AN205554

F²MC-16FX Family Series With Source Clock Timers

This application note describes the functionality of the Source Clock Timers and gives some examples.

Contents

1 Introduction....1
1.1 Key Features....1
2 The Source Clock Timers....1
2.1 Source Clock Timers....1
2.2 Registers....5
3 Source Clock Timers Example....8
3.1 Source Clock Timers Interrupt....8
4 Additional Information....9
5 Document History....10

1 Introduction

This application note describes the functionality of the Source Clock Timers and gives some examples.

1.1 Key Features

- 3 different timers: Main Clock, Sub Clock and RC Clock Timer
- Interrupt generation in selectable intervals
  - For RC Clock between 2^8/CLKRC and 2^{23}/CLKRC
  - For Main Clock between 2^8/CLKMC and 2^{23}/CLKMC
  - For Sub Clock between 2^{10}/CLKSC and 2^{17}/CLKSC
- All timers can be reset via software

2 The Source Clock Timers

The Basic Functionality of the Source Clock Timers

2.1 Source Clock Timers

There are three Clock Timers available:

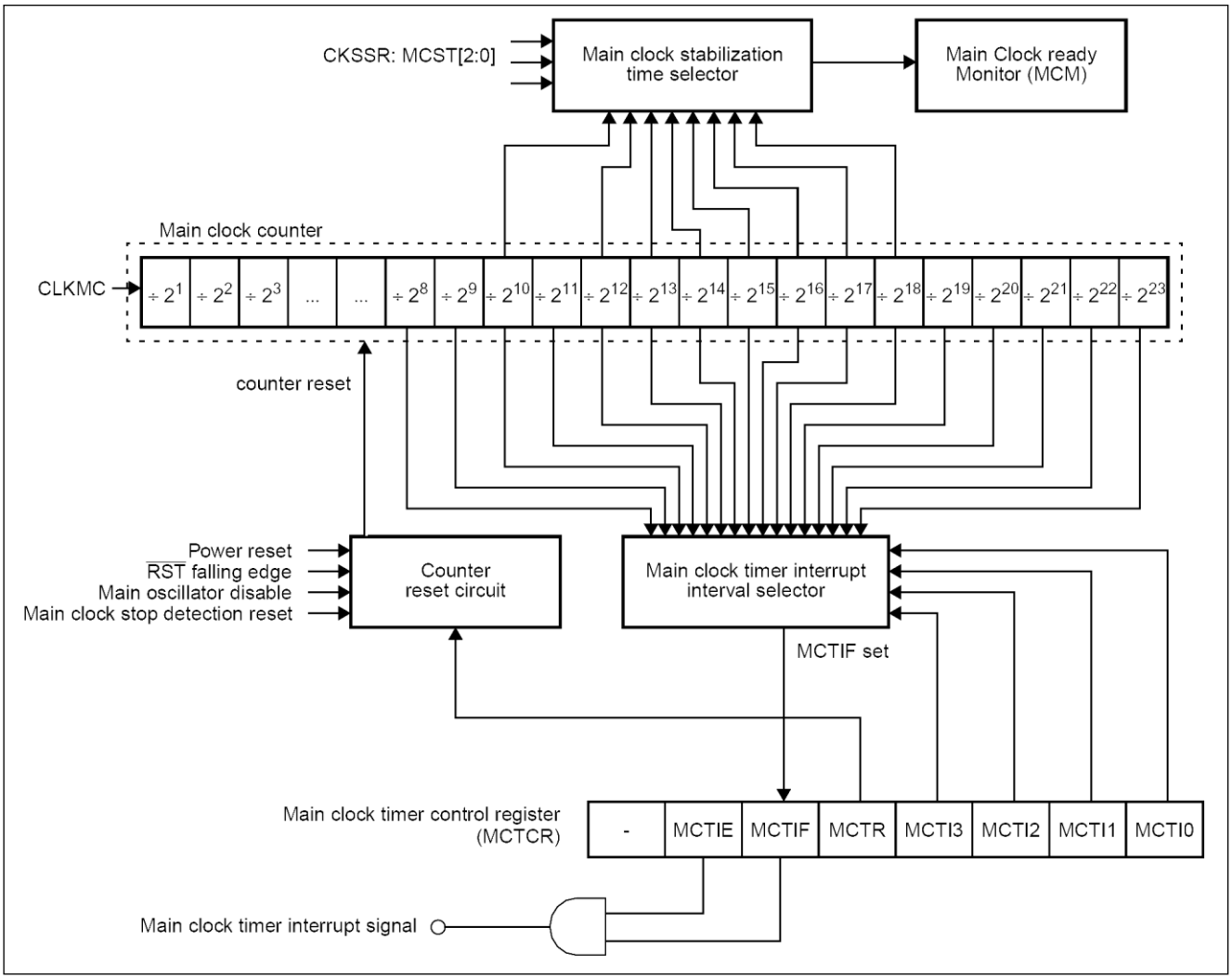
- Main Clock Timer
- Sub Clock Timer
- RC Clock Timer

