AN1137

Interfacing an External Processor to the SL811HS

Author: Rich Peng
Associated Project: No
Associated Part Family: SL811HS
Software Version: None
Related Application Notes: AN1215

AN1137 describes the method of using an external processor interface with SL811HS in each signal. It also provides the reference schematic designs for six processors.

1 Introduction

The SL811HS is a dual-role capable, that is, host or peripheral, embedded USB controller. It is designed to connect to a variety of external embedded processors ranging from an 8051 to a StrongARM as a memory mapped peripheral. The signal descriptions and transaction methods described here equally apply to the SL811S peripheral only device.

This application note describes the typical methods used to connect the SL811HS/S to an embedded processor. It also describes a typical configuration of support circuitry needed when a USB controller is incorporated into a USB-enabled embedded system. Two configurations are demonstrated including host-only and peripheral only.

2 Signal Basics

The SL811HS/S incorporates an industry-standard address/ data bus. The requirements of the embedded processor signals are laid out in the following list:

- Active LOW CHIP SELECT signal
- Active LOW READ signal
- Active LOW WRITE signal
- Active HIGH INTERRUPT signal
- Address bus or GPIO
- Data bus, at least 8-bits wide
- GPIO to drive various signals such as RESET, USB bus power enable, various resistors and so on. The number of GPIO required is dependent on the controller's configuration. See the schematics later in this document for more information.

2.1 Chip Select (nCS)

nCS is used to enable the SL811HS/S interface and read or write the SL811HS/S registers/memory. nCS essentially signals that the transaction is intended for "this chip" as opposed to another one that might share the same read or write signals. nCS must be asserted by the embedded processor for at least 65 ns during a transaction in order for the transaction to be valid. With some embedded processors this may require that the firmware set an additional number of wait-states so that nCS does not cycle too fast. Wait-state generation is processor dependent, so no information will be given here on how to set additional wait states. If the SL811HS/S is the only IC on the embedded processor's data bus, nCS can be continuously asserted.

2.2 Read (nRD)

nRD is an active LOW signal driven by the embedded processor that is used to signal a register or memory read. Before a read can take place, the desired address to read must be written into the SL811HS/S. During a read nCS must also be asserted in order for the SL811HS/S to recognize the assertion of nRD. The minimum pulse width of the nRD pulse is 65 ns. 65 ns after the assertion of nRD the D[7:0] signals switch from hi-z to driving mode and drive the data bus until 5 ns after nRD is deasserted. The minimum spacing in between nRD assertions is 85 ns.

2.3 Write (nWR)

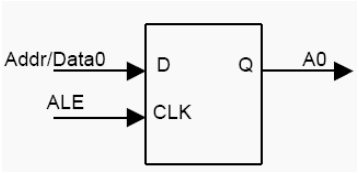
nWR is an active LOW signal driven by the embedded processor that is used to write an address, register, or memory location. In conjunction with nWR, nCS must also be asserted in order for the SL811HS/S to recognize the assertion of the nWR signal. nWR is asserted LOW for a minimum of 65 ns. Data is written from the embedded processor to the SL811HS/S on the rising edge of nWR. The data must remain valid on the bus for 5 ns after nWR is deasserted in order for it to be properly latched by the SL811HS/S. The minimum spacing between nWR assertions is 85 ns.

2.3.1 Address (A_0)

The A_0 signal is driven by the embedded processor and is used in conjunction with the nWR signal to define a write as an address pointer or data. If A_0 is LOW during a write, the write goes to an address pointer register. If A_0 is HIGH, the write goes to a register or memory location pointed to by the address pointer register. For instance, if we want to write to the register at address 00h we would first perform a write with $D_{[7:0]}$ set to 00h and A_0 set LOW. Then we would perform another write with $D_{[7:0]}$ set to the register value and A_0 set HIGH. The value of A_0 must be held for 10 ns after the assertion of nWR in order for the write to be properly recognized. Typically this signal would be connected to address bit 0 on an 8-bit processor, address bit 1 on a 16-bit processor, or address bit 2 on a 32-bit processor.

In some cases the embedded processor may be using an older Intel®-type bus with multiplexed address and data pins, and will typically have an address latch enable (ALE) signal. If this is the case, an external flip-flop will be required to latch the value of the A_0 pin on the ALE edge (edge may depend on the particular processor) as shown in Figure 1.

