AN205549

F²MC-16FX Family, Delayed Interrupt

This application note describes the functionality of the Delayed Interrupt and gives an example.

1 Introduction

This application note describes the functionality of the Delayed Interrupt and gives an example.

1.1 Key Features

- Interrupt Generation for OS Task Switching

2 Delayed Interrupt

The Basic Functionality of the Delayed Interrupt

2.1 Functionality

The Delayed Interrupt is mainly used by operating systems. It can be used for dispatching after all tasks were finished. Therefore the Delayed Interrupt should have the lowest priority, so that all other Interrupts are preferred (such as scheduler-timer).

2.2 Register

2.2.1 Delayed Interrupt Request Register (DIRR)

The DIRR consists of the following bit:

Table 1. DIRR

Bit No.	Bit Name	Initial Value	Description
7 … 1	–	X	*Undefined Bits; write "0" to them*
0	R0	0	"1" sets Interrupt Request; "0" clear Interrupt Reset

If "1" is written to the R0 Bit and no other interrupt with higher or same priority is pending, the Interrupt service routine of the Delayed Interrupt is called.

The request is cleared by writing "0" to the R0 Bit.

3 Delayed Interrupt Example

Example for the Delayed Interrupt Usage

3.1 Generate Delayed Interrupt Request

The following sample code shows how to generate a Delayed Interrupt Request. Please note that vectors.c is contained in our standard template project.

vectors.c

```
void InitIrqLevels(void)
{
       . . .

       ICR = (12 << 8) | 6;       /* priority level 6 for Delayed interrupt of
                                     16FX Family */

       . . .
}
/* ISR prototype */
__interrupt void ISR_Delayed_Interrupt (void);

       . . .

#pragma intvect ISR_Delayed_Interrupt 12       /* Delayed Interrupt of 16FX
                                                  Family  */
       . . .
```

Main.c

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

    /* do something */

    . . .

    /* set Delayed Interrupt here. If no other higher/same priority interrupt
           is pending */
    /* Delayed Interrupt is executed after next instruction.        */

    DIRR = 0x01;

    /* do something */

    . . .
}

__interrupt void ISR_Delayed_Interrupt (void)
{
    DIRR = 0x00;     /* clear Delayed Interrupt request */

    /* do something else */

    . . .
}
```

4 Additional Information

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

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

5 Document History

Document Title: AN205549 - F²MC-16FX Family, Delayed Interrupt

Document Number:002-05549

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	-	NOFL	06/23/2006	Initial Release
			02/21/2007	Reviewed the document and updated with review findings
			08/10/2007	Clarification in SW example
*A	5074677	NOFL	03/02/2016	Migrated Spansion Application Note MCU-AN-300211-E-V12 to Cypress format
*B	5868077	AESATP12	08/30/2017	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.