With these timers the user can generate cyclic interrupts with an interval specified by a dedicated divider. This can be used to wake-up from a Timer standby-mode periodically. Especially when using a Sub Clock with an oscillation frequency of 32.768 kHz, exactly 1 s intervals (and fractions and multiples thereof) can be implemented easily.

2.1.1 Main Clock Timer Block Diagram

Figure 1 shows the internal block diagram of the Main Clock Timer.

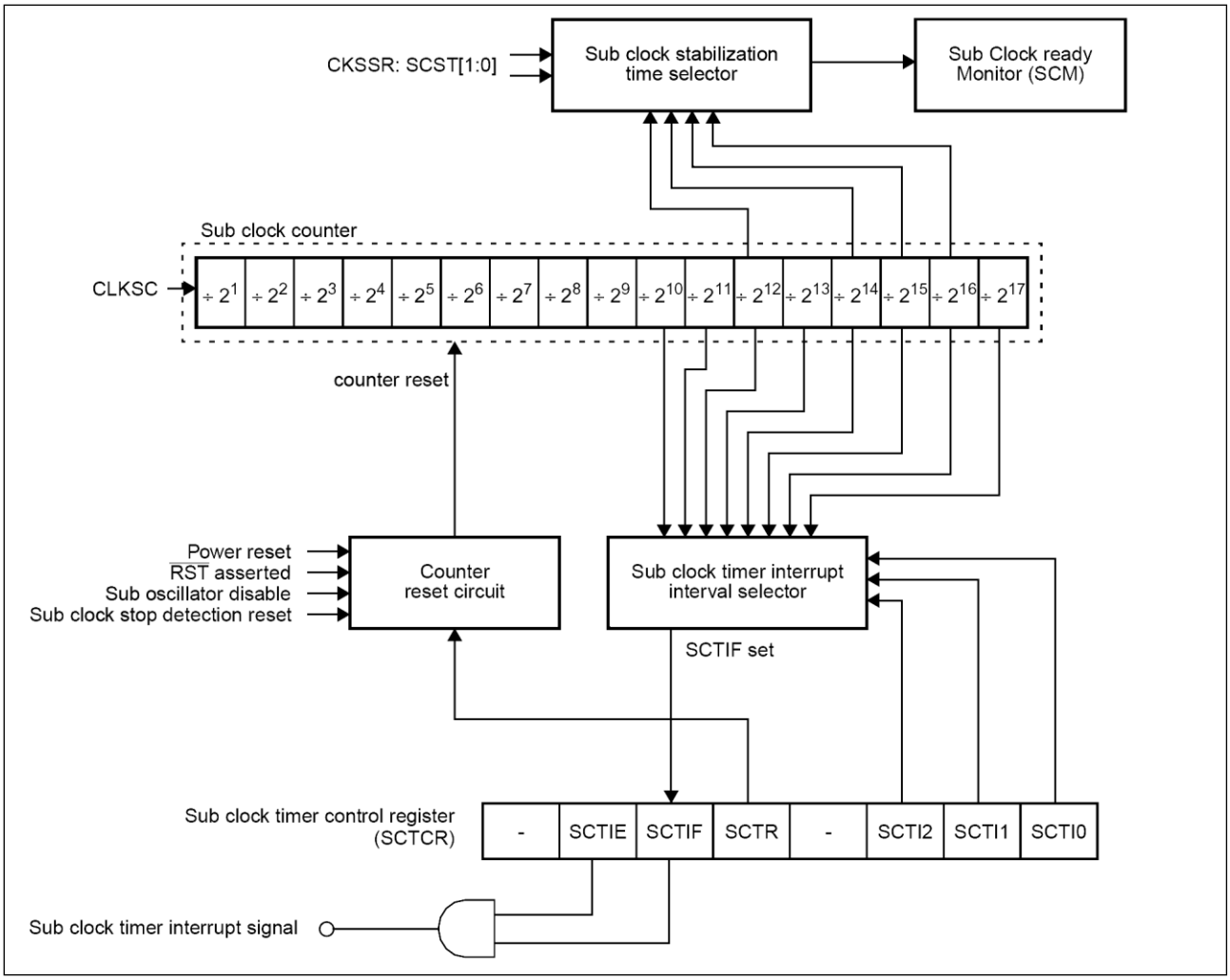
Figure 1. Main Clock Timer Block Diagram



2.1.2 Sub Clock Timer Block Diagram

Figure 2 shows the internal block diagram of the Sub Clock Timer.

