AN2014

Application note

How to best use STMicroelectronics serial EEPROMs

Introduction

Electrically erasable and programmable memory (EEPROM) devices are standard products used for the nonvolatile storage of data parameters, with fine granularity.

This document details the internal architecture of Automotive Serial EEPROM, SPD EEPROM, and Standard Serial EEPROM products. It also describes the functionalities of these products, such as storage mechanism, interface circuits, optimal settings of hardware, software, and data management.

Thanks to the guidelines in this document, designers can have a better understanding of the device operation, and significantly improve the reliability of the application.

1 EEPROM cell and memory array architecture

1.1 Floating gate operation

From the user's point of view, the EEPROM is a circuit for storing digital information. To interface with it, a set of standard instructions are used. Behind this simple interface, however, there are several physical processes.

Figure 1 shows the key component of a single EEPROM cell, the floating gate (FLOTOX) transistor. Figure 2 shows how it can be considered to be like any other MOSFET device. As the voltage V_g increases on the control gate electrode, so the does the current flowing through the drain, I_d. For clarity purposes, it is possible to assume that this is a linear relationship.

Figure 1. Structure of an EEPROM floating gate transistor, and circuit symbol

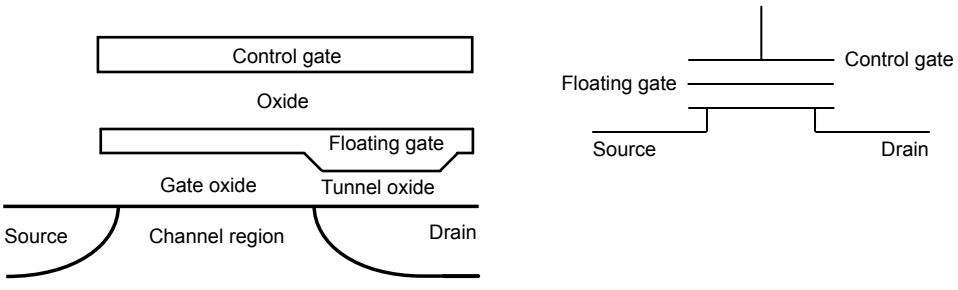


Figure 2. MOSFET-like operation

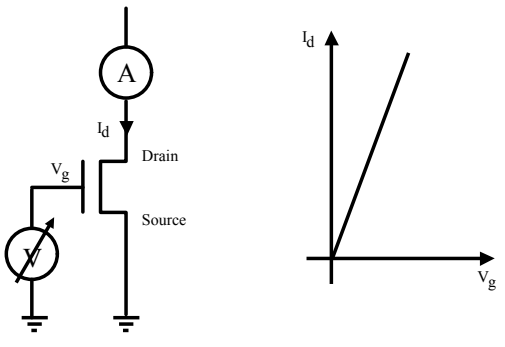
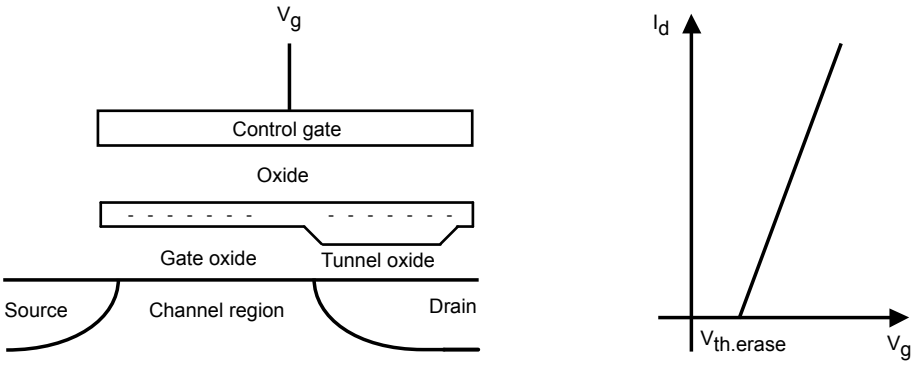


Figure 3 shows what happens if the floating gate can be made more negatively charged, by filling it with extra electrons. This is used for the erased state of the EEPROM cell. Figure 4 shows what happens if the floating gate can be made less negatively charged, by removing electrons. This is used for the written state of the EEPROM cell.

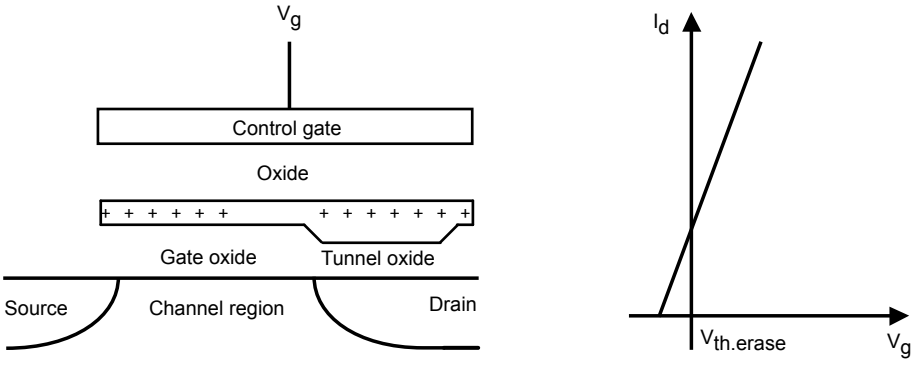
Figure 3. Floating gate with excess of electrons (erased state)



Note: 1. *Control gate threshold value ($V_{th.erase}$) is positive.*

2. *$I_d = f(V_g)$ characteristic shows $I_d = 0$ for $V_g < V_{th.erase}$.*

Figure 4. Floating gate with deficit of electrons (written state)



Note: 1. *Control gate threshold value ($V_{th.write}$) is negative.*

2. *$I_d = f(V_g)$ characteristic shows $I_d = 0$ for $V_g < V_{th.write}$.*

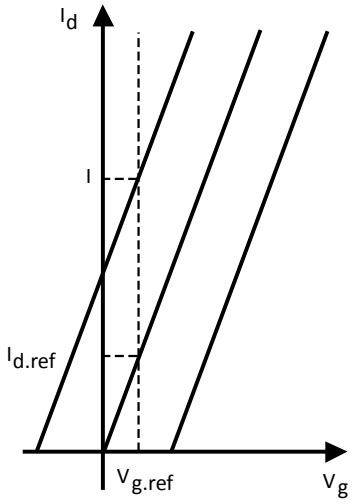
The effect, seen from the channel region, is that V_g is offset by an extra negative or positive amount. Viewed from the outside, the charge on the floating gate has the effect of moving the threshold voltage, V_{th}, at which the linear conduction region begins. In other words, a FLOTOX transistor is a MOS transistor with a variable control gate threshold value, V_{th}.

The floating gate acts as the storage element, and, being completely surrounded by insulating oxide, keeps its charge even when there is no power supply.

1.1.1 Reading the value stored in a memory cell

Figure 5 puts the three curves together. It shows that, for a given control gate voltage, V_g, the current that flows through the drain, I_d, is detectably higher or lower than that of the neutral device, depending on whether there is an excess or a deficit of electrons in the floating gate. This makes possible to read the memory cell and to understand if it is written or erased. At a fixed voltage ($V_{g.ref}$) a written cell draws a current $I_{\mu A} > I_{d.ref}$, while an erased cell does not draw any current (0 µA).

Figure 5. Using the voltage on the control gate to determine the charge on the floating gate



DT74616

A predetermined biasing condition on the control gate and the drain makes it possible to compare the current absorbed by the FLOTOX transistor with that of a reference. With the predetermined biasing condition an erased FLOTOX cell does not sink as much current as the reference (ideally the transistor is off). On the other hand, a written FLOTOX cell sinks a current highr than the reference (the transistor is on). By comparing to a reference current, the device is able to retrieve the stored information as a digital signal on the output pins of the memory device.

1.1.2 Writing a new value to the memory cell

The next question, of course, is how the charge can be changed on the floating gate, given that it is so well insulated by oxide, and keeps its charge even when there is no power supply. The answer is that the tunnel oxide is very thin, and can be used to transfer charge, thanks to voltages significantly higher than those applied during read operations.

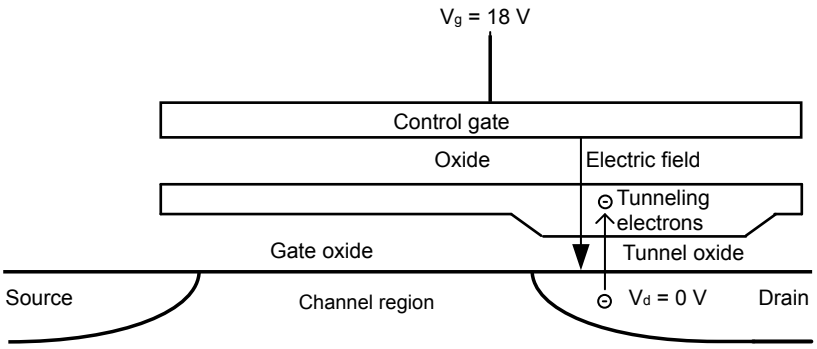
Moving electrons into the floating gate is called erase. After erase, the FLOTOX transistor is in the erased state (see Figure 3). Pulling out negative charges from the floating gate is called program. After program, the FLOTOX transistor is in the written state (see Figure 4). One state is used to represent the logic 0, and the other the logic1 (the choice is manufacturer- and product-dependent).

Both operations use the Fowler-Nordheim tunneling effect: a high electric field (10 MV/cm, or more) is needed to make electrons pass through the thin tunnel oxide. For a tunnel oxide thickness of 100 Å, this voltage must exceed 10 V.

Voltages in the 15 to 18 V range are normally used, to reduce the time taken for the operation. Higher voltages cannot be used, since they would damage the tunnel oxide.

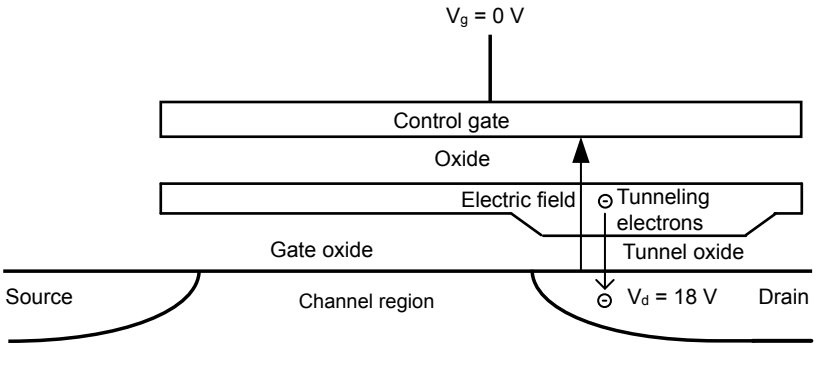
For erase, the cell control gate is made positive, and the source-drain regions are grounded (as shown in Figure 6). The electric field makes electrons move from the substrate towards the floating gate, increasing the threshold voltage (as shown in Figure 3).

Figure 6. Erase operation



For write, the control gate is grounded and the source-drain region is made positive (as shown in Figure 7). The electric field is the opposite of that for erase, and so electrons move out from the floating gate, decreasing the threshold voltage (as shown in Figure 4).

Figure 7. Write operation



Typically, erase/write cycles require a voltage of about 15 to 18 V for approximately 4 or 5 ms.

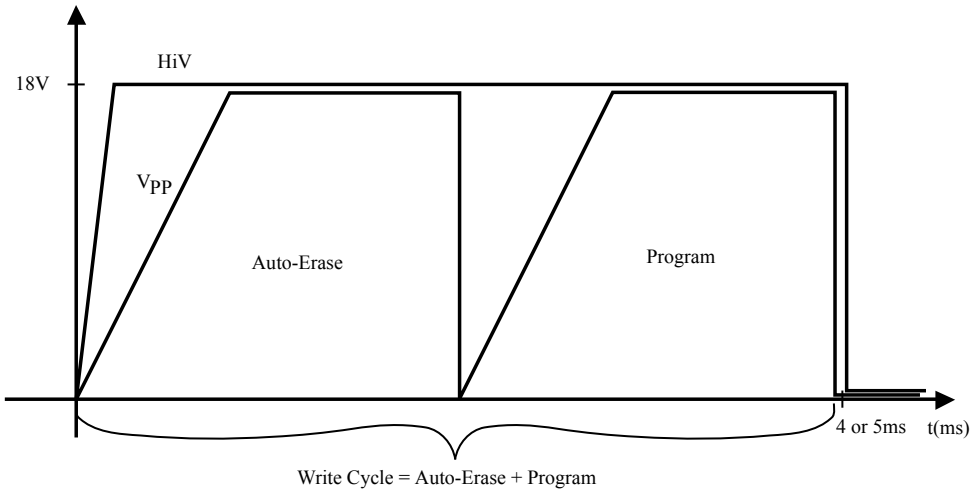
As EEPROM devices use a single supply voltage, the high voltage must be generated and managed internally. A set of analog circuits is available to generate and control the high voltage from the single external power supply:

- Voltage and current references to control oscillators and timings.
- A regulated charge pump that generates a stable 15 to 18 V voltage, HiV, from the single external power supply.
- A ramp generator that, from the stable HiV voltage, makes the specific waveform (shown in Figure 8) to apply to the cells.

V_{PP} is the high voltage directly applied to the FLOTOX cell, as described earlier. The precise shape of the waveform is critical, and has a direct effect on the reliability and endurance of the memory cells. The slope, plate time and maximum level are tightly controlled parameters.

Writing new data in an EEPROM array triggers an auto-erase of all the addressed bytes, resets them all to the erased state, and then selectively programs the bits to set to the written state.

Figure 8. V_{PP} signal applied to EEPROM cells (HiV is the output of the charge pump)



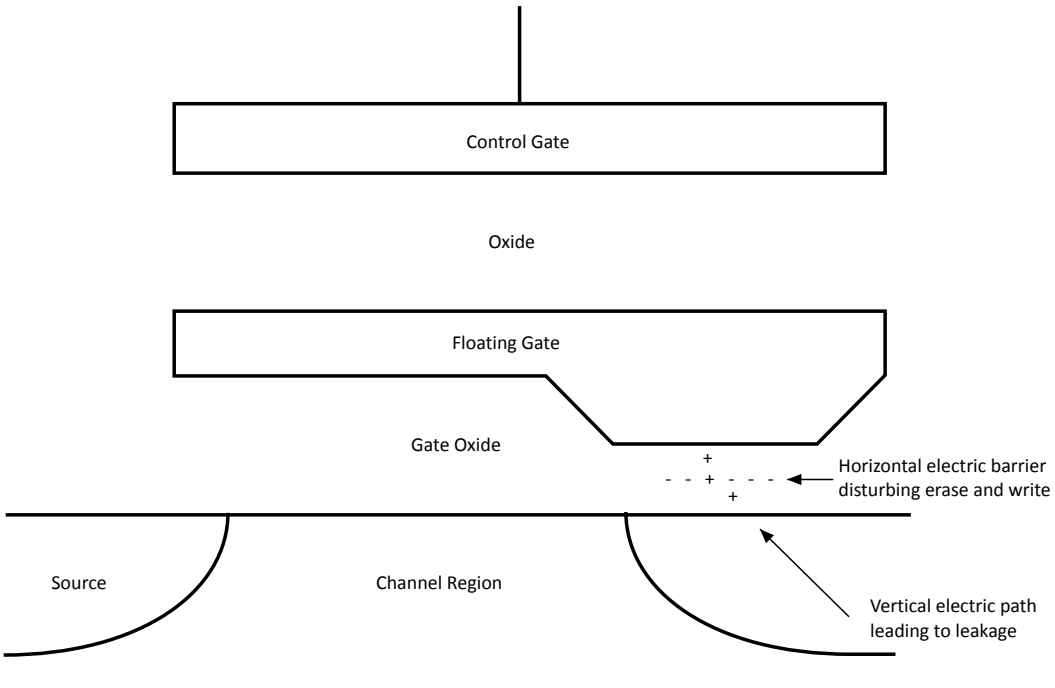
To summarize: Binary information is coded by means of a FLOTOX transistor. The floating gate is a reservoir filled with negative electric charges that modify its electrical characteristics. The electric charges can migrate into or out of the reservoir by applying a high voltage to a thin tunnel oxide. The binary information is read by comparing the cell (FLOTOX transistor) current to a reference.

1.1.3 Cycling limit of EEPROM cells

When a cell is cycled (repeatedly erased and programmed) two common phenomena occur and are amplified during the memory cell lifetime. When tunneling, negative charges can either be trapped in some imperfection of the oxide or damage the tunnel oxide:

1. **Charge trapping**
   The accumulation of negative charges in the thin tunnel oxide creates an electric barrier in the tunnel oxide. The high voltage needed for the tunneling effect becomes even higher: programming high voltages are no more able to move enough charges to program the cell properly. The erased and written states become undifferentiated.
2. **Stress on oxide**
   When the tunnel oxide deteriorates, a positive charge path may appear, that facilitates undesirable leakage through the tunnel oxide. The floating gate is no more 100% insulated, and loses its charges, and so the data retention time drops drastically.

Figure 9. **Accumulation of negative or positive charges in the tunnel oxide**



Charge trapping and oxide damage are accelerated at high temperatures. They are directly involved in cell cycling and endurance limitations.

Permanent digital information storage has to cope with physical phenomena and analog nonlinear behaviors that have natural limits and are sensitive to wear-out and improper use conditions.

