



**Please note that Cypress is an Infineon Technologies Company.**

The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

**Continuity of document content**

The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

**Continuity of ordering part numbers**

Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

# WHITEPAPER

**Shivendra Singh**  
**Applications Engineer Principal**  
Cypress Semiconductor Corp.



## **Probability of Occurrence of a Soft Sequence in Asynchronous nvSRAMs**

### **Abstract**

Cypress's nonvolatile static RAM (nvSRAM) combines the features of a fast access asynchronous interface SRAM with reliable non-volatility. In addition to the standard SRAM accesses, the nvSRAM offers the other additional features, such as software STORE, software RECALL, AutoStore Enable, and AutoStore Disable. These features are executed by issuing specific soft sequences, which are initiated by reading six specified address locations in a specific order.

This whitepaper highlights the different soft sequences of nvSRAMs and the probability that a soft sequence will occur during normal operation.

### **nvSRAM Soft Sequences: An Overview**

Cypress's nvSRAM supports specific nonvolatile operations, including software STORE, software RECALL, AutoStore Enable, and AutoStore Disable. These soft sequences are initiated by performing a sequence of SRAM read operations from six specified address locations.

The normal SRAM read in nvSRAM is address-controlled, chip enable ( $\overline{CE}$ )-controlled, or output enable ( $\overline{OE}$ )-controlled. In contrast, the read required for software sequences must be either  $\overline{CE}$  or  $\overline{OE}$  controlled. The address-controlled read will not initiate the soft sequence. For software sequences, the address is set when  $\overline{CE}$  and/or  $\overline{OE}$  are HIGH and after the address bits have settled.  $\overline{CE}$  and/or  $\overline{OE}$  must go LOW. It does not matter whether  $\overline{CE}$  or  $\overline{OE}$  goes LOW first as long as the required address setup time ( $t_{SA}$ ), address hold time ( $t_{HA}$ ), and clock pulse width ( $t_{CW}$ ) are met. The write enable ( $\overline{WE}$ ) must remain HIGH during all of the six consecutive read operations. The various soft sequences of nvSRAM are shown in [Table 1](#).

This whitepaper discusses the total address sequences reserved to initiate a soft sequence in the nvSRAM and the probability that a soft sequence will occur during normal read operation from the nvSRAM.

Soft Sequences	nvSRAM Density	
	8Mbit, 4Mbit, 1Mbit	256Kbit
	Address Range	
	A15-A0	A14-A0
AutoSTORE Disable	0x4E38	0x0E38
	0xB1C7	0x31C7
	0x83E0	0x03E0
	0x7C1F	0x3C1F
	0x703F	0x303F
	0x8B45	0x0B45
AutoSTORE Enable	0x4E38	0x0E38
	0xB1C7	0x31C7
	0x83E0	0x03E0
	0x7C1F	0x3C1F
	0x703F	0x303F
	0x4B46	0x0B46
Software STORE to Nonvolatile	0x4E38	0x0E38
	0xB1C7	0x31C7
	0x83E0	0x03E0
	0x7C1F	0x3C1F
	0x703F	0x303F
	0x8FC0	0x0FC0
Software RECALL from Nonvolatile	0x4E38	0x0E38
	0xB1C7	0x31C7
	0x83E0	0x03E0
	0x7C1F	0x3C1F
	0x703F	0x303F
	0x4C63	0x0C63

Table 1. Soft Sequences for Nonvolatile Cycles

**Notes:**

1. Table 1 does not include a comprehensive list of all nvSRAM variants. Refer to the device datasheet for address sequences.
2. The six consecutive addresses must be read in the order listed in Table 1. The  $\overline{WE}$  must be HIGH during all six consecutive read cycles.
3. The data output data at the sixth read may be invalid because the output is disabled at  $t_{DELAY}$  time.
4. Only 13 address lines A[14:2] are used to control the software sequence; the remaining address lines are treated as “don’t care.”

