Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!


## Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

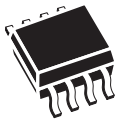
# M95320-W M95320-R M95320-DF

## 32-Kbit serial SPI bus EEPROM with high-speed clock

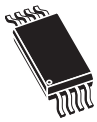
**Datasheet – production data**

## Features

- Compatible with the Serial Peripheral Interface (SPI) bus
- Memory array
  - 32 Kb (4 Kbytes) of EEPROM
  - Page size: 32 bytes
- Write
  - Byte Write within 5 ms
  - Page Write within 5 ms
- Additional Write lockable page (Identification page)
- Write Protect: quarter, half or whole memory array
- High-speed clock: 20 MHz
- Single supply voltage:
  - 2.5 V to 5.5 V for M95320-W
  - 1.8 V to 5.5 V for M95320-R
  - 1.7 V to 5.5 V for M95320-DF
- Operating temperature range: from -40°C up to +85°C
- Enhanced ESD protection
- More than 4 million Write cycles
- More than 200-year data retention
- Packages
  - RoHS compliant and halogen-free (ECOPACK®)



SO8 (MN)
150 mil width



TSSOP8 (DW)
169 mil width

UFDFPN8
2 × 3 mm (MLP)

# Contents

# List of tables

# List of figures

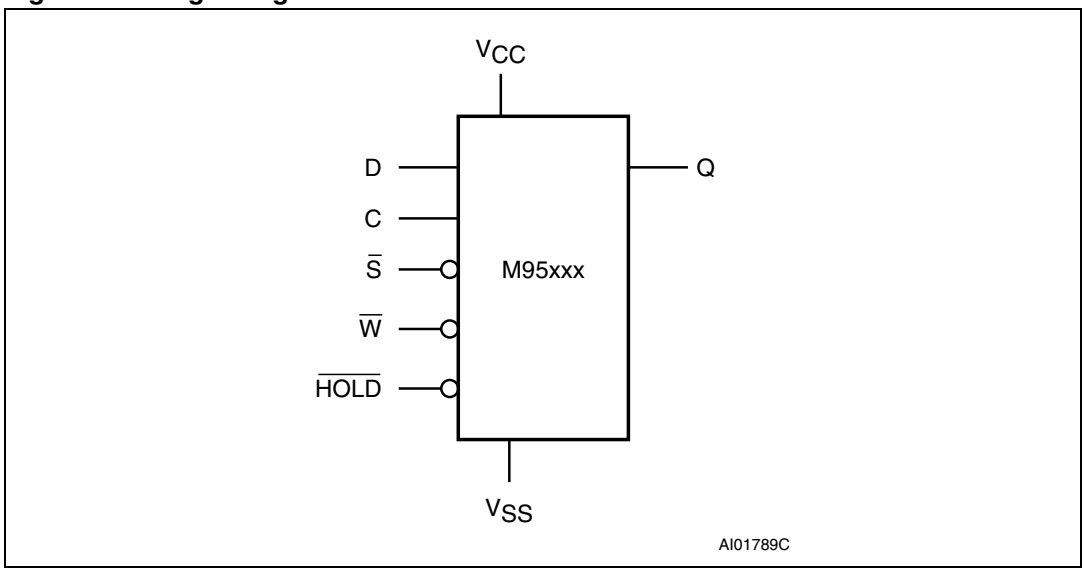
# 1 Description

The M95320 devices are Electrically Erasable PROgrammable Memories (EEPROMs) organized as 4096 x 8 bits, accessed through the SPI bus.

The M95320 devices can operate with a supply range from 1.7 V up to 5.5 V, and are guaranteed over the -40 °C/+85 °C temperature range.

The M95320-D offers an additional page, named the Identification Page (32 bytes). The Identification Page can be used to store sensitive application parameters which can be (later) permanently locked in Read-only mode.

**Figure 1. Logic diagram**

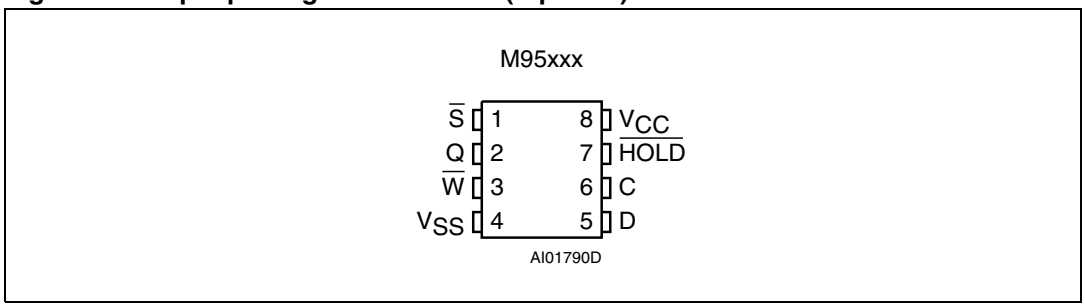



The SPI bus signals are C, D and Q, as shown in *Figure 1* and *Table 1*. The device is selected when Chip Select ($\overline{S}$) is driven low. Communications with the device can be interrupted when the $\overline{HOLD}$ is driven low.

**Table 1. Signal names**

| Signal name | Function | Direction |
|---|---|---|
| C | Serial Clock | Input |
| D | Serial Data Input | Input |
| Q | Serial Data Output | Output |
| $\overline{S}$ | Chip Select | Input |
| $\overline{W}$ | Write Protect | Input |
| $\overline{HOLD}$ | Hold | Input |
| $V_{CC}$ | Supply voltage | |
| $V_{SS}$ | Ground | |

