Power Saving with Cypress’s 65-nm Asynchronous PowerSnooze™ SRAM

Author: Nilesh Badodekar
Associated Project: No
Associated Part Family: CY7S10xxG/CY7S10xxGE

This application note explains the PowerSnooze™ feature of Cypress’s 65-nm asynchronous fast SRAM devices (CY7S10xxG family). PowerSnooze allows the SRAM to enter a low-power mode during long chip disable periods. A user-controlled pin (DS) enables a seamless transition between high-speed mode and low-power mode. This application note also describes critical timing parameters for the mode transitions as well as a sample SRAM interface configuration to use the PowerSnooze feature in application systems.

Introduction

With the advent of mobile technology and portable battery-backed devices, power consumption has become one of the key factors in system design. System designers face the dilemma of selecting between faster operating speeds and lower power consumption for the microcontrollers/ASICs, peripherals, and memory devices in their systems.

Microcontrollers from Texas Instruments and NXP Semiconductors have special low-power modes such as deep power-down and deep sleep. Applications running on these controllers can use these features to save power. The controller can run at full speed during normal operation but go into low-power mode afterwards, saving power. During this low-power mode, peripherals and memory devices are also expected to save power, which is a challenge to memory devices interfaced to such systems. Asynchronous SRAMs are typically used as off-chip cache memories, scratch pad memory, or other similar scenarios that need faster access times on the order of tens of nanoseconds.

Fast SRAMs offer access times between 10 ns and 20 ns but consume significant power during both active and standby modes. The low-power SRAMs, on the other hand, have slower access times of about 45 ns to 70 ns. Traditionally, system designers have used low-power SRAMs in low-power applications, but today, they need both: fast access times for their memory interface as well as low power consumption when their system is idle.

Cypress’s 65-nm asynchronous SRAMs with the PowerSnooze™ feature offer the best of both worlds to make a true high-speed, low-power device.

A True Fast, Low-Power Asynchronous SRAM

Cypress’s 65-nm asynchronous SRAM with the PowerSnooze feature allows system designers to enjoy the benefits of fast access time and low-power consumption in the same chip. Cypress SRAMs with the letter “S” in their ordering code (for example, CY7S1061G30), which stands for 16-Mb asynchronous fast PowerSnooze SRAM, have an additional input pin, called DS, that is apart from the conventional asynchronous SRAM interface. This pin is user controlled, which allows users to change the operating modes of SRAM on the fly.

During normal operation, the controller/ASIC/FPGA1 can access SRAM data at a full speed of 100 MHz (access time of 10 ns). However, when the controller enters low-power modes, it can control the DS pin to put the SRAM in deep-sleep mode. In deep-sleep mode, these SRAMs consume a current of 22 μA at 85 °C for a 16-Mb SRAM, making this device a true high-speed, low-power device. Figure 1 shows a simplified architecture of a PowerSnooze SRAM.

---

1 The terms “controller,” “ASIC,” “microcontroller,” and “FPGA” are used interchangeably in this document.
**Modes of Operation**

As shown in Figure 1, PowerSnooze SRAMs have an additional power management block that allows users to operate the SRAM in the following three modes:

- **Active mode**
- **Standby mode**
- **Deep-sleep mode**

**Active Mode**

When the DS pin is in a logic HIGH state and SRAM is enabled (‘CE is logic LOW’), then it is in active mode. In this mode, the device can operate with an access time of 10 ns, and it will consume active current $I_{CC}$ (typically in the order of milliamps). Refer to the SRAM datasheet for the timing and DC parameters.

**Standby Mode**

When the DS pin is in a logic HIGH state and SRAM is disabled (‘CE is logic HIGH’), the SRAM is in standby mode, also referred to as “chip disable mode.” In this mode, no access is allowed to the SRAM; it will consume standby current $I_{SB}$ (typically in milliamps, but an order less than active current). Data stored in SRAM prior to entering this mode will be retained while it is in the standby mode. Refer to the SRAM datasheet for standby mode conditions.