1.2 Architecture of serial EEPROM arrays

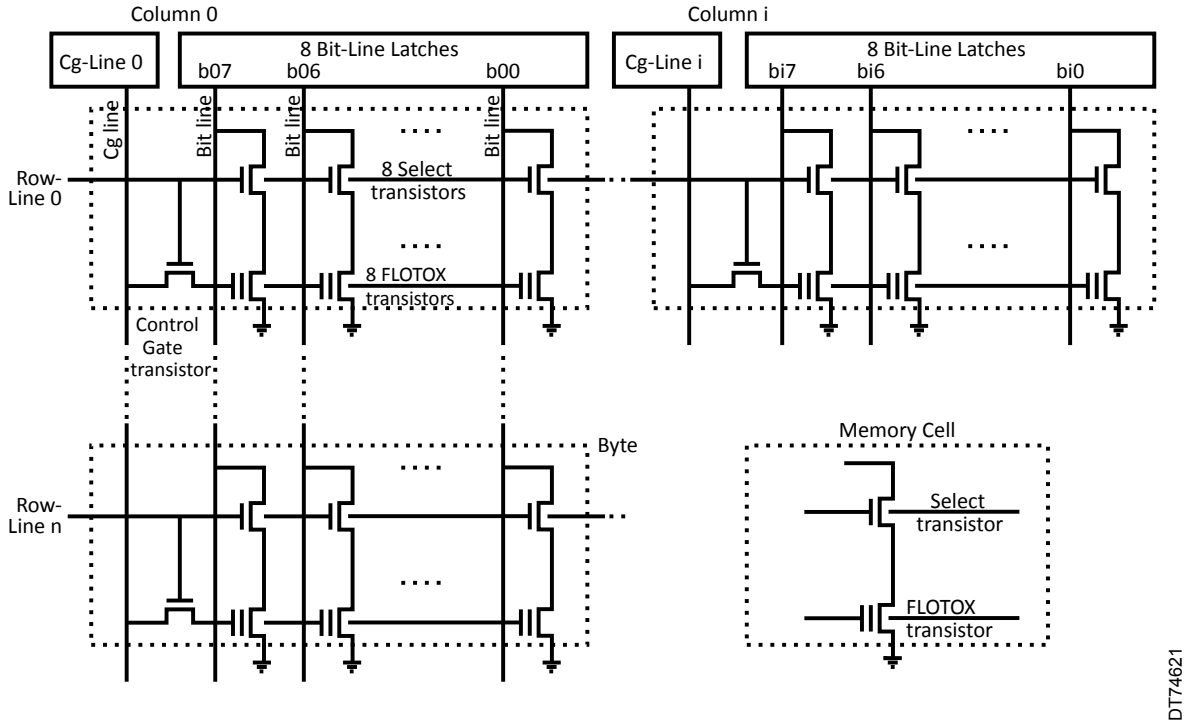
In the previous section, the EEPROM functionality was considered at the single bit level. We zoom out of the memory cell to the full array, to give an overview of the architecture of an EEPROM device.

1.2.1 Memory array architecture

An EEPROM device is made of an array of memory cells whose organization allows byte granularity, the automatic erasing of the addressed bytes (erased state), and the programming of only those bits that are to be changed to '1' (written state). The array (as shown in Figure 10) is organized as follows:

- Each memory cell consists of one select transistor in series with a FLOTOX transistor and each byte is made up of eight memory cells and a control gate transistor with a drain that is common to the control gates of all eight FLOTOX transistors.
- Rows (in the horizontal direction) are made up of 16 bytes (or more, depending on the memory size (the number of bytes within each row being a function of the array size). For each row, all select transistors and all control gate transistors are connected to the row line.
- Columns are grouped by eight bit-lines plus one Cg-line. This is then repeated as many times as the number of bytes in a row.
- A bit-line is common to all the drains of the select transistors of each memory cell located in the column. A Cg-line is common to all the sources of the control gate transistors of the column.

Figure 10. Architecture of the memory array (showing the grouping in bytes)

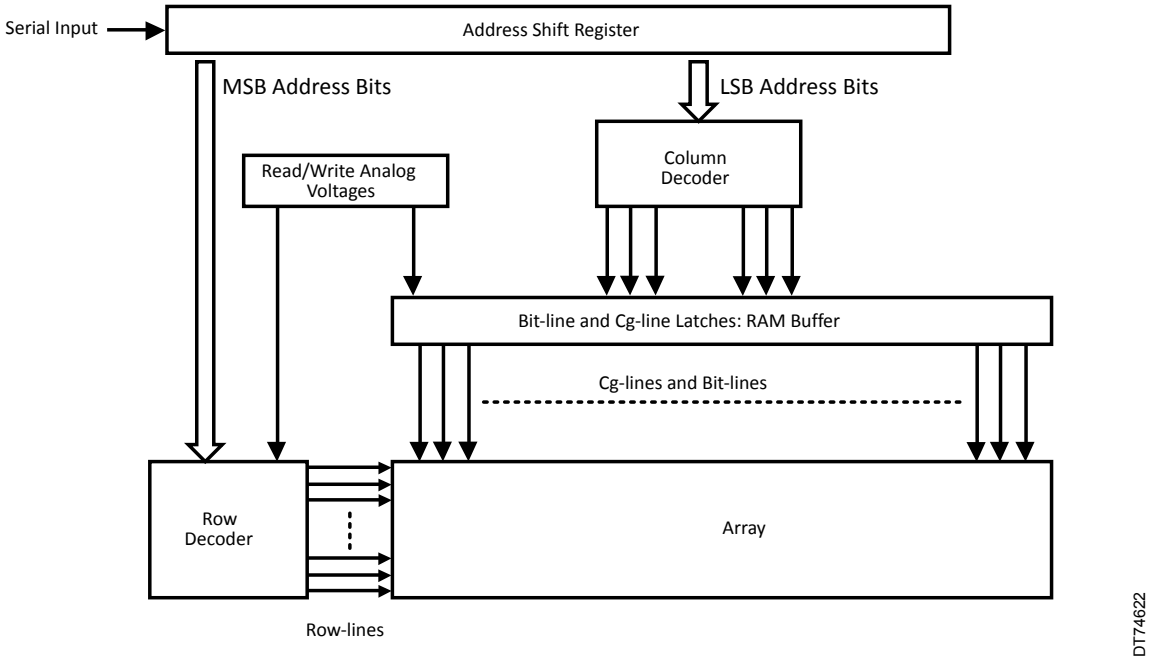


Note: Figure 10 corresponds to a memory array without ECC (error correction code) logic (see Section 5.5 and AN2440 Embedded ECC in F8H process automotive EEPROM: device architecture and related application guidelines for more details).

1.2.2 Decoding architecture

To address a single byte in a full array, decoding circuits are necessary. One logical address is associated with one byte location. The address bits are inserted serially into a shift register. Then, with parallel output, the decoding structures receive all of the bits at the same time, to perform the decoding and addressing. The row decoder decodes and brings correct biasing to a single row line. As one or more bytes of the same row can be programmed at the same time, the column decoder decodes one or more column(s), and a RAM buffer memorizes the data to write, and enables the right path for Cg-line and Bit-line biasing.

Figure 11. **Decoding block diagram**



1.2.3 Intrinsic electrical stress induced by programming

Whatever the data value to be programmed and whether the request is made by byte or page, all high-voltage circuits(1.) are stressed by HiV (a high voltage ranging between 15 and 18 V). In particular, the internal nodes of the charge pump can see voltages equal to HiV + V_{CC} (as much as 23 V). All circuits that receive and carry HiV (ramp generator, regulation, decoding, latches) are submitted to higher stress than active low voltage transistors. The overall time during which the high voltage circuits are active is relatively short compared to the product lifetime (10 ms x 1 Mcycles = 10000 seconds, or less than 3 hours).

A standard ST EEPROM device includes a few hundred high voltage transistors, for low memory density products (1Kbit). This number can rise to a few thousand for high memory density products (4 Mbit).

Consider, by way of example, the stress induced on the array elements when programming one single byte in a 1Kbit EEPROM, organized as 128 x8 bit. The memory array is composed of 8 pages (or rows) of 16 bytes (or columns).

Erase cycle: the complete row (page) that contains the addressed byte receives the V_{PP} signal, on the selected row-line, as does the complete column, on the selected Cg-line:

- Control gates of all the select transistors in the given row: 1 row x 16 bytes x 8 bits = 128
- Control cates of all the control gate transistors in the given row: 1 row x 16 bytes = 16
- Drains of all the control gate transistors that are connected to the given Cg-line: 1 column x 8 rows = 8

The Bit-lines of the addressed bytes are floating.

Note: 1. The high voltage is required to erase and program an EEPROM cell.

Note: ***Write cycle:*** *the complete row (page) that contains the addressed byte receives the V_{PP} signal, on the selected row-line:*

- *Control gates of all the select transistors in the given row: 1 row x 16 bytes x 8 bits = 128*
- *Control gates of all the control gate transistors in the given row: 1 row x 16 bytes = 16*
- *The Cg-line of the addressed byte is held at ground voltage*
- *The Bit-lines are left floating or receive V_{PP} depending on data to be written. The worst case is when FFh is to be written, and all Bit-lines receive the V_{PP} signal*
- *Drains of all the Select transistors sharing the same 8 Bit-lines: 1 column x 8 rows x 8 bits = 64*

This example shows how one single byte, being erased or programmed, incurs a lot of high voltage stress on elements that share the same row, column and bit-line as the one addressed. For a 1 Kbit EEPROM, programming one single byte to FFh induces stress on 128 select transistors and 24 control gate transistors during auto-erase, and 192 select transistors and 16 control gate transistors during the write cycle, even though only 17 transistors (8 select transistors, 8 FLOTOX transistors, 1 MOS transistor) were really being addressed for the data change.

The bigger the memory array, the larger the number of additional transistors that are involved. This is why when high cycling performance is required, it is recommended to group N contiguous bytes inside one page and use the write page mode so that the overall number of write cycles remains to its lowest value.

2 Choosing a suitable EEPROM for your application

ST EEPROM products offer a very high quality level, as they are produced with mature processes, there is an efficient testing coverage on each single production part. Nevertheless, the reliability on EEPROM products is also closely linked to the way they are controlled in the end application. The aim of the following chapters is to provide a set of recommendations to significantly improve the reliability and robustness in the end application.

In the case of automotive applications, it is recommended to use products that are classified as automotive grade, that is, devices referenced as MXXxxx-125 (previous generation) and MXXxxx-A125/A145 (new generation with improved features). These devices are designed to satisfy the most stringent quality requirements and are tested with STMicroelectronics' High Reliability Certified Flow.

2.1 Choosing a memory type suited to the task to be performed

EEPROM devices are particularly suited to the tasks of code traceability and parameter storage. The Serial protocol offers the best compromise of performance versus cost where the access time is not critical.

2.2 Choosing the appropriate memory interface

ST offers serial access EEPROMs based on three main protocols: I²C, SPI, and Microwire (see Table 1). Noise immunity, ESD, latchup, and cycling endurance are basic features of each device, independently from the protocol.

The choice of the most appropriate product depends mainly on the hardware resources of the controller and on the architecture built around it.

- The I²C bus offers a 2-wire protocol working at a maximum clock rate of 1 MHz and so is preferred when the hardware resources are limited and the data rate is not a constraint at all. The multiple-receiver configuration requires no extra hardware and is managed by software.
- The SPI and Microwire buses are 4-wire protocols allowing higher communication speed (determined by each manufacturer design and technology). The number of receivers is unlimited, but N receivers require N additional controller I/Os for each chip select line. Both buses can be reduced to only three wires, provided that D and Q pins are tied together to a bidirectional I/O.

Data write protection is different for each protocol family and is also a key factor when selecting the memory interface. I²C (on CMOSF9V), SPI, and Microwire products provide both hardware and software protection. Refer to Section 5.3: Write protection.

Table 1. Serial bus protocols

Interface	I²C	SPI	Microwire
Products	M24xxx, 1-Kbit to 4-Mbit	M95xxx, 1-Kbit to 4-Mbit	M93Cxxx, 1- to 16-Kbit M93Sxxx, 1- to 4-Kbit
Interface	Two wires: single I/O line, clock	Four wires: data in, data out, clock, and CS	
Clock rate (max)	Up to 1 Mbit/s	Up to 20 Mbit/s	Up to 2 Mbit/s
Data management	By byte - Page: 16 to 512 bytes		By byte or by word Page: four words
Specific features	• Hardware write control • Software write control for four blocks • Software addressing on 5-pin pinout • Up to eight devices cascadable on the same bus	• Hold mode (input pin) • Write control for four blocks	• Block write protection defined by software for M93Sxxx products

2.3 Choosing an appropriate supply voltage and temperature range

These are essential parameters that will define the device reliability when operating in the application. The V_{CC} values and the temperature values of the application must always stay within the limits defined in ST datasheets.

3 Recommendations to improve reliability

3.1 Electrostatic discharges (ESD)

ESD damage can happen any time during the product lifetime, from the moment it is delivered to the final field service operation. ESD damage can be destructive or latent. In the first case, a simple functional test can screen faulty devices; in the second case, the part is partially damaged and may be able to operate correctly, but its operating life may be drastically reduced, causing the device to fail prematurely in field service.

3.1.1 What is ESD?

Static electricity results from the contact and separation of two bodies, which creates an unbalance in the number of electrons at the surface of the bodies. Practically, the bodies become charged to a specific electrical potential that depends on the material from which they are made (see Table 2). An electrostatic discharge is defined as the transfer of charges between two bodies at different electrical potentials. It is instantaneous (a few nanoseconds) and thus induces high energy peaks which are very difficult to control and predict.

3.1.2 How to prevent ESD?

An ESD can be managed if the discharge is driven through a known and controlled path on the silicon die. Specific design rules and techniques can be used by designers to better protect against ESDs, such as Faraday shields, perimeter ground lines or ground planes.

In a production line, the part handling until the assembly line has to be carefully ESD-protected.

Table 2. ESD generation

ESD generation means	Static voltage levels
Walking across a carpet	1,500 V to 35,000 V
Worker on a bench	100 V to 6,000 V
Chair with urethane foam	1,500 V to 18,000 V

Note: The charge unbalance depends on many factors such as the contact area, separation speed and relative humidity.

3.1.3 ESD protection

ST EEPROM devices offer a specific protection circuit against human body model ESDs, in accordance with AEC-Q100-002.

During write operations, the EEPROM is much more sensitive to ESDs because of the architecture of its internal high voltage generator. Applications exposed to ESD should avoid writing data in the EEPROM when an ESD is more likely to occur.

3.2 Electrical overstress and latchup

Electrical overstress (EOS) and latchup are also damaging stresses that are either immediately destructive, or may create latent defects leading to premature failure.

3.2.1 What are EOS and latchup?

In comparison with ESDs, EOS and latchup are lower-intensity events that last much longer (sometimes more than a few seconds). That is why the energy induced by an EOS is higher than the ESD energy. EOS and latchup induce current injections inside the EEPROM when an overvoltage stress is applied on one or more package pins. Latchup occurs when a charge injection triggers the I/O parasitic thyristors (also called SCR) thus generating a very high current between V_{DD} and V_{SS}. This phenomenon lasts until the V_{CC} power supply is turned off.

3.2.2 How to prevent EOS and latchup events

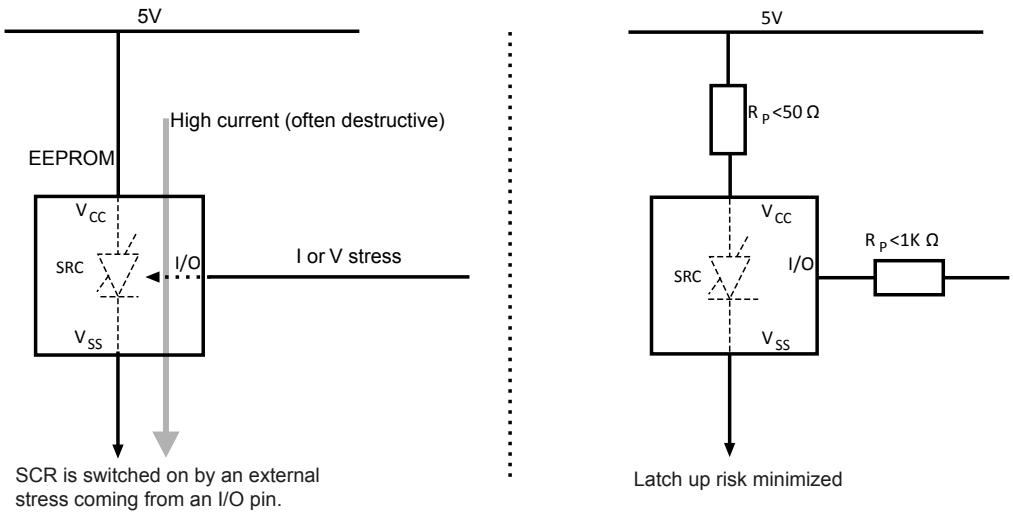
Typically power supply cycling leads to EOS situations. During the power-up and power-down phases, the EEPROM I/Os interfaced with other ICs may temporary see voltages greater than V_{CC} or lower than V_{SS}. When outside absolute maximum ratings, these biasing conditions may lead to positive and negative current injections, respectively. This kind of stress cannot always be completely prevented but it can be minimized. The switching sequence of the different interfaced ICs must be carefully determined, and if necessary protection resistor (< 1 kΩ) can be placed on critical pins or sometimes directly on V_{CC} pin (< 50 Ω) to limit eventual latchup current. Refer to Section 4: Hardware considerations for more details.

Overshoots and undershoots may occur on external device pins when the application is running. They can be generated by radiations, power supply disturbances or even some ICs. The very first protection is provided by the semiconductor manufacturer (ST) which offers the best possible robustness against EOS and latchup. If extra protection is needed, the application designer can add small value resistors (< 1 kΩ) in series on all interfaced lines and (< 50 Ω) in series on V_{CC} line so that it can be compatible with the communication speed constraints and power supply range. Refer to the Section 4: Hardware considerations for more details.

Manufacturing and handling devices are also sources of EOS: all voltage levels applied to the device must be checked accurately and regularly. In addition all equipment should be constantly calibrated.

