AN68174

Migrating from Serial Peripheral Interface (SPI) EEPROM to SPI nvSRAM

Author: Shivendra Singh

Associated Part Family: CY14xxxxPA, CY14xxxxQxA

Related Application Notes: AN64574

To get the latest version of this application note, please visit http://www.cypress.com/go/AN68174.

This application note highlights the benefits of using SPI nvSRAM in an application and provides design guidelines to migrate from an SPI EEPROM based design to an SPI nvSRAM based design.

Contents

1 Introduction

Serial SPI EEPROM devices are frequently used for the nonvolatile storage of data. However, their low nonvolatile write speed and limited endurance causes bottlenecks when used in designs that need frequent updates into the nonvolatile memory at bus speed. Many system designers have tried to solve the endurance problems by using wear-leveling techniques that increase effective endurance cycles but at the cost of increasing EEPROM density and software overheads. Other designers have adopted the buffered memory architecture in which the system saves data only at power down or in power fail conditions using a reliable backup power source. Both of these approaches have been proven to be expensive because of the increased BOM cost, board area, processor I/O usage, design complexity, and software overhead.

Cypress SPI nvSRAM solution solves these problems by providing an easy-to-use, fast random access SRAM memory cells. With the Cypress solution, you can write the entire memory array serially at speed without incurring any additional nonvolatile write delay and also provide infinite read/write endurance cycle. SPI nvSRAMs are available in a pin configuration compatible with the standard serial SPI EEPROM devices.

This application note provides design guidelines for migrating from SPI EEPROM to SPI nvSRAM. It describes key benefits of SPI nvSRAM over EEPROM and provides example circuits for designing with various nvSRAM package options. The application note also discusses nvSRAM device features and the associated differences from EEPROM in terms of software changes and offers system-level considerations for an easy migration.

For more information on designing with SPI nvSRAM, refer to the Cypress application note AN64574 – Designing with Serial Peripheral Interface (SPI) nvSRAM.

2 nvSRAM Technology

The Cypress nvSRAM cell integrates an SRAM cell and a silicon-oxide-nitride-oxide-silicon (SONOS) based nonvolatile cell into a single nvSRAM cell. The nvSRAM combines the best features of an SRAM and an EEPROM, which makes it the fastest and most reliable nonvolatile memory solution in the industry. The SRAM is read from and written to an infinite number of times, while independent nonvolatile data resides in the nonvolatile elements. Data in the SRAM is transferred to the nonvolatile cell automatically during power down (the Store operation) or from the nonvolatile cell to the SRAM at power up (the Recall operation). All Store and Recall operations in nvSRAM occur in parallel because of its unique cell architecture. The high-speed, infinite read/write endurance and reliability of nvSRAM make it the nonvolatile memory of choice for most applications.

For more information on nvSRAM technology and functionalities, see the Cypress white paper Nonvolatile SRAM (nvSRAM) Basics.

Some SPI nvSRAM parts status have changed to 'Not Recommended for New Design' (NRND). It is always recommended to check the device status at www.cypress.com before designing with the parts that you have chosen for your design.

3 Benefits of SPI nvSRAM Over EEPROM

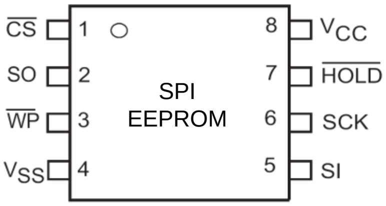
- Faster Memory
  - Random access: No page reads/writes needed
  - Full memory write at bus speed without any internal page program delay after each page write
- Easier Design
  - No software overhead for managing page boundary as with EEPROM
  - Infinite SRAM write endurance that does not require wear leveling
  - Available in industry-standard packages
- Data Security
  - Reliable silicon-oxide-nitride-oxide semiconductor (SONOS) technology
  - 1 million Store cycles
  - AutoStore: Automatically stores data on power down
- Additional Features
  - Software Store: User-controlled NV store
  - Software Recall: Allows unique data recall to its SRAM cell from the NV cell in run time
  - Available with full-featured real-time clock (RTC)
  - Pb-free technology

4 Replacing SPI EEPROM With SPI nvSRAM

The SPI nvSRAM is pin configuration compatible with SPI EEPROM, which makes it easier for EEPROM users to replace their current solution with the nvSRAM. The SPI nvSRAM offers a density up to 1 Mb and an SPI operating frequency up to 104 MHz in the 8-SOIC and 16-SOIC package options.

