The following document contains information on Cypress products. The document has the series name, product name, and ordering part numbering with the prefix “MB”. However, Cypress will offer these products to new and existing customers with the series name, product name, and ordering part number with the prefix “CY”.

How to Check the Ordering Part Number

1. Go to www.cypress.com/pcn.
2. Enter the keyword (for example, ordering part number) in the **SEARCH PCNS** field and click **Apply**.
3. Click the corresponding title from the search results.
4. Download the Affected Parts List file, which has details of all changes

For More Information

Please contact your local sales office for additional information about Cypress products and solutions.

About Cypress

Cypress is the leader in advanced embedded system solutions for the world's most innovative automotive, industrial, smart home appliances, consumer electronics and medical products. Cypress' microcontrollers, analog ICs, wireless and USB-based connectivity solutions and reliable, high-performance memories help engineers design differentiated products and get them to market first. Cypress is committed to providing customers with the best support and development resources on the planet enabling them to disrupt markets by creating new product categories in record time. To learn more, go to www.cypress.com.

AN205273

F²MC - 8FX Family, MB95200H/210H Series, I/O Port

This application note describes how to use IO Port. The application note describes the function and usage. Please note that in this manual each port number is given with a 1-digit placeholder. “x. yz” always means the bit position 0 - 7.

Contents

1 Introduction....1
2 I/O Port....2
2.1 Key Features....2
2.2 Block Diagram....2
2.3 Registers....3
2.4 Input Mode....4
2.5 Pull-up Register....4
2.6 Output Mode....5
3 Port Input / Unused Pins....6
3.1 Port Input / Unused Pins....6
4 Technical information....7
4.1 Hysteresis Input....7
5 Tips and Tricks....8
5.1 Initial Value....8
5.2 Byte Instructions....8
5.3 RMW Instructions....8
6 Appendix....9
6.1 Sample Code....9
7 Additional Information....10

1 Introduction

This application note describes how to use IO Port.

The application note describes the function and usage.

Please note that in this manual each port number is given with a 1-digit placeholder.

“x. yz” always means the bit position 0 - 7.

For example, “PDR0_P03” means Port 0 Bit 3.

2 I/O Port

This chapter describes the basic functions of I/O port

2.1 Key Features

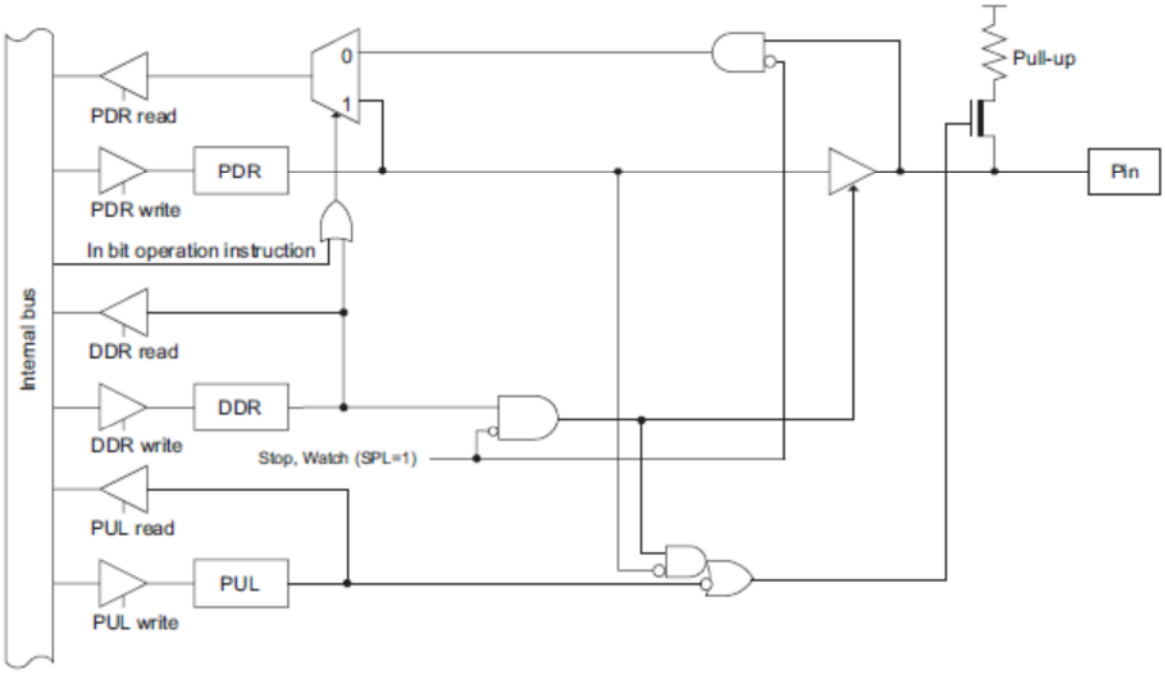
- Operation as an output or an input port
- Operation as a peripheral function output or input
- Operation of pull-up control register
- Operation of input level selection register

