BST25VF040B Type NORFLASH

Product Brochure

Ver.1

Revision History

Version	date	Change Description	Change Person
1	2024-06-10	Initial release	Que Min

Table of contents

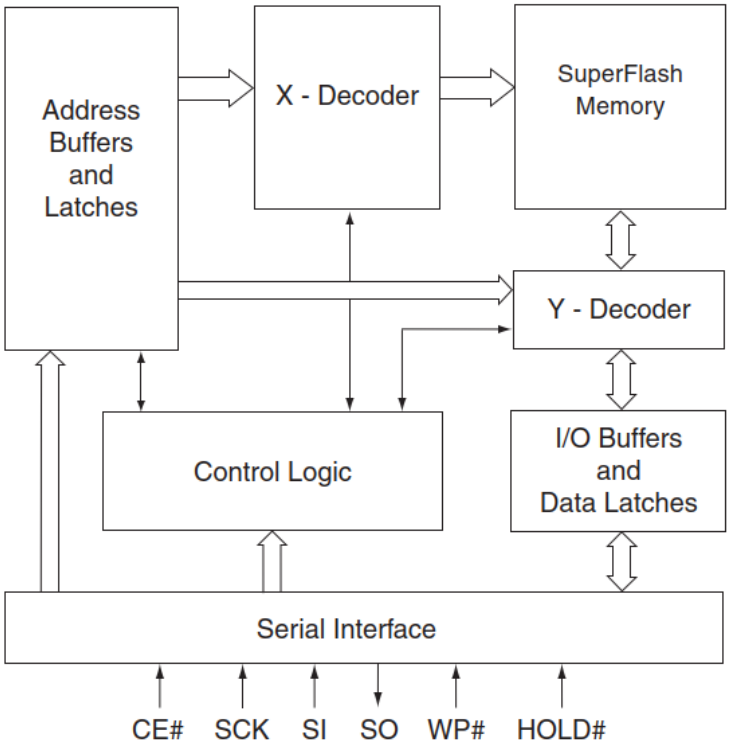
I. Product Features

- Single voltage read and write operation
    - 2.7~3.6V
- Serial interface architecture
    - SPI compatible: Mode 0 and Mode 3
- High clock frequency
    - Up to 50MHz
- High reliability
    - Durability: 100,000Erasing times (Typical value,25℃)
    - Data retention time:10 years (Typical value,25℃)
- Flexible erase function

  - Unified 4K byte sectors
  - Unified 32K byte overlay blocks
  - Unified 64K byte overlay blocks
- Automatic Address Increment (AAI) programming
  - Reduce total chip programming time compared to byte programming operations
- Write end detection
  - Poll the BUSY bit in the status register via software
  - In AAI mode, the busy status can be read through the SO pin
- HOLD pin (HOLD#)
  - Suspend serial operations to memory without deselecting the device
- Write Protect (WP#)
  - Enable/disable the lock function of the status register
- Software write protection
  - Write protection via block protection bit in status register
- Temperature range
- -55°C to +125°C
- Encapsulation
  - CSOP8

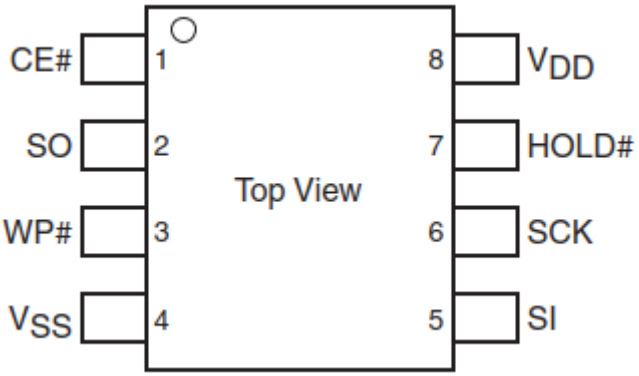
II. Product Description

This device is a 4Mbit, 3V single power supply (2.7~3.6V) Nor Flash memory. It uses a four-wire, SPI-compatible interface with a lower pin count, which occupies less board space and ultimately reduces the cost of the entire system. The SPI frequency is up to 50MHz and supports automatic address increment (AAI) programming, which can reduce the total programming time of the chip.



Picture 2-1. Functional Block Diagram

III. Pin Definition



Picture 3-1. Pin Arrangement

Table 3-1. Pin Description

conform to	Pin Name	Function

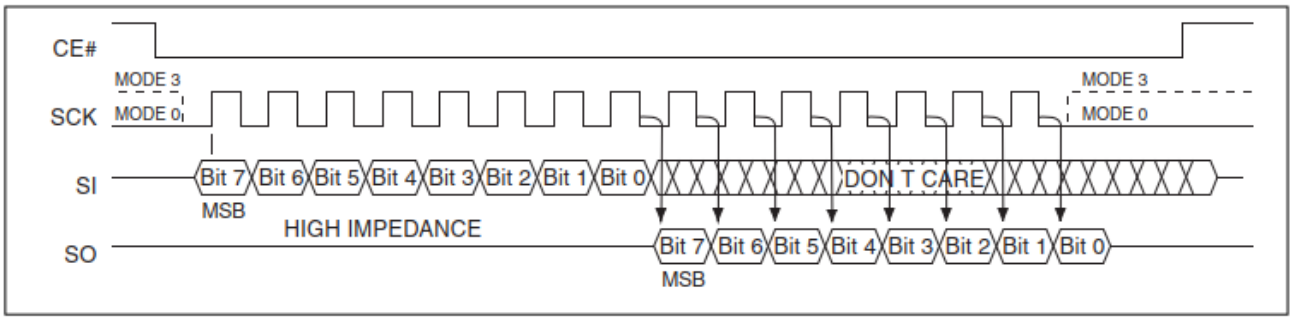
conform to	Pin Name	Function
SCK	Serial Clock	Provides timing control for the serial interface. Commands, addresses, or input data are latched on the rising edge of the clock input, and output data are shifted out on the falling edge of the clock input.
SI	Serial data input	Used to transfer commands, addresses or data serially into the device. Input data is latched on the rising edge of the serial clock.
SO	Serial data output	Used to serially output data out of the device. Data is shifted out on the falling edge of the serial clock. During AAI programming, when reconfigured as RY/BY# pin, outputs Flash busy status.
CE#	Chip Select	The device is enabled by a high-to-low transition of CE#. CE# must remain low for the duration of any command sequence.
WP#	Write protection	The Write Protect (WP#) pin is used to enable/disable the BPL bit in the status register.
HO;D#	Keep	Used to temporarily stop serial communication with the SPI Flash memory without resetting the device.
VDD	power supply	Provide power supply voltage: 2.7~3.6V
VSS	land	

IV. Device Operation

The device is accessed via a protocol compatible with the SPI (Serial Peripheral Interface) bus. The SPI bus consists of four control lines; Chip Select (CE#) is used to select the device, and data is accessed via Serial Data Input (SI), Serial Data Output (SO), and Serial Clock (SCK).

This device supports two modes of SPI bus operation: Mode 0 (0, 0) and Mode 3 (1, 1).Figure 4-1As shown in Figure 1, the difference between the two modes is the state of the SCK signal when the bus master is in standby mode and not transmitting data. For Mode 0, the SCK

signal is low, while for Mode 3, the SCK signal is high. For both modes, the serial data input (SI) is sampled on the rising edge of the SCK clock signal, while the serial data output (SO) is driven after the falling edge of the SCK clock signal.



Picture 4-1.SPI Protocol

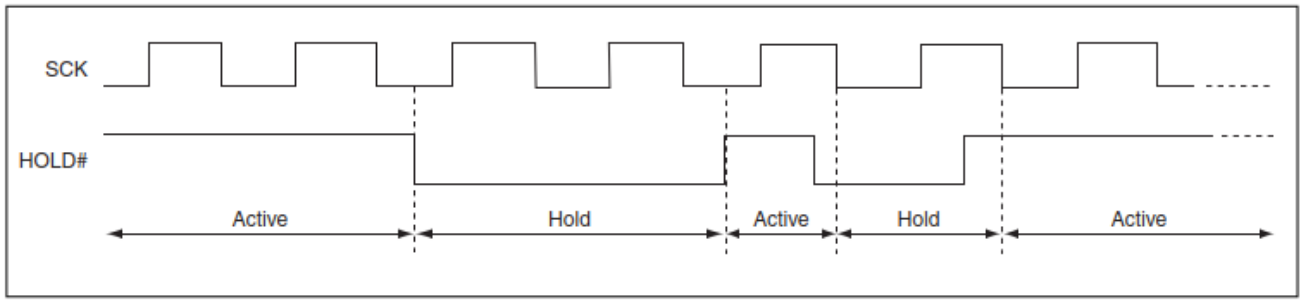
4.1. Hold Operation

The HOLD# pin is used to suspend ongoing serial operations with the SPI Flash memory without resetting the clock sequence. To activate the HOLD# mode, CE# must be in an active low state. The HOLD# mode begins when the low state of SCK coincides with the falling edge of the HOLD# signal. The HOLD# mode ends when the rising edge of the HOLD# signal coincides with the low state of SCK.

If the falling edge of the HOLD# signal does not coincide with the low state of SCK, the device enters hold mode the next time SCK goes low. Similarly, if the rising edge of the HOLD# signal does not coincide with the low state of SCK, the device exits hold mode the next time SCK goes low. For a waveform diagram of the HOLD condition, refer toFig.42.

Once the device enters hold mode, SO will be in high impedance state, while SI and SCK can be VIL (low level input voltage) or VIH (high level input voltage).

If CE# is driven high during the hold condition, the internal logic of the device is reset. The memory remains in the hold state as long as the HOLD# signal is low. To resume communication with the device, HOLD# must be driven high and CE# must be driven low.



Picture 4-1. Hold operation waveform

4.2. Write Protection

This device provides a software write protection feature. The Write Protect pin (WP#) is used to enable or disable the lock feature of the status register. The Block Protection bits (BP3, BP2, BP1, BP0, and BPL) in the Status Register provide write protection for the memory array and the Status Register. For a description of the Block Protection, refer toTable 4-3.

