XMC4000

XMC4500 Family

CCU4 and Floating Prescaler

Autonomic Signal Measurement Range Adaption

User Guide

Device Guide

<Revision 1.0>, <2012-30-03>

Microcontroller

Edition <2012-30-03>

Published by
Infineon Technologies AG
81726 Munich, Germany

© 2012 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.

Confidential

Revision History	
Page or Item	**Subjects (major changes since previous revision)**
<Revision 1.0>, <2012-30-03>	

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

1 Autonomic Signal Measurement Range Adaption 5
1.1 Floating Prescaler Use Cases 5
1.2 The Floating Prescaler Principle Function 5
1.3 The Floating Prescaler in Signal Measurements by Capture 6
1.4 The Floating Prescaler usage with Compare operations 7
1.5 Reload of Floating Prescaler parameters 8
1.5.1 Shadow Transfer of Compare and Floating Prescaler values 8
1.6 Top-Level Control of Event Requests to/from a Timer with Floating Prescaler 8

2 Getting Started with CCU4 and Floating Prescaler 8
2.1 Ultimate Initialization Sequence 8
2.1.1 Ultimate Start-Up Enable 8
2.1.2 Prescaler Start-Up 8
2.2 Floating Prescaler Setup Control Registers 8
2.3 Floating Prescaler Shadow Transfer Control Registers 8
2.4 Using the Floating Prescaler Associated Control Register Bitfields 9

List of Figures

Figure 1 Timer Slice Prescaler with Normal or Floating Prescaler Capability Selectable 5
Figure 2 Floating Prescaler Principle 6
Figure 3 Floating Prescaler for Automatic Dynamic Range Adaption in Time Measurement by Capture 7
Figure 4 Using Floating Prescaler for Pulse Width Control by Compare (e.g. PWM) 8

1 Autonomic Signal Measurement Range Adaption

Each CAPCOM4 (CCU40/../43) is a multi-purpose timer unit for many use cases. It is designed with repetitive structures with multiple timer slices that have all the same base functionalities. This allows the development of re-usable SW low level drivers. The main function blocks of 1 of the 4 CC4y slices on a CCU4x are shown here. See Figure 1.

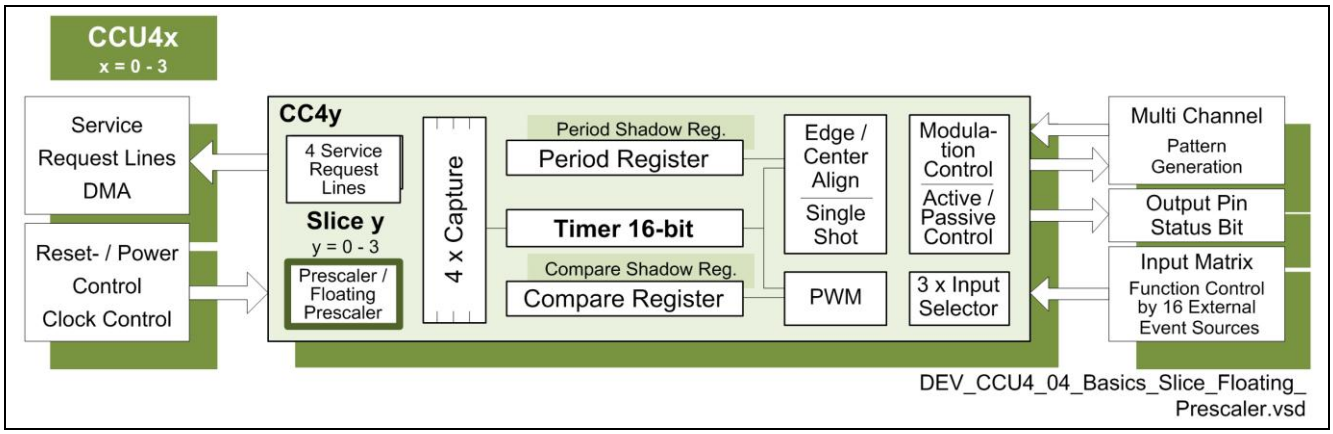

Figure 1 Timer Slice Prescaler with Normal or Floating Prescaler Capability Selectable

1.1 Floating Prescaler Use Cases

Floating Prescaler is in the first place designated signal monitoring by Capture but fits also Compare operations. By successive halving of the timer clock frequency periodically, when there was no capture (or compare) event, the dynamic range may be autonomously adapted to any time length, limited just by the timer configurations.

