XMC4000

XMC4500 Family

CCU4 and Capture Mode Features

Advanced Signal Measurement

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 Advanced Signal Measurement 5
1.1 Slice Timer Setup in Capture Mode 5
1.2 The Capture Algorithm 6
1.2.1 Disabling the Full-Flags 6

2 Capture by External Events Control 6
2.1.1 Timer Inputs for Capture 7
2.2 External Control by Capture Events 7
2.3 Top-Level Control of Event Requests to/from a Timer in Capture Mode 7

3 Getting Started with CCU4 and Capture Mode Features 8
3.1 Ultimate Initialization Sequence 8
3.1.1 Ultimate Start-Up Enable 8
3.2 External Events Control Setup Registers 8
3.2.1 Using the External Capture Control Register Bitfields 8
3.2.2 Pseudo Code for the Capture0 or Capture1 Setup 8

List of Figures

Figure 1 Timer Slice four Capture Registers 5
Figure 2 Slice Capture Logic 6
Figure 3 Capture by External Events Control 7

1 Advanced Signal Measurement Advanced Signal Measurement

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 is shown here.

See Figure 1.

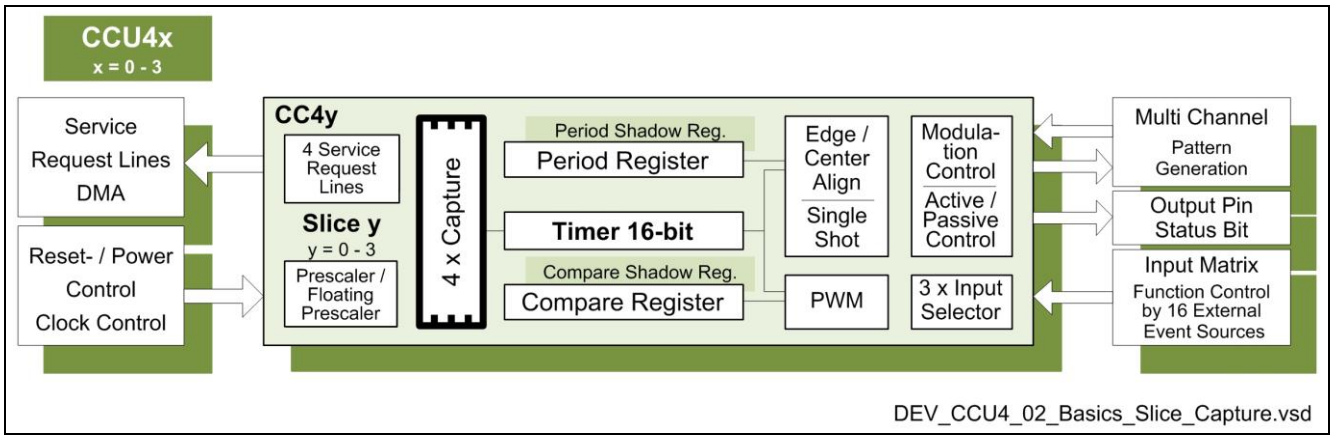

Figure 1 Timer Slice four Capture Registers

1.1 Slice Timer Setup in Capture Mode

There are 4 CAPCOM4 (CCU4x) with 4 timer-slices each - and each slice has 4 *capture value registers*, split into 2 pairs that will capture on selected event control input: *Capt0* or *Capt1* – according to 2 possible pair schemes: either as 2 pairs for *different* events to *Capt0* resp. *Capt1* or cascaded for the *same* event via Capt1.

See Figure 2.

