M95020-A125
M95020-A145

Automotive 2-Kbit serial SPI bus EEPROMs with high-speed clock

Datasheet - production data

SO8 (MN)
150 mil width

TSSOP8 (DW)
169 mil width

WFDFPN8 (MF)
2 x 3 mm
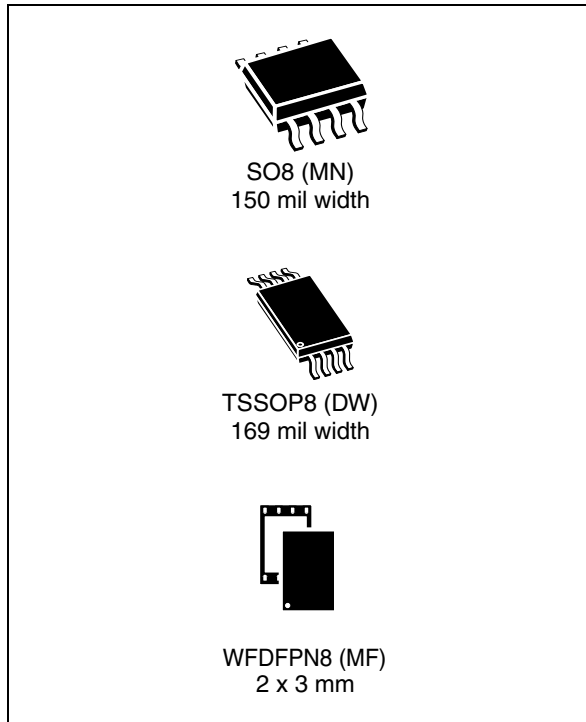
Features

- Compatible with the Serial Peripheral Interface (SPI) bus
- Memory array
  - 2 Kbit (256 Kbyte) of EEPROM
  - Page size: 16 byte
  - Write protection by block: 1/4, 1/2 or whole memory
  - Additional Write lockable Page (Identification page)
- Extended temperature and voltage ranges
  - Up to 125 °C (V_{CC} from 1.7 V to 5.5 V)
  - Up to 145 °C (V_{CC} from 2.5 V to 5.5 V)
- High speed clock frequency
  - 20 MHz for $V_{CC} \geq 4.5$ V
  - 10 MHz for $V_{CC} \geq 2.5$ V
  - 5 MHz for $V_{CC} \geq 1.7$ V
- Schmitt trigger inputs for noise filtering
- Short Write cycle time
  - Byte Write within 4 ms
  - Page Write within 4 ms
- Write cycle endurance
  - 4 million Write cycles at 25 °C
  - 1.2 million Write cycles at 85 °C
  - 600 k Write cycles at 125 °C
  - 400 k Write cycles at 145 °C
- Data retention
  - 50 years at 125 °C
  - 100 years at 25 °C
- ESD Protection (Human Body Model)
  - 4000 V
- Packages
  - RoHS-compliant and halogen-free (ECOPACK2®)

Contents

List of tables

List of figures

1 Description

The M95020-A125 and M95020-A145 are 2-Kbit serial EEPROM Automotive grade devices operating up to 145°C. They are compliant with the very high level of reliability defined by the Automotive standard AEC-Q100 grade 0.

The devices are accessed by a simple serial SPI compatible interface running up to 20 MHz.

The memory array is based on advanced true EEPROM technology (Electrically Erasable PROgrammable Memory). The M95020-A125 and M95020-A145 are byte-alterable memories (256 × 8 bits) organized as 16 pages of 16 byte in which the data integrity is significantly improved with an embedded Error Correction Code logic.

The M95020-A125 and M95020-A145 offer an additional Identification Page (16 byte) in which the ST device identification can be read. This page can also be used to store sensitive application parameters which can be later permanently locked in read-only mode.

Figure 1. Logic diagram
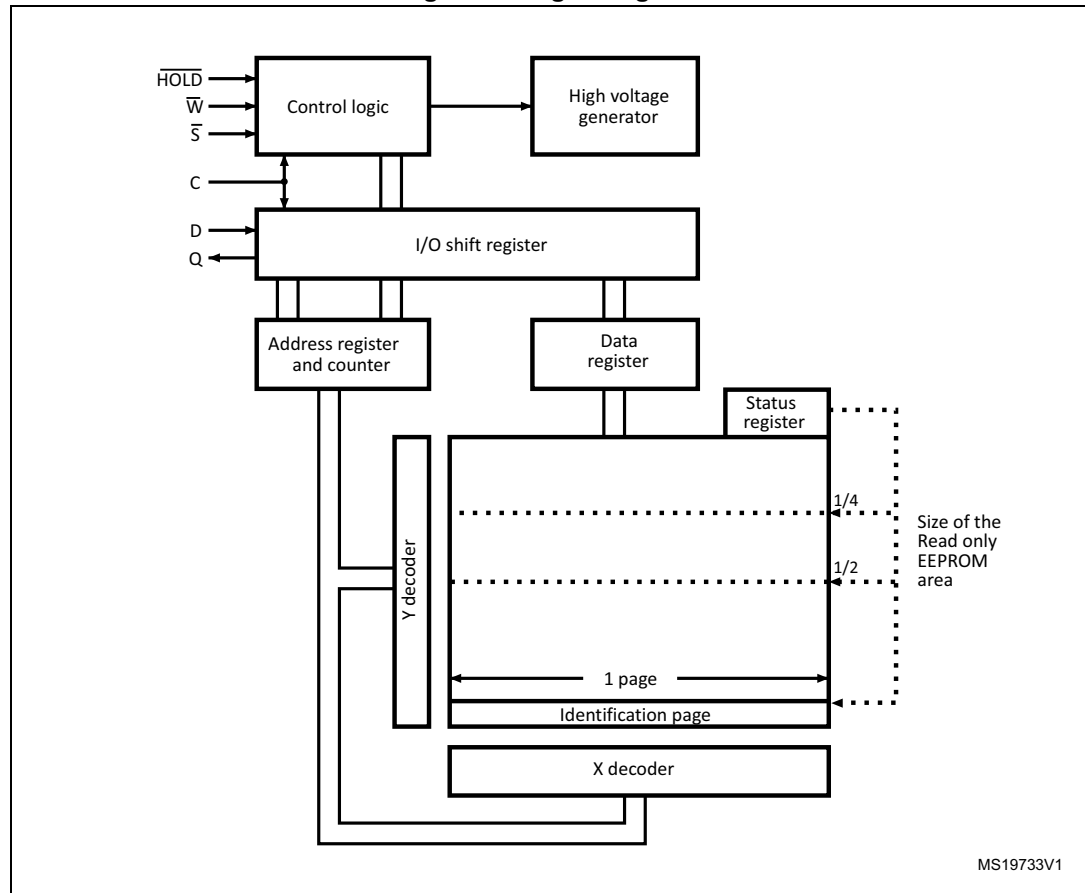
Figure 2. 8-pin package connections

```
                          M95xxx

                    ┌─────────────┐
             S̅ ─┤ 1         8 ├─ V_CC
             Q ─┤ 2         7 ├─ HOLD
             W̅ ─┤ 3         6 ├─ C
           V_SS ─┤ 4         5 ├─ D
                    └─────────────┘

                                        AI01790D
```
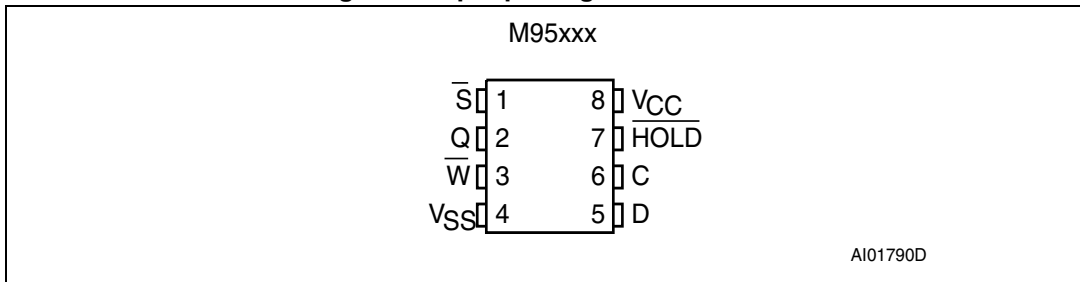
1. See *Package mechanical data* section for package dimensions and how to identify pin-1.