1.2 The Floating Prescaler Principle Function

The time diagrams for usage of floating prescaler in the different counting modes can be explained as follows. There is a *Prescaler Initial Value* (PSIV) table, which sets the start level of the *prescaler value* (PVAL) that will be *doubled periodically* as long as no timer event occurs – and then resumed again for a next monitoring cycle. See Figure 2.

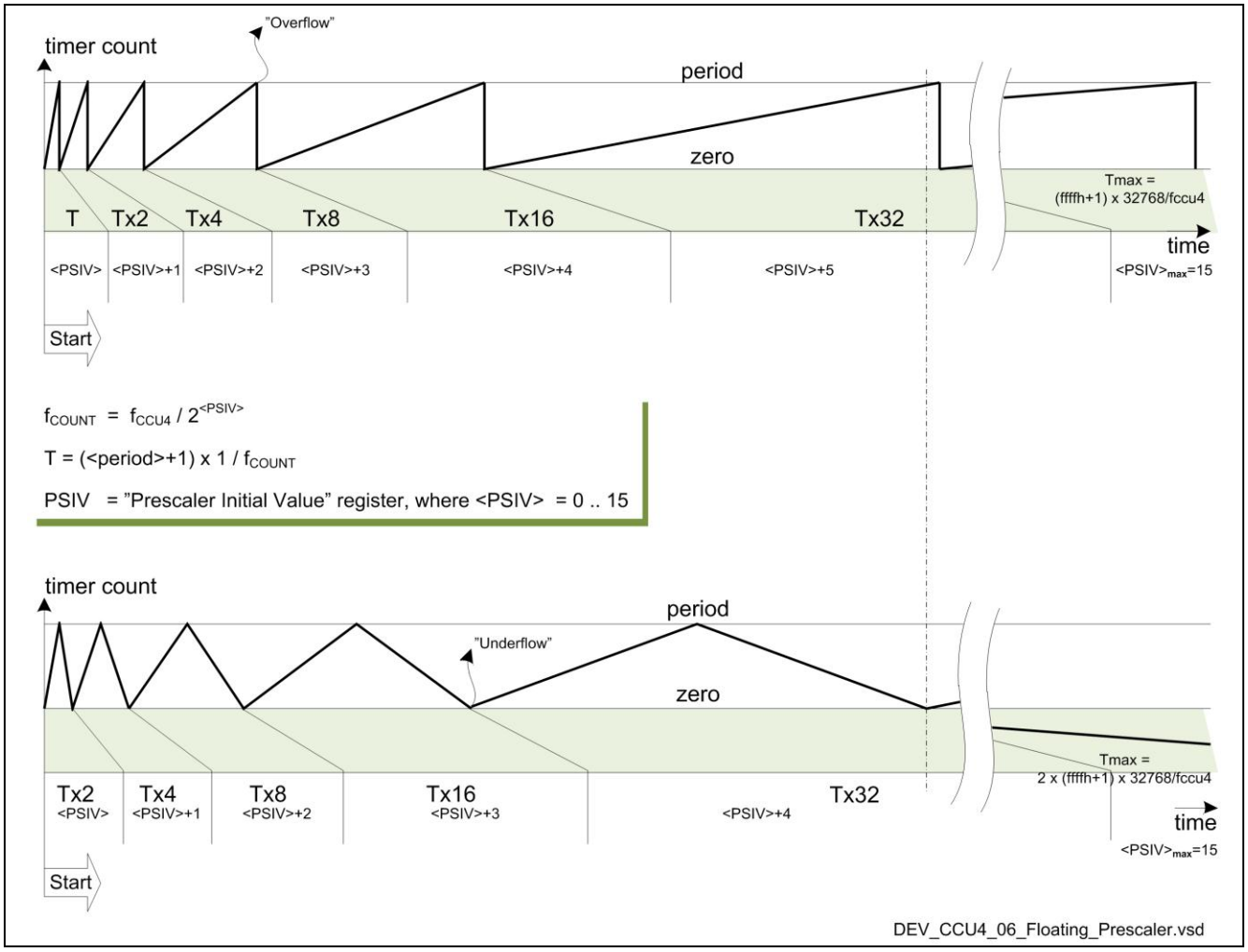

Figure 2 Floating Prescaler Principle

1.3 The Floating Prescaler in Signal Measurements by Capture

The cyclic capture procedure with floating prescaler is useful for signals with unpredictable events. By using floating prescaler in order to automatically tune