Timing of erase, program, and write operations for page EEPROMs

Introduction

This document concerns the products listed in Table 1, and describes the erase, program, and write algorithms of page EEPROM devices. It provides useful informations to efficiently manage the timings of these operations, depending upon the executed algorithm.

Table 1. Applicable products

Series	Root part number
Standard Serial Page EEPROM	M95P32-E M95P32-I
	M95P16-E M95P16-I
	M95P08-E M95P08-I

1 Page EEPROM erase definition

1.1 Erase operation

An erase sets all bits to "1" in a specific area of the memory. There are four different erase operations:

- Page erase (one page, 512 bytes)
- Sector erase (eight pages, 4 Kbytes)
- Block erase (128 pages, 64 Kbytes)
- Chip erase (whole memory)

If a part of the targeted area is protected, the erase operation is not performed. The erase is carried out anyway on already erased areas.

1.2 Erase steps

The timing of this operation is impacted by three different steps:

1. **Page refresh**
   The objective is to reprogram the highly cycled cells affected by charge loss. This step is performed after each counter erase increment.
2. **Counter erase**
   The architecture of the page EEPROM incorporates data memory units (DMU). Each DMU contains a page dedicated to counter erase. After all erase operations (except chip erase), the counter is incremented, and a page refresh operation is executed. The refreshed page is different after each counter increment. When the counter reaches 1024, it is erased with a page erase operation, and then it restarts the count.
3. **Erase pulse**
   To set a bit to "1", a voltage (called pulse) is applied to the memory cell. If a first pulse is not sufficient to erase the cell, a second pulse is applied with a higher voltage value. The last pulse value is stored in a dedicated page and becomes the reference for the next erase operation. Only high cycled cells can trigger a new occurrence of a second pulse.

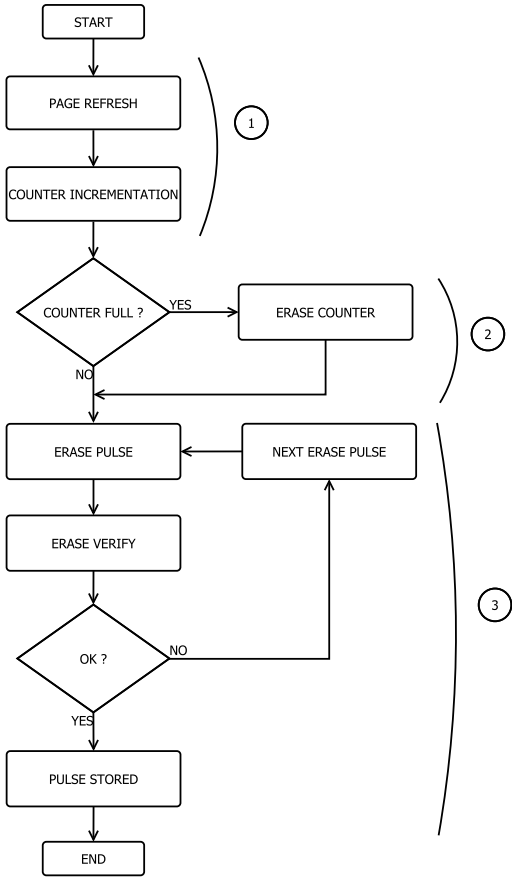
All these steps are part of the erase algorithm and improve the robustness of the products.

2 Page EEPROM erase timing

2.1 Erase algorithm

The different steps of the erase algorithm are shown in Figure 1. All these steps are available for page, sector, and block erases.

Figure 1. Erase algorithm



Referring to Figure 1:

1. After the erase operation request, one page in the DMU is refreshed, and the erase counter is incremented.
2. When the counter reaches 1024 increments, it is erased with a page erase operation.
3. A pulse is applied to the cells. If not high enough, an additional pulse is generated.

Chip erase operation is impacted only by step 3.

Note: As the erase counter is a page erase operation, the erase counter step can follow the step 3 and generate an additional pulse.

Different sequences can occur:

- Page refresh + erase pulse x1
- Page refresh + counter erase pulse x1 + erase pulse x1
- Page refresh + erase pulse x2
- Page refresh + counter erase pulse x2 + erase pulse x2
- Page refresh + counter erase pulse x2 + erase pulse x3

These sequences define the erase timing. The erase counter (step 2) and the next erase pulse (step 3) are not executed for each erase operation. The frequency of their execution explains the gap between the typical and the maximum erase timing values specified in the datasheets.

2.2 Erase timing

Erase time depends upon the algorithm being executed (page refresh, counter erase, double pulse erase). The gap between typical and maximum timing values for page, sector, or block erase operations is explained in Table 2.

Table 2. Erase steps frequencies

Algorithm	Frequency	Comment
Page refresh + erase pulse x1	99.9%	Typical flow
Page refresh + counter erase pulse x1 + erase pulse x1	0.1%	1 over 1024
Page refresh + erase pulse x2	0.01%	1 over 10 000
Page refresh + counter erase pulse x2 + erase pulse x2	1 per 1 000 000	Very low probability
Page refresh + counter erase pulse x2 + erase pulse x3	1 per 10 000 000	Extremely low probability

Table 3 shows an example of these probabilities applied to M95P32 devices, for the page erase operation.