Table 1. Signal names

Signal name	Description
C	Serial Clock
D	Serial data input
Q	Serial data output
$\overline{S}$	Chip Select
$\overline{W}$	Write Protect
$\overline{HOLD}$	Hold
V_CC	Supply voltage
V_SS	Ground

2 Signal description

All input signals must be held high or low (according to voltages of V_{IH} or V_{IL}, as specified in *Table 12* and *Table 13*). These signals are described below.

2.1 Serial Data output (Q)

This output signal is used to transfer data serially out of the device during a Read operation. Data is shifted out on the falling edge of Serial Clock (C), most significant bit (MSB) first. In all other cases, the Serial Data output is in high impedance.

2.2 Serial Data input (D)

This input signal is used to transfer data serially into the device. D input receives instructions, addresses, and the data to be written. Values are latched on the rising edge of Serial Clock (C), most significant bit (MSB) first.

2.3 Serial Clock (C)

This input signal allows to synchronize the timing of the serial interface. Instructions, addresses, or data present at Serial Data Input (D) are latched on the rising edge of Serial Clock (C). Data on Serial Data Output (Q) changes after the falling edge of Serial Clock (C).

2.4 Chip Select ($\overline{S}$)

Driving Chip Select ($\overline{S}$) low selects the device in order to start communication. Driving Chip Select ($\overline{S}$) high deselects the device and Serial Data output (Q) enters the high impedance state.

2.5 Hold ($\overline{HOLD}$)

The Hold ($\overline{HOLD}$) signal is used to pause any serial communications with the device without deselecting the device. During the Hold condition, the Serial Data Output (Q) is high impedance, and the Serial Data Input (D) and Serial Clock (C) are Don't Care. To start the Hold condition, the device must be selected, with Chip Select (S) driven low.

2.6 Write Protect ($\overline{W}$)

This pin is used to write-protect the Status Register.

2.7 V_{SS} ground

V_{SS} is the reference for all signals, including the V_{CC} supply voltage.

2.8 V_{CC} supply voltage

V_{CC} is the supply voltage pin. Refer to *Section 3.1: Active power and Standby power modes* and to *Section 5.1: Supply voltage (VCC)*.

3   Operating features

3.1   Active power and Standby power modes

When Chip Select ($\overline{S}$) is low, the device is selected and in the Active power mode.

When Chip Select ($\overline{S}$) is high, the device is deselected. If a Write cycle is not currently in progress, the device then goes in to the Standby power mode, and the device consumption drops to I_{CC1}, as specified in *Table 12* and *Table 13*.

3.2   SPI modes

The device can be driven by a microcontroller with its SPI peripheral running in either of the two following modes:

- CPOL=0, CPHA=0
- CPOL=1, CPHA=1

For these two modes, input data is latched in on the rising edge of Serial Clock (C), and output data is available from the falling edge of Serial Clock (C).

The difference between the two modes, as shown in *Figure 3*, is the clock polarity when the bus master is in Stand-by mode and not transferring data:

- C remains at 0 for (CPOL=0, CPHA=0)
- C remains at 1 for (CPOL=1, CPHA=1)

Figure 3. SPI modes supported
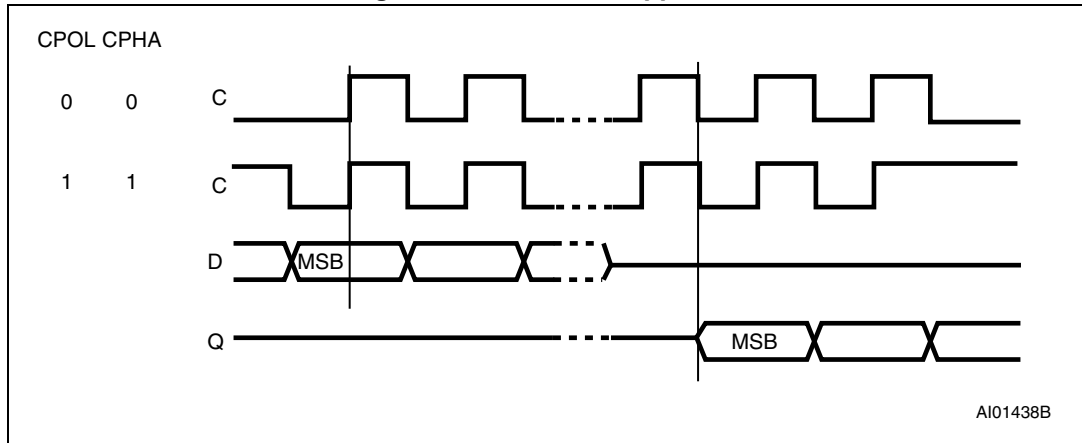
3.3   Hold mode

The Hold ($\overline{\text{HOLD}}$) signal is used to pause any serial communications with the device without resetting the clocking sequence.