During write operations, an EEPROM device is more sensitive to overvoltages on its power supply pin because the internal high voltage generator is directly fed by the voltage applied to the power supply pin.

Figure 12. Latchup mechanism and protection

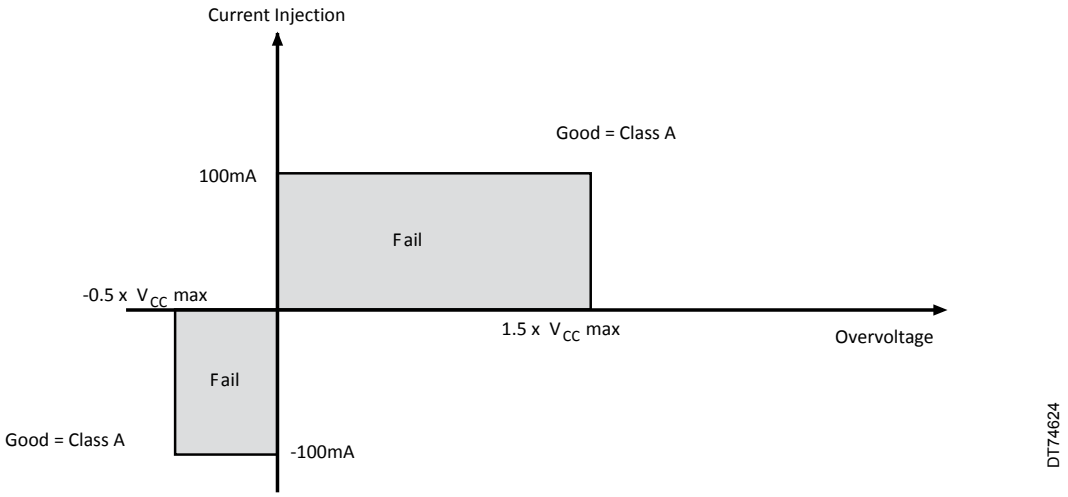


Note: 1. Protection is only recommended if latchup risk is identified.

3.2.3 Latchup protection

During the qualification process, samples from three different lots are tested for voltage overshoots (positive and negative injections). Figure 13 shows the levels of stress applied to the tested devices.

Figure 13. Latchup test conditions



Note: The device does not latch up within the gray areas.

3.3 Power supply considerations

The power supply also has a major impact on the operating reliability of the EEPROM device.

3.3.1 Power-up and power-on-reset sequence

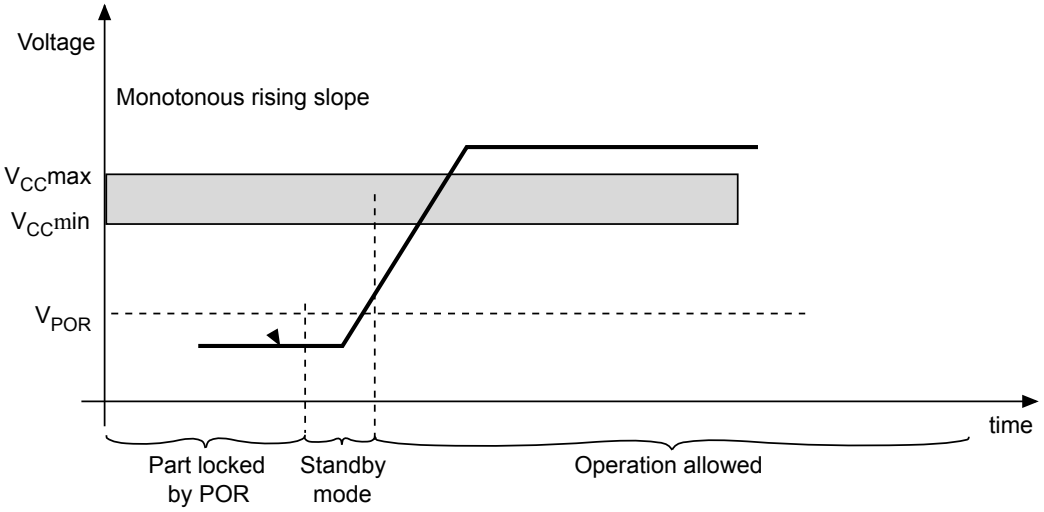
During power-up, ensure that the V_{CC} supply voltage rises monotonously from V_{SS} to the final stabilized V_{CC} value to correctly reset the device.

The POR (power-on-reset) circuit is activated first, as it locks the part before the internal logic is able to run: at power-up, the device does not respond to any instruction until V_{CC} has reached the power-on-reset threshold voltage, V_{POR}. When V_{CC} exceeds V_{POR}, the device is reset, in standby power mode. Do not send instructions until the power supply has reached a stabilized DC value comprised between V_{CC}min and V_{CC}max (see Figure 14).

Before a controlled power-down sequence (continuous decrease in V_{CC}), the device must be placed in the standby power mode.

If the power supply drops, it is recommended to carry out a safe power-down sequence by pulling V_{CC} to 0 V (or at least below V_{POR}), and then to perform a safe power-up sequence, as described previously. This secures the device re-initialization.

Figure 14. **Power-up**



Note: Power-up is safe with a monotonous rising slope.

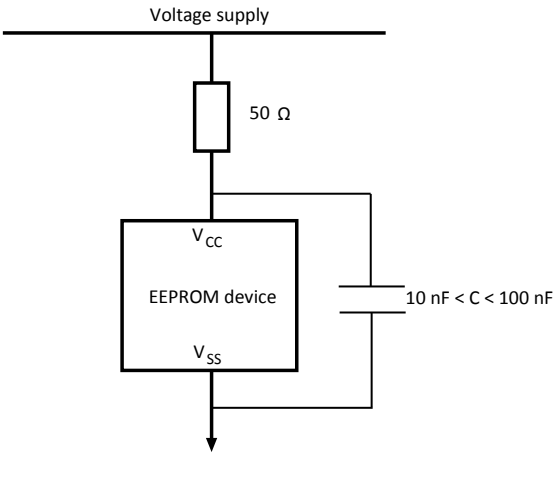
Table 3. **Typical POR threshold values**

Bus protocol	I²C	SPI	Microwire
Device voltage range	All	All	All
POR threshold limits over the whole temperature range	1 V ± 0.5 V	1 V ± 0.5 V	1 V ± 0.5 V

3.3.2 Stabilized power supply voltage

The value of the stabilized power supply voltage, including potential variations, must always stay within the operating V_{CC} range specified in the datasheet, where device operation is reliable and guaranteed. Application designers deal mostly with transient peak currents and voltages. Transient peak current generated by EEPROM during read, write and output buffer transitions induces transient voltage disturb on power supply lines. Therefore, the use of high-frequency, low-inductance capacitors located as close as possible to the device V_{CC} and V_{SS} pins are also recommended. Some applications with a limited power supply driving capability may require a small value resistor (< 50 Ω) connected to the EEPROM V_{CC} pin, so that the peaks of current sunk by the EEPROM during the write cycle (10 mA typical during a few nanoseconds, with a duty cycle of less than 1/100) are mostly supplied by the decoupling capacitor and so, induce less disturbance on the voltage supply line of the application. See Figure 15.

Figure 15. **Local EEPROM supply filtering**



Note: 1. *Capacitor should be placed as close as possible to V_{CC} and V_{SS} pins to avoid parasitic inductive effects.*
2. *Resistor must never be placed between the decoupling capacitor and the V_{CC} pin of the EEPROM*

3.3.3 Absolute maximum ratings

Absolute maximum ratings are not operating values for the device. They provide an additional security margin for temporary operating deviations. Temporary operation within this margin will not cause the device to be damaged. However, the normal operation of the EEPROM is neither guaranteed nor reliable under absolute maximum rating conditions that are above or below normal operating conditions.

4 Hardware considerations

EEPROM connections are essential for an application's robustness. During power up, power down and application reset, input signals must be fully controlled to avoid hazardous behavior or random operation. For each product family a good hard-wiring design can protect the parts from uncontrolled behavior.

4.1 I²C products (M24xxx devices)

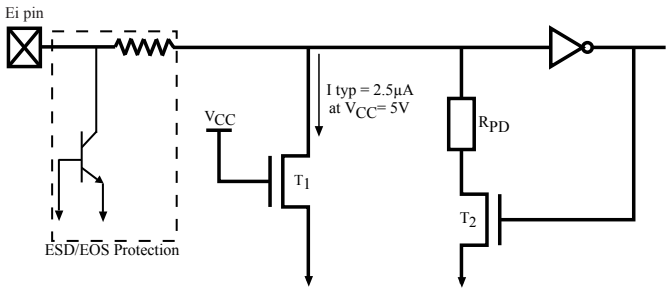
4.1.1 Chip enable (E0, E1, E2)

The chip enable (E0, E1, E2) inputs have an internal pull-down resistor. It is possible to have the three inputs unconnected (chip enable address is decoded as 000). However, this configuration should be avoided since inputs are still prone to antenna-like pick-up or other cross coupling effects.

To achieve a robust design, Ei inputs must not be driven dynamically, but must be directly tied to V_{CC} or V_{SS}.

Note: Not applicable for I²C products manufactured in CMOSF9V, as there are no longer physical chip enable inputs.

Figure 16. Chip enable inputs E0, E1, E2



The input leakage current on the Ei pins depends on the input voltage value (Table 4).

Table 4. Connecting the Ei inputs of I²C products

I²C products	1 to 16 Kbits	32 to 128 Kbits	256 Kbits to 2 Mbits
Internal E0, E1, E2 pull-down resistance (R_{PD}) typical value (devices identified with process letters K and T)	50 kΩ		
Internal E0, E1, E2 pull-down resistance (R_{PD}) typical value (all other devices)	30 kΩ	100 kΩ	50 kΩ
Input leakage I_{li} of Ei pins	I_{li} = 0 for $V_{Ei} > V_{IH}$ $I_{li} = V_{Ei} / R_{PD}$ for $0 < V_{Ei} < V_{IH}$		
Recommended connection of Ei pins	Direct connection to V_{CC} or V_{SS}		

4.1.2 Serial data (SDA)

The serial data (SDA) input/output is a bidirectional signal. The SDA pin is internally connected to a CMOS Schmitt trigger input buffer and an open drain output transistor (see Figure 17). The SDA line must be pulled to the V_{CC} of the device through a pull-up resistor (R_{BUS}), whose value depends upon the capacitive load of SDA line, the controller, and the buffer characteristics. More in detail:

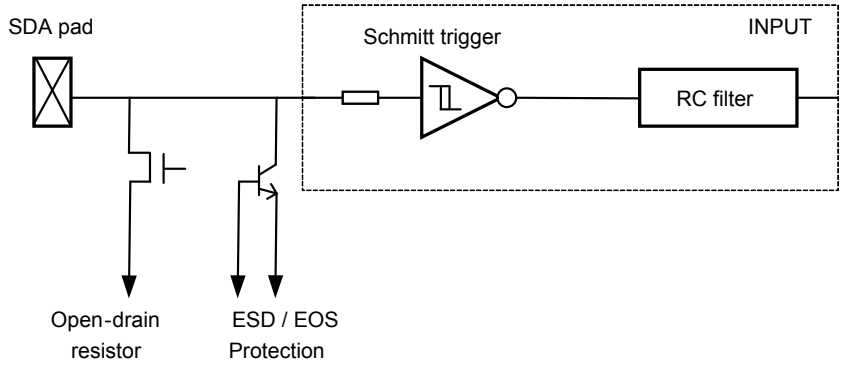
- For the maximum value, R_{BUS} defines the SDA line rise time t_R (defined in the M24xxx datasheets), depending on the C_{BUS} parasitic value (1)
- For the minumum value, use the maximum value of {(V_{CC} - $V_{IL\ EEPROM}$)/$I_{OL\ Controller}$ or (V_{CC} - $V_{IL\ Controller}$)/$I_{OL\ EEPROM}$}

1. *Refer to the maximum R_{BUS} value versus bus parasitic capacitance (C_{BUS}) for an I²C bus at the maximum frequency figure in the I²C datasheet.*

Note: The smaller R_{BUS}, the faster the clock frequency. The higher R_{BUS}, the lower the operating current, the slower the transitions, and the lower the electromagnetic interference.

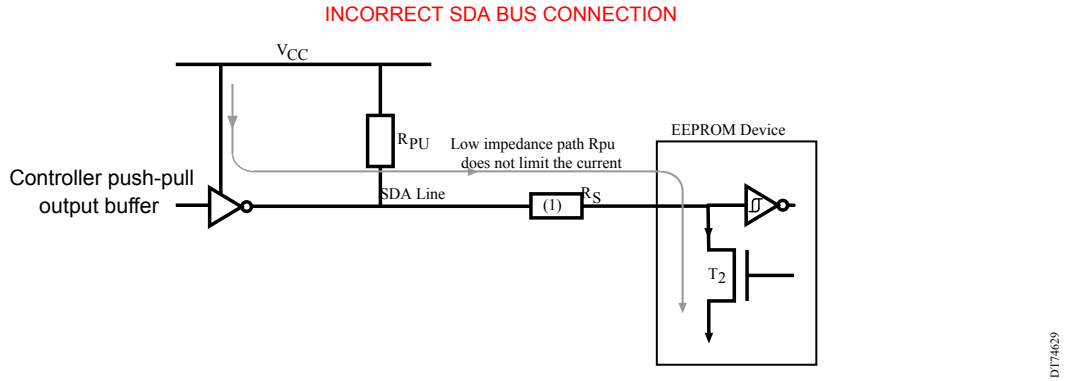
The input pin leakage is negligible (typically a few nA). The input schematic (including the protection circuit) does not offer any open path to the V_{SS} or V_{CC}, therefore the SDA level can be set before the EEPROM power up, and remains high even after power down, with no risk of leakage or EOS.

Figure 17. Serial data input/output SDA



On the controller side, the SDA must be an open drain output. The controller SDA output must not be a push-pull buffer, as this would lead to a conflict when the controller drives a high SDA line and when the EEPROM drives low the SDA line (this induces a high current between the power supply of the controller and the ground of the EEPROM device). This event occurs each time the I²C device acknowledges an instruction from the controller (see Figure 18).

Figure 18. SDA bus conflict with push-pull buffers (NOT RECOMMENDED)



Note: $R_S > V_{CC}/I_O$ with I_O = min (controller I_{OH}, EEPROM I_{OL}). Without the R_S resistor the current is limited by the controller buffer and transistor T2 producing overstress at both controller and EEPROM side.

4.1.3 Serial clock (SCL)

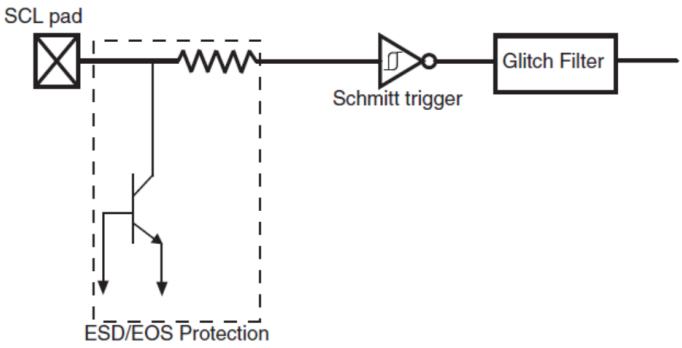
The M24xxx serial clock (SCL) input is a CMOS Schmitt trigger. In applications with a multiple controller configuration, the controller must have an open drain output with an external pull-up resistor. In applications using a single controller configuration, the SCL line can be driven by a controller buffer configured as a push-pull (therefore the pull-up resistor on the SCL line is not required).

For a safe design, the SCL line must never be left floating (high-Z) and must be driven low when SDA transitions are not under control to avoid undesired START and STOP conditions. As a consequence, power up, power down phases as well as reset states, can be secured using a pull-down resistor on the SCL line. This minimizes the chances of a parasitic STOP/START condition, when the controller releases the I²C bus.

Note: A STOP condition can be decoded as the trigger of a write cycle if this STOP condition appears at the end of a data byte inside a transmitted write command.

The input pin leakage is negligible (typically a few nA). Input schematic (including a protection circuit) does not offer any open path to V_{SS} or V_{CC}. Therefore the SCL level can be set before the EEPROM powers up and can remain high even after power down with no risk of high leakage or EOS.

Figure 19. Serial clock input SCL

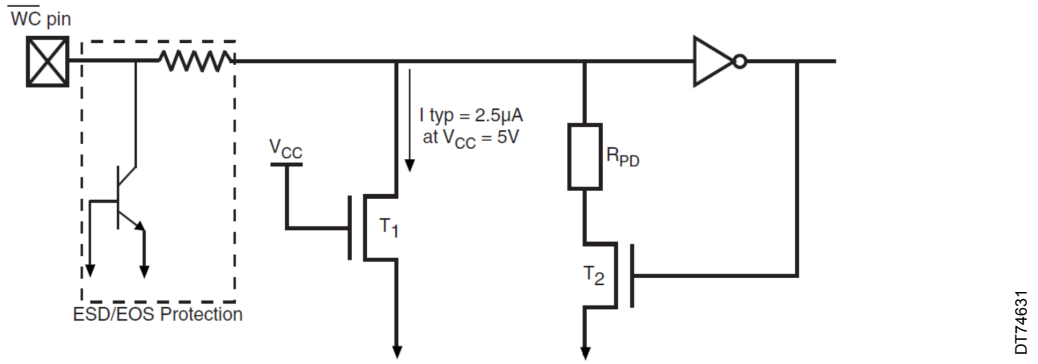


4.1.4 Write control ($\overline{WC}$)

The write control ($\overline{WC}$) input includes an internal pull-down resistor, in case the application designer leaves it floating (see Figure 20). If no write protection is necessary, tie it to V_{SS}. The best write protection is obtained by connecting the write control input to a controller output and a pull-up resistor to V_{CC} (see Figure 22), thus allowing a default write protection also during the critical power-up, power down and application reset phase. Prior to issuing any write instruction the $\overline{WC}$ pin must be driven low, and kept low after the STOP condition ending the write command.

