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.

AN205320

FR Family, MB91460 Microcontroller Software Watchdog

This application note describes the functionality of the Watchdog Timer and Watchdog Reset with examples.

Contents

1 Introduction 1
1.1 Key Features 1
2 The Watchdog Timer 1
2.1 Block Diagram 2
2.2 Registers 3
2.3 Operation of Watchdog timer 5
3 Watchdog Timer Examples 6
3.1 Timer Enable and Reset 6
3.2 Determining Watchdog Reset 7
4 Additional Information 7
Document History 8

1 Introduction

This application note describes the functionality of the Watchdog Timer and Watchdog Reset and gives some examples.

The Watchdog is an up-counting Timer which has to be cleared in a given time interval. Otherwise a reset is performed. It can be used for restarting an application, if the program gets stuck.

1.1 Key Features

- Generates the watchdog reset (INIT) with the overflow from one-bit counter
- Counter counts bit output from the time base timer (either Bit 20,22,24 or 26)
- Timer clear by Successively writing “A5”“5A” to watchdog reset generation delay register `WPR` by the software
- This timer starts to operate once it writes data to the watchdog control register `RSRR` for the first time after the reset (RST). This timer stops by reset (RST, INIT, SWWD reset, HWWD reset).
- Timer stopped in Stop Mode, Sleep Mode
- Timer stopped in break when using the emulator debugger and monitor debugger, only if DSU4 is mounted
- Timer stopped in break when using the embedded debug support unit (only if EDSU and EMMODE is enabled)
- Period between the time when executing the INTE command and when executing RETI, only if DSU4 is mounted
- Timer Interval using base clock equal to 80MHz : ~13 ms to ~838ms
- Timer Interval using base clock equal to 2MHz: ~0.5s to ~33 s
- Timer Interval using base clock equal to 32.768kHz: ~32s to ~2048 s

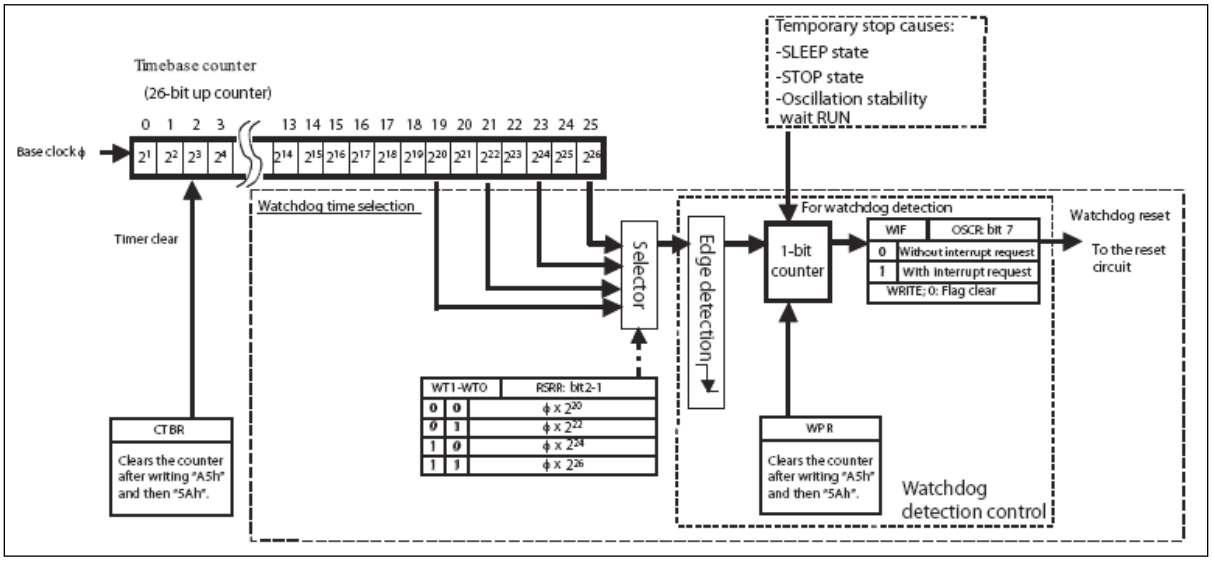
2 The Watchdog Timer

The Basic Functionality of the Watchdog Timer

2.1 Block Diagram

Figure 1 shows the internal block diagram of the Watchdog Timer.