The Hold mode starts when the Hold ($\overline{\text{HOLD}}$) signal is driven low and the Serial Clock (C) is low (as shown in *Figure 4*). During the Hold mode, the Serial Data output (Q) is high impedance, and the signals present on Serial Data input (D) and Serial Clock (C) are not decoded. The Hold mode ends when the Hold ($\overline{\text{HOLD}}$) signal is driven high and the Serial Clock (C) is or becomes low.

Figure 4. Hold mode activation
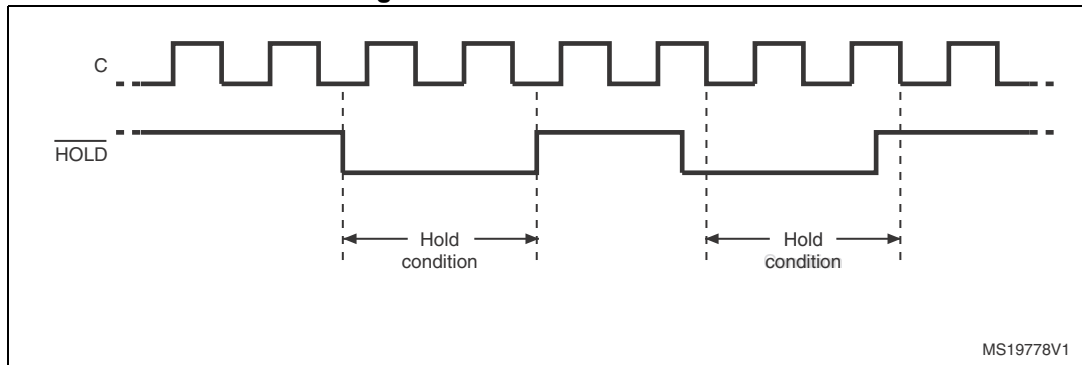
Deselecting the device while it is in Hold mode resets the paused communication.

3.4   Protocol control and data protection

3.4.1   Protocol control

The Chip Select ($\overline{\text{S}}$) input offers a built-in safety feature, as the $\overline{\text{S}}$ input is edge-sensitive as well as level-sensitive: after power-up, the device is not selected until a falling edge has first been detected on Chip Select ($\overline{\text{S}}$). This ensures that Chip Select ($\overline{\text{S}}$) must have been high prior to going low, in order to start the first operation.

For Write commands (WRITE, WRSR, WRID, LID) to be accepted and executed:

- the Write Enable Latch (WEL) bit must be set by a Write Enable (WREN) instruction
- a falling edge and a low state on Chip Select ($\overline{\text{S}}$) during the whole command must be decoded
- instruction, address and input data must be sent as multiple of eight bits
- the command must include at least one data byte
- Chip Select ($\overline{\text{S}}$) must be driven high exactly after a data byte boundary

Write command can be discarded at any time by a rising edge on Chip Select ($\overline{\text{S}}$) outside of a byte boundary.

To execute Read commands (READ, RDSR, RDID, RDLS), the device must decode:

- a falling edge and a low level on Chip Select ($\overline{\text{S}}$) during the whole command
- instruction and address as multiples of eight bits (byte)

From this step, data bits are shifted out until the rising edge on Chip Select ($\overline{\text{S}}$).

3.4.2 Status Register and data protection

The Status Register format is shown in *Table 2* and the status and control bits of the Status Register are as follows:

Table 2. Status Register format

b7	b6	b5	b4	b3	b2	b1	b0
1	1	1	1	BP1	BP0	WEL	WIP

Block Protect bits

Write Enable Latch bit

Write In Progress bit
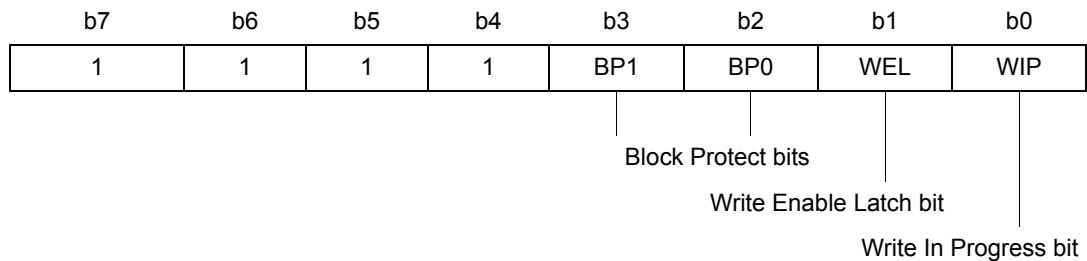
Note: *Bits b7, b6, b5 and b4 are always read as* 1*.*

WIP bit

The WIP bit (Write In Progress) is a read-only flag that indicates the Ready/Busy state of the device. When a Write command (WRITE, WRSR, WRID, LID) has been decoded and a Write cycle (t_W) is in progress, the device is busy and the WIP bit is set to 1. When WIP=0, the device is ready to decode a new command.

During a Write cycle, reading continuously the WIP bit allows to detect when the device becomes ready (WIP=0) to decode a new command.

WEL bit

The WEL bit (Write Enable Latch) bit is a flag that indicates the status of the internal Write Enable Latch. When WEL is set to 1, the Write instructions (WRITE, WRSR, WRID, LID) are executed; when WEL is set to 0, any decoded Write instruction is not executed.

The WEL bit is set to 1 with the WREN instruction. The WEL bit is reset to 0 after the following events:

- Write Disable (WRDI) instruction completion
- Write instructions (WRITE, WRSR, WRID, LID) completion including the write cycle time t_W
- Power-up

BP1, BP0 bits

The Block Protect bits (BP1, BP0) are non-volatile. BP1,BP0 bits define the size of the memory block to be protected against write instructions, as defined in *Table 2*. These bits are written with the Write Status Register (WRSR) instruction.

Table 3. Write-protected block size

Status Register bits		Protected block	Protected array addresses
BP1	**BP0**		
0	0	None	None
0	1	Upper quarter	C0h - FFh
1	0	Upper half	80h - FFh
1	1	Whole memory	00h - FFh plus Identification page

3.5 Identification page

The M95020-A125 and M95020-A145 offer an Identification page (16 byte) in addition to the 2 Kbit memory. The Identification page contains two fields:

- Device identification: the three first byte are programmed by STMicroelectronics with the Device identification code, as shown in *Table 4*.

- Application parameters: the bytes after the Device identification code are available for application specific data.

Note: *If the end application does not need to read the Device identification code, this field can be overwritten and used to store application-specific data. Once the application-specific data are written in the Identification page, the whole Identification page should be permanently locked in Read-only mode.*

The Read, Write, Lock Identification Page instructions are detailed in *Section 4: Instructions*.

