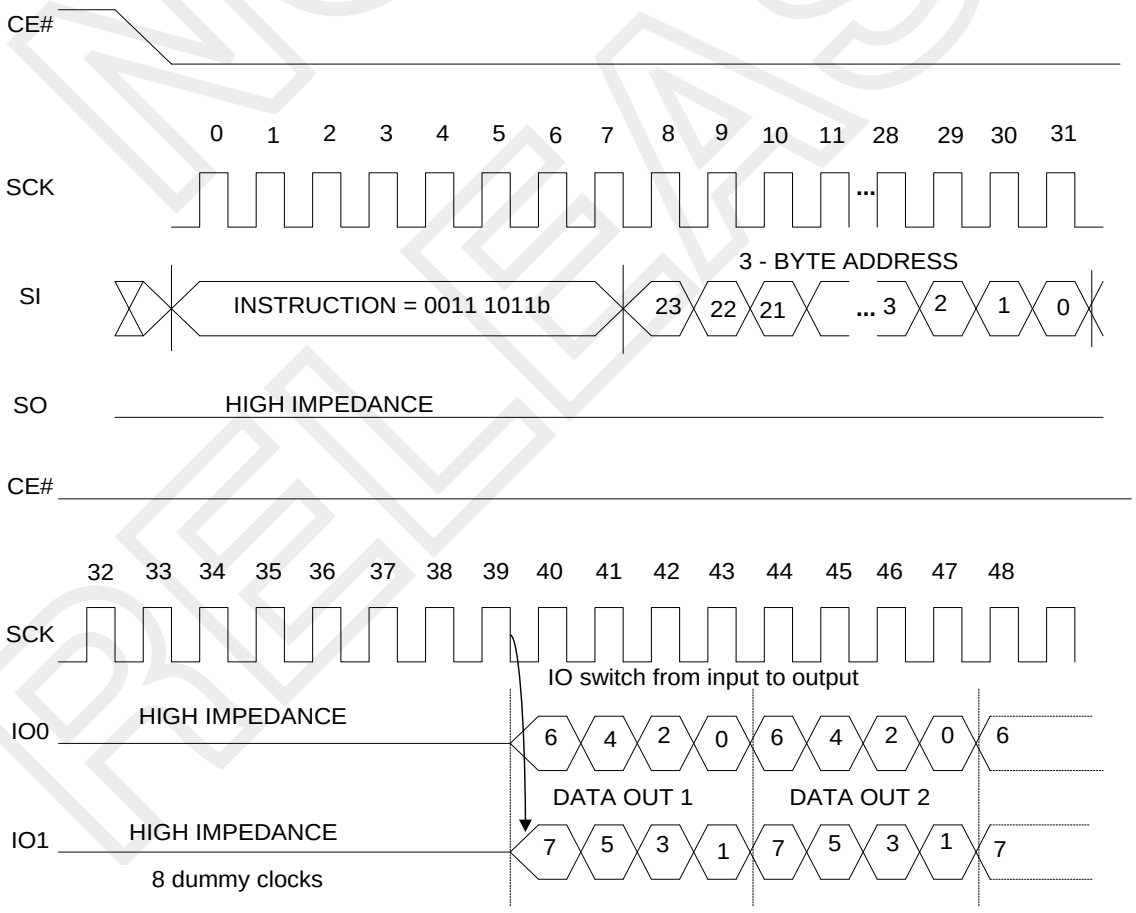
### FRDO COMMAND (FAST READ DUAL OUTPUT) OPERATION

The FRDO instruction is used to read memory data on two output pins each at up to a 104 MHz clock.

The FRDO instruction code is followed by three address bytes (A23 – A0) and a dummy byte (8 clocks), transmitted via the SI line, with each bit latched-in during the rising edge of SCK. Then the first data byte addressed is shifted out on the SO and SIO lines, with each pair of bits shifted out at a maximum frequency $f_{CT}$, during the falling edge of SCK. The first bit (MSb) is output on SO, while simultaneously the second bit is output on SIO.

The first byte addressed can be at any memory location. The address is automatically incremented after each byte of data is shifted out. When the highest address is reached, the address counter will roll over to the 000000h address, allowing the entire memory to be read with a single FRDO instruction. FRDO instruction is terminated by driving CE# high ($V_{IH}$). If a FRDO instruction is issued while an Erase, Program or Write cycle is in process (WIP=1) the instruction is ignored and will not have any effects on the current cycle

**Figure 12. Fast Read Dual-Output Sequence**




NOT RELEASED

## DEVICE OPERATION (CONTINUED)

### FRDIO COMMAND (FAST READ DUAL I/O) OPERATION

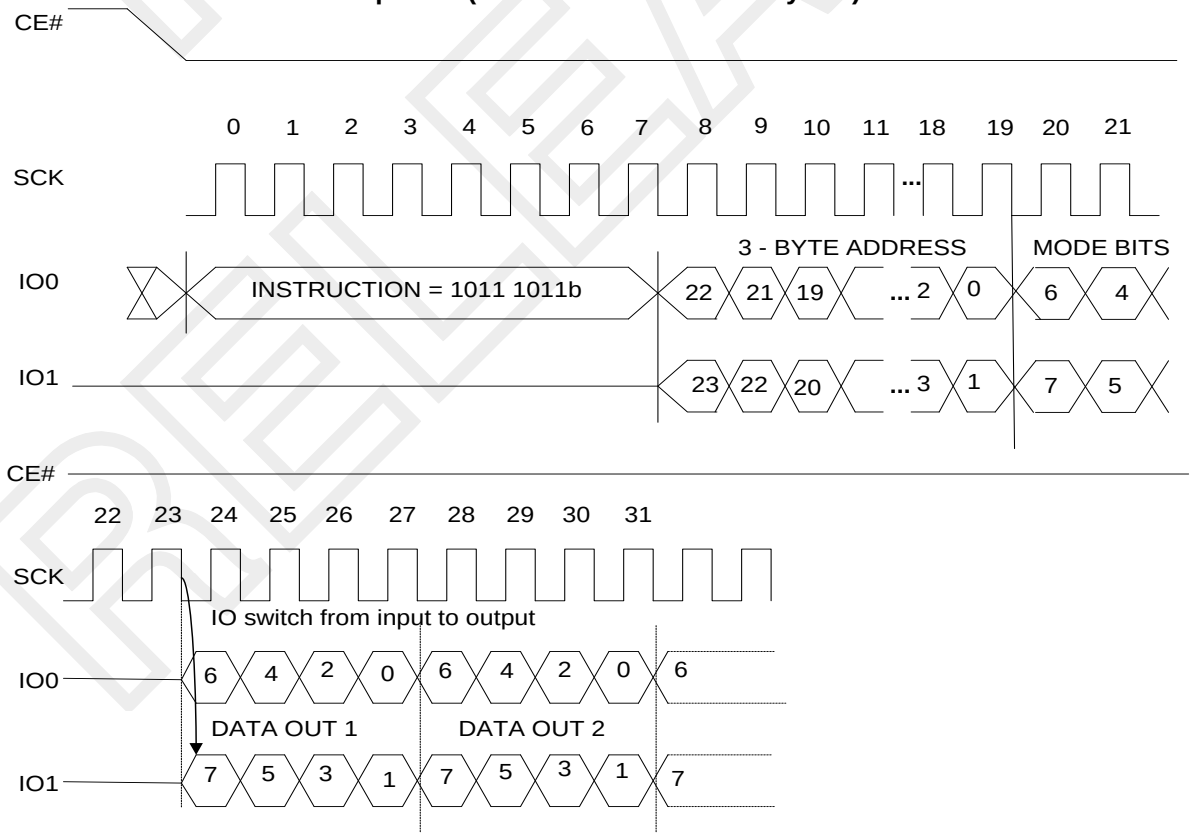
The FRDIO instruction is similar to the FRDO instruction, but allows the address bits to be input two bits at a time. This may allow for code to be executed directly from the SPI in some applications.

The FRDIO instruction code is followed by three address bytes (A23 – A0) and a mode byte, transmitted via the IO0 and IO1 lines, with each pair of bits latched-in during the rising edge of SCK. The address MSb is input on IO1, the next bit on IO0, and continues to shift in alternating on the two lines. The mode byte contains the value Ax, where x is a "don't care" value. Then the first data byte addressed is shifted out on the IO1 and IO0 lines, with each pair of bits shifted out at a maximum frequency $f_{CT}$, during the falling edge of SCK. The MSb is output on IO1, while simultaneously the second bit is output on IO0. Figure 13 illustrates the timing sequence.

The first byte addressed can be at any memory location. The address is automatically incremented after each byte of data is shifted out. When the highest address is reached, the address counter will roll over to the 000000h address, allowing the entire memory to be read with a single FRDIO instruction. FRDIO instruction is terminated by driving CE# high ($V_{IH}$).

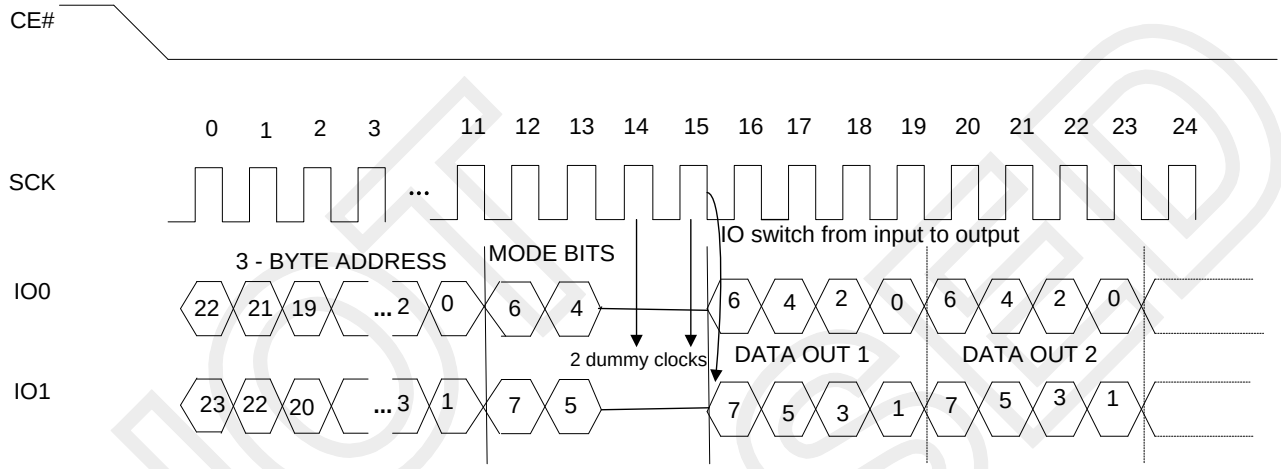
The device expects the next operation will be another FRDIO. It remains in this mode until it receives a Mode Reset (FFh) command. In subsequent FRDIO execution, the command code is not input, saving timing cycles as described in Figure 14. If a FRDIO instruction is issued while an Erase, Program or Write cycle is in process (WIP=1) the instruction is ignored and will not have any effects on the current cycle

**Figure 13. Fast Read Dual I/O Sequence (with command decode cycles)**



## DEVICE OPERATION (CONTINUED)

**Figure 14. Fast Read Dual I/O Sequence (without command decode cycles)**




### FRQO COMMAND (FAST READ QUAD OUTPUT) OPERATION

The FRQO instruction is used to read memory data on four output pins each at up to a 104 MHz clock.

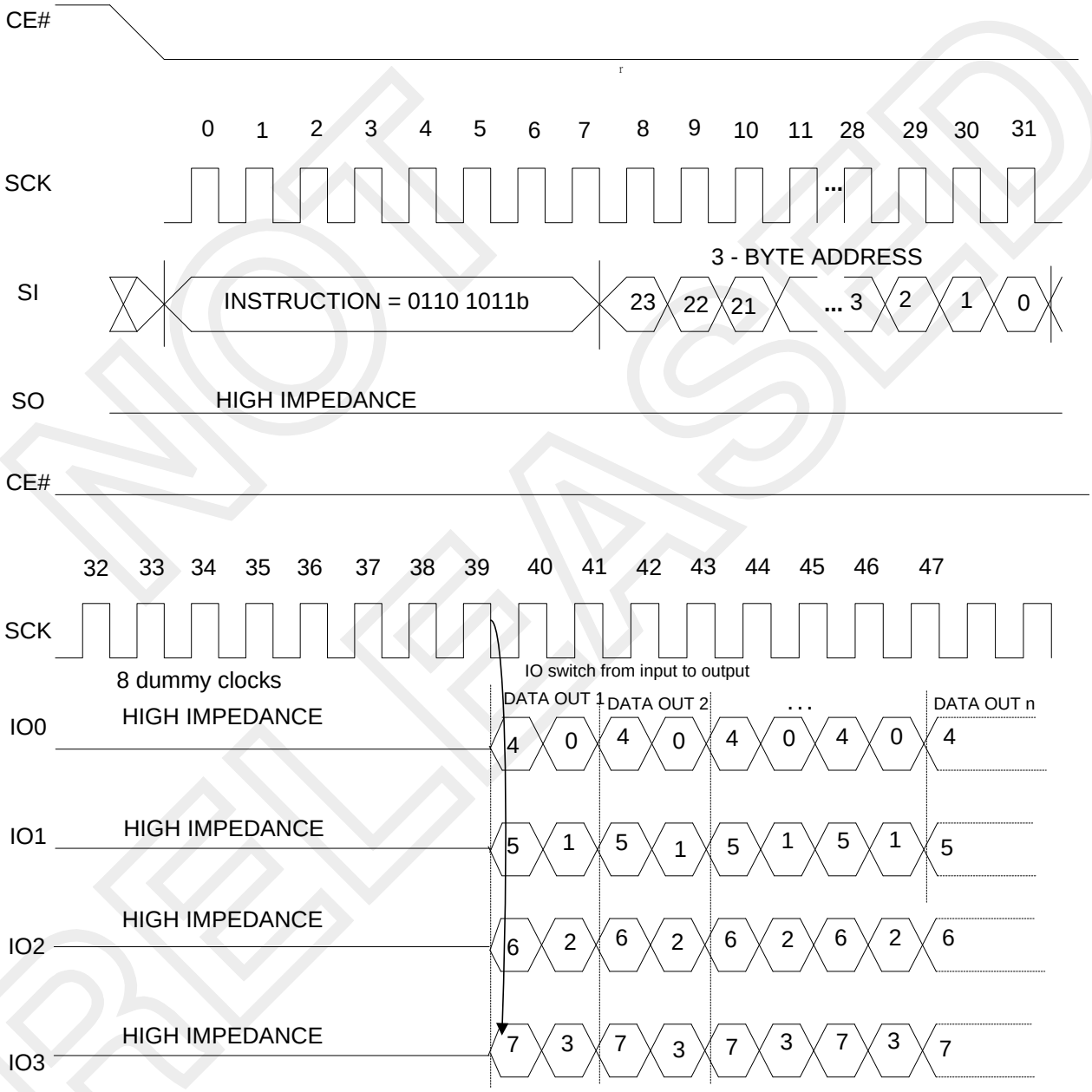
The FRQO instruction code is followed by three address bytes (A23 – A0) and a dummy byte (8 clocks), transmitted via the SI line, with each bit latched-in during the rising edge of SCK. Then the first data byte addressed is shifted out on the IO3, IO2, IO1 and IO0 lines, with each group of four bits shifted out at a maximum frequency $f_{CT}$, during the falling edge of SCK. The first bit (MSb) is output on IO3, while simultaneously the second bit is output on IO2, the third bit is output on IO1, etc.