The SPI EEPROM is popularly available in an 8-pin SOIC package as shown in Figure 1. The nvSRAM is also available in 8-pin SOIC, making it pin compatible with the SPI EEPROM.

Figure 1. SPI EEPROM Pin Configuration (8-SOIC)



The SPI EEPROM includes the interface pins listed in Table 1.

Table 1. SPI EEPROM Pin Definition

Pin	Pin Definition
$\overline{\text{CS}}$	Chip Select used to enable or disable the device
SCK	Serial clock used to synchronize all data access
SI	Serial Input pin to the EEPROM
SO	Serial output from the EEPROM
$\overline{\text{WP}}$	Write Protect pin for hardware write protection
$\overline{\text{HOLD}}$	Hold pin to pause the serial communication

The first four pins ($\overline{\text{CS}}$, SCK, SI, and SO) in Table 1 are critical to the SPI communication in a system. The $\overline{\text{WP}}$ and $\overline{\text{HOLD}}$ pins provide additional functionalities. The $\overline{\text{WP}}$ pin is used to protect the memory from unintended write operation and is mostly enabled when the EEPROM memory is used as a read-only program memory to store a controller's firmware code. The $\overline{\text{HOLD}}$ pin is used to pause or suspend the ongoing SPI communication in a multislave SPI configuration.

The SPI nvSRAM is available in two different pin configurations— nvSRAM without AutoStore and nvSRAM with AutoStore—in 8-pin SOIC package options.

The SPI nvSRAM without the AutoStore feature has a pin configuration that is identical to the SPI EEPROM and can replace it directly on the same footprint as shown in Figure 2.

The nvSRAM with the AutoStore feature needs a dedicated V_{CAP} pin. Hence, in this configuration, the V_{CAP} pin replaces the $\overline{\text{WP}}$ pin as shown in Figure 3. You must connect a capacitor on the V_{CAP} pin to use the nvSRAM AutoStore feature.

Figure 2. SPI nvSRAM Pin Configuration Without AutoStore (Identical to SPI EEPROM)

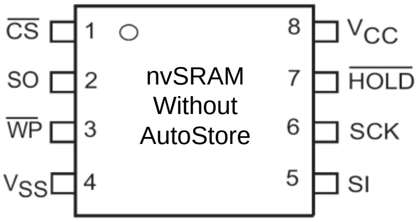
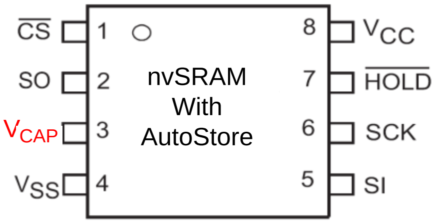


Figure 3. SPI nvSRAM Pin Configuration With AutoStore



The following sections describe relevant changes that you should consider making in the hardware and/or software for an error-free migration.

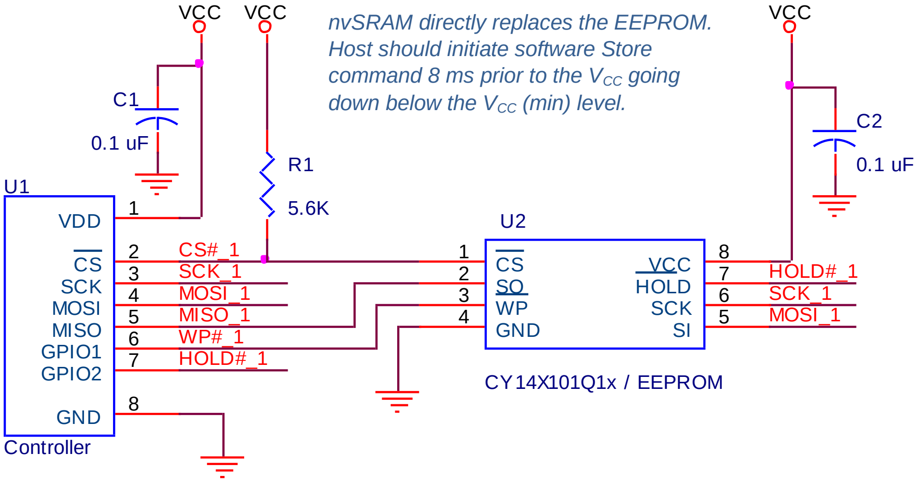
4.1 nvSRAM Without AutoStore (With Software Store)