Table 4. Device identification bytes

Address in Identification page	Content	Value
00h	ST Manufacturer code	20h
01h	SPI Family code	00h
02h	Memory Density code	08h (2 Kbit)

4 Instructions

Each command is composed of bytes (MSBit transmitted first), initiated with the instruction byte, as summarized in *Table 5*.

If an invalid instruction is sent (one not contained in *Table 5*), the device automatically enters a Wait state until deselected.

Table 5. Instruction set

Instruction	Description	Instruction format
WREN	Write Enable	0000 X110[1]
WRDI	Write Disable	0000 X100[1]
RDSR	Read Status Register	0000 X101[1]
WRSR	Write Status Register	0000 X001[1]
READ	Read from Memory Array	0000 X011
WRITE	Write to Memory Array	0000 X010
RDID	Read Identification Page	1000 0011
WRID	Write Identification Page	1000 0010
RDLS	Reads the Identification Page lock status.	1000 0011
LID	Locks the Identification page in read-only mode.	1000 0010

1. X = Don't Care.

For read and write commands to memory array and Identification Page, the address is defined by one byte as explained in *Table 6*.

Table 6. Significant bits within the address bytes[1]

| Instructions	Bit b3 of the instruction byte	Address byte							
		b7	b6	b5	b4	b3	b2	b1	b0
READ or WRITE	x	A7	A6	A5	A4	A3	A2	A1	A0
RDID or WRID	0	0	0	0	0	A3	A2	A1	A0
RDLS or LID	0	1	0	0	0	0	0	0	0

1. A: Significant address bit.

4.1 Write Enable (WREN)

The WREN instruction must be decoded by the device before a write instruction (WRITE, WRSR, WRID or LID).

As shown in *Figure 5*, to send this instruction to the device, Chip Select ($\overline{S}$) is driven low, the bits of the instruction byte are shifted in (MSB first) on Serial Data Input (D) after what the Chip Select ($\overline{S}$) input is driven high and the WEL bit is set (Status Register bit).

Figure 5. Write Enable (WREN) sequence
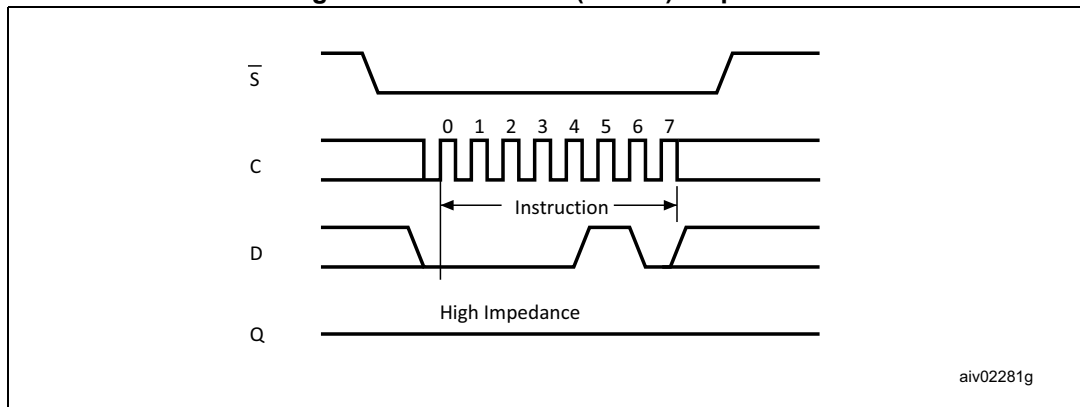
4.2 Write Disable (WRDI)

One way of resetting the WEL bit (in the Status Register) is to send a Write Disable instruction to the device.

As shown in *Figure 6*, to send this instruction to the device, Chip Select ($\overline{S}$) is driven low, and the bits of the instruction byte are shifted in (MSB first), on Serial Data Input (D), after what the Chip Select ($\overline{S}$) input is driven high and the WEL bit is reset (Status Register bit).

If a Write cycle is currently in progress, the WRDI instruction is decoded and executed and the WEL bit is reset to 0 with no effect on the ongoing Write cycle.

In fact, the Write Enable Latch (WEL) bit becomes reset by any of the following events:

- Power-up
- WRDI instruction execution
- WRSR instruction completion
- WRITE instruction completion
- Write Protect (W) line being held low.

Figure 6. Write Disable (WRDI) sequence
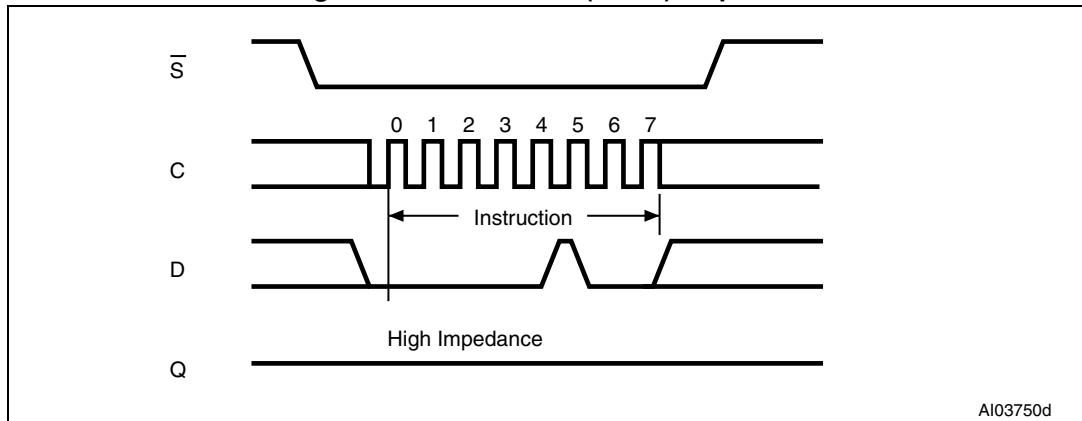
4.3   Read Status Register (RDSR)

The Read Status Register (RDSR) instruction is used to read the content of the Status Register.

As shown in *Figure 7*, to send this instruction to the device, Chip Select ($\overline{\text{S}}$) is first driven low. The bits of the instruction byte are shifted in (MSB first) on Serial Data Input (D), the Status Register content is then shifted out (MSB first) on Serial Data Output (Q).

If Chip Select ($\overline{\text{S}}$) continues to be driven low, the Status Register content is continuously shifted out.

The Status Register can always be read, even if a Write cycle (t_W) is in progress. The Status Register functionality is detailed in *Section 3.4.2: Status Register and data protection*.