The first byte addressed can be at any memory location. The address is automatically incremented after each byte of data is shifted out. When the highest address is reached, the address counter will roll over to the 000000h address, allowing the entire memory to be read with a single FRQO instruction. FRQO instruction is terminated by driving CE# high ($V_{IH}$). If a FRQO instruction is issued while an Erase, Program or Write cycle is in process (WIP=1) the instruction is ignored and will not have any effects on the current cycle

NOT RELEASED

## DEVICE OPERATION (CONTINUED)

**Figure 15. Fast Read Quad-Output Sequence**




NOT RELEASED

## DEVICE OPERATION (CONTINUED)

### FRQIO COMMAND (FAST READ QUAD I/O) OPERATION

The FRQIO instruction is similar to the FRQO instruction, but allows the address bits to be input four bits at a time. This may allow for code to be executed directly from the SPI in some applications.

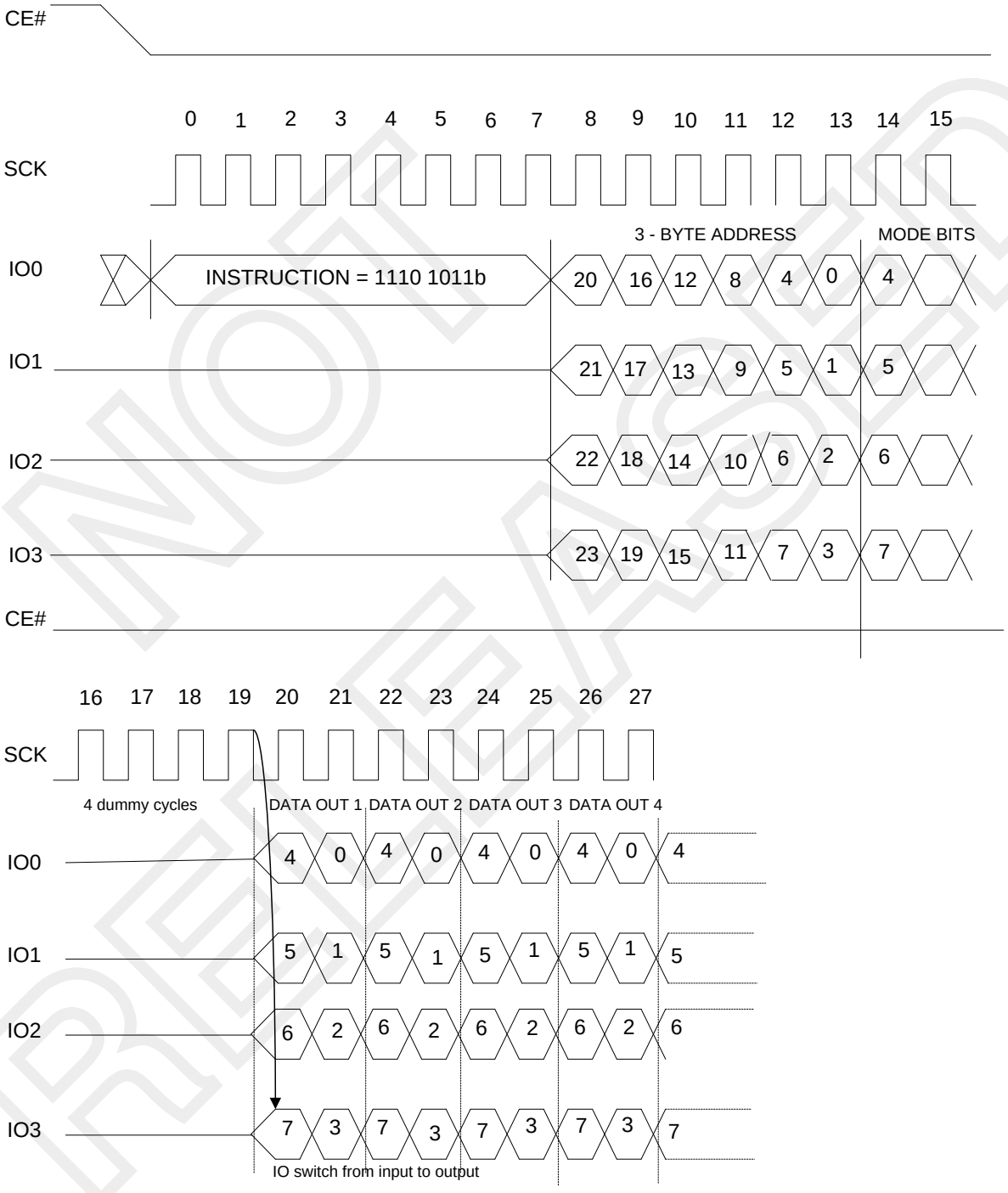
The FRQIO instruction code is followed by three address bytes (A23 – A0) and a mode byte, transmitted via the IO3, IO2, IO0 and IO1 lines, with each group of four bits latched-in during the rising edge of SCK. The address MSb is input on IO3, the next bit on IO2, the next bit on IO1, the next bit on IO0, and continue to shift in alternating on the four. The mode byte contains the value Ax, where x is a "don't care" value. After four dummy clocks, the first data byte addressed is shifted out on the IO3, IO2, IO1 and IO0 lines, with each group of four bits shifted out at a maximum frequency $f_{CT}$, during the falling edge of SCK. The first bit (MSb) is output on IO3, while simultaneously the second bit is output on IO2, the third bit is output on IO1, etc. Figure 16 illustrates the timing sequence.

The first byte addressed can be at any memory location. The address is automatically incremented after each byte of data is shifted out. When the highest address is reached, the address counter will roll over to the 000000h address, allowing the entire memory to be read with a single FRQIO instruction. FRQIO instruction is terminated by driving CE# high ($V_{IH}$).

The device expects the next operation will be another FRQIO. It remains in this mode until it receives a Mode Reset (FFh) command. In subsequent FRQIO execution, the command code is not input, saving cycles as described in Figure 17. If a FRQIO instruction is issued while an Erase, Program or Write cycle is in process (WIP=1) the instruction is ignored and will not have any effects on the current cycle

NOT RELEASED

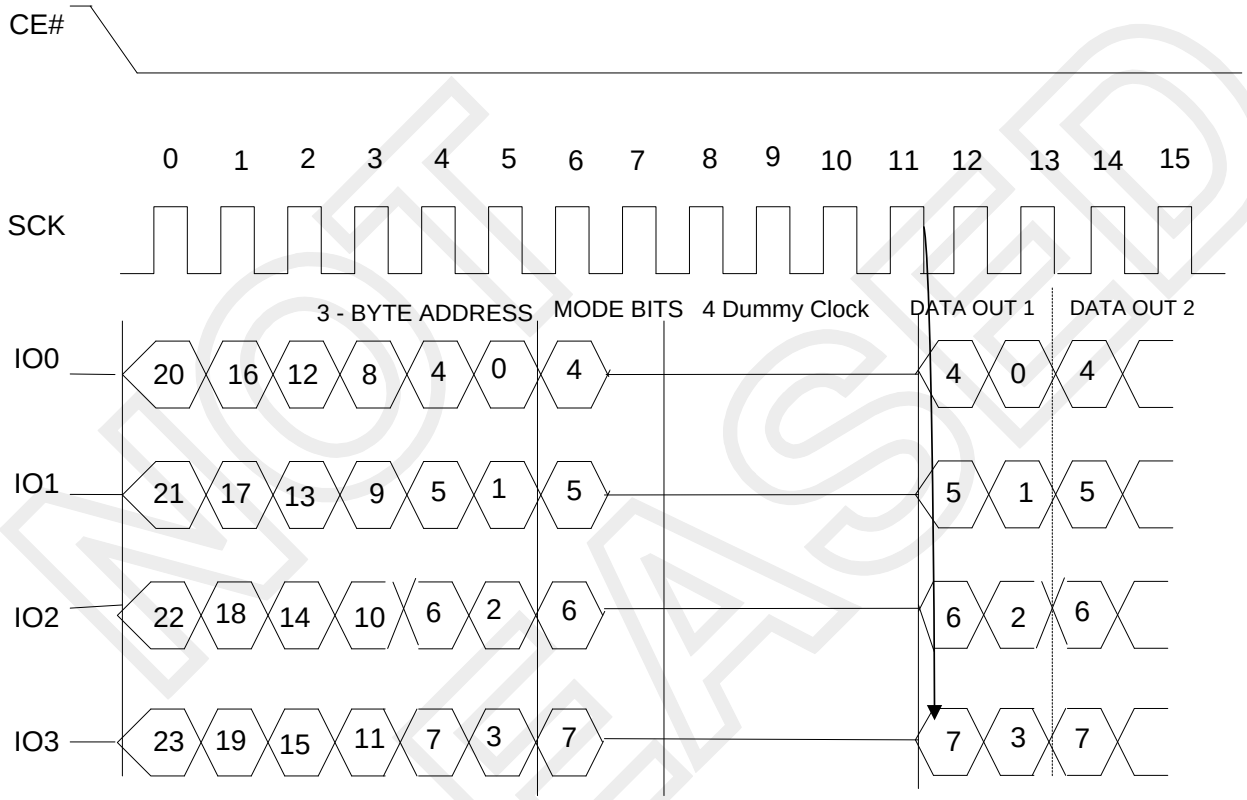
**Figure 16. Fast Read Quad I/O Sequence (with command decode cycles)**




NOT RELEASED

## DEVICE OPERATION (CONTINUED)

**Figure 17. Fast Read Quad I/O Sequence (without command decode cycles)**

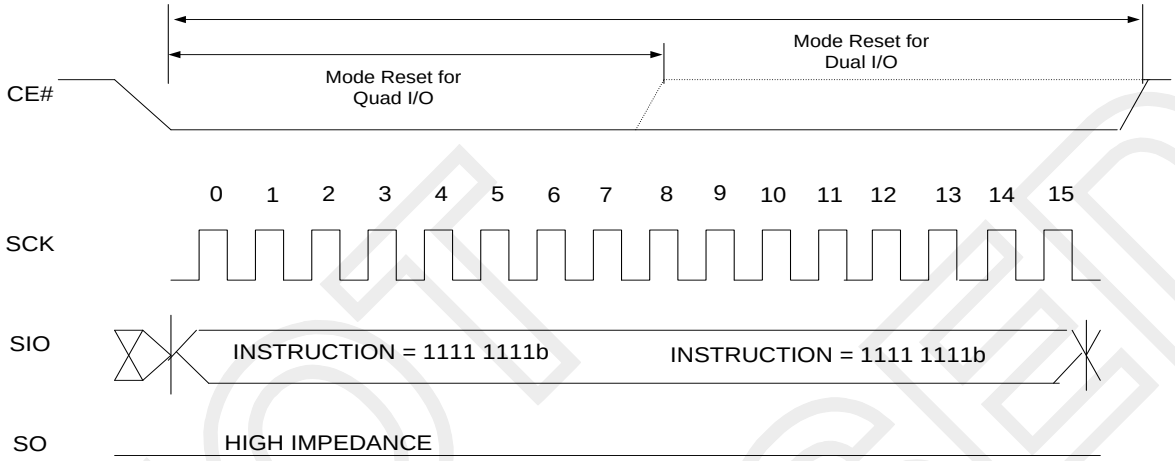



### MR COMMAND (MODE RESET) OPERATION

The Mode Reset command is used to conclude subsequent FRDIO and FRQIO operations. It resets the Mode bits to a value that is not Ax. It should be executed after an FRDIO or FRQIO operation, and is recommended also as the first command after a system reset. The timing sequence is different depending whether the MR command is used after an FRDIO or FRQIO, as shown in Figure 18.

NOT RELEASED

**Figure 18, Mode Reset Command**




NOT RELEASED

## DEVICE OPERATION (CONTINUED)

### PAGE_PROG COMMAND (PAGE PROGRAM) OPERATION

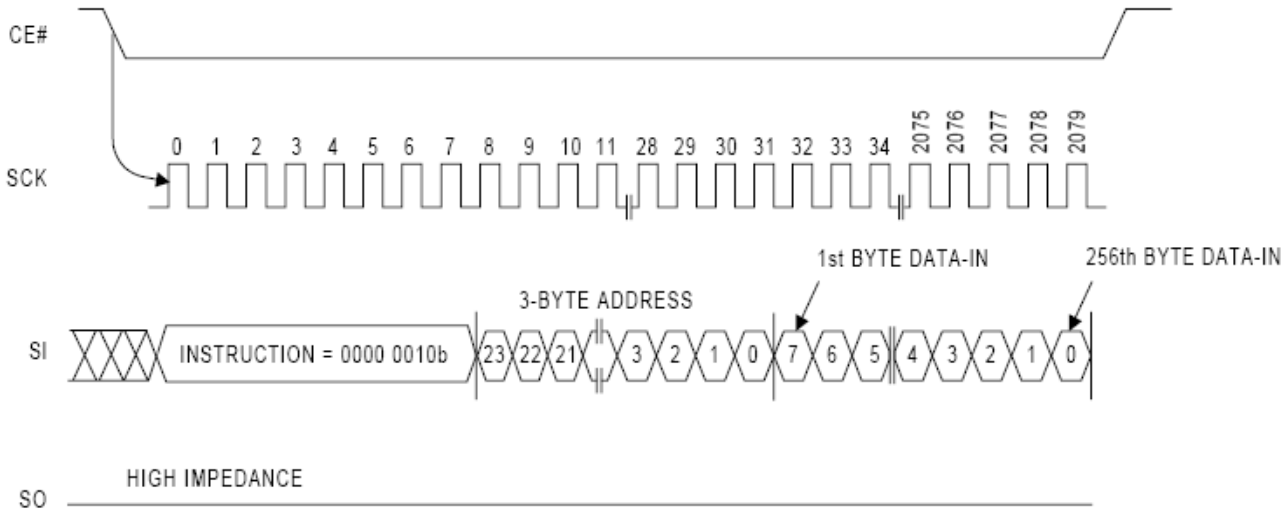
The Page Program (PAGE_PROG) instruction allows up to 256 bytes data to be programmed into memory in a single operation. The destination of the memory to be programmed must be outside the protected memory area set by the Block Protection (BP2, BP1, BP0) bits. A PAGE_PROG instruction which attempts to program into a page that is write-protected will be ignored. Before the execution of PAGE_PROG instruction, the Write Enable Latch (WEL) must be enabled through a Write Enable (WREN) instruction.

The PAGE_PROG instruction code, three address bytes and program data (1 to 256 bytes) are input via the SI line. Program operation will start immediately after the CE# is brought high, otherwise the PAGE_PROG instruction will not be executed. The internal control logic automatically handles the programming voltages and timing. During a program operation, all instructions will be ignored except the RDSR instruction. The progress or completion of the program operation can be determined by reading the WIP bit in Status Register via a RDSR instruction. If the WIP bit is "1", the program operation is still in progress. If WIP bit is "0", the program operation has completed.

If more than 256 bytes data are sent to a device, the address counter rolls over within the same page, the previously latched data are discarded, and the last 256 bytes data are kept to be programmed into the page. The starting byte can be anywhere within the page. When the end of the page is reached, the address will wrap around to the beginning of the same page. If the data to be programmed are less than a full page, the data of all other bytes on the same page will remain unchanged.

Note: A program operation can alter "1"s into "0"s, but an erase operation is required to change "0"s back to "1"s. A byte cannot be reprogrammed without first erasing the whole sector or block.