The leakage current depends upon the pin voltage. See Table 5.

Figure 20. **Write control input ($\overline{WC}$)**



The $\overline{WC}$ pull-up resistor must be compatible with the controller output driving capability, as detailed in Table 5.

Table 5. **Connecting $\overline{WC}$ inputs in I²C products**

I²C	1 to 16 Kbits	32 Kbits and higher
$\overline{WC}$ internal pull-down resistance (R_{PD})(1) (devices identified with process letters K or T)	50 kΩ	
$\overline{WC}$ internal pull-down resistance (R_{PD})(1) (all previous devices)	15 kΩ	30 kΩ
External pull-up (R_{PU})	R_{PD} / (R_{PU} + R_{PD}) > 0.7, therefore R_{PU} < 3 R_{PD} / 7	
$\overline{WC}$ inputs not connected (left floating)	$\overline{WC}$ is read as 0	
Input leakage I_{li}	I_{li} = 0 for V_i = 0 V I_{li} = V_i/R_{PD} for 0 < V_i < V_{IH} I_{li} < 5 µA for V_i > V_{IH}	

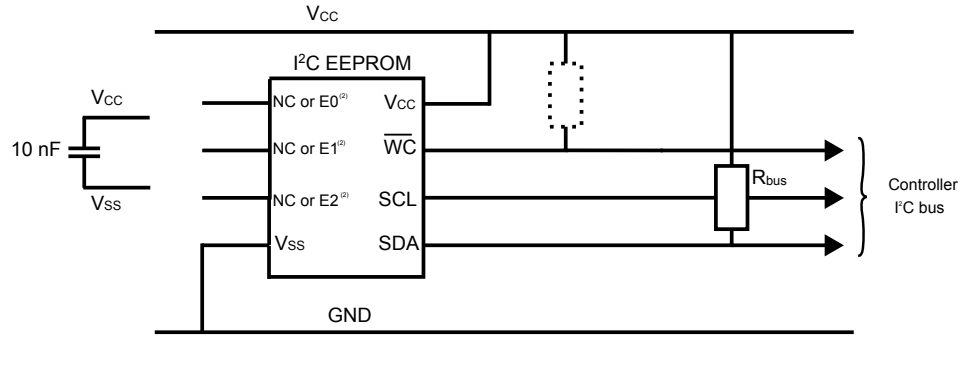
1. *These pull-down values can change within the range authorized in the datasheet without previous notice.*

4.1.5 Recommended I²C connections

The recommended connections are shown in Figure 21 and Figure 22. Note that for both circuits:

1. The decoupling capacitor (10 nF min) must be placed as close as possible to package pins V_{CC} and V_{SS}.
2. A pull-up resistor can be used only when the $\overline{WC}$ pin is driven by a controller I/O. If unused, it must be connected to the ground or left floating.
3. The I²C specification recommends to connect a 220 Ω serial resistor on SCL and SDA. This is not useful unless identified overstress can occur on these pins, or electromagnetic disturbances must be reduced.

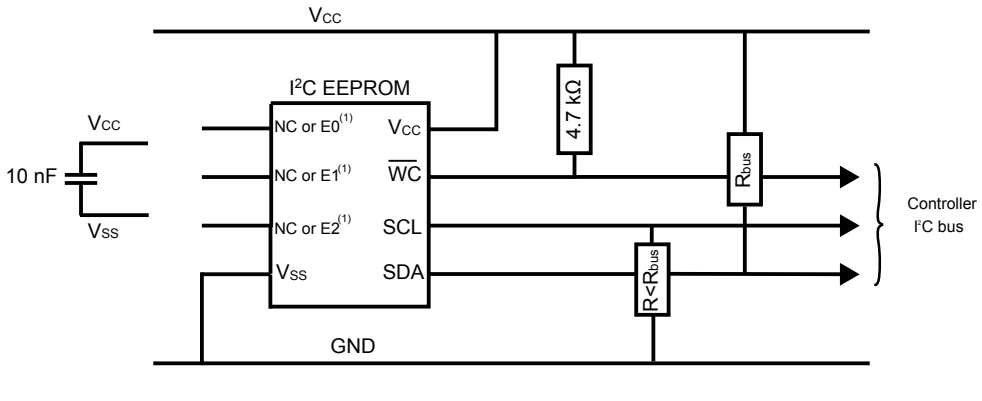
Figure 21. **Recommended I²C connections - Safe design**



Note:

1. *The pull-up resistor values must be large enough so that the I²C-bus controller can pull these lines low (current sink capability of the I²C-bus controller I/Os).*
2. *NC for I²C CMOSF9V products. For others, E0/E1/E2 must be connected either to V_{CC} or to GND.*

Figure 22. **Recommended I²C connections - Robust design**



Note:

1. *NC for I²C CMOSF9V products. For others, E0/E1/E2 must be connected either to V_{CC} or to GND.*
2. *The use of external pull-up and pull-down resistors is strongly recommended even if the controller I/Os for the I²C bus are already providing them. Refer to section Section 6.1.1 for more details.*

4.2 SPI products (M95xxx devices)

4.2.1 Chip select ($\overline{S}$)

Chip select ($\overline{S}$) is a CMOS Schmitt trigger input buffer protected from ESD and EOS spikes, as shown in Figure 23 .

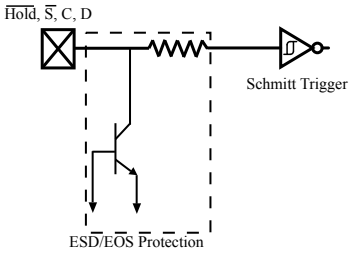
Chip select ($\overline{S}$) must never be left floating, and must be constantly held at V_{CC} outside the communication slots. It is strongly recommended to add a pull-up resistor to ensure a high level on the chip select pin at any time and in particular during power-up, power-down and during the reset phase of the controller, since, during these phases, the controller usually leaves its I/Os in high impedance.

The pull-up resistance should be calculated as a function of the bus capacitive load so that the voltage on the $\overline{S}$ pin always remains above $V_{IH} = 0.7\ V_{CC}$ during power-up.

The input pin leakage is negligible, typically a few nA. The input schematics, including the protection circuit, does not offer any open path to the V_{SS} or V_{CC}.

Note: To filter out wrong selections, the M95xxx devices are internally selected only when the following two conditions are met: $\overline{S}$ falling edge and $\overline{S}$ remains low.

Figure 23. Chip select, clock, data, hold input pins



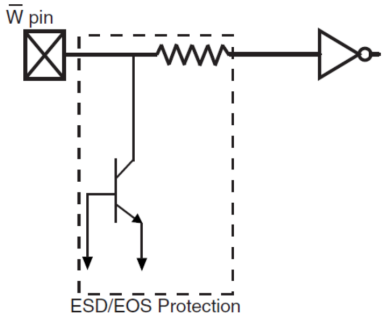
4.2.2 Write protect ($\overline{W}$)

This is a CMOS input used to enable or disable write protection. To ensure the write protection at power-up and power-down, its default state is low. It is therefore recommended to add a pull-down resistor (whose value must be smaller than the pull-up resistor on $\overline{S}$ line) on $\overline{W}$, to optimize write protection latency when the controller releases the SPI bus in high-Z (power-up, power-down, and controller reset phases). In this case the write protect ($\overline{W}$) line goes low before the chip select ($\overline{S}$) line goes high (since the time constant of the $\overline{W}$ pull-down resistor is smaller than the time constant of the $\overline{S}$ pull-up resistor, see Section 4.2.6: Recommended SPI connections), thus preventing the potential execution of an ongoing write command.

Note: For 1 to 4 Kbits SPI devices, the $\overline{W}$ protects the write to memory and for devices of density above 8 Kbit/s, the $\overline{W}$ protects the write in the status register.

The input pin leakage is negligible, typically a few nA. The input schematic, including the protection circuit, does not offer any open path to V_{SS} or V_{CC}.

Figure 24. Write protect input $\overline{W}$



4.2.3 Serial data input (D) and serial clock (C)

The serial data input (D) and serial clock (C) signals are connected to a CMOS Schmitt trigger input buffer and should be controlled by push-pull buffers (from the SPI controller bus). An external pull-down resistor on serial clock (C) signal prevents “out-of-specification” configurations like simultaneous rising edges on $\overline{S}$ and C when the controller releases the SPI bus (violation of the t_{SHCH} and t_{CHSH} timings). An external pull-down resistor on the serial data input (D) (see Figure 27) will optimize the signal control and the device standby current.

The pull-down resistor value on C is optimized if its value is larger than the pull-up resistor value on the chip select ($\overline{S}$) line. In this case, if the SPI bus is released (high-Z state), the chip select ($\overline{S}$) line goes high faster than the clock (C) line goes low and so deselects the device before the clock signal crosses the input buffer trigger point (around $V_{CC}/2$).

The input pin leakage is negligible, typically a few nA. The input schematic, including the protection circuit, does not offer any open path to the V_{SS} or V_{CC}.

Note: If the clock (C) line cannot be pulled down (and must be pulled up due to other system constraints), it is recommended to choose a lower pull-up resistor value (resistor value at least three times lower) than the pull-up resistor value on the chip select ($\overline{S}$) line. Moreover, the "out-of-specification" configuration can also be minimized by connecting the hold ($\overline{HOLD}$) pin to the reset signal (active low) of the controller: if the controller leaves the SPI bus in high-Z, the hold ($\overline{HOLD}$) line goes low, locking the clock to a low level (if already low), thus preventing the occurrence of a rising edge on both the clock and chip select lines (this prevents the violation of the t_{CHSH} and t_{SHCH} timings).

4.2.4 Hold ($\overline{HOLD}$)

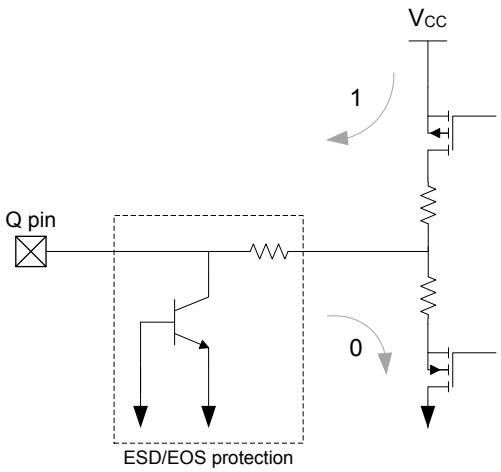
The hold ($\overline{HOLD}$) is a CMOS Schmitt trigger input buffer used to pause communication. It should be driven by a push-pull buffer (SPI controller bus) for a better timing control, or tied directly to V_{CC} if unused. The hold pin cannot be left floating. The hold input can be set before the EEPROM power up and can remain high after power down. The hold input will not sink a current even if a voltage higher than V_{CC} is applied to it.

The input pin leakage is negligible, typically a few nA. The input schematic, including the protection circuit, does not offer any open path to the V_{SS} or V_{CC} (see Figure 23).

4.2.5 Serial data output (Q)

The serial data output (Q) is a push-pull tri-state output buffer. The serial data output being often in the high-Z state (for example, during a write) and connected to a controller input pin, it may be useful to connect it to a pull-up (or pull-down) resistor to set a default value on the bus and thus prevent the controller input from toggling (see Figure 23. Chip select, clock, data, hold input pins). Application designers must be aware that connecting several devices on the serial data output (Q) increases the capacitive load of the line. The access time of the device is tested with a 100 pF or 30 pF capacitive load, depending on the clock frequency (refer to the M95xxx device datasheet for the values of the capacitive load and clock frequency).

Figure 25. Output pin tri-state buffer



Note: The push-pull upper transistor (NMOS transistor connected to V_{CC}) offers a parasitic reverse diode to V_{CC}, therefore the voltage applied on Q must never be higher than V_{CC}.

4.2.6 Recommended SPI connections

The recommended connections are shown in Figure 26 and Figure 27. Note that for both circuits:

1. The decoupling capacitor (10 nF min) must be placed as close as possible to package pins V_{CC} and V_{SS}.
2. $\overline{S}$ input is pulled high with R_{PU}, and C input can be pulled low with R_{PD}. In doing so, if the SPI controller bus leaves $\overline{S}$ and C in high-Z, the SPI EEPROM is deselected and it is ensured that clock C remains low or falls low (unlike a case where $\overline{S}$ and C can rise together, that is t_{CHSH} = 0, out of a specification event).
3. If unused, the hold and $\overline{W}$ pins must be directly connected to V_{CC}, as a CMOS input must never be left floating.

Figure 26. Recommended SPI connections - Safe design

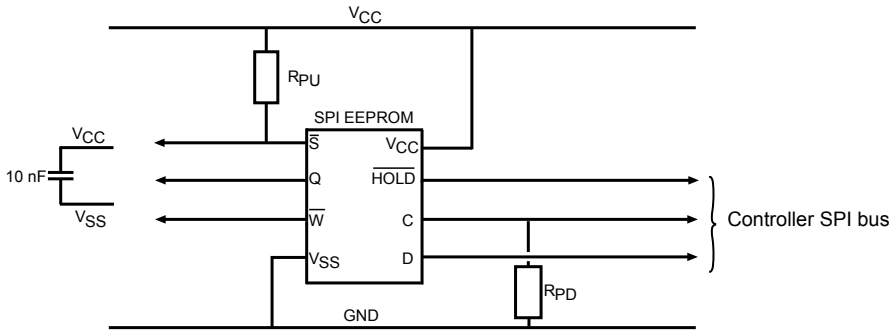
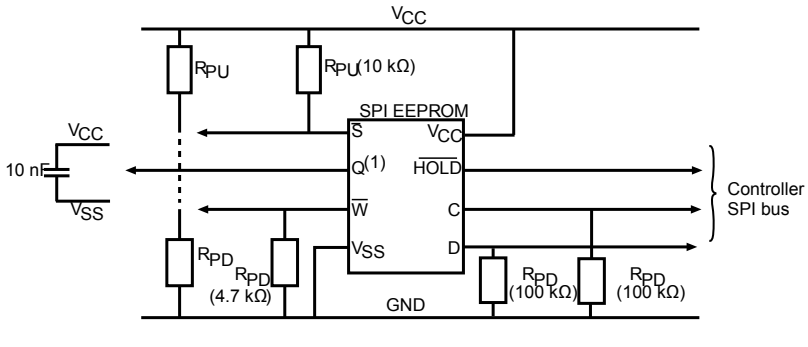


Figure 27. Recommended SPI connections - Robust design



Note: Q must be connected either to R_{PD} or R_{PU}.

Table 6. Calculation for external pull-up and pull-down resistors in SPI products

Resistor	EEPROM input pins ($\overline{S}$, $\overline{W}$, $\overline{HOLD}$, C, D)	EEPROM output pin Q
R_{PD}	R_{PD} > V_{IH} EEPROM / I_{OH} controller	R_{PD} > V_{IH} controller / I_{OH} EEPROM
R_{PU}	R_{PU} > (V_{CC} - V_{IL} EEPROM) / I_{OL} controller	R_{PU} > (V_{CC} - V_{IL} controller) / I_{OL} EEPROM

4.3 Microwire products (M93Cxxx and M93Sxxx devices)

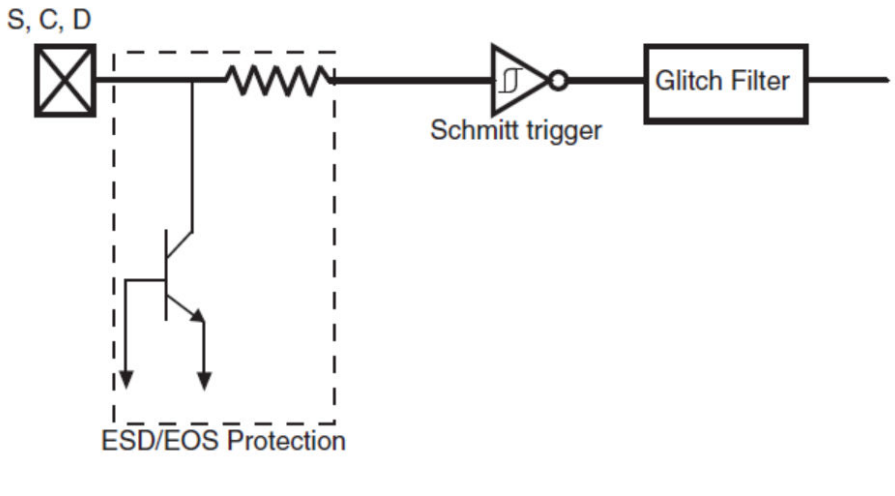
4.3.1 Chip select (S)

This input pin is connected to a CMOS Schmitt trigger input buffer (see Figure 28). It is recommended that the controller bus controls it with a push-pull buffer.

S must never be left floating. It is therefore strongly recommended to add a pull-down resistor to ensure a low level on the chip select input at any time, so that the device remains deselected, including during periods when the S line is in high-Z, such as power-up, power-down, and the reset phase of the controller.

The input pin leakage is negligible, typically a few nA. The input schematic, including the protection circuit, does not offer any open path to V_{SS} or V_{CC}.

Figure 28. Chip select, clock, data input pins



4.3.2 Serial data (D) and serial clock (C)

These input pins are connected to a CMOS Schmitt trigger input buffer (see Figure 28). It is recommended that the controller bus controls them with a push-pull buffer. An external pull-down resistor (R_{PD}) on a serial clock (C) signal prevents from an out-of-specification configuration, such as a clock rising edge while chip select goes low when the controller releases the bus (high-Z state, violation of the t_{SLCH} timing)). An external pull-down resistor on serial data input (D) optimizes signal control and standby current.