4.2.1. Write Protect Pin (WP#)

WriteThe Protect (WP#) pin is used to enable the lockout feature of the BPL bit (bit 7) in the Status Register. When WP# is pulled low, the behavior of the Write Status Register (WRSR) instruction depends on the value of the BPL bit (seeTable 4-1). When WP# is high, the lock function of the BPL bit is disabled.

Table 4-1. Conditions for executing the WRSR instruction

WP#	BPL	Execute the WRSR instruction
L	1	Not allowed
L	0	allow
H	X	allow

V. Status Register

The software status register provides information about whether the memory array is available for any read or write operation, whether the device is write enabled, and the write

protection status. During an internal erase or program operation, the status register can only be read to determine if the ongoing operation is complete.Table 4-2Describes the function of each bit in the software status register.

Table 4-1. Software Status Register

Bit	name	Function	Power-on default value	Read/Write
0	BUSY	Internal write operation in progress: 1 No internal write operation in progress: 0	0	Read-only (R)
1	WEL	Device storage write enable: 1 Device storage not write enabled: 0	0	Read-only (R)
2	BP0	Indicates the current block write protection level (seeTable 4-3)	1	Read/Write (R/W)
3	BP1	Indicates the current block write protection level (seeTable 4-3)	1	Read/Write (R/W)
4	BP2	Indicates the current block write protection level (seeTable 4-3)	1	Read/Write (R/W)
5	BP3	Indicates the current block write protection level (seeTable 4-3)	0	Read/Write (R/W)
6	AAI	Automatic address increment programming status: 1 = AAI programming mode 0 = Byte Programming Mode	0	Read-only (R)
7	BPL	1 = BP3, BP2, BP1, BP0 bits are read-only 0 = BP3, BP2, BP1, BP0 are readable and writable	0	Read/Write (R/W)

5.1. Busy state

The busy bit determines whether there is an internal erase or program operation in progress. A busy bit of "1" indicates that the device is busy with an operation. A "0" indicates that the device is ready for the next valid operation.

5.2. Write Enable Latch (WEL)

The Write Enable Latch bit indicates the status of the internal memory write enable latch. If

the Write Enable Latch bit is set to “1”, it indicates that the device is write enabled. If the bit is set to “0” (reset), it indicates that the device is not write enabled and will not accept any memory write (program/erase) commands. The Write Enable Latch bit is automatically reset under the following conditions:

- Power on
- Write Disable (WRDI) instruction completed
- Byte Program Instruction Complete
- Automatic Address Increment (AAI) programming is complete or reaches its highest unprotected memory address
- Sector erase command completed
- Block Erase Command Completed
- Chip erase command completed
- Write Status Register Instruction Completed

5.3. Automatic Address Increment (AAI)

The Auto Address Increment Programming Status bit indicates whether the device is currently in AAI Programming Mode or Byte Programming Mode. The power-on default is Byte Programming Mode.

5.4. Block protection (BP3, BP2, BP1, BP0)

The block protection (BP3, BP2, BP1, BP0) bits define the size of a memory area (e.g.Table 4-3), this area is software protected against any memory write (program or erase) operation. The Write Status Register (WRSR) instruction can be used to program the BP3, BP2, BP1, and BP0 bits as long as WP# is high or the Block Protection Lock (BPL) bit is 0. A chip erase operation can only be performed when all block protection bits are 0. After power-on, BP3, BP2, BP1, and BP0 are set to 1.

5.5. Block Protection Lock (BPL)

When the WP# pin is pulled low (VIL), the Block Protection Lock (BPL) bit is enabled. When BPL is set to 1, it prevents further changes to the BPL, BP3, BP2, BP1, and BP0 bits. When the WP# pin is pulled high (VIH), the BPL bit has no effect and its value is "Don't Care". After power-on, the BPL bit is reset to 0.

Table 4-2. Software status register block protection setting

Protection Level	Status Register Bits				Protected Address
	BP3	BP2	BP1	BP0	4Mbit
No protection	X	0	0	0	none
Upper 1/8	X	0	0	1	70000H~7FFFFH
Upper 1/4	X	0	1	0	60000H~7FFFFH
Upper 1/2	X	0	1	1	40000H~7FFFFH
All blocks	X	1	0	0	0000H~7FFFFH
All blocks	X	1	0	1	0000H~7FFFFH
All blocks	X	1	1	0	0000H~7FFFFH
All blocks	X	1	1	1	0000H~7FFFFH

Note:

- "X" in the table means it is not important, and the default value is "0";
- The power-on default value of BP2, BP1, and BP0 is "111" (all blocks are protected).

VI. Instruction

Instructions are used for read, write (erase and program), and configuration. The instruction bus cycle is 8 bits for command (opcode), data, and address. A Write Enable (WREN) instruction must be executed before any Byte Program, Auto Address Increment (AAI) Program, Sector Erase, Block Erase, Write Status Register, or Chip Erase instructions.
A complete instruction listing is inTable 4-4All instructions are synchronized with the transition of CE# from high to low. Input begins on the rising edge of SCK and is received

most significant bit first. CE# must be pulled low before inputting instructions and must be pulled high after the last bit of the instruction is shifted in (except for Read, Read ID, and Read Status Register instructions). Any low-to-high transition of CE# before the last bit of the instruction bus cycle is received will terminate the instruction in progress and return the device to Standby mode. Instruction commands (opcodes), addresses, and data are all input starting with the most significant bit (MSB).

Table 4-1. Operation Instructions

Instruction	Describe	Opcode Cycle 1	Address Cycle 2	Idle Cycle	Data Cycle
Read	Reading Storage Arrays	0000 0011b (03H)	3	0	1 to ∞
High-speed reading	Reading storage arrays at higher speeds	0000 1011b (0BH)	3	1	1 to ∞
4 KByte sector erase 3	Erase 4 KByte of memory array	0010 0000b (20H)	3	0	0
32 KByte Block Erase 4	Erase 32 KByte memory block	0101 0010b (52H)	3	0	0
64 KByte Block Erase 5	Erase 64 KByte memory block	1101 1000b (D8H)	3	0	0
Chip Erase	Erase the entire storage array	0110 0000b (60H) or 1100 0111b (C7H)	0	0	0
Byte Programming	Program a data byte	0000 0010b (02H)	3	0	1
AAI-Word Programming 6	Automatic address increment programming	1010 1101b (ADH)	3	0	2 to ∞
RDSR7	Read Status Register	0000 0101b (05H)	0	0	1 to ∞
EWSR	Enable write status register	0101b 0000b (50H)	0	0	0
WRSR	Write Status Register	0000 0001b (01H)	0	0	1

Instruction	Describe	Opcode Cycle 1	Address Cycle 2	Idle Cycle	Data Cycle
WREN	Write Enable	0000 0110b (06H)	0	0	0
WRDI	Write Disable	0000 0100b (04H)	0	0	0
RDID8	Read ID	1001 0000b (90H) or 1010 1011b (ABH)	3	0	1 to ∞
RDID8	Read ID	1001 0000b (90H) or 1010 1011b (ABH)	3	0	1 to ∞
JEDEC-ID	JEDEC ID Reading	1001 1111b (9FH)	0	0	3 to ∞
EBSY	During AAI programming, SO outputs RY/BY# status	0111 0000b (70H)	0	0	0
DBSY	Disable SO as RY/BY# state during AAI programming	1000 0000b (80H)	0	0	0

Note:

- One bus cycle is eight clock cycles.
- Address bits above the most significant bit can be VIL or VIH.
- 4 KByte sector erase address: Use AMS-A12, the rest of the addresses are irrelevant but must be set to VIL or VIH.
- 32 KByte block erase address: Use AMS-A15, the rest of the addresses are irrelevant but must be set to VIL or VIH.
- 64 KByte block erase address: Use AMS-A16, the rest of the addresses are irrelevant but must be set to VIL or VIH.
- To continue programming to the next sequential address location, enter the 8-bit command ADH followed by the 2 bytes of data to be programmed. Data Byte 0 will

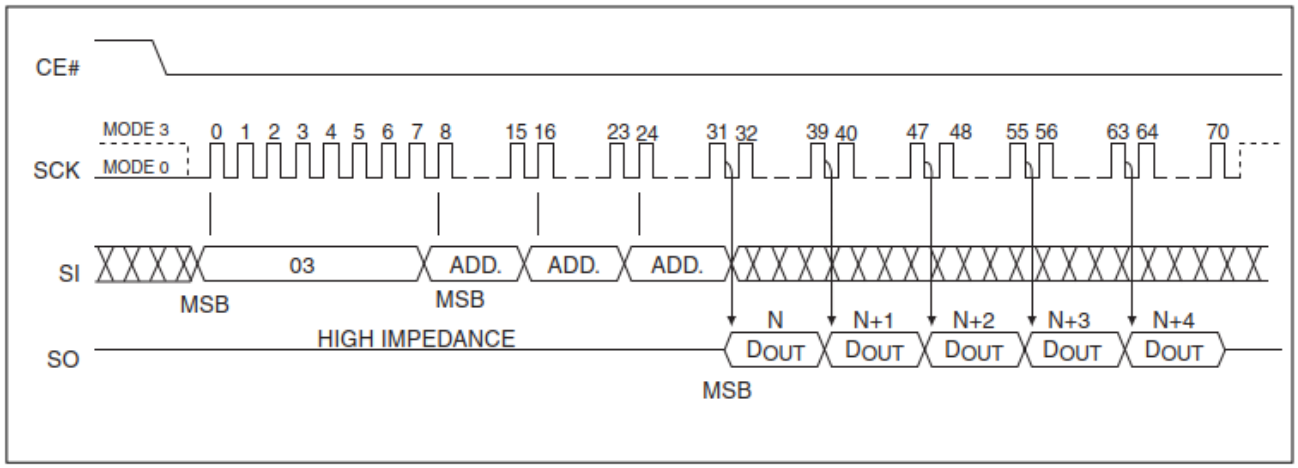
be programmed to the initial address [A23-A1] with A0=0 and Data Byte 1 will be programmed to the initial address [A23-A1] with A0=1.

- The read status register operation will continue with the continuous clock cycle until CE# is pulled high.
- The Manufacturer ID is read when A0 = 0 and the Device ID is read when A0 = 1. All other address bits are 00H. The output stream of the Manufacturer ID and Device ID will continue until a low-to-high transition occurs on CE#.

