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.

AN205314

FR, MB91460, Time Base Counter and Time Base Timer

The Time Base Counter is a 26-bit up-counter that counts the base clock. When recovering from a state in which the selected clock source for the MCU has been, or may have been, halted, the MCU automatically changes to the oscillation stabilization wait state to avoid any unstable output from the oscillator

Contents

1 Introduction....1
1.1 Key Features....1
2 Time Base Counter and Time Base Timer....2
2.1 Block Diagrams....2
2.2 Registers....3
3 Operations....5
4 Software Example....8
4.1 Basic Functionality of the Time Base Timer....8
4.2 Recovering from STOP state....9
5 Additional Information....9
Document History....10

1 Introduction

The Time Base Counter is a 26-bit up-counter that counts the base clock. When recovering from a state in which the selected clock source for the MCU has been, or may have been, halted, the MCU automatically changes to the oscillation stabilization wait state to avoid any unstable output from the oscillator. During the oscillation stabilization wait time, supply of internal and external clocks is halted and only the Time Base Counter continues to operate until the time set by the oscillation stabilization wait time setting has elapsed.

The Time Base Counter and the Time Base Timer are based on the same hardware module. The Time Base Counter is used in this chapter to generate the oscillation stabilization wait time. Later, when the software is running the Time Base Counter is used as a timer (Time Base Timer).

1.1 Key Features

1.1.1 Time Base Counter

- 26-bit up counter
- Base clock is fed in as a clock source
- Cleared automatically when changing to oscillation stabilization wait state

1.1.2 Time Base Timer

- Generates an interval interrupt
- Interval time is selected from Base Clock * 2^{11} to Base Clock * 2^{56}
- Continuously writing “A5H”, “5AH” in the time base counter clear register (CTBR) clears the Time Base Counter immediately after writing “5AH”. (All bits are “0”) Input.

2 Time Base Counter and Time Base Timer

The basic functionality of time base counter and time base timer is discussed here

2.1 Block Diagrams

Figure 1 shows the internal block diagram of a Time Base Counter and Timer.

Figure 1. Time Base Counter and Timer

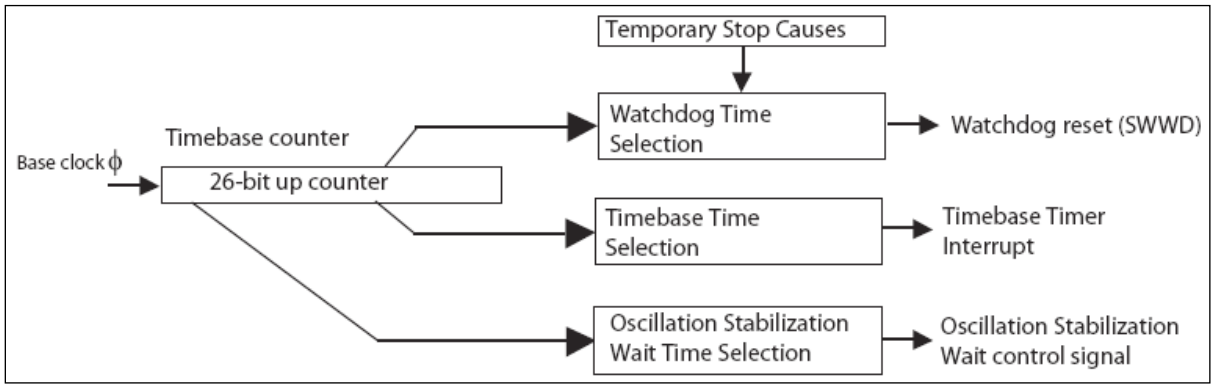


Figure 2. Time Base Counter when used to generate the oscillation stabilization wait