**Figure 2. 8-pin package connections (top view)**

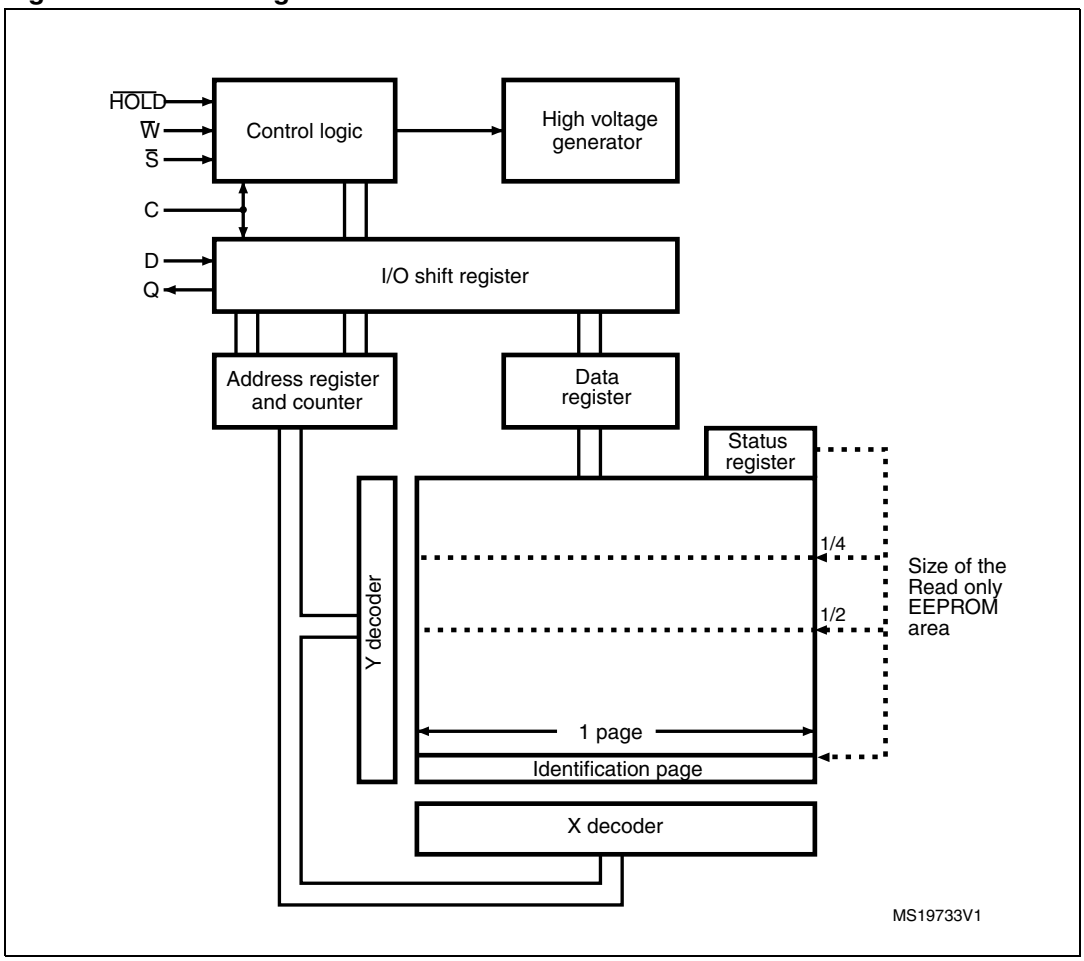



1. See *Section 10: Package mechanical data* section for package dimensions, and how to identify pin 1.

# 2 Memory organization

The memory is organized as shown in the following figure.

**Figure 3. Block diagram**



# 3 Signal description

During all operations, $V_{CC}$ must be held stable and within the specified valid range: $V_{CC}$(min) to $V_{CC}$(max).

All of the input and output signals must be held high or low (according to voltages of $V_{IH}$, $V_{OH}$, $V_{IL}$ or $V_{OL}$, as specified in *Section 9: DC and AC parameters*). These signals are described next.

## 3.1 Serial Data Output (Q)

This output signal is used to transfer data serially out of the device. Data is shifted out on the falling edge of Serial Clock (C).

## 3.2 Serial Data Input (D)

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

## 3.3 Serial Clock (C)

This input signal provides 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) change from the falling edge of Serial Clock (C).

## 3.4 Chip Select ($\overline{S}$)

When this input signal is high, the device is deselected and Serial Data Output (Q) is at high impedance. The device is in the Standby Power mode, unless an internal Write cycle is in progress. Driving Chip Select ($\overline{S}$) low selects the device, placing it in the Active Power mode.

After power-up, a falling edge on Chip Select ($\overline{S}$) is required prior to the start of any instruction.

## 3.5 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 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 ($\overline{S}$) driven low.

## 3.6 Write Protect ($\overline{W}$)

The main purpose of this input signal is to freeze the size of the area of memory that is protected against Write instructions (as specified by the values in the BP1 and BP0 bits of the Status Register).

This pin must be driven either high or low, and must be stable during all Write instructions.

## 3.7 $V_{CC}$ supply voltage

$V_{CC}$ is the supply voltage.

## 3.8 $V_{SS}$ ground

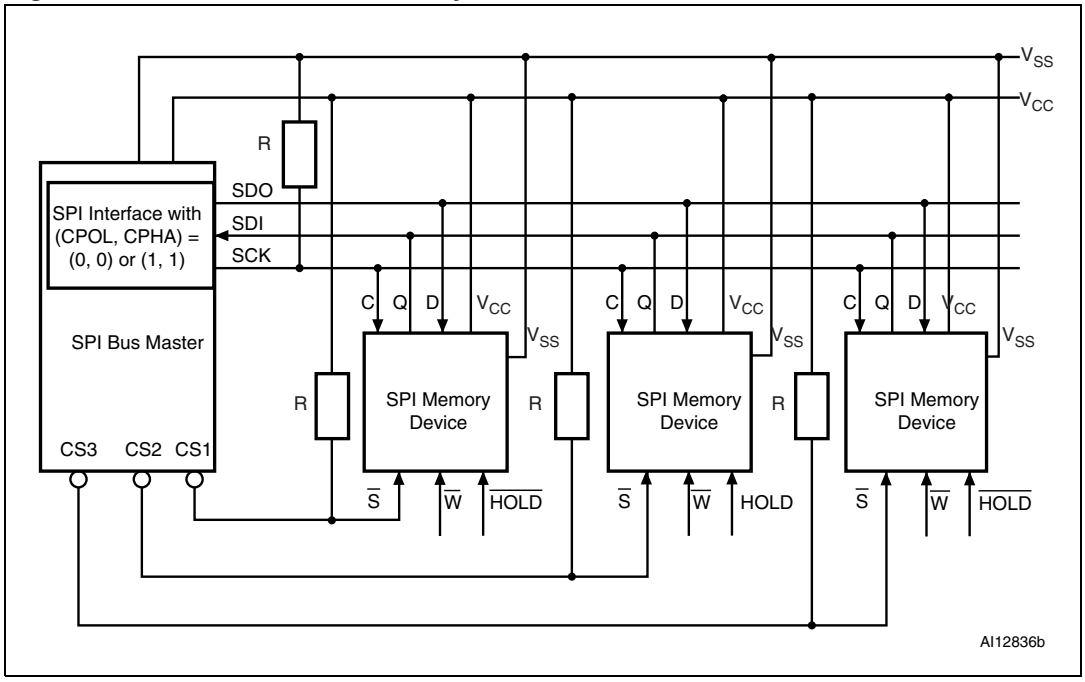
$V_{SS}$ is the reference for all signals, including the $V_{CC}$ supply voltage.