Figure 1. Using ALE on a Multiplexed Bus



2.4 Interrupt (INTRQ)

The interrupt signal is asserted HIGH by the SL811HS/S during a programmable interrupt event. Some examples may include the completion of a transaction or the connection of a new peripheral device. INTRQ is asserted HIGH until the interrupt event is cleared by writing to the associated interrupt clearing register in the SL811HS/S. The interrupt polarity is not programmable, so an external inverter may be required if a particular processor does not support active HIGH interrupt signaling.

2.4.1 DATA BUS ($D_{[7:0]}$)

The SL811HS/S bidirectional data bus is used to transfer data in and out of registers or memory. The data bus is normally held in a high-impedance state unless both nCS and nRD are asserted during a read transaction. The D[7:0] pins should be connected to the least significant byte of the embedded processor's data bus.

2.5 Reset (NRST)

Reset must be asserted LOW at power-on by the embedded processor or an external POR circuit. NRST is asserted for 16 clock cycles of the CLK signal. Further transactions with the SL811HS/S should not take place before 16 cycles of CLK after NRST is deasserted.

2.6 Role (M/S)

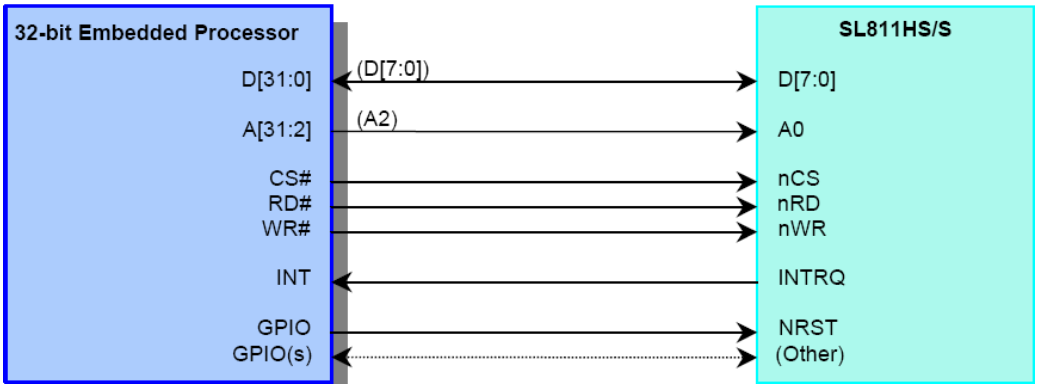
This signal determines the operating role for the SL811HS at the assertion of an external reset. At the assertion of NRST the value of the M/S pin is latched into the internal M/S register bit. If M/S is held LOW, the SL811HS acts as a USB host. If M/S is held HIGH, the SL811HS is a USB peripheral. During normal operation the M/S bit does not have any effect on the operation of the SL811HS. The operating role of the SL811HS may be changed without external reset by software running on the embedded processor that changes the SL811HS internal M/S register bit.

2.7 DMA Signals (nDACK, nDRQ)

These peripheral-only DMA related signals are typically not used with an embedded processor so they are not described here. Please see the SL811HS/S data sheet for more details on nDACK and NDRQ. The SL811HS/S I/Os are 5-volt tolerant, but will only drive its own I/Os to 3.3 volts. As long the embedded processor has TTL or 3.3-volt CMOS level inputs the SL811HS/S interface should not require voltage translation buffering.

Figure 2 shows a typical example of the connection of a SL811HS/S to generic embedded processor bus.

Figure 2. Example Connection of a 32-bit Embedded Processor to the SL811HS/S



3 Example Transactions

Two example transactions are shown in Figure 3 and Figure 4. Figure 3 shows a simple write transaction where a register or memory location is being written to. Figure 4 shows a simple read transaction of a register or memory location.

Figure 3. Example Write Transaction (not to scale)