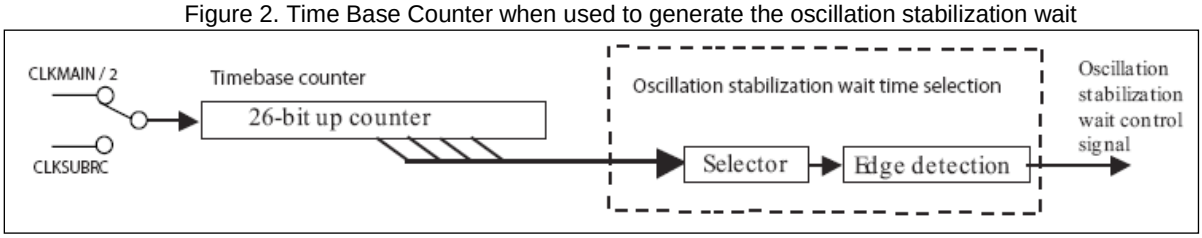
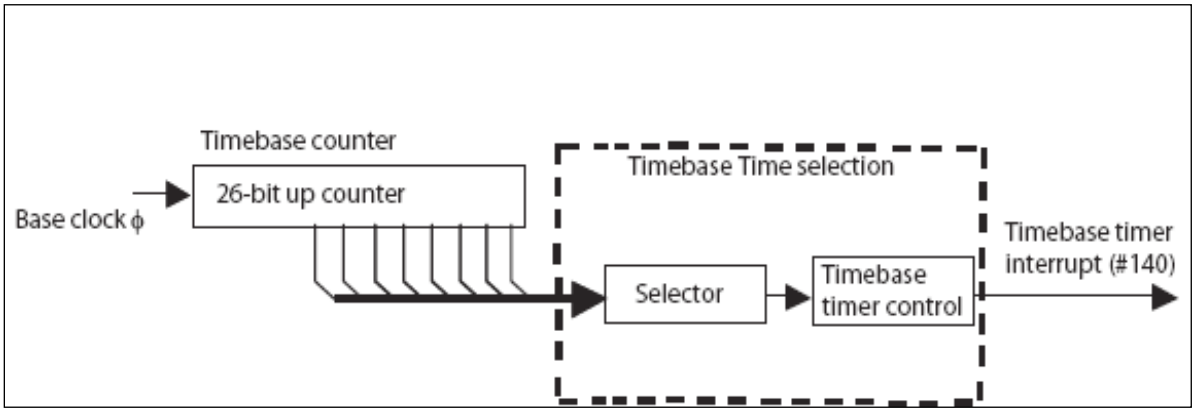


Figure 3. Time Base Counter used to generate the Time Base Timer Interrupt



2.2 Registers

2.2.1 Time Base Counter

2.2.1.1 Standby Control Register (STCR)

This Byte contains several status and control bits.

Table 1. STCR

Bit No.	Name	Explanation	Value	Operation		
7	STOP	STOP state	0			
			1	Change to Stop state		
6	SLEEP	SLEEP state	0			
			1	Change to Sleep state		
5	HIZ	High impedance mode	0	Pins maintain the same states they have on entering STOP state		
			1	pin outputs go to high impedance (Hi-z) during STOP state		
4	SRST	Software reset	0	Software reset		
			1			
3,2	OS1, OS0	Oscillation stabilization time selection		Oscillations stabilization wait time	When using Main clock (4.0MHz)	When using Sub clock (32.768kHz)
			0, 0	Base clock * 2^1	1.0µs	61µs
			0, 1	Base clock * 2^{11}	1.0ms	62.5ms
			1, 0	Base clock * 2^{16}	32ms	2.0s
			1, 1	Base clock * 2^{22}	2s	128s
1	OSCD1	Sub clock oscillation halt	0	Initialization by Reset or Clear Bit		
			1	Initialization by Reset, Clear Bit, or Compare Register (OCCP)		
0	OSCD2	Main clock oscillation halt	0	Write: no effect		
			1	Write: Clear to “0000”		

2.2.1.2 Clock Source Control Register (CLKR)

Table 2. CLKR

Bit No.	Name	Explanation	Value	Operation
7-4	-	*Undefined*	-	*Write always 1.* *The read value is value written*
3	SCKEN	Sub clock selection enable	1	enables the Sub clock to be selected
2	PLL1EN	Main PLL operation enable	1	starts Main PLL operation
1,0	CLKS1, CLKS0	Clock source selection	00	The Main clock input from X0/X1 divided by 2
			01	The Main clock input from X0/X1 divided by 2
			10	Main PLL
			11	Sub clock

2.2.2 Time Base Timer

2.2.2.1 Time Base Timer Control Register (TBCR)

Table 3. TBCR

Bit No.	Name	Explanation	Value	Operation for Channel *(2n+0)* or *(2n+1)*		
7	TBIF	Time Base timer interrupt flag	0	Read: No Interrupt Write: Flag is cleared		
			1	Read: Interrupt Write: No effect		
6	TBIE	Time Base timer interrupt request enable	0	Interrupt request disable		
			1	Interrupt request enable		
5,4,3	TBC2, TBC1, TBC0	Selecting the Time Base timer interval time		Interval time	32MHz, Base Clock	32.768kHz, Base clock
			000	Ø * 2^{11}	62µs	62.5ms
			001	Ø * 2^{12}	128µs	125ms
			010	Ø * 2^{13}	256µs	250ms
			011	Ø * 2^{22}	131ms	128s
			100	Ø * 2^{23}	262ms	256s
			101	Ø * 2^{24}	524ms	512s
			110	Ø * 2^{25}	1048ms	1024s
			111	Ø * 2^{26}	2097ms	2048s
2	-	*Undefined*	-	*Writing does not affect the operation. The read value is indefinite*		
4	SYNCR	Enabling the synchronous reset operation	0	Asynchronous reset operation		
			1	Synchronous reset operation enable		
4	SYNCS	Synchronous standby operation enable	0	Asynchronous reset operation (Not permitted on this model).		
			1	Enable synchronous standby operation (always set this before changing to a standby mode).		