# 4 Connecting to the SPI bus

All instructions, addresses and input data bytes are shifted in to the device, most significant bit first. The Serial Data Input (D) is sampled on the first rising edge of the Serial Clock (C) after Chip Select ($\overline{S}$) goes low.

All output data bytes are shifted out of the device, most significant bit first. The Serial Data Output (Q) is latched on the first falling edge of the Serial Clock (C) after the instruction (such as the Read from Memory Array and Read Status Register instructions) have been clocked into the device.

**Figure 4. Bus master and memory devices on the SPI bus**




1. The Write Protect ($\overline{W}$) and Hold ($\overline{HOLD}$) signals should be driven, high or low as appropriate.

*Figure 4* shows an example of three memory devices connected to an SPI bus master. Only one memory device is selected at a time, so only one memory device drives the Serial Data Output (Q) line at a time. The other memory devices are high impedance.

The pull-up resistor R (represented in *Figure 4*) ensures that a device is not selected if the Bus Master leaves the $\overline{S}$ line in the high impedance state.

In applications where the Bus Master may leave all SPI bus lines in high impedance at the same time (for example, if the Bus Master is reset during the transmission of an instruction), the clock line (C) must be connected to an external pull-down resistor so that, if all inputs/outputs become high impedance, the C line is pulled low (while the $\overline{S}$ line is pulled high): this ensures that $\overline{S}$ and C do not become high at the same time, and so, that the $t_{SHCH}$ requirement is met. The typical value of R is 100 kΩ.

## 4.1 SPI modes

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

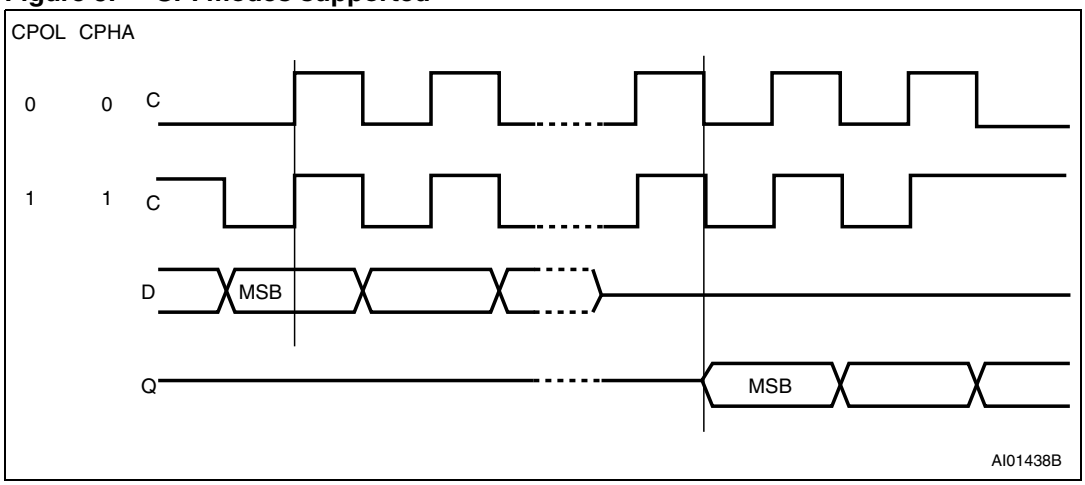
- CPOL=0, CPHA=0
- CPOL=1, CPHA=1

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 5*, 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 5. SPI modes supported**



# 5 Operating features

## 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 Operating conditions in *Section 9: DC and AC parameters*). 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 $V_{CC}$/$V_{SS}$ device pins.

### 5.1.2 Device reset

In order to prevent erroneous instruction decoding and 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 VCC reaches the POR threshold voltage. This threshold is lower than the minimum $V_{CC}$ operating voltage (see Operating conditions in *Section 9: DC and AC parameters*).

At power-up, when $V_{CC}$ passes over the POR threshold, the device is reset and is in the following state:

- in Standby Power mode,
- deselected,
- Status Register values:
  - The Write Enable Latch (WEL) bit is reset to 0.
  - The Write In Progress (WIP) bit is reset to 0.
  - The SRWD, BP1 and BP0 bits remain unchanged (non-volatile bits).

It is important to note that the device must not be accessed until $V_{CC}$ reaches a valid and stable level within the specified [$V_{CC}$(min), $V_{CC}$(max)] range, as defined under Operating conditions in *Section 9: DC and AC parameters*.

### 5.1.3 Power-up conditions

When the power supply is turned on, $V_{CC}$ rises continuously 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 4*).

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

The $V_{CC}$ voltage has to rise continuously from 0 V up to the minimum $V_{CC}$ operating voltage defined under Operating conditions in *Section 9: DC and AC parameters*, and the rise time must not vary faster than 1 V/µs.

### 5.1.4 Power-down

During power-down (continuous decrease of the $V_{CC}$ supply voltage below the minimum $V_{CC}$ operating voltage defined under Operating conditions in *Section 9: DC and AC parameters*), the device must be:

- deselected (Chip Select $\overline{S}$ should be allowed to follow the voltage applied on $V_{CC}$),
- in Standby Power mode (there should not be any internal write cycle in progress).