**Figure 19. Page Program Sequence**



## DEVICE OPERATION (CONTINUED)

### Quad Input Page Program operation

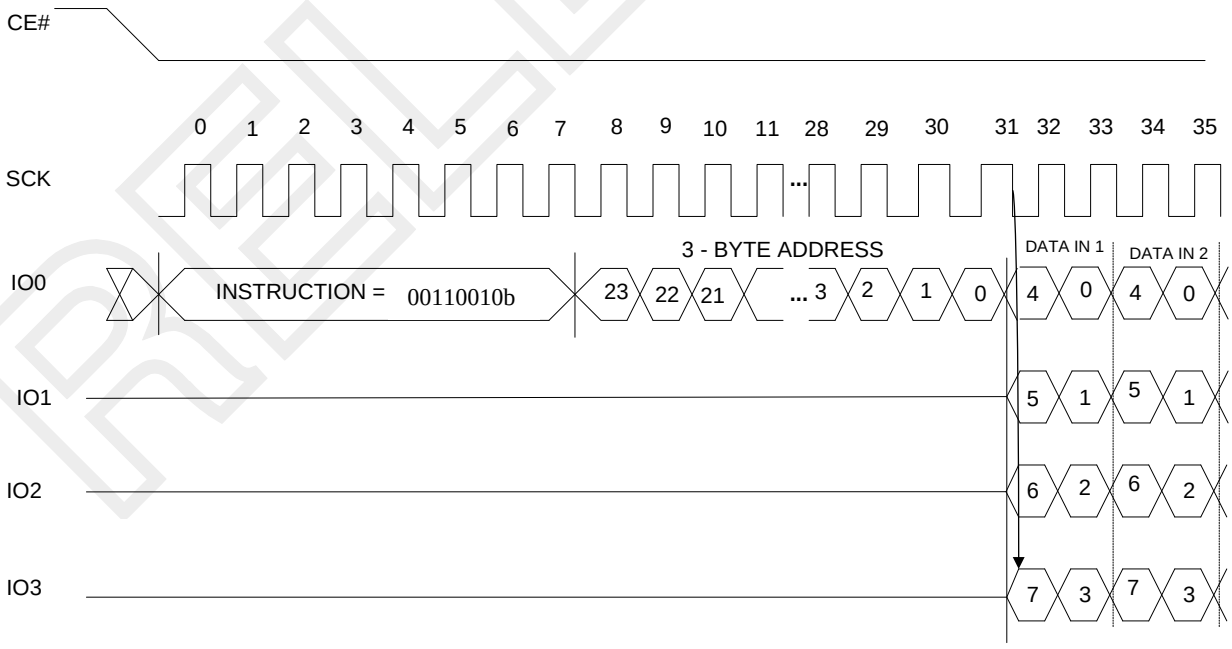
The Quad Input Page Program instruction allows up to 256 bytes data to be programmed into memory in a single operation with four pins (IO0, IO1, IO2 and IO3). The destination of the memory to be programmed must be outside the protected memory area set by the Block Protection (BP3, BP2, BP1, BP0) bits. A Quad Input Page Program instruction which attempts to program into a page that is write-protected will be ignored. Before the execution of Quad Input Page Program instruction, the QE bit in the status register must be set to "1" and the Write Enable Latch (WEL) must be enabled through a Write Enable (WREN) instruction.

The Quad Input Page Program instruction code, three address bytes and program data (1 to 256 bytes) are input via the four pins (IO0, IO1, IO2 and IO3). Program operation will start immediately after the CE# is brought high, otherwise the Quad Input Page Program instruction will not be executed. The internal control logic automatically handles the programming voltages and timing. During a program operation, all instructions will be ignored except the RDSR instruction. The progress or completion of the program operation can be determined by reading the WIP bit in Status Register via a RDSR instruction. If the WIP bit is "1", the program operation is still in progress. If WIP bit is "0", the program operation has completed.

If more than 256 bytes data are sent to a device, the address counter rolls over within the same page, the previously latched data are discarded, and the last 256 bytes data are kept to be programmed into the page. The starting byte can be anywhere within the page. When the end of the page is reached, the address will wrap around to the beginning of the same page. If the data to be programmed are less than a full page, the data of all other bytes on the same page will remain unchanged.

Note: A program operation can alter "1"s into "0"s, but an erase operation is required to change "0"s back to "1"s. A byte cannot be reprogrammed without first erasing the whole sector or block.

**Figure 20. Quad Page Program Sequence**



## DEVICE OPERATION (CONTINUED)

### Dual Input Page Program operation

The Dual Input Page Program instruction allows up to 256 bytes data to be programmed into memory in a single operation with two pins (IO0, IO1). The destination of the memory to be programmed must be outside the protected memory area set by the Block Protection (BP3, BP2, BP1, BP0) bits. A Dual Input Page Program instruction which attempts to program into a page that is write-protected will be ignored. Before the execution of Dual Input Page Program instruction, the Write Enable Latch (WEL) must be enabled through a Write Enable (WREN) instruction.

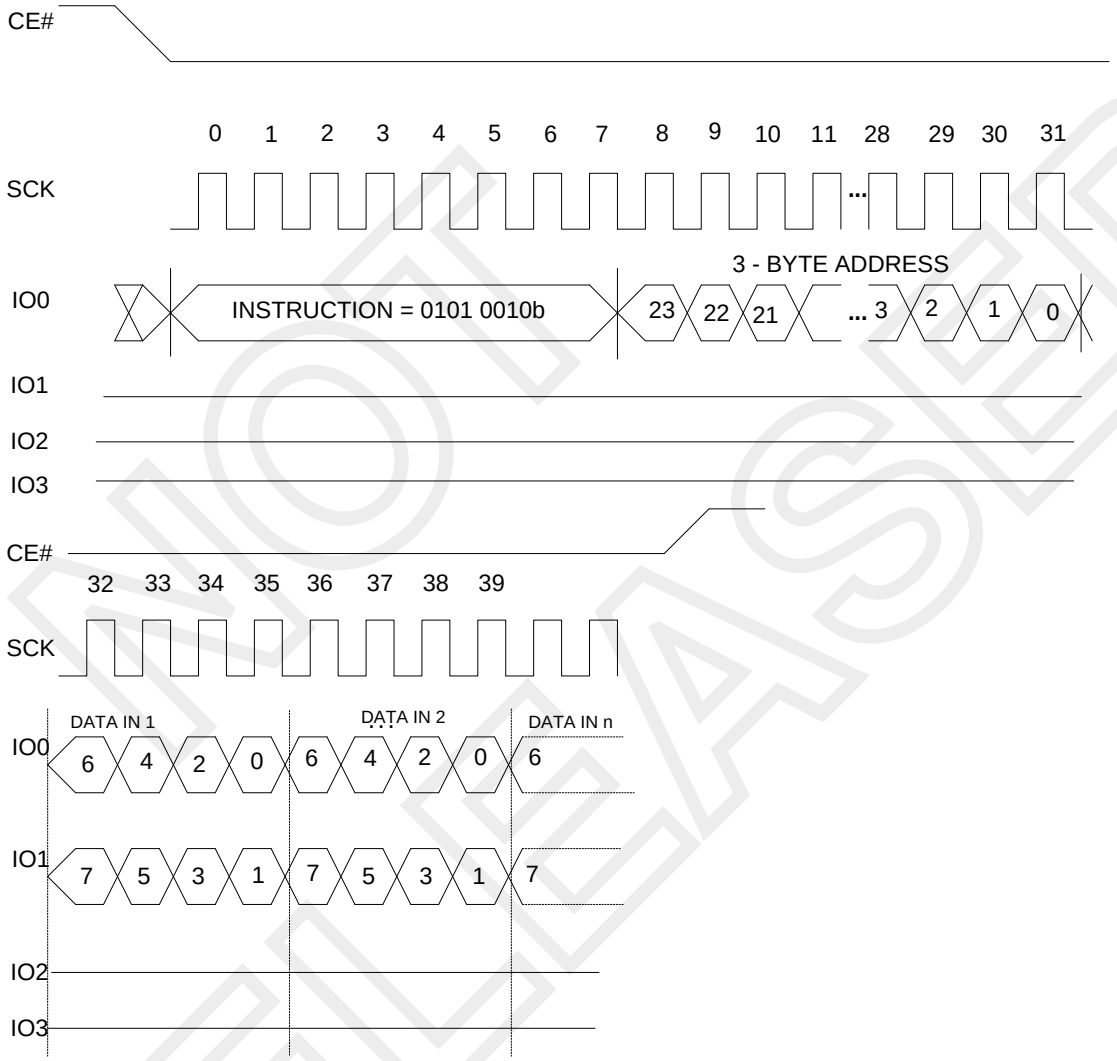
The Dual Input Page Program instruction code, three address bytes and program data (1 to 256 bytes) are input via the two pins (IO0, IO1). Program operation will start immediately after the CE# is brought high, otherwise the Dual Input Page Program instruction will not be executed. The internal control logic automatically handles the programming voltages and timing. During a program operation, all instructions will be ignored except the RDSR instruction. The progress or completion of the program operation can be determined by reading the WIP bit in Status Register via a RDSR instruction. If the WIP bit is "1", the program operation is still in progress. If WIP bit is "0", the program operation has completed.

If more than 256 bytes data are sent to a device, the address counter rolls over within the same page, the previously latched data are discarded, and the last 256 bytes data are kept to be programmed into the page. The starting byte can be anywhere within the page. When the end of the page is reached, the address will wrap around to the beginning of the same page. If the data to be programmed are less than a full page, the data of all other bytes on the same page will remain unchanged.

Note: A program operation can alter "1"s into "0"s, but an erase operation is required to change "0"s back to "1"s. A byte cannot be reprogrammed without first erasing the whole sector or block.

NOT RELEASED

**Figure 21. Dual Input Page Program Sequence**




NOT RELEASED

## DEVICE OPERATION (CONTINUED)

### ERASE OPERATION

The memory array of the IS25WQ080 is organized into uniform 4 Kbyte sectors or 64 Kbyte uniform blocks (a block consists of sixteen adjacent sectors).

Before a byte can be reprogrammed, the sector or block that contains the byte must be erased (erasing sets bits to "1"). In order to erase the devices, there are three erase instructions available: Sector Erase (SECTOR_ER), Block Erase (BLOCK_ER) and Chip Erase (CHIP_ER). A sector erase operation allows any individual sector to be erased without affecting the data in other sectors. A block erase operation erases any individual block. A chip erase operation erases the whole memory array of a device. A sector erase, block erase or chip erase operation can be executed prior to any programming operation.

### SECTOR_ER COMMAND (SECTOR ERASE) OPERATION

A SECTOR_ER instruction erases a 4 Kbyte sector Before the execution of a SECTOR_ER instruction, the Write Enable Latch (WEL) must be set via a Write Enable (WREN) instruction. The WEL bit is reset automatically after the completion of sector an erase operation.

A SECTOR_ER instruction is entered, after CE# is pulled low to select the device and stays low during the entire instruction sequence The SECTOR_ER instruction code, and three address bytes are input via SI. Erase operation will start immediately after CE# is pulled high. The internal control logic automatically handles the erase voltage and timing. Refer to Figure 22 for Sector Erase Sequence.

During an erase operation, all instruction will be ignored except the Read Status Register (RDSR) instruction. The progress or completion of the erase operation can be determined by reading the WIP bit in the Status Register using a RDSR instruction. If the WIP bit is "1", the erase operation is still in progress. If the WIP bit is "0", the erase operation has been completed.

### BLOCK_ER COMMAND (BLOCK ERASE) OPERATION

A Block Erase (BLOCK_ER) instruction erases a 32K/64 Kbyte block of the IS25WQ080. Before the execution of a BLOCK_ER instruction, the Write Enable Latch (WEL) must be set via a Write Enable (WREN) instruction. The WEL is reset automatically after the completion of a block erase operation.

The BLOCK_ER instruction code and three address bytes are input via SI. Erase operation will start immediately after the CE# is pulled high, otherwise the BLOCK_ER instruction will not be executed. The internal control logic automatically handles the erase voltage and timing. Refer to Figure 23 for Block Erase Sequence.

### CHIP_ER COMMAND (CHIP ERASE) OPERATION

A Chip Erase (CHIP_ER) instruction erases the entire memory array of a IS25WQ080. Before the execution of CHIP_ER instruction, the Write Enable Latch (WEL) must be set via a Write Enable (WREN) instruction. The WEL is reset automatically after completion of a chip erase operation.

The CHIP_ER instruction code is input via the SI. Erase operation will start immediately after CE# is pulled high, otherwise the CHIP_ER instruction will not be executed. The internal control logic automatically handles the erase voltage and timing. Refer to Figure 24 for Chip Erase Sequence.

NOT RELEASED

## DEVICE OPERATION (CONTINUED)

**Figure 22. Sector Erase Sequence**

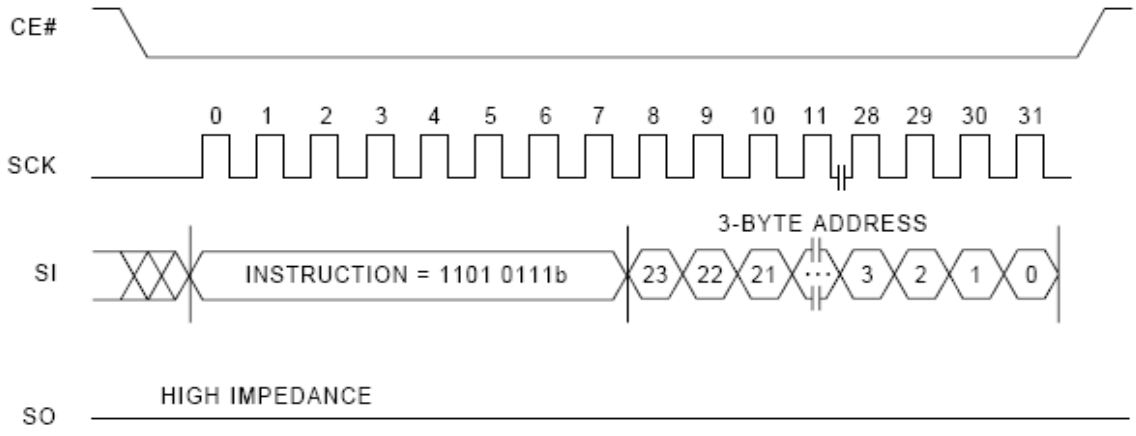



**Figure 23. Block Erase Sequence**

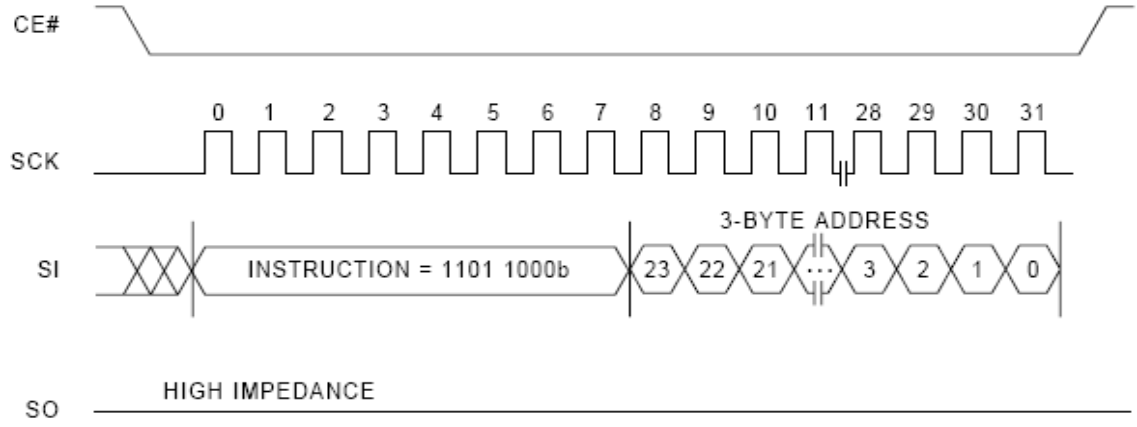
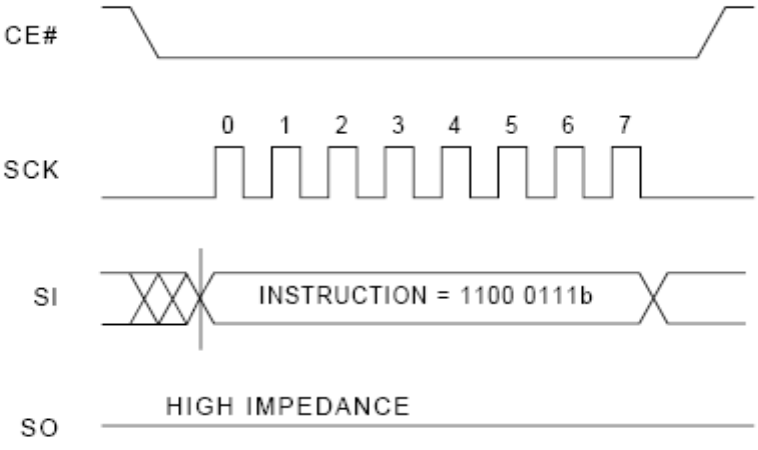



**Figure 24. Chip Erase Sequence**



## DEVICE OPERATION (CONTINUED)

### Deep Power Down

The Power-down (DP) instruction is for setting the device on the minimizing the power consumption (enter into Power-Down mode), the standby current is reduced from Isb1 to Isb2). During the Power-down mode, the device is not active and all Write/Program/Erase instruction are ignored. The instruction is initiated by driving the CE# pin low and shifting the instruction code “B9h” as show in the figure 25. The CE# pin must be driven high after the instruction has been latched. If this is not done the Power-Down will not be executed. After CE# pin driven high, the power-down state will entered within the time duration of $t_{DPD}$. While in the power-down state only the Release from Power-down / RDID instruction, which restores the device to normal operation, will be recognized. All other instructions are ignored. This includes the Read Status Register instruction, which is always available during normal operation. Ignoring all but one instruction makes the Power Down state a useful condition for securing maximum write protection.

