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.

AN205908

Hardware Watchdog Timer Setting/Usage

Associated Part Family: FM3 Family 32-bit Microcontroller
MB9A100/MB9B100/MB9B300/MB9B400/MB9B500 Series

This application note describes how to use the hardware watchdog timer.

Contents

1 Preface 1
2 Hardware Watchdog Timer 2
2.1 Interrupt (NVIC) 2
2.2 Interrupt (Interrupt Request Batch Read Register 00) 2
2.3 Register 3
2.4 Hardware watchdog reset 4
2.5 APB0 Bus Clock 4
3 Setting Example (Sample Program) 5
3.1 Sample Program 5
3.2 Setting Procedure Example 6
4 Precautions 7
Document History 8
Worldwide Sales and Design Support 9

1 Preface

The watchdog timer is a function to detect runaway of user program.

The hardware watchdog timer in this series has following features:

- The hardware watchdog timer is activated by turning on the device, and after releasing all the resets except software resets without an intervention of software.
- The hardware watchdog timer can be stopped by accessing a register by software.
- Low-speed CR clock (CLKLC) is used for a count clock.
- It counts cycles while CLKLC is operating, and it stops counting while CLKLC is stopped in the standby mode (stop mode). The count value is retained so that it continues counting after returning from the standby mode.

This application note describes how to use the hardware watchdog timer.

It also shows setting example of changing the count value of hardware watchdog timer to an arbitrary value in sample program (in the case of MB9BF506).

2 Hardware Watchdog Timer

Note: For features, operations, block diagrams of hardware watchdog timer, refer to CHAPTER 11: Watchdog timer of MB9Axxx/MB9Bxxx Series PERIPHERAL MANUAL.

2.1 Interrupt (NVIC)

Table 2-1 shows vectors of exception and interrupt factor input to NVIC.

Note: For detail on NVIC, refer to CHAPTER 8: Nested Vectored Interrupt Controller of Cortex-M3 Technical Reference Manual.

Table 2-1 Base Timer Interrupt Vector

Exception and Interrupt Factor	Vector No.	IRQ No.	Vector Offset
Non-maskable interrupt (NMI) Hardware watchdog timer	2	-	0x08

Since vector offset is ”0x08”, the interrupt address will be stored in ”0x00000008”.

2.2 Interrupt (Interrupt Request Batch Read Register 00)

By using the interrupt request batch read register, you can batch-read the interrupt request of interrupt vector No.2 including base timer interrupt. This enables to confirm which interrupt request is generated, the request of hardware watchdog timer or that of NMI.

Note: For details of interrupt request batch read register, refer to CHAPTER 6: Interrupt of MB9Axxx/MB9Bxxx Series PERIPHERAL MANUAL.

2.3 Register

Setting registers differs according to hardware and software. Table 2-22 shows the registers that can be set by hardware watchdog timer.

Note: For details on register setting, refer to CHAPTER 11: Watchdog Timer of MB9Axxx/MB9Bxxx Series PERIPHERAL MANUAL.

Table 2-2 Registers Used by Hardware Watchdog Timer

Abbreviation	Register Name	Address
WDG_LDR	Hardware watchdog timer load register	Base + 0x000
WDG_VLR	Hardware watchdog timer value register	Base + 0x004
WDG_CTL	Hardware watchdog timer control register	Base + 0x008
WDG_ICL	Hardware watchdog timer clear register	Base + 0x00C
WDG_RIS	Hardware watchdog timer interrupt status register	Base + 0x010
WDG_LCK	Hardware watchdog timer lock register	Base + 0xC00

Note: Hardware watchdog timer base address: 0x40011000

2.3.1 WDG_LCK (Hardware Watchdog Timer Lock Register)

When accessing to a register of hardware watchdog timer, you need to write a certain value to WDG_LCK to unlock the register. If you access to the register of hardware watchdog timer after releasing it, it is locked again automatically. Thus, you need to release the lock every time you access to the registers.
The read value of WDG_LCK while it is locked is “0x01”, and the read value of WDG_LCK while it is unlocked is “0x00”.
When registers other than WDG_CTL writes “0x1ACCE551” to WDG_LCK and WDG_CTL writes “0x1ACCE551” to WDG_LCK, access to the register of hardware watchdog timer after writing “0xE5331AAE” (the reversal value) to WDG_LCK.
Table 2-3 shows the procedure of releasing a lock for each register.