## 5.2 Active Power and Standby Power modes

When Chip Select ($\overline{S}$) is low, the device is selected, and in the Active Power mode. The device consumes $I_{CC}$.

When Chip Select ($\overline{S}$) is high, the device is deselected. If a Write cycle is not currently in progress, the device then goes into the Standby Power mode, and the device consumption drops to $I_{CC1}$, as specified in DC characteristics (see *Section 9: DC and AC parameters*).

## 5.3 Hold condition

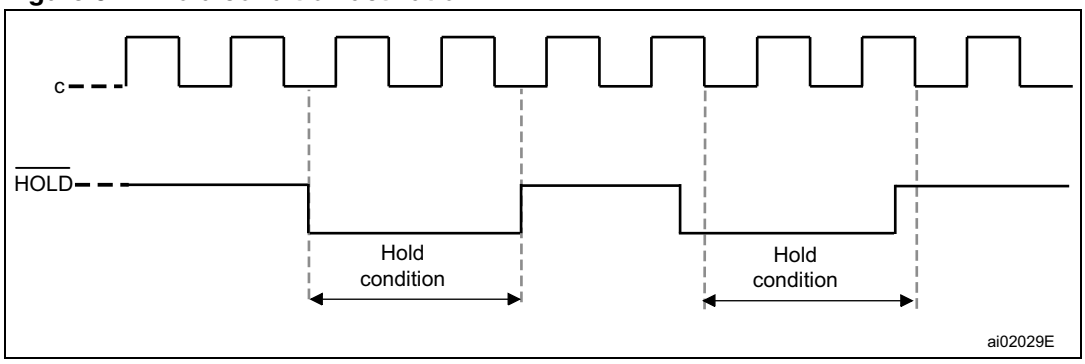
The Hold ($\overline{HOLD}$) signal is used to pause any serial communications with the device without resetting the clocking sequence.

To enter the Hold condition, the device must be selected, with Chip Select ($\overline{S}$) low.

During the Hold condition, the Serial Data Output (Q) is high impedance, and the Serial Data Input (D) and the Serial Clock (C) are Don't Care.

Normally, the device is kept selected for the whole duration of the Hold condition. Deselecting the device while it is in the Hold condition has the effect of resetting the state of the device, and this mechanism can be used if required to reset any processes that had been in progress.[(a)(b)]

**Figure 6. Hold condition activation**




The Hold condition starts when the Hold (HOLD) signal is driven low when Serial Clock (C) is already low (as shown in *Figure 6*).

a. This resets the internal logic, except the WEL and WIP bits of the Status Register.

b. In the specific case where the device has shifted in a Write command (Inst + Address + data bytes, each data byte being exactly 8 bits), deselecting the device also triggers the Write cycle of this decoded command.

The Hold condition ends when the Hold (HOLD) signal is driven high when Serial Clock (C) is already low.

*Figure 6* also shows what happens if the rising and falling edges are not timed to coincide with Serial Clock (C) being low.

## 5.4 Status Register

The Status Register contains a number of status and control bits that can be read or set (as appropriate) by specific instructions. See *Section 6.3: Read Status Register (RDSR)* for a detailed description of the Status Register bits.

## 5.5 Data protection and protocol control

The device features the following data protection mechanisms:

- Before accepting the execution of the Write and Write Status Register instructions, the device checks whether the number of clock pulses comprised in the instructions is a multiple of eight.
- All instructions that modify data must be preceded by a Write Enable (WREN) instruction to set the Write Enable Latch (WEL) bit.
- The Block Protect (BP1, BP0) bits in the Status Register are used to configure part of the memory as read-only.
- The Write Protect ($\overline{W}$) signal is used to protect the Block Protect (BP1, BP0) bits in the Status Register.

For any instruction to be accepted, and executed, Chip Select ($\overline{S}$) must be driven high after the rising edge of Serial Clock (C) for the last bit of the instruction, and before the next rising edge of Serial Clock (C).

Two points should be noted in the previous sentence:

- The “last bit of the instruction” can be the eighth bit of the instruction code, or the eighth bit of a data byte, depending on the instruction (except for Read Status Register (RDSR) and Read (READ) instructions).
- The “next rising edge of Serial Clock (C)” might (or might not) be the next bus transaction for some other device on the SPI bus.

**Table 2. Write-protected block size**

| Status Register bits | | Protected block | Protected array addresses |
|---|---|---|---|
| BP1 | BP0 | | |
| 0 | 0 | none | none |
| 0 | 1 | Upper quarter | 0C00h - 0FFFh |
| 1 | 0 | Upper half | 0800h - 0FFFh |
| 1 | 1 | Whole memory | 0000h - 0FFFh |

# 6 Instructions

Each instruction starts with a single-byte code, as summarized in *Table 4*.

If an invalid instruction is sent (one not contained in *Table 4*), the device automatically deselects itself.

**Table 3. Instruction set**

| Instruction | Description | Instruction format |
|---|---|---|
| WREN | Write Enable | 0000 0110 |
| WRDI | Write Disable | 0000 0100 |
| RDSR | Read Status Register | 0000 0101 |
| WRSR | Write Status Register | 0000 0001 |
| READ | Read from Memory Array | 0000 0011 |
| WRITE | Write to Memory Array | 0000 0010 |

**Table 4. M95320-D instruction set**

| Instruction | Description | Instruction format |
|---|---|---|
| WREN | Write Enable | 0000 0110 |
| WRDI | Write Disable | 0000 0100 |
| RDSR | Read Status Register | 0000 0101 |
| WRSR | Write Status Register | 0000 0001 |
| READ | Read from Memory Array | 0000 0011 |
| WRITE | Write to Memory Array | 0000 0010 |
| Read Identification Page | Reads the page dedicated to identification. | 1000 0011(1) |
| Write Identification Page | Writes the page dedicated to identification. | 1000 0010(1) |
| Read Lock Status | Reads the lock status of the Identification Page. | 1000 0011(2) |
| Lock ID | Locks the Identification page in read-only mode. | 1000 0010(2) |

1. Address bit A10 must be 0, all other address bits are Don't Care.
2. Address bit A10 must be 1, all other address bits are Don't Care.