into an appropriate magnitude of timer clock frequency, you are avoiding CPU load. Time calculation is based on all successively doubled period lengths and the capture value. See Figure 3.

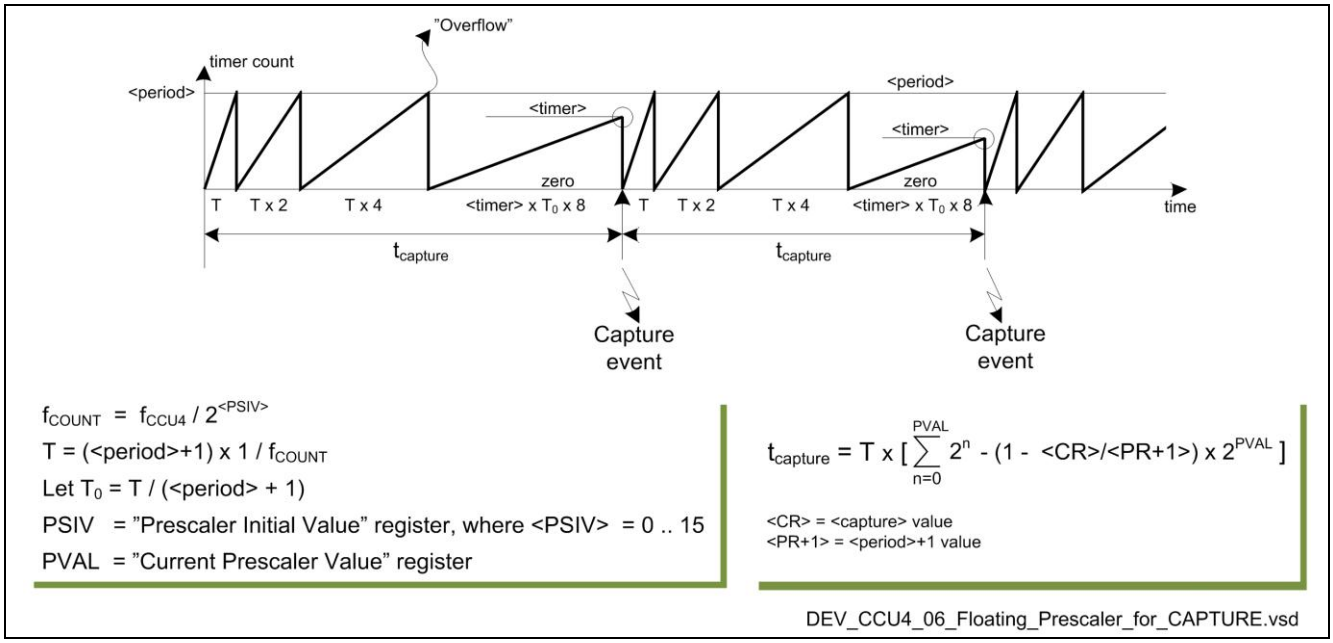

Figure 3 Floating Prescaler for Automatic Dynamic Range Adaption in Time Measurement by Capture

1.4 The Floating Prescaler usage with Compare operations

A signal time interval, a count range or PWM periods of arbitrary lengths may be achieved by using Floating Prescaler. Period or pulse lenghts may be changed coherently by shadow transfers. A timer compare event is due *if* it occurs during a period when the prescaler current value PVAL equals the preset PCMP compare value.