Pull-down resistor values on serial clock (C) and serial data (D) are optimized if they are at least three times bigger than the pull-down value on the chip select (S) line. In this case, if the SPI bus is released, the S line goes low faster than the clock (C) line, and so, deselects the device before the clock signal crosses the input buffer trigger point (area around $V_{CC}/2$ is always sensitive).

If the clock line cannot be pulled down and must be pulled up due to other system constraints, it is recommended to choose a weaker pull-up value (at least three times weaker) than the pull-down value on chip select.

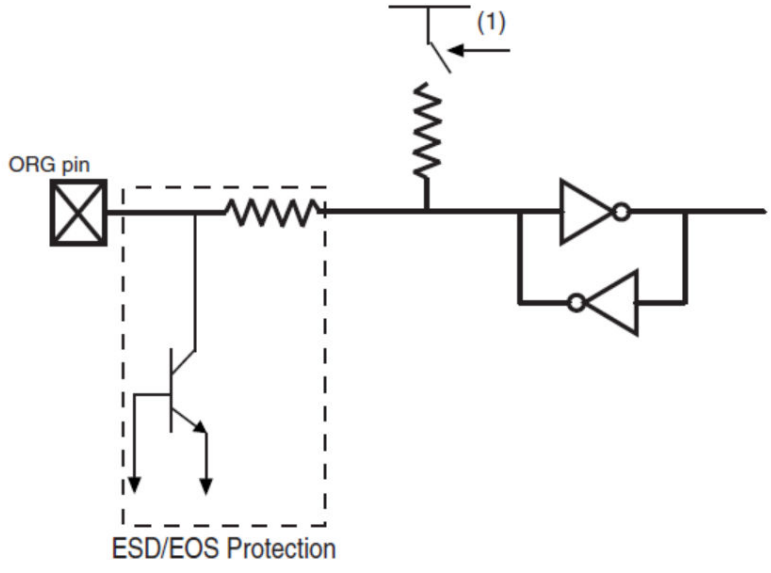
The input pin leakage is typically a few nA. The input schematic, including the protection circuit, does not offer any open path to V_{SS} or V_{CC}.

4.3.3 Organization select (ORG)

ORG is not a CMOS input. If left floating, it is interpreted internally as high. It is strongly recommended to connect it directly to the V_{CC} or V_{SS} pin of the device. Driving it dynamically implies that application software can handle specific Microwire dual organization and switch from Single data byte management to word data management.

The input pin leakage is negligible, in standby mode, typically a few nA. The input schematic, including the protection circuit, does not offer any open path to the V_{SS} or V_{CC} (see Figure 29).

Figure 29. Organization input ORG



Note: 1. *The pull-up resistor is only active for a short period of time (right after the device selection) to latch the level read on the ORG pin when it is left floating.*

4.3.4 Serial data output (Q)

This is a push-pull tri-state output buffer. As the serial data output is often in the high-Z state and connected to a controller input pin, it may be useful to connect a pull-up resistor to set a default value (corresponding to the Ready state) on the bus and thus prevent the controller input from toggling. Application designers must be aware that connecting several devices on serial data output (Q) increases the capacitive load of the line. Access time of M93Cxxx devices is tested with a 100 pF load (see Figure 25).

4.3.5 Don't use (DU)

This pin does not contribute to the normal operation of the device. It is reserved for use by STMicroelectronics during test sequences. The pin may be left unconnected or may be connected to V_{CC} or V_{SS}. No other connection is allowed. Direct connection of DU to V_{SS} is recommended for the lowest standby power consumption mode.

4.3.6 Recommended Microwire connections

the recommended connections are shown in Figure 30 and Figure 31. Note that for both circuits:

1. A decoupling capacitor (10 nF min) must be placed as close as possible to the package pins V_{CC} and V_{SS}.
2. A 50 Ω resistor can be connected to V_{CC} if extra filtering on V_{CC} is needed or if an identified latchup risk must be minimized

Figure 30. **Recommended Microwire connections - Safe design**

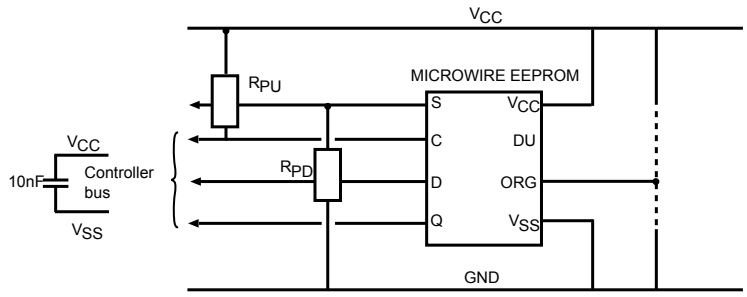


Figure 31. **Recommended Microwire connections - Robust design**

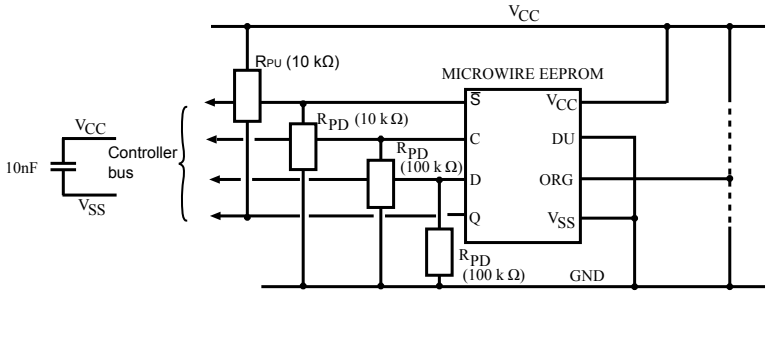


Table 7. **Calculating external pull-up and pull-down resistors**

Recommended connection	EEPROM input pins ($\overline{S}$, C, D, $\overline{W}$, ORG)	EEPROM output pin Q
R_{PD}	$R_{PD} > V_{IH}$ EE / I_{OH} controller	$R_{PD} > V_{IH}$ controller / I_{OH} EE
R_{PU}	$R_{PU} > (V_{CC} - V_{IL}$ EE) / I_{OL} controller	$R_{PU} > (V_{CC} - V_{IL}$ controller) / I_{OL} EE

4.4 PCB layout considerations

The full system layout becomes ever more critical because of space constraints, high communication speed, noise due to interference and all EMC constraints.

4.4.1 Cross coupling

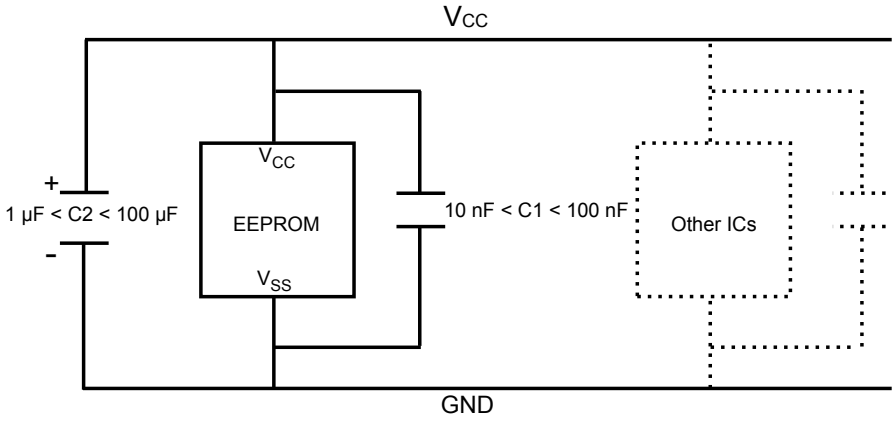
The cross-coupling effect increases with the frequency and fine PCB technology. All floating signals or pins, weak pull-up, or pull-down connections are very sensitive to cross coupling and antenna-like pick-up. All unused pins should be tied correctly (in general to V_{CC} or V_{SS}).

4.4.2 Noise and disturbances on power supply lines

Noise and disturbances on power supply lines must be filtered out. For a robust design it is recommended to place local decoupling capacitors. Low inductance capacitors in the range of 10 nF to 100 nF are usually preferred.

Electrolytic capacitors in the range of 1 µF to 100 µF can also be placed close to the power supply source of the system. Avoid Ground and V_{CC} loops on the PCB as it increases the sensitivity to electromagnetic fields. (See Figure 32)

Figure 32. PCB decoupling



5 Software considerations

The purpose of the suggestions presented below is to help the application designers make the most of ST's EEPROM products, and to help to improve the system robustness and compatibility with future EEPROM devices.

5.1 Electrical parameters

Low-level drivers (hardware dependent) must follow the electrical parameters for correct communication. EEPROM samples are not absolute references for software validation as they are not representative of production variations. The timings given in the device datasheets are unique reference characteristics. They correspond to minimum and maximum timing values to be taken into account for hardware and software calibration.

Typical errors to be avoided in applications are:

- Input voltage levels not compatible with the specifications, $V_{IL} < 0.3\ V_{CC}$ and $V_{IH} > 0.7\ V_{CC}$.
- Excessive current requested from EEPROM data output buffer (output CMOS levels are no more guaranteed).
- Write time not completed (t_W = 4 or 5 ms) before issuing a new command.
- Data setup time in applications using high clock rate or very smooth waveforms with slow transitions (as a general recommendation, signal rise and fall time must be less than 10% of the clock period).
- Out of specification (too short) pulses on the clock signal, chip select signal or on Start/Stop conditions.

The behavior of EEPROM devices operating out of specification can never be guaranteed and is not always predictable. Moreover, major compatibility issues may arise when switching to a new device version or using a compatible device from another supplier. When using a double source supplier for EEPROMs, use the worst value of each single timing as the reference for direct compatibility.

5.2 Optimal write control

EEPROM devices are simple products with few operating modes and instructions. It is nevertheless worth focusing on the features that can be used to improve performance and application robustness.

5.2.1 Page mode

The memory array is divided into pages. The size of a page is given on the first page of the product datasheet (it can be 8, 16, 32, 64, 128, 256 or 512 bytes).

The write page mode is used to write a block of data bytes in a single shot. The write page mode sequence consists of a write instruction with the start address and one or more data bytes directly followed by the internal execution of the operation (t_W).

- The maximum number of bytes programmed during a write page is limited by the page length of the product.
- A data block can be programmed starting at any offset inside the page.
- The address of the first data byte to program is given in the instruction, other data bytes are programmed in consecutive addresses.
- If the last location in a page is reached when shifting in the data bytes during a write command, the internal address pointer rolls over to the first byte inside the same page. It is therefore not possible to store data in two different pages with a single write page instruction.
- If more data bytes than the page length (for instance 32) are shifted in, only the last received data bytes (last 32 bytes) are programmed in the page (The 33rd data byte shifted in will replace the 1st data byte shifted in, and so on).

To write to the EEPROM, it is recommended to use the page mode instead of the byte mode whenever possible. The programming time (t_W) is independent of the number of bytes to program and the page mode has two main advantages:

1. It speeds up the application when storing or updating data.
2. It minimizes the high-voltage programming stress and naturally extends the cycling endurance.

5.2.2 Data polling

Data polling is a very safe and optimal way of managing the write time (t_W). The aim is to check the EEPROM status before sending the next instruction, so as to prevent bad controller/target communications.

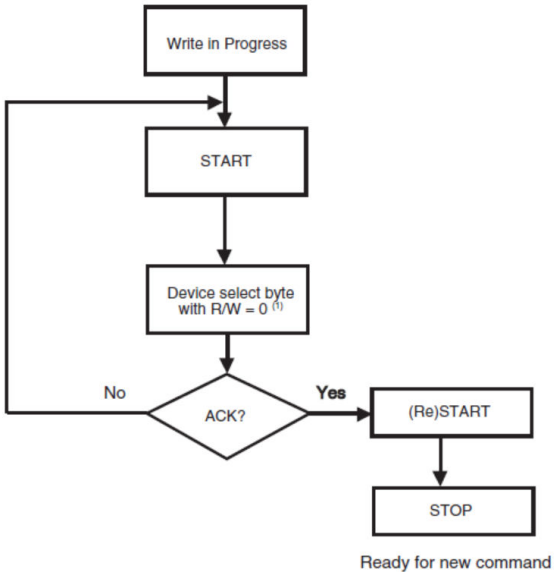
Data polling is a software loop used to optimize the write wait time and control the correct operation of the device. Moreover, software, which has data polling is able to adapt to different devices regardless of the specified write time.

This data polling algorithm must be coupled to a time-out counter to limit the data polling time and avoid some endless polling. The timeout limit should be higher than the maximum write time of all devices used (typically 15 ms should be enough).

I²C products

In I²C products, the device does not respond (NoAck) when a programming operation is in progress. Data polling thus consists in sending a device code in a loop mode and tracking the EEPROM acknowledgment. It is highly recommended to poll the device with a write instruction.

Figure 33. I²C data polling algorithm

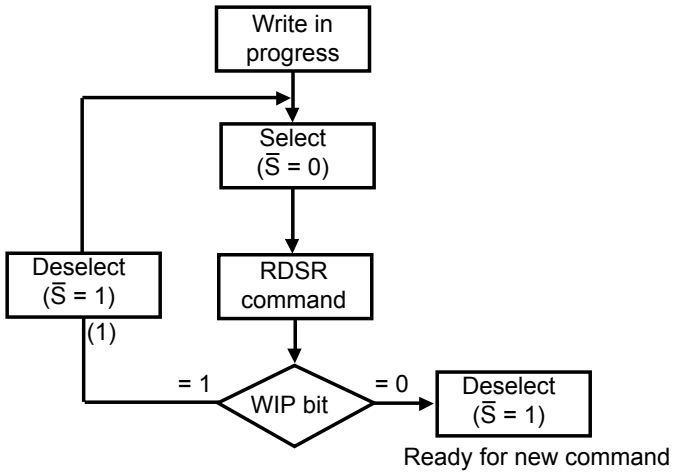


Note: 1. Using the READ device code (R/W = 1) is hazardous due to I²C protocol constraints.

SPI products

In SPI products, a specific instruction (read status register - RDSR) is used to check the status of the WIP (write in progress) bit in the status register. A loop on the RDSR command (RDSR instruction + data byte) checks the WIP bit status. As soon as it returns to 0, the device is ready to accept new commands. For compatibility reasons, it is recommended to send the full RDSR command each time instead of continuously reading the status register.

Figure 34. SPI data polling algorithm

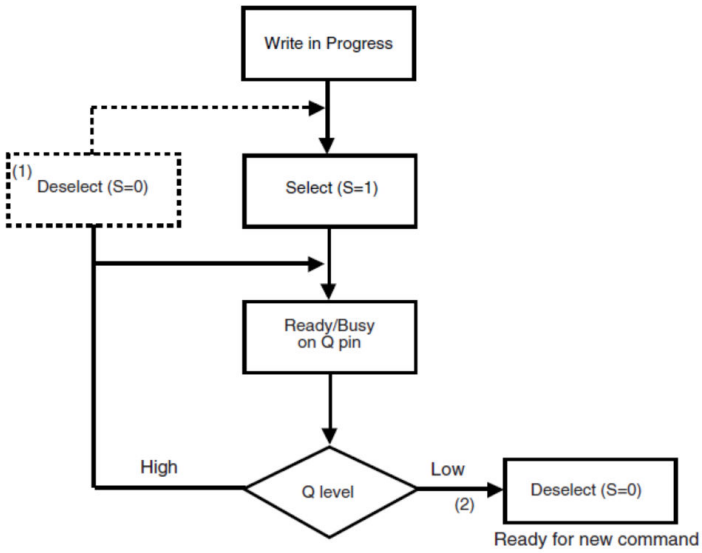


Note: *1. Although ST EEPROMs allow continuous read of the status register it is, for compatibility reasons, recommended to send each time the full RDSR command.*

Microwire products

In Microwire products, the data output pin (Q) indicates the ready/busy status when the chip select pin ($\overline{S}$) is driven high. Once the device is ready, the Q output goes high-Z after a start condition. It is strongly recommended not to operate the clock during the data polling sequence because as soon as the chip is ready, the logic starts to decode incoming bits.

Figure 35. Microwire data polling algorithm



Note:
1. *There is no difference in the data polling process if chip is deselected between two ready/busy checks*
2. *It is strongly recommended not to operate the clock during the data polling sequence because as soon as the chip is ready, the logic starts to decode incoming bits*

5.3 Write protection

5.3.1 Software write protection

Use the software write protection features to protect sensitive data items:

- Only I²C products manufactured in CMOSF9V offer software write protection through the SWP register. The two nonvolatile BP0 and BP1 bits are dedicated to define the size of the memory block to protect. The nonvolatile WPA bit enables or disables the write protection.
- In SPI products, two nonvolatile status register bits (BP0, BP1) are dedicated to the software write protection. The upper quarter, the upper half or the whole memory array can be set as read-only.

Data items like trace codes, identification codes, manufacturing configurations, default parameters and all sensitive data in general, can be software protected against corruption during field service. Software protection bits are nonvolatile bits and therefore offer the same cycling and data retention performances than the EEPROM memory bits.

5.3.2 Hardware write protection

By default the hardware write protection feature should be set, and the time during which the device is left unprotected should be limited to the time required to issue and execute write instructions. The hardware write protection is very effective against parasitic or hazardous instructions transiting on the interface bus.

Also use the hardware write protection features during power-up, power-down, and normal operation (refer to the Section 4: Hardware considerations).

- The $\overline{WC}$ pin in I²C products protects the entire array.
- The $\overline{W}$ pin protects the entire array by resetting the write enable latch (WEL) bit in 1 Kbit to 4 Kbit SPI products whereas it protects the nonvolatile bits of the status register in 8 Kbit to 4 Mbit SPI products.