Figure 7. Read Status Register (RDSR) sequence
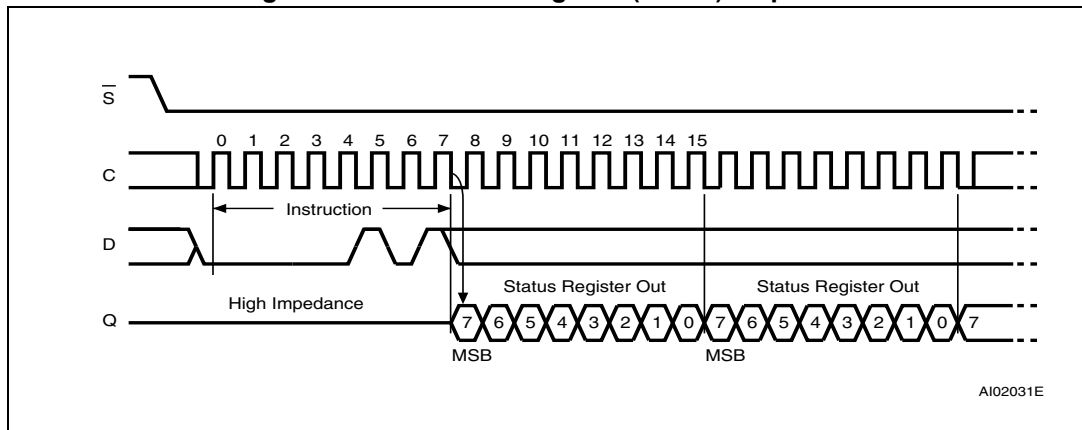
4.4   Write Status Register (WRSR)

The Write Status Register (WRSR) instruction allows new values to be written to the Status Register. Before it can be accepted, a Write Enable (WREN) instruction must previously have been executed.

The Write Status Register (WRSR) instruction is entered (MSB first) by driving Chip Select ($\overline{S}$) low, sending the instruction code followed by the data byte on Serial Data input (D), and driving the Chip Select ($\overline{S}$) signal high.

This instruction allows the user to change the values of the BP1 and BP0 bits which define the size of the area that is to be treated as read only, as defined in *Table 3: Write-protected block size*.

The contents of the BP1, BP0 bits are updated after the completion of the WRSR instruction, including the Write cycle (t_W).

The Write Status Register (WRSR) instruction has no effect on the b6, b5, b4, b1 and b0 bits in the Status Register (see *Table 2: Status Register format*). Bits b7, b6, b5, b4 are always read as 1.

The Status Register functionality is detailed in *Section 3.4.2: Status Register and data protection*.

The instruction is not accepted, and is not executed, under the following conditions:

- if the Write Enable Latch (WEL) bit has not been set to 1 (by executing a Write Enable instruction just before)
- if a write cycle is already in progress
- if the device has not been deselected, by Chip Select (S) being driven high, after the eighth bit, b0, of the data byte has been latched in
- if Write Protect (W) is low during the WRSR command (instruction, address and data)

Figure 8. Write Status Register (WRSR) sequence
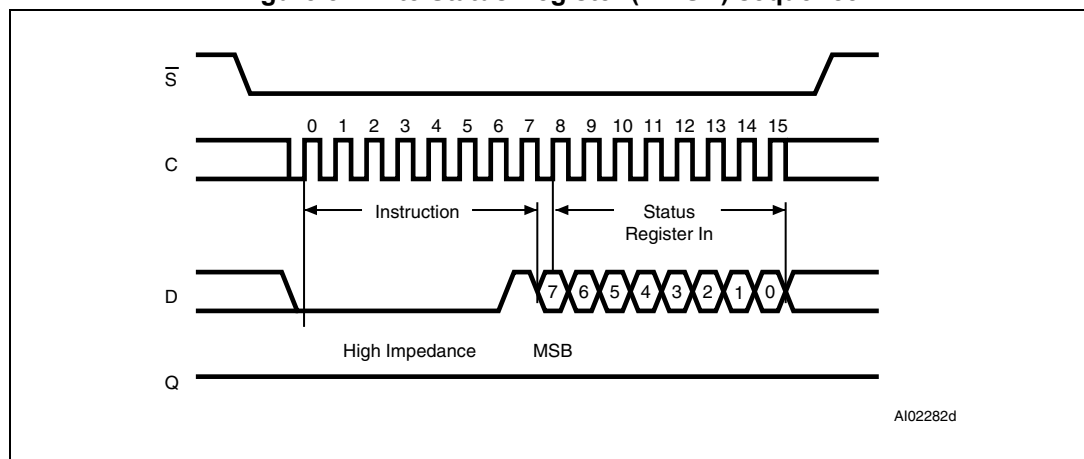
4.5   Read from Memory Array (READ)

The READ instruction is used to read the content of the memory.

As shown in *Figure 9*, to send this instruction to the device, Chip Select ($\overline{S}$) is first driven low.

The bits of the instruction byte and address byte are then shifted in, on Serial Data Input (D).The address is loaded into an internal address register, and the byte of data at that address is shifted out, on Serial Data Output (Q).

If Chip Select ($\overline{S}$) continues to be driven low, the internal address register is automatically incremented, and the next byte of data is shifted out. The whole memory can therefore be read with a single READ instruction.

When the highest address is reached, the address counter rolls over to zero, allowing the Read cycle to be continued indefinitely.

The Read cycle is terminated by driving Chip Select ($\overline{S}$) high at any time when the data bits are shifted out on Serial Data Output (Q).

The instruction is not accepted, and is not executed, if a Write cycle is currently in progress.

Figure 9. Read from Memory Array (READ) sequence
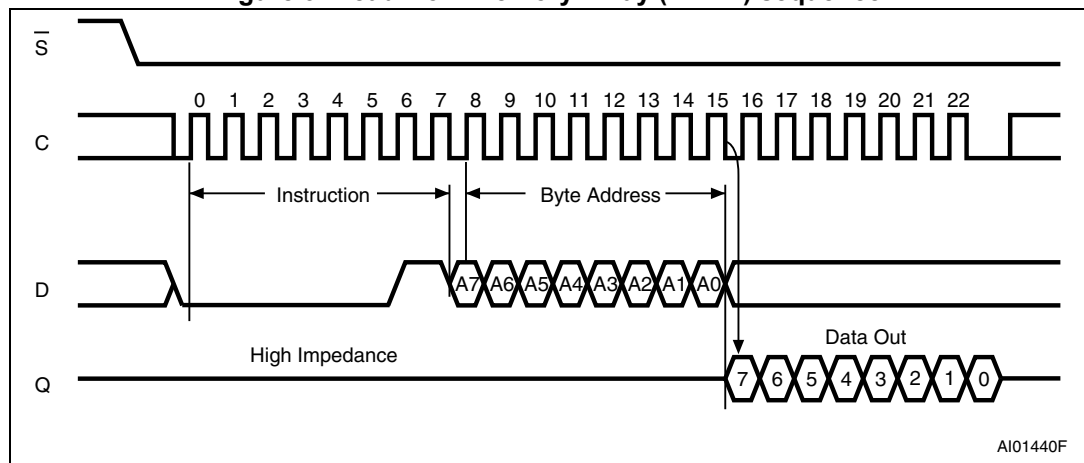
4.6 Write to Memory Array (WRITE)

The WRITE instruction is used to write new data in the memory.