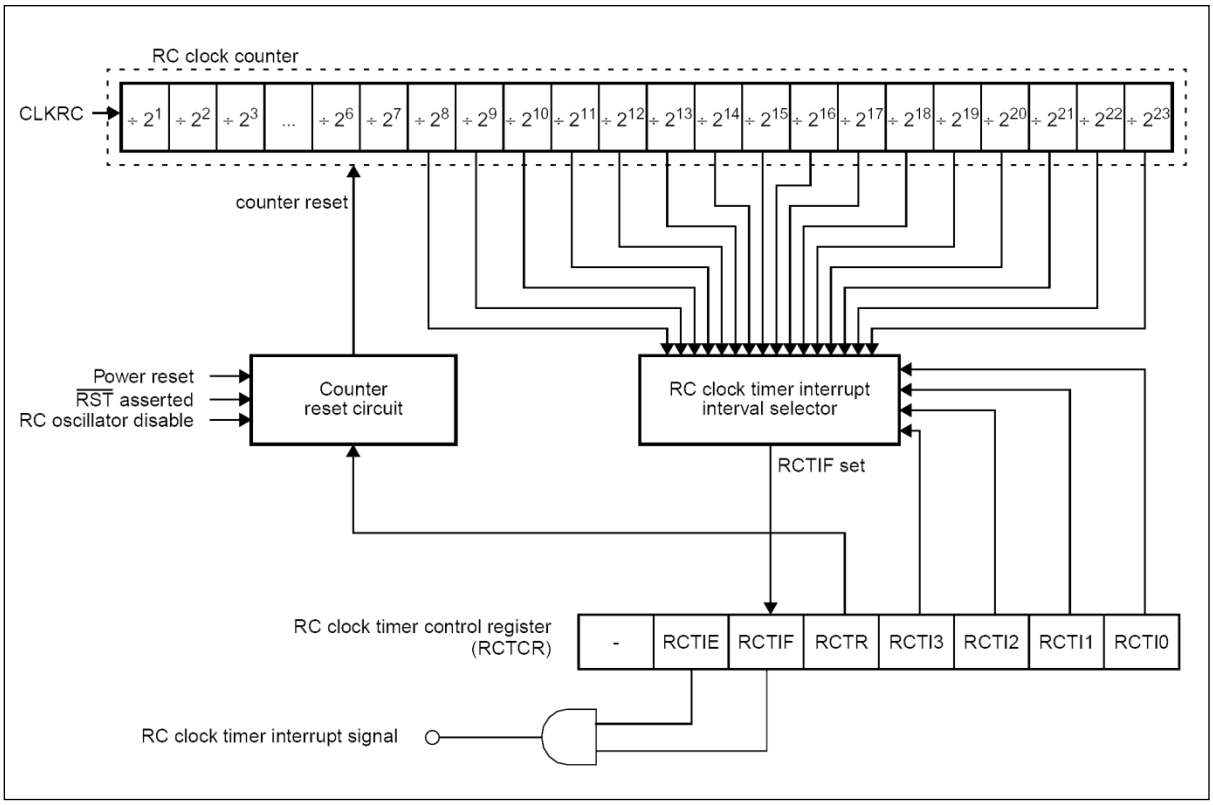
Figure 2. Sub Clock Timer Block Diagram



2.1.3 RC Clock Timer Block Diagram

Figure 3 shows the internal block diagram of the RC Clock Timer.

Figure 3. RC Clock Timer Block Diagram



2.2 Registers

2.2.1 Main Clock Timer Control Register (MCTCR)

This register controls Main clock timer's interrupt interval and also its reset function.