It is recommended to change the state of the write-protect pin only if no data transfer or program cycle is in progress. The write protect and write control pins should be controlled with very conservative timings:

- 1 clock cycle time clearance (with no data transfer) before the select (or start) event.
- 1 clock cycle time clearance (with no data transfer) after the deselect (or stop) event.
- Wait for the write cycle (t_W) completion before changing the protection.

This conservative sequence will not affect the communication speed but will ensure the safe operation of the products (see Figure 36 and Figure 37).

The write protect signal ($\overline{W}$ for SPI, $\overline{WC}$ for I²C) is glitch sensitive and a short (parasitic) pulse could cause a write request to be aborted. This feature can also be of great help in emergency situations like power loss or controller reset. See Section 6: Power supply loss and application reset for details.

Figure 36. Recommended use of the $\overline{WC}$ pin - I²C products

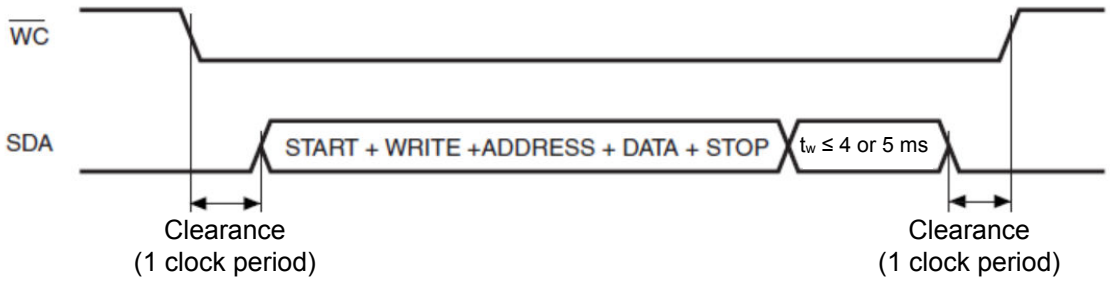
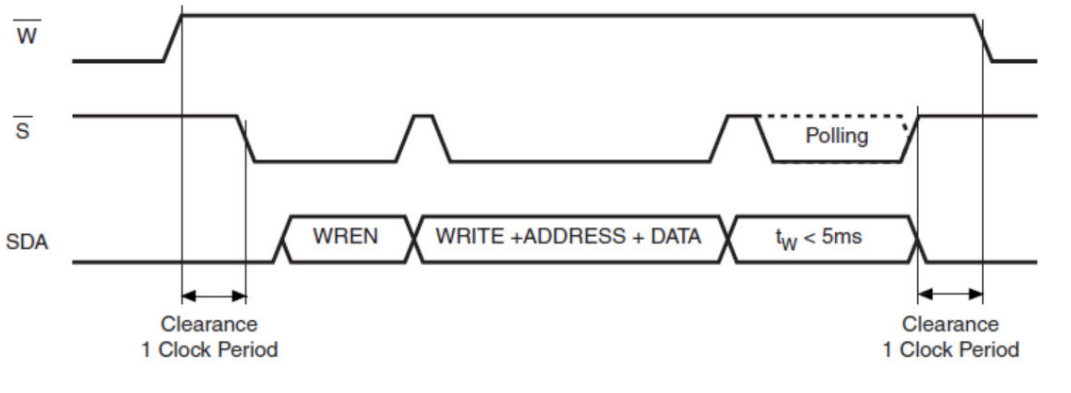


Figure 37. Recommended use of the $\overline{W}$ pin - SPI products



5.4 Data integrity

In several applications, the implementation of a data integrity strategy is mandatory. Several strategies are possible, but whatever the solution used, extra memory density is required to hold the extra data.

5.4.1 Checksum

This is perhaps the more commonly used method to prevent data loss, data corruption and poor communication. It consists in computing a checksum of the data to write, and in storing it into the memory array as an additional data byte. Checksums are particularly suitable for the secure communication of parameters that are often read and updated.

To give applications more robustness, more elaborated routines (like error code correction) can be used to correct errors.

5.4.2 Data redundancy

Data redundancy is a technique used to prevent data loss and corruption, particularly indicated for the read-only data stored in the EEPROM. Typically, these data are programmed once during module manufacturing, and they are read during the whole application lifetime. As they are not refreshed, there is a higher probability of retention loss. With redundancy, there is a backup on each byte.

The efficiency of redundancy depends on the physical structure of the memory. As described previously, the memory array is organized as rows (or pages) and columns where each byte location (address) is the intersection of a row and a column. Redundant data should not share the same row (or page) and column (byte location inside one page) but should be located at physically independent addresses (see Figure 38).

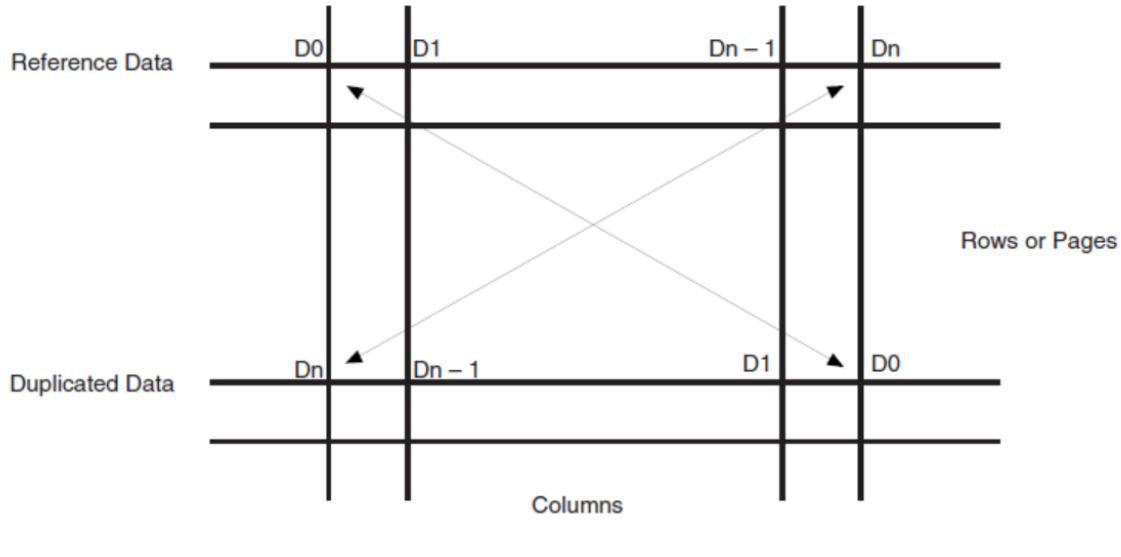
Table 8. Column and page address bits according to page length

Page length	Column address bits	Page address bits
16 bytes	4 least significant bits	All other bits
32 bytes	5 least significant bits	
64 bytes	6 least significant bits	
128 bytes	7 least significant bits	
256 bytes	8 least significant bits	
512 bytes	9 least significant bits	

It is preferrable to not locate redundant data in the same page as the reference data. The address of the duplicated data should differ from the original address by at least 1 bit for the the column, and 1 bit for the page (see Table 8).

Note: More detailed information on memory array, data scrambling and address decoding are available on request.

Figure 38. **Example of how to duplicate data safely**



Note: N = Page length (number of bytes defining a page)
Di = Data byte at page location "i" (i < N)
n = Multiple of 4 (n < N).

5.4.3 Checksum and data redundancy

Combining checksums and data redundancy is the best strategy.

Applying the redundancy and checksum strategy to read-only data (two or three copies with a checksum byte for each data block) improves the EEPROM robustness and, with it, data integrity.

5.4.4 Extra redundancy

Default backup parameters can also be stored in another nonvolatile external or embedded flash memory. The micro should then have the ability to copy back the data in the EEPROM when necessary.

5.5 Cycling endurance and data retention

Even if EEPROM devices are able to withstand a very high number of write cycles, the EEPROM should not be used in replacement for a nonvolatile RAM buffer. The cycling budget during application life has to be considered.

5.5.1 Cycling and data retention qualification procedures

Cycling qualification

During the cycling qualification, devices are cycled with a dedicated test program, which allows to program in one single write cycle the whole memory array with the same byte value. ST can thus guarantee that, for each write operation performed with this test program:

- Each memory array cell is cycled once;
- the logic and the associated internal voltages that control the write cycles are cycled once.

Therefore, in a device specified as 4 Mcycles, the qualification results allow to consider that each byte (1.) can be cycled 4 million times.

Note: 1. Byte for device with ECC X 1 or without ECC and word of 4 bytes for devices with ECC x 4

Data retention qualification

The data retention qualification tests check that the data written to the EEPROM remain available with a correct programming level after a bake at 150 °C during 1000 hours (high temperature significantly accelerates the data retention drift).

5.5.2 Optimal cycling with ECC

Some ST EEPROM devices embed an internal ECC (error correction code) logic, which improves data retention performance. This ECC logic must be taken into account when defining the application cycling budget.

For devices embedding the ECCx4, the ECCx4 logic compares each group of 4 bytes with six additional EEPROM ECC bits associated with each group of 4 bytes. As a result, if a single bit out of 4 bytes of data happens to be erroneous during a read operation, the ECC detects it and corrects it with the correct value in the output stream of data (read data). The EEPROM cell read reliability is therefore improved by using this feature.

However, even if a single byte has to be written, 4 bytes are internally modified (plus the ECC bits), that is, the addressed byte is cycled together with the three other bytes making up the group.

For example:

- Writing one byte at address 0000h internally programs this byte plus the three following bytes and ECC bits.
- Writing the next byte at address 0001h internally programs this byte plus the three contiguous bytes (address 0000h, 0002h, 0003h), and the ECC bits.
- Writing the next byte at address 0002h internally programs this byte plus the three contiguous bytes (address 0000h, 0001h, 0003h), and the ECC bits.
- Wring the next byte at address 0003h, internally programs this byte plus the three contiguous bytes (address 0000h, 0001h, 0002h), and the ECC bits.

This example shows that each byte was cycled four times, although each byte was written only once. We can therefore conclude that writing data by groups of 4-bytes benefit of the highest number of write cycles.

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

Refer to AN2440 for more details concerning the ECC.

For devices embedding the ECCx1, the ECC logic compares each byte with four additional EEPROM ECC bits associated with each byte. As a result, if a single bit in a byte of data happens to be erroneous during a read operation, the ECC detects it and corrects it with the correct value in the stream of read data. The EEPROM cell read reliability is therefore improved by using this feature.

5.5.3 Cycling and temperature dependence

Write cycling and retention endurance are independent from the value of V_{CC}, but depend directly upon the operating temperature. The higher the temperature, the lower the cycling performance (see Section 1.1.3).

For a robust application design, the safe cycling operating area (by manufacturing technology) shown in Figure 40, Figure 39, Figure 41, and Figure 42 must be considered as a safe maximum cycling value for each byte of the memory. Going above this safe operating area is not recommended.

Figure 39. Cycling performance versus temperature for CMOSF8H EEPROMs (2 Mbits products excluded)

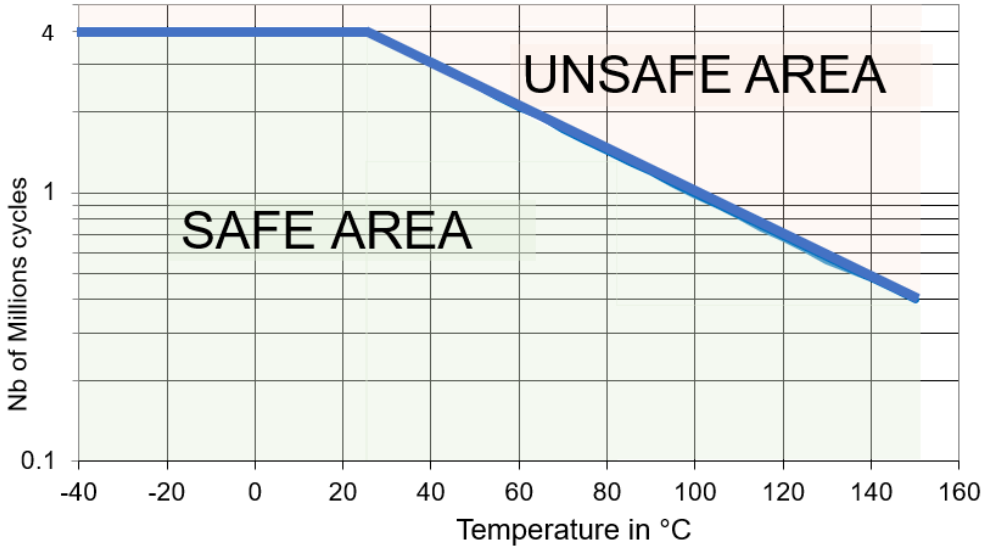
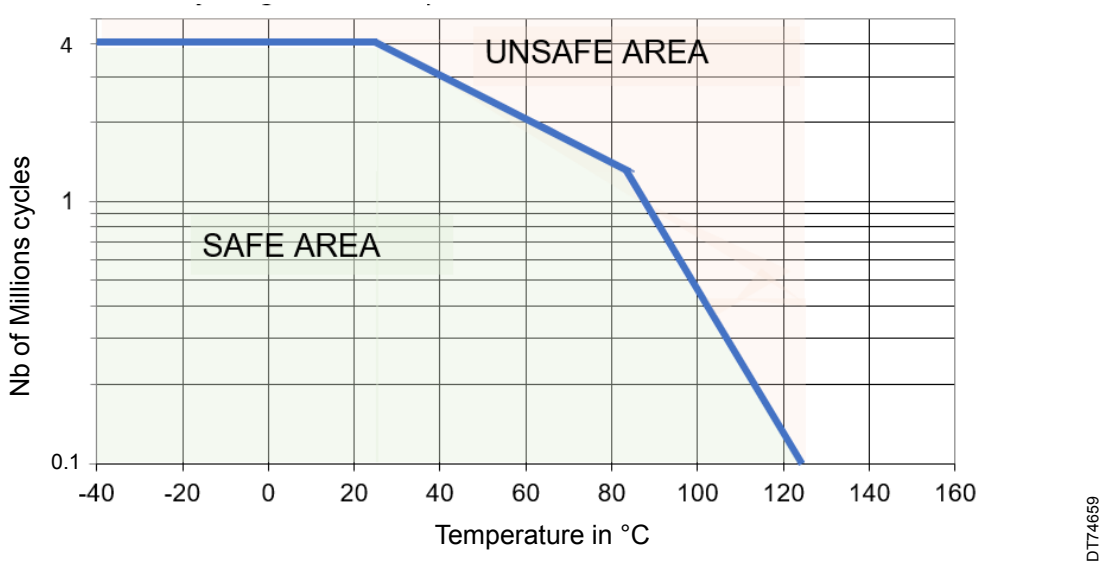
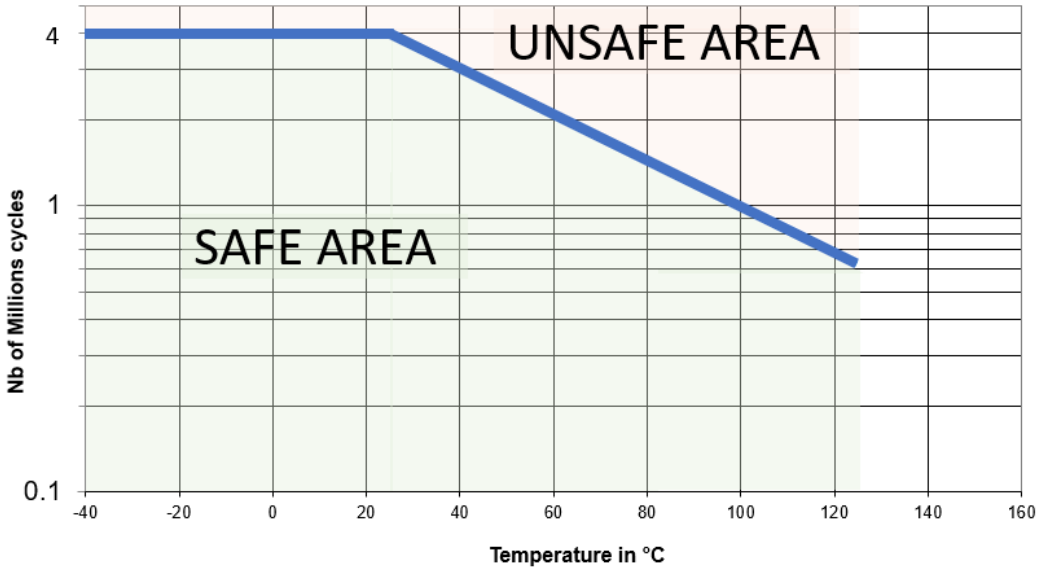


Figure 40. Cycling performance versus temperature for the 2 Mbits EEPROM manufactured in CMOSF8H



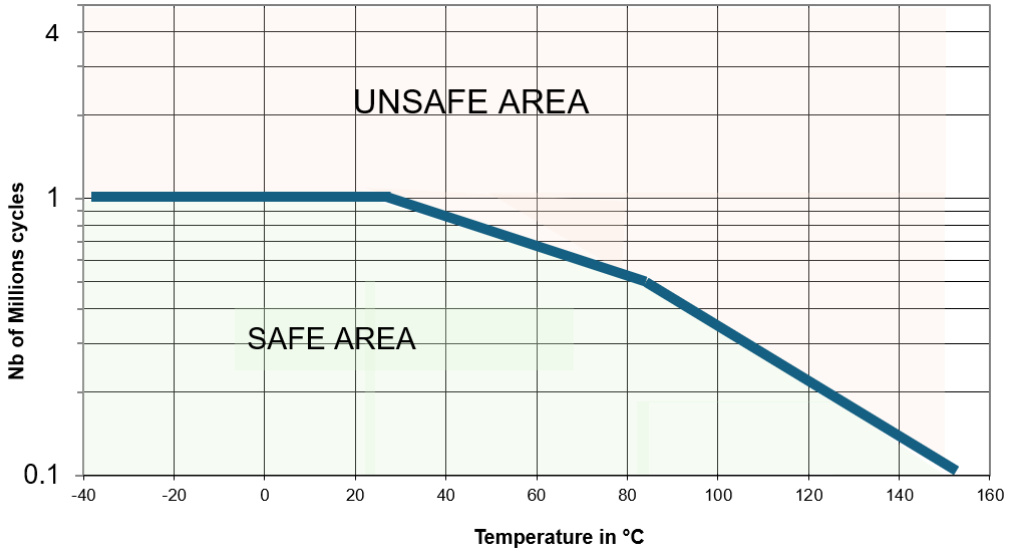
Note: The intrinsic (belonging to the essential nature or constitution of the die) cycling limits measured on the CMOS F8H automotive devices are much higher than the safe conditions suggested in Figure 40.