Figure 1. Watchdog Timer block diagram



2.2 Registers

2.2.1 Watchdog Timer Control Register (RSRR)

Table 1. RSRR

Bit No.	Name	Explanation	Value	Operation
7	INIT	Initialization reset occurred flag	0	No INIT has been triggered by the INITX pin input
			1	INIT has been triggered by the INITX pin input
6	HSTB	Hardware Standby reset occurred flag	0	No INIT has been triggered by the HSTX pin input
			1	INIT has been triggered by the HSTX pin input
5	WDOG	Watchdog reset occurred flag	0	No INIT has been triggered by the watchdog timer
			1	INIT has been triggered by the watchdog timer
4	ERST	External reset occurred flag	0	No RST has been triggered by the RSTX pin input
			1	RST has been triggered by the RSTX pin input
3	SRST	Software reset occurred flag	0	No RST has been triggered by a software reset
			1	RST has been triggered by a software reset
2	LINIT	Low voltage reset occurred flag	0	No INIT has been triggered by the low voltage detection
			1	INIT has been triggered by the low voltage detection
1,0	WT1,WT0	Watchdog interval time selection		The minimum writing interval required for WPR so that the watchdog timer may not be reset
			00	Ø x 2^{20}
			01	Ø x 2^{22}
			10	Ø x 2^{24}
			11	Ø x 2^{26}

Ø = base Clock

Note: Reading the reset request cause returns the reset cause flags and then clears the flag values to “0”. If multiple resets occur prior to reading the register, the resulting flag values contain the bitwise OR of the flags for each reset. That is, more than one flag may be set to “1”.

2.2.2 Watchdog Reset Generation Postponement Register (WPR)

This register is used to postpone the generation of watchdog reset.

If “A5H” and “5AH” are successively written in the watchdog reset generation postponement register and immediately after writing “5AH” the 1-bit counter used to detect the watchdog is set to “0” to postpone the generation of a watchdog reset.

Although there are no restrictions on the write timings for “A5H” and “5AH”, if “A5H” and a value other than “5AH” are written, “A5H” must be written again. If not, writing “5AH” does not set the 1-bit counter to “0”.

Both “A5H” and “5AH” must be written within the specified interval as shown below to prevent the watchdog reset from being generated. The intervals are shown in the following table according to the watchdog interval time selection bit (`RSRR.WT [1:0]`).

WT1	WT0	Minimum interval required for writing data in WPR
0	0	With in Base Clock * 2^{20}
0	1	With in Base Clock * 2^{22}
1	0	With in Base Clock * 2^{24}
1	1	With in Base Clock * 2^{26}