**Figure 25. Power Down Sequence**

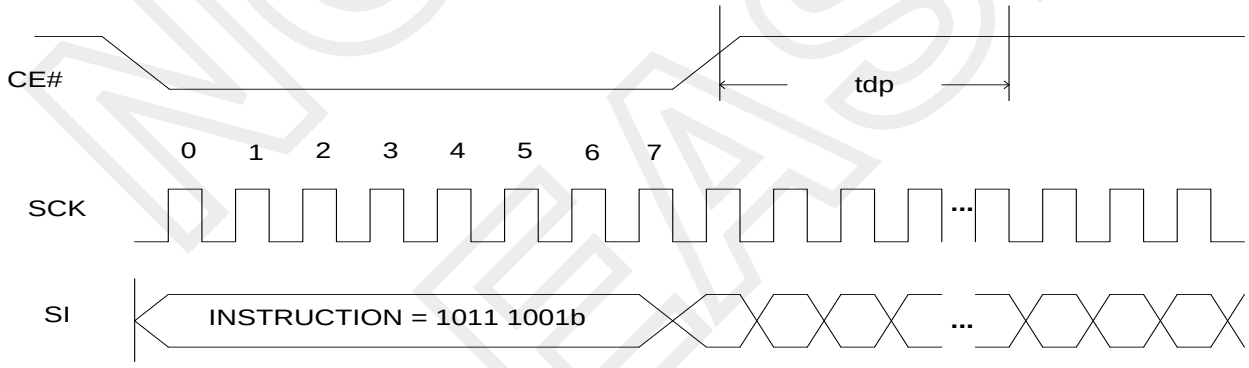
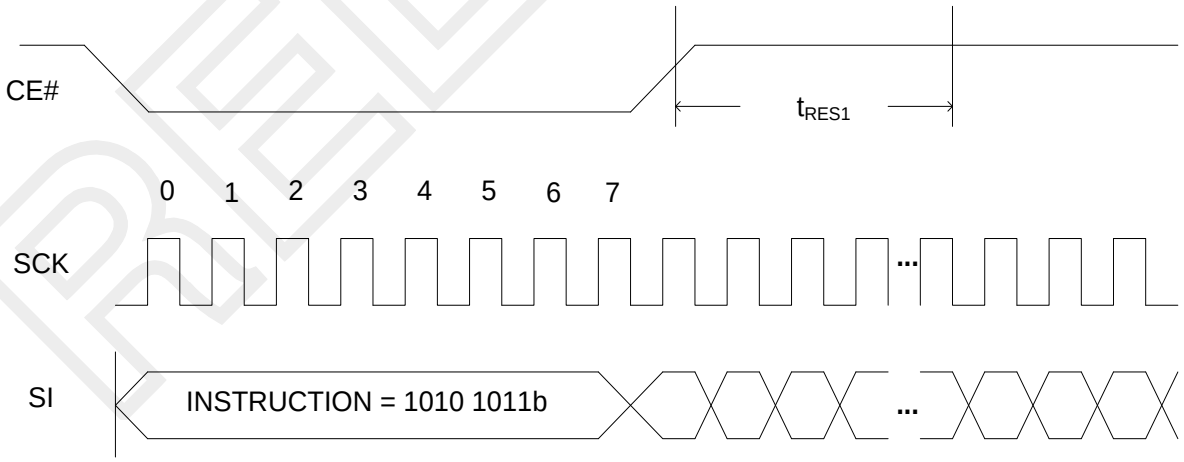



**Figure 26. Release Power Down Sequence**




NOT RELEASED

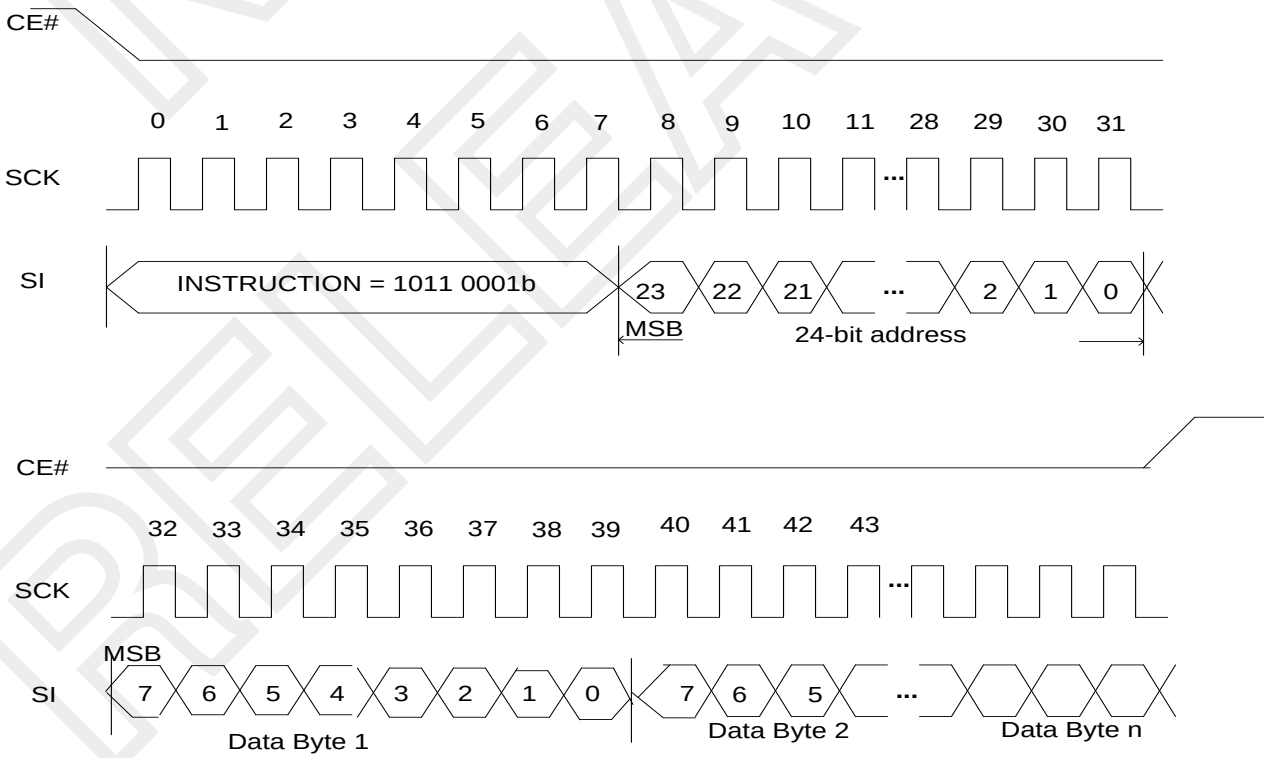
## DEVICE OPERATION (CONTINUED)

### Program Security information Row instruction (PSIR)

The PSIR instructions can read and programmed (Erase) using three dedicated instructions. The program information Raw instruction is used to program at most 256 bytes to the security memory area (by changing bits from '1' to '0', only). Before it can be accepted, a write enable (WREN) instruction must previously have been executed. After the write enable (WREN) instruction has been decoded, the device sets the write enable latch (WEL) bit. The program information Row instruction is entered by driving CE# pin Low, followed by the instruction code, three address bytes and at least one data byte on serial data input (SI). CE# pin must be driven High after the eighth bits of the last data byte has been latched in, otherwise the Program information Row instruction is not executed. If more than 256 bytes data are sent to a device, the address counter can not roll over.

After CE# pin is driven High, the self-timed page program cycle (whose duration is $t_{potp}$) is initiated. While the program OTP cycle is in progress, the status register may be read to check the value of the write in progress (WIP) bit. The write in progress (WIP) bit is 1 during the self-timed program cycle, and it is 0 when it is completed. At some unspecified time before the cycle is complete, the write enable latch (WEL) bit is reset.

**Figure 27. Program information Raw Sequence**




Note: 1 ≤ n ≤ 256

Note: 1. The SIR address is from 000000h to 0000FFh.
2. The SIR protection bit is in the address 000100h



NOT RELEASED

## DEVICE OPERATION (CONTINUED)

**To lock the OTP memory:**

Bit 0 of the OTP control byte, that is byte256, is used to permanently lock the OTP memory array.

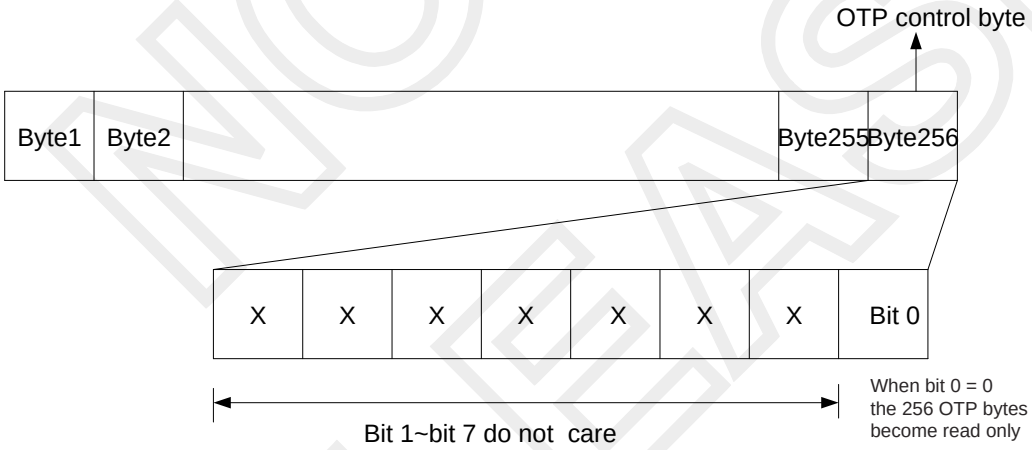
- When bit 0 of byte 256 = '1', the 256 bytes of the OTP memory array can be programmed.
- When bit 0 of byte 256 = '0', the 256 bytes of the OTP memory array are read-only and cannot be programmed anymore.

Once a bit of the OTP memory has been programmed to '0', it can no longer be set to '1'.

Therefore, as soon as bit 0 of byte 256 (control byte) is set to '0', the 256 bytes of the OTP memory array become read-only in a permanent way.

Any program OTP (POTP) instruction issued while an erase, program or write cycle is in progress is rejected without having any effect on the cycle that is in progress

**Figure 28. OTP area**



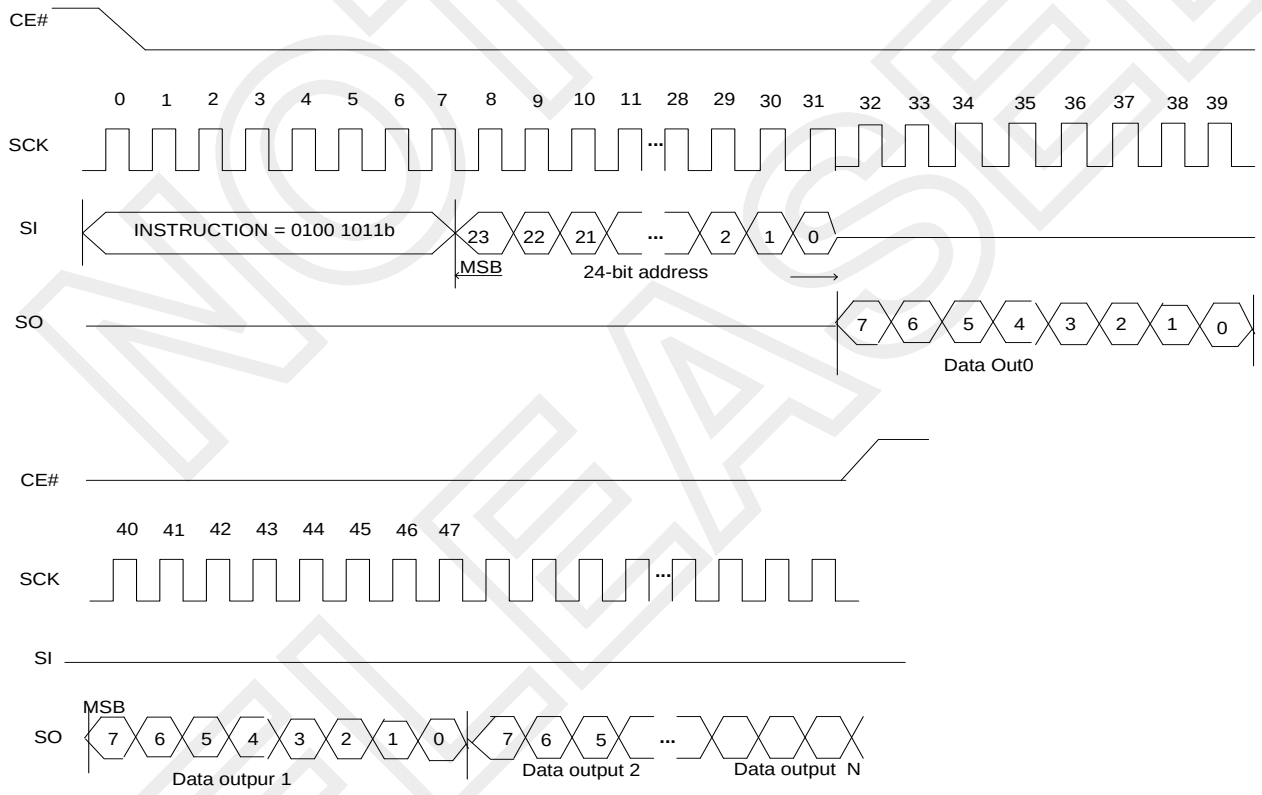

NOT RELEASED

## DEVICE OPERATION (CONTINUED)

### Read Security Information Row (RSIR)

The RSIR instruction read the security information Row. There is no rollover mechanism with the read OTP (ROTP) instruction. This means that the read OTP (ROTP) instruction must be sent with a maximum of 256 bytes to read, since once the 256th byte has been read, the same (256th) byte keeps being read on the SO pin.

**Fig 29. Read Security information Row instruction**




NOT RELEASED

## DEVICE OPERATION (CONTINUED)

### Program/Erase Suspend Resume

The device allow the interruption of Sector-Erase, Block-Erase or Page-Program operations and conduct other operations.
To enter the suspend/ resume mode: issuing 75h/B0h for suspend; 7Ah/30h for resume
Read function register bit2 (PSUS) and bit1 (ESUS) to check suspend ready information.
Suspend to suspend ready timing: 20us.
Resume to another suspend timing: 1ms.