The nvSRAM without AutoStore depends on the system's ability to perform a Store operation to secure the data from SRAM to the nonvolatile memory when the system power goes down. This is a perfect fit for those systems where a sag or failure in the system power supply can be detected in advance. The controller should initiate a software Store and secure the data in nonvolatile memory before shutting down the system's power. Alternatively, systems can include a mechanism to provide a power supply to the nvSRAM for an extended duration (up to 8 ms) after detecting the power failure. This will help ensure that the nvSRAM successfully completes the software Store cycle initiated by the controller after detecting the power failure.

4.1.1 Hardware Changes

The nvSRAM without the AutoStore option has a pin configuration identical with the EEPROM. Therefore, no hardware change is required to replace the EEPROM with nvSRAM. Figure 4 shows an example of SPI nvSRAM connections in a system.

Figure 4. SPI nvSRAM Schematic (Without AutoStore Option)



nvSRAM directly replaces the EEPROM. Host should initiate software Store command 8 ms prior to the V_{CC} going down below the V_{CC} (min) level.

4.1.2 Software Changes

Systems that use the SPI nvSRAM without the AutoStore feature need their software updated for including the software Store sequence to save the data in the nonvolatile memory cells of nvSRAM prior to the V_{CC} power down. The software Store operation is initiated in the SPI nvSRAM through a single-byte instruction (opcode). The software Store operation takes 8 ms to complete the NV Store cycle. Therefore, it requires the system to hold the nvSRAM V_{CC} power for at least 8 ms to successfully complete the Store operation once it is initiated.

Refer to the section Software Changes While Migrating From SPI EEPROM to SPI nvSRAM for further details.

4.2 nvSRAM With AutoStore (With V_{CAP} Pin)

The SPI nvSRAM with the AutoStore feature requires a small capacitor to be connected to its V_{CAP} pin to perform a nonvolatile Store automatically at power down. The V_{CAP} pin replaces the hardware Write Protect pin ($\overline{\text{WP}}$) in this package. Figure 5 shows the SPI nvSRAM with AutoStore pin configuration.

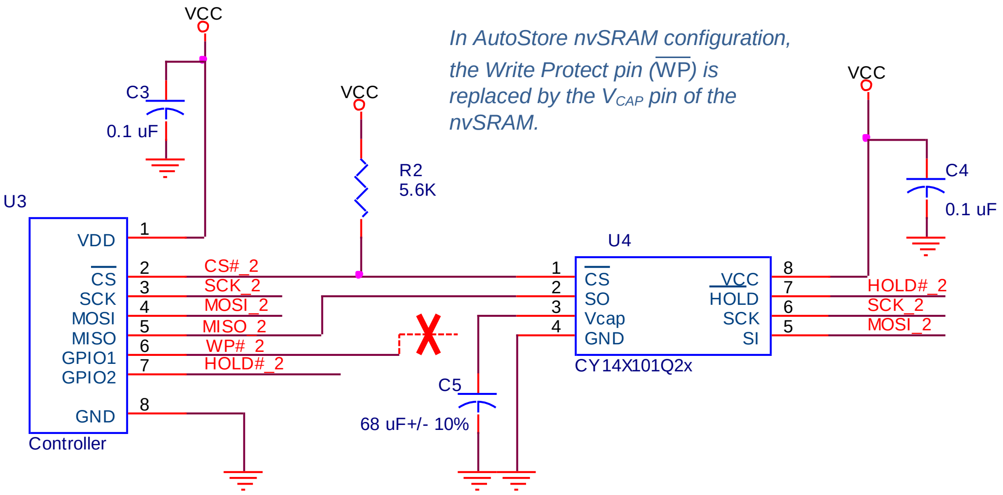
4.2.1 Hardware Changes

Using the nvSRAM with AutoStore feature requires the following hardware changes:

- Remove the controller’s GPIO connection to Pin 3 ($\overline{\text{WP}}$ pin) of the EEPROM.
- Connect the V_{CAP} capacitor to Pin 3. Typically the value of the V_{CAP} capacitor lies in the range of 61 µF–180 µF for most nvSRAM devices. Refer to the device datasheet for the appropriate capacitor value.

Figure 5 shows an example schematic of the SPI nvSRAM in a system where the Write Protect ($\overline{\text{WP}}$) pin is replaced by the V_{CAP} pin for using the AutoStore feature in nvSRAM.

Figure 5. SPI nvSRAM Schematic (With AutoStore Option)



In AutoStore nvSRAM configuration, the Write Protect pin ($\overline{\text{WP}}$) is replaced by the V_{CAP} pin of the nvSRAM.

4.2.2 Software Changes

The advantage of using nvSRAM with the AutoStore feature is that the controller does not need to initiate a software Store operation at power down. Refer to the section Software Changes While Migrating From SPI EEPROM to SPI nvSRAM for further details.

In some applications where the hardware write protect feature is enabled through the hardware pin ($\overline{\text{WP}}$), replacing those sockets with the nvSRAM with AutoStore will require some software modification to keep the write protect feature enabled. The following section discusses the write protection use model and provides an overview of its implementation at the system level.

4.3 Write Protection Use Model

This section explains write protection implementation in applications.

4.3.1 Write Protection Through the $\overline{\text{WP}}$ Pin

Typically, applications use the $\overline{\text{WP}}$ pin in one of the following two configurations.

- The $\overline{\text{WP}}$ pin is connected to a dedicated GPIO of the controller. The controller toggles the $\overline{\text{WP}}$ pin to LOW to prevent any accidental write in the status register and/or protected memory block. Similarly to remove the write protection, the controller requires pulling the $\overline{\text{WP}}$ pin to HIGH.

- The $\overline{WP}$ pin is tied to ground (V_{SS}) in the hardware to save one controller pin by permanently disabling writes to the status register. This use case mostly applies when memory is used as program memory and needs to be configured as read-only memory.

The $\overline{WP}$ pin is most commonly tied to HIGH or connected to a controller pin. If this pin is tied HIGH, the system does not use this feature; therefore, replacing the EEPROM with the nvSRAM does not pose an issue. If this pin is tied to the controller in your system, the use of write protection needs to be verified. However, if this pin is tied LOW in your system, you may want to consider whether the hardware write protection is necessary or not. If so, choose the nvSRAM with the software Store (without AutoStore) option. All the other nvSRAM benefits remain the same over EEPROM.

4.3.2 Write Protection Through Software

For most applications, it is preferable to use software write protection instead of hardware protection to save one controller GPIO. Typically, the SPI nonvolatile memories support two levels of write protection through the software.

- Write protection through the write enable latch (WEN) bit. Any write operation must be preceded by a Write Enable (WREN) instruction to set the write enable (WEN) bit. Any attempt to write into nvSRAM is ignored until WEN is set to ‘1’ prior to writing.
- Write protection through the memory block protect. In this scheme, the block protect bits BP0 and BP1 of the status register are set accordingly to inhibit the write access to one-fourth, half, or the entire memory array. The block protect bits are nonvolatile in nature.

If the $\overline{WP}$ is enabled by pulling this pin to LOW, it prevents writing into the status register. However, it does allow writing into unprotected memory block. Therefore, most systems can afford to do away with the hardware write protection feature. If write protection is required, then block-based software write protection can be enabled by setting the appropriate block protect bits (BP0 and BP1) in the status register.