Ø = Base Clock

2.2.2.2 Time Base Counter Clear Register (CTBR)

This an 8-Bit register. Continuously writing in sequence, “A5H”, “5AH” in the Time Base Counter Clear register, clears the Time Base counter immediately after writing “5AH”. (All bits are “0”) There are no time restrictions between “A5H” and “5AH”, but if writing of “A5H” is not followed by “5AH”, one must write “A5H” again. If not, the Time Base counter cannot be cleared.

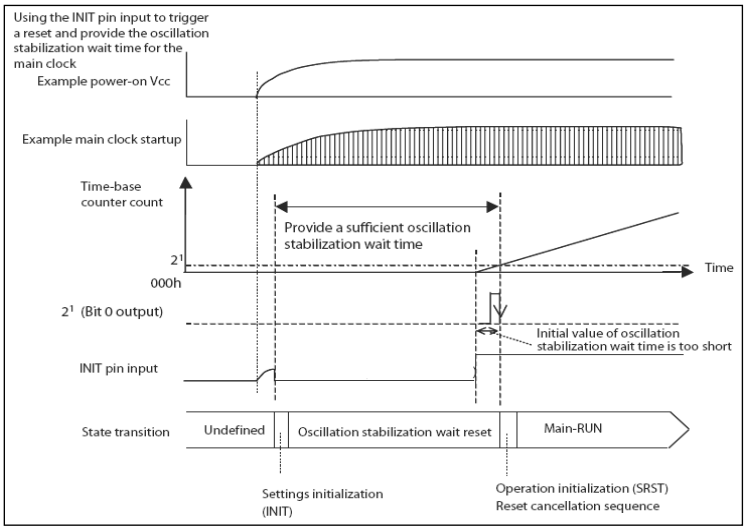
3 Operations

This section describes events that require an oscillation stabilization wait time and its operation in each case.

3.1.1 INITX Pin Input

An oscillation stabilization wait is required after power on.

Figure 4. Using the Width of the Pin Input to Provide the Oscillation Stabilization Wait Time



Since oscillation stabilization wait time provided by time base timer/counter after reset is not sufficient it is recommended to hold the reset signal to provide a sufficient time for the Main clock oscillation to stabilize.

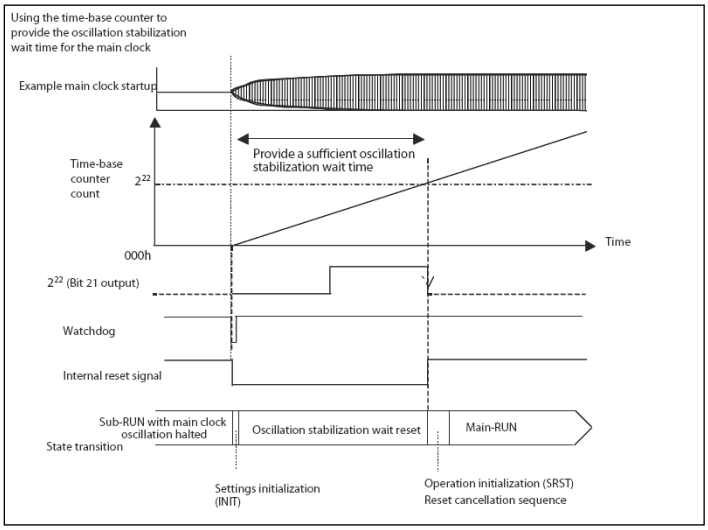
When Main clock running and INITX Pin input, the device goes to the operation initialization reset (RST)* state automatically after the minimum oscillation stabilization wait time elapses

* Refer Hardware manual for more information

3.1.2 Watchdog Reset

If the interval time is set beforehand to provide an adequate oscillation stabilization wait time and if a watchdog reset occurs while the Main clock oscillation is halted, the oscillation stabilization wait time is generated automatically. (See figure below.)