Table 1. MCTCR

Bit No.	Name	Explanation	Initial Value	Value	Operation
15	-	*Reserved*	X	0	Always write "0" to this Bit
14	MCTIE	Interrupt Enable	0	0	Timer interrupt disabled
				1	Timer interrupt enabled
13	MCTIF	Interrupt Flag	0	0	Read: No Interrupt Write: Clear Interrupt
				1	Read: Interrupt requested Write: No Effect
12	MCTR	Main Clock Timer Reset	1	0	Read: Always read 1 Write: Clear Main clock timer
				1	Read: Always read 1 Write: No Effect
11, 10, 9, 8	MCTI3 … MCTI0	Interrupt Interval	0, 0, 0, 0	0, 0, 0, 0	2^8 / CLKMC (~ 64 µs*)
				0, 0, 0, 1	2^9 / CLKMC (~ 128 µs*)
				0, 0, 1, 0	2^{10} / CLKMC (~ 256 µs*)
				0, 0, 1, 1	2^{11} / CLKMC (~ 512 µs*)
				0, 1, 0, 0	2^{12} / CLKMC (~ 1 ms*)
				0, 1, 0, 1	2^{13} / CLKMC (~ 2 ms*)
				0, 1, 1, 0	2^{14} / CLKMC (~ 4 ms*)
				0, 1, 1, 1	2^{15} / CLKMC (~ 8 ms*)
				1, 0, 0, 0	2^{16} / CLKMC (~ 16 ms*)
				1, 0, 0, 1	2^{17} / CLKMC (~ 33 ms*)
				1, 0, 1, 0	2^{18} / CLKMC (~ 66 ms*)
				1, 0, 1, 1	2^{19} / CLKMC (~ 131 ms*)
				1, 1, 0, 0	2^{20} / CLKMC (~ 262 ms*)
				1, 1, 0, 1	2^{21} / CLKMC (~ 524 ms*)
				1, 1, 1, 0	2^{22} / CLKMC (~ 1049 ms*)
				1, 1, 1, 1	2^{23} / CLKMC (~ 2097 ms*)

* These time intervals are calculated for 4 MHz Main Clock frequency.

2.2.2 Sub Clock Timer Control Register (SCTCR)

This register controls Sub clock timer's interrupt interval and also its reset function.

Table 2. SCTCR

Bit No.	Name	Explanation	Initial Value	Value	Operation
7	-	*Reserved*	X	0	Always write "0" to this Bit
6	SCTIE	Interrupt Enable	0	0	Timer interrupt disabled
				1	Timer interrupt enabled
5	SCTIF	Interrupt Flag	0	0	Read: No Interrupt Write: Clear Interrupt
				1	Read: Interrupt requested Write: No Effect
4	SCTR	Sub Clock Timer Reset	1	0	Read: Always read 1 Write: Clear Sub clock timer
				1	Read: Always read 1 Write: No Effect
3	-	*Reserved*	X	0	Always write "0" to this Bit
2, 1, 0	SCTI2 … SCTI0	Interrupt Interval	0, 0, 0	0, 0, 0	2^{10} / CLKMC (~ 31.25 ms*)
				0, 0, 1	2^{11} / CLKMC (~ 62.5 ms*)
				0, 1, 0	2^{12} / CLKMC (~ 125 ms*)
				0, 1, 1	2^{13} / CLKMC (~ 250 ms*)
				1, 0, 0	2^{14} / CLKMC (~ 500 ms*)
				1, 0, 1	2^{15} / CLKMC (~ 1 s*)
				1, 1, 0	2^{16} / CLKMC (~ 2 s*)
				1, 1, 1	2^{17} / CLKMC (~ 4 s*)

* These time intervals are calculated for 32.768 kHz Sub Clock frequency.

2.2.3 RC Clock Timer Control Register (RCTCR)

This register controls RC clock timer's interrupt interval and also its reset function.