4.4 Software Changes While Migrating From SPI EEPROM to SPI nvSRAM

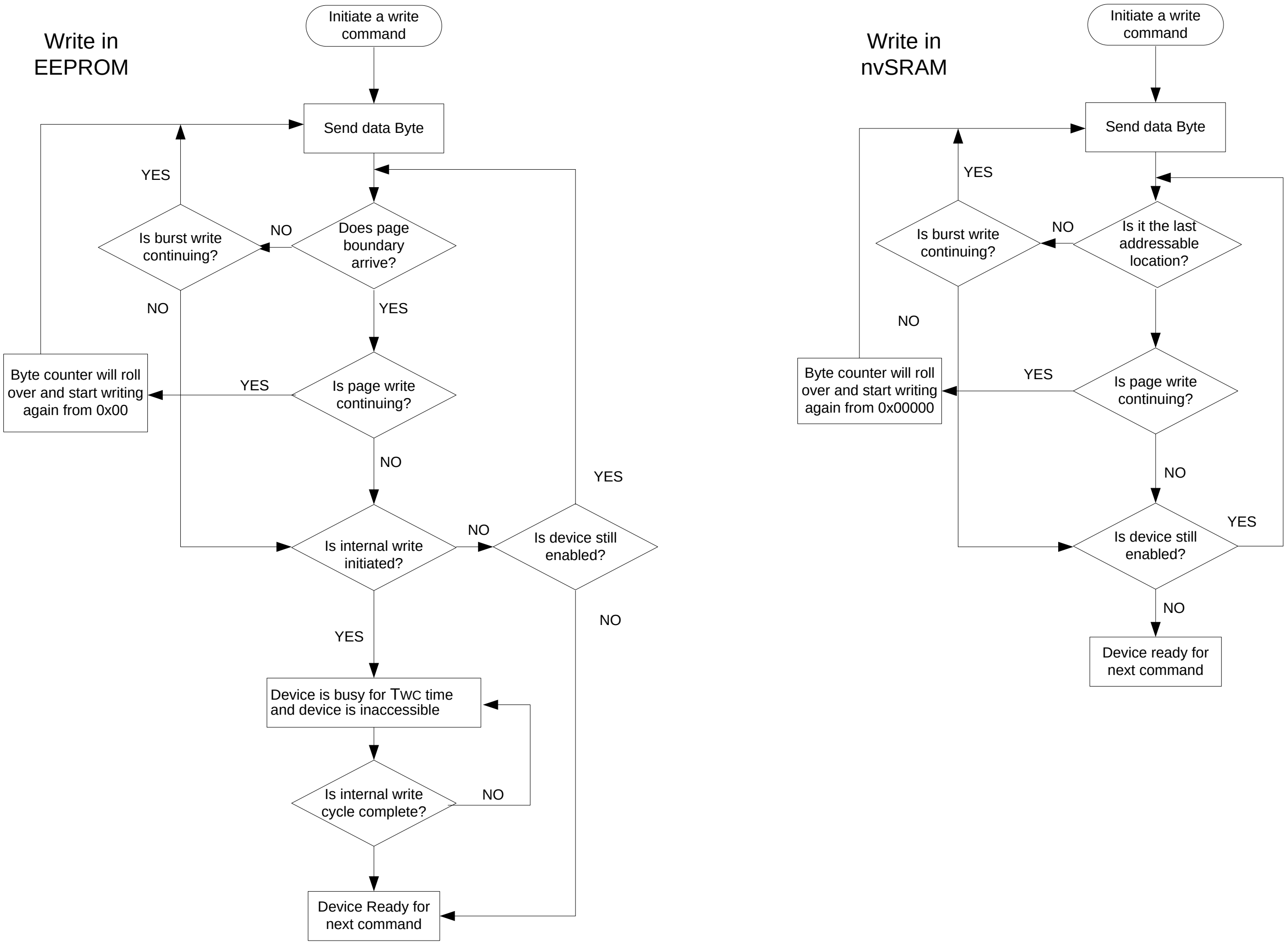
This section discusses the architectural differences between SPI EEPROM and SPI nvSRAM. You must consider them in your firmware updates.

