XMC1000, XMC4000

Microcontroller Series
for Industrial Applications

Migration from Timer2 to CCU4/CCU8

Migration Guide

V1.0 2013-06

Microcontrollers

Edition 2013-06

Published by
Infineon Technologies AG
81726 Munich, Germany

© 2013 Infineon Technologies AG
All Rights Reserved.

Legal Disclaimer

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics. With respect to any examples or hints given herein, any typical values stated herein and/or any information regarding the application of the device, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation, warranties of non-infringement of intellectual property rights of any third party.

Information

For further information on technology, delivery terms and conditions and prices, please contact the nearest Infineon Technologies Office (**www.infineon.com**).

Warnings

Due to technical requirements, components may contain dangerous substances. For information on the types in question, please contact the nearest Infineon Technologies Office.

Infineon Technologies components may be used in life-support devices or systems only with the express written approval of Infineon Technologies, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

Revision History

Page or Item	Subjects (major changes since previous revision)
V1.0, 2013-06	

Trademarks of Infineon Technologies AG

AURIX™, C166™, CanPAK™, CIPOS™, CIPURSE™, EconoPACK™, CoolMOS™, CoolSET™, CORECONTROL™, CROSSAVE™, DAVE™, EasyPIM™, EconoBRIDGE™, EconoDUAL™, EconoPIM™, EiceDRIVER™, eupec™, FCOS™, HITFET™, HybridPACK™, I²RF™, ISOFACE™, IsoPACK™, MIPAQ™, ModSTACK™, my-d™, NovalithIC™, OptiMOS™, ORIGA™, PRIMARION™, PrimePACK™, PrimeSTACK™, PRO-SIL™, PROFET™, RASIC™, ReverSave™, SatRIC™, SIEGET™, SINDRION™, SIPMOS™, SmartLEWIS™, SOLID FLASH™, TEMPFET™, thinQ!™, TRENCHSTOP™, TriCore™.

Other Trademarks

Advance Design System™ (ADS) of Agilent Technologies, AMBA™, ARM™, MULTI-ICE™, KEIL™, PRIMECELL™, REALVIEW™, THUMB™, µVision™ of ARM Limited, UK. AUTOSAR™ is licensed by AUTOSAR development partnership. Bluetooth™ of Bluetooth SIG Inc. CAT-iq™ of DECT Forum. COLOSSUS™, FirstGPS™ of Trimble Navigation Ltd. EMV™ of EMVCo, LLC (Visa Holdings Inc.). EPCOS™ of Epcos AG. FLEXGO™ of Microsoft Corporation. FlexRay™ is licensed by FlexRay Consortium. HYPERTERMINAL™ of Hilgraeve Incorporated. IEC™ of Commission Electrotechnique Internationale. IrDA™ of Infrared Data Association Corporation. ISO™ of INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. MATLAB™ of MathWorks, Inc. MAXIM™ of Maxim Integrated Products, Inc. MICROTEC™, NUCLEUS™ of Mentor Graphics Corporation. Mifare™ of NXP. MIPI™ of MIPI Alliance, Inc. MIPS™ of MIPS Technologies, Inc., USA. muRata™ of MURATA MANUFACTURING CO., MICROWAVE OFFICE™ (MWO) of Applied Wave Research Inc., OmniVision™ of OmniVision Technologies, Inc. Openwave™ Openwave Systems Inc. RED HAT™ Red Hat, Inc. RFMD™ RF Micro Devices, Inc. SIRIUS™ of Sirius Satellite Radio Inc. SOLARIS™ of Sun Microsystems, Inc. SPANSION™ of Spansion LLC Ltd. Symbian™ of Symbian Software Limited. TAIYO YUDEN™ of Taiyo Yuden Co. TEAKLITE™ of CEVA, Inc. TEKTRONIX™ of Tektronix Inc. TOKO™ of TOKO KABUSHIKI KAISHA TA. UNIX™ of X/Open Company Limited. VERILOG™, PALLADIUM™ of Cadence Design Systems, Inc. VLYNQ™ of Texas Instruments Incorporated. VXWORKS™, WIND RIVER™ of WIND RIVER SYSTEMS, INC. ZETEX™ of Diodes Zetex Limited.