6.1. Read (25/33 MHz)

The read instruction 03H supports read speeds up to 25MHz. The device starts outputting data from the specified address location. The data output stream continues through all addresses until it is terminated by a low-to-high transition on CE#. The internal address pointer automatically increments until the highest memory address is reached. Once the highest memory address is reached, the address pointer automatically increments back (wraps around) to the beginning of the address space. Once the data from address location 1FFFFFH is read, the next output will start from address location 0000000H.

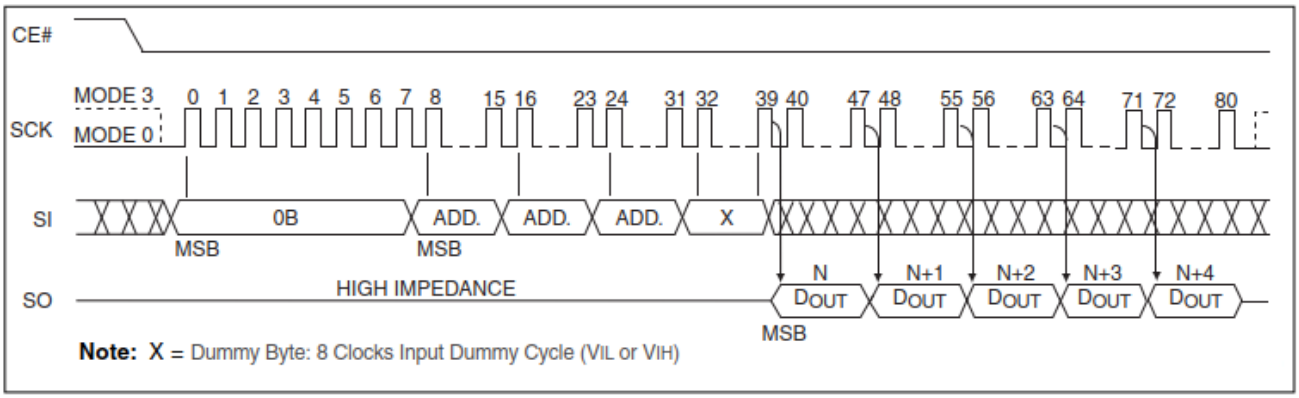
The read instruction starts by executing an 8-bit command 03H, followed by the address bits [A23-A0]. During the read cycle, CE# must remain active low.



Picture 4-1. Read sequence

6.1.1. High-speed read (50/80 MHz)

The high-speed read instruction supports read speeds up to 50 MHz by executing an 8-bit command starting with 0BH, followed by the address bits [A23-A0] and a dummy byte. During the high-speed read cycle, CE# must remain active low.



Picture 4-2. High-speed read sequence

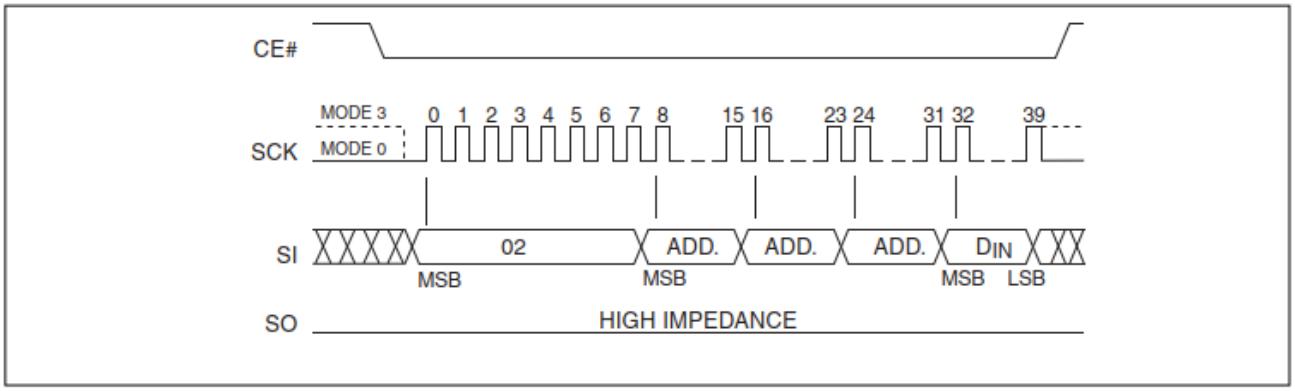
After the dummy cycle, the high-speed read instruction starts outputting data from the specified address location. The data output stream continues through all addresses until it is terminated by a low-to-high transition on CE#. The internal address pointer automatically increments until the highest memory address is reached. Once the highest memory address is reached, the address pointer automatically increments back (wraps around) to the beginning of the address space. Once the data from address location 7FFFFH is read, the next output will start from address location 000000H.

6.2. Byte-Program

The Byte Program instruction programs the bits in the selected byte to the desired data. The selected byte must be in the erased state (FFH) when initiating the programming operation. Applying a Byte Program instruction to a protected memory area will be ignored.

Prior to any write operation, a Write Enable (WREN) instruction must be executed. CE# must remain active low throughout the Byte Program instruction. The Byte Program instruction begins by executing an 8-bit command 02H, followed by the address bits [A23-A0]. Following the address, data is entered in order from the most significant bit (bit 7) to the least significant

bit (bit 0). CE# must be driven high before executing the instruction. The user can complete the internally self-timed Byte Program operation by polling the Busy bit in the Software Status Register or waiting for TBP.



Picture 4-3. Byte Programming Sequence

6.3. Automatic Address Increment (AAI) Word Programming

The AAI programming instruction allows multiple bytes of data to be programmed without resending the next sequential address. This feature reduces the total programming time when multiple bytes or the entire memory array need to be programmed. AAI word programming instructions that point to protected memory areas will be ignored. The selected address range must be in the erased state (FFH) when the AAI word programming operation is initiated. In the AAI word programming sequence, only the following instructions are valid: the instructions for software write end detection are AAI word (ADH), WRDI (04H), and RDSR (05H); the instructions for hardware write end detection are AAI word (ADH) and WRDI (04H). There are three ways to determine the completion of each AAI word programming cycle: hardware detection through serial output, software detection by polling the BUSY bit in the software status register, or waiting for TBP. Please refer to the "Write End Detection" section for details.

Prior to any write operation, a Write Enable (WREN) instruction must be executed. The AAI Word Program instruction is initiated by executing an 8-bit command, ADH, followed by the

address bits [A23-A0]. Following the address, two data bytes are entered sequentially, each from the most significant bit (Bit 7) to the least significant bit (Bit 0). The first data byte (D0) is programmed to the initial address [A23-A1] with A0 = 0, and the second data byte (D1) is programmed to the initial address [A23-A1] with A0 = 1. CE# must be driven high before executing the AAI Word Program instruction. Check for BUSY status before entering the next valid instruction. Once the device indicates that it is no longer busy, the data for the next two consecutive addresses can be programmed, followed by the next two, and so on.

When programming the last expected byte or the highest unprotected memory address, use the hardware or software (RDSR instruction) method to check the busy status to confirm that programming is complete. After programming is complete, terminate AAI using the appropriate method. If the device is in software write end detection mode, execute the Write Disable (WRDI) instruction, which is 04H. If the device is in AAI hardware write end detection mode, execute the Write Disable (WRDI) instruction 04H first, and then execute the 8-bit DBSY command 80H. Once the highest unprotected memory address is reached, there is no wraparound mode during AAI programming. Refer to the AAI word programming sequence.Figure 4-8andFigure 4-9.

6.4. Write end detection

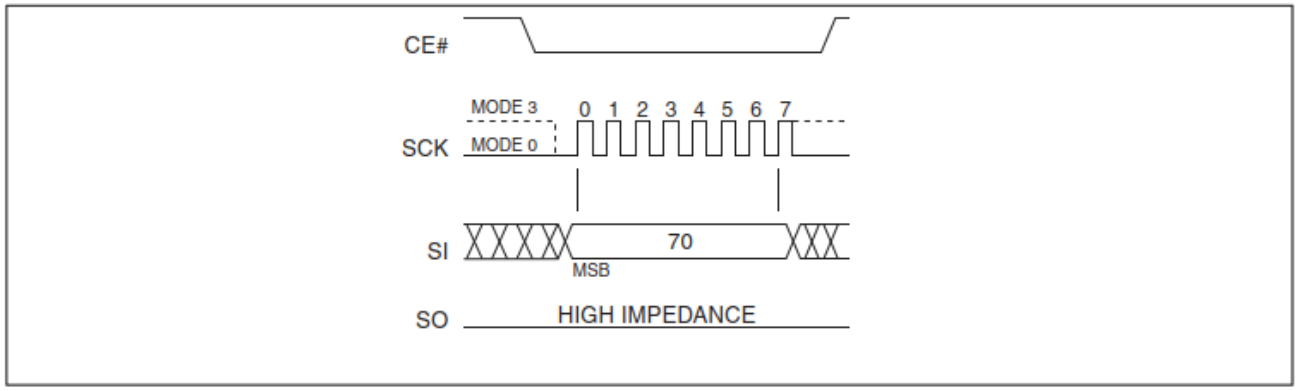
During the AAI word programming process, there are three methods that can be used to determine whether the programming cycle is complete: hardware detection by reading the serial output, software detection by polling the BUSY bit in the software status register, or waiting for TBP. The hardware write end detection method is described in the following section.