*Note: It needs 500ns delay time from write command to suspend command

**Program/Erase Suspend During Sector-Erase or Block-Erase**

After erase suspend, WEL bit will be clear, only read related, resume and reset command can be accepted.
(03h, 0Bh, BBh, Ebh, 05h, Abh, 9Fh, 90h, 4Bh)
To execute a Program/Erase Suspend operation, the host drives CE# low, sends the Program/Erase Suspend command cycle (75h/B0H), then drives CE# high. The Function register indicates that the erase has been suspended by changing the ESUS bit from '0' to '1,' but the device will not accept another command until it is ready. To determine when the device will accept a new command, poll the WIP bit in the Status register or wait TWS. When ESUS bit is issued, the Write Enable Latch (WEL) bit will be reset.

**Program/Erase Suspend During Page Programming**

Program suspend allows the interruption of all program operations.
After program suspend, WEL bit will be cleared, only read related, resume and reset command can be accepted.
(03h, 0Bh, BBh, Ebh, 05h, Abh, 9Fh, 90h, 4Bh)
To execute a Program/Erase Suspend operation, the host drives CE# low, sends the Program/Erase Suspend command cycle (75h/B0H), then drives CE# high. The Function register indicates that the programming has been suspended by changing the PSUS bit from '0' to '1,' but the device will not accept another command until it is ready. To determine when the device will accept a new command, poll the WIP bit in the Status register or wait TWS.

**Program/Erase Resume**

Program/Erase Resume restarts a Program/Erase command that was suspended, and changes the suspend status bit in the (ESUS or PSUS) back to '0'.
To execute a Program/Erase Resume operation, the host drives CE# low, sends the Program/Erase Resume command cycle (7Ah/30H), then drives CE# high. To determine if the internal, self-timed Write operation completed, poll the WIP bit in the Status register, or wait the specified time TSE, TBE or TPP for Sector- Erase, Block-Erase, or Page-Programming, respectively. The total write time before suspend and after resume will not exceed the uninterrupted write times TSE, TBE or TPP.

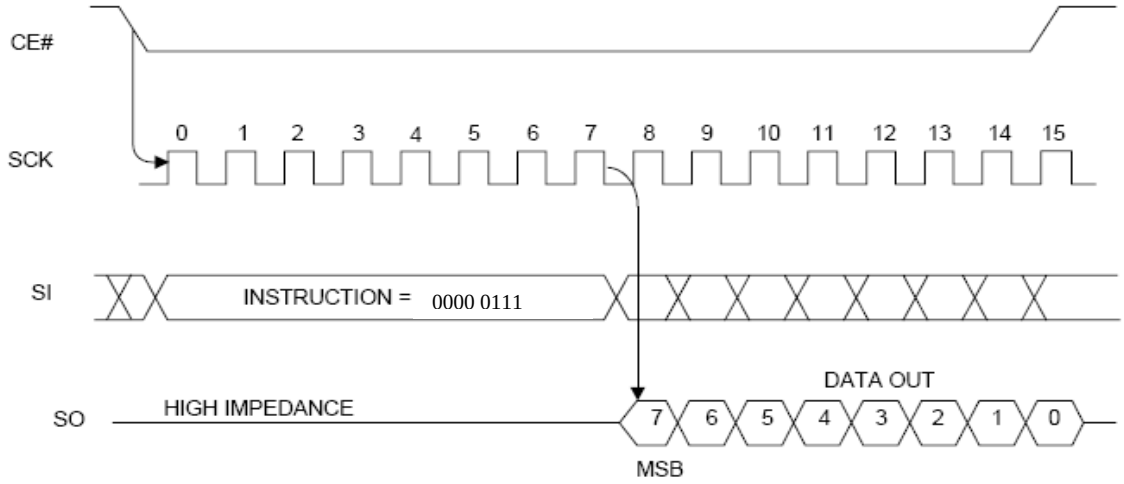
NOT RELEASED

## DEVICE OPERATION (CONTINUED)

### RDFR COMMAND (READ FUNCTION REGISTER) OPERATION

The Read Function Register (RDFR) instruction provides access to the Erase/Program suspend register. During the execution of a program, erase or write status register suspend, which can be used to check the suspend status.

**Figure 30. Read Function Register Sequence**




## ABSOLUTE MAXIMUM RATINGS (1)

| | | |
|---|---|---|
| Storage Temperature | | -65°C to +150°C |
| Surface Mount Lead Soldering Temperature | Standard Package | 240°C 3 Seconds |
| | Lead-free Package | 260°C 3 Seconds |
| Input Voltage with Respect to Ground on All Pins (2) | | -0.5 V to VCC + 0.5 V |
| All Output Voltage with Respect to Ground | | -0.5 V to VCC + 0.5 V |
| VCC (2) | | -0.5 V to +4.0 V |

Notes:
1. Applied conditions greater than those listed in “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only. The functional operation of the device conditions that exceed those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating condition for extended periods may affect device reliability.

2. Maximum DC voltage on input or I/O pins is $V_{CC}$ + 0.5 V. During voltage transitions, input or I/O pins may overshoot $V_{CC}$ by + 2.0 V for a period of time not to exceed 20 ns. Minimum DC voltage on input or I/O pins is -0.5 V. During voltage transitions, input or I/O pins may undershoot GND by -2.0 V for a period of time not to exceed 20 ns.

## DC AND AC OPERATING RANGE

| Part Number | IS25WQ080 |
|---|---|
| Operating Temperature (Extended Grade) | -40°C to 105°C |
| Operating Temperature (Industrial Grade) | -40°C to 85°C |
| Operating Temperature (Automotive, A1 Grade) | -40°C to 85°C |
| Operating Temperature (Automotive, A2 Grade) | -40°C to 105°C |
| Operating Temperature (Automotive, A3 Grade) | -40°C to 125°C |
| Vcc Power Supply | 1.65 V –1.95 V |

## DC CHARACTERISTICS

Applicable over recommended operating range from:
$V_{CC}$ = 1.65 V to 1.95 V (unless otherwise noted).

| Symbo | Parameter | Condition | | Min | Typ | Max | Units |
|---|---|---|---|---|---|---|---|
| $I_{CC1}$ | Vcc Active Read Current | $V_{CC}$ = 1.95 V at 33 MHz, SO = Open | | | 10 | 15 | mA |
| $I_{CC2}$ | Vcc Program/Erase Current | $V_{CC}$ = 1.95 V at 33 MHz, SO = Open | | | 15 | 20 | mA |
| $I_{SB1}$ | Vcc Standby Current CMOS | $V_{CC}$ = 1.95 V, CE# = $V_{CC}$ | | | | 50 | □A |
| $I_{SB2}$ | Vcc Standby Current TTL | $V_{CC}$ = 1.95 V, CE# = $V_{IH}$ to $V_{CC}$ | | | | 3 | mA |
| $I_{LI}$ | Input Leakage Current | $V_{IN}$ = 0V to $V_{CC}$ | | | | 1 | □A |
| $I_{LO}$ | Output Leakage Current | $V_{IN}$ = 0V to $V_{CC}$, $T_{AC}$ = 0°C to 130°C | | | | 1 | □A |
| $V_{IL}$ | Input Low Voltage | | | -0.5 | | 0.3Vcc | V |
| $V_{IH}$ | Input High Voltage | | | 0.7$V_{CC}$ | | $V_{CC}$ + 0.4 | V |
| $V_{OL}$ | Output Low Voltage | 1.65V < $V_{CC}$ < 1.95 V | $I_{OL}$ = 100 □A | | | 0.2 | V |
| $V_{OH}$ | Output High Voltage | | $I_{OH}$ = -100 □A | $V_{CC}$ – 0.2 | | | V |

NOT RELEASED

## AC CHARACTERISTICS

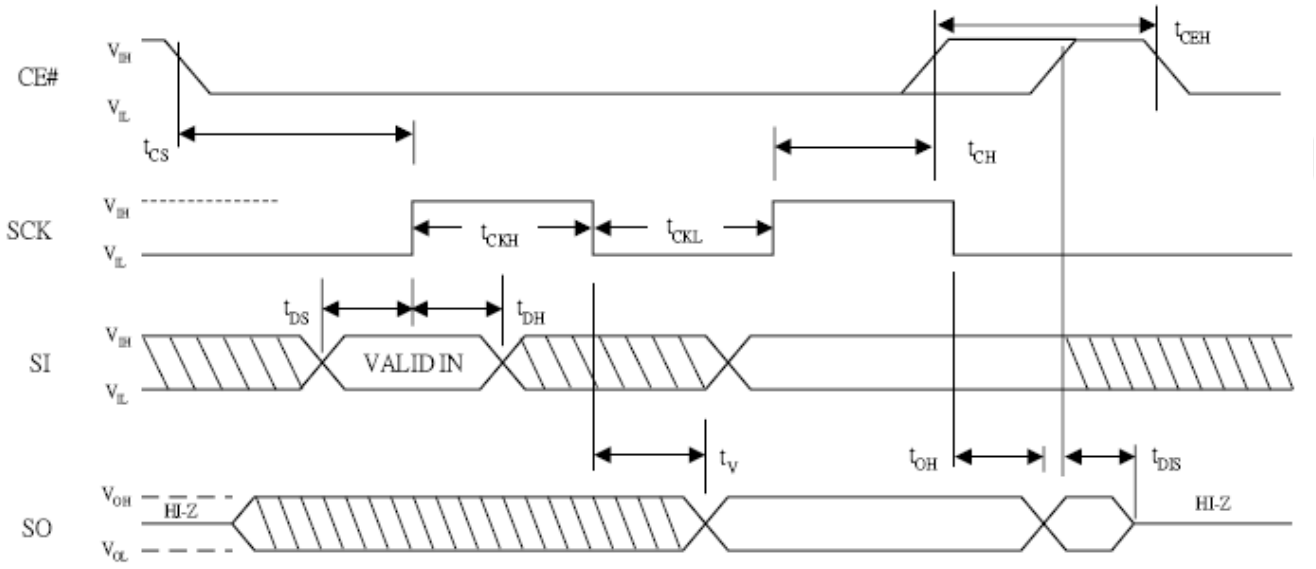
Applicable over recommended operating range from $V_{CC}$ = 1.65 V to 1.95 V
$C_L$ = 1 TTL Gate and 30 pF (unless otherwise noted).

| Symbol | Parameter | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| $f_{CT}$ | Clock Frequency for fast read mode | 0 | | 104 | MHz |
| $f_C$ | Clock Frequency for read mode | 0 | | 33 | MHz |
| $t_{RI}$ | Input Rise Time | | | 8 | ns |
| $t_{FI}$ | Input Fall Time | | | 8 | ns |
| $t_{CKH}$ | SCK High Time | 4 | | | ns |
| $t_{CKL}$ | SCK Low Time | 4 | | | ns |
| $t_{CEH}$ | CE# High Time | 25 | | | ns |
| $t_{CS}$ | CE# Setup Time | 10 | | | ns |
| $t_{CH}$ | CE# Hold Time | 5 | | | ns |
| $t_{DS}$ | Data In Setup Time | 2 | | | ns |
| $t_{DH}$ | Data in Hold Time | 2 | | | ns |
| $t_{HS}$ | Hold Setup Time | 15 | | | ns |
| $t_{HD}$ | Hold Time | 15 | | | ns |
| $t_V$ | Output Valid | | | 8 | ns |
| $t_{OH}$ | Output Hold Time Normal Mode | 0 | | | ns |
| $t_{LZ}$ | Hold to Output Low Z | | | 200 | ns |
| $t_{HZ}$ | Hold to Output High Z | | | 200 | ns |
| $t_{DIS}$ | Output Disable Time | | | 100 | ns |
| $t_{EC}$ | Sector Erase Time | | 70 | 150 | ms |
| | Block Erase Time (32KB) | | 0.12 | 0.5 | s |
| | Block Erase Time(64KB) | | 0.15 | 0.5 | s |
| | Chip Erase Time (8Mb) | | 2 | 5 | s |
| | Chip Erase Time (16Mb) | | 5 | 12 | s |
| $t_{PP}$ | Page Program Time | | 0.6 | 0.7 | ms |
| $t_{VCS}$ | $V_{CC}$ Set-up Time | 50 | | | □s |
| $t_{res1}$ | Release Deep power down | | | 10 | □s |
| $t_{dp}$ | Deep power down | | | 10 | □s |
| $t_W$ | Write Status Register time | | 10 | 15 | ms |
| tsus | Suspend to suspend ready | | | 20 | us |
| trs | Resume to another suspend | | | 1 | ms |
| tsrst | Software Reset cover time | | | 15 | ms |
| Tws | Suspend time | | | 20 | us |

NOT RELEASED

## AC CHARACTERISTICS (CONTINUED)

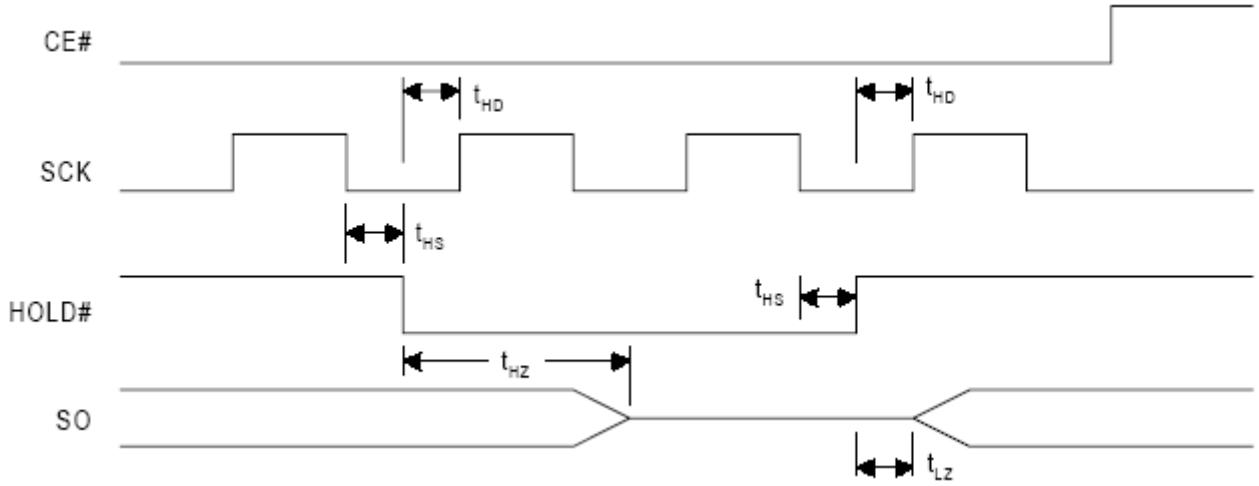
### SERIAL INPUT/OUTPUT TIMING (1)




Note: 1. For SPI Mode 0 (0,0)

## AC CHARACTERISTICS (CONTINUED)

### HOLD TIMING

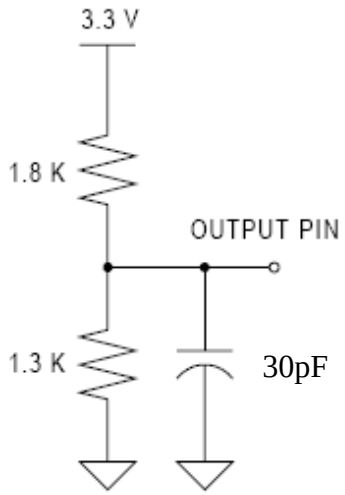



**PIN CAPACITANCE** (f = 1 MHz, T = 25°C )

| | Typ | Max | Units | Conditions |
|---|---|---|---|---|
| $C_{IN}$ | 4 | 6 | pF | $V_{IN}$ = 0 V |
| $C_{OUT}$ | 8 | 12 | pF | $V_{OUT}$ = 0 V |

Note: These parameters are characterized but not 100% tested.