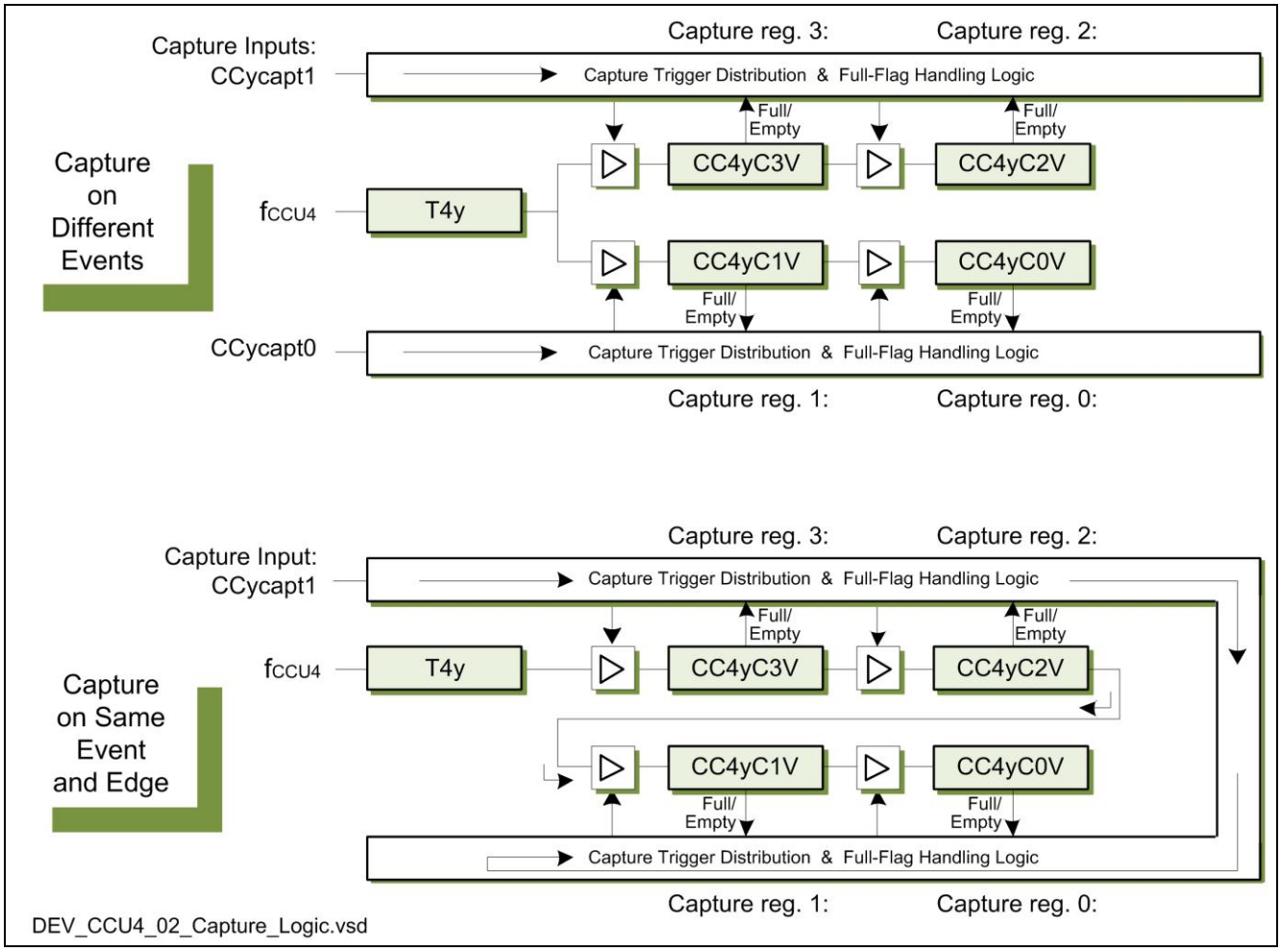

Figure 2 Slice Capture Logic

1.2 The Capture Algorithm

Each capture register has a *Full-Flag* that will be set on a *capture-to* and cleared on a *read-from* the register. At a Capture Input Event (*Capt1* or *Capt0*), each register will capture data from the next higher indexed register - *IF* that higher one is full *AND* a lower indexed register than that is empty. The timer is seen as of highest index.

1.2.1 Disabling the Full-Flags

A continous capturing without any effect from Full-Flags may be set by setting the bit CC4yTC.CCS = 1. In this case registers will capture data on the Capture Input Events without regarding the Full-Flags status.

2 Capture by External Events Control

This scenario regards how to link the *Capture0* or the *Capture1* register pairs to external trigger requests – from any of the following: *GPIO*, *ERU*, *POSIF*, *CAN*, *CCU4x*, *USIC*, *ADC*, *CCU8x* ***or*** *SCU*. A **Connection Pin Table** is given by the **Top-Level Interconnection Matrix**. A setup is done by the registers **CC4yINS** and **CC4yCMC**.

See Figure 3.

