XMC4000

XMC4000 Family

POSIF Quadrature Decoder

Direction, Position and Revolution Decode

User Guide

Device Guide

<Revision 1.0>, <2012-08-06>

Microcontroller

Edition <2012-08-06>

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.

Revision History	
Page or Item	**Subjects (major changes since previous revision)**
<Revision 1.0>, <2012-08-06>	

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

1 Direction, Position and Revolution Decode 6
1.1 Quadrature Clock Generation Basics 6

2 Quadrature Decoder Mode 6
2.1 POSIF Pin Connections 6
2.1.1 Input Pin Connections in Quadrature Mode 7
2.1.2 Output Pin Connections in Quadrature Mode 7
2.1.3 Output Pin Functions in Quadrature Mode 8
2.1.4 Synchronous Start Output Pin in Quadrature Mode 8

3 Direction Count Mode 8

4 POSIF Interrupt Registers 8
4.1 POSIF Interrupts/Event requests in Quadrature Decoder Mode 8
4.2 POSIF Interrupts/Event requests in Multi-Channel Mode 8

5 Getting Started with POSIF Quadrature Decoder 9
5.1 POSIF Global Registers 9
5.2 POSIF Setup when using Quadrature Decoder Mode 9
5.3 POSIF Position Decoder Mode selection 9
5.3.1 POSIF Input Selections in Quadrature Mode 9
5.3.2 POSIF Operation Mode Associated Registers 9
5.3.3 POSIF Quadrature Decoder Control (QDC) Modes 9
5.4 POSIF Associated Configurations of CCU4x/-8x Timers 9
5.4.1 POSIF Synchronized Start of Timers by HW Initialization 9
5.4.2 Initialize and Start the Timer by SW 10
5.5 Timer Setup 10
5.5.1 Ultimate Initialization Sequence 10
5.5.2 Ultimate Start-Up Enable 10
5.5.3 Setting the Timer Counting Modes 10
5.6 Using the POSIF Global Registers for Quadrature Decoder Mode Setup 10
5.7 Using the POSIF Global and Multi-Channel (MC) Mode Setup Registers 10
5.8 Using the Timer Control Registers for COMPARE Mode 11
5.9 Using the Timer Control Registers for CAPTURE Mode 11
5.10 Using the External Events Control Setup Registers 11
5.10.1 External Event Control Register Bitfields 11
5.10.2 Using the External Start Control Register Bitfields 11
5.10.3 Using the External Capture Event Control Register Bitfields 12
5.11 Getting Started Example 12
5.11.1 Timer Input Connections 12
5.11.2 Timer Input Functions 12
5.11.3 Setup Strategy 12
5.12 Timer Count and Direction Control Setup (example) 13

List of Figures

Figure 1 Standard Quadrature Clock Decoding 6
Figure 2 POSIF in Standard Quadrature Mode (when just using position monitoring) 7
Figure 3 POSIF in Standard Quadrature Mode 7
Figure 4 POSIF in Direction Count Mode 8

1 Direction, Position and Revolution Decode

A POSIFx (x=0-1) is a universal *Position Interface* unit. In conjunction with the CAPCOM units CCU4 or CCU8 it offers powerful solutions for motion control systems that use various position sensors or rotary encoders in the feedback loop. The *Quadrature Decoder* is one of the main subunits beside the *Hall* and *Multi-Channe* subunits.

1.1 Quadrature Clock Generation Basics

See Figure 1.

Each Quadrature Clock Cycle is a 4-state, 4 clocks pulse train that is generated by two phase-shifted signals from a sensor-pair, (A) and (B). Since the state order is significant for which sensor that is in the sense position of the leading phase, each state transition is unique for the motion direction: *clockwise* or *counterclockwise*.