**OUTPUT TEST LOAD**




**INPUT TEST WAVEFORMS AND MEASUREMENT LEVEL**




Note: 1. Input Pulse Voltage : 0.2Vcc to 0.8Vcc.
2. Input Timing Reference Voltages :
0.3Vcc to 0.7Vcc.
3. Output Timing Reference Voltage : Vcc/2.

**POWER-UP AND POWER-DOWN**

At Power-up and Power-down, the device must not be selected (CE# must follow the voltage applied on Vcc) until Vcc reaches the correct value:

1. Vcc(min) at Power-up, and then for a further delay of
tVCE
0 Vss at Power-down

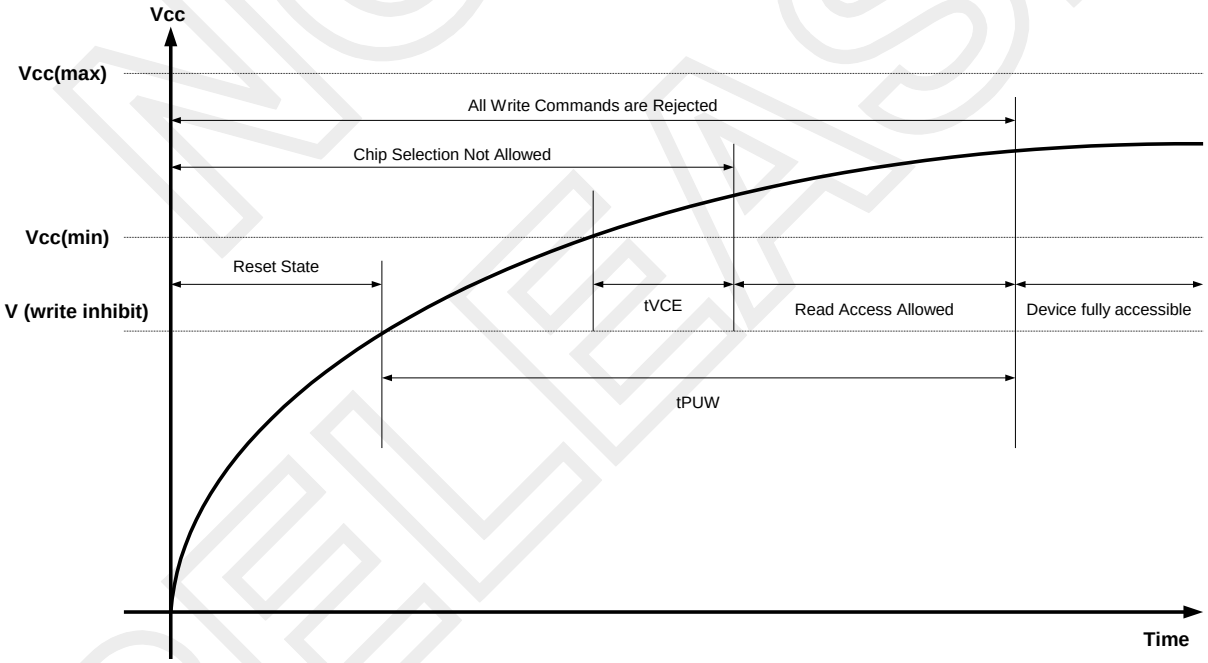
Usually a simple pull-up resistor on CE# can be used to insure safe and proper Power-up and Power-down. To avoid data corruption and inadvertent write operations during power up, a Power On Reset (POR) circuit is included. The logic inside the device is held reset while Vcc is less than the POR threshold value (Vwi) during power up, the device does not respond to any instruction until a time delay of tPUW has elapsed after the moment that Vcc rised above the VWI threshold. However, the correct operation of the device is not guaranteed if, by this time, Vcc is still below Vcc(min). No Write Status Register, Program or Erase instructions should be sent until the later of:

- tPUW after Vcc passed the VWI threshold
- tVCE after Vcc passed the Vcc(min) level

At Power-up, the device is in the following state:

- The device is in the Standby mode
- The Write Enable Latch (WEL) bit is reset

At Power-down, when Vcc drops from the operating voltage, to below the Vwi, all write operations are disabled
and the device does not respond to any write instruction.




| Symbol | Parameter | Min. | Max. | Unit |
|---|---|---|---|---|
| $t_{VCE}$ *1 | Vcc(min) to CE# Low | 10 | | us |
| $t_{PUW}$ *1 | Power-Up time delay to Write instruction | 1 | 10 | ms |
| $V_{WI}$ *1 | Write Inhibit Voltage | 1.3 | | V |
| Note : *1. These parameters are characterized only. | | | | |



NOT RELEASED

## PROGRAM/ERASE PERFORMANCE

| Parameter | Unit | Typ | Max | Remarks |
|---|---|---|---|---|
| Sector Erase Time | ms | 70 | 150 | From writing erase command to erase completion |
| Block Erase Time | s | 0.15 | 0.5 | From writing erase command to erase completion |
| Chip Erase Time (16Mb) | s | 5 | 12 | From writing erase command to erase completion |
| Chip Erase Time (8Mb) | s | 2 | 6 | From writing erase command to erase completion |
| Page Programming Time | ms | 0.6 | 0.7 | From writing program command to program completion |
| Byte Program | us | 8 | 25 | First byte |
| Additional Byte program | us | 2.5 | 12 | After first byte |

Note: These parameters are characterized and are not 100% tested.

## RELIABILITY CHARACTERISTICS

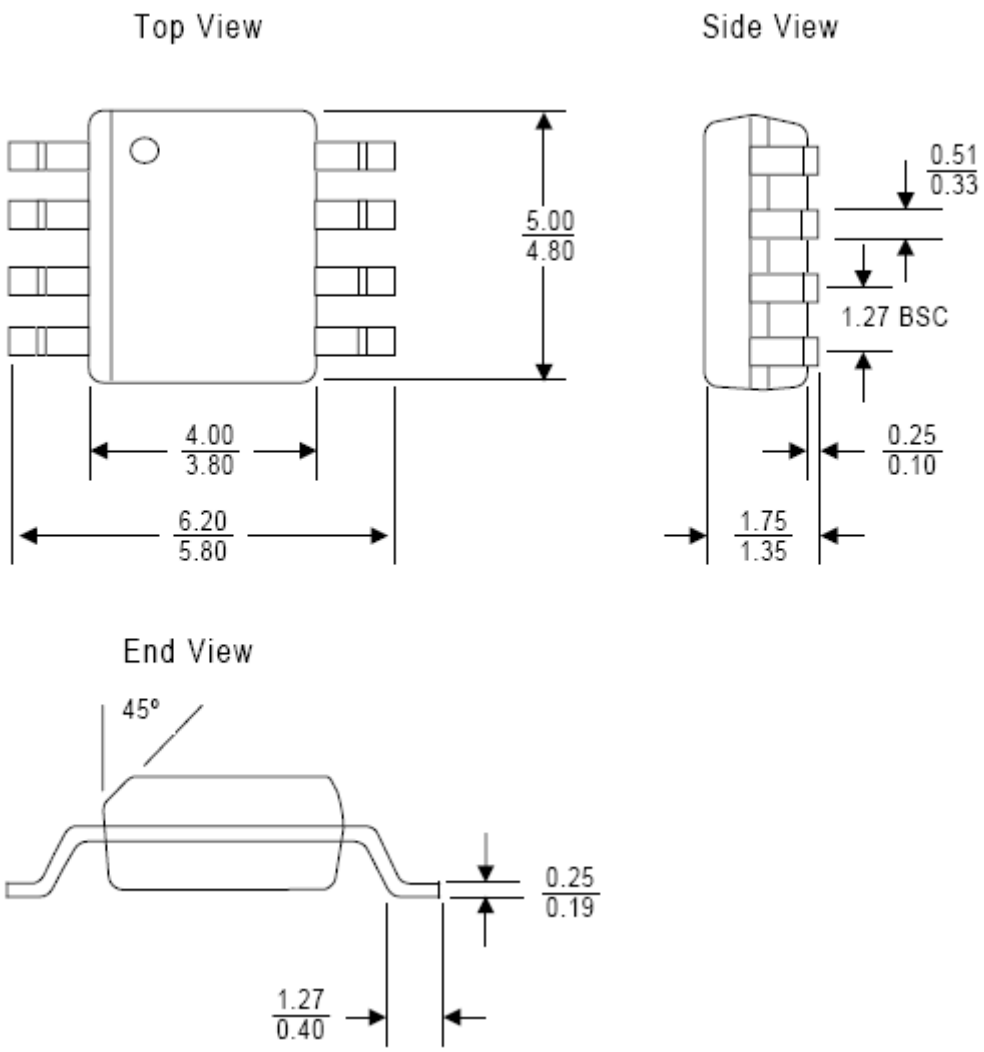
| Parameter | Min | Typ | Unit | Test Method |
|---|---|---|---|---|
| Endurance | 100,000 | | Cycles | JEDEC Standard A117 |
| Data Retention | 20 | | Years | JEDEC Standard A103 |
| ESD – Human Body Model | 2,000 | | Volts | JEDEC Standard A114 |
| ESD – Machine Model | 200 | | Volts | JEDEC Standard A115 |
| Latch-Up | 100 + $I_{CC1}$ | | mA | JEDEC Standard 78 |

Note: These parameters are characterized and are not 100% tested.

NOT RELEASED

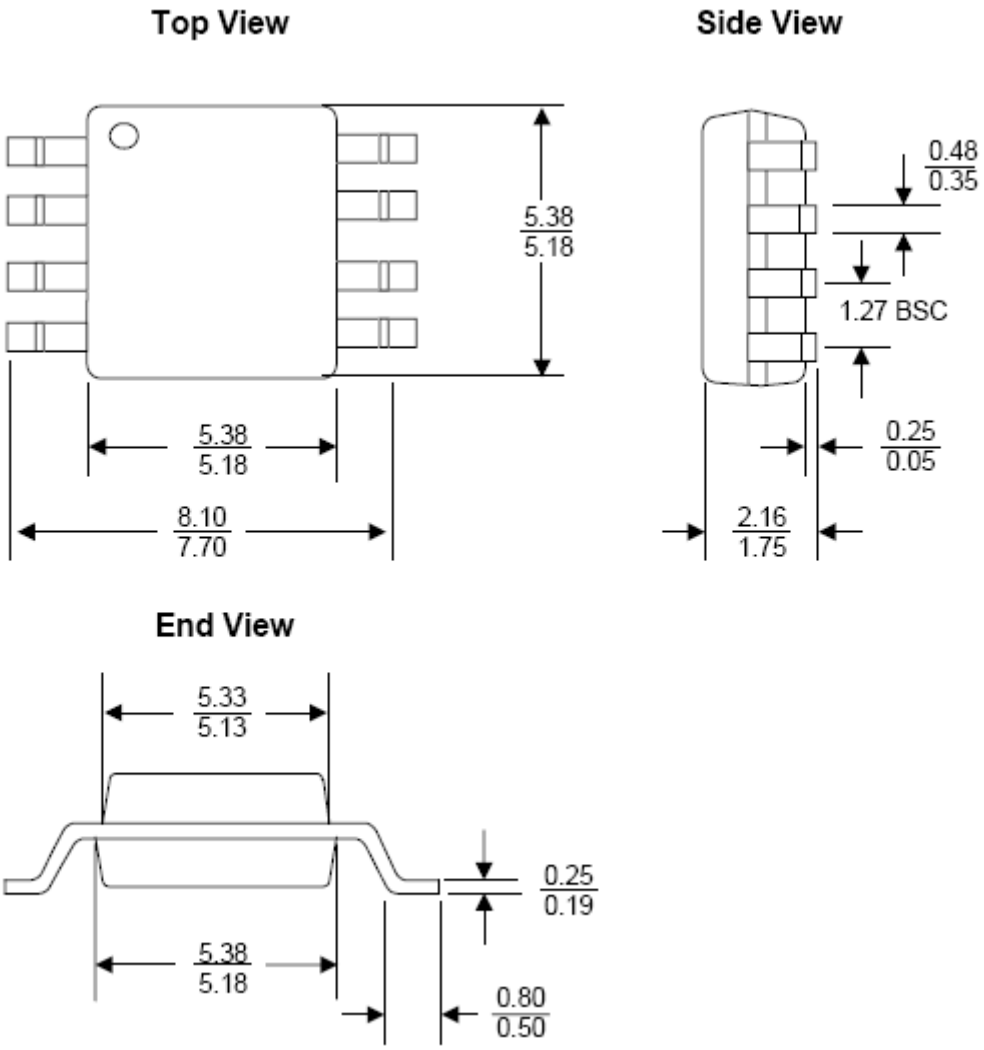
## PACKAGE TYPE INFORMATION

**JN**
**8-Pin JEDEC 150mil Broad Small Outline Integrated Circuit (SOIC) Package**
**(measure in millimeters)**



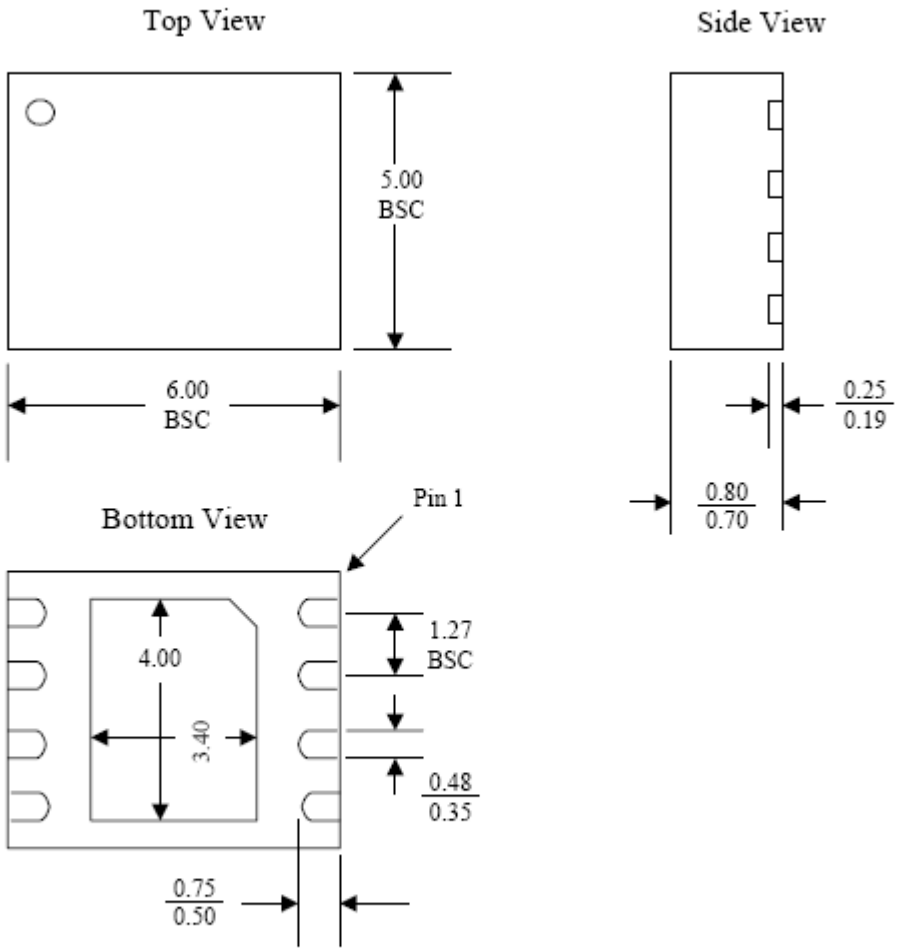
## PACKAGE TYPE INFORMATION (CONTINUED)