Table 2-3 Unlocking Procedure for Hardware Watchdog Timer Registers

Register	Unlocking Procedure
WDG_LDR	Write “0x1ACCE551” to WDG_LCK
WDG_CTL	After writing “0x1ACCE551” to WDG_LCK, write “0xE5331AAE” to WDG_LCK
WDG_ICL	Write “0x1ACCE551” to WDG_LCK

2.4 Hardware Watchdog Reset

When a reset by hardware watchdog timer occurs, refer to the reset factor register to confirm it. Table 2-4 shows the conditions and flags.

Table 2-4 Hardware Watchdog Reset

Generation factor	Generated by two underflows without clearing the counter of hardware watchdog timer.
Release factor	Released automatically after issuing a reset.
Initialization target	Initializes all register settings and hardware except the debug circuit. Note: The reset factor register is not initialized.
Flag	Bit 5 (HWDT) of reset factor register (RST_STR) = 1

2.5 APB0 Bus Clock

The bus clock for hardware watchdog timer of MB9BF506 is APB0 (refer to Block Diagram of FM3 MB9B500 Series Data Sheet) and its maximum internal operating clock frequency is 40 MHz. When CPU/AHB bus clock is to be set over 40 MHz, you need to divide frequency by setting APB0 prescaler register (APBC0_PSR).

Note: For details on APB2 bus clock, refer to CHAPTER 2-1: Clock of MB9Axxx/MB9Bxxx Series PERIPHERAL MANUAL.

3 Setting Example (Sample Program)

3.1 Sample Program

This section describes the sample program that configures the hardware watchdog timer.

The following shows the setting conditions of this sample program.

Clock conditions:

- Set PLL clock of 80 MHz as a master clock (multiply after inputting external main clock of 4 MHz to X0 and X1)
- APB2 prescaler: 1/2 frequency (APB1 bus clock: 40 MHz)

According to the conditions above, process as follows in sample program:

- Set the hardware watchdog timer before main loop (counter value and reset setting) and clear the hardware watchdog counter for each main loop.
- If the counter underflows because of abnormal operation, underflow interrupt of hardware watchdog timer counter is generated to transit to interrupt function. If an underflow is generated again, hardware watchdog reset will occur.

3.1.1 Hardware Watchdog Underflow Cycle in Sample Program

The following shows the hardware watchdog underflow cycle in sample program.

Low-speed internal CR: 100 kHz (Typ) → 1 cycle: 10 µs

Interval cycle of hardware watchdog (initial value): WDG_LDR = 0x0000FFFF

Count value: 65,535

Interval cycle of hardware watchdog (setting value): WDG_LDR = 0x0001FFFF

Count value: 131,071

Underflow frequency after power-on is calculated as follows:

10 µs*65535 = 655350 µs = 0.65535 s (hereinafter approx. 0.65 s)

Underflow frequency set by sample program is calculated as follows:

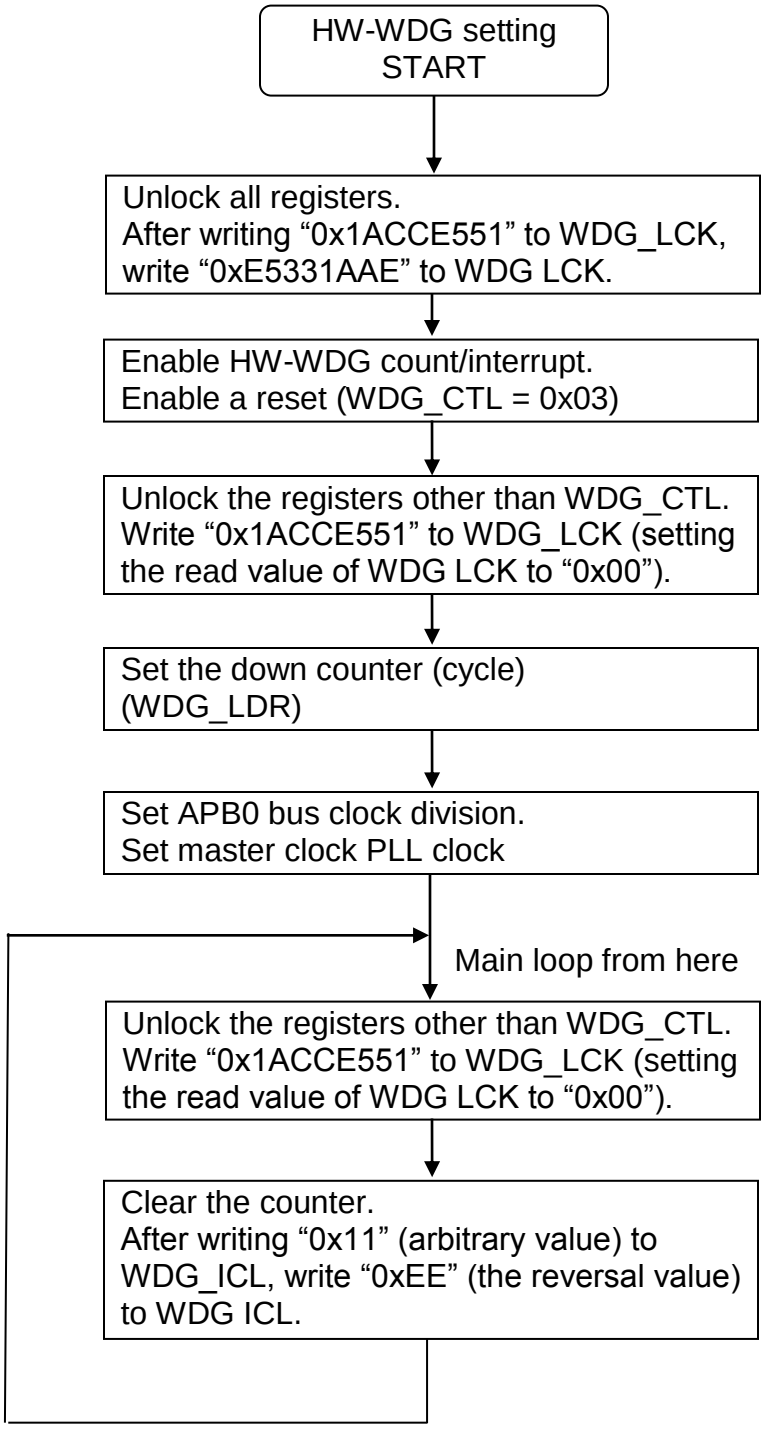
10 µs*131071 = 1310710 µs = 1.310710 s (hereinafter approx. 1.3 s)

After turning the power on, underflow interrupt is generated if the hardware watchdog counter is not cleared within 0.65 s after a change of load register and clear by clear register.
Moreover, if it is still not cleared by clear register within 0.65 s after this, hardware watchdog reset is generated.
Since the sample program has changed the interval cycle, underflow interrupt is generated if the hardware watchdog counter is not cleared within 1.3 s after the change. Also, hardware watchdog reset is generated if the hardware watchdog counter is not cleared within 1.3 s after this.

3.2 Setting Procedure Example

The following procedure shows the setting of hardware watchdog timer in sample program.

Fig. 3.1 Setting Procedure Example in Sample Program



4 Precautions

- After turning the power on, the hardware watchdog timer starts its operation. Since the initial value of WDG_CTL register is “0x03” (enabling resets, interrupts, and counts), disable it (WDG_CTL = 0x00) if you do not use the hardware watchdog timer. The initial value of WDG_LDR is “0x0000FFFF”. Therefore, the hardware watchdog timer interrupt is generated if it is stopped within 0.65 s as described in 3.1.1, and if the count value is not changed or not cleared.
- The sample project attached to this application note is created by IAR Embedded Workbench for ARM.

Document History

Document Title: AN205908 - Hardware Watchdog Timer Setting/Usage

Document Number: 002-05908

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	-	NNAK	02/02/2011	Migrated to Cypress and assigned document number 002-05908. No change to document contents or format.
*A	5868729	NNAK	08/31/2017	Updated 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
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, 2011-2017. 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.