4.4.1 Multiple EEPROM Pages Versus a Single nvSRAM Page

EEPROM devices are written or programmed on a page-by-page basis. The typical page size of an EEPROM device varies between 1/512th to 1/1024th of the main memory array size. This means, to write in the entire EEPROM memory array, the controller must initiate between 512 to 1,024 page write operations. Also, the controller has to track the counts of the total data bytes written in a page during the page write operation; otherwise, the byte address counter rolls over from the last addressable location in a page and starts writing from the 0x00 location again in the same page, overwriting the previously written data.

The nvSRAM does not have any restriction on the page size, and the entire memory array is treated as one page. So, burst write operation in the nvSRAM writes the entire memory array by initiating a single write command at the beginning of the write cycle. Once the internal counter reaches the last addressable location in the nvSRAM memory array, the counter rolls over to the start address location 0x00000. As the write operation constitutes a single page write in nvSRAM compared with multiple page writes in EEPROM, the host controller is required to keep track of just one counter for nvSRAM as opposed to tracking multiple counters for page count and byte count in a page. Use of nvSRAM simplifies the firmware design by reducing the number of steps while writing into it. Figure 6 demonstrates writing in EEPROM versus writing in nvSRAM.

Figure 6. Write Operation in EEPROM and nvSRAM


Write in EEPROM
Initiate a write command
Send data Byte
Does page boundary arrive?
NO
Is burst write continuing?
YES
NO
YES
Is page write continuing?
YES
Byte counter will roll over and start writing again from 0x00
NO
Is internal write initiated?
NO
Is device still enabled?
YES
NO
YES
Device is busy for TWC time and device is inaccessible
Is internal write cycle complete?
NO
Device Ready for next command
Write in nvSRAM
Initiate a write command
Send data Byte
Is it the last addressable location?
NO
Is burst write continuing?
YES
NO
Byte counter will roll over and start writing again from 0x00000
YES
Is page write continuing?
NO
Is device still enabled?
YES
NO
Device ready for next command

4.4.2 Removing the Delay Cycle for Write Operations

The nvSRAM performs write operations at bus speed; therefore, no delay cycle is required to complete write operations in nvSRAM. In contrast, the serial EEPROM requires a wait time of 5 ms to 7 ms to transfer every page of data from the page buffer of the nonvolatile cell. Page size varies depending upon the EEPROM density. Typically a 1-Mb serial EEPROM consists of a 256-byte page. Any such delays included in the firmware code can be removed.

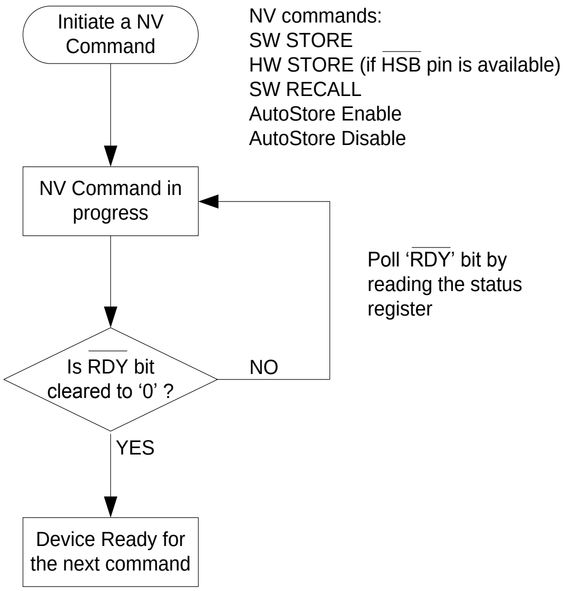
4.4.3 Software Store (for nvSRAM Without AutoStore)

Systems that use nvSRAM without AutoStore need no hardware change. However, these systems must implement software changes to initiate a Store operation when a droop on VCC is detected. A Store operation can be initiated in SPI nvSRAM through a single-byte instruction (opcode), and it takes 8 ms to complete the nonvolatile Store cycle. This requires the system to hold the nvSRAM VCC power for 8 ms to successfully complete the Store operation.