**Table 5. Address range bits**

| | |
|---|---|
| Address significant bits | A11-A0(1) |

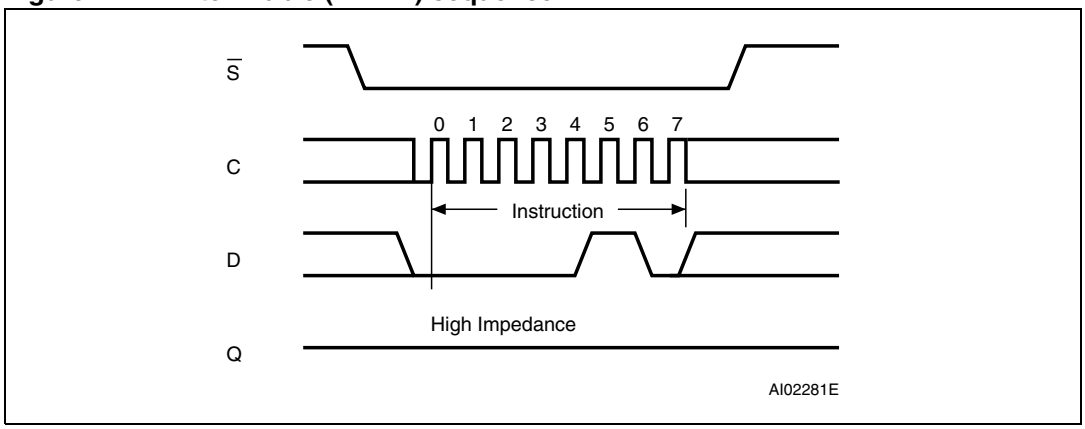
1. Upper MSBs are Don’t Care.

## 6.1 Write Enable (WREN)

The Write Enable Latch (WEL) bit must be set prior to each WRITE and WRSR instruction. The only way to do this is to send a Write Enable instruction to the device.

As shown in *Figure 7*, to send this instruction to the device, Chip Select ($\overline{S}$) is driven low, and the bits of the instruction byte are shifted in, on Serial Data Input (D). The device then enters a wait state. It waits for the device to be deselected, by Chip Select ($\overline{S}$) being driven high.

**Figure 7. Write Enable (WREN) sequence**



## 6.2 Write Disable (WRDI)

One way of resetting the Write Enable Latch (WEL) bit is to send a Write Disable instruction to the device.

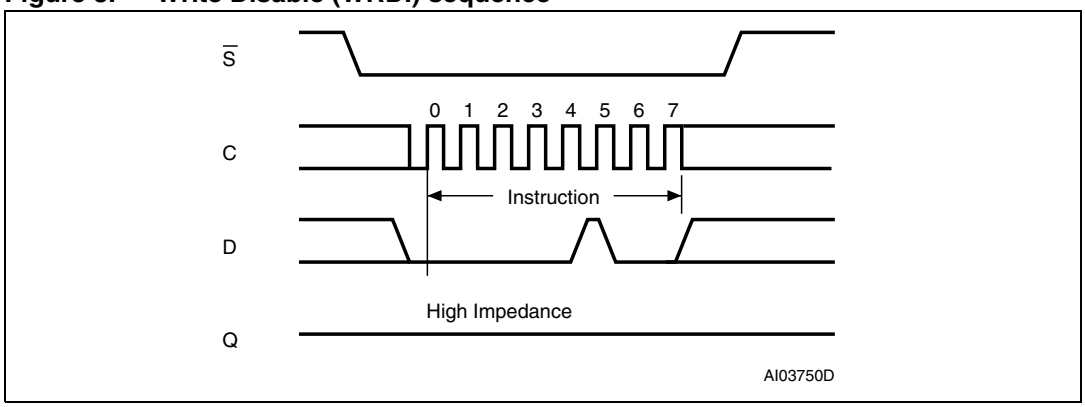
As shown in *Figure 8*, to send this instruction to the device, Chip Select ($\overline{S}$) is driven low, and the bits of the instruction byte are shifted in, on Serial Data Input (D).

The device then enters a wait state. It waits for a the device to be deselected, by Chip Select ($\overline{S}$) being driven high.

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

- Power-up
- WRDI instruction execution
- WRSR instruction completion
- WRITE instruction completion.

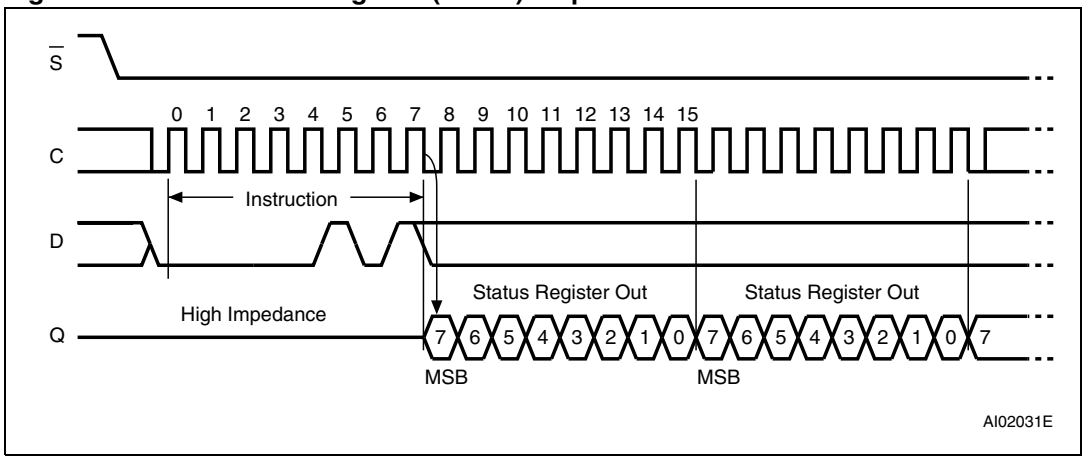
**Figure 8. Write Disable (WRDI) sequence**



## 6.3 Read Status Register (RDSR)

The Read Status Register (RDSR) instruction is used to read the Status Register. The Status Register may be read at any time, even while a Write or Write Status Register cycle is in progress. When one of these cycles is in progress, it is recommended to check the Write In Progress (WIP) bit before sending a new instruction to the device. It is also possible to read the Status Register continuously, as shown in *Figure 9*.