Figure 41. Cycling performance versus temperature for CMOSF9V EEPROMs (4 Mbits automotive products excluded)



DT74661

Figure 42. Cycling performance versus temperature for CMOSF9V EEPROMs (4 Mbits automotive products)



DT74662

5.5.4 Defining the application cycling strategy

ST EEPROM products have no built-in function to limit cycling and therefore application designers have to evaluate the number of write cycles that will be executed during the life of the end application. To ensure the safest conditions, it is strongly recommended to define a temperature profile of the write cycles performed by the EEPROM.

- Define the main temperature stages at which the EEPROM is operating in the end application.
- For each temperature, estimate the number of write cycles executed for each data block.
- For each data block (with different cycling profiles), calculate the cumulated cycling effect using Table 9. Application cycling profile evaluation .

Table 9. Application cycling profile evaluation

Temperature	Number of cycles (1)	% of specification max
25 °C	w	(w/1M) × 100 = a%
55 °C	x	(x/600K) × 100 = b%
85 °C	y	(y/300K) × 100 = c%
125 °C	z	(z/150K) × 100 = d%
Total	w + x + y + z	a + b + c + d%

1. *w, x, y, and z are the forecast number of cycles for a specific data block*

Note: The table can be adapted according to the temperature profile by taking care of putting down the maximum cycling for each temperature (using Figure 39, Figure 40, Figure 41, Figure 42).

If the total percentage of cumulated cycles (last row in Table 8) is lower than 100%, the data stored in the EEPROM are safely cycled.

If the total percentage of cumulated cycles is above 100%, the intrinsic safe margin for cycling is exceeded and a data relocation strategy must be defined.

Cycling on each EEPROM cell is not infinite (as shown in Figure 40), it is therefore wise to define a data relocation strategy by distributing the total number of cycles over several memory locations. To do this:

- Define a cycling limit for each data block according to the application needs and product performance (see Table 9. Application cycling profile evaluation).
- Count the number of cycles executed on each data block (counter value can be stored in the EEPROM).
- When the counter exceeds the defined limit, the cycled data block must be relocated to another physically independent memory address. The software developer should not move it to a location in the same page (when possible, not in the same column either) as the reference column/page. This means that the two addresses should differ by at least 1 bit in the page address and, if possible, by 1 bit in the column address. See Table 8 for page and column address bits. The counter is then reset and must also change address.

In addition, to optimize the number of cycles in the EEPROM and preserve the other data blocks in the memory array:

- Define data groups or classes (located in the same page) where data with similar update rates are gathered. This optimizes the use of the page mode instead of the byte mode.
- The area containing the read-only parameters and the cycled items should be separated and made independent as much as possible. Two types of data should not share the same pages and, where possible, not the same columns.

Following the above rules, in the laboratory environment, ST EEPROM devices have been demonstrated to reach hundreds of millions of cycles, safely.

5.5.5 Overall number of write cycles

As explained above, each EEPROM cell must not be cycled more than the maximum cycling value specified in the datasheet. However, the overall number of cycles executed by the memory can exceed this maximum value as several memory blocks can be cycled individually.

The safe value of the overall amount of cycling is 10 million cycles for any ST EEPROM. However, for products manufactured in CMOSF8H, characterization trials performed on products equal or larger than 32 Kbit, the safe value of the overall number of cycles at 25 °C is 128 M cycles.

6 Power supply loss and application reset

6.1 Application reset

During the application runtime, the controller may be reset by some external reset condition like a watchdog timer, a power supply monitor, or an ESD. In such cases, the serial bus is not controlled (it is left floating) while the power supply stays stable at its nominal value. When this occurs, the challenge is to either control the completion of the ongoing write cycle or to stop the communication with the EEPROM.

When the controller is reset, the EEPROM can be:

- Deselected in standby mode. This is the best and safest case. The EEPROM was in the idle state and will keep this state if the hardware connections are correct. The EEPROM is ready to accept any new command when the controller is restarted.
- Deselected while performing an internal write cycle. It is not a problem as at the end of the self-timed internal write process, the device returns to the standby state. Note that if the controller restarts while the EEPROM is still programming, the controller has to check that the EEPROM is ready by issuing a write cycle polling sequence.
- Selected while receiving a command or answering to a read command. This case has to be handled specifically according to the EEPROM protocol family.

The solutions discussed hereafter concern only the case where an EEPROM is selected. The aim of these recommendations is to properly stop the communication with the device to avoid further potential disturbances.

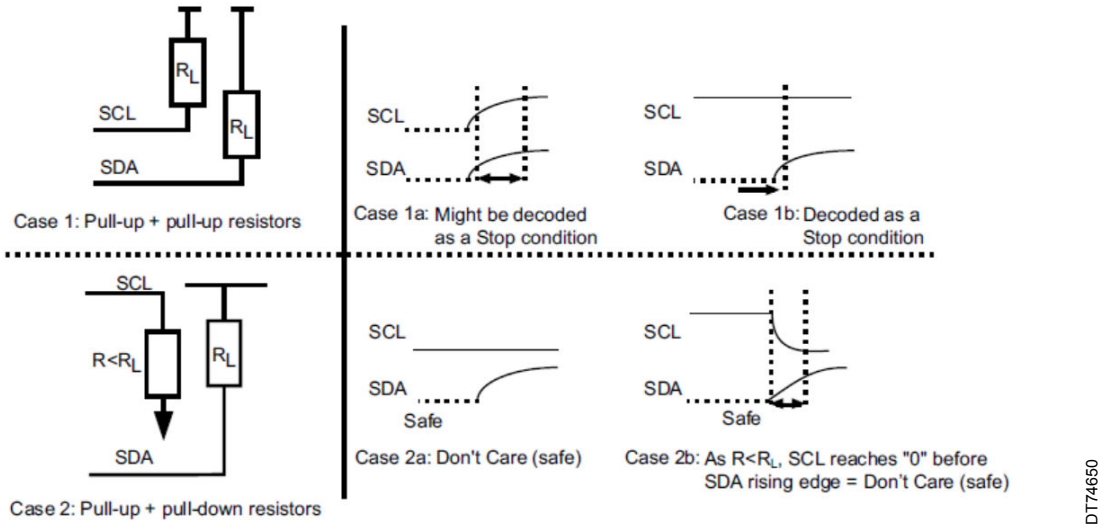
6.1.1 I²C products

The basic principle to protect an I²C transaction is to avoid issuing a stop condition when a reset occurs. This is because a stop condition can be decoded as the trigger of an undesired write cycle if the command was a write and if this stop condition occurs right at the end of a data byte.

On the other hand, the Start condition is a safe event, as it resets the internal state machine, and it is decoded by a specific logic block, always active.

Smart connections on the I²C bus lines help to avoid erroneous stop conditions when the controller is reset (I²C bus in high-Z), as described in Figure 43.

Figure 43. I²C bus enters the high-Z state (controller reset)



When the controller is reset and releases the I²C bus, the commonly used pull-up resistors on the SCL and SDA pins (case 1) increase the probability of sourcing an erroneous stop condition. The recommended I²C connections for a robust design are also detailed in Section 4.1.5.

When the transmission of an I²C bus command is interrupted before completion (before the stop condition), the EEPROM is paused in its communication until the controller is able to take again the control of the I²C bus. Before accessing the EEPROM, the controller must follow the sequence below:

- The controller must first send a resynchronization sequence to the EEPROM. It consists of nine START conditions plus one STOP condition to reinitialize the internal state machine and deselect the device safely. Refer to AN1471 *What happens to the M24xxx I²C EEPROM if the I²C bus communication is stopped?* for additional details.
- The controller must check that the device is ready (no write cycle in progress) by sending a data polling sequence. Refer to Section 5.2.2: Data polling.

Note: For the first read access, it is recommended to define the internal address pointer with a random read instruction as a current address read does not change/define the address pointer value.

These recommendations allow to maximize the control in case of inadvertent controller resets.

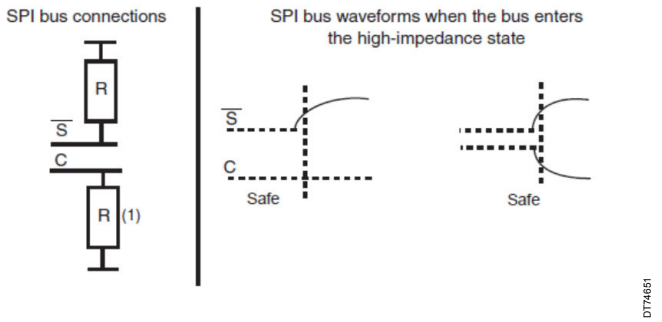
6.1.2 SPI products

The main recommendation for protecting an SPI transaction is to deselect the device in a safe way. The chip must be deselected taking care of the timings of chip select respecting to the clock rising edge.

- Smart connections of EEPROM pins help to avoid deselect timing violation (t_{CHSL} and $t_{CHSH)}$ when the controller releases the SPI bus. Refer to Section 4: Hardware considerations for safe recommendations. At deselect, EEPROM goes into the standby state (see Figure 44).
- However, a write cycle may be triggered if the EEPROM is deselected between two data bytes of a write instruction. On 4-Kbit or lower-density SPI devices, setting the $\overline{W}$ pin to low before deselecting the memory will prevent the write cycle execution.
- When the controller restarts, it must run a data polling sequence to check that the EEPROM is ready (and that no write cycle is in progress). Refer to Section 5.2: Optimal write control.
- As soon as the SPI device is ready, a WRDI instruction must be issued if the WEL bit in the status register is still set to 1. In so doing, the device is protected against any parasitic write instruction.

These recommendations maximize the control on the EEPROM in case of inadvertent controller reset. Recommendations for controller restart can also be the default sequence each time controller comes out of the reset state like after power-up.

Figure 44. SPI bus enters the high impedance state (controller reset)



Note: 1. A pull-down resistor on C prevents any t_{SHCH} timing violation (as a pull-up resistor on C causes C and $\overline{S}$ to rise at the same time, inducing t_{SHCH} = 0).

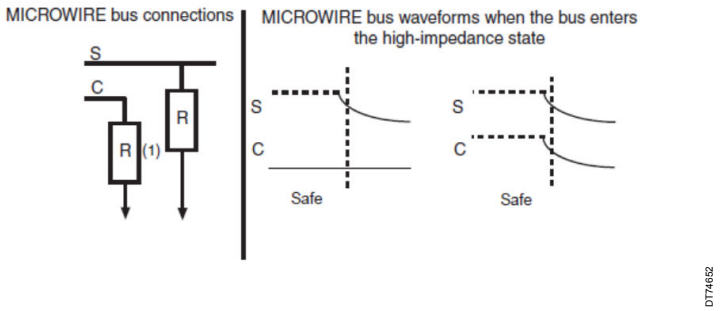
6.1.3 Microwire products

The main recommendation for these products is to deselect the device in a safe way. The chip must be deselected taking care of the timings of chip select respecting to the clock rising edge.

- Smart connections of EEPROM pins help to avoid deselect critical configurations when the controller releases the MICROWIRE bus. Refer to Section 4: Hardware considerations for safe connections. At deselect, EEPROM goes into the standby state (see Figure 45)
- However, when deselecting the EEPROM, a write cycle may be triggered if the EEPROM is deselected between two data bytes of a write instruction.
- When the controller restarts, it must run a data polling sequence to check that EEPROM is ready (no write cycle in progress).
- As soon as the device is ready, an Erase/Write disable (EWDS) instruction must be issued to disable any WRITE instruction. In this way, the device is protected against any parasitic WRITE instruction.

These recommendations maximize the control in case of inadvertent controller reset. Recommendations for controller restart can also be the default sequence each time controller comes out of the reset state like after power-up.

Figure 45. MICROWIRE bus enters the high-Z state (controller reset)



Note: 1. A pull-down resistor on C prevents any t_{SLCH} timing violation (as a pull-up resistor on C causes C to rise at the same time as $\overline{S}$ goes low, inducing t_{SLCH} = 0).

6.2 Power supply loss

Nonvolatile memory data integrity of is a key condition as applications rely on these stored values, mainly at system start-up power loss is critical for an EEPROM device when a write instruction is being issued or executed. In this event, the on-going write request or the internal write process in the EEPROM may not have been completed, leading to data corruption and data inconsistency.

6.2.1 Hardware recommendations

Application designers finds below some guidelines and recommendations to handle in the best conditions power supply loss on systems designed with STMicroelectronics EEPROM memories.

The power supply must designed in such a way that power loss is detected and a backup supply is supplied for a time allowing a safe emergency ending of the system operations. The list below gives some useful elements to build a robust power management system:

- Use voltage regulators including an output voltage sensor. It gives power loss information to the controller before the supply is too low for system operation.
- Use available MCU features such as the auxiliary voltage detector and external voltage detector pin to create a delay between the detection of the low voltage and the system reset.
- Use diodes, bipolar transistors, or analog switches to create specific areas with backup power capacitors.

The extra delay time gained should be used either to allow the EEPROM to complete any on-going write process or to allow the controller to finish or interrupt safely the current communication with the EEPROM. In a running application it is not possible to distinguish these two possibilities, therefore the below recommendations must be considered all together.

6.2.2 Supply voltage energy tank capacitor

In case of inadvertent power loss, applications are very often faced with the situation where the EEPROM is operating while the power supply is falling down.

It is not recommended to operate the device and in particular to initiate write operations when the device is undergoing steady V_{CC} transitions. ST EEPROM devices can however handle write cycles during smooth power supply transitions. A power supply transition is considered smooth when it allows a complete write cycle to be completed while V_{CC} is continuously falling or rising within the authorized V_{CC} range. Taking advantage of this possibility, a power backup capacitor can be designed to allow for the EEPROM to complete its on-going self-timed write operation in case of inadvertent power loss.

The capacitor value is calculated so as to allow for the full write cycle to be executed:

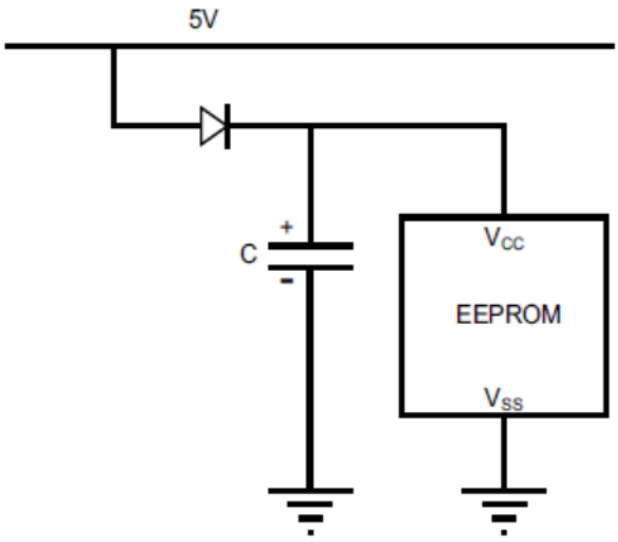
- I is the EEPROM supply current (I_{CC} max)
- t is the EEPROM write time (t_W)
- U is the voltage drop from the nominal value to V_{CC} min of the EEPROM

$Q = C \times U = I \times t \Rightarrow C = (I \times t) / U$

For instance:

$C = (3\ mA \times 5\ ms) / 2\ V = 7.5\ \mu F$

Figure 46. EEPROM power backup capacitor



Note: 1. *Sometimes filtering capacitors placed after voltage regulators, are big enough to allow the EEPROM device to finish the write operation. In this case, backup capacitors are no longer necessary.*

For a complete and detailed calculation, the discharge current through the MCU protection diodes must be taken into account. The EEPROM inputs/outputs do not draw any current during write operations and only the pull-up resistors connected to the EEPROM V_{CC} pin will discharge the backup capacitor through the MCU connection.

Optimum robustness is obtained by adding a discharge path to ground. At power-down, the EEPROM is usually in the standby mode, where it draws little current (no more than a few µA) and the backup capacitor takes a long time to discharge. The system designer must either:

- take into consideration the long discharge time to allow the EEPROM V_{CC} supply to reach ground level (0 V) before setting the application supply voltage active

or

- Add a discharge path to ground to accelerate the discharge if the system may or must be restarted after a short time

When applying this recommendation, read also Section 3.3.1: Power-up and power-on-reset sequence.