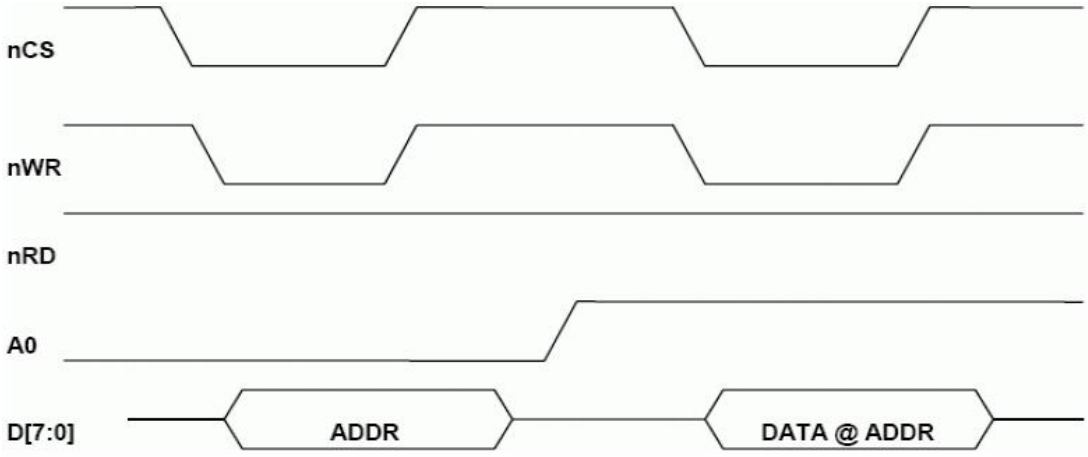
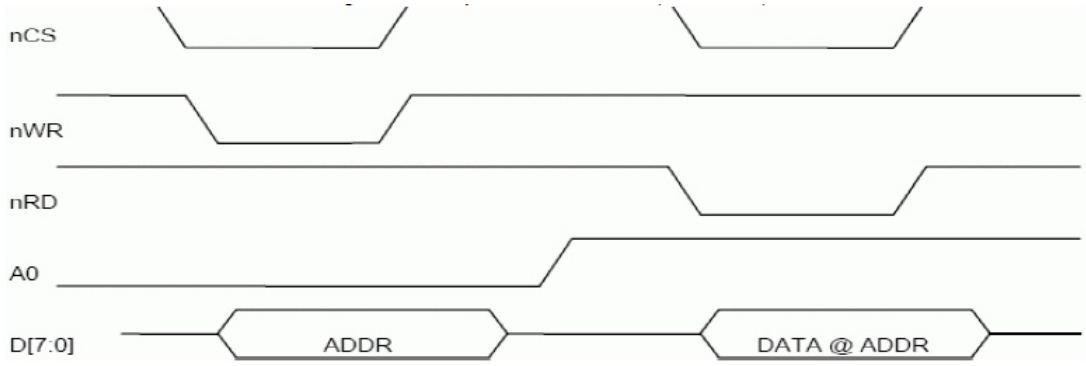


Figure 4. Example Read Transaction (not to scale)



4 Example SL811HS/S Circuits

Two typical circuit configurations are shown in Figure 5 and Figure 6. The 48-pin version of the SL811HS is shown in each of the figures; however the same pin configurations apply to the 28-pin package. Figure 5 represents a typical USB embedded host with power protection and all associated components. The clock may be supplied from a 3.3-volt 12-/48-MHz CMOS oscillator or a 12-/48-MHz crystal. The chosen clock source must meet the jitter and accuracy requirements of the USB 2.0 specification, meaning that driving the clock from an external processor timer/counter output may or may not be possible. Reset is generated via a GPIO on the embedded processor. Reset could also be generated from a dedicated POR circuit.

Figure 6 shows a typical USB peripheral configuration for the SL811HS or SL811S (minus the M/S pin). In this situation the USB data line pull-up resistor must be able to be disabled while power is disconnected; therefore two GPIO are required on the embedded processor. In Figure 4 the pull-up resistor is connected to D+, meaning that the circuit is configured for full-speed USB operation. If low-speed USB operation is desired, the pull up resistor should be connected to the D– signal. Total capacitance on the USB Vbus must be below 10 µF to meet the USB 2.0 specification. A voltage regulator that converts the Vbus 5 volts to 3.3 volts is not shown in the schematic, but may be required if the device is bus powered.