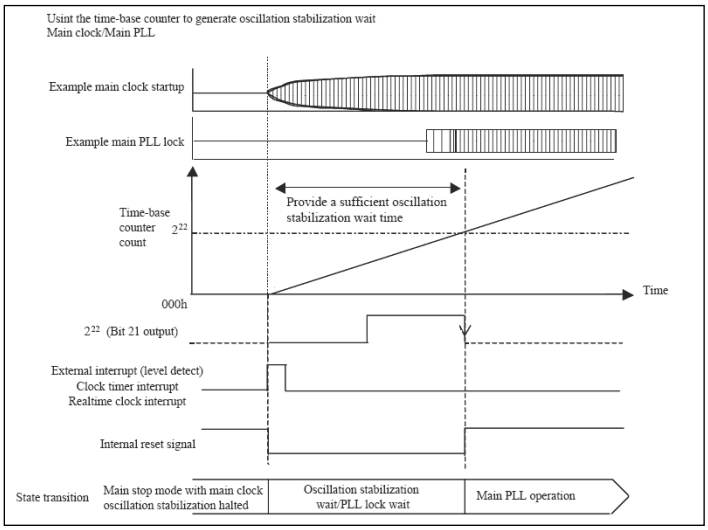
Figure 5. Watchdog Reset when Main Clock Halted (Sub-RUN)



When Main clock operating and Watchdog is reset, although no oscillation stabilization wait is required, the specified wait time is generated automatically.

3.1.3 Recovering from STOP state via an Interrupt

The Main oscillation circuit generates the selected oscillation stabilization time automatically. (See figure below.)



When changing to STOP state without halting the clock oscillation circuit (Main PLL/Main/ Sub), although no oscillation stabilization wait is required in this case, a wait is generated automatically. Accordingly, it is recommended that the interval time is set to its minimum value before changing to STOP state.

When recovering from STOP state, the device goes to the oscillation stabilization wait state immediately after STOP state is released. The next state after the oscillation stabilization wait completes depends on what triggered recovery from STOP state. If recovery was triggered by an enabled external interrupt, sub oscillation stabilization timer interrupt, or main oscillation stabilization wait timer interrupt, the device goes to the normal operating state (RUN).

3.1.4 The lock wait time for the Main PLL

Using the Time base timer interrupt is recommended after Main PLL operation enabled. Using the time base timer interrupt is recommended after Main PLL multiplier modified

3.1.5 Changing from Sub clock mode to Main Clock Mode

When Main clock continues to run during Sub clock mode and If not using Main PLL after changing clock than no oscillation stabilization wait time is required. If, however, using Main PLL after changing clock than Main PLL lock wait is required

When Main clock halts during Sub clock mode, A Main clock oscillation stabilization wait is required before changing clock. If using the Main PLL than further wait is required for the Main PLL to lock

3.1.6 Recovering from an Abnormal State with the Main PLL Selected

When the Main PLL is set as the clock source and a problem of some sort occurs in Main PLL control (such as the multiplier setting being changed or the Main PLL enable bit modified during Main PLL operation), the device goes to the oscillation stabilization wait state automatically to provide the Main PLL lock time. The device then goes to normal operating mode after the oscillation stabilization wait elapses.

4 Software Example

Example for time base timer

4.1 Basic Functionality of the Time Base Timer

The following example shows how to set up Time base Timer.

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

void InitFRTimer0(void)
{
  TBCR_TBIE = 1;     // Interrupt Enable
  TBCR_TBC  = 3;     // Base clock/2^22
}

. . .

__interrupt void FRTimer0 (void)
{
   TBCR_TBIF = 0;    // clear ISR request flag

   . . .
}
```

At 64 MHz Base Clock, the above example generates interrupts at an interval of 0.065s.

Please note, that the corresponding interrupt vector and level has to be defined in the *vectors.c* module of our standard template project.

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

void InitIrqLevels(void)
{
. . .

ICR62 = 30;                          // Timebase Overflow
. . .
}

__interrupt void TbTimerIR (void);// prototype

    . . .


    . . .
```

4.2 Recovering from STOP state

Following example shows how to recover from stop state using Time Base timer.

```
/*                          SAMPLE CODE                                    */
/*-------------------------------------------------------------------------*/
void Stop_Mode_wakeup(void)
{
    /* Turn Pll on */
    CLKR |= 0x4;

    /* Pll Stabilization Wait Time */
    TBCR = 0x10;            // Base clock / 2^13
    CTBR = 0xA5;            // Clear Base Timer register
    CTBR = 0x5A;
    while (TBCR_TBIF == 0)  // Wait for interrupt flag to set
      HWWD = 0x10;

    /* Pll Gear up */
    PLLCTRL_GRUP = 0;
    CLKR |= 0x02;           // Clock source selected to MAINPLL
    while(!PLLCTRL_GRUP)
HWWD = 0x10;
    PLLCTRL_GRUP = 0;

   return;

}
```

5 Additional Information

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

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

The software examples related to this application note is:

91460_tbt_tbc

91460_lpm_stop-v12

It can be found on the following Internet page:

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

Document History

Document Title: AN205314 - FR, MB91460, Time Base Counter and Time Base Timer

Document Number: 002-05314

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	-	NOFL	04/28/2008	V1.0, First draft, HPi
*A	5090038	NOFL	04/06/2016	Converted Spansion Application Note “MCU-AN-300063-E-V10” to Cypress format
*B	5870355	AESATMP9	09/01/2017	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, 2008-2017. 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.