Table 3.RCTCR

Bit No.	Name	Explanation	Initial Value	Value	Operation
15	-	*Reserved*	X	0	Always write "0" to this Bit
14	RCTIE	Interrupt Enable	0	0	Timer interrupt disabled
				1	Timer interrupt disabled
13	RCTIF	Interrupt Flag	0	0	Read: No Interrupt Write: Clear Interrupt
				1	Read: Interrupt requested Write: No Effect
12	RCTR	Main Clock Timer Reset	1	0	Read: Always read 1 Write: Reset all Bits
				1	Read: Always read 1 Write: No Effect
11, 10, 9, 8	RCTI3 … RCTI0	Interrupt Interval	0, 0, 0, 0	0, 0, 0, 0	2^8 / CLKRC (~ 128 µs/2.5 ms*)
				0, 0, 0, 1	2^9 / CLKRC (~ 256 µs/5.1 ms*)
				0, 0, 1, 0	2^{10} / CLKRC (~ 512 µs/10.2 ms*)
				0, 0, 1, 1	2^{11} / CLKRC (~ 1ms/20.5 ms*)
				0, 1, 0, 0	2^{12} / CLKRC (~ 2 ms/41 ms*)
				0, 1, 0, 1	2^{13} / CLKRC (~ 4 ms/82 ms*)
				0, 1, 1, 0	2^{14} / CLKRC (~ 8 ms/164 ms*)
				0, 1, 1, 1	2^{15} / CLKRC (~ 16 ms/328 ms*)
				1, 0, 0, 0	2^{16} / CLKRC (~ 33 ms/655 ms*)
				1, 0, 0, 1	2^{17} / CLKRC (~ 66 ms/1.3 s*)
				1, 0, 1, 0	2^{18} / CLKRC (~ 131 ms/2.6 s*)
				1, 0, 1, 1	2^{19} / CLKRC (~ 262 ms/5.2 s*)
				1, 1, 0, 0	2^{20} / CLKRC (~ 524 ms/10.4 s*)
				1, 1, 0, 1	2^{21} / CLKRC (~ 1.05 s/21 s*)
				1, 1, 1, 0	2^{22} / CLKRC (~ 2.1 s/42 s*)
				1, 1, 1, 1	2^{23} / CLKRC (~ 4.2 s/84 s*)

* These time intervals are calculated for 2 MHz and 100 kHz RC Clock frequency respectively.

3 Source Clock Timers Example

Examples for the Source Clock Timers

3.1 Source Clock Timers Interrupt

Main.c

```
void Init_Port00(void)
{
    PDR00 = 0x00;    /* switch off all port pins                     */
    DDR00 = 0xFF;    /* set parallel port direction register: output */
}

void Init_MC_Timer(void)
{
    MCTCR = 0x5E;    /* Enable interrupt, interval: approx. 1.049 s  */
}

void Init_SC_Timer(void)
{
    SCTCR = 0x56;    /* Enable interrupt, interval: approx. 2 s      */
}

void main(void)
{
    InitIrqLevels();
    __set_il(7);     /* allow all levels          */
    __EI();      /* globally enable interrupts */

    Init_Port00();
    Init_MC_Timer();
    Init_SC_Timer();

    while(1);     /* wait for interrupts       */
}

//-----------------------------------------------------------------------
__interrupt void ISR_MC_Timer(void)
{
    MCTCR_MCTIF = 0; /* clear interrupt bit       */

    PDR00 ^= 0x01;   /* invert Port 00_0          */
}

__interrupt void ISR_SC_Timer(void)
{
    SCTCR_SCTIF = 0; /* clear interrupt bit       */

    PDR00 ^= 0x02;   /* invert Port 00_1          */
}
```

The above sample code demonstrates to configure the Main Clock Timer at an interrupt interval of 1.049 seconds (approximately) and to configure Sub Clock Timer interrupt at an interval of 2 seconds (approximately). In the Main Clock Timer interrupt service routine Port 0 bit 0 is toggled and in the Sub Clock Timer interrupt service routine Port 0 bit 1 is toggled.

vectors.c

```
void InitIrqLevels(void)
{
  ICR = (14 << 8) | 2;            /*Priority Level 2 for Main Clock Timer of
                                    MB9634x Series */
  ICR = (15 << 8) | 2;            /*Priority Level 2 for Sub Clock Timer of
                                    MB9634x Series */
}
        . . .

/* ISR prototype */
__interrupt void ISR_MC_Timer(void);
__interrupt void ISR_SC_Timer(void);

        . . .

#pragma intvect ISR_MC_Timer    14    /* Main Clock Timer of MB9634x Series */
#pragma intvect ISR_SC_Timer    15    /* Sub Clock Timer of MB9634x Series  */

        . . .
```

4 Additional Information

Information about Cypress Microcontrollers can be found on the following Internet page:

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

The software example related to this application note is:

96340_src_clk_tmr

It can be found on the following Internet page:

http://www.cypress.com/16lx

Document History

Document Title: AN205554 - F²MC-16FX Family series with Source Clock Timers

Document Number: 002-05554

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	-	NOFL	07/10/2006	Initial release
			03/02/2007	Reviewed the document and updated with review findings
			08/27/2007	Corrected typos, updated basic functionality, added list of tables and Figures
*A	5074691	NOFL	03/02/2016	Migrated Spansion Application Note MCU-AN-300216-E-V12 to Cypress format
*B	5872363	AESATP12	09/06/2017	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, 2006-2017. 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.