2.2.3 Time base Counter Clear Register (CTBR)

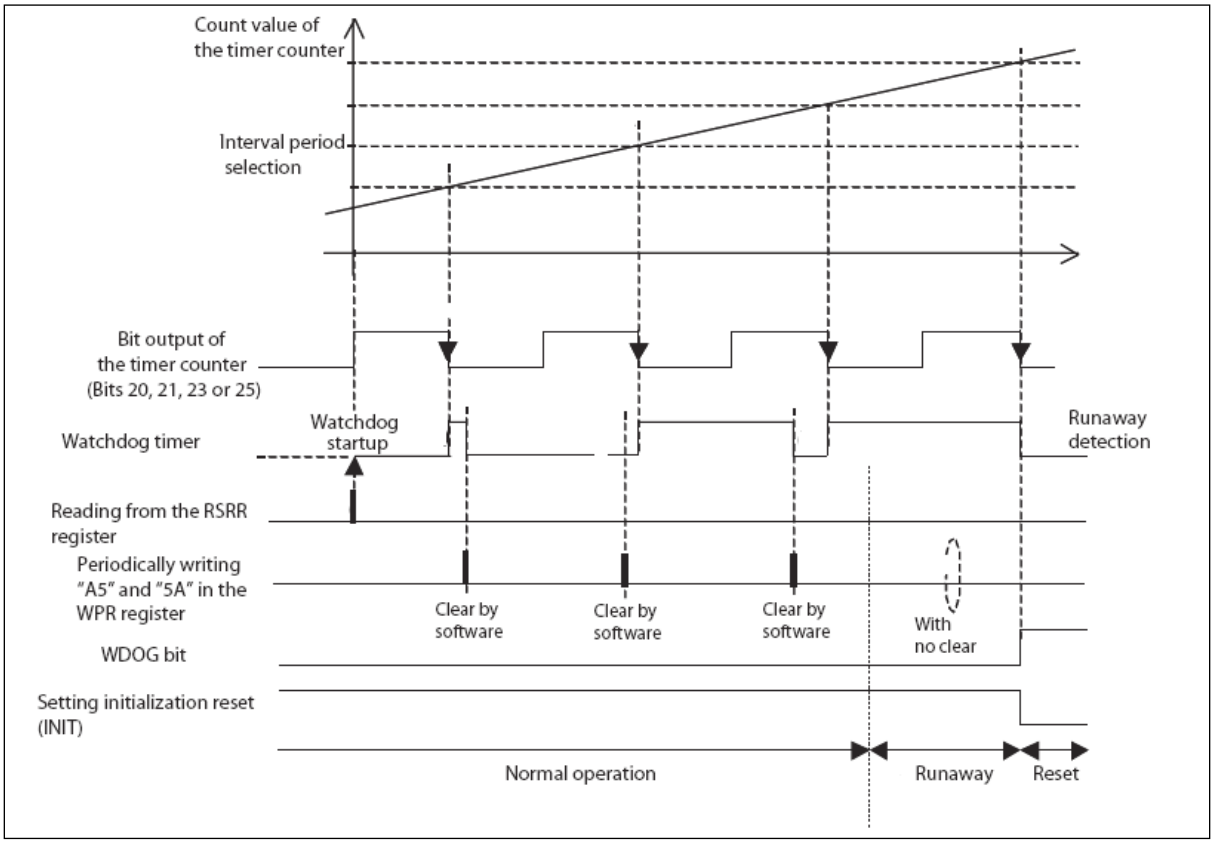
This register is used to initialize the time base counter.

For more information refer to application note AN205314 - FR, MB91460, Time Base Counter and Time Base Timer.

2.3 Operation of Watchdog timer

Following diagram describes operation of watchdog timer. Watchdog timer start immediately after writing `RSRR: WT` register and counts signal output from time base counter. The watchdog timer is cleared by writing to the WPR register with “A5” and “5A” within the interval time. If it is not done than runaway of MCU will be detected. WODOG flag will be changed to “1” and watchdog reset (INIT) is generated.

Figure 2. Watchdog Timer operation



3 Watchdog Timer Examples

Examples for the Watchdog

3.1 Timer Enable and Reset

```
        /*SAMPLE CODE
        /*-------------------------------------------------------------------------
        ---*/

        void InitWatchdog(void)
        {
            RSRR = 0x03;        // 2^26..2^27 (Base Clock = 64MHz:1.048576..
        2.097152 Sec)
        }

        void WatchdogReset(void)
        {
            WPR = 0xa5;         // postpone watchdog reset
            WPR = 0x5a;
            CTBR = 0xA5;        // Clear time base timer
            CTBR = 0x5A;
        }

        void main(void)
{
    . . .

    InitWatchdog();   // Start Watchdog Timer

    . . .

    WatchdogReset();  // Reset Watchdog Timer

    . . .
}
```

3.2 Determining Watchdog Reset

```
/* SAMPLE CODE
/*-------------------------------------------------------------------------
---*/
void DetermineWtReset(void)
{
       unsigned char reset_cause;

       reset_cause = RSRR; // Read reset cause, also clear it

   if (reset_cause & 0x20) // Is reset caused by watchdog overflow?
   {
       // Reset caused by watchdog overflow,
       // Take appropriate action
       }
}
```

4 Additional Information

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

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

Document History

Document Title: AN205320 - FR Family, MB91460 Microcontroller Software Watchdog

Document Number: 002-05320

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	—	NOFL	06/05/2008	Initial release
*A	5115218	NOFL	04/15/2016	Converted Spansion Application Note “MCU-AN-300069-E-V10” to Cypress template.
*B	5833510	AESATP12	07/27/2017	Updated logo and copyright.
*C	6059522	NOFL	02/05/2018	Updated hyperlinks across the document. Updated to new template. Completing Sunset Review.

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 MCU

Cypress Developer Community

Community | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2008-2018. 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.