**Deep-Sleep Mode**

Cypress offers the PowerSnooze feature in both 4-Mb and 16-Mb asynchronous SRAMs. These two densities involve a minor difference in the way deep-sleep mode is entered and exited. Systems designed with 4-Mb PowerSnooze SRAMs will need minor firmware changes when upgrading to 16-Mb PowerSnooze SRAMs. Systems designed with 16-Mb PowerSnooze SRAM can use 4-Mb PowerSnooze SRAMs with no firmware changes. The following sections explain the deep-sleep mode for 16-Mb and 4-Mb SRAMs in detail.

---

2 For all dual chip enable devices, ‘CE’ is the logical combination of ‘CE1’ and ‘CE2’. When ‘CE1’ is LOW and ‘CE2’ is HIGH, ‘CE’ is LOW; when ‘CE1’ is HIGH or ‘CE2’ is LOW, ‘CE’ is HIGH.
Deep-Sleep Mode for 16-Mb SRAM

The 16-Mb PowerSnooze SRAM can enter the deep-sleep mode only from standby mode (CE is logic HIGH$^2$). After the chip is disabled for at least $t_{CEDS}$ time, the DS pin must be asserted (to logic LOW) to enter deep-sleep mode. Refer to Figure 2 for the timing requirements. On assertion of the DS pin (to logic LOW), the SRAM enters deep-sleep mode after $t_{DS}$ time. During deep-sleep mode, DS must remain asserted and the SRAM must remain in standby mode (CE is logic HIGH). While entering deep-sleep mode ($t_{CEDS} + t_{DS}$ time), SRAM will consume standby current $I_{SB2}$; once in deep-sleep mode, it will consume the deep-sleep current $I_{DS}$.

To exit the deep-sleep mode, the DS pin must be deasserted (to logic HIGH$^2$) while SRAM is still disabled. SRAM will take $t_{DSCE}$ time to return to standby mode. After $t_{DSCE}$ time, SRAM can be enabled (CE to logic LOW$^2$) to enter active mode.

It must be noted that any violations of these timing parameters do not guarantee successful entry to or exit from deep-sleep mode. Refer to Table 1 for a summary of the timing requirements.

Deep-Sleep Mode for 4-Mb SRAM

The 4-Mb PowerSnooze SRAM can enter the deep-sleep mode either from standby mode (CE is logic HIGH$^2$) or active mode. On assertion of the DS signal (to logic LOW$^2$), the SRAM starts its transition to deep-sleep mode. If DS continues to be asserted for at least $t_{DS}$ time, the SRAM enters deep-sleep mode and consumes deep-sleep current $I_{DS}$. Refer to Figure 3 for the timing requirements. During deep-sleep mode, DS must remain asserted. While entering deep-sleep mode ($t_{DS}$ time), SRAM will consume standby current $I_{SB2}$; once in deep-sleep mode, it will consume the deep-sleep current $I_{DS}$. If the DS signal gets asserted during a write operation, then the SRAM cannot guarantee successful operation, as the device starts transitioning to deep-sleep mode.

To exit the deep-sleep mode, the DS pin must be deasserted (to logic HIGH). If DS is asserted for at least $t_{DS}$ time, then SRAM will be disabled internally for $t_{DSCE}$ time. This time ($t_{DSCE}$) allows the external controller to access CE and disable it externally. The device must continue to be disabled until $t_{DSCE}$ time. After $t_{DSCE}$ time from the deassertion of DS, SRAM can be accessed by enabling it (CE to logic LOW) to enter active mode. If $t_{DS}$ is not met by the system, then $t_{DSCE}$ is not guaranteed by the SRAM. During this period (when $t_{DS} < t_{DS(min)}$), SRAM must be disabled externally until $t_{DSCE}$ time to avoid corruption of the SRAM data. Refer to Table 2 for a summary of these timing requirements.