2.2 Block Diagram

Figure 2-1 shows the internal block diagram of an external I/O Port (Port G).

Up to 8 I/O-pins can be controlled within one register. The registers are described as follows.

Figure 1. Block Diagram of Port G



Please note that the block diagram only contains the basic functions of a general I/O Port, such as input, output and pull-up resister.

2.3 Registers

Please refer to Chapter 9 in Hardware Manual of MB95200H Series for detailed register setting.

2.3.1 Port Data Register (PDR)

This register contains the data bits, if the corresponding port acts as a simple digital output. The values are output values, if the Port Direction Register is set to output mode.

Please note that a resource output control bit overwrites the bit of PDR register bit.

PDRx_Pyz	Pin Function
0	Pin state low (VSS)
1	Pin state high (VDD)

2.3.2 Data Direction Register (DDR)

This register contains the bit information of the corresponding pins if they act as input or output.

DDRx_Pyz	Peripheral Function Output	Pin Function
0	Disable	Port Input
1	Disable	Port Output
invalid	Enable	Peripheral Function Output

2.3.3 Pull-up Control Register (PUL)

This register connects an internal pull-up resistor to a port pin.

PULx_Pyz	Pull-up Resistor
0	Disable
1	Enable

Please see Data Sheet for the resistor value.

2.3.4 Input level Selection Register (ILSR)

With this register, one of the following input levels can be chosen.

ILSR	Input Level	VIL	VIH
0x04	CMOS	0.3 V_{CC}	0.7 V_{CC}
0x00	Hysteresis	0.3 V_{CC}	0.7 V_{CC}

Please note that only PDR0_P04 has this function.

2.4 Input Mode

In general, if a pin acts as an input, the corresponding bit in the Data Direction Register (2.3.2) should be set to “0”.

If the connected external source may change to high-Z state, please use an external pull-up or pull-down resistor or set the corresponding bit in the Pull-up Register (2.3.3).

There are three kinds of input modes as follows:

- Digital Input means the port is used as a general I/O.
- ADC Input means the port is only used for analog input function.
- Peripheral function Input means the port is used as a peripheral input such as external interrupt input.

2.4.1 Digital Port Input or Peripheral Function Input (except ADC)

The following register settings shall be done, if a pin acts as a digital port or peripheral function input.

```
DDRx_Pyz = 0;
AIDRL_Pyz = 1;      // Only if pin shares ADC input
```

2.4.2 ADC Input

The following register settings shall be done, if a pin acts as ADC input.

```
DDRx_Pyz = 0;
AIDRL_Pyz = 0;      // Only if pin shares ADC input
```

2.5 Pull-up Register

The P0 and PG ports in input mode may enable an internal pull-up resistor (about 50 KΩ, please see Data Sheet for the exact value) by programming the pull-up register (2.3.3).