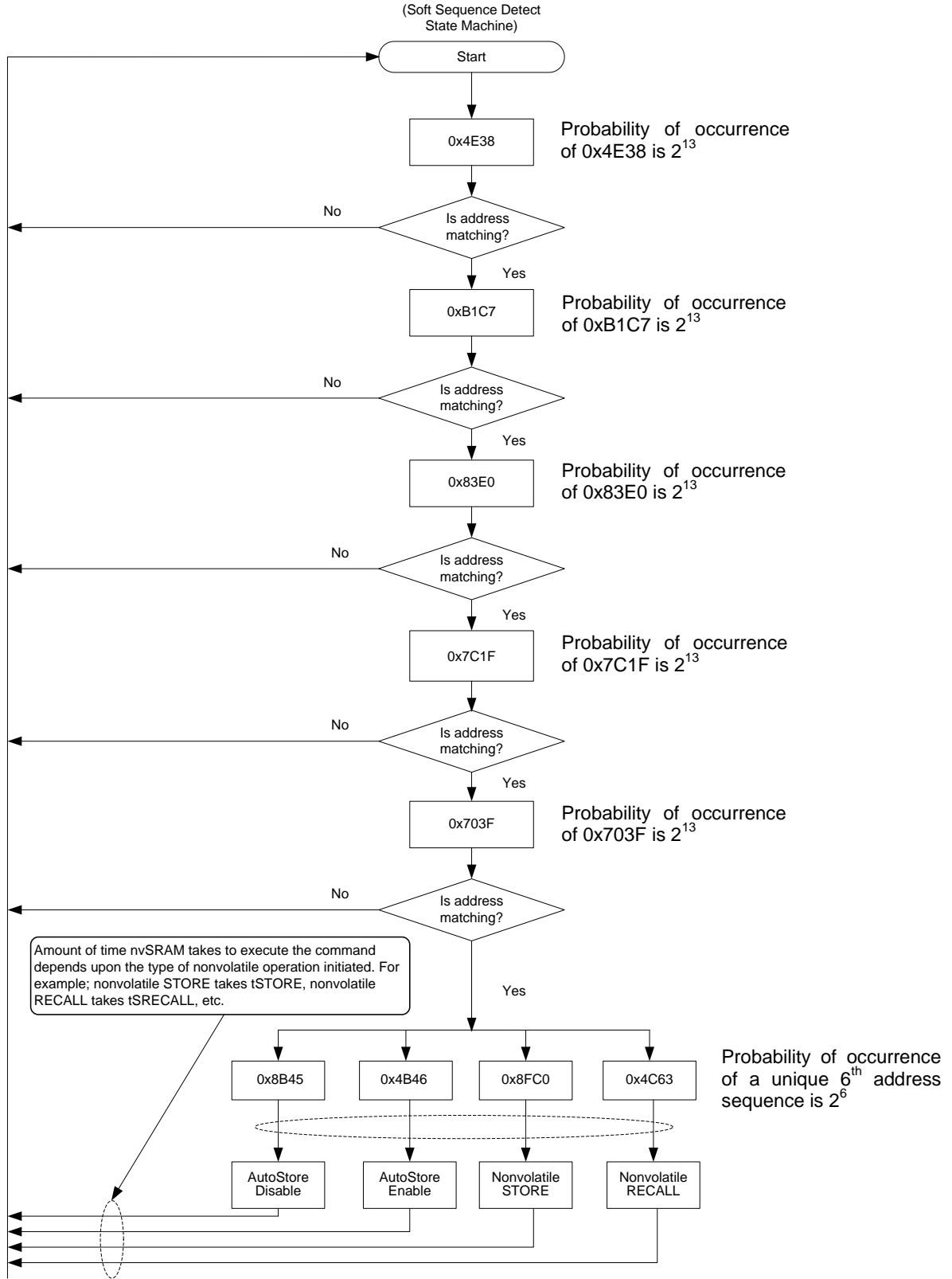


Figure 1. nvSRAM Soft Sequence Flowchart for 1Mbit and Above Density

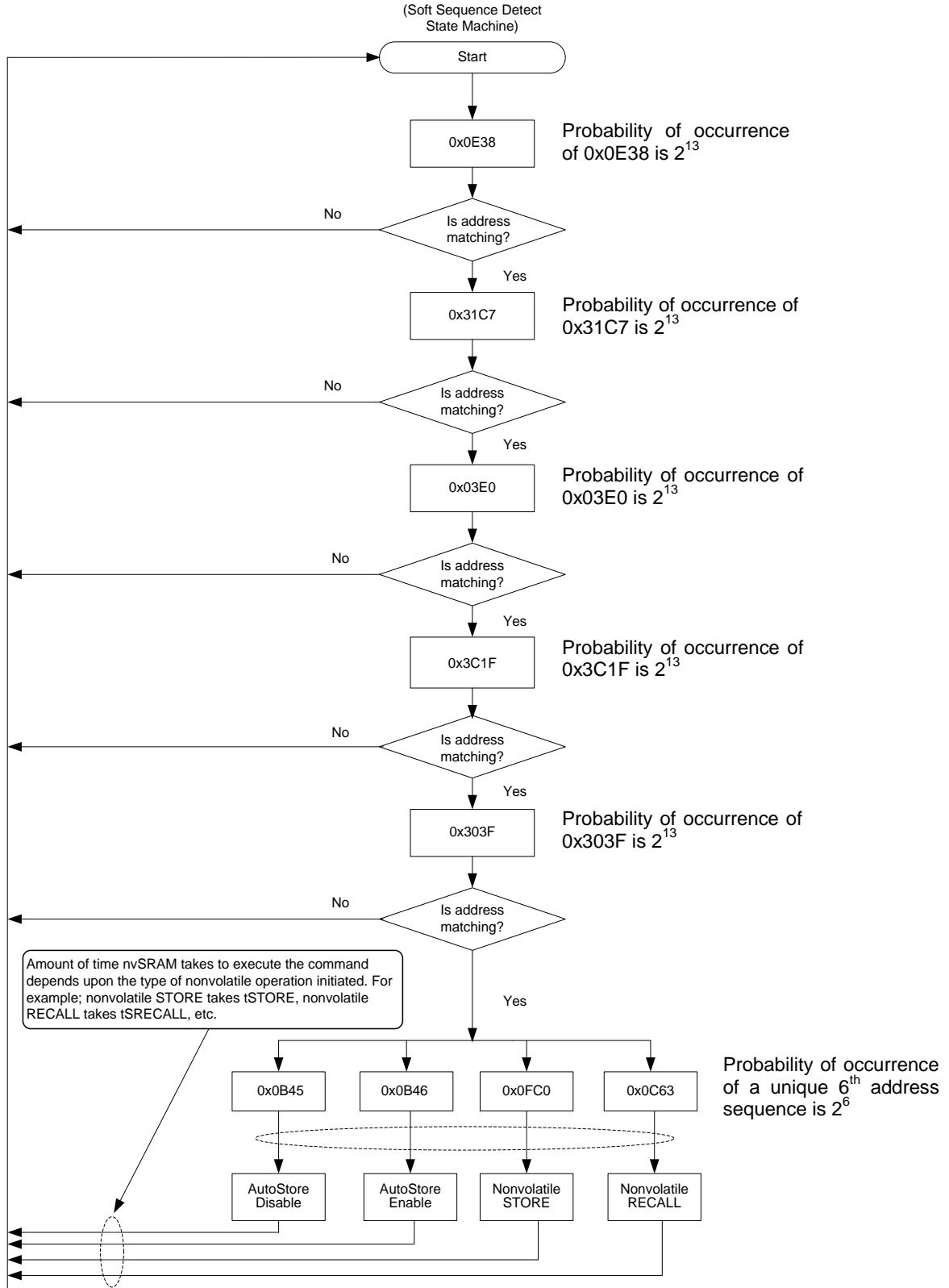


Figure 2. nvSRAM Soft Sequence Flowchart for 256Kbit Density

## Probability Calculation

The nvSRAM soft sequences are triggered by reading six fixed address locations in a specific order by using only 13 lower bits A [14:2] of its address lines. The other address bits other than A[14:2] are “don’t care” and, therefore, they are ignored by the device internally for determining a soft sequence.

The soft sequence detect state machine in nvSRAM validates all 13 address bits A[14:2] for the first five reads in a soft sequence, whereas the 6<sup>th</sup> read is checked for only 6 of 13 address bits to trigger one of the four soft sequences. Therefore, the total number of possible unique address sequences generated by toggling 13 address lines six times is calculated as follows:

$$2^{13} \times 2^{13} \times 2^{13} \times 2^{13} \times 2^{13} \times 2^6 = 2^{71} \text{ or } 2.361 \times 10^{21}.$$

Based on the flowcharts shown in [Figure 1](#) and [Figure 2](#), only four out of a possible  $2.361 \times 10^{21}$  address sequences can initiate a soft sequence. Therefore, the probability that a soft sequence will occur during normal read is 4 out of  $2.361 \times 10^{21}$  or 1 out of  $5.903 \times 10^{20}$  read operations.

To further explain, when the device performs continuous read operations with 25-ns (40-MHz) access speed, the above probability equates to striking an unwanted soft sequence after every 467,954 years.

$$5.903 \times 10^{20} / (40 \times 10^6 \times 60 \times 60 \times 24 \times 365) = 4.68 \times 10^5 \text{ years.}$$

## Preventing an Unwanted Soft Sequence

According to the above calculation, the probability of striking an unwanted soft sequence in the nvSRAM is very low. Therefore, it is almost impossible for this to happen in a user application even once during a product’s life cycle.