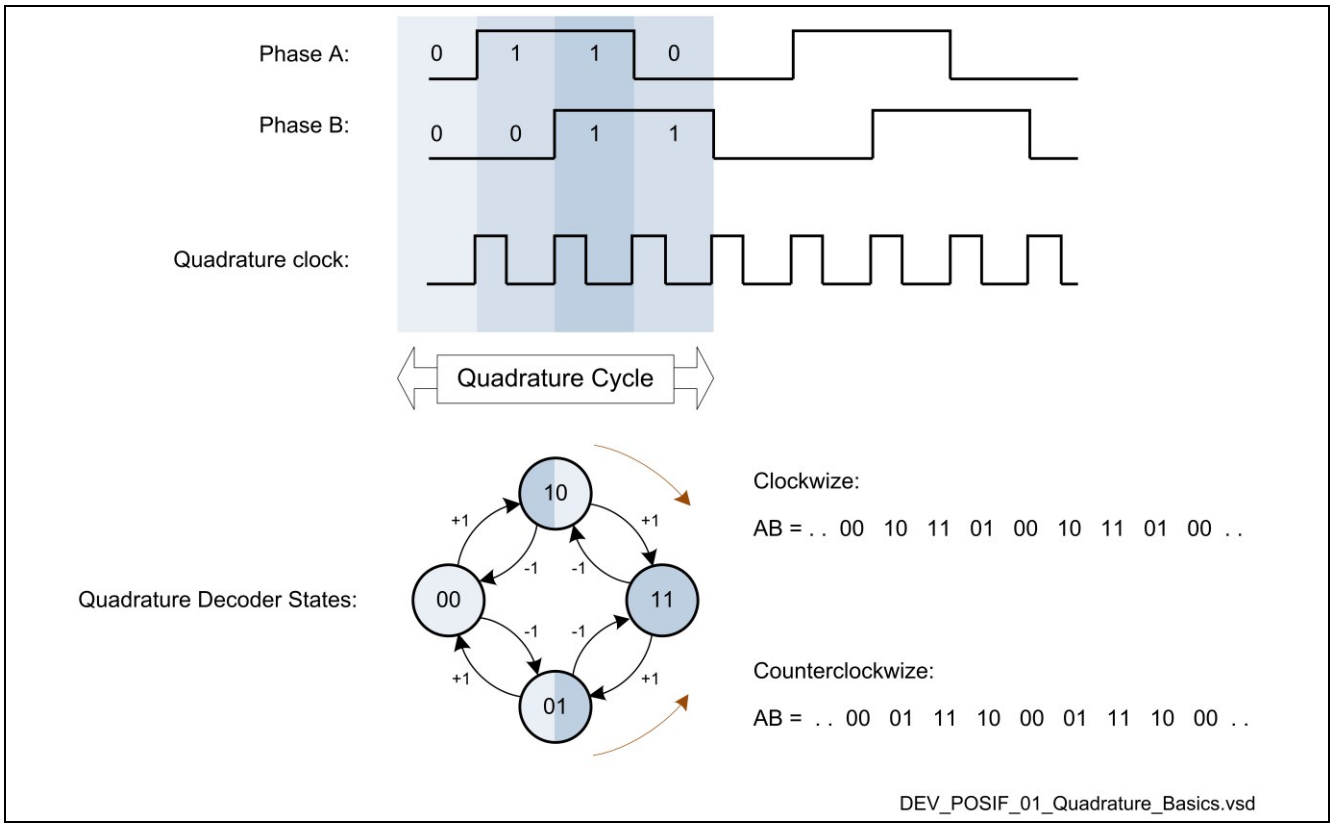

Figure 1 Standard Quadrature Clock Decoding

2 Quadrature Decoder Mode

2.1 POSIF Pin Connections

Sinse the POSIF module has flexible configuration schemes, in order to cover a large universe of motion control applications, the *Pin Connections* have *Alternative Functions* - depending on the bitfield **PCONF**.FSEL setup:

00_B: Hall Sensor Mode,

01_B: **Quadrature Decoder Mode**,

10_B: Stand-alone Multi-Channe Mode,

11_B: Stand-alone Multi-Channe & **Quadrature Decoder Mode**

2.1.1 Input Pin Connections in Quadrature Mode

A freely selectable pin in each quad input group POSIFx.IN**0**[D...A], POSIFx.IN**1**[D...A]) and POSIFx.IN**2**[D...A] may represent the *phase A*, *phase B* and *Index*/*Zero-Mark signal* respectively. These pins are not used for *Multi Channel Mode* specifically - thus they can be used in a *combined Multi-Channe & Quadrature Decoder Mode*.