6.5. Hardware write end detection

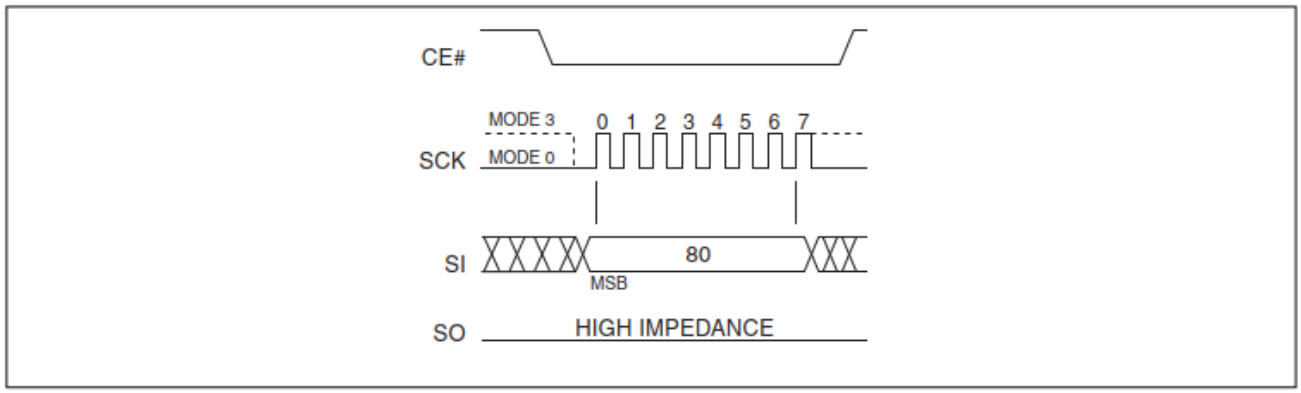
The hardware end-of-write detection method eliminates the overhead of polling the BUSY bit in the software status register during AAI word programming operations. The 8-bit command 70H configures the serial output (SO) pin to indicate the Flash busy status during AAI word

programming (seeFigure 4-6). Before executing the AAI word programming instruction, the 8-bit command 70H must be executed. Asserting CE# immediately displays the internal Flash status on the SO pin once the internal programming operation begins. A '0' indicates that the device is busy and a '1' indicates that the device is ready to receive the next instruction. De-asserting CE# returns the SO pin to tri-state. In AAI and Hardware Write End Detection mode, the only valid instructions are AAI Word (ADH) and WRDI (04H).

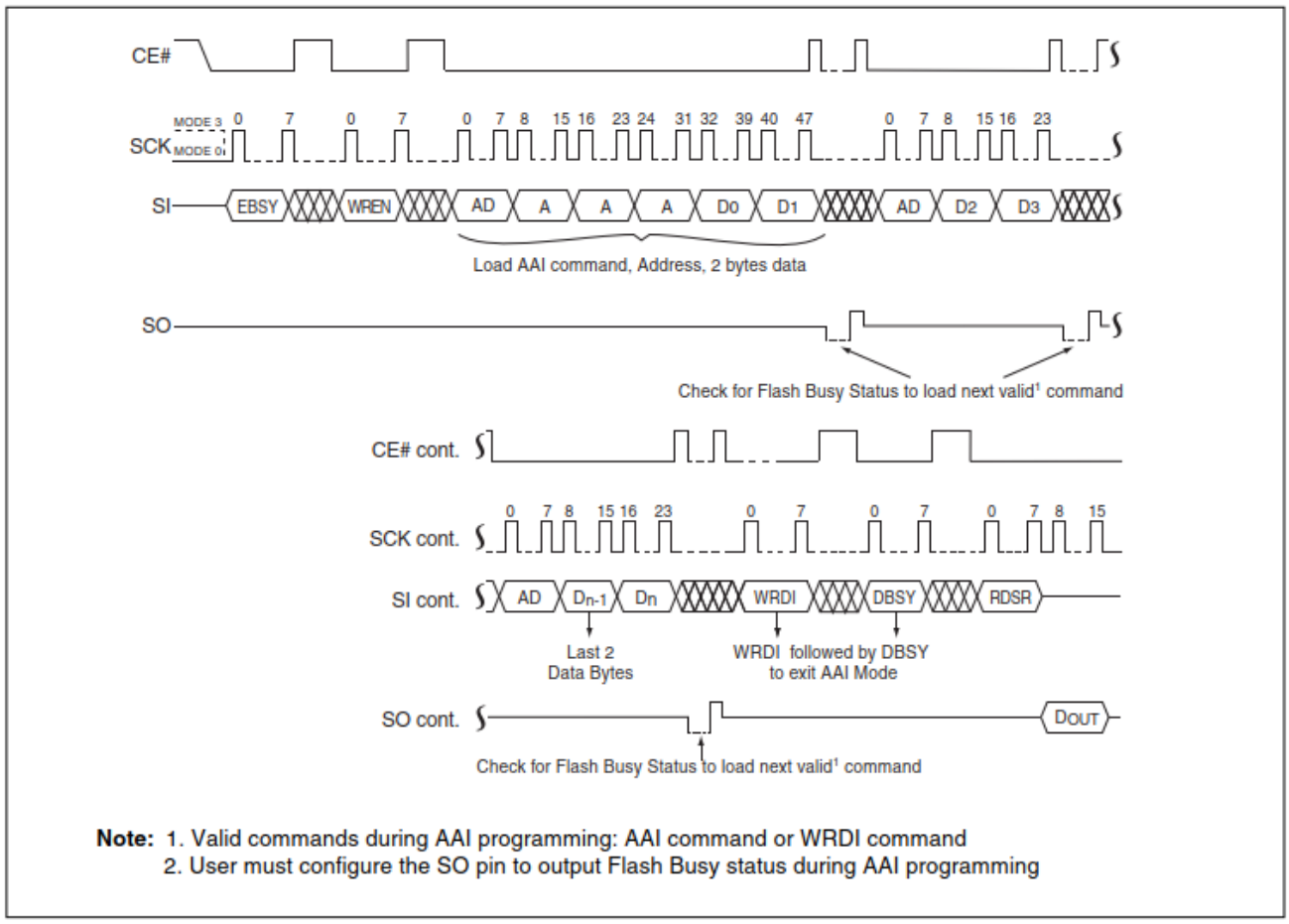
To exit the AAI hardware write end detection, first execute the WRDI instruction 04H to reset the write enable latch bit (WEL=0) and the AAI bit. Then execute the 8-bit DBSY command 80H to disable the RY/BY# state during the AAI command Figure 4-7 and Figure 4-8.



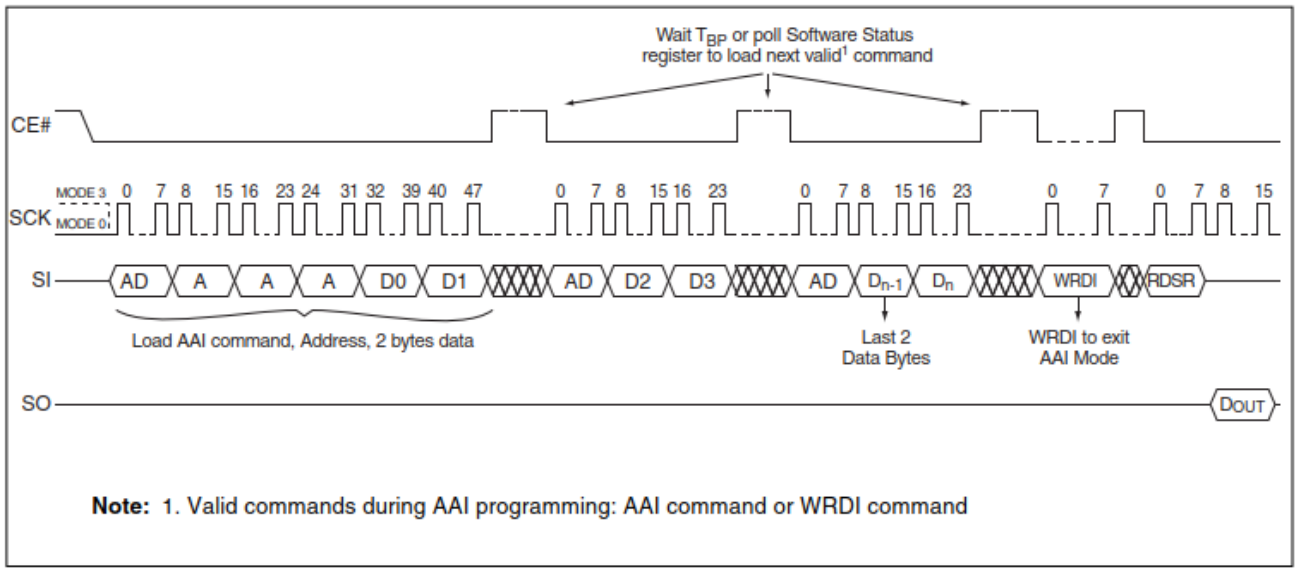
Picture 4-4. Enable SO as hardware RY/BY during AAI programming



Picture 4-5. Disable SO as hardware RY/BY during AAI programming#



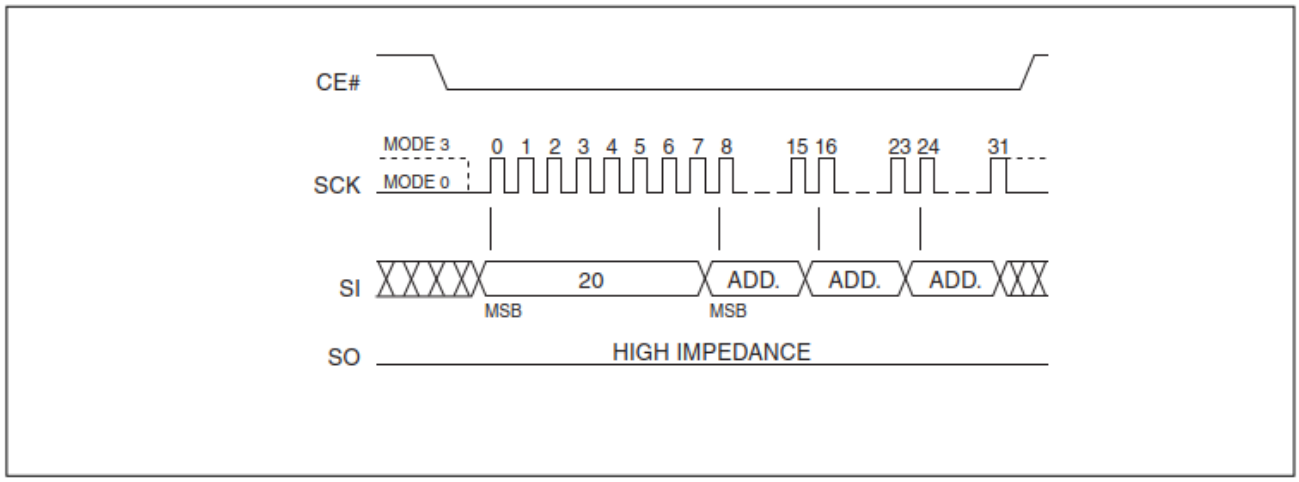
Picture 4-6. During AAI programming sequence (hardware end-of-write detection)



Picture 4-7. During the AAI programming sequence (software write end detection)

6.6 4-KByte sector erase

The Sector Erase instruction clears all bits in the selected 4-KByte sector to FFH. Sector Erase instructions applied to protected memory areas will be ignored. A Write Enable (WREN) instruction must be executed prior to any write operation. CE# must remain active low during any instruction sequence. The Sector Erase instruction begins by executing an 8-bit command 20H, followed by the address bits [A23-A0]. The address bits [AMS-A12] (AMS = Most Significant Address) are used to determine the sector address (SAX), and the remaining address bits can be VIL or VIH. CE# must be driven high before executing the instruction. The user can complete the internal self-timed sector erase cycle by polling the busy bit in the software status register or waiting for TSE.