The address-controlled read operation doesn’t trigger the soft sequence. Therefore, using an address-controlled read in an application will further reduce the probability of an unwanted soft sequence occurring.

Another approach would be to mask the access of one of the first five addresses used in initiating a soft sequence. Some software management in the memory controller is required for this approach.

**Note:** When a soft sequence triggers in the nvSRAM, the internal state machine generates an interrupt and completes any ongoing access within the  $t_{\text{DELAY}}$  (25 ns). The device takes  $t_{\text{SS}}$  time (soft sequence processing time) of 100  $\mu\text{s}$  to process the soft sequence first to determine which specific function it has to execute. After detecting a soft sequence, the nvSRAM executes the associated operation and available for any further access only after the current operation is over. The nvSRAM disables all of its I/Os during  $t_{\text{SS}}$  time and keeps them disabled until the nvSRAM completes its operation.

## Summary

The soft sequence in the nvSRAM enables a few additional features unique to the device but widely used in applications in which on-demand nonvolatile STORE or RECALL is preferred. These soft sequences are triggered by reading from six fixed memory locations in a specified order. Even through soft sequences are executed merely by reading six address locations, the probability of triggering an unwanted soft sequence is extremely low. In fact, it is almost impossible for an unwanted soft sequence to occur during a product’s life cycle.

Cypress Semiconductor  
198 Champion Court  
San Jose, CA 95134-1709  
Phone: 408-943-2600  
Fax: 408-943-4730  
<http://www.cypress.com>

© Cypress Semiconductor Corporation, 2013-2016. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit [cypress.com](http://www.cypress.com). Other names and brands may be claimed as property of their respective owners.