

Design Recommendation for Battery-Backed SRAMs Using Cypress MoBL® SRAMs

Author: Anuj Chakrapani
Associated Part Family: MoBL® SRAMs

AN44517 gives the design recommendation for battery-backed SRAMs using a Cypress MoBL® SRAM (also called ultra low-power SRAM or micropower SRAM) and a microprocessor supervisory chip. This recommendation ensures better data integrity in SRAM, in case of power failures in a battery-backed application, when compared to other techniques that are also described in this document.

1 Introduction

Applications such as point-of-sale (POS) terminals, network processing engines, servers, and medical implant devices use battery-backed memory solutions with SRAMs.

Because SRAMs are volatile memories (that is, they cannot retain their contents on loss of power), they need an alternate source of power to ensure data integrity in the event of a power failure.

A basic battery-backed application block diagram consisting of a microprocessor (or FPGA), a flash memory, and an SRAM is illustrated in [Figure 1](#). The application has two power sources: main (on-board regulated) power supply and a battery (some applications also employ a super capacitor instead of a battery).

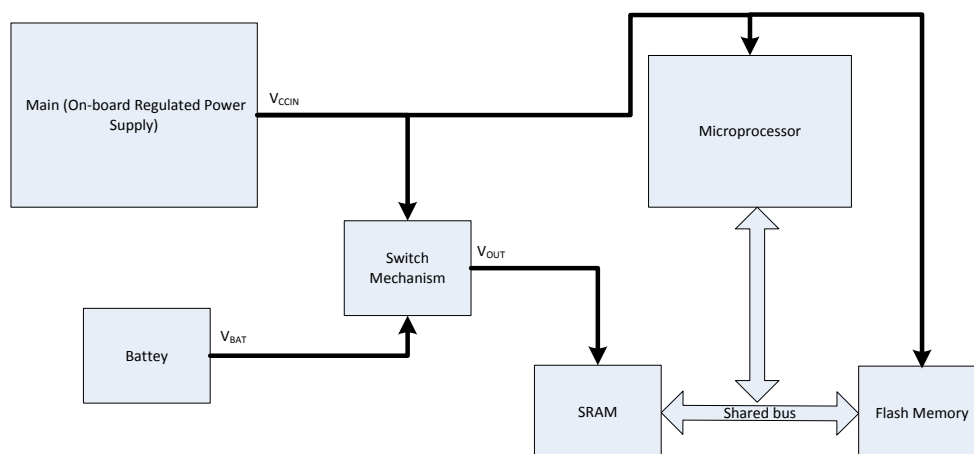
During normal operation, the main power supply is used for all communication between the microprocessor^[1], flash memory, and SRAM devices. In the event of a power failure, it is important to back up critical data and secure it in the SRAM. You can accomplish this using a battery that supplies power only to the SRAM during power failure and an associated switch mechanism. This ensures data integrity in the SRAM, while the rest of the application remains without power.

While in battery mode, the application is designed to draw as little current as possible to ensure longer battery life. To achieve this, place the SRAM in standby mode, so it is disabled and no read or write accesses are made. This ensures that the data remains intact. Normal operation resumes when the main power is restored. Cypress MoBL® SRAMs typically draw standby current less than 10 μ A. This ensures longer battery life and makes them ideal for battery-operated applications.

The rest of this application note discusses recommendations to design battery-backed memories using Cypress MoBL SRAMs.

¹ An FPGA or microcontroller can also be used instead of the microprocessor. The points discussed in this application note are independent of the type of device driving the on-board memories. The rest of the document will refer to this device as a "microprocessor", but it can be any logic device.