Figure 2. Deep-Sleep Mode for 16-Mb SRAM – Entry and Exit Sequence
Table 1. Key Timing Parameters for 16-Mb PowerSnooze SRAM

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
<th>Timing</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_{CEDS} )</td>
<td>Time between deassertion of CE and assertion of DS</td>
<td>100 ns –</td>
</tr>
<tr>
<td>( t_{DS} )</td>
<td>DS assertion to deep-sleep mode transition time</td>
<td>– 1 ms</td>
</tr>
<tr>
<td>( t_{DSCE} )</td>
<td>Time between deassertion of DS and assertion of CE</td>
<td>1 ms –</td>
</tr>
</tbody>
</table>

Figure 3. Deep-Sleep Mode for 4-Mb SRAM– Entry and Exit Sequence

Table 2. Key Timing Parameters for 4-Mb PowerSnooze SRAM

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
<th>Timing</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_{PDS} )</td>
<td>Minimum time for DS to be LOW for part to successfully exit deep-sleep mode</td>
<td>100 ns –</td>
</tr>
<tr>
<td>( t_{DS} )</td>
<td>DS assertion to deep-sleep mode transition time</td>
<td>– 1 ms</td>
</tr>
<tr>
<td>( t_{DSCE} )</td>
<td>Time between deassertion of DS and chip disable (if ( t_{PDS} \geq t_{PDS(MIN)} ))</td>
<td>– 100 µs</td>
</tr>
<tr>
<td>( t_{DISC} )</td>
<td>Time between deassertion of DS and chip disable (if ( t_{PDS} &lt; t_{PDS(MIN)} ))</td>
<td>– 0 µs</td>
</tr>
<tr>
<td>( t_{DSCA} )</td>
<td>DS# deassertion to chip access (active/standby) (if ( t_{PDS} \geq t_{PDS(MIN)} ))</td>
<td>300 µs</td>
</tr>
<tr>
<td>( t_{DSCA} )</td>
<td>DS# deassertion to chip access (active/standby) (if ( t_{PDS} &lt; t_{PDS(MIN)} ))</td>
<td></td>
</tr>
</tbody>
</table>

As the SRAM is disabled for the entire period when it is in deep-sleep mode, data stored in SRAM prior to entering this mode is retained. Refer to the SRAM datasheet for the deep sleep-mode conditions. Table 1 and Table 2 summarize the key timing parameters that must be followed for PowerSnooze SRAMs.
Interfacing PowerSnooze SRAM in Systems

High-end microcontrollers like AM18XX series from TI or LPC 177X/LPC178X series from NXP have special deep-sleep modes that the controller can enter to save power consumption. TI’s AM18XX series controllers have an input signal, called “DEEPSLEEP”, which can put the controller in low-power mode. This signal can be interfaced with the DS pin of PowerSnooze SRAMs. The LPC177x/178x series controller can enter low-power mode through a software configuration, which sequentially disables the peripherals and memory devices connected to it. This sequence can be used to generate an extra signal apart from normal SRAM interface signals to control the DS pin of PowerSnooze SRAMs.

In situations where system designers intend to use PowerSnooze SRAMs in existing applications or want to interface to an ASIC or microcontroller without a special low-power mode, the DS pin can be controlled with a normal GPIO. Figure 4 illustrates an interfacing scenario where a GPIO is interfaced to the DS pin of the PowerSnooze SRAM. During active and standby modes, the DS pin should always be at logic HIGH.

To enter deep-sleep mode, the microcontroller must follow the timing sequences explained in the previous section for CE and DS. Thus, by performing minor modifications in the software and hardware, system designers can use the PowerSnooze SRAM in existing applications.

Figure 4. Interfacing PowerSnooze SRAM
Summary

Cypress’s asynchronous PowerSnooze SRAM offers the best of both worlds—a fast and low-power SRAM. PowerSnooze SRAMs are available in multiple options and packages. Table 3 summarizes the 16-Mb and 4-Mb PowerSnooze SRAM product portfolio. Refer to the datasheet for information on choosing the right configuration and ordering.