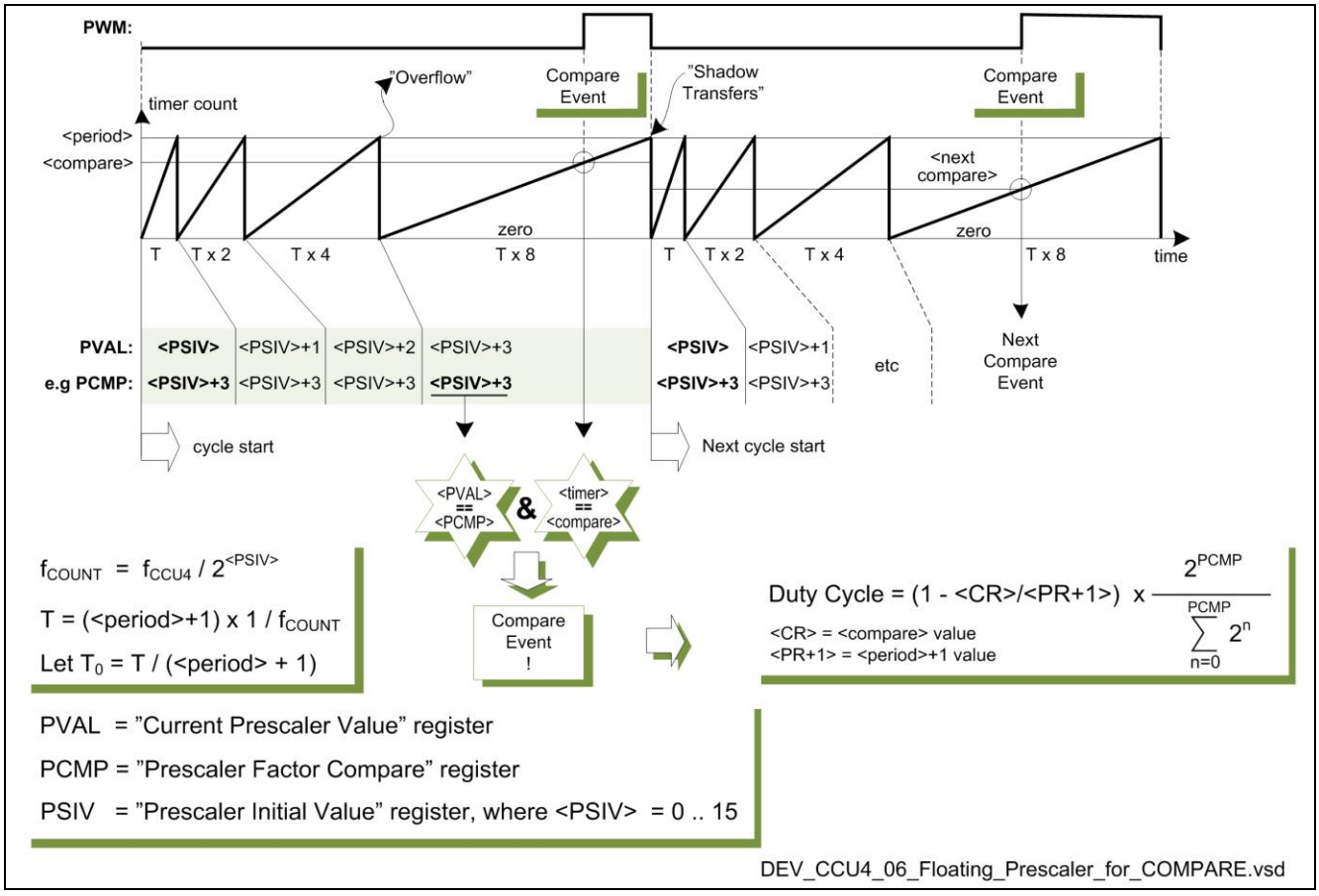

Figure 4 Using Floating Prescaler for Pulse Width Control by Compare (e.g. PWM)

1.5 Reload of Floating Prescaler parameters

All CAPCOM4 timers, in any slice configuration, no matter how complex or for what signal pattern, are assured *coherent update* by HW of all concerned timer function parameters – by update values in shadow registers that, on *a global* preset request, are transferred simultaneously to resp. function registers on **Period-** or **One-Match**.