Figure 1. Basic Battery Backed Application: Simplified Block Diagram



2 Details

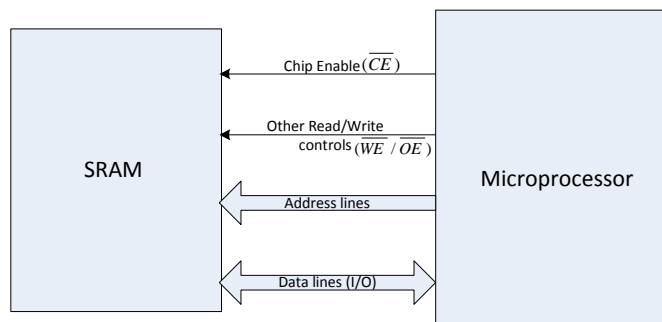
Figure 2 takes a closer look at the SRAM interface and its associated control signals. All address, data, and control pins including the Chip Enable pin (\overline{CE}) are driven by the microprocessor. In the event of a power failure, the battery keeps the SRAM powered, while the microprocessor ceases to be powered. This can result in the microprocessor's outputs either getting discharged to ground (GND) or floating to an intermediate voltage level. Because the outputs of the microprocessor drive the SRAM, a floating node on one of the SRAM Chip Enable (\overline{CE}) inputs can result in the latter drawing excess current. This is because the \overline{CE} pin of the SRAM needs to be in the de-asserted state and at high CMOS levels for meeting the standby current specifications as mentioned in the datasheet. Floating the \overline{CE} input can result in it being biased to an intermediate voltage level, thereby resulting in high current that can potentially reduce the battery life of the application.

To prevent the above condition and to ensure the SRAM is in standby mode during power failure in a battery-backed application, three design techniques are discussed in the following sections:

- Legacy Design Technique using Pull-Up Resistors (Not Recommended)
- Recommended Design Technique: Using Microprocessor Supervisory Circuits
- Alternate Design Technique: Using RESET Monitors

The legacy technique and the design technique using the reset monitor have some limitations and hence, Cypress recommends that you use a supervisory chip to guarantee successful transition from the normal mode to the battery mode and vice versa, while ensuring long battery life.

Figure 2. Microprocessor – SRAM Interface



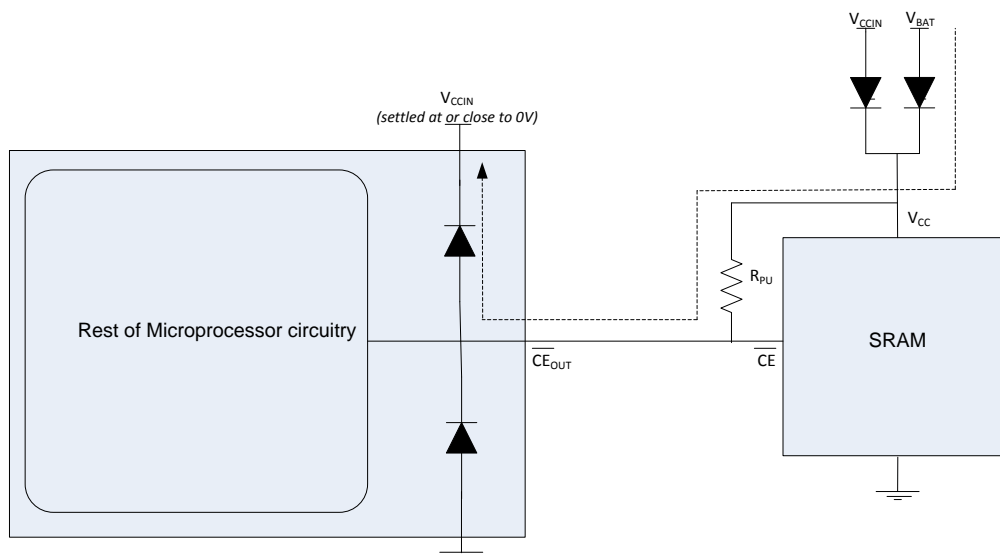
3 Legacy Design Technique using Pull-Up Resistors (Not Recommended)

Consider an SRAM that has an active low chip enable (\overline{CE}) pin. For placing the SRAM in the standby mode, this pin needs to be de-asserted (disabled) to the logic HIGH state, while other controls, address, and data pins can be floated under this condition. An external weak pull-up resistor of the order of kilo ohms on the \overline{CE} pin connected to the V_{CC} pin of SRAM can be used to place the device in the standby mode because the SRAM V_{CC} is equal to the battery voltage in the battery mode. Thus, the pull-up resistor will bias the \overline{CE} input to HIGH when it is not actively driven by the microprocessor. However, this scheme has a limitation, as explained in the following paragraph.