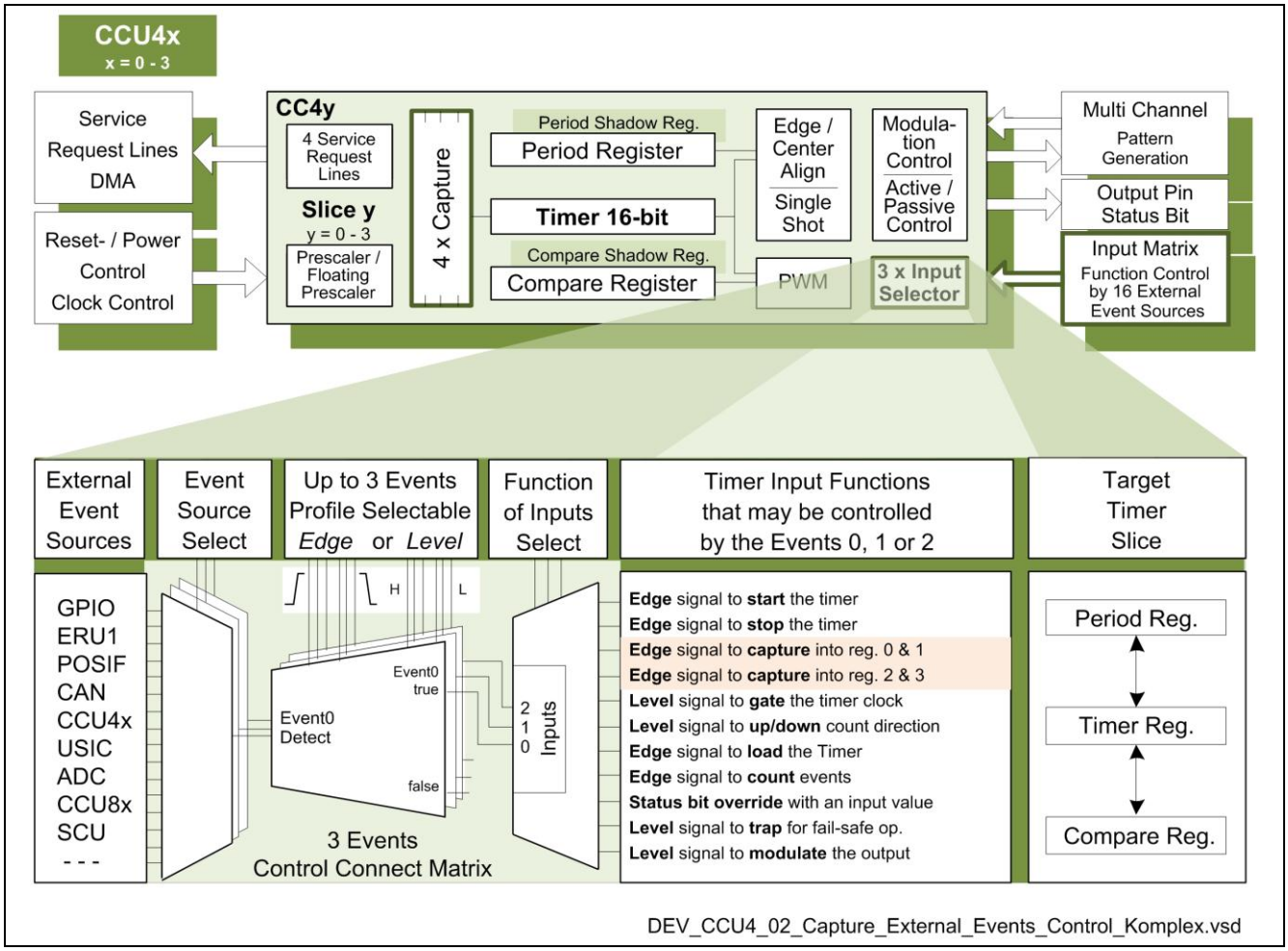

Figure 3 Capture by External Events Control

2.1.1 Timer Inputs for Capture

There are 3 selectable input lines with configurable *source-event-condition* profiles available for Capture by external event control functions, extendable in the register **CC4yTC** and a read access register **ECRD** too that simplifies administration of capture registers and full-flags, when more than one slice is used in Capture mode.

2.2 External Control by Capture Events

A capture event may *trigger external actions* via the **Top-Level Interconnect** matrix or *request for an Interrupt*. Each CAPCOM4 has four **Service Request Lines** and each slice has a dedicated output signal **CC4ySR[3...0]** selectable to a line by **CC4ySRS.** E.g. a capture event may request for some other unit action or for an interrupt.