Last Trademarks Update 2011-02-24

Table of Contents

Revision History **3**

Table of Contents **4**

1 Overview **5**
1.1 Intended audience 5
1.2 Scope and purpose 5
1.3 References 5

2 Using software to start and stop the timer **6**

3 Using hardware to start the timer **7**

4 Up/Down count enabled **8**

5 Captured Mode **9**

6 Counter Mode **10**

7 Reloading the register on overflow **11**

8 Prescaler Option **12**

9 Conclusion **13**

1 Overview

The Timer2 peripheral, a 16-bit general purpose timer found in the Infineon XC800 8-bit family of microcontrollers, is often used for signal monitoring/conditioning, and for Pulse Width Modulation (PWM) signal generation. The CCU4 and CCU8 peripherals found in the XMC1000 and XMC4000 family of products from Infineon, provide all the functionality of the Timer2 peripheral, but also offer many new feature and performance enhancements.

1.1 Intended audience

This document is intended for users who are already familiar with the Timer2 module, and want to know how to implement similar functionality in XMC1000 or XMC4000 products.

1.2 Scope and purpose

Although the CCU4 and CCU8 support all of the Timer2 peripheral functionality, it is not possible to create a simple formula or table for a one-to-one conversion between peripherals. For example, it is not possible to say: “If bit x is set in the Timer2, set bit y in the CCU4 to achieve the same functionality”.

Instead we focus on a feature comparison between the Timer2 and the CCU4 and CCU8, with references to specific User Guides that describe the detail of how to implement the required feature.

We focus specifically on the following Timer2 features:

- **Using software to start and stop the timer**
- **Using hardware to start the timer**
- **Up/Down count enabled**
- **Captured Mode**
- **Counter Mode**
- **Reloading the register on overflow**
- **Prescaler Option**

1.3 References

For the complete list of available documents, please refer to the XMC web pages:

- **XMC1000** (See the ‘User Guide’ section for peripheral Device Guides)
- **XMC4000** (See the ‘User Guide’ section for peripheral Device Guides)
- **www.infineon.com/xmc**

2 Using software to start and stop the timer

Timer2

- T2_T2CON.TR2 = 1
  - Set the run bit of the timer to start the timer
- T2_T2CON.TR2 = 0
  - Clear the run bit of the timer to stop the timer

CCU4 and CCU8

For the CCU4 (and CCU8) module, the timer start and stop is controlled from a different register bit:

- CC4yTCSET.TRBS = 1
  - Set the run bit of the timer to start the timer
- CC4yTCCLR.TCC = 1
  - Clear the run bit of the timer to stop the timer

Note: For CCU8, replace '4' in the name with '8' to derive the register name; e.g. CC4yTCSET becomes CC8yTCSET.

3 Using hardware to start the timer

Timer2

Timer2 is started by hardware when T2_T2MOD.T2RHEN is set.

T2_T2MOD.T2REGS defines the event on pin T2EX, falling or rising edge, which can set the run bit TR2 by hardware.

- T2RHEN = 0
  - Timer2 external start is disabled
- T2RHEN = 1
  - Timer2 external start is enabled
- T2REGS = 0
  - The falling edge external start at pin T2EX is selected
- T2REGS = 1
  - The rising edge external start at pin T2EX is selected

CCU4 and CCU8

To select an external start function, map one of the events (output of the input selector) to a specific input signal, by:

- Setting the required value in the CC4yINS.EVxIS field
- Indicating the active edge of the signal on the CC4yINS.EVxEM field

This event is then mapped to the start functionality by setting CC4yCMC.STRTS.

Note: For CCU8, replace '4' in the name with '8' to derive the register name; e.g. CC4yINS becomes CC8yINS.

Document references

For more information please refer to the following chapters of the respective User Guides (See the 'User Guide' section of the **XMC1000** or **XMC4000** Infineon web pages):

- **Device Guide – CCU4**
  - External Events Control
- **Device Guide – CCU8**
  - External Events Control

4 Up/Down count enabled

Timer2

- The up-down count selection is enabled when T2_T2MOD.DCEN = 1.
- The direction of count is determined by the level at input pin T2EX.
- The logic 1 at pin T2EX sets the Timer2 to ‘up’ counting mode.
- The logic 0 at pin T2EX sets the Timer2 to ‘down’ counting mode.

CCU4 and CCU8

To select as ‘up’ or ‘down’ count functionality, map one of the events (output of the input selector) to a specific input signal, by:

- Setting the required value in the CC4yINS.EVxIS field
- Indicating the active edge of the signal on the CC4yINS.EVxEM field.

This event is then mapped to the up/down functionality by setting the CC4yCMC.UDS.

Note: For CCU8, replace ‘4’ in the name with ‘8’ to derive the register name; e.g. CC4yINS becomes CC8yINS.

5 Captured Mode

Timer2

When T2_T2MOD.CP =1 and T2_T2MOD.EXEN2 = 1, Timer2 functions as a 16-bit timer and always counts up to 0xFFFF.

After that point an overflow condition occurs and Timer2 reloads its register with 0x0000.