Figure 5. SL811HS in a Typical USB Host Configuration

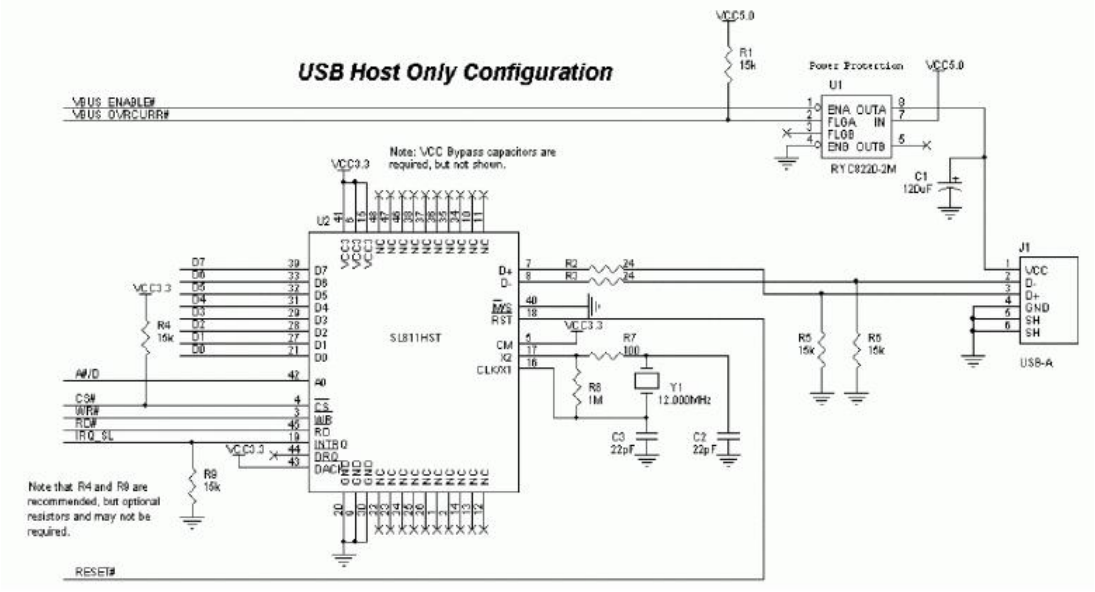
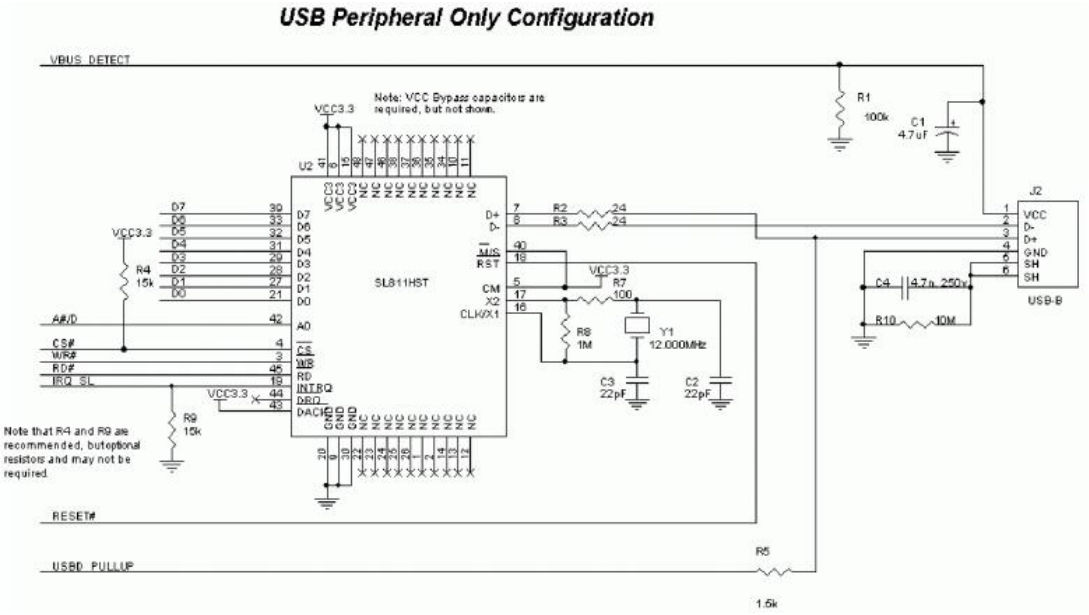


Figure 6. SL811HS/S in a Typical USB Peripheral Configuration



5 Hardware Reference Design

5.1 Interfaced ICs

A variety of microcontrollers and microprocessors are interfaced using simple "glue logic". Some chips may require a couple of gates to separate the R/W line into /Rd and /Wr line. Some of them are:

- EP7312-208LQFP
- ARM940
- H8S/2328-TFP120
- SA1110
- MPC855T
- TMS320VC33

5.1.1 EP7312

EP7312 is high-performance, low-power SoC with SDRAM & enhanced digital audio interface from Cirrus Logic®. It has an ARM®720T processor at its core. Its advanced audio decoder/decompression capability supports bit streams with adaptive bit rates, featuring multiple audio decompression algorithms (MP3, WMA, AAC, etc.) It is targeted for ultra-low-power portable and line-powered applications such as portable consumer entertainment devices, home and car audio juke box systems, and devices that feature the added capability of digital audio compression/decompression. Refer to AN1137.zip folder for schematics of the reference design to interface SL811HS with EP7312.