**Figure 9. Read Status Register (RDSR) sequence**




The status and control bits of the Status Register are as follows:

### 6.3.1 WIP bit

The Write In Progress (WIP) bit indicates whether the memory is busy with a Write or Write Status Register cycle. When set to 1, such a cycle is in progress, when reset to 0, no such cycle is in progress.

### 6.3.2 WEL bit

The Write Enable Latch (WEL) bit indicates the status of the internal Write Enable Latch. When set to 1, the internal Write Enable Latch is set. When set to 0, the internal Write Enable Latch is reset, and no Write or Write Status Register instruction is accepted.

The WEL bit is returned to its reset state by the following events:

- Power-up
- Write Disable (WRDI) instruction completion
- Write Status Register (WRSR) instruction completion
- Write (WRITE) instruction completion

### 6.3.3 BP1, BP0 bits

The Block Protect (BP1, BP0) bits are non volatile. They define the size of the area to be software-protected against Write instructions. These bits are written with the Write Status Register (WRSR) instruction. When one or both of the Block Protect (BP1, BP0) bits is set to 1, the relevant memory area (as defined in *Table 2*) becomes protected against Write (WRITE) instructions. The Block Protect (BP1, BP0) bits can be written provided that the Hardware Protected mode has not been set.

### 6.3.4 SRWD bit

The Status Register Write Disable (SRWD) bit is operated in conjunction with the Write Protect ($\overline{W}$) signal. The Status Register Write Disable (SRWD) bit and Write Protect ($\overline{W}$) signal enable the device to be put in the Hardware Protected mode (when the Status Register Write Disable (SRWD) bit is set to 1, and Write Protect ($\overline{W}$) is driven low). In this mode, the non-volatile bits of the Status Register (SRWD, BP1, BP0) become read-only bits and the Write Status Register (WRSR) instruction is no longer accepted for execution.

**Table 6. Status Register format**

| b7 | | | | | | | b0 |
|---|---|---|---|---|---|---|---|
| SRWD | 0 | 0 | 0 | BP1 | BP0 | WEL | WIP |

Status Register Write Protect

Block Protect bits

Write Enable Latch bit

Write In Progress bit

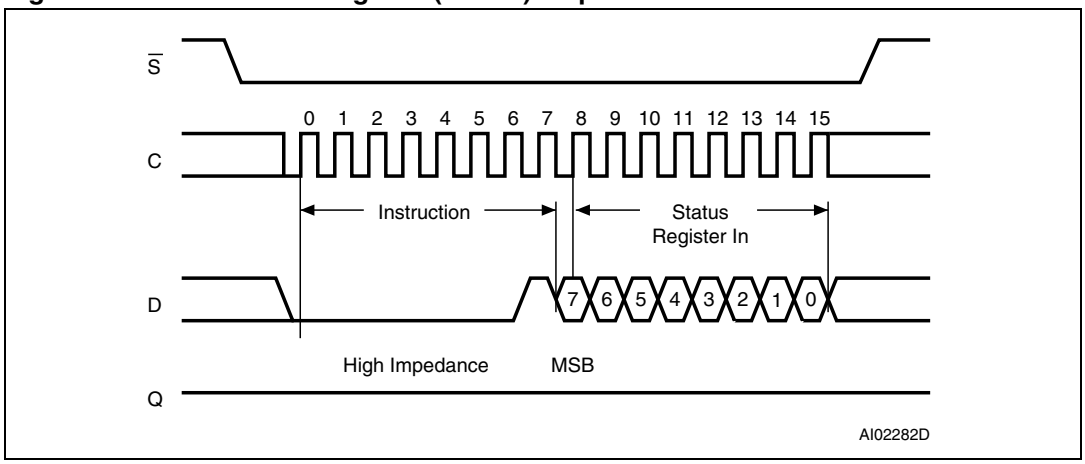
## 6.4 Write Status Register (WRSR)

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

The Write Status Register (WRSR) instruction is entered by driving Chip Select ($\overline{S}$) low, followed by the instruction code, the data byte on Serial Data input (D) and Chip Select ($\overline{S}$) driven high. Chip Select ($\overline{S}$) must be driven high after the rising edge of Serial Clock (C) that latches in the eighth bit of the data byte, and before the next rising edge of Serial Clock (C). Otherwise, the Write Status Register (WRSR) instruction is not executed.

The instruction sequence is shown in *Figure 10*.

**Figure 10. Write Status Register (WRSR) sequence**



Driving the Chip Select ($\overline{S}$) signal high at a byte boundary of the input data triggers the self-timed Write cycle that takes $t_W$ to complete (as specified in AC tables under *Section 9: DC and AC parameters*).

While the Write Status Register cycle is in progress, the Status Register may still be read to check the value of the Write in progress (WIP) bit: the WIP bit is 1 during the self-timed Write cycle $t_W$, and 0 when the Write cycle is complete. The WEL bit (Write Enable Latch) is also reset at the end of the Write cycle $t_W$.

The Write Status Register (WRSR) instruction enables the user to change the values of the BP1, BP0 and SRWD bits:

- The Block Protect (BP1, BP0) bits define the size of the area that is to be treated as read-only, as defined in *Table 2*.
- The SRWD (Status Register Write Disable) bit, in accordance with the signal read on the Write Protect pin ($\overline{W}$), enables the user to set or reset the Write protection mode of the Status Register itself, as defined in *Table 7*. When in Write-protected mode, the Write Status Register (WRSR) instruction is not executed.

The contents of the SRWD and BP1, BP0 bits are updated after the completion of the WRSR instruction, including the $t_W$ Write cycle.

The Write Status Register (WRSR) instruction has no effect on the b6, b5, b4, b1, b0 bits in the Status Register. Bits b6, b5, b4 are always read as 0.