Picture 4-8. Sector Erase Sequence

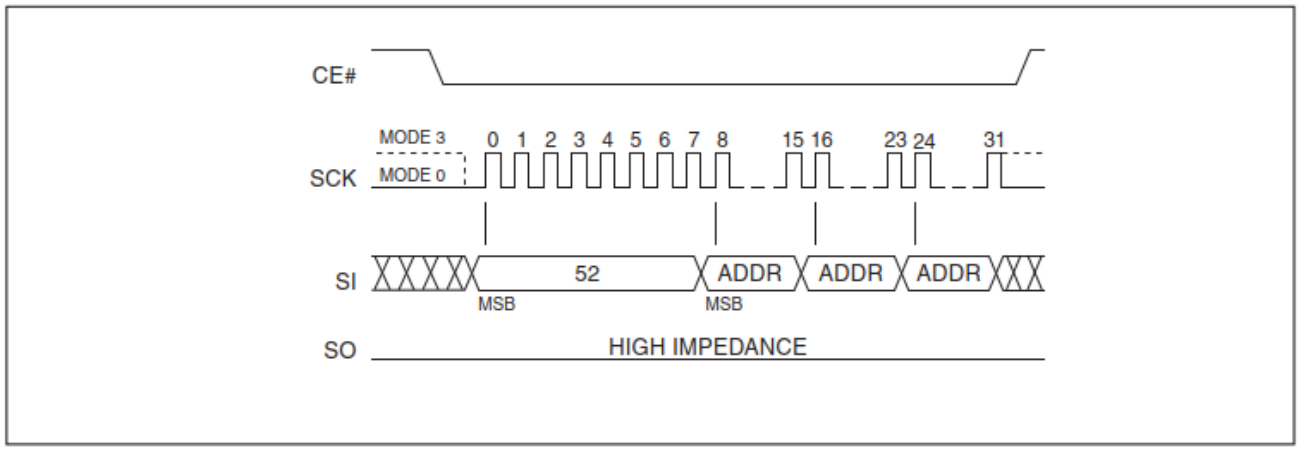
6.7. 32-KByte and 64-KByte Block Erase

The 32-KByte Block Erase instruction clears all bits within the selected 32-KByte block to FFH. Block Erase instructions applied to protected memory areas will be ignored. The 64-KByte Block Erase instruction clears all bits within the selected 64-KByte block to FFH. Likewise, Block Erase instructions applied to protected memory areas will be ignored. Prior to any write operation, a Write Enable (WREN) instruction must be executed. CE# must remain active low during any command sequence.

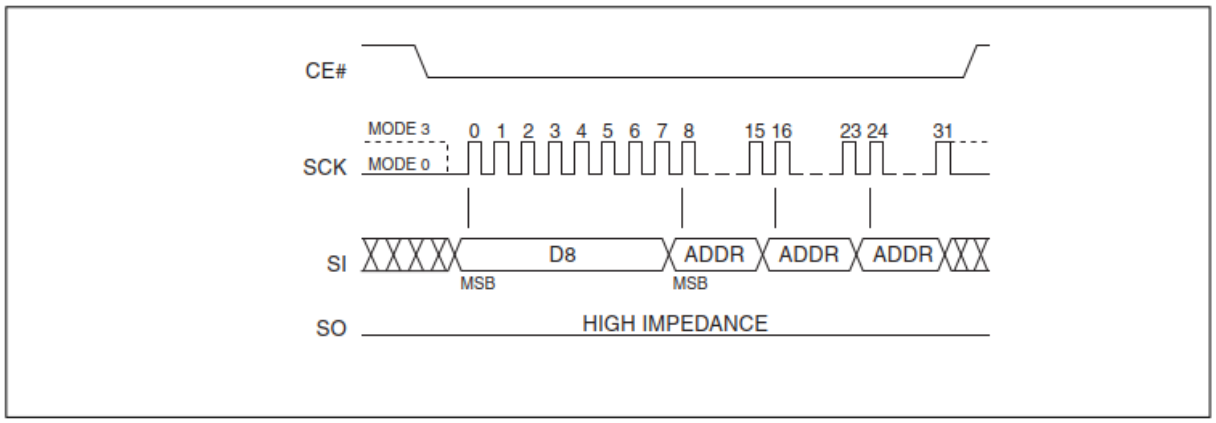
The 32-KByte Block Erase instruction starts by executing an 8-bit command 52H, followed by the address bits [A23-A0]. The address bits [AMS-A15] (AMS = Most Significant Address) are used to determine the block address (BAX), and the remaining address bits can be VIL or VIH. CE# must be driven high before executing the instruction.

The 64-KByte block erase instruction starts by executing an 8-bit command D8H, followed by the address bits [A23-A0]. The address bits [AMS-A15] are used to determine the block address (BAX), and the remaining address bits can be VIL or VIH. CE# must also be driven high before executing the instruction.

The user can complete the internal self-timed 32-KByte Block Erase or 64-KByte Block Erase cycle by polling the Busy bit in the software status register or waiting for TBE.



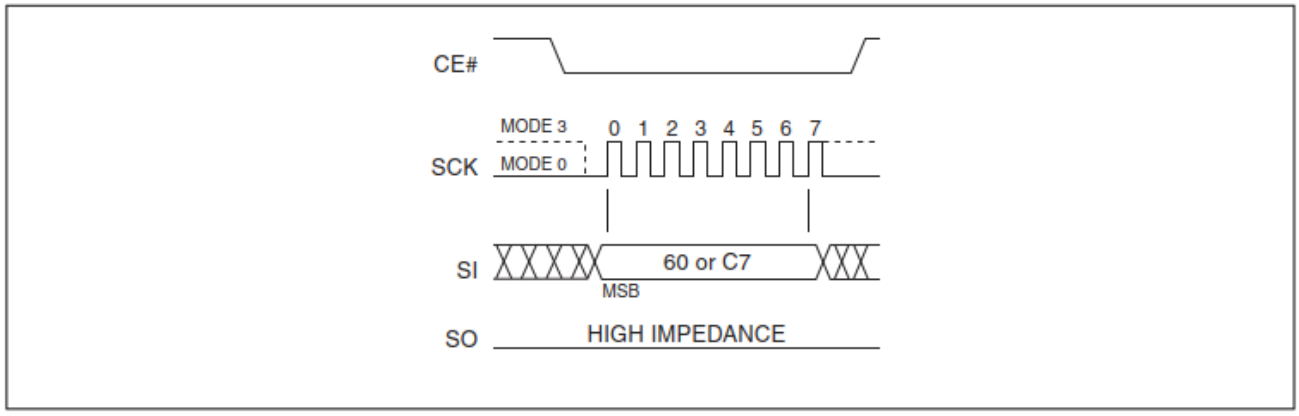
Picture 4-9. 32-KByte Block Erase Sequence



Picture 4-10. 64-KByte Block Erase Sequence

6.8. Chip Erase

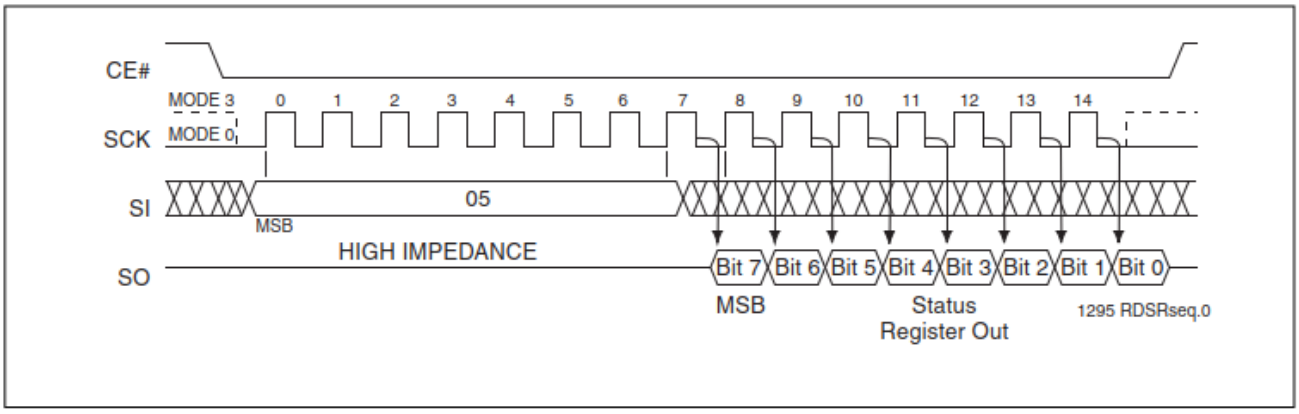
The Chip Erase instruction clears all bits in the device to FFH. If any memory area is protected, the Chip Erase instruction will be ignored. A Write Enable (WREN) instruction must be executed prior to any write operation. CE# must remain active low during the Chip Erase instruction sequence. The Chip Erase instruction is initiated by executing an 8-bit command 60H or C7H. CE# must be driven high before executing the instruction. The user can complete the internal self-timed Chip Erase cycle by polling the Busy bit in the Software Status Register or waiting for TCE.