5.1.2 ARM940

ARM940 is one of the ARM9TDMI family of general-purpose microprocessors with core plus cache and protection unit from ARM. It is a Harvard architecture device implemented using a five stage pipeline. Its standalone core can be embedded into more complex devices which has simple bus interface, allowing design of flexible, user-defined caches and memory systems around the core. It supports both 32-bit ARM and 16-bit Thumb instruction sets, allowing tradeoff between high performance and high code density. Refer to AN1137.zip folder for schematics of the reference design to interface SL811HS with ARM940.

5.1.3 H8S/2328

H8S/2328 is a high-performance microcontroller from Renesas with internal 32-bit H8S/2000 CPU core and a set of on-chip supporting functions for system configuration, which supports a concise, optimized instruction set. It features 16-bit timer pulse unit, programmable pulse generator, 8-bit timer, watchdog timer, serial communication interface, A/D converter, D/A converter, on-chip DMA controller and data transfer controller, enabling high-speed data transfer without CPU intervention. Refer to AN1137.zip folder for schematics of the reference design to interface SL811HS with H8S_2328.

5.1.4 SA1110

SA1110 is the Intel® StrongARM microprocessor. It is highly integrated communications microcontroller that incorporates a 32-bit StrongARM RISC processor core, system support logic, multiple communication channels, LCD controller, memory and PCMCIA controller, and twenty-eight general-purpose I/O ports. It features 230 Kbps UART, and ports for touch-screen, audio, telecom, Infrared data, and synchronous serial (SPI, UCB110, …). It has wide range of applications as it is a powerful general-purpose microprocessor. Refer to AN1137.zip folder for schematics of the reference design to interface SL811HS with SA1110.

5.1.5 MPC855T

MPC855T is a versatile one-chip integrated microprocessor from Freescale™ Semiconductor designed for lower cost access equipment that requires fast Ethernet support capable of 100 Mbps. It combines MPC8xx core with Freescale's Communication Processor Module, an independent RISC engine specifically designed to offload communication tasks. It is capable to run at 105 Machine Instructions per second (105 MIPS) featuring up to 32-bit data bus (dynamic 8, 16, 32 bits), 32-bit address bus, four baud rate generators, one serial communication controller, two serial management channels, one SPI port, one I2C port, PCMCIA Interface and Debug Interface. More details are available in the datasheet. Refer to AN1137.zip folder for schematics of the reference design to interface SL811HS with MPC855T.

5.1.6 TMS320VC33

TMS320VC33 is Texas Instrument's TMS320C3x generation, high performance, floating-point Digital Signal Processor. It is capable of 16/32-bit integer operations and 32/40-bit floating-point operations featuring parallel arithmetic-logic unit/multiplier execution in single cycle. It has simple interface to I/O and memory devices, 32-bit instructions, 24-bit addresses, and EDGEMODE selectable external interrupts. The special instruction set and internal bus gives it the speed and flexibility to execute up to 150 million floating-point operations per second (MFLOPS). It greatly enhances general purpose applications. More details are available in the datasheet. Refer to AN1137.zip folder for schematics of the reference design to interface SL811HS with TMS320VC33.

Note Refer the SL811HS datasheet – 38-08008 for read/write waveforms and timing parameters.

6 Summary

The SL811HS/S offers an easy-to-use interface for embedded processors. The use of standard data bus signals and 5-volt tolerant I/Os allow the SL811HS/S to connect to most embedded processors without glue-logic. For further questions and assistance please contact Cypress USB applications support.

About the Author

Name:	Rich Peng
Title:	Product Apps Group Lead

Document History

Document Title: AN1137 - Interfacing an External Processor to the SL811HS

Document Number: 001-43147

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	1779366	LIP	11/28/2007	Re-catalogued application note.
*A	3180128	LIP	02/23/2011	Template updates. Included section “Hardware Reference Design”.
*B	3217954	LIP	04/06/2011	Updated section “Hardware Reference Design” to include more information.
*C	3711279	LIP	08/22/2012	Updated Abstract. Updated Introduction. Updated in new template.
*D	4882301	LIP	09/10/2015	Updated template
*E	5824234	AESATMP8	07/19/2017	Updated logo and Copyright.

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, 2007-2017. 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.