See Figure 2

See Figure 3.

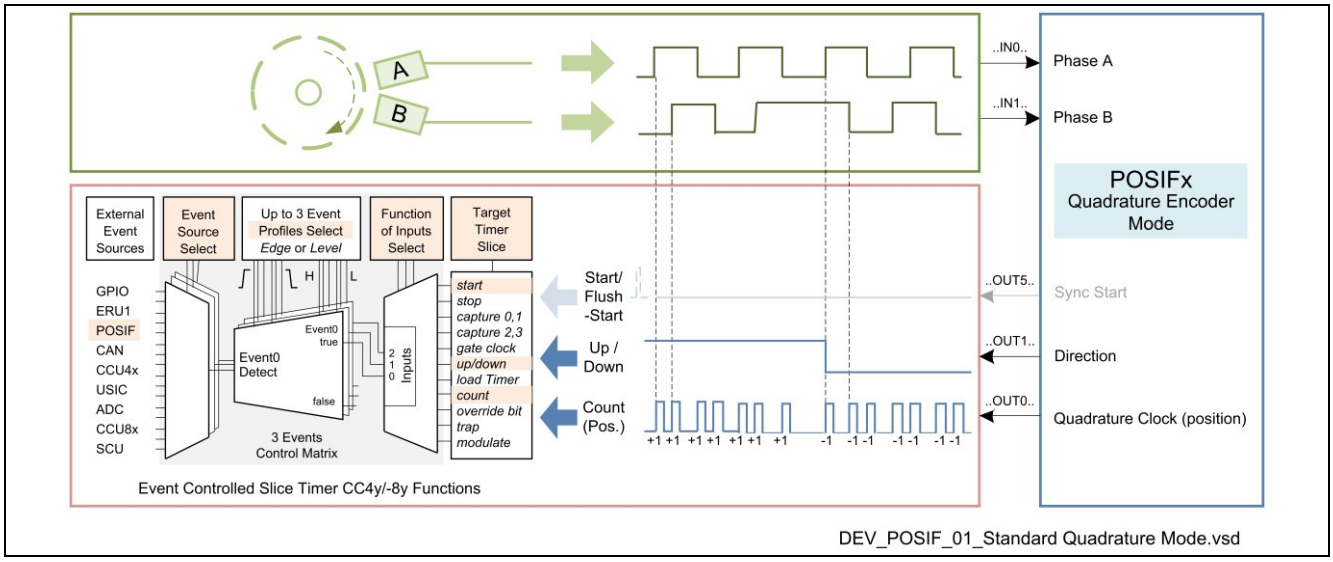

Figure 2 POSIF in Standard Quadrature Mode (when just using position monitoring)

2.1.2 Output Pin Connections in Quadrature Mode

There are six output pins, POSIFx.OUT0 - POSIFx.OUT5, handling the following Output Functions in this order: **0:** *Quadrature Clock*, **1:** *Direction*, **2:** *Period clock*, **3:** *Index Occurrence*, **4:** *Index* (*revolution*) and **5:** *Synchronous Start*. These pins can be used in the combined *Multi-Channe & Quadrature Decoder Mode* too.