In the normal mode, the output of the microprocessor \overline{CE}_{OUT} actively drives the \overline{CE} pin of the SRAM for memory access. During power failure, the microprocessor ceases driving its outputs; the external pull-up resistor then places the SRAM in the standby mode by pulling the \overline{CE} pin HIGH. Though this technique works in theory, it is not a robust solution and can result in undesirable high current. The reason is as follows: the \overline{CE}_{OUT} pin of the microprocessor may be an I/O and can have protection diodes inside, as shown in [Figure 3](#).

These diodes protect the microprocessor against high voltage on the pins, and hence turn on if the applied voltage exceeds the rails by a voltage equal to one diode drop. Now, if an external pull-up resistor is used, because the supply of the microprocessor (V_{CCIN}) would have settled at or close to 0 V on power down, pulling its \overline{CE}_{OUT} pin high turns on these protection diodes, as shown in [Figure 3](#). The continuous diode current can potentially drain the battery. It can also turn on circuits inside the microprocessor, because the internal core voltage of the microprocessor now rises to maintain the diode current. To avoid these undesirable effects, it is recommended to use microprocessor supervisory circuits described in the section, [Recommended Design Technique: Using Microprocessor Supervisory Circuits](#) on page 5.

Figure 3. Protection Diodes inside the Microprocessor Chip



Notes

- SRAM is powered by V_{CCIN} during normal operation and by V_{BAT} during battery mode. The two external diodes, shown in the V_{CCIN} and V_{BAT} paths to the SRAM, prevent reverse current
- The dotted arrow indicates current flow through the pull-up resistor and diode when V_{CCIN} trips.
- R_{PU} indicates the external pull-up resistor to disable the chip.

4 Recommended Design Technique: Using Microprocessor Supervisory Circuits

Microprocessor supervisory circuits (simply called supervisors) are semiconductor devices that monitor voltage levels in power supplies and provide battery backup control. They also provide write protection during power failures to prevent data corruption in battery-backed solutions. Due to the limitations of using external pull-up resistors, most battery-backed solutions today use a supervisor chip instead of external resistors.

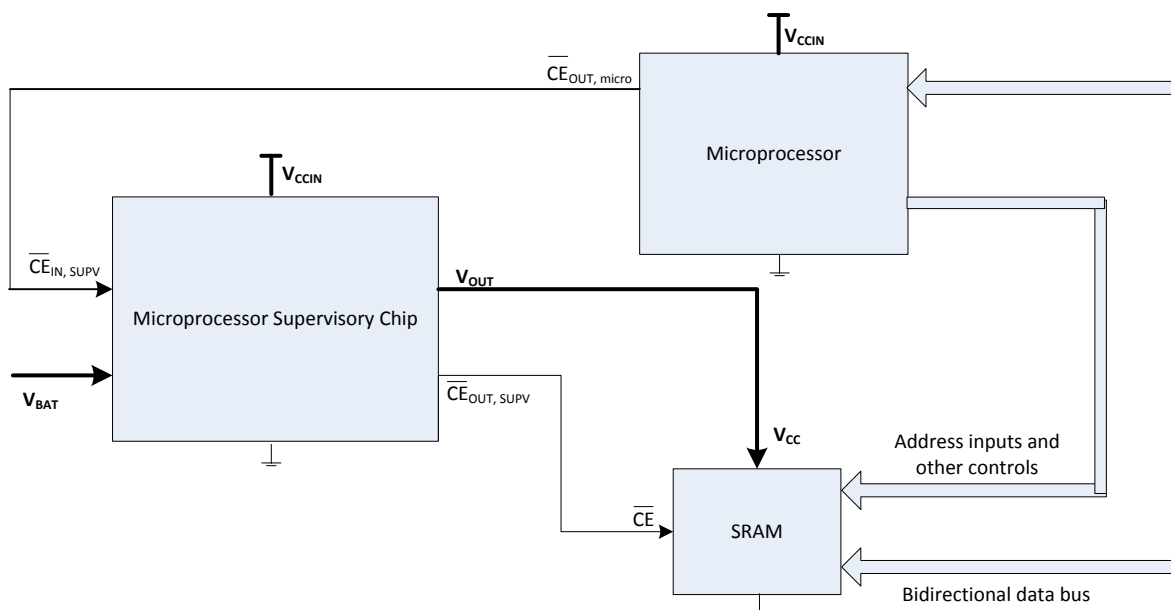
A typical supervisor chip has input pins V_{CCIN} , V_{BAT} , \overline{CE}_{IN} , and output pins V_{OUT} , \overline{CE}_{OUT} , as illustrated in Figure 4.