Picture 4-11. Read Status Register Sequence

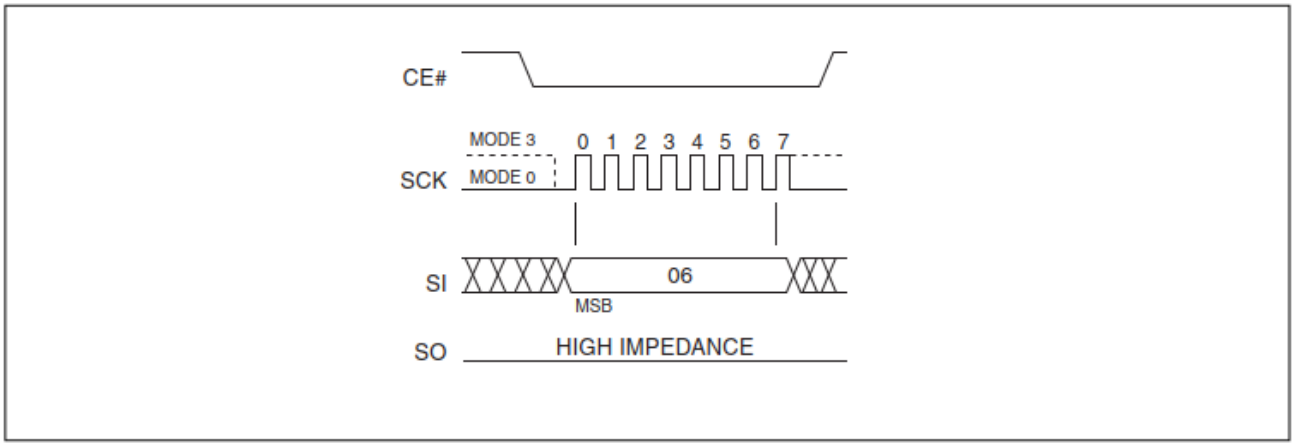
6.9. Read Status Register (RDSR)

The Read Status Register (RDSR) instruction allows the contents of the Status Register to be read. The Status Register can be read at any time, even during a Write (Program/Erase) operation. When a Write operation is in progress, the Busy bit can be checked before any new commands are sent to ensure that the new commands are properly received by the device. CE# must be pulled low before entering the RDSR instruction and remain low until the status data is read. The Read Status Register operation continues with each clock cycle until it is terminated by a low-to-high transition of CE#.



6.10. Write Enable (WREN)

The Write Enable (WREN) instruction sets the Write Enable Latch bit in the Status Register to 1 to allow write operations to occur. The WREN instruction must be executed prior to any write (program/erase) operation. The WREN instruction can also be used to allow the Write Status Register (WRSR) instruction to be executed; however, the Write Enable Latch bit in the Status Register will be cleared on the rising edge of CE# on the WRSR instruction. CE# must be driven high prior to executing the WREN instruction.

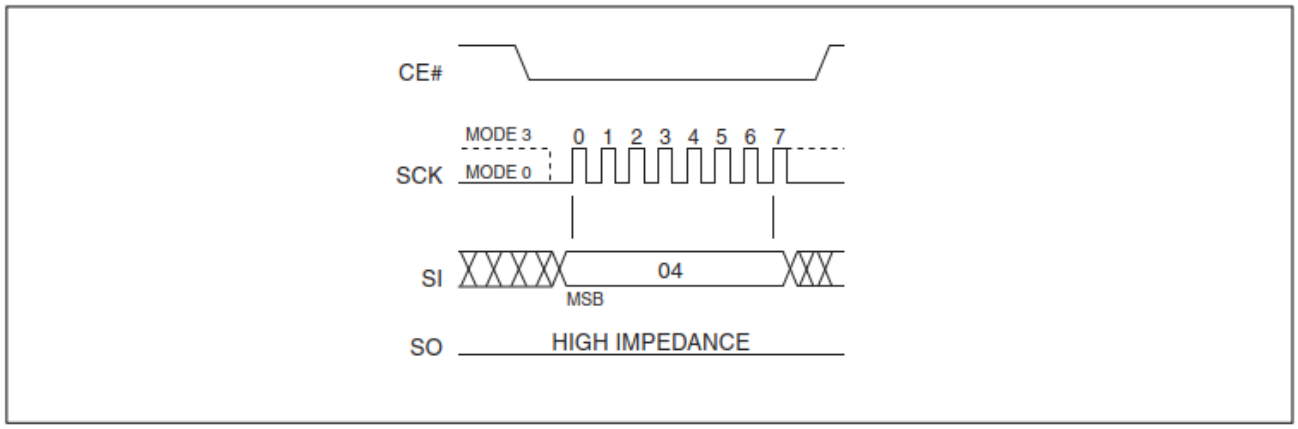


Picture 4-12. Write Enable (WREN) Sequence

6.11. Write Disable (WRDI)

The Write Disable (WRDI) instruction resets the Write Enable Latch bit and the AAI bit to 0, thereby preventing any new write operations from occurring. The WRDI instruction does not terminate any programming operation in progress. After executing the WRDI instruction, any

programming operation in progress may still continue until the TBP (Programming Time Interval) expires. The Chip Select (CE#) signal must be driven high before executing the WRDI instruction.



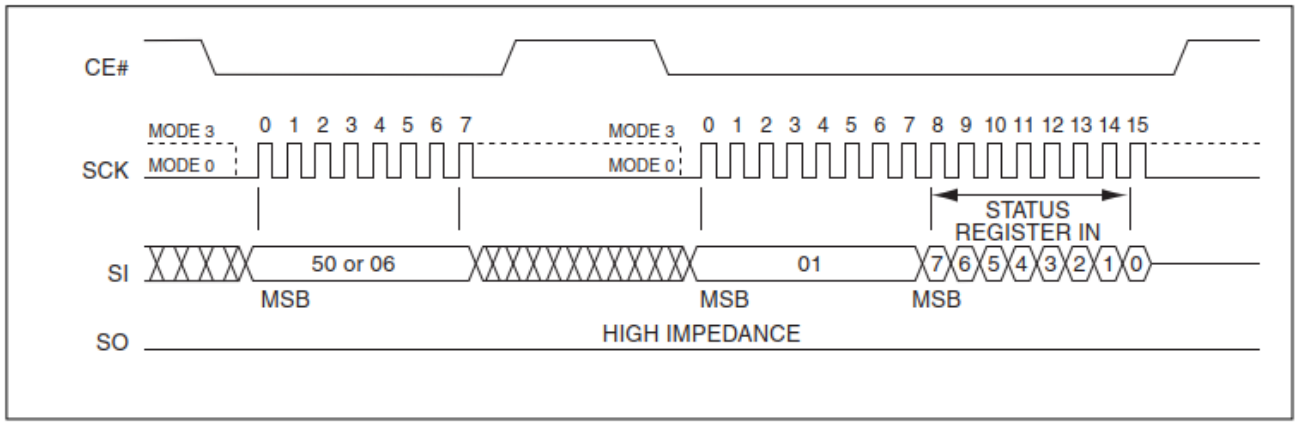
Picture 4-13. Write Disable (WRDI) Sequence

6.12. Enable Status Register Write (EWSR)

The Enable Status Register Write (EWSR) instruction prepares the Write Status Register (WRSR) instruction and opens the Status Register for modification. The Write Status Register instruction must be executed immediately after the EWSR instruction is executed. This two-step instruction sequence of the EWSR instruction followed by the WRSR instruction is similar to the SDP (Software Data Protection) command structure, which prevents accidental changes to the Status Register value. CE# must be pulled low before entering the EWSR instruction and must be pulled high before executing the EWSR instruction.

6.13. Write Status Register (WRSR)

The Write Status Register instruction writes new values to the BP3, BP2, BP1, BP0, and BPL bits of the Status Register. CE# must be pulled low before entering the command sequence for the WRSR instruction, and must be pulled high before executing the WRSR instruction. For EWSR or WREN and WRSR instruction sequences, see Figure 4-17.

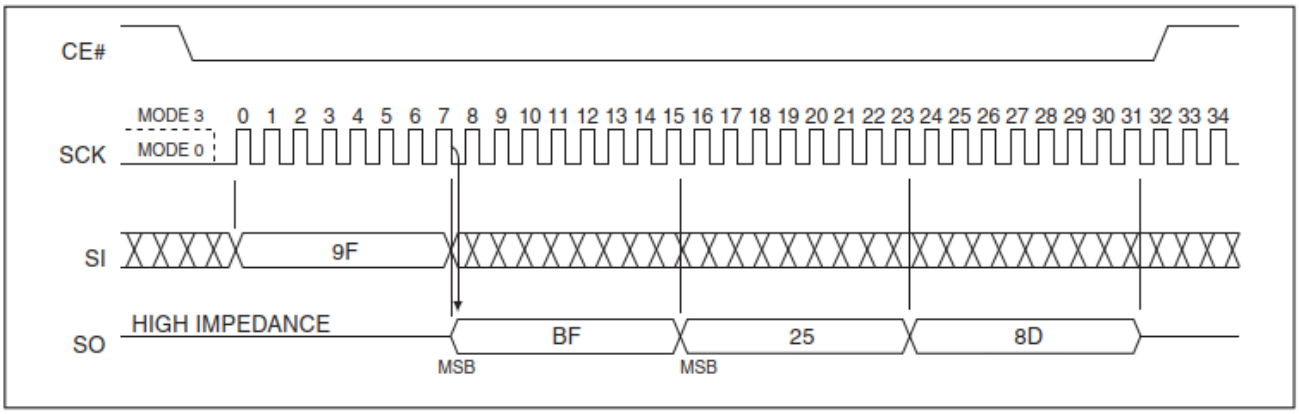


Picture 4-14. EWSR or WREN and WRSR sequences

Executing a Write Status Register instruction while WP# is low and the BPL bit is set to "1" will be ignored. When WP# is low, the BPL bit can only be set from "0" to "1" to lock the status register, but it cannot be reset from "1" to "0". When WP# is high, the lock function of the BPL bit is disabled and the BPL, BP0, BP1, and BP2 bits in the status register can be changed. All bits in the status register can be changed by a WRSR instruction as long as the BPL bit is set to 0 or the WP# pin is driven high (VIH) before the low-to-high transition of the CE# pin at the end of the WRSR instruction. In this case, a single WRSR instruction can both set the BPL bit to "1" to lock the status register and change the BP0, BP1, and BP2 bits simultaneously. For an overview of WP# and BPL functionality, see Table 4-1.