**Table 7. Protection modes**

| $\overline{W}$ signal | SRWD bit | Mode | Write protection of the Status Register | Memory content | |
|---|---|---|---|---|---|
| | | | | Protected area(1) | Unprotected area(1) |
| 1 | 0 | Software-protected (SPM) | Status Register is writable (if the WREN instruction has set the WEL bit). The values in the BP1 and BP0 bits can be changed. | Write-protected | Ready to accept Write instructions |
| 0 | 0 | | | | |
| 1 | 1 | | | | |
| 0 | 1 | Hardware-protected (HPM) | Status Register is Hardware write-protected. The values in the BP1 and BP0 bits cannot be changed. | Write-protected | Ready to accept Write instructions |

1. As defined by the values in the Block Protect (BP1, BP0) bits of the Status Register. See *Table 2*.

The protection features of the device are summarized in *Table 7*.

When the Status Register Write Disable (SRWD) bit in the Status Register is 0 (its initial delivery state), it is possible to write to the Status Register (provided that the WEL bit has previously been set by a WREN instruction), regardless of the logic level applied on the Write Protect ($\overline{W}$) input pin.

When the Status Register Write Disable (SRWD) bit in the Status Register is set to 1, two cases should be considered, depending on the state of the Write Protect ($\overline{W}$) input pin:

- If Write Protect ($\overline{W}$) is driven high, it is possible to write to the Status Register (provided that the WEL bit has previously been set by a WREN instruction).
- If Write Protect ($\overline{W}$) is driven low, it is not possible to write to the Status Register even if the WEL bit has previously been set by a WREN instruction. (Attempts to write to the Status Register are rejected, and are not accepted for execution). As a consequence, all the data bytes in the memory area, which are Software-protected (SPM) by the Block Protect (BP1, BP0) bits in the Status Register, are also hardware-protected against data modification.

Regardless of the order of the two events, the Hardware-protected mode (HPM) can be entered by:

- either setting the SRWD bit after driving the Write Protect ($\overline{W}$) input pin low,
- or driving the Write Protect ($\overline{W}$) input pin low after setting the SRWD bit.

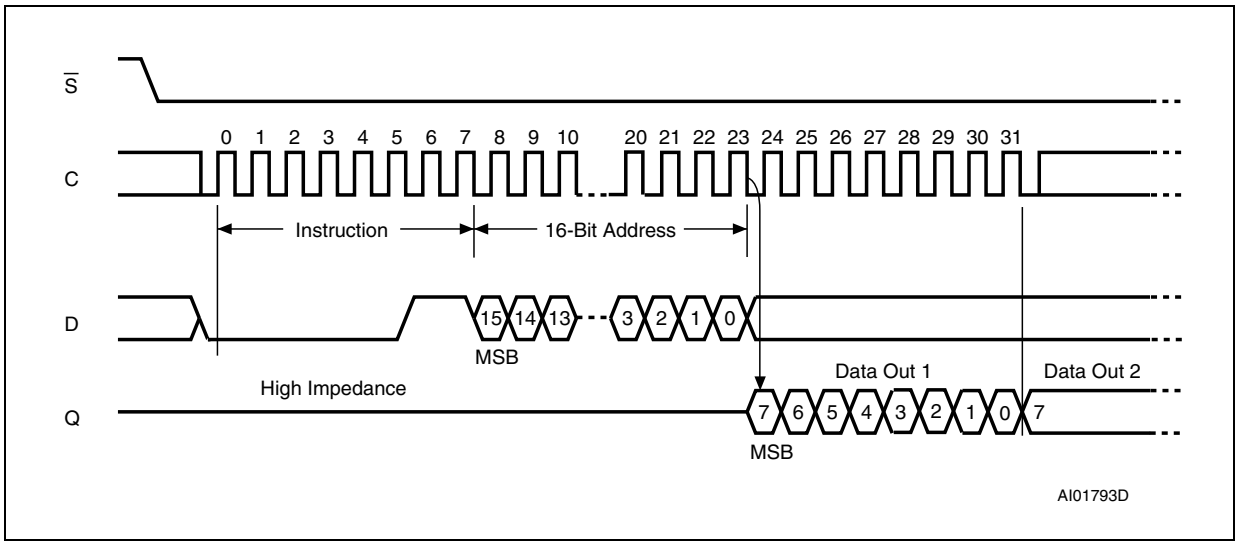
Once the Hardware-protected mode (HPM) has been entered, the only way of exiting it is to pull high the Write Protect ($\overline{W}$) input pin.

If the Write Protect ($\overline{W}$) input pin is permanently tied high, the Hardware-protected mode (HPM) can never be activated, and only the Software-protected mode (SPM), using the Block Protect (BP1, BP0) bits in the Status Register, can be used.

## 6.5 Read from Memory Array (READ)

As shown in *Figure 11*, to send this instruction to the device, Chip Select ($\overline{S}$) is first driven low. The bits of the instruction byte and address bytes 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).

**Figure 11. Read from Memory Array (READ) sequence**




1. Depending on the memory size, as shown in *Table 5*, the most significant address bits are Don't Care.

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

When the highest address is reached, the address counter rolls over to zero, allowing the Read cycle to be continued indefinitely. The whole memory can, therefore, be read with a single READ instruction.

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 during the cycle.

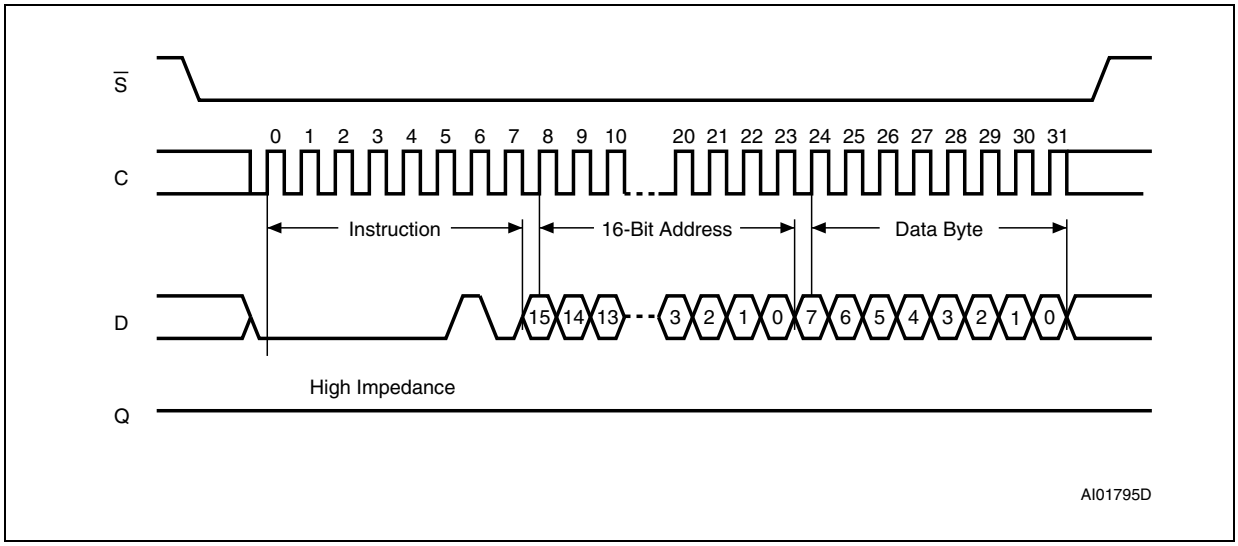
The instruction is not accepted, and is not executed, if a Write cycle is currently in progress.