The input pins, V_{CCIN} and V_{BAT} , are power supplies from the mains and battery respectively. V_{OUT} is the output pin that tracks either V_{CCIN} (in normal mode) or V_{BAT} (in battery mode) and is connected to the V_{CC} pin of SRAM, so that SRAM has uninterrupted supply.

The chip enable output of the microprocessor is connected to the chip enable input pin of the supervisory chip, whose \overline{CE}_{OUT} output is connected to the \overline{CE} (chip enable) of the SRAM. This sequence of connections enables the supervisor to be a “transparent” device in the normal mode. In other words, it “tracks” the chip enable output of the microprocessor in the normal mode. Therefore, it is the latter that drives the SRAM chip enable. In the battery mode, however, the supervisor pulls its \overline{CE}_{OUT} output HIGH, thereby disabling the SRAM and placing it in standby mode.

Thus, the supervisor chip stays transparent and allows the microprocessor to drive the chip enable of the SRAM and actively control the operation and accesses of the SRAM in the normal mode. However, when the main supply trips and the battery takes over, the supervisor chip assumes direct control of disabling the SRAM, thereby placing the device in standby mode. Thus, at all times, it ensures that the chip enable, \overline{CE} , of the SRAM is always biased at the right voltage levels, and does not float to an intermediate state.

Figure 4. Recommended Battery Backed Design using Supervisory Chip



4.1 Other Considerations

Some SRAMs have multiple chip enable pins that may be active LOW or active HIGH. To place the SRAM in the standby mode, it is necessary that all chip enables must be at fixed CMOS levels (not floating) with at least one of them being in the disabled state. For example, for an SRAM with dual chip enables - \overline{CE}_1 (active LOW) and CE_2 (active HIGH) – consider the following:

- Connect SRAM \overline{CE}_1 to the supervisor in the same manner as for the \overline{CE} shown in the previous section.

- Connect SRAM CE_2 directly to the microprocessor I/O if the latter has an I/O dedicated for SRAM controls. Ensure that an external weak pull-down resistor to ground (GND) is available to avoid the input getting biased to an intermediate state in the event of a power failure. See Figure 5 or an illustration of this scheme.
- If no microprocessor I/O is available, connect SRAM CE_2 to its V_{CC} through a weak pull-up resistor (of the order of kilo ohms). Because there is no active device driving this line, there is no risk of current flow in any other direction. This is shown in Figure 6 on page 7.

Many supervisor chips have a *RESET* output that can be used in different ways. This output is driven LOW if the main supply drops below a threshold level (for example, battery mode), while it is driven HIGH when it is above the threshold level (for example, normal mode). This output can be used to drive the microprocessor to trigger power-down or reset-related events. It can also be used to drive the second chip enable, CE_2 , of the SRAM. See Figure 7 on page 7 for this connection scheme.

Based on the application scenario the logic can be extended or modified for the SRAM with multiple chip enables.