6.14. JEDEC Read ID

The JEDEC Read ID command is used to identify the device and identify the manufacturer. The device information can be read by executing the 8-bit command 9FH. After executing the JEDEC Read ID command, the device outputs the 8-bit manufacturer ID BFH. Subsequently, a 16-bit device ID is output bit by bit through the SO (serial output) pin. The first byte BFH identifies the manufacturer. The second byte 25H identifies the memory type as SPI serial flash. The third byte 8DH specifically identifies the device as BST25VF040. At any time during the data output, the JEDEC Read ID command is terminated by a low-to-high jump of the CE# pin.



Picture 4-18. JEDEC Read ID Sequence

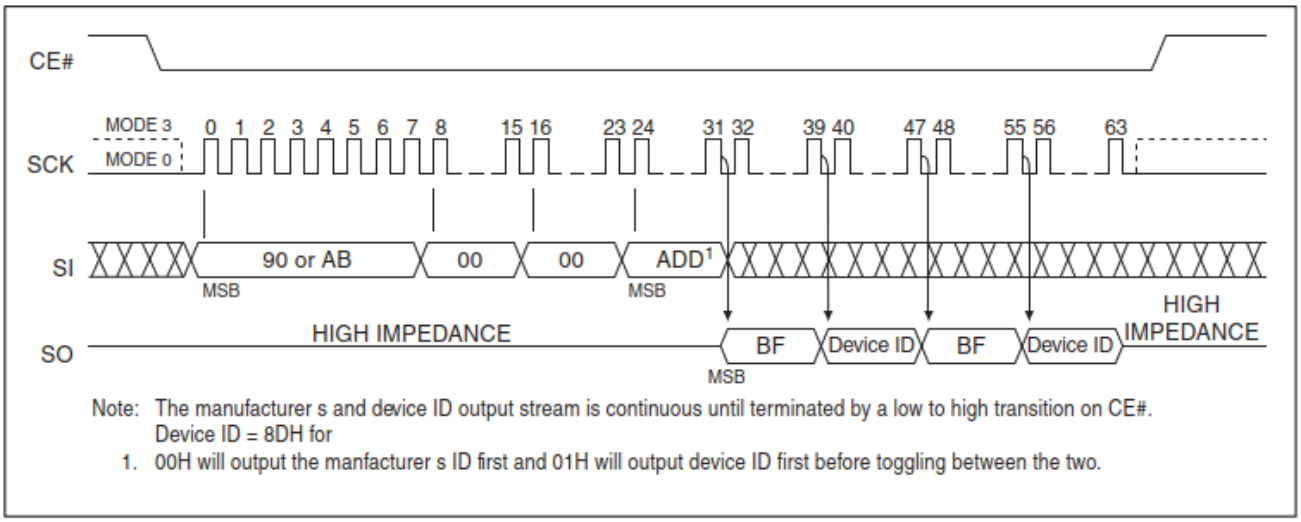
Table 4-2. JEDEC read ID data

Manufacturer ID	Device ID	
	type	capacity
Byte 1	Byte 2	Byte 3
BFH	25H	8DH

6.15. Read ID (RDID)

The Read ID command (RDID) is used to identify the device as well as the manufacturer. It should be used as the default device identification command when using a multi-version SPI serial flash device in the design. The device information can be read by executing an 8-bit command 90H or ABH followed by the address bits [A23-A0]. After executing the Read ID command, the Manufacturer ID is located at address 00000H and the Device ID is located at address 00001H. Once the device enters the Read ID mode, the output data of the Manufacturer ID and Device ID will toggle between addresses 00000H and 00001H until the process is terminated by a low-to-high transition of the CE# pin.

For device identification data, please refer toTable 4-5andTable 4-6.



Picture 4-19. Read ID sequence

Table 4-3. JEDEC read ID data

	Address	Data
Manufacturer ID	0000H	BFH
Device ID	0001H	8DH

VII. Electrical specifications

7.1. Absolute Maximum Ratings

Absolute Maximum Stress Ratings (Application conditions exceeding those listed under the “Absolute Maximum Stress Ratings” list may cause permanent damage to the device. This is a description of the stress level only and does not imply that functional operation of the device at these or other conditions beyond those defined in the operational sections of this data sheet is guaranteed. Exposure to absolute maximum stress rating conditions may affect device reliability.)

Operating temperature . . . . . -55℃~ +125℃

Storage temperature . . . . . -65℃~+150℃

DC voltage on any pin with respect to ground . . . . . VDD + 0.5V

Surface mount reflow temperature . . . . . 250℃, lasts 10 seconds

7.2. DC parameters

Table 5-1. DC parameters

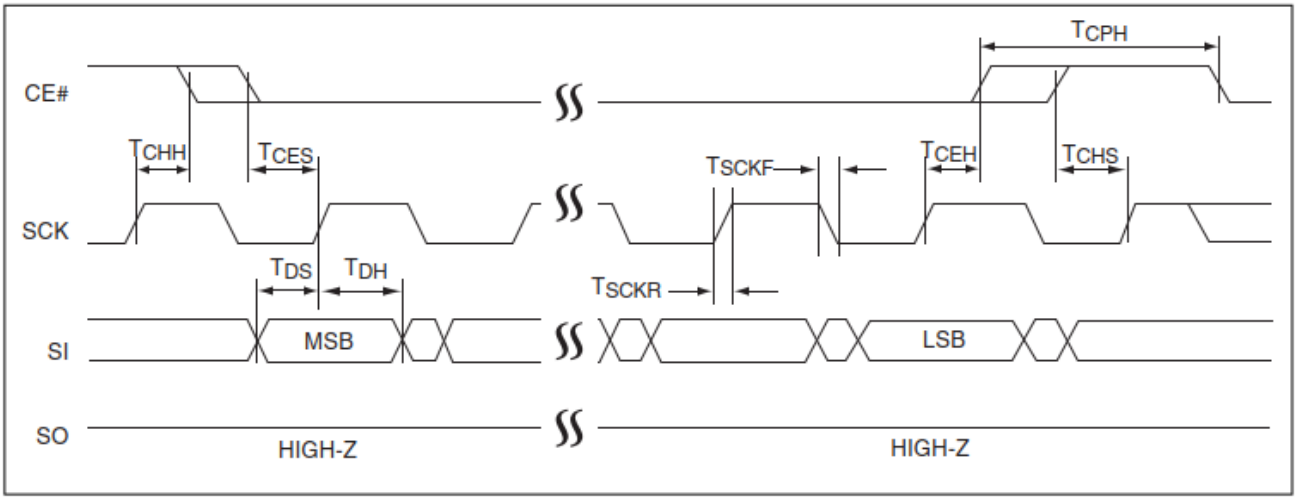
Symbol	Parameter	Condition	Minimum	Maximum	Unit
IDDR	Read current	CE#=0.1VDD/0.9VDD@50MHz, SO open, VDD=3.6V	—	15	mA
IDDW	Program and erase current	CE=VDD=3.6V	—	30	mA
ISB	Standby current	CE=VDD=3.6V, VIN=VDD or VSS	—	8	mA
ILI	Input leakage current	VIN = VSS to VDD, VDD = 3.6V	—	1	μA
ILO	Output leakage current	VOUT = VSS to VDD, VDD = 3.6V	—	1	μA
VIL	Input low level voltage	VDD=2.7V	—	0.8	V
VIH	Input high level voltage	VDD=3.6V	0.7 VDD	—	V
VOL	Output low level voltage	VDD=2.7V, IOL=100μA	—	0.2	V
VOL2	Output low level voltage	VDD=2.7V, IOL=1.6mA	—	0.4	V
VOH	Output high level voltage	VDD=2.7V, IOL=-100μA	VDD-0.2	—	V

7.3. AC parameters

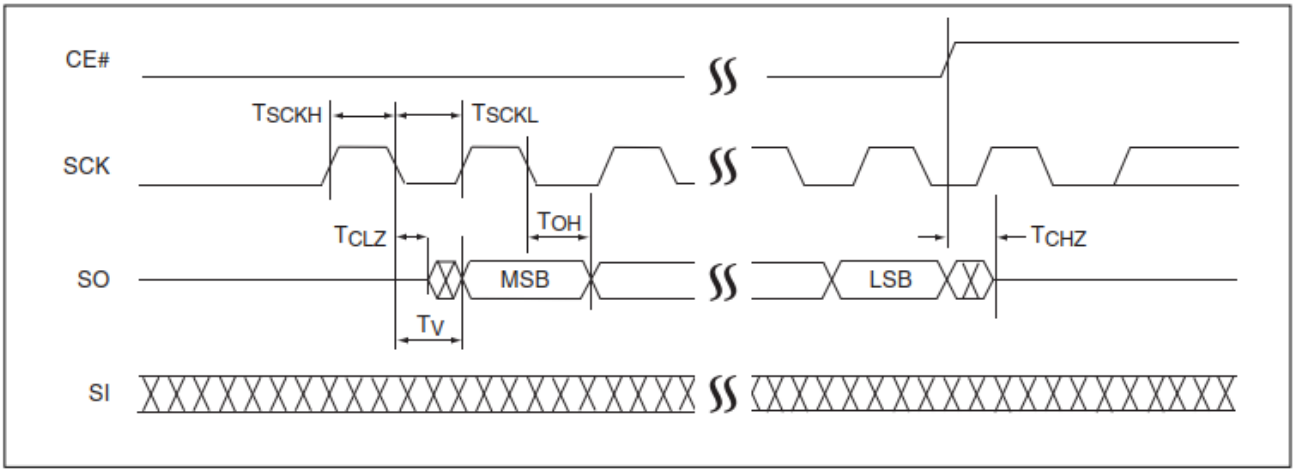
Table 5-2. AC parameters

Symbol	Characteristic	Condition	Minimum	Maximum	Unit
fCLK	Serial clock frequency	V_{DD}=3.3V	—	50	MHz
ikB	Serial clock high time	V_{DD}=3.3V, fCLK≤25MHz	18	—	ns
		V_{DD}=3.3V, 25MH<fCLK≤50MHz	9	—	ns