As shown in *Figure 10*, to send this instruction to the device, Chip Select ($\overline{S}$) is first driven low. The bits of the instruction byte, address byte, and at least one data byte are then shifted in (MSB first), on Se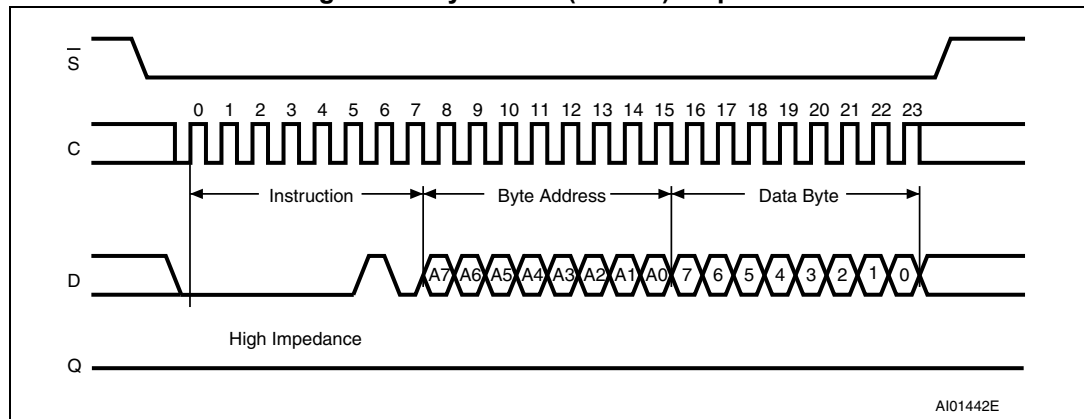rial Data Input (D). The instruction is terminated by driving Chip Select ($\overline{S}$) high at a data byte boundary. *Figure 10* shows a single byte write.

Figure 10. Byte Write (WRITE) sequence

A Page write is used to write several bytes inside a page, with a single internal Write cycle.

For a Page write, Chip Select ($\overline{S}$) has to remain low, as shown in *Figure 11*, so that the next data bytes are shifted in. Each time a new data byte is shifted in, the least significant bits of the internal address counter are incremented. If the address counter exceeds the page boundary (the page size is 16 byte), the internal address pointer rolls over to the beginning of the same page where next data bytes will be written. If more than 16 byte are received, only the last 16 byte are written.

For both Byte write and Page write, the self-timed Write cycle starts from the rising edge of Chip Select ($\overline{S}$), and continues for a period t_W (as specified in *Table 14*).

The instruction is discarded, and is not executed, under the following conditions:

- if a Write cycle is already in progress
- if Write Protect (W) is low or if the addressed page is in the area protected by the Block Protect (BP1 and BP0) bits
- if one of the conditions defined in *Section 3.4.1* is not satisfied

Note: *The self-timed Write cycle t_W is internally executed as a sequence of two consecutive events: [Erase addressed byte(s)], followed by [Program addressed byte(s)]. An erased bit is read as "0" and a programmed bit is read as "1".*

Figure 11. Page Write (WRITE) sequence
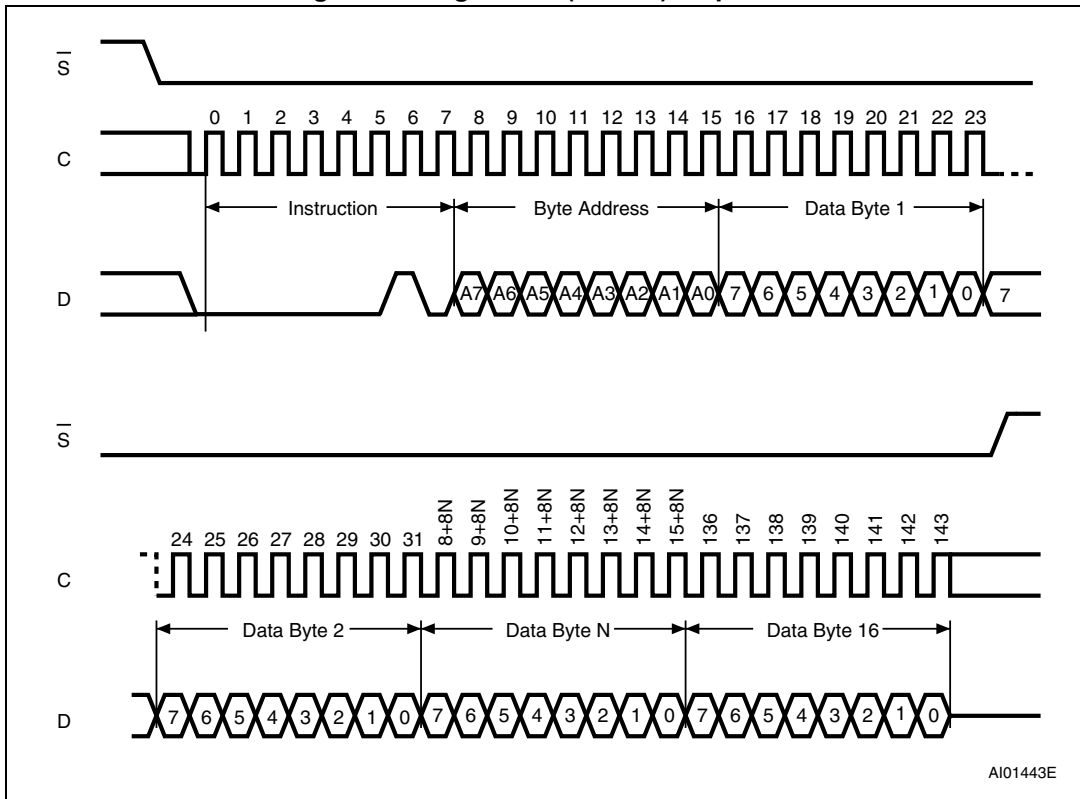
4.7   Read Identification Page (RDID)

The Read Identification Page instruction is used to read the Identification Page (additional page of 16 byte which can be written and later permanently locked in Read-only mode).