**JB**
**8-Pin JEDEC 208mil Broad Small Outline Integrated Circuit (SOIC) Package**
**(measure in millimeters)**



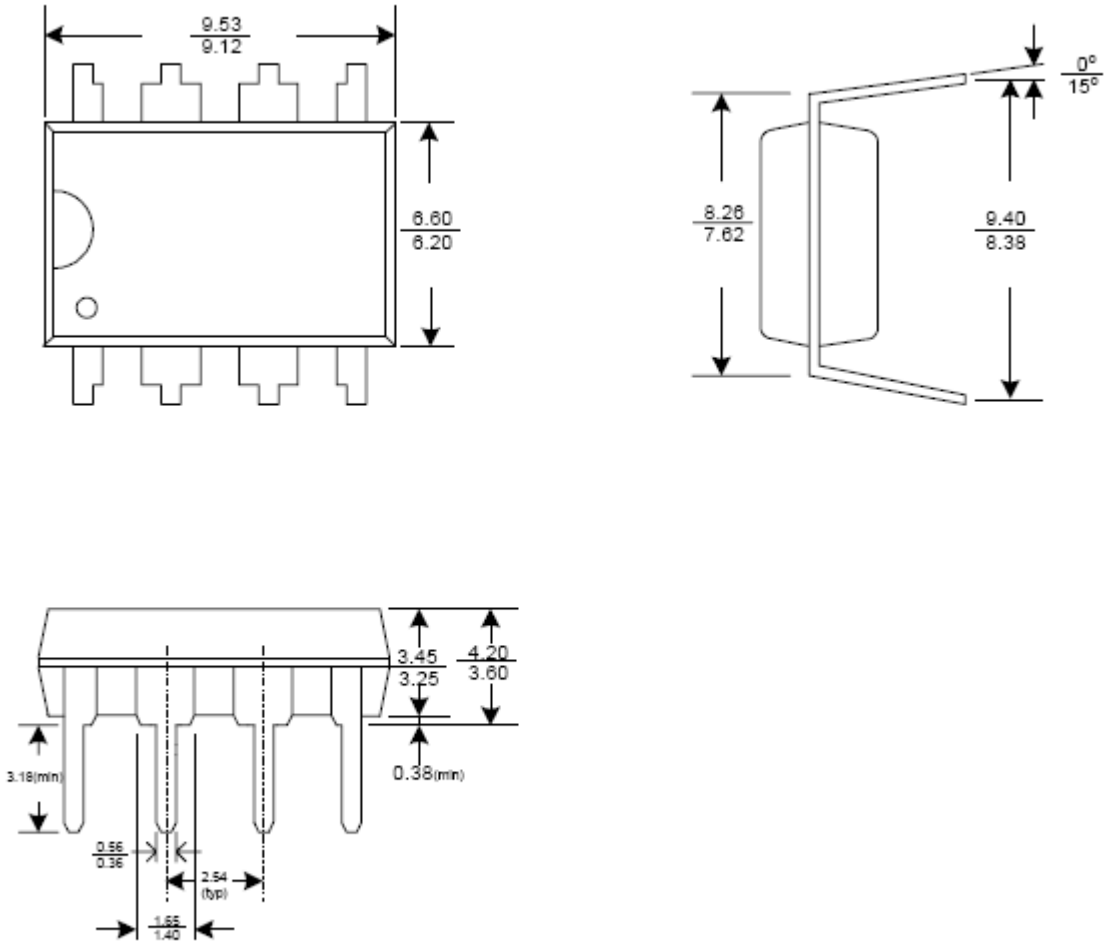
## PACKAGE TYPE INFORMATION (CONTINUED)

**JK**
**8-pin Ulta-Thin Small Outline No-Lead (WSON) Package (measure in millimeters)**



## PACKAGE TYPE INFORMATION (CONTINUED)

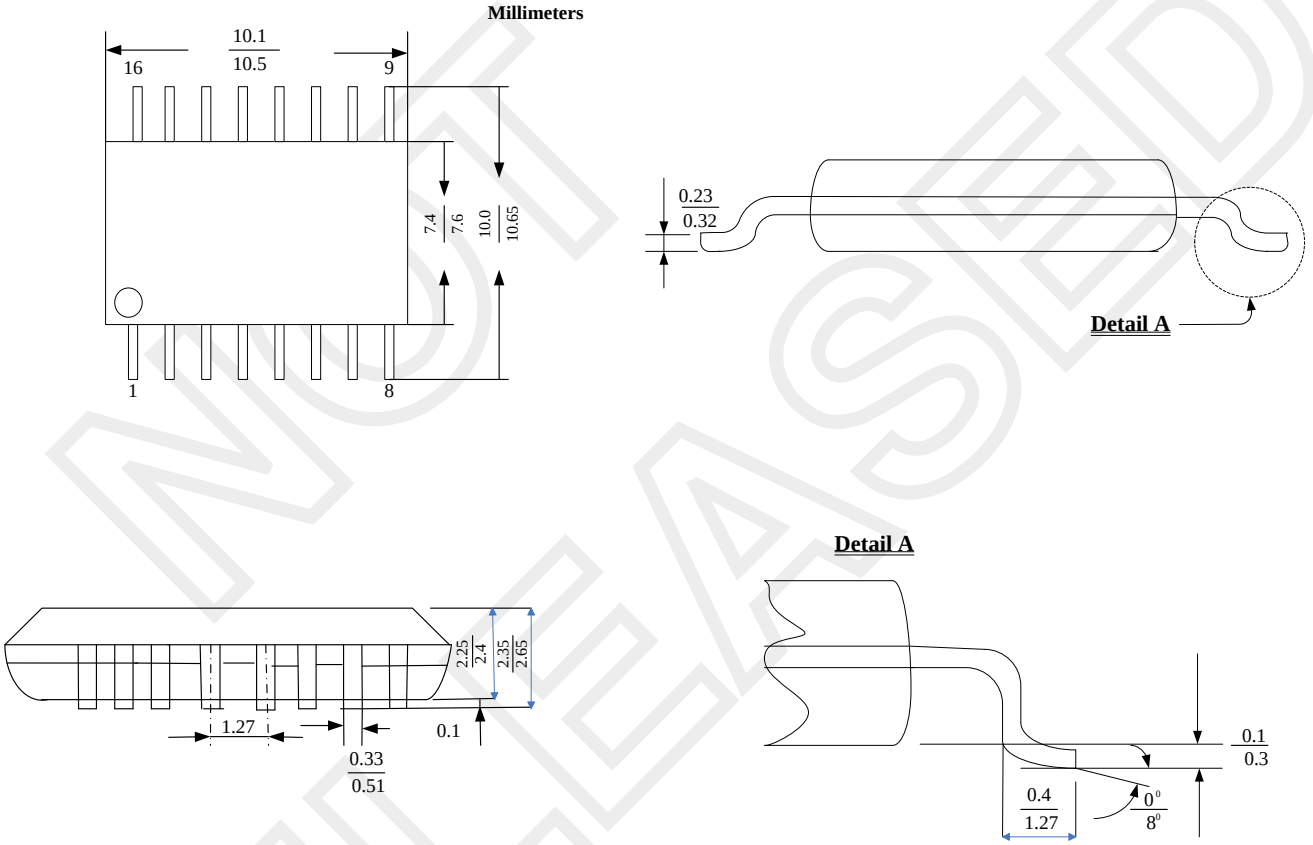
**JA**
**8-pin 300mil wide body, Plastic Dual In-Line Package PDIP (measure in millimeters)**



## PACKAGE TYPE INFORMATION (CONTINUED)

### JM

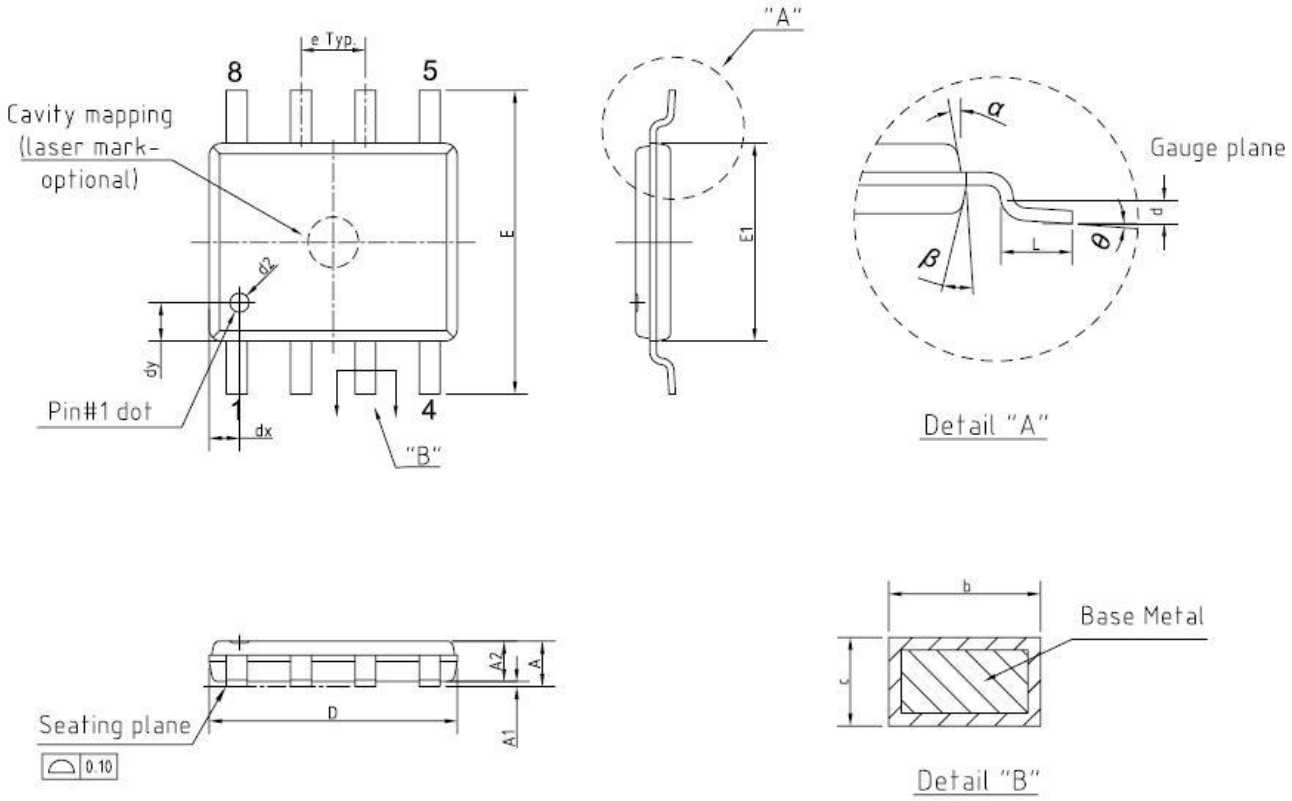
**16 pin – 16-lead Plastic Small Outline, 300 mils body width, package outline**




NOT RELEASED

## PACKAGE TYPE INFORMATION (CONTINUED)

### JV
### 8-pin VVSOP 150mil



| SYMBOL | DIMENSION (MM) | | | DIMENSION (MIL) | | |
|---|---|---|---|---|---|---|
| | MIN. | NOM. | MAX. | MIN. | NOM. | MAX. |
| A | --- | --- | 0.90 | --- | --- | 35.4 |
| A1 | 0.05 | 0.10 | 0.15 | 2.0 | 3.9 | 5.9 |
| A2 | 0.72 | 0.74 | 0.75 | 28.3 | 29.1 | 29.5 |
| b | 0.33 | 0.41 | 0.51 | 13.0 | 16.1 | 20.1 |
| c | 0.13 | --- | 0.18 | 5.1 | --- | 7.1 |
| D | 4.80 | 4.90 | 5.00 | 189.0 | 192.9 | 196.9 |
| d | --- | 0.25 | --- | --- | 10.0 | --- |
| d1 | --- | 0.50typ | --- | --- | 19.7typ | --- |
| dx | --- | 0.60typ | --- | --- | 23.6typ | --- |
| dy | --- | 0.76typ | --- | --- | 29.9typ | --- |
| E | 5.80 | 6.00 | 6.20 | 228.3 | 236.2 | 244.1 |
| E1 | 3.80 | 3.90 | 4.00 | 149.6 | 153.5 | 157.5 |
| e | --- | 1.27 | --- | --- | 50.0 | --- |
| L | 0.41 | 0.71 | 1.27 | 16.1 | 28.0 | 50.0 |
| $\alpha$ | --- | 10° | --- | --- | 10° | --- |
| $\beta$ | --- | 10° | --- | --- | 10° | --- |
| $\theta$ | 0° | --- | 10° | 0° | --- | 10° |

NOTES:
1. MOLD BODY DIMENSION DOES NOT INCLUDE END FLASH.
2. THE MOLDED END FLASH IS MAX. 0.15 mm.
3. LEAD WIDTH DIMENSION DOES NOT INCLUDE DAMBAR PROTRUSION / INTRUSION.

## Appendix1: Safe Guard function

Safe Guard function is a security function for customer to protect by sector (4Kbyte).
Every sector has one bit register to decide it will under safe guard protect or not. (“0”means protect and “1” means not protect by safe guard.) IS25WQ080 (sector 0~sector 255). IS25WQ016 (sector 0~sector 511)

## Mapping table for safe guard register

| | Address | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|
| Sector0 | 000h | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| Sector1 | 000h | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| Sector2 | 000h | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| Sector3 | 000h | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| Sector4 | 000h | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| Sector5 | 000h | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| Sector6 | 000h | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| Sector7 | 000h | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Sector8 | 001h | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| Sector9 | 001h | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| Sector10 | 001h | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| Sector11 | 001h | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| Sector12 | 001h | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| Sector13 | 001h | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| Sector14 | 001h | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| Sector15 | 001h | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| Sector248 | 01Fh | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| Sector249 | 01Fh | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| Sector250 | 01Fh | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| Sector251 | 01Fh | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| Sector252 | 01Fh | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| Sector253 | 01Fh | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| Sector254 | 01Fh | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| Sector255 | 01Fh | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

## Read Safe Guard register

The READ Safe Guard instruction code is transmitted via the SIO line, followed by three address bytes (A23 – A0) of the first register location to be read. The first byte data (D7 – D0) addressed is then shifted out on the SO line, MSb first. The address is automatically incremented after each byte of data is shifted out. The read operation can be terminated at any time by driving CE# high ($V_{IH}$) after the data comes out.

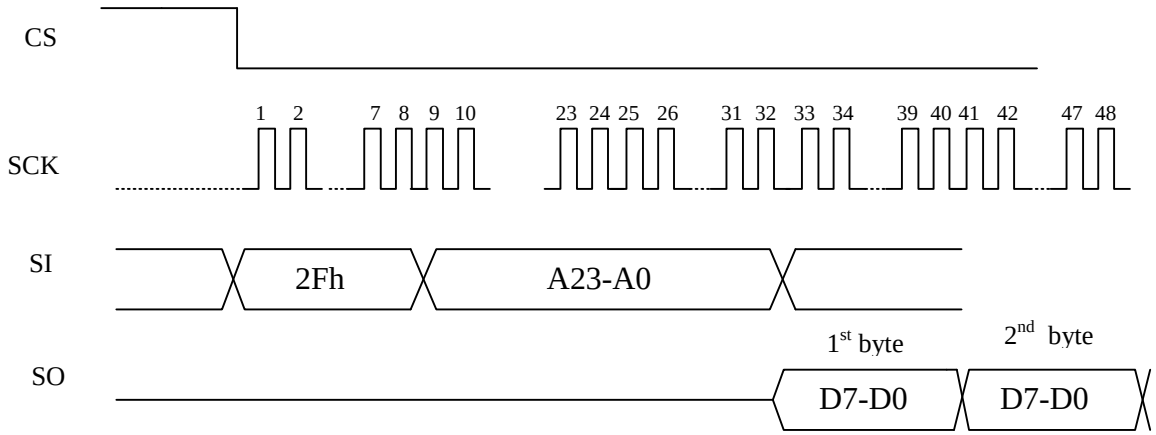



Fig a. Timing waveform of Read Safe guard register

## Erase Safe Guard register

If we want to erase the safe guard register to let the flash into unprotect status, it needs five continuous instructions. If any instruction is wrong, the erase command will be ignored. Erase wait time follow product erase timing spec.
Fig b. shows the complete steps for Erase safe guard register.