Figure 5. SRAM Second Chip Enable Driven by Microprocessor

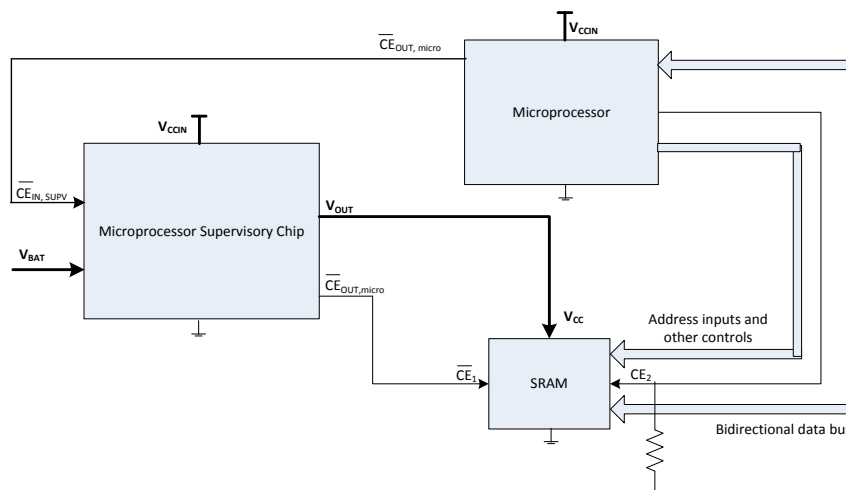


Figure 6. SRAM Second Chip Enable Pulled Up Permanently

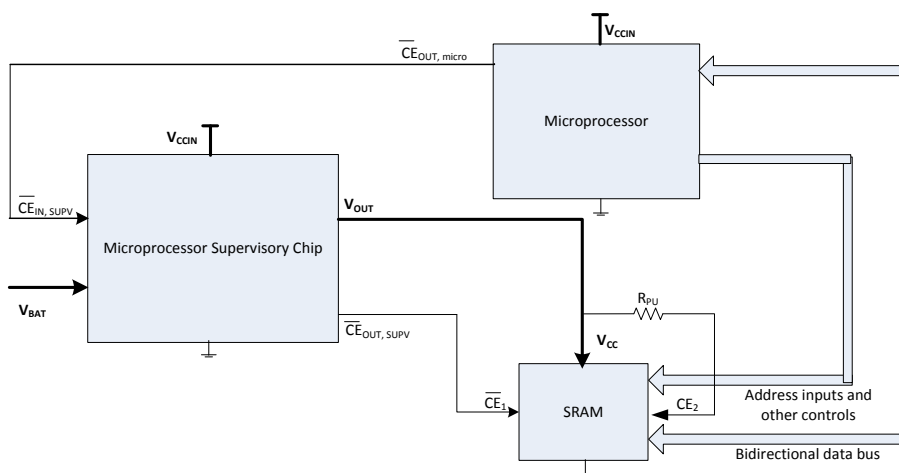
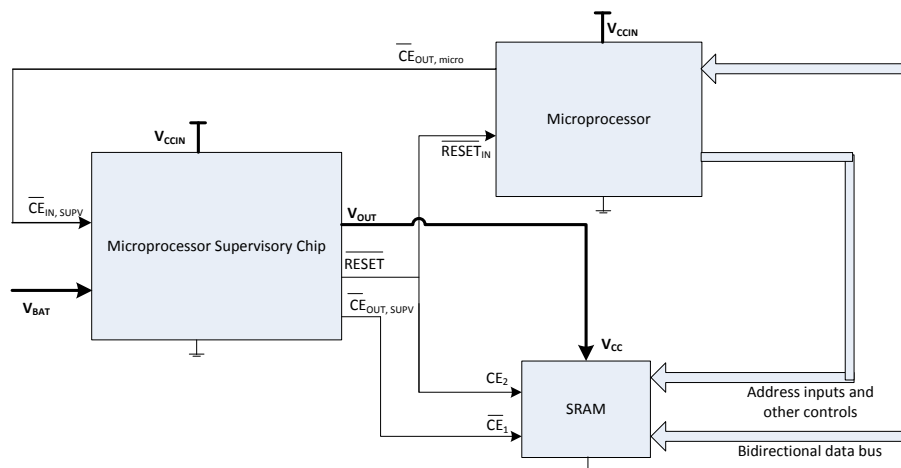


Figure 7. SRAM Second Chip Enable Driven by RESET Pin of Supervisory Chip



5 Alternate Design Technique: Using RESET Monitors

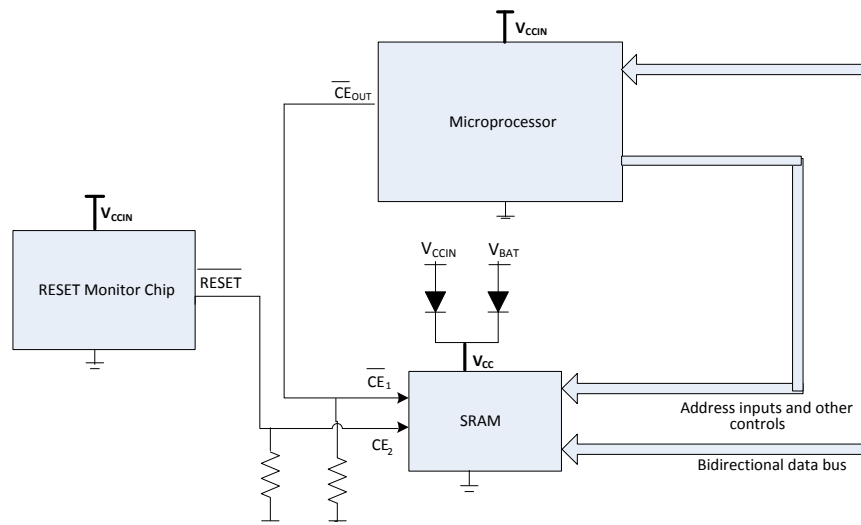
This section introduces the technique of using reset monitors for battery-backed solutions. Reset monitors are similar to supervisors, and are easy to implement with minimal external connections. Typically, a reset monitor chip has three pins - V_{CCIN} , GND and an active low output, \overline{RESET} . The reset monitor chip continuously monitors the supply voltage on its input pin, V_{CCIN} , and provides a reset signal by comparing V_{CCIN} against a set threshold voltage limit. The reset monitor chip normally draws very low supply current, thereby making it suitable for battery-backed applications.