The Chip Select ($\overline{S}$) signal is first driven low, the bits of the instruction byte and address bytes are then shifted in (MSB first) on Serial Data input (D). Address bit A7 must be 0 and the other address bits are Don't Care except the lower address bits [A3:A0] (it might be easier to define these bits as 0, as shown in *Table 6*). Data are then shifted/clocked out (MSB first) on Serial Data output (Q).

The first byte addressed can be any byte within the identification page.

If Chip Select ($\overline{S}$) continues to be driven low, the internal address register is automatically incremented and the byte of data at the new address is shifted out.

Note that there is no roll over feature in the Identification Page. The address of bytes to read must not exceed the page boundary.

The read cycle is terminated by driving Chip Select ($\overline{S}$) high. The rising edge of the Chip Select ($\overline{S}$) signal can occur at any time when the data bits are shifted out.

The instruction is not accepted, and is not executed, if a Write cycle is currently in progress.

Figure 12. Read Identification Page sequence
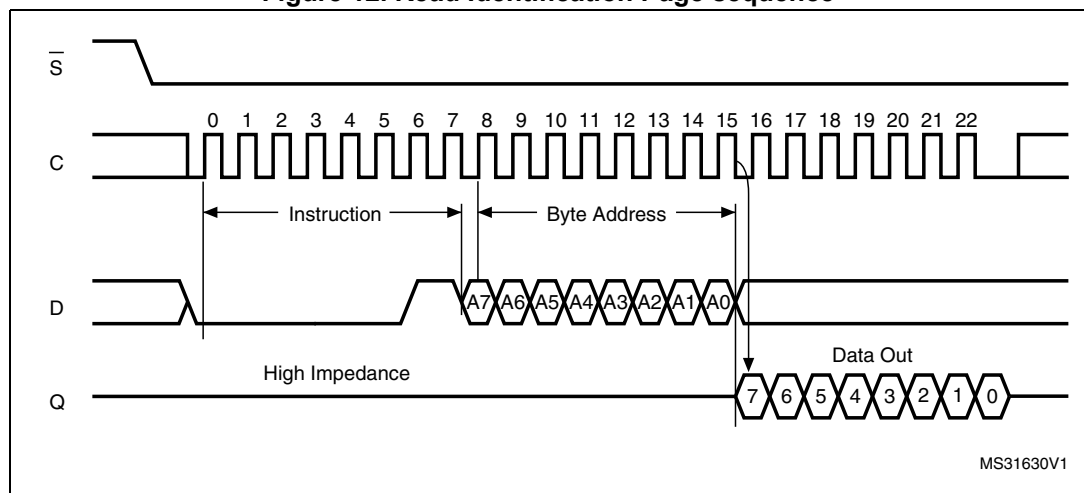
The first three bytes of the Identification page offer information about the device itself. Please refer to *Section 3.5: Identification page* for more information.

4.8   Write Identification Page (WRID)

The Write Identification Page instruction is used to write the Identification Page (additional page of 16 byte which can also be permanently locked in Read-only mode).

The Chip Select signal ($\overline{S}$) is first driven low, and then the bits of the instruction byte, address bytes, and at least one data byte are shifted in (MSB first) on Serial Data input (D). Address bit A7 must be 0 and the other address bits are Don't Care except the lower address bits [A3:A0] (it might be easier to define these bits as 0, as shown in *Table 6*).

The self-timed Write cycle starts from the rising edge of Chip Select ($\overline{S}$), and continues for a period t_W (as specified in *Table 14*).

Figure 13. Write Identification Page sequence
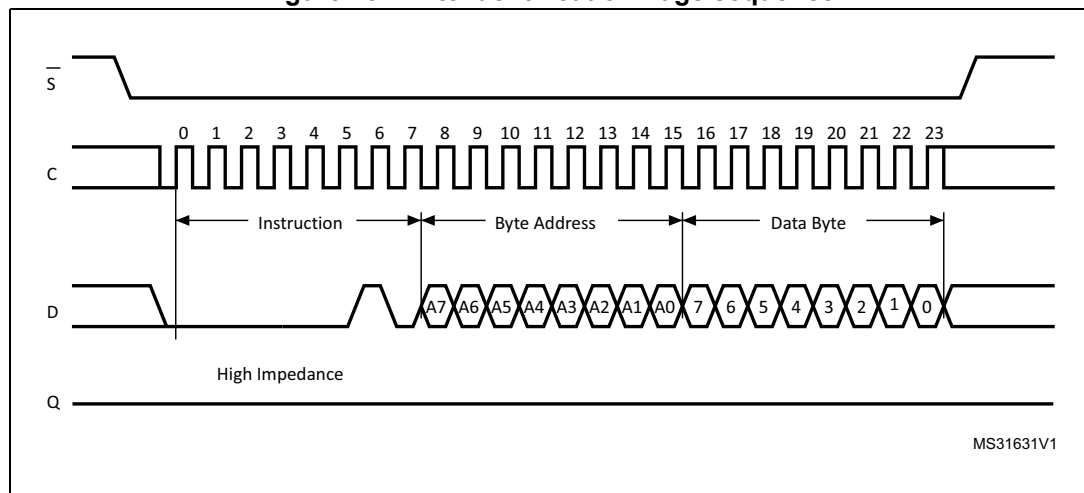
The instruction is discarded, and is not executed, under the following conditions:

- If a Write cycle is already in progress
- If the Block Protect bits (BP1,BP0) = (1,1)
- If one of the conditions defined in *Section 3.4.1: Protocol control* is not satisfied.

4.9   Read Lock Status (RDLS)

The Read Lock Status instruction is used to read the lock status.

To send this instruction to the device, Chip Select ($\overline{S}$) first has to be driven low. The bits of the instruction byte and address bytes are then shifted in (MSB first) on Serial Data input (D). Address bit A7 must be 1; all other address bits are Don't Care (it might be easier to define these bits as 0, as shown in *Table 6*). The Lock bit is the LSB (Least Significant Bit) of the byte read on Serial Data output (Q). It is at '1' when the lock is active and at '0' when the lock is not active. If Chip Select ($\overline{S}$) continues to be driven low, the same data byte is shifted out.

The read cycle is terminated by driving Chip Select ($\overline{S}$) high. The instruction sequence is shown in *Figure 14*.

The Read Lock Status instruction is not accepted and not executed if a Write cycle is currently in progress.