2.3 Top-Level Control of Event Requests to/from a Timer in Capture Mode

Top-Level control means also *conditional* control of events requests between a slice and other action providers. The Event Request Unit (**ERU1**) *plus* the **Top-Level Interconnect** matrix may *combine, control and link* event signals according to *user defined* request-to-action event patterns - e.g. capture on timer *and* other event status.

3 Getting Started with CCU4 and Capture Mode Features

3.1 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*.

3.1.1 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 >>>

3.2 External Events Control Setup Registers

The *Input Selection* register **CC4yINS** maps *External-Event-Sources* and *Edge/Level Mode* conditions to the *Event[2:0] Input Lines*. In turn the *Connection Matrix Control* register **CC4yCMC** maps any *Event[2:0] Input Line* to any of the 11 *Input Functions*. The registers **CC4yTC** + **ECRD** enable *Extended Capture Read Mode* options.

3.2.1 Using the External Capture Control Register Bitfields

1. **CC4yCMC.**CAP0S — Input Capture0 Selection and Event Number x=0-2 Input Line Selection bitfield
2. **CC4yINS.**<Event-x bitfields> — Event-x Source EVxIS (**CC4yINS**[P:A]) and Edge Mode EVxEM bitfields
3. **CC4yCMC.**CAP1S — Input Capture1 Selection and Event Input Line Number x=0-2 Selection bitfield
4. **CC4yINS.**<Event-x bitfields> — Event-x Source EVxIS (**CC4yINS**[P:A]) and Level Mode EVxEM bitfields
5. **CC4yTC.**CMOD — CAPCOM Mode bit should be set =1. (CMOD=0 is Compare Mode by default)
6. **CC4yTC.**ECM — Extended Capture Mode enable bit (=0 by default, if no extended read mode)
7. **ECRD** — Extended Capture Read Register (A Global read-only register, written by HW)

Note: The External Capture functions are Edge Sensitive.

Note: When using the index 'x' in event nomenclature, it should be understood as 'Actual Event[2:0] Number '.

Note: When using the index 'x' in CCU4x nomenclature, it should be understood as 'Actual CCU4 Unit Number'.

Note: The Ultimate Event Number 'x' for the CAP0S Function is always x == <CAP0S>-1 (when <CAP0S> != 0).

Note: The Ultimate Event Number 'x' for the CAP1S Function is always x == <CAP1S>-1 (when <CAP1S> != 0).

3.2.2 Pseudo Code for the Capture0 or Capture1 Setup

```
// This is an independent 32-bit based Pseudo Code, regarding that there are no
connections to a certain CAPCOM4 Unit (CCU4x) nor specific Variable Declarations.
// step1
// Make MASK OPERATION first for bitfield focusing – if not first time – else:
CC4yCMC |= CAP0S_FUNCTION_AND_EVENT_NUMBER_BITFIELD //Init External Capture0
// step 2
CC4yINS &= CAP0S_SOURCE_AND_EVENT_MODES_BITFIELDS_MASK; // Focus on bitfields
CC4yINS |= CAPOS_SOURCE_AND_EVENT_MODES_BITFIELDS_PATTERN; //Paste setup pattern
// step3
// Make MASK OPERATION first for bitfield focusing – if not first time – else:
CC4yCMC |= CAP1S_FUNCTION_AND_EVENT_NUMBER_BITFIELD //Init External Capture1
```

```
// step 4
CC4yINS &= CAP1S_SOURCE_AND_EVENT_MODES_BITFIELDS_MASK; //Focus on bitfields
CC4yINS |= CAP1S_SOURCE_AND_EVENT_MODES_BITFIELDS_PATTERN; //Paste setup pattern

// step 1 and 3 could naturally be merged into one operation sequence, as follows:
// Make a MASK OPERATIONS first for bitfield focusing - if not first time - else:
CC4yCMC |= CAP01S_FUNCTIONS_AND_EVENTS_NUMBERS_BITFIELD //Init External Capture0&1

// step 2 & 4 could naturally be merged into one operation sequence, as follows:
CC4yINS &= CAP01S_SOURCES_AND_EVENTS_MODES_BITFIELDS_MASK; //Focus on bitfielfs
CC4yINS |= CAP01S_SOURCES_AND_EVENTS_MODES_BITFIELDS_PATTERN; //Paste both setups!
```

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

Published by Infineon Technologies AG