Table 3. Page erase timing for M95P32 devices

Page erase algorithm	Frequency	Timing (ms)
Page refresh + page erase pulse x1	99.9%	1.1(1)
Page refresh + counter erase pulse x1 + page erase pulse x1	0.1%	1.6
Page refresh + page erase pulse x2	0.01%	1.6
Page refresh + counter erase pulse x2 + page erase pulse x2	1 per 1 000 000	3.5
Page refresh + counter erase pulse x2 + page erase pulse x3	1 per 10 000 000	4.5(2)

1. *Datasheet typical value.*
2. *Datasheet max value.*

- The page erase typical time of 1.1 ms occurs in 99.9% of the cases.
- One page erase over 1024 is 1.6 ms long, due to counter erasing.
- One page erase over approximately 10 000 is 1.6 ms long, due to a double erase pulse on cycled cells. The voltage value of this second pulse is stored and directly applied for the following erase operations, to keep on with the typical erase timing.
- Longer page erase times occur sporadically, and only if the device is intensively cycled. Even if the maximum values are indicated in the datasheet, their occurrence is unlikely in most of the applications.

Sector and blocked erase operations have the same algorithms and frequencies, and are as stable as the page erase operation detailed in Figure 1.

Chip erase

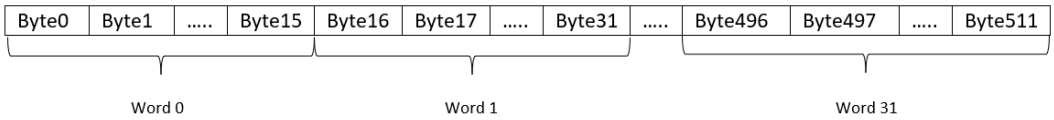
Chip erase operation generates a high voltage to erase all the memory cells in a single shot. If one or more addresses are protected, the chip erase operation is automatically discarded. This limits the chip erase endurance to 100 cycles, skipping the erase algorithm (illustrated in Figure 1), with a typical time of 15 ms (maximum 25 ms).

3 Page EEPROM program definition

3.1 Program operation

Page EEPROM has pages of 512 bytes, and each page has 32 words of 16 bytes, as shown in Figure 2.

Figure 2. Page architecture



The programming operation allows 1 to 512 bytes of data initially in the erased state (FFh) to be programmed to 0. To program a byte within a word, the entire word must be erased. If a page program operation includes two bytes in two successive words, the two words must be in the erased state.

3.2 Program steps

Page program timing is determined by three different steps:

1. **Check erased state**
   This step ensures that the requested memory area is in an erased state (FFh). If not, the page program operation is automatically discarded.
2. **Memory cell programming**
   A voltage pulse is applied to the memory, it can program a maximum of 48 bits. The number of pulses ranges between 1 (48 bits, or 6 bytes) and 86 (4 Kbits, or 512 bytes). As a single pulse can be not enough for high cycled cells, the programmed words are checked later.
3. **Programming check**
   To ensure that a word has been programmed correctly, its content is compared with the SPI buffer. If the comparison shows at least one difference, the word is reprogrammed with a higher voltage. Each word is checked if the page differs from the contents of the SPI buffer.

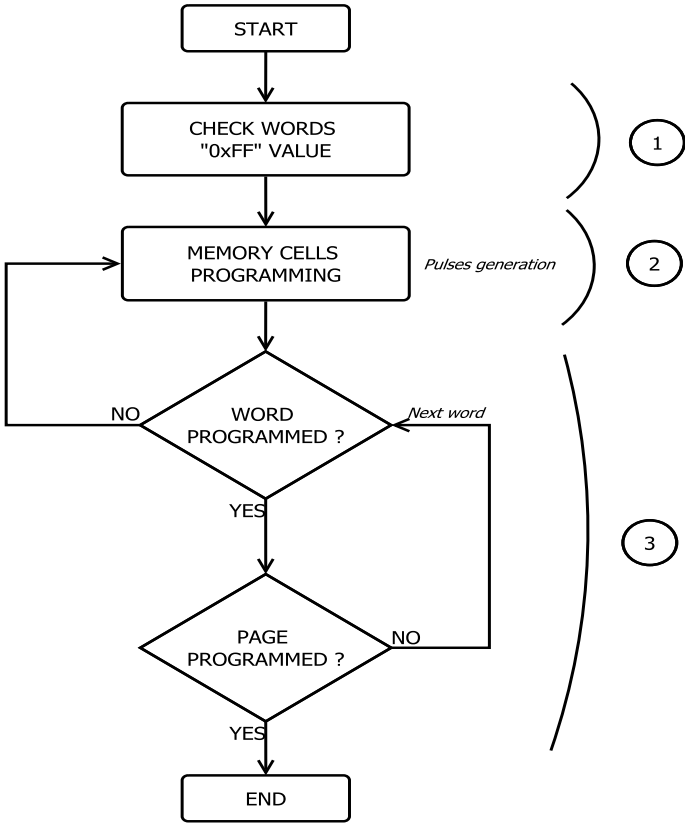
All of these steps prevent possible page program errors.