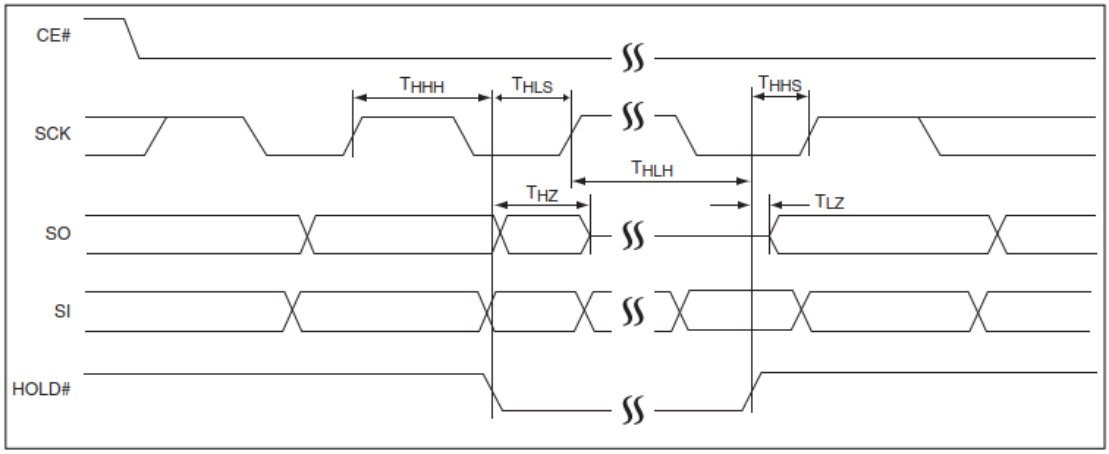
Symbol	Characteristic	Condition	Minimum	Maximum	Unit
tKL	Serial clock low time	V_{DD}=3.3V, fCLK≤25MHz	18	—	ns
		V_{DD}=3.3V, 25MH<fCLK≤50MHz	9	—	ns
tCES	CE valid establishment time	V_{DD}=3.3V	5	—	ns
CE	CE effective holding time	V_{DD}=3.3V	5	—	ns
tC	CE invalid establishment time	V_{DD}=3.3V	8	—	ns
HkDJ	CE invalid hold time	V_{DD}=3.3V	8	—	ns
tC	CE high level time	V_{DD}=3.3V	50	—	ns
oeLh	CE high level to high impedance output time	V_{DD}=3.3V	—	15	ns
tZ	SCK low level to low impedance output time	V_{DD}=3.3V	0	—	ns
tDS	Data creation time	V_{DD}=3.3V	2	—	ns
tD	Data retention time	V_{DD}=3.3V	5	—	ns
oeLh	HOLD low level establishment time	V_{DD}=3.3V	5	—	ns
HkDJ	HOLD high level establishment time	V_{DD}=3.3V	5	—	ns
HkDJ	HOLD low level holding time	V_{DD}=3.3V	5	—	ns
tHHH	HOLD high level holding time	V_{DD}=3.3V	5	—	ns
HkDJ	HOLD low level to high impedance output time	V_{DD}=3.3V	—	15	ns
tZ	HOLD high level to low impedance output time	V_{DD}=3.3V	—	15	ns
T	SCK changes to output hold time	V_{DD}=3.3V	0	—	ns
tV	SCK output valid time	V_{DD}=3.3V	—	12	ns
tT	Sector Erase	V_{DD}=3.3V	—	50	ms
tB	Block Erase	V_{DD}=3.3V	—	75	ms
tSCE	Full chip erase	V_{DD}=3.3V	—	75	ms
TbB	Byte Programming	V_{DD}=3.3V	—	75	μs



Picture 5-1. Serial Input Timing Diagram



Picture 5-2. Serial Output Timing Diagram

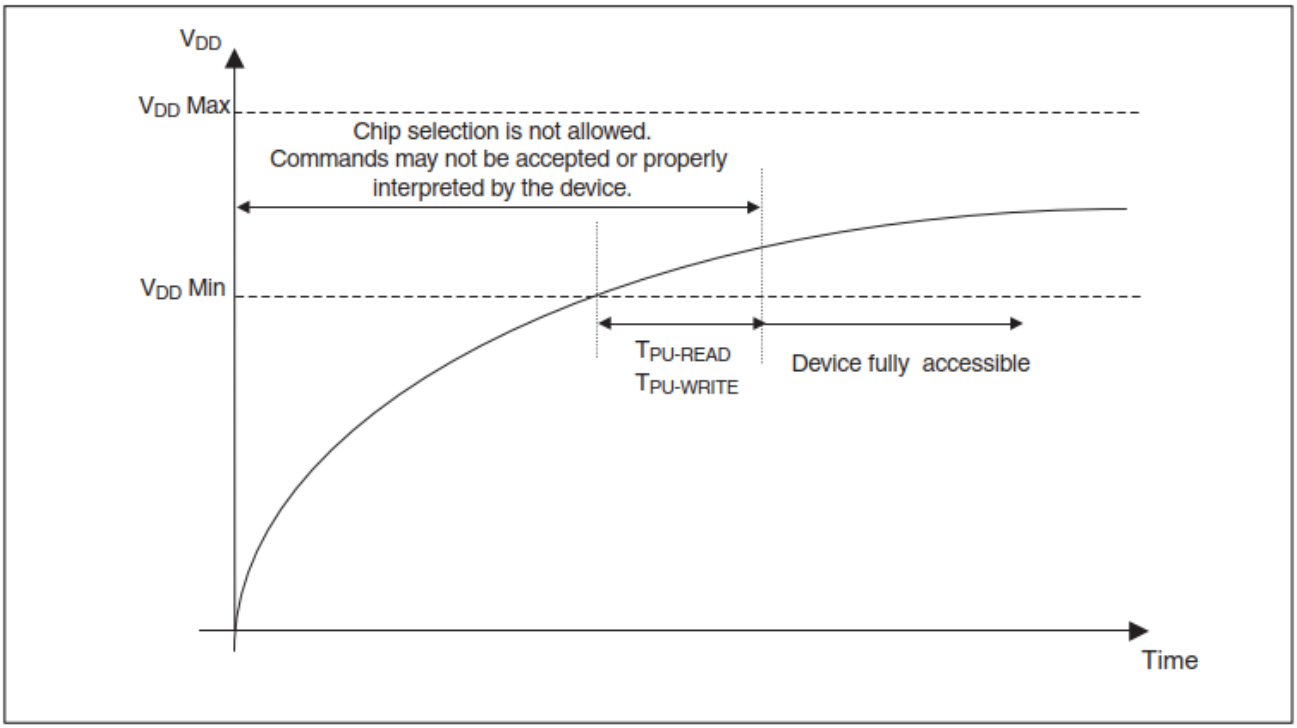


Picture 5-3. Hold (HOLD) Timing Diagram

7.4. AC parameters (reset/temporary release protection)

Table 5-3. System power-on parameters

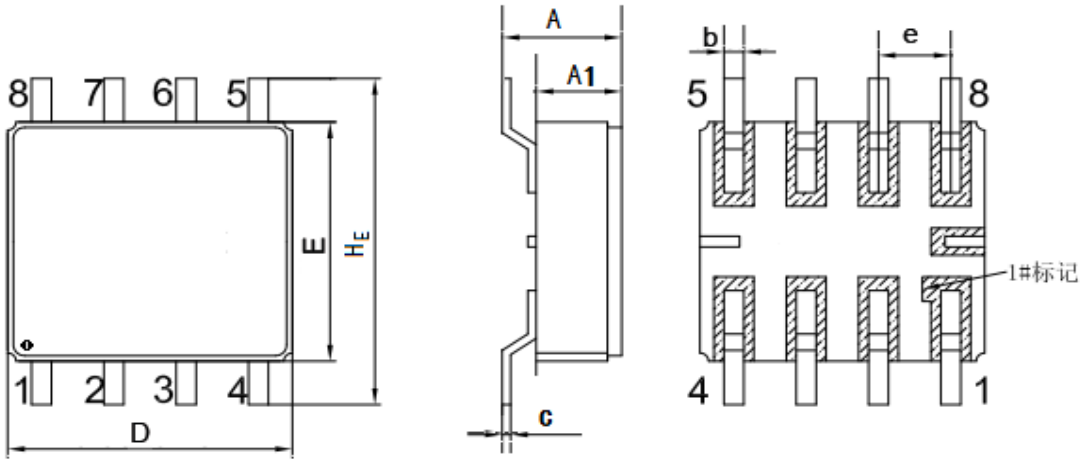
Conform To	Parameter	Minimum	Unit
TPU-READ	VDD minimum to read operation	1	ms
TPU-WRITE	VDD minimum to write operation	1	ms



Picture 5 4. Power-on timing diagram

VIII. Packaging information

8.1. CSOP8



Picture 6 1. CSOP8 package outline

Surface 6-1. CSOP8 Package Dimensions

Dimension Symbols	Value (Unit: Mm)		
	Minimum	Nominal	Maximum
D	5.05	-	5.35
E	5.05	-	5.35
HE	7.70	-	8.10
A	1.85	-	2.50
A1	-	-	1.80
b	0.25	-	0.45
c	0.10	-	0.20
e	-	1.27	-

IX. Precautions for use

The device must be operated with anti-static measures. The precautions for operating the device are as follows:

- The device should be operated on an anti-static workbench;

- Test equipment and apparatus should be grounded;
- Do not touch the device leads;
- Devices should be stored in containers made of conductive materials;
- Avoid using plastic, rubber or silk fabrics that can cause static electricity in the MOS area;
- If feasible, maintain relative humidity above 50%.

XX. Ordering Information

Table 8-1. Ordering Information

model	Package Name	Quality Grade	Detailed regulation number
BST25VF040B	CSOP8	GJB: B	Q/BST 20552-2020

Note 1:

- Q/B/B1 grade devices meet the screening requirements of Q/B/B1 grade of GJB 597A-1996 or GJB 597B-2012 "General Specification for Semiconductor Integrated Circuits".
- N/N1 level devices meet the N/N1 level screening requirements of GJB 7400 "General Specification for Semiconductor Integrated Circuits for Qualified Manufacturer Certification"
- I/Industrial temperature range devices meet the screening requirements of Q/BST40022-2021 "General Specifications for HUAWEI Industrial Temperature Range Products", and their operating environment temperature range should be -40℃～+85℃.
- Military temperature grade devices meet the screening requirements of

Q/BST40020-2018 "General Specifications for Huawei Military Temperature Grade Products", and their operating environment temperature range should be -55℃～+125℃.

Note 2:

The product ordering information is for our existing products or devices that have been confirmed to be developed. We can develop devices in other packaging forms according to user needs.