The initial value of “0” disconnects the internal pull-up resistor, writing “1” to the corresponding bit in the PULx enables the resistor.

If the port-pin is used as an output, the value of the register bit has no meaning and the pull-up resistor is disabled. (Exceptions: For UART output SOT, the internal pull-up resistor can be used if not provided by line driver).

Enabled pull-up resistors will be disabled while the microcontroller is in stop mode or timer mode.

The resistor is also disabled if the pin is used as ADC input.

For a pin shared with other peripheral functions, disable the output of such peripheral functions. In addition, set the corresponding bit in the PUL register to “0”.

.

2.6 Output Mode

There are two kinds of output modes as follows:

- Digital output means the port is used as a general I/O.
- Peripheral function output means the port is used as peripheral resource output such as 8/16-bit composite timer's output.

2.6.1 Digital Port Output

The following register settings shall be done, if a pin acts as a digital port output.

```
DDRx_Pyz = 1;
PDRx_Pyx = [pin output state];
[Resource-Output enable] = 0;    // disable resource output
```

Note that the resource output enable bit, which may share a port pin, shall be disabled for digital port output function.

2.6.2 Resource Output

```
[Resource-Output enable] = 1;    // enable resource output
```

The following register settings shall be done, if a pin acts as a peripheral output.

3 Port Input / Unused Pins

This chapter describes how to connect input port pins and how to proceed with unused pins

3.1 Port Input / Unused Pins

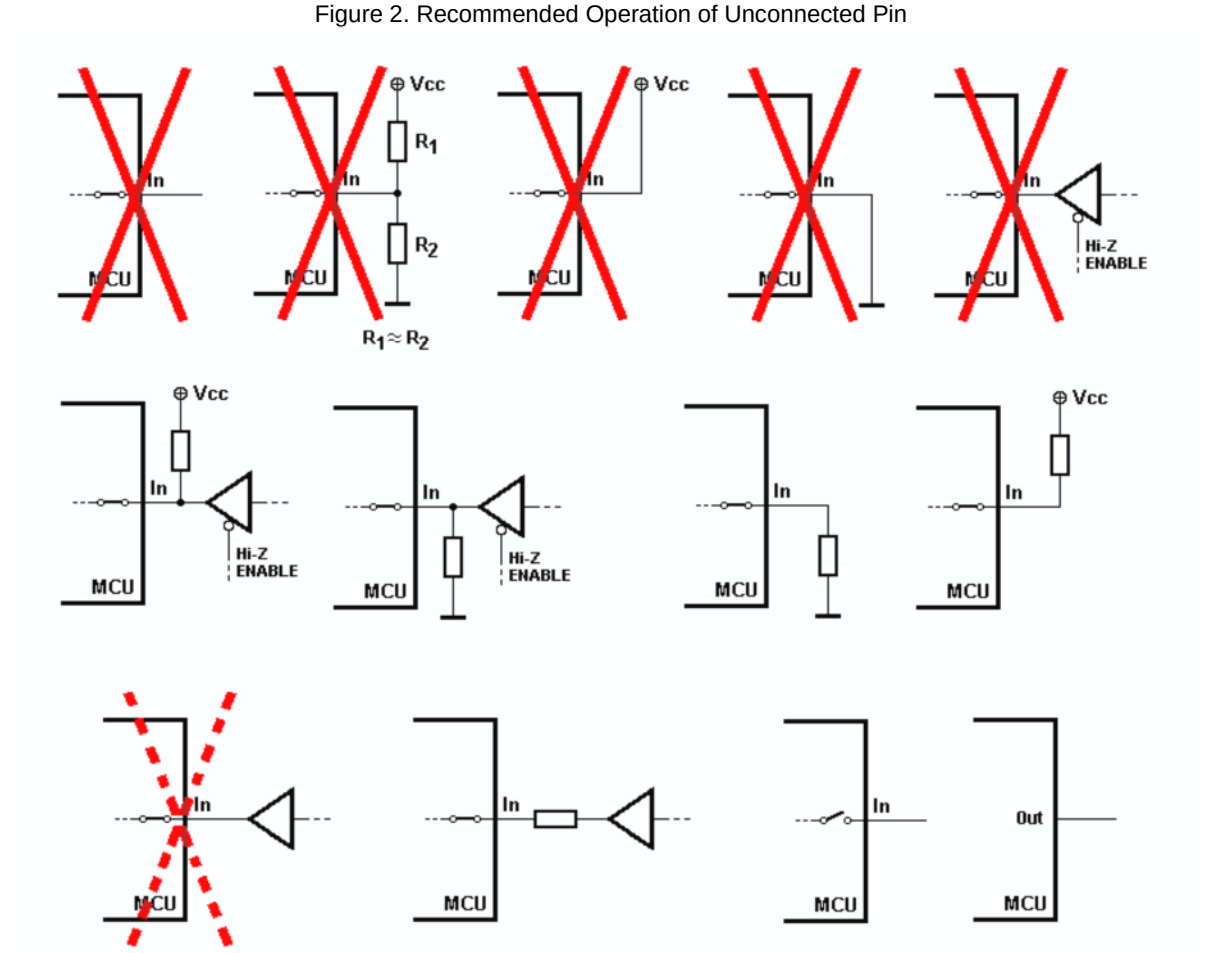
It is strongly recommended not to leave the pins unconnected, while they are switched to input and are enabled. Otherwise, these pins may enter a so-called floating state. This may cause a high I_{CC} current, which is adverse to low power consumption modes, and may cause a damage of the MCU.