Figure 14. Read Lock Status sequence
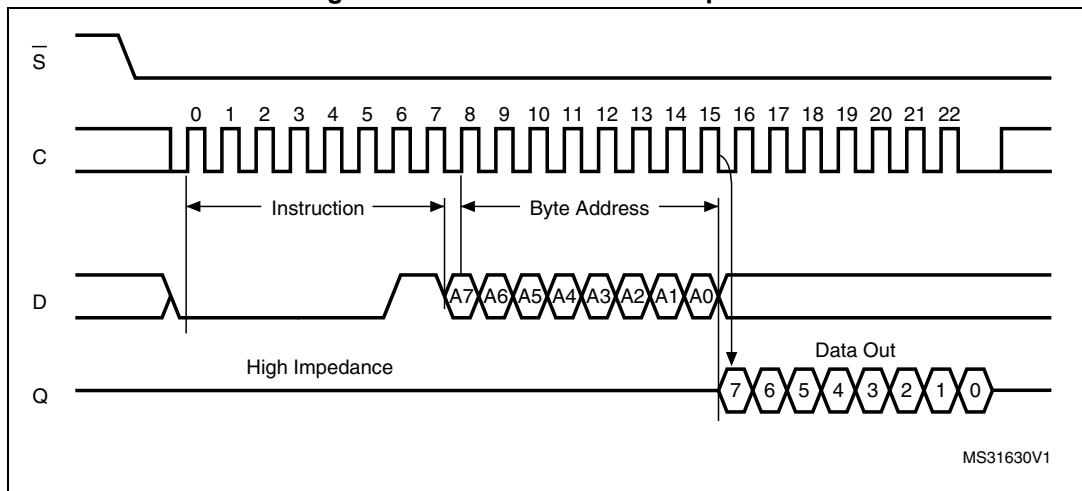
4.10 Lock Identification Page (LID)

The Lock Identification Page (LID) command is used to permanently lock the Identification Page in Read-only mode.

The LID instruction is issued by driving Chip Select (S) low, sending (MSB first) the instruction code, the address and a data byte on Serial Data input (D), and driving Chip Select (S) high. In the address sent, A7 must be equal to 1. All other address bits are Don't Care (it might be easier to define these bits as 0, as shown in *Table 6*). The data byte sent must be equal to the binary value xxxx xx1x, where x = Don't Care. The LID instruction is terminated by driving Chip Select (S) high at a data byte boundary, otherwise, the instruction is not executed.

Figure 15. Lock ID sequence
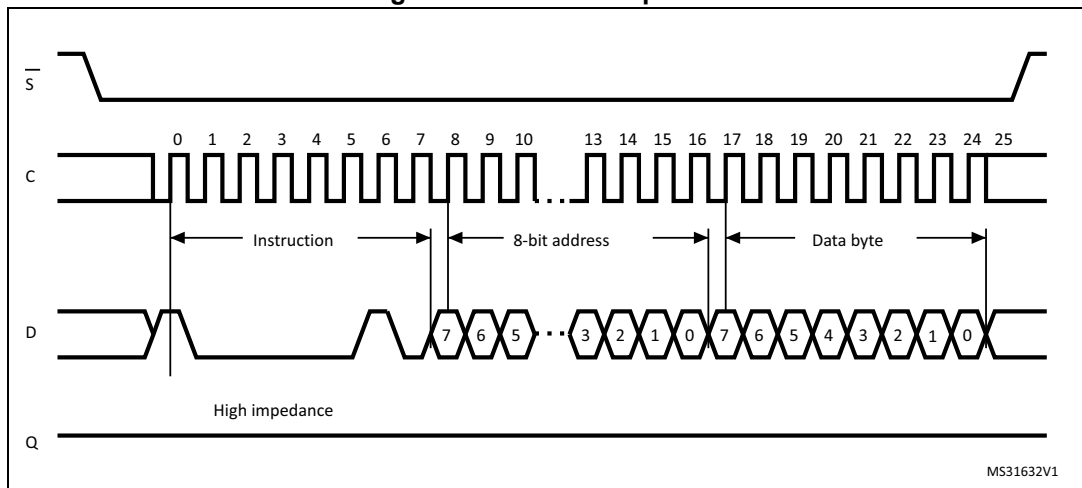
Driving Chip Select (S) high at a byte boundary of the input data triggers the self-timed Write cycle which duration is t_W (specified in *Table 14*). The instruction sequence is shown in *Figure 15*.

The instruction is discarded, and is not executed, under the following conditions:

- If a Write cycle is already in progress
- If the Block Protect bits (BP1,BP0) = (1,1)
- If one of the conditions defined in *Section 3.4.1: Protocol control* is not satisfied.

5   Application design recommendations

5.1   Supply voltage (V$_{CC}$)

5.1.1   Operating supply voltage (V$_{CC}$)

Prior to selecting the memory and issuing instructions to it, a valid and stable V$_{CC}$ voltage within the specified [V$_{CC(min)}$, V$_{CC(max)}$] range must be applied (see *Table 9* and *Table 10*).

This voltage must remain stable and valid until the end of the transmission of the instruction and, for a Write instruction, until the completion of the internal Write cycle (t$_W$). In order to secure a stable DC supply voltage, it is recommended to decouple the V$_{CC}$ line with a suitable capacitor (usually of the order of 10 nF to 100 nF) close to the V$_{CC}$/V$_{SS}$ package pins.

5.1.2   Power-up conditions

When the power supply is turned on, V$_{CC}$ continuously rises from V$_{SS}$ to V$_{CC}$. During this time, the Chip Select ($\overline{S}$) line is not allowed to float but should follow the V$_{CC}$ voltage. It is therefore recommended to connect the $\overline{S}$ line to V$_{CC}$ via a suitable pull-up resistor (see *Figure 16*).

The V$_{CC}$ voltage has to rise continuously from 0 V up to the minimum V$_{CC}$ operating voltage defined in *Table 12* and *Table 13*.

In order to prevent inadvertent write operations during power-up, a power-on-reset (POR) circuit is included.

At power-up, the device does not respond to any instruction until V$_{CC}$ reaches the internal threshold voltage (this threshold is defined in the DC characteristics tables *12* and *13* as VRES).

When V$_{CC}$ passes over the POR threshold, the device is reset and in the following state:

*   in the Standby power mode
*   deselected
*   Status register values:
    –   Write Enable Latch (WEL) bit is reset to 0.
    –   Write In Progress (WIP) bit is reset to 0.
    –   BP1 and BP0 bits remain unchanged (non-volatile bits).
*   not in the Hold condition

As soon as the V$_{CC}$ voltage has reached a stable value within [V$_{CC}$(min), V$_{CC}$(max)] range, the device is ready for operation.