## 6.6 Write to Memory Array (WRITE)

As shown in *Figure 12*, 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, on Serial Data Input (D).

The instruction is terminated by driving Chip Select ($\overline{S}$) high at a byte boundary of the input data. The self-timed Write cycle, triggered by the Chip Select ($\overline{S}$) rising edge, continues for a period $t_W$ (as specified in AC characteristics in *Section 9: DC and AC parameters*), at the end of which the Write in Progress (WIP) bit is reset to 0.

**Figure 12. Byte Write (WRITE) sequence**




1. Depending on the memory size, as shown in *Table 5*, the most significant address bits are Don't Care.

In the case of *Figure 12*, Chip Select ($\overline{S}$) is driven high after the eighth bit of the data byte has been latched in, indicating that the instruction is being used to write a single byte. However, if Chip Select ($\overline{S}$) continues to be driven low, as shown in *Figure 13*, the next byte of input data is shifted in, so that more than a single byte, starting from the given address towards the end of the same page, can be written in a single internal Write cycle.

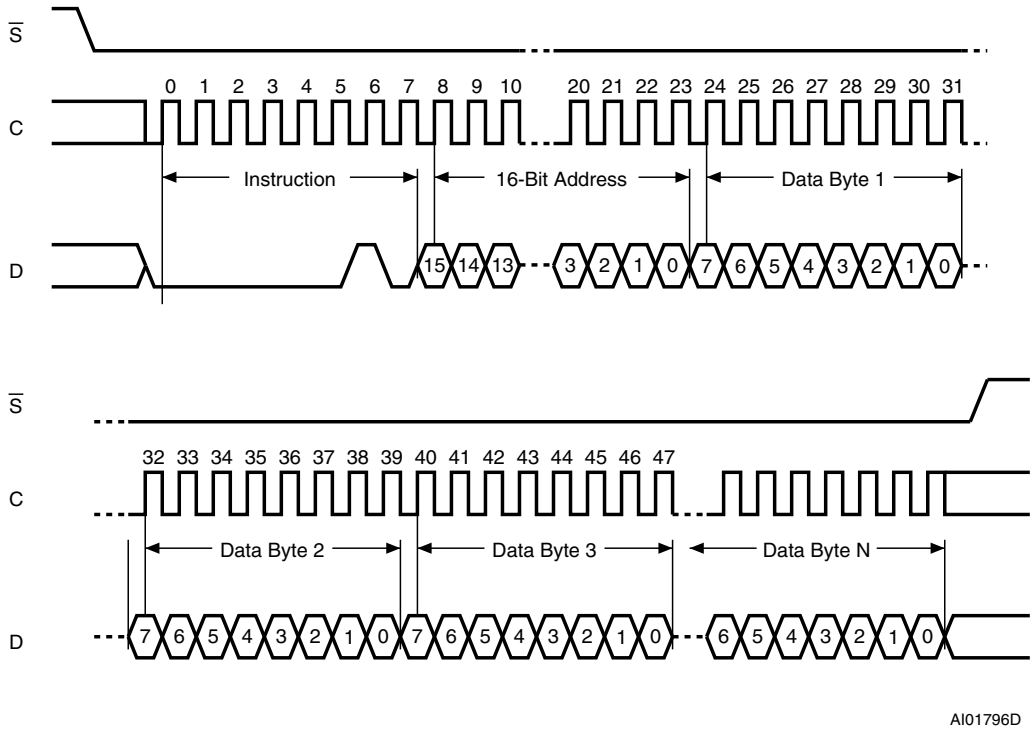
Each time a new data byte is shifted in, the least significant bits of the internal address counter are incremented. If more bytes are sent than will fit up to the end of the page, a condition known as "roll-over" occurs. In case of roll-over, the bytes exceeding the page size are overwritten from location 0 of the same page.

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 driving high Chip Select ($\overline{S}$), at a byte boundary (after the eighth bit, b0, of the last data byte that has been latched in),
- if the addressed page is in the region protected by the Block Protect (BP1 and BP0) bits.

*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 13. Page Write (WRITE) sequence**




1. Depending on the memory size, as shown in *Table 5*, the most significant address bits are Don't Care.

### 6.6.1 Cycling with Error Correction Code (ECC)

M95320-D devices offer an Error Correction Code (ECC) logic. The ECC is an internal logic function which is transparent for the SPI communication protocol.

The ECC logic is implemented on each group of four EEPROM bytes[c]. Inside a group, if a single bit out of the four bytes happens to be erroneous during a Read operation, the ECC detects this bit and replaces it with the correct value. The read reliability is therefore much improved.

Even if the ECC function is performed on groups of four bytes, a single byte can be written/cycled independently. In this case, the ECC function also writes/cycles the three other bytes located in the same group[c]. As a consequence, the maximum cycling budget is defined at group level and the cycling can be distributed over the four bytes of the group: the sum of the cycles seen by byte0, byte1, byte2 and byte3 of the same group must remain below the maximum value defined in *Table 14*.

c. A group of four bytes is located at addresses [4*N, 4*N+1, 4*N+2, 4*N+3], where N is an integer.