With a falling/rising edge on pin T2EX, the contents of the timer register (THL2) are captured into the RC2 register.

CCU4 and CCU8

To implement capture mode functionality, map one of the events (output of the input selector) to a specific input signal, by:

- Setting the required value in the CC4yINS.EVxIS field
- Indicating the active edge of the signal on the CC4yINS.EVxEM field

This event is then mapped to the capture functionality by setting CC4yCMC.CAP0S.

The contents of the timer register will be captured into Capture Register 1 (CC4yC1V.CAPTV).

Note: For CCU8, replace '4' in the name with '8' to derive the register name; e.g. CC4yINS becomes CC8yINS.

Document references

For more information please refer to the following chapters of the respective User Guides (See the 'User Guide' section of the **XMC1000** or **XMC4000** Infineon web pages):

- **Device Guide – CCU4**
  - Advanced Signal Measurement using the Capture Mode
- **Device Guide – CCU8**
  - Advanced Signal Measurement using the Capture Mode

6 Counter Mode

Timer2

When C/T2 = 1, Timer2 behaves as a counter that counts 1-to-0 transition at the input pin T2.

CCU4 and CCU8

To implement counting event functionality, map one of the events (output of the input selector) to a specific input signal, by:

- Setting the required value in the CC4yINS.EVxIS field
- Indicating the active edge of the signal on the CC4yINS.EVxEM field

This event must then mapped to the external count functionality by setting CC4yCMC.CNTS.

Note: For CCU8, replace '4' in the name with '8' to derive the register name; e.g. CC4yINS becomes CC8yINS.

7 Reloading the register on overflow

Timer2

The Timer2 auto-reload mode is selected with the bit CP/RL2 in register T2CON=0.

The timer will count to an overflow value (0xFFFF) and then reload the count register contents with a 16-bit start value for a fresh counting sequence.

CCU4 and CCU8

In the edge aligned mode (CC4yTC.TCM = 0), the CCU4 timer will count up from 0x0000 until it matches the value programmed in the period register (CC4yPR).

When the period match is detected the timer is cleared to 0x0000 and then it begins incrementing again.

Note: For CCU8, replace '4' in the name with '8' to derive the register name; e.g. CC4yTC becomes CC8yTC.

8 Prescaler Option

Timer2

T2_T2MOD.PREN allows the user to select whether to use the default divider setting (1/12) or follow the configuration of T2PRE bits.

- When PREN =0, the default 12 divider takes effect.
- When PREN = 1, it will follow the T2PRE bits for the configuration of the prescaler divider.

000	$f_{T2} = f_{T2CCU}$
001	$f_{T2} = f_{T2CCU}/2$
010	$f_{T2} = f_{T2CCU}/4$
011	$f_{T2} = f_{T2CCU}/8$
100	$f_{T2} = f_{T2CCU}/16$
101	$f_{T2} = f_{T2CCU}/32$
110	$f_{T2} = f_{T2CCU}/64$
111	$f_{T2} = f_{T2CCU}/128$

Figure 1 Prescaler divider option for Timer2

CCU4 and CCU8

CC4yPSC.PSIV is used to configure the different possible division values for the timer clock prescaler.

CC4yPSC.PSIV	Resulting clock
0000_B	f_{ccu4}
0001_B	$f_{ccu4}/2$
0010_B	$f_{ccu4}/4$
0011_B	$f_{ccu4}/8$
0100_B	$f_{ccu4}/16$
0101_B	$f_{ccu4}/32$
0110_B	$f_{ccu4}/64$
0111_B	$f_{ccu4}/128$
1000_B	$f_{ccu4}/256$
1001_B	$f_{ccu4}/512$
1010_B	$f_{ccu4}/1024$
1011_B	$f_{ccu4}/2048$
1100_B	$f_{ccu4}/4096$
1101_B	$f_{ccu4}/8192$
1110_B	$f_{ccu4}/16384$
1111_B	$f_{ccu4}/32768$

Figure 2 Prescaler divider option for CCU4

Note: For CCU8, replace '4' in the name with '8' to derive the register name; e.g. CC4yPSC becomes CC8yPSC.

9 Conclusion

The CCU4 and CCU8 peripherals offer the same functionality as the Timer2 peripheral, but with significant additional features. These peripherals are designed with repetitive structures that make automatic code generation easier.

The DAVE™ 3 tool with DAVE™ Apps can be used to handle the most common use cases for the CCU4 and CCU8. Example projects and Users Guides are available for free download from the Infineon website, to ease the learning curve and help speed-up development and implementation.

w w w . i n f i n e o n . c o m

Published by Infineon Technologies AG