6.2.3 Interruption of an EEPROM request

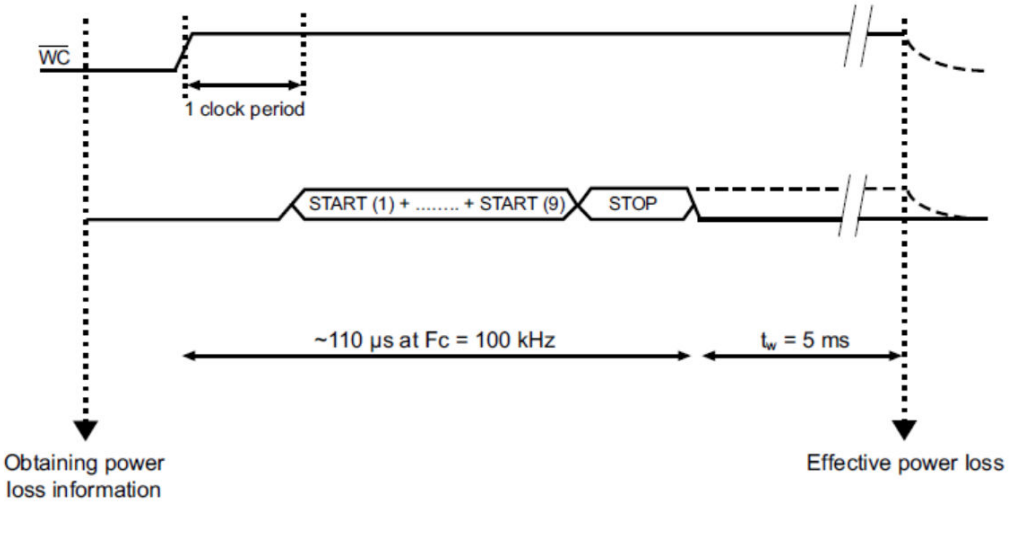
When the power loss occurs while the controller is still sending a command, it is strongly recommended to have an emergency software procedure able to safely interrupt the request being sent to the EEPROM. The set of recommendations below is adapted to each product family.

I²C products

Emergency procedure to interrupt an I²C request:

- Drive the $\overline{WC}$ pin high. One SCL clock pulse inhibits the current write request.
- Send a START condition followed by a STOP condition (a START condition resets the device and a STOP condition sets it in standby mode). The resynchronizations sequence described in AN1471 can also be used. See Figure 47.

Figure 47. Emergency sequence - I²C products



Note: This emergency sequence is detailed in the AN1471.

After the emergency sequence, the power supply of the EEPROM (through keep alive systems or a backup capacitor at the EEPROM level) should remain high enough to allow an eventual engaged write cycle to end correctly (t_W = 4 or 5 ms).

SPI products (1 to 4 Kbits)

Emergency procedure to interrupt an SPI request:

- Drive the $\overline{W}$ pin low while the device is selected. One clock pulse resets the WEL bit in the status register (any current write request is ignored).
- Deselect the EEPROM by driving the $\overline{S}$ pin high.

Warning: *If the SPI device is deselected between two data bytes of a write request and $\overline{W}$ has not been driven low, a write cycle may be triggered.*

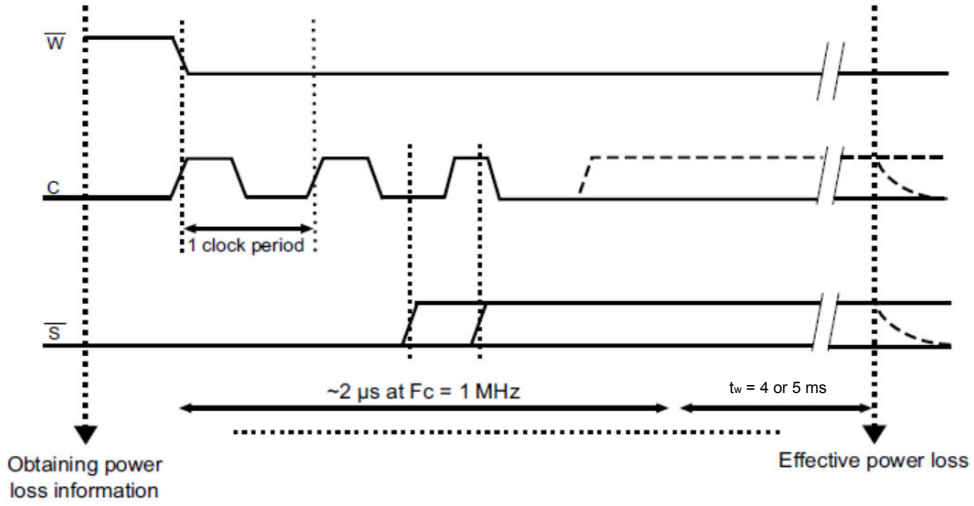
SPI products (8 Kbits and higher)

Emergency procedure to interrupt a SPI request:

- Drive $\overline{W}$ pin low while device is selected. Only the WRSR instruction is ignored, the execution of a write (to memory) is not discarded by the $\overline{W}$ pin for 8-Kbit products and higher.
- Deselect the EEPROM driving $\overline{S}$ pin high.

Warning: *If the chip is deselected between two data bytes of a write request, a write cycle may be triggered.*

Figure 48. Emergency sequence - SPI products



Note: If the emergency software sequence can be executed so that the EEPROM is not deselected at a data byte boundary (multiple of 8 bits), there is no risk of triggering a write cycle.

After the emergency sequence, the power supply of the EEPROM (through keep alive systems or a backup capacitor at the EEPROM level) must remain high enough to allow an eventual write cycle to complete correctly.

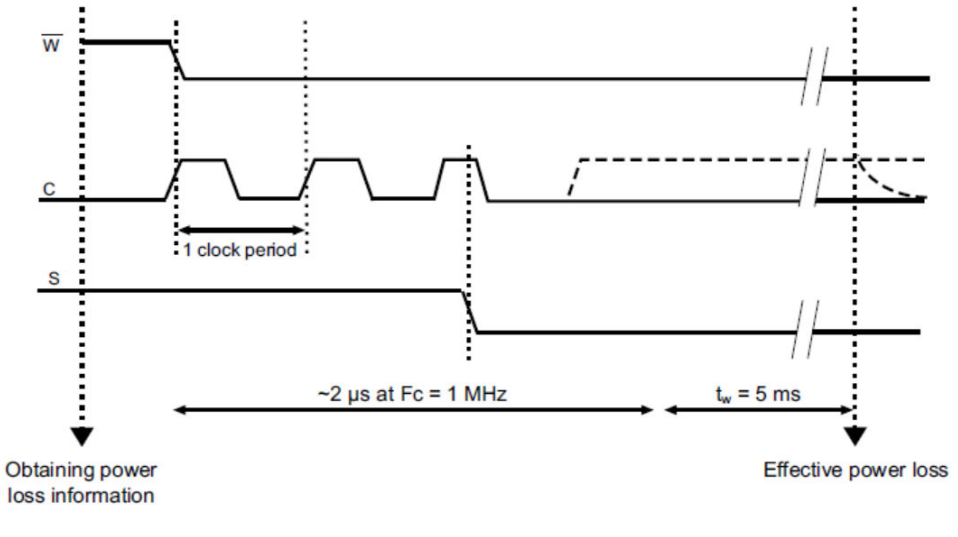
Microwire products

Emergency procedure to interrupt a Microwire request:

- Drive $\overline{W}$ pin low for M93Sxx devices(1.). One clock pulse discards the current write request.
- Deselect the EEPROM by driving the S pin low.

Warning: *If the chip is deselected between two data bytes of a write request and $\overline{W}$ has not been driven low, a write cycle may be triggered (see Figure 49. Emergency sequence - Microwire products).*

Figure 49. **Emergency sequence - Microwire products**



Note: If the emergency software sequence can be executed so that the EEPROM is not deselected at a data byte boundary (multiple of 8 bits), there is no risk of triggering a write cycle.

After the emergency sequence, the power supply of the EEPROM (through keep alive systems or a backup capacitor at the EEPROM level) must remain high enough to allow an ongoing write cycle to be correctly completed.

Note: 1. $\overline{W}$ pin is available on M93Sxx products (not available on M93Cxx products).

6.3 Robust software and default operating mode

In sensitive applications such as automotive, safety or medical applications, it is not acceptable for a system to enter a locked state or an endless loop, because of bad EEPROM communications. In many cases, simple software rules can help to secure the operation of the application.

- WRITE: It is recommended to add a time-out counter to the write data polling loop to prevent the application from being locked if, for some unknown reason, the application software cannot exit from an endless write data polling loop. After each write cycle, the software should always verify that the data has been correctly programmed by reading back the data.
- READ: When reading data from the EEPROM, the application software should check whether the data value is within an acceptable range and, if not, switch to a default value allowing continuity in the application operation.
- As recommended, data in the EEPROM can be duplicated and associated with a checksum and an error code correction mechanism. In particular default parameters can be stored in a protected part of the memory array (Read-Only array, the write-lock being defined by software) or in another available nonvolatile memory. The MCU should then be able to access an external memory to copy the missing parameters back to the EEPROM.
- Moreover, it is safer to have a default operating mode that can run with a reduced set of default parameters.

Refer also to Section 5.3: Write protection and Section 5.4: Data integrity.

7 Operating conditions

There are many other operating conditions, imposed by the final application environment, that may also have an adverse affect on the EEPROM device (shortened lifetime or unreliable operation). They should be studied, and solutions must be found to minimize them.

7.1 Temperature

The temperature should be kept as low as possible, since high temperatures accelerate wear-out. At high temperatures, cycling endurance and data retention capability are reduced because of charge trapping in the thin oxide of the memory cells. When applications are designed to run in hot environments with high cycling requirements, it is strongly recommended to establish a temperature profile and discuss it with the ST EEPROM quality support (refer to Section 5.5: Cycling endurance and data retention).

7.2 Humidity and chemical vapors

Boards should always operate in a clean and dry environment. Humidity and dirt of any kind can cause corrosion and short circuits between package pins and tracks.

7.3 Mechanical stress

EEPROM packages cannot withstand excessive weight, local pressure or strong shocks.

8 Conclusions

Electrically erasable and programmable memory (EEPROM) devices are standard products used for the nonvolatile storage of parameters, with fine-granularity data.

There is no single memory technology (SRAM, DRAM, EEPROM, flash memory, EPROM, ROM) that meets all application needs perfectly. In the case of an EEPROM, an application designer needs to know the particular strengths and weaknesses of the device technology and device architecture to define an optimal control of parameters for his application. In doing so, the application remains within the specification, with the best performance and reliability level.

9 References

- AN1471: *What happens to the M24xxx I²C EEPROM if the I²C bus communication is stopped?*
- AN2440: *Embedded ECC in F8H process automotive EEPROM: device architecture and related application guidelines*

Revision history

Table 10. **Document revision history**

Date	Version	Changes
28-Oct-2005	1	Initial release.
15-May-2006	2	Section 1.1.3: Cycling limit of EEPROM cells, Section 5.2.1: Page mode, Section 5.4.1: The checksum, Section 5.5: Cycling endurance and data retention, Section 6.1: Application reset, Section 6.2: Power supply loss modified. Small text changes.
26-Oct-2006	3	Small text changes. Figure 14: Power-up and Figure 39 modified (paragraph added to explain Figure 39). Table 5: Calculation rules for pullup resistor on SDA and Note 1 modified. Note modified below Figure 22. Section 5.4: Data integrity modified. Section 6.2.2: Supply voltage energy tank capacitor modified. “General recommendation applying to all EEPROM products” paragraph removed and content transferred to Section 3.3.1 on page 21.
15-Jan-2007	4	Small text changes. In the SPI family the S pin must remain above VIH = 0.7VCC during powerup (see Section 4.2.1: Chip Select (S)) and Write Protect (W) behavior specified. Pull-up and pull-down resistances discussed in Section 4.2.3: Serial Data input (D) and Serial Clock (C) (SPI) and Section 4.3.2: Serial Data (D) and Serial Clock (C) (MICROWIRE). Pull-down resistor value on C modified and pull-down resistor added to D line in Figure 27: Recommended SPI connections - robust design and Figure 31: Recommended MICROWIRE connections - robust design. Maximum C2 value modified in Figure 32: PCB decoupling.
26-Jun-2008	5	Small text changes. Section 3.3.2: Stabilized power supply voltage, Section 3.3.1: Power-up and power-on-reset sequence and Section 4.1.5: Recommended I2C EEPROM connections clarified. Section 4.4.3: Communication lines removed. Section 6.1: Application reset modified, Figure 40: I2C bus enters the high impedance state (Master reset), Figure 41: SPI bus enters the high impedance state (Master reset) and Figure 42: MICROWIRE bus enters the high impedance state (Master reset) modified.
04-Feb-2010	6	Removed reference to application note AN1001 in Section 2.2: Choosing an appropriate memory interface. Updated POR threshold in Table 3: Typical POR threshold values. Updated programming time in Section 5.1: EEPROM electrical parameters. Updated Section 5.5.1: Cycling and data retention qualification procedures and Section 5.5.3: Cycling and temperature dependence. Added Section 5.5.2: Optimal cycling with ECC.
01-Apr-2011	7	Updated: • Section 4.1.5: Recommended I2C EEPROM connections • Section 4.2.1: Chip Select (S) • Section 4.2.2: Write Protect (W) • Section 4.2.3: Serial Data input (D) and Serial Clock (C) • Section 4.2.4: Hold (HOLD) • Section 4.2.6: Recommended SPI EEPROM connections • Figure 30: Recommended MICROWIRE connections - safe design • Section 5.5.1: Cycling and data retention qualification procedures • Added Section 5.5.5: Overall number of write cycles
15-Mar-2012	8	Updated Figure 2: MOSFET-like operation. Updated Section 1.1.2: Writing a new value to the memory cell. Renamed Figure 8: VPP signal applied to EEPROM cells (HiV is the output of the charge pump). Updated Section 1.1.3: Cycling limit of EEPROM cells. Updated Section 1.2.1: Memory array architecture. Updated Table 1: Three serial bus protocols.

Date	Version	Changes
15-Mar-2012	8 cont'd	Updated Section 2.3: Choosing an appropriate supply voltage and temperature range. Updated Section 3.1.2: How to prevent ESD? and Section 3.1.3: ST EEPROM ESD protection. Updated Section 3.2.1: What are EOS and latchup? and Section 3.2.2: How to prevent EOS and latchup events, and Section 3.2.3: ST EEPROM latchup protection. Section 4.1: I2C family (M24xxx devices): updated Section 4.1.1: Chip enable (E0, E1, E2) and Figure 16: Chip Enable inputs E0, E1, E2, Section 4.1.2: Serial data (SDA), Section 4.1.4: Write control (WC), Table 6: Connecting WC inputs in I2C products, Section 4.1.5: Recommended I2C EEPROM connections. Section 4.2: SPI family (M95xxx devices): updated Section 4.2.1: Chip Select (S), Section 4.2.3: Serial Data input (D) and Serial Clock (C), Section 4.2.5: Serial Data output (Q), Figure 26: Recommended SPI connections - safe design and Figure 27: Recommended SPI connections - robust design. Section 4.3: MICROWIRE® family (M93Cxxx and M93Sxxx devices): updated Section 4.3.1: Chip Select (S), Section 4.3.3: Organization Select (ORG), Figure 30: Recommended MICROWIRE connections - safe design, and Figure 31: Recommended MICROWIRE connections - robust design. Updated Section 5.1: EEPROM electrical parameters, Section 5.2.1: Page mode. Updated Section 5.5.1: Cycling and data retention qualification procedures, Section 5.5.2: Optimal cycling with ECC, Section 5.5.3: Cycling and temperature dependence, Section 5.5.4: Defining the application cycling strategy, and Section 5.5.5: Overall number of write cycles. Updated Section 6.1: Application reset.
31-Mar-2014	9	Modified Introduction and Conclusions. Updated clock rate data in Table 1. Text updated throughout the document, but more specifically: • Chapter 2 • Section 3.1.3: ST EEPROM ESD protection • Section 3.3.1: Power-up and power-on-reset sequence • Section 4.1.1: Chip enable (E0, E1, E2) • Section 4.1.2: Serial data (SDA) • Section 4.1.3: Serial clock (SCL) • Section 4.1.4: Write control (WC) • Section 4.1.5: Recommended I2C EEPROM connections • Section 4.2.2: Write Protect (W) • Section 4.2.3: Serial Data input (D) and Serial Clock (C) • Section 4.2.5: Serial Data output (Q) • Section 5.4.2: Data redundancy • Section 5.5: Cycling endurance and data retention • Section 5.5.1: Cycling and data retention qualification procedures • Section 5.5.2: Optimal cycling with ECC • Section 5.5.3: Cycling and temperature dependence • Section 5.5.5: Overall number of write cycles • Section 6.3: Robust software and default operating mode Changed Figure 39: Write cycling versus temperature. Added specific values relating to process letter K.
17-Feb-2015	10	Updated Section 4.1.3: Serial clock (SCL), Section 5.3.1: Software write protection and Section 5.5.4: Defining the application cycling strategy. Updated Note below Figure 10 and Footnote 1 of Figure 29. Modified Figures 21, 22, 25, 32, 39, 44, 45 and 46. Other text updates throughout the document, without any context change.
01-Oct-2024	11	Updated document title, Introduction, Section 1.2.3: Intrinsic electrical stress induced by programming, Section 2.2: Choosing the appropriate memory interface, Section 3.3.1: Power-up and power-on-reset sequence, Section 4.3.1: Chip select (S), Section 4.3.2: Serial data (D) and serial clock (C), Section 5.1: Electrical parameters, Section 5.2.1: Page mode, Section 5.3.1: Software write protection, Section 5.3.2: Hardware write protection, and Section 5.5.3: Cycling and temperature dependence.

Date	Version	Changes
01-Oct-2024	11 cont'd	Updated Table 1. Serial bus protocols, Table 3. Typical POR threshold values, and Table 8. Column and page address bits according to page length. Updated Figure 6. Erase operation, Figure 7. Write operation, Figure 17. Serial data input/output SDA, Figure 21. Recommended I²C connections - Safe design, Figure 22. Recommended I²C connections - Robust design, Figure 36. Recommended use of the $\overline{WC}$ pin - I²C products, Figure 37. Recommended use of the $\overline{W}$ pin - SPI products, Figure 49. Emergency sequence - Microwire products, and figures in Section 5.5.3. Added note to Section 4.1.1: Chip enable (E0, E1, E2) and to Section 5.5.1: Cycling and data retention qualification procedures. Minor text edits across the whole document.

Contents

List of tables

List of figures