Application Report
SLAA739–June 2017

Proper Sleep and Interrupt Use on the SimpleLink™ MSP432™ ARM® Cortex®-M4 Microcontrollers

Evan Wakefield *Connected MCU Applications*

ABSTRACT

This application note helps developers understand the proper way to go to sleep using the Wait For Interrupt (WFI) instruction in an ARMv7-M or ARMv6-M application, particularly when using the SimpleLink™ MSP432™ microcontrollers. If System Control Register (SCR) bits are not set in a particular order and a Data Synchronization Barrier (DSB) is not executed to flush the data memory transfer pipeline, the device can potentially behave in an unexpected manner if the SCR access side effects are potentially needed immediately after being set.

Contents

Trademarks

SimpleLink, MSP432 are trademarks of Texas Instruments.
ARM, Cortex are registered trademarks of ARM Ltd.
All other trademarks are the property of their respective owners.

1 Introduction

ARMv6-M and ARMv7-M architectures provide memory barrier instructions that are used to ensure the synchronization of events by an ARM® Cortex®-M microcontroller. One of the specific memory barrier instructions used is the Data Synchronization Barrier (DSB) that is used to ensure that all explicit data memory transfers before the DSB instruction are completed before any instruction after the DSB is executed [1].

When not using the DSB instruction, a developer that has manipulated SLEEPDEEP or SLEEPONEXIT bits in the System Control Register (SCR) and then tries to immediately go to sleep by executing the Wait For Interrupt (WFI) and an interrupt occurs, unexpected behavior can occur.

2 Examples

This section describes scenarios that can potentially showcase the unexpected behavior and the appropriate way to correct them. The following examples show correct and incorrect coding to access the SCR register, set bits, and use the DSB before going to sleep.

Scenario A (Not Recommended)

```
SCB->SCR |= SCB_SCR_SLEEPDEEP_Msk;      // Set SLEEPDEEP
SCB->SCR |= SCB_SCR_SLEEPONEXIT_Msk;    // Set SLEEPONEXIT
__WFI();                                // Go to sleep
```

Scenario B (Not Recommended)

```
SCB->SCR |= SCB_SCR_SLEEPONEXIT_Msk;    // Set SLEEPONEXIT
SCB->SCR |= SCB_SCR_SLEEPDEEP_Msk;      // Set SLEEPDEEP
__DSB();                                // Flushes pipeline before sleep
__WFI();                                // Go to sleep
```

Scenario C (Recommended)

```
SCB->SCR |= SCB_SCR_SLEEPDEEP_Msk;      // Set SLEEPDEEP
SCB->SCR |= SCB_SCR_SLEEPONEXIT_Msk;    // Set SLEEPONEXIT
__DSB();                                // Ensures SLEEPONEXIT is set
                                        // immediately before sleep
__WFI();                                // Go to sleep
```

In Scenario A, two SCR accesses occur, in which the SLEEPDEEP and the SLEEPONEXIT masks are set before going to sleep. However, in this scenario, if the WFI instruction executes before the data memory transfers that contain the access to the SCR registers are completed, the device may go to sleep without having set the appropriate masks. This can cause unexpected behavior. The solution is to use the Data Synchronization Barrier (DSB) instruction that ensures the instructions before the DSB instruction are completed.

In Scenario B, the DSB is executed to ensure all the explicit data memory transfers have been completed before executing the WFI. However, this is still not correct and is susceptible to the unexpected behavior described in Scenario A. In Scenario B, The SLEEPDEEP mask is set after setting the SLEEPONEXIT mask, but if an interrupt occurs between the two SCR accesses, the processor enters SLEEP mode upon leaving the interrupt, but it might not be the SLEEPDEEP mode the developer wished.

In Scenario C, both of these issues are resolved. By ensuring the SLEEPDEEP mask is set before the SLEEPONEXIT mask, and by using DSB to ensure all the explicit data memory transfers have been completed before executing the WFI, the developer can be sure that the application cannot enter a corner case of unpredictable behavior.

The following examples show correct and incorrect ways to access the System Control Register and manage the sleep power management settings to ensure that unpredictable behavior does not occur during an interrupt service routine (ISR). In Scenario D, if the SLEEPONEXIT mask is disabled at the end of an ISR, there is a chance that the device might not receive the command before exiting the ISR, and the device might actually go to sleep after exiting. To ensure that this does not happen, ARM documentation recommends that you execute a DSB after accessing the SCR register to ensure that the mask has been properly set, which ensures the behavior of the code. [1]

Scenario D (Not Recommended)

```
void eUSCI_ISR_Handler(void)
{
   ...
   SCB->SCR &= ~SCB_SCR_SLEEPONEXIT_Msk;   // Disable SLEEPONEXIT
}
```

Scenario E (Recommended)

```
void eUSCI_ISR_Handler(void)
{
   ...
   SCB->SCR &= ~SCB_SCR_SLEEPONEXIT_Msk;   // Disable SLEEPONEXIT
   __DSB();                                // Ensures SLEEPONEXIT is set
                                           // immediately before exiting ISR
}
```

Another good practice to ensure the predictability of the device is to ensure that the SLEEPONEXIT mask is set at the end of the initialization stage. This setting ensures the device does not enter sleep if an interrupt occurs during the middle of the initialization stage. [2]

3 Conclusion

In conclusion, use the Data Synchronization Barrier to ensure that all data in the pipeline has been flushed and executed before executing a WFI after accessing the SCR. For more information, see the documentation in Section 4.

4 References

1. *ARM® Cortex®-M Programming Guide to Memory Barrier Instructions*
2. *The Definitive Guide to ARM® Cortex®-M3 and Cortex®-M4 Processors, 3rd Edition*, Joseph Yiu

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ('TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products http://www.ti.com/sc/docs/stdterms.htm), evaluation modules, and samples (http://www.ti.com/sc/docs/sampterms.htm).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265