In this case, use the internal pull-up resistors or use external pull-up or pull-down resistors to define the input-level.

It is recommended to set the port input enable bit to “0”, if a port pin is unconnected.

Never connect a potential divider with almost the same resistor values.

Figure 2. Recommended Operation of Unconnected Pin



Take care of the connection of input pins with other devices, which may enter into High-Z state. Always use internal pull-up or external pull-up or pull-down resistors in this case.

Outputs from external circuits should always be connected via a serial resistor to a MCU input pin, in order to prevent latch-up caused by undershoots or overshoots.

Debouncing and decoupling capacitors should be chosen as small as possible.

All pins are set to input enabled (i.e. corresponding DDR bits are set to “0”), after power-on reset.

Do not connect any input port directly to V_{CC} or V_{SS} (GND)! Always use pull-up or pull-down resistors.

4 Technical information

This chapter describes electrical characteristics related to hysteresis input

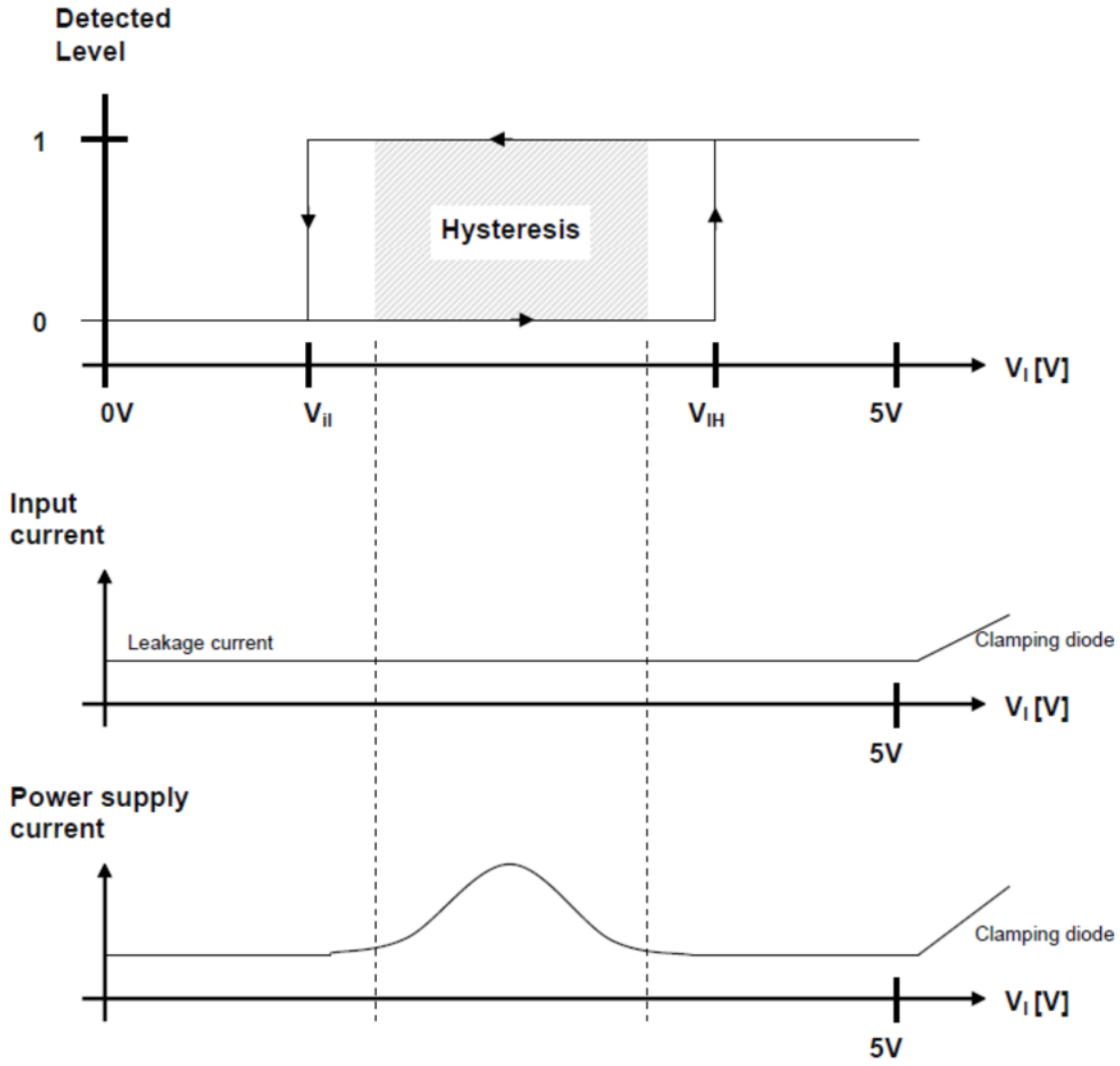
4.1 Hysteresis Input

A hysteresis indicates that the behaviors of input levels, at which '1' is detected and the level at which"0" is detected, are different.

Please refer to the section "Electrical Characteristics" in Data Sheet of MB95200H/210H series for detailed description.

Kindly note that the power supply current (i.e. the power consumption) of the device may increase, while the input voltage is within the hysteresis area. However, the input current of I/O pin remains constant.

Figure 3. Power Consumption Level on Hysteresis Input



VIL Port hysteresis input level (low-level) specified in Data Sheet

VIH Port hysteresis input level (high-level) specified in Data Sheet

Hysteresis area

5 Tips and Tricks

This chapter gives some hints on using I/O port.

5.1 Initial Value

Take care that the port data is defined before the pin direction is changed to output.

Otherwise, the undefined data might be output to I/O pin, until PDRx is written.

```
PDRx = initial_value;      // define initial value before port is set to
                           output
DDRx = 0xff;               // set port x to output, after initial value is
```

5.2 Byte Instructions

Instead of bit instructions, use byte-instructions which will be executed faster.