## Program Safe Guard register

If we want to erase the safe guard register to let the flash into unprotect status, it needs five continuous instructions. If any instruction is wrong, the program command will be ignored. The Program safe guard instruction allows up to 256 bytes data to be programmed into memory in a single operation. Program wait time follow product program timing spec.

Fig c. shows the complete steps for program safe guard register.

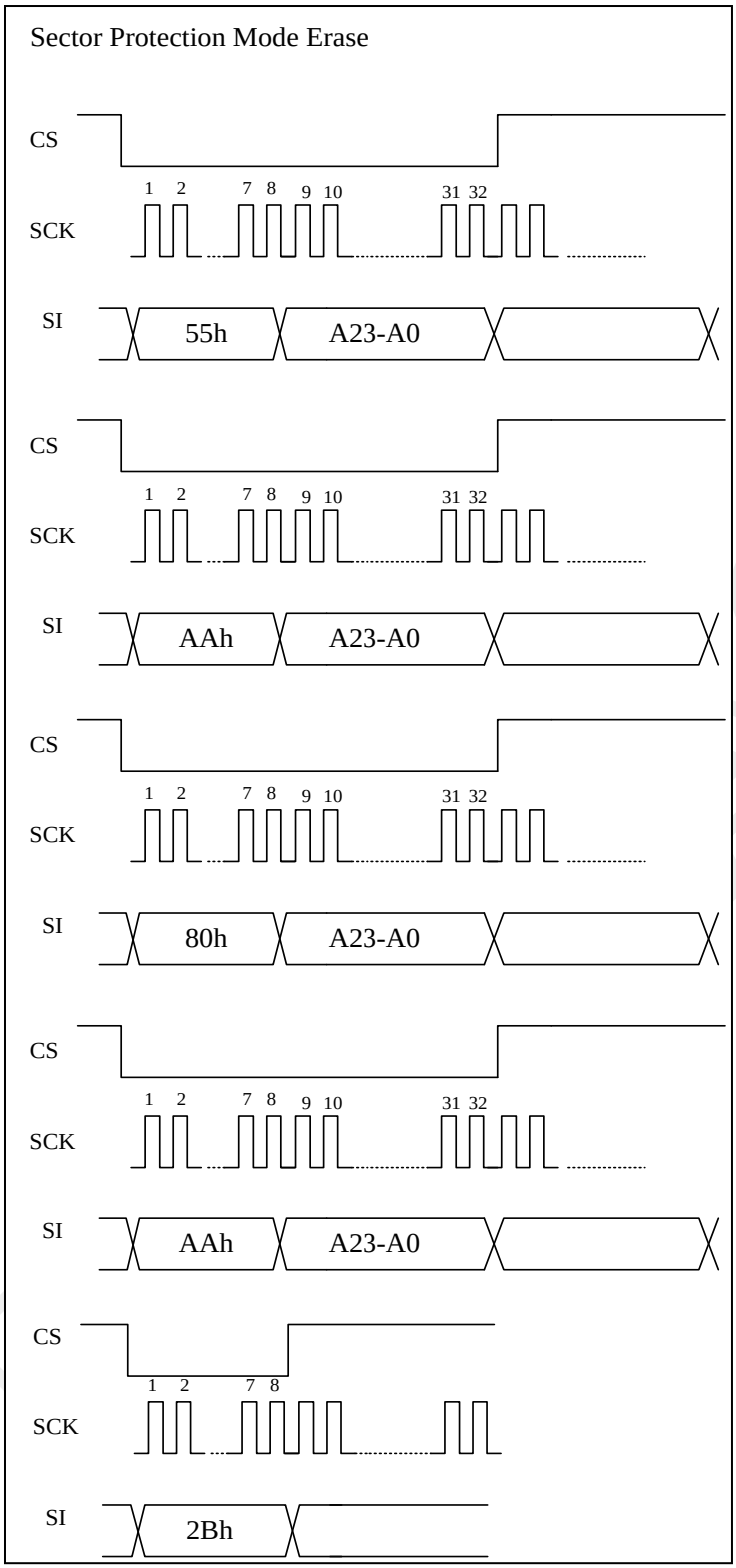



Fig b. Erase safe guard register

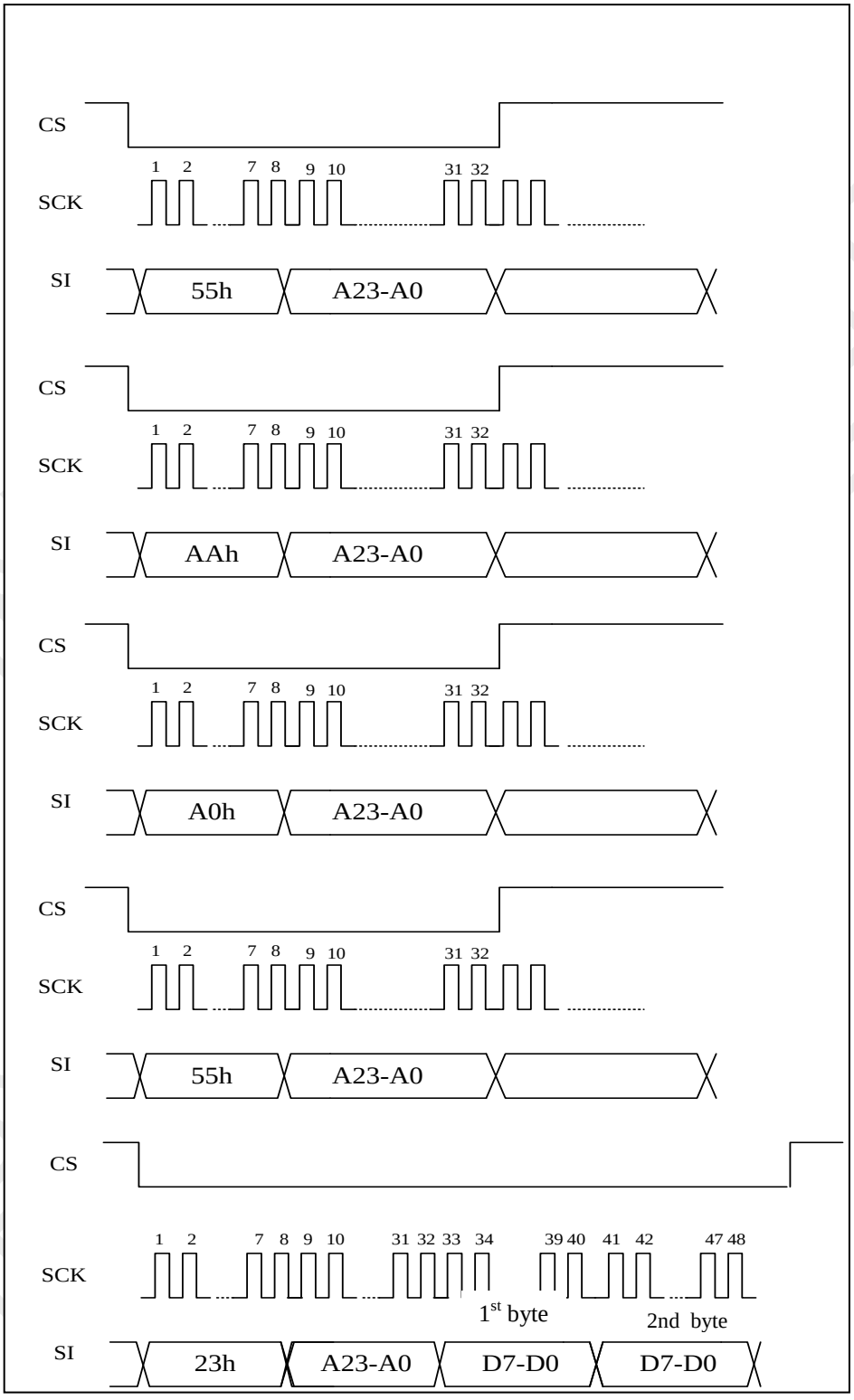



Fig c. program safe guard register

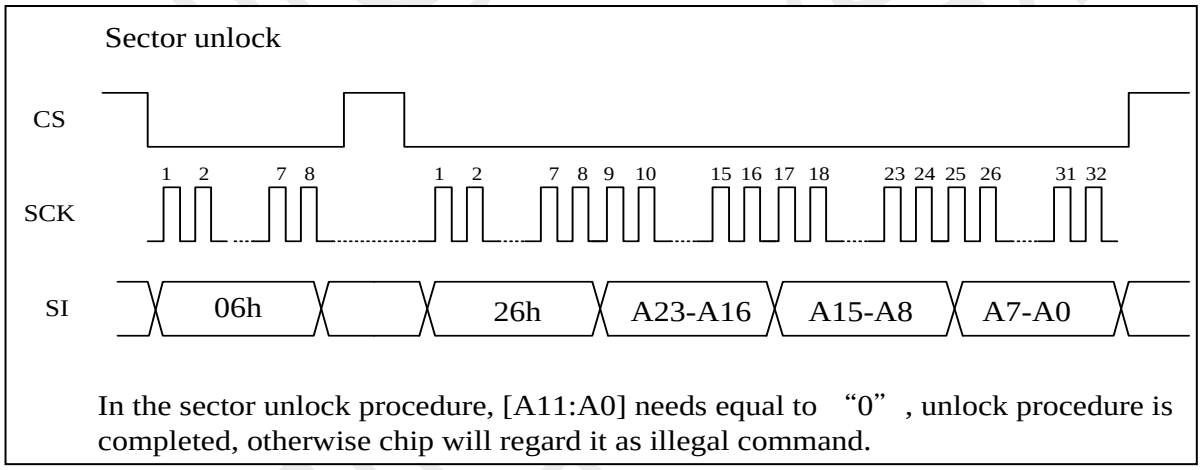
## Appendix2: Sector Unlock function

| Instruction Name | Hex Code | Operation | Command Cycle | Maximum Frequency |
|---|---|---|---|---|
| SECT_UNLOCK | 26h | Sector unlock | 4 Bytes | 104 MHz |
| SECT_LOCK | 24h | Sector lock | 1 Byte | 104 MHz |

**SEC_UNLOCK COMMAND OPERATION**

The Sector unlock command allows the user to select a specific sector to allow program and erase operations. This instruction is effective when the blocks are designated as write-protected through the BP0, BP1, BP2 and BP3 bits in the status register. Only one sector can be enabled at any time. To enable a different sector, a previously enabled sector must be disabled by executing a Sector Lock command. The instruction code is followed by a 24-bit address specifying the target sector, but A0 through A11 are not decoded. The remaining sectors within the same block remain in read-only mode.

**Figure d. Sector Unlock Sequence**

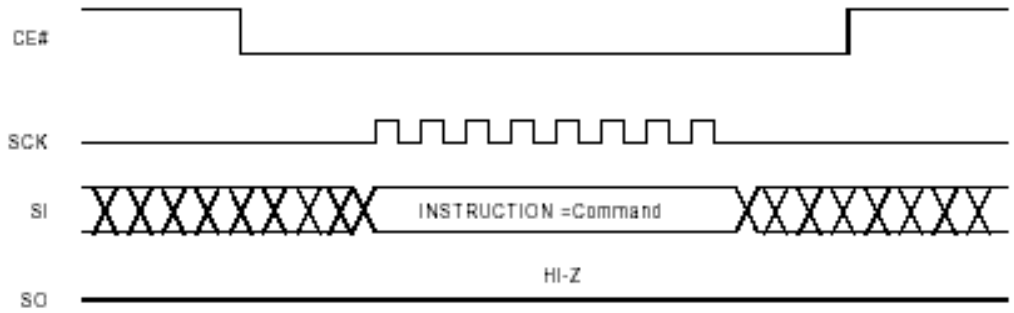



Note: 1.If the clock number will not match 8 clocks(command)+ 24 clocks (address), it will be ignored.
2.It must be executed write enable (06h) before sector unlock instructions.

### SECT_LOCK COMMAND OPERATION

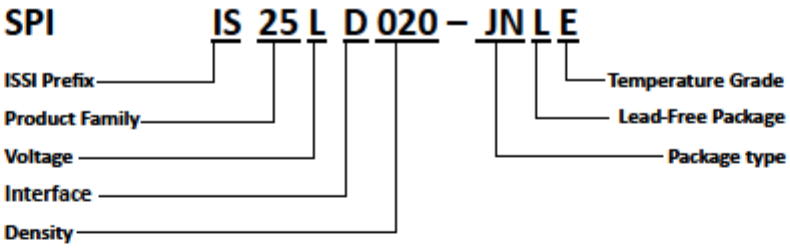
The Sector Lock command reverses the function of the Sector Unlock command. The instruction code does not require an address to be specified, as only one sector can be enabled at a time. The remaining sectors within the same block remain in read-only mode.

**Figure e. Sector Lock Sequence**




RELEASED

## ORDERING INFORMATION:




**Flash Product Family**
SPI = 25

**SPI Operating Voltage Range**
L = 2.30V-3.60V
W = 1.65V-1.95V
C = 2.70-3.60V

**SPI Interface**
D = x2 output
Q = x4 output

**Density**
025 = 256Kb
512 = 512Kb
010 = 1Mb
020 = 2Mb
040 = 4Mb
080 = 8Mb
016 = 16Mb
032 = 32Mb
064 = 64Mb
128 = 128Mb
256 = 256Mb

**Package Type**
JN = 8 pin SOIC 150mil
JB = 8 pin SOIC 208mil
JM = 16 pin SOIC 300mil
JD = 8 pin TSSOP
JE = 48 pin TSSOP
JT = 56 pin TSSOP
JK = 8 pin WSON (5x6 mm)
JL = 8 pin WSON (6x8mm)
JU = 8 pin USON (2x3mm)
JF = 8 pin VSOP 208mil
JV = 8 pin VVSOP 150mil
JG = 24 ball (BGA 6x8mm) (Call Factory)
JH = 48 ball (BGA 6x8mm) (Call Factory)
JI = 64 ball (BGA 11x13mm) (Call Factory)
JS = WLCSP (Call Factory)
JW = KGD (Call Factory)

**Lead-Free Package**
All flash products are lead free

**Temperature Grade**
E = Extended grade (-40C to +105C)
A1 = Automotive grade (-40C to +85C)
A2 = Automotive grade (-40C to +105C)
A3 = Automotive grade (-40C to +125C)

REL

| Density | Frequency (MHz) | Order Part Number | Package |
|---|---|---|---|
| 8M | 104 | IS25WQ080-JBLE | 8-pin SOIC 208mil |
| | | IS25WQ080-JNLE | 8-pin SOIC 150mil |
| | | IS25WQ080-JKLE | 8-pin WSON (5x6mm) |
| | | IS25WQ080-JVLE | 8-pin VVSOP 150mil |
| | | IS25WQ080-JBLA* | 8-pin SOIC 208mil (Call Factory) |
| | | IS25WQ080-JNLA* | 8-pin SOIC 150mil (Call Factory) |
| | | IS25WQ080-JKLA* | 8-pin WSON (5x6mm) (Call Factory) |
| | | IS25WQ080-JVLA* | 8-pin VVSOP 150mil (Call Factory) |

A* = A1,A2,A3 Automotive Temperature Ranges

NOT RELEASED