1.5.1 Shadow Transfer of Compare and Floating Prescaler values

The compare values that are targeted for reload operations must be written in the **CC4yCRS** and **CC4yFPCS** shadow registers - *AND the respective* **Transfer Set Enable** bits, **SySE**, **SyPSE** in **GCSS**, *must be preset latest within the clock cycle of* **Period Match** (in *Edge Aligned Mode*) or **Period/One Match** (in *Center Aligned Mode*)

1.6 Top-Level Control of Event Requests to/from a Timer with Floating Prescaler

This scenario regards how to invoke *conditional* distribution of event requests signals to/from external actors. The Event Request Unit (**ERUx**) may combine events via the **Top-Level Interconnection Matrix**, whereby a conditional signal flow can be top-level controlled, according to a request-to-action event pattern set by the user.

2 Getting Started with CCU4 and Floating Prescaler

2.1 Ultimate Initialization Sequence

Before start and execution of a timer slice SW first time, the CAPCOM4 must have been initialized appropriately according to a specific **CAPCOM4 Initialization Sequence:** 1. *Apply Reset*, 2. *Release Reset*, 3. *Enable Clock*, 4. *Enable Prescaler Block*, 5. *Configure Global Control*, 6. *Configure Slice(s) Functions*, *Interrupts* and *Start-up*.

2.1.1 Ultimate Start-Up Enable

In the last part of the ultimate CAPCOM4 **Initialization Sequence** the startup value(s) for a specific Compare Channel Status of the Timer Slice(s) may be configured by the respective **GCSS.SyTS** bit. After that the default *IDLE mode has to be removed* from the Timer Slice(s) in register **GIDLC** and Start or Global Start can be done!

All these topics are covered in detail by the Reference Manual: <<< Link to 23.6.1 Initialization Sequence >>>

2.1.2 Prescaler Start-Up

The run bit of the prescaler can be set or cleared by SW by writing into the register bits **GIDLC.SPRB** and **GIDLS.CPRB** for set and clear respectively. One may also use the run bit of a specific slice by making it responsible for the set/clear. To select the slice to perform this, one should program the **GCTRL.**PRBC bitfield.

2.2 Floating Prescaler Setup Control Registers

Each **CCU4xCC4y** slice has an individual register bitfield **CC4yPSC.**PSIV as *index* to select a clock frequency for each timer in the **Timer Clock Division Table**. Each register bit **CC4yTC.**FPE selects *normal* or *floating* prescaler mode - and adds a *compare register* **CC4yFPC**.PCMP plus its *shadow register*, if *floating* mode is set.

2.3 Floating Prescaler Shadow Transfer Control Registers

The reload value in the **Shadow Register** bitfield **CC4yFPCS.**PCMPS **AND** the corresponding **Slice Transfer Set Enable** bit **SyPSE** in the **GCSS** register *must be written latest within the clock cycle of* a **Period Match** or **Period/One Match** (regarding *edge* or *center aligned mode*) to get a prescaler compare register PCMP update!

2.4 Using the Floating Prescaler Associated Control Register Bitfields

1. **CC4yPSC.**PSIV Look-up index for a *Prescaler Select Initial Value*, used as start value in floating mode.
2. **CC4yTC.**FPE Select FPE =1 for ***F****loating* ***P****rescaler* ***E****nable* (else default mode: *Normal*, by FPE= 0).
3. **CC4yPSC.**PVAL *Current Prescaler Value* registers bitfield, read/writable by the user. Updated by HW.
4. **CC4yPSC.**PCMP *Floating Prescaler Compare* value. The comparison is triggered by the compare match.
5. **CC4yFPCS.**PCMPS *Floating Prescaler Compare Shadow* registers bitfield.
6. **GCSS.**SyPSE Slice y (y=0-3) Prescaler Shadow transfer Set Enable registers bit.

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

Published by Infineon Technologies AG