Table 3. PowerSnooze SRAM Product Portfolio

<table>
<thead>
<tr>
<th>Parameters</th>
<th>PowerSnooze SRAMs (High-Speed Low-Power)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>16 Mb</td>
</tr>
<tr>
<td>Temperature range</td>
<td>–40 °C to +85 °C</td>
</tr>
<tr>
<td>Technology</td>
<td>65 nm</td>
</tr>
<tr>
<td>Operating current $I_{CC}$ (max)</td>
<td>110 mA</td>
</tr>
<tr>
<td>Standby current $I_{SB2}$ (max)</td>
<td>30 mA</td>
</tr>
<tr>
<td>Deep-sleep current $I_{DS}$ (max)</td>
<td>22 µA</td>
</tr>
<tr>
<td>Packages</td>
<td>48-pin TSOP I, 54-pin TSOP II, 48-ball VFBGA</td>
</tr>
<tr>
<td>Speed</td>
<td>10 ns</td>
</tr>
<tr>
<td>Copper lead frame</td>
<td>Yes</td>
</tr>
<tr>
<td>Pb free and leaded</td>
<td>RoHS</td>
</tr>
<tr>
<td>Availability</td>
<td>Production</td>
</tr>
</tbody>
</table>

About the Author

Name: Nilesh Badodekar
Title: Applications Engineer Staff
Background: Nilesh Badodekar has a Master’s degree in Visual Information Processing and Embedded Systems from I.I.T. Kharagpur, India.
Document History

Document Title: AN89371 – Power Saving With Cypress’s 65-nm Asynchronous PowerSnooze™ SRAM

Document Number: 001-89371

<table>
<thead>
<tr>
<th>Revision</th>
<th>ECN</th>
<th>Orig. of Change</th>
<th>Submission Date</th>
<th>Description of Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>**</td>
<td>4364981</td>
<td>NILE</td>
<td>04/29/2014</td>
<td>New Application Note.</td>
</tr>
<tr>
<td>*A</td>
<td>4623709</td>
<td>NILE</td>
<td>03/30/2015</td>
<td>Added details about Cypress’s 4-Mb PowerSnooze SRAMs. Added Table 1 and Table 2. Updated Table 3 for specifications of 4-Mb PowerSnooze SRAMs. Updated template</td>
</tr>
</tbody>
</table>
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

- **Automotive**
  - [cypress.com/go/automotive](http://cypress.com/go/automotive)
- **Clocks & Buffers**
  - [cypress.com/go/clocks](http://cypress.com/go/clocks)
- **Interface**
  - [cypress.com/go/interface](http://cypress.com/go/interface)
- **Lighting & Power Control**
  - [cypress.com/go/powerpsoc](http://cypress.com/go/powerpsoc)
- **Memory**
  - [cypress.com/go/memory](http://cypress.com/go/memory)
- **PSOC**
  - [cypress.com/go/psoc](http://cypress.com/go/psoc)
- **Touch Sensing**
  - [cypress.com/go/touch](http://cypress.com/go/touch)
- **USB Controllers**
  - [cypress.com/go/usb](http://cypress.com/go/usb)
- **Wireless/RF**
  - [cypress.com/go/wireless](http://cypress.com/go/wireless)

## PSoC® Solutions

- [psoc.cypress.com/solutions](http://psoc.cypress.com/solutions)
  - PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

## Cypress Developer Community

- [Community](http://community.cypress.com)
- [Forums](http://forum.cypress.com)
- [Blogs](http://blogs.cypress.com)
- [Video](http://video.cypress.com)
- [Training](http://training.cypress.com)

## Technical Support

- [cypress.com/go/support](http://cypress.com/go/support)

PSOC is a registered trademark and PSoC Creator is a trademark of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are the property of their respective owners.

© Cypress Semiconductor Corporation, 2014-2015. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

This Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress’ product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.