4.4.4 nvSRAM $\overline{\text{RDY}}$ Bit in Status Register

Similar to the write in progress (WIP) bit defined in the status register of serial EEPROMs, nvSRAM defines a read-only bit ($\overline{\text{RDY}}$). The $\overline{\text{RDY}}$ bit indicates the ready status of the device. The device sets this bit to ‘1’ when it is busy doing a software Store or Recall operation. Once the device completes its ongoing NV operation, the nvSRAM clears the $\overline{\text{RDY}}$ bit to ‘0’. If needed, you can poll this bit by reading the status register to learn the device ready/busy status after initiating the software Store or Recall command. Figure 7 shows the flow diagram for polling the $\overline{\text{RDY}}$ bit status. The EEPROM defines this bit location as write in progress (WIP). When any CHIP ERASE, SECTOR ERASE, PAGE ERASE, and PAGE WRITE cycle is in progress, the WIP bit is set to ‘1’, indicating a device busy status. The host controller polls this bit to learn the EEPROM ready/busy status.

Figure 7. nvSRAM Ready Bit ($\overline{\text{RDY}}$) Polling


Initiate a NV Command
NV Command in progress
Is $\overline{\text{RDY}}$ bit cleared to '0' ?
NO
YES
Device Ready for the next command

NV commands:
SW STORE
HW STORE (if $\overline{\text{HSB}}$ pin is available)
SW RECALL
AutoStore Enable
AutoStore Disable

Poll '$\overline{\text{RDY}}$' bit by reading the status register

4.4.5 Opcode Difference

Table 2 lists the combined set of opcodes that are available for EEPROM and nvSRAM. Opcodes related to EEPROM cell operation, such as PE, SE, CE, and RDID (highlighted in Table 2), are not applicable to the nvSRAM and become redundant. nvSRAM supports all the standard opcodes of serial EEPROM. The SPI nvSRAM supports many more opcodes, which are not applicable to EEPROM devices. You can use these opcodes to provide additional features in the system while using nvSRAM.

Table 2. Instruction (Opcode) Comparison Between EEPROM and nvSRAM

Instruction Name	Opcode	Opcode Description	SPI EEPROM	SPI nvSRAM
READ	0x03	Read	√	√
WRITE	0x02	Write	√	√
WREN	0x06	Write enable	√	√
WRDI	0x04	Reset the write enable latch	√	√
RDSR	0x05	Read STATUS register	√	√
WRSR	0x01	Write STATUS register	√	√
PE	0x42	Page erase (Not available as standard instruction)	√	X
SE	0xD8	Sector erase (Not available as standard instruction)	√	X
CE	0xC7	Chip erase (Not available as standard Instruction)	√	X
RDID	0xAB	Release from deep power down (Not available as standard instruction)	√	X
DPD	0xB9	Deep Power-down mode (Not available as standard instruction)	√	√
FAST_RDSR	0x09	Read STATUS register in Fast mode SPI at 104 MHz	X	√

Instruction Name	Opcode	Opcode Description	SPI EEPROM	SPI nvSRAM
FAST_READ	0x0B	Read SRAM in Fast mode SPI at 104 MHz	X	√
WRTC	0x12	Write in RTC register	X	√
RDRTC	0x13	Read from RTC register	X	√
FAST_RDRTC	0x1D	Read from RTC registers in Fast mode SPI at 104 MHz	X	√
STORE	0x3C	Initiate software store	X	√
RECALL	0x60	Initiate software recall	X	√
ASENB	0x59	Enable AutoStore	X	√
ASDISB	0x19	Disable AutoStore	X	√
WRSN	0xC2	Write serial number	X	√
RDSN	0xC3	Read serial number	X	√
FAST_RDSN	0xC9	Read serial number in Fast mode SPI at 104 MHz	X	√
RDID	0x9F	Read device ID	X	√
FAST_RDID	0x99	Read device ID in Fast mode SPI at 104 MHz	X	√

4.5 Additional System-Level Considerations

Table 3 lists the key feature and function differences between the SPI EEPROM devices and Cypress SPI nvSRAM devices. You should take these differences into consideration at the system level for a proper migration.