The method discussed in this section is only applicable for dual chip enable SRAMs. Cypress's asynchronous SRAMs can be placed in the standby mode by keeping at least one of the chip-enables in the disable state as stated earlier. The connections for this scheme are shown in [Figure 8](#).

V_{CCIN} and V_{BAT} are power supplies from the mains and battery respectively. The SRAM V_{CC} is equal to V_{CCIN} in the normal mode of operation, and to V_{BAT} in the battery mode. V_{CCIN} or V_{BAT} is selected through the diode connections (similar to the configuration in the Legacy Design Technique).

Chip enable $\overline{CE_1}$ of SRAM, driven by the microprocessor in the normal mode, is pulled low by a weak pull-down resistor connected externally on this pin. The active high CE_2 signal of SRAM is connected to the \overline{RESET} output of the reset monitor chip. The Reset chip monitors the external supply, V_{CCIN} , continuously in the normal mode and maintains output \overline{RESET} high for V_{CCIN} greater than the threshold voltage. It drives \overline{RESET} low when the V_{CCIN} falls below the threshold voltage. When V_{CCIN} drops, a weak pull-down to GND on CE_2 of SRAM avoids this pin going into an unknown (floating) state. Thus, by virtue of CE_2 being LOW, the SRAM is placed in the standby mode.

Figure 8. Battery-Backed SRAM Design using RESET Monitor



There are a few limitations in using this technique:

1. This scheme can be used only for dual chip enable SRAMs
2. The effective voltage supplied to V_{CC} of SRAM is reduced by a diode drop (~ 0.7 V), which makes V_{CC} more susceptible to noise

Hence, it is recommended that a microprocessor supervisory chip be used for your battery-backed application design.

6 Summary

For new designs using Cypress MoBL SRAM in battery-backed applications, use the supervisory techniques as mentioned in this application note.

Document History

Document Title: AN44517 - Design Recommendation for Battery-Backed SRAMs Using Cypress MoBL® SRAMs

Document Number: 001-44517

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	2385026	AJU	04/15/2008	New application note
*A	3162588	AJU	02/04/2011	Replaced all instances of "Micropower" with "MoBL". Updated Design 1: Using Supervisor Chip (Recommended) (included information on dual chip enable SRAMs).
*B	3269754	RAME	06/01/2011	Changed Title and Abstract Changed Design 2 heading to Legacy Design technique Changed Design 1 heading to Recommended Design
*C	3352188	AJU	08/23/2011	Major rewrite in sections, such as Introduction, Recommended Design Technique, Legacy Design Technique, and Other Considerations. Added summary section.
*D	3697827	AJU	07/30/2012	Document title changed to "Design Recommendation for Battery Backed SRAMs Using Cypress MoBL® SRAMs". Updated template according to current Cypress standards.
*E	3800059	MEMJ	11/02/2012	Changed all the instances of \overline{CE}_2 with CE_2 .
*F	4249034	MEMJ	02/14/2014	Introduced an alternate design technique as point (3) on page 2. Added new design technique under section –"Alternate Design Technique –Using RESET monitors". Formatted Figures 4 to 8.
*G	4410986	NILE	06/18/2014	No Updates
*H	5732929	NILE	05/10/2017	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

© Cypress Semiconductor Corporation, 2008-2017. 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. Other names and brands may be claimed as property of their respective owners.