5.3 RMW Instructions

Accessing to the Port Data Register (2.3.1) via a read-modify-write instruction always returns register value during a read cycle (of the same read-modify-write instruction), no matter whether the resource output of the corresponding pin is enabled or not.

6 Appendix

6.1 Sample Code

6.1.1 Project Name: Digital_IO

I/O Port is used as a digital input and output

main.c

```
#include “mb95200.h”
/*-----------------------------------------------------------------------
   name: Init();
   function: initial I/O Port as digital input or output
-----------------------------------------------------------------------*/
void Init (void)
{
       PDR0_P00 = 1;
       DDR0_P00 = 1;// set P0_P00 output
       PDR0_P01 = 0;
       AIDRL_P01 = 1;      // disable AN1 input
       DDR0_P01 = 0;// set P0_P01 input

       PDR6_P62 = 1;
       DDR6_P62 = 1;// set P6_P62 output
       PDR6_P63 = 0;
       DDR6_P63 = 0;// set P6_P63 input
}
/*-----------------------------------------------------------------------
   name: main ();
   function: main loop
-----------------------------------------------------------------------*/
void main (void)
{
       Init ();
       while(1);
}
```

6.1.2 Project Name: ADC_Input

I/O Port is used as analog input

main.c

```
/*-----------------------------------------------------------------------
   name: Init();
   function: initial I/O Port as analog input
-----------------------------------------------------------------------*/
void Init (void)
{
       DDR0_P01 = 0;// P0_P01 inputs
       AIDRL_P01 = 0;      // enable AN1 input
}
/*-----------------------------------------------------------------------
   name: main ();
   function: main loop
-----------------------------------------------------------------------*/
void main (void)
{
       Init ();
       while(1);
}
```

6.1.3 Project Name: Peripheral_Output

IO Port is used as Peripheral Output (e.g. 8/16-bit composite timer)

main.c

```
#include "mb95200.h"
/*-----------------------------------------------------------------------------
   name: InitCompTimer();
   function: initial timer for interval timer (One-shot) function
-----------------------------------------------------------------------------*/
void InitCompTimer (void)
{
       T00DR = 0xFF;       // set count value
       TMCR0 = 0x00;       // 8-bit, no filtering
       T00CR0 = 0x00;             // interval timer with one-shot mode
       T00CR1 = 0x00;             // disable interrupt
       T00CR1_OE = 1;             // enable output
}
/*-----------------------------------------------------------------------------
   name: main();
   function: main loop, P0_P05 will be output while counter value
                    matches with pre-set value
-----------------------------------------------------------------------------*/
void main(void)
{
       InitCompTimer();

       T00CR1_STA = 1;            // start timer
       while(1)
       {
              while (T00CR1_STA == 0)
              {
                     T00CR1_STA = 1;
              }
       }
}
```

7 Additional Information

For more information on Cypress Microcontrollers Products, please visit the following websites:

http://www.cypress.com/cypress-microcontrollers

http://www.cypress.com/cypress-mcu-product-softwareexamples

Document History

Document Title: AN205273 – F²MC - 8FX Family, MB95200H/210H Series, I/O Port

Document Number: 002-05273

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	-	Levi, Zhang	20/03/2008	V1.0, First Draft
			18/07/2008	V1.1, Delete description about I^2C and external bus in 2.5 Pull-up register; add URL in chapter 6 Additional Information.
*A	5261592	HUAL	05/06/2016	Migrated Spansion Application Note “MCU-AN-500003-E-11” to Cypress format.

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
Lighting & Power Control	cypress.com/powerpsoc
Memory	cypress.com/memory
PSoC	cypress.com/psoc
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless/RF	cypress.com/wireless

PSoC® Solutions

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

Cypress Developer Community

Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

PSoC is a registered trademark and PSoC Creator is a trademark of Cypress Semiconductor Corporation. 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

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com

© Cypress Semiconductor Corporation,2008-2016. 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, 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.