Table 3. Feature Comparison Between SPI EEPROMs and SPI nvSRAM

Feature/Function	EEPROM	nvSRAM Without V_{CAP}	nvSRAM With V_{CAP}
		CY14XXXXQ1A	CY14XXXXQ2A
V_{CC}	1.8 V to 5.5 V	2.5 V (Typ) – 2.4 V to 2.6 V (CY14CXXXX) 3 V (Typ) – 2.7 V to 3.6 V (CY14BXXXX) 5 V (Typ) – 4.5 V to 5.5 V (CY14EXXXX) (nvSRAM supports three operating voltage ranges. Depending upon the application requirement, the nvSRAM operating voltage option should be selected.)	
PCB change	–	No change	Add a V_{CAP} at Pin 3
Power up	Immediate, after V_{CC} min arrives.	20 ms (max), after the V_{SWITCH} level arrives. V_{SWITCH} is typically 50 mV below the V_{CC} min.	
Page write delay	5–10 ms/page write.	No delay. All writes happen in SRAM location.	
No of pages	512–1,025 pages per device. Counter rollover from the page boundary. For example, with 256-byte page size, counter rolls over to start address after writing 256 bytes in the page.	Single-page device. Counter rollover from the page boundary. For example, with 1-Mb (or 128K x 8) nvSRAM density, counter rolls over to the start of memory address location after writing 128 KB in nvSRAM page.	
Hardware write protect	Yes	Yes	No (No $\overline{WP}$ pin)
Status register	Bit 6 is ‘0’.	Bit 6 is SNL (serial number lock). Factory default is ‘0’. You can set this bit to ‘1’ to lock the serial number after writing a new 8-byte serial number. You can leave this bit at the factory default setting ‘0’ if not used.	

Feature/Function	EEPROM	nvSRAM Without V_{CAP}	nvSRAM With V_{CAP}
		CY14XXXXQ1A	CY14XXXXQ2A
	Bit 0 is WIP/$\overline{\text{RDY}}$.	Bit 0 is $\overline{\text{RDY}}$. Default value is ‘0’. It is set to ‘1’ during Store cycle, which is required only at power fail.	
Power Down STORE	Automatic	Host is required to initiate a software Store and save the SRAM data into NV memory before power down.	Automatic
Last access to power down	0 µs	8 ms, during which Store needs to be initiated.	0 µs; part automatically performs a Store for 8 ms.
Time to enter sleep	100 µs (if supported)	8 ms	
Wake from sleep	100 µs (if supported)	20 ms	
Device ID	1 byte (if supported)	4 bytes (Available in all nvSRAMs)	
Serial Number	Not applicable.	8 bytes, set by the user (Available in all nvSRAMs)	

5 Summary

Cypress nvSRAM allows faster, easier, and more reliable data storage than EEPROM. The nvSRAM does not require any additional memory management software or firmware, which makes integration easier with all the controllers that support the standard SPI interface. Cypress nvSRAM devices are available in compatible packages, making it easy to replace EEPROM with nvSRAM in any application.

6 Related Application Note

AN64574 – Designing with Serial Peripheral Interface (SPI) nvSRAM provides a few key design considerations and firmware tips to guide users designing with SPI nvSRAM. An associated project for PSoC® 1 and a library component for PSoC 3 are also provided as an example project, which demonstrates SPI nvSRAM access by a standard SPI master controller.

Document History

Document Title: AN68174 - Migrating from Serial Peripheral Interface (SPI) EEPROM to SPI nvSRAM

Document Number: 001-68174

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	3239539	ZSK	04/25/2011	New application note.
*A	3338089	ZSK	08/11/2011	Content Update. Added Figure 6 and Figure 7. Added Table 1 and Table 2.
*B	3638304	ZSK	06/07/2012	Added table 1 to replace the verbiage section on EEPROM pin description. Updated template
*C	4387067	ZSK	05/22/2014	Updated application note template. Fixed wrong AN reference in the header under the “Related Application Notes”: AN65574 to AN64574. Revised abstract section to align with the application note content. Replaced obsolete application note title “Nonvolatile SRAM (nvSRAM) Basics – AN6023” with the white paper title “Nonvolatile SRAM (nvSRAM) Basics.” Added a “Related Application Note” section after the “Summary.”
*D	5744565	ZSK	05/22/2017	Updated template

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

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.