4 Page EEPROM program timing

4.1 Program algorithm

The page program algorithm is sketched in Figure 3.

Figure 3. Page program algorithm



Referring to Figure 3:

1. SPI buffer is loaded with value(s) to be programmed. The targeted word(s) is(are) read in the memory array to check the erased state. If not, the page program is discarded.
2. Voltage pulses are generated to program memory cells (several pulses can be applied).
3. The previously programmed word is compared with the contents of the SPI buffer. If the comparison shows differences, the algorithm goes back to step 2 to program the unprogrammed bits with a higher pulse level. Then, the next word is checked, as long as the page differs from the contents of the SPI buffer.

The return cycle of step 2 is limited. After a certain number of incremental pulses, the page program is aborted and the PRF bit in the safety register is set to 1.

The page program algorithm prevents unprogrammed bits. Its execution time depends upon the amount of data to be programmed and the number of pulses generated to program targeted words.

4.2 Program timing

Page program timing value depends upon the three steps shown in Figure 3. To program one page, step 1 is executed once, so the time needed for programming depends upon step 2 and step 3, whose execution depends upon the amount of data to program, and how many of targeted cells are cycled.

The typical 1.2 ms value is related to 512 bytes to be programmed. The maximum value of 1.5 ms is reached only if the targeted cells have been heavily cycled. If the amount of data is less than 512 bytes, the typical timing value is less than 1.2 ms, as illustrated in Table 4.

Table 4. Page program typical time

Programmed bytes	Time (µs)
1	100
2	100
4	100
8	117
16	134
32	167
64	234
128	368
256	637
512	1175

Below 6 programmed bytes, the page program time value is 100 µs. This is the minimum programming time to start the algorithm and the different analog signals. The programming time increases linearly with the amount of data with the following equations:

Page program time = 100 µs

$1 \leq n \leq 6$

Page program time (µs) = 2.1*n* + 100

$n > 6$

Where *n* is the number of bytes to be programmed.

Note: These equations are for information only. They give a typical estimation time value for the page program operation.

5 Page EEPROM write timing

Page write operation is a page erase followed by a page program operation. This operation (managed internally by the device) writes from 1 to 512 bytes of data in a single instruction. When a page write operation is performed, the page is first erased, then written (updating the bits to be modified, and not the ones to keep unchanged).

Page erase and page program operations performed during the page write operation use the same algorithms described in, respectively, Figure 1 and Figure 3. Page write operation is as stable and robust as the page erase and page program operations.

6 Conclusion

Page EEPROM products prevent charge loss, possible failures, and potential unsuccessful programing. This robustness is achieved thanks to dedicated algorithms during erase and program operations. The values of the typical and maximum erasing and program times are defined according to these algorithms.

The erase algorithm makes the erase time very stable throughout the device lifetime. For instance, the page erase has a very short-typical timing at 1.1 ms with very few occurrences at 1.6 ms. In case of intensive cycling, page erase can reach a maximum of 4.5 ms before returning to typical time at the next operation. Sector, block, and chip erase operations are as stable as the page erase example illustrated above.

The page program operation (512 bytes) has small variations over the device lifetime, with a typical timing at 1.2 ms, and very rare instances at 1.5 ms. Moreover, the page program algorithm ensures that each data has been programmed correctly checking after each word programmed.

The page write operation is a combination of page erase and page program operations, hence it features the same benefits. For example, in term of timing, page write is very stable with a typical value of 2.3 ms, and very rare occurrences at 6 ms.

Monitoring these timings by polling the status register and the write in progress (WIP) bit makes possible to get the typical timing in 99% of the cases over the device lifetime, and to manage the few occurrences with higher values.

Revision history

Table 5. **Document revision history**

Date	Version	Changes
07-Feb-2023	1	Initial release.

Contents

1 **Page EEPROM erase definition** 2
1.1 Erase operation 2
1.2 Erase steps 2
2 **Page EEPROM erase timing** 3
2.1 Erase algorithm 3
2.2 Erase timing 4
3 **Page EEPROM program definition** 5
3.1 Program operation 5
3.2 Program steps 5
4 **Page EEPROM program timing** 6
4.1 Program algorithm 6
4.2 Program timing 6
5 **Page EEPROM write timing** 8
6 **Conclusion** 9
Revision history 10
List of tables 12
List of figures 13

List of tables

Table 1. Applicable products . . . 1
Table 2. Erase steps frequencies . . . 4
Table 3. Page erase timing for M95P32 devices . . . 4
Table 4. Page program typical time . . . 7
Table 5. Document revision history . . . 10

List of figures

Figure 1. Erase algorithm . . . 3
Figure 2. Page architecture . . . 5
Figure 3. Page program algorithm . . . 6