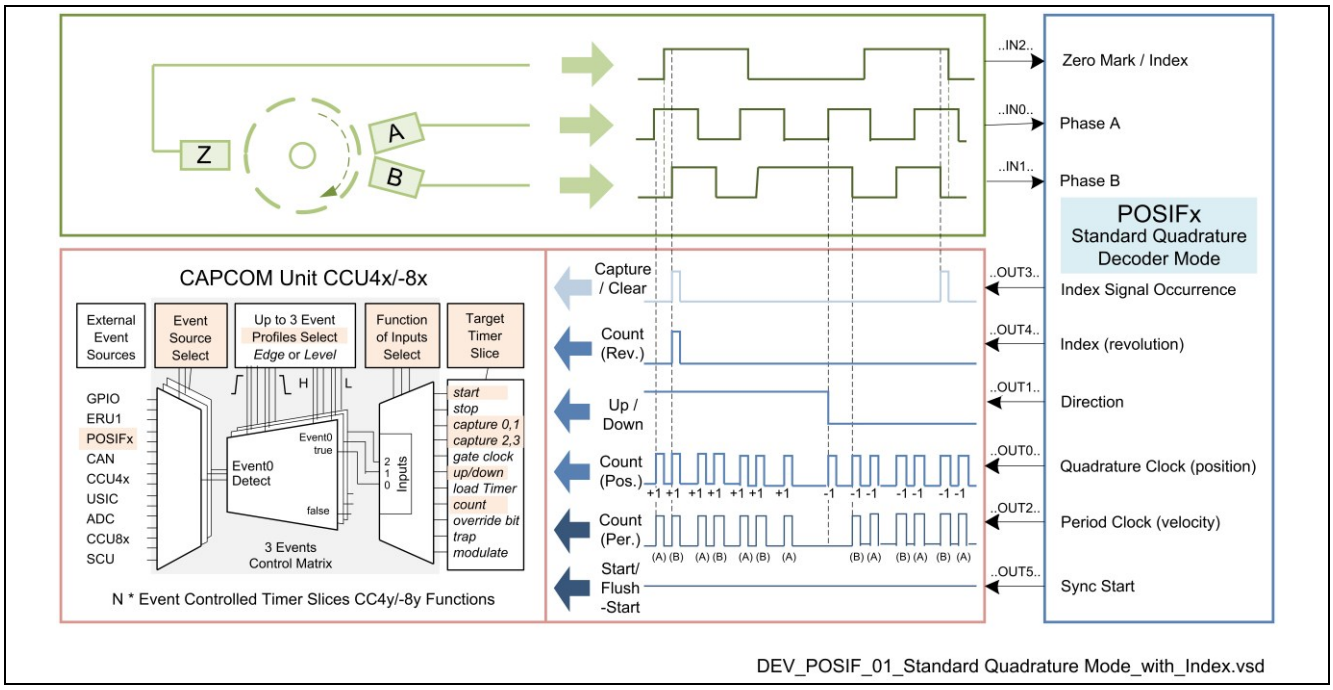

Figure 3 POSIF in Standard Quadrature Mode

2.1.3 Output Pin Functions in Quadrature Mode

The *Quadrature clock* gives *position* up/down count information in conjunction with the *Direction* signal, whilst the *Period* clock contains sequences of (*AB*) or (*BA*) pulse-pairs for *velocity* detection. The *Index Occurrence* pin asserts *Z-mark/Index* (on **QDC**.ICM conditions), whilst the *Index revolution* asserts *only once* per revolution.

2.1.4 Synchronous Start Output Pin in Quadrature Mode

The POSIF module has a *Synchronous Start* output, POSIFx.OUT5, that is linked with the module run bit. This can be used for a complete synchronous start together with the CAPCOM units. However, those CAPCOM units have to be preset for *External Events Control*, e.g. *External Start* and *Extended Start* for *Start* or *Flush/Start* etc.

3 Direction Count Mode

See Figure 4.

Position encoders that work in *Direction Count Mode* provide just two signals: *Clock* and *Direction* information. The POSIFx gets the *Direction Count Mode* by setting **PCONF**.QDCM =1_B. The inputs should be mapped to POSIFx.IN**0**[D...A]/ POSIFx.IN**1**[D...A]); The output pins are POSIFx.OUT0/-1/-5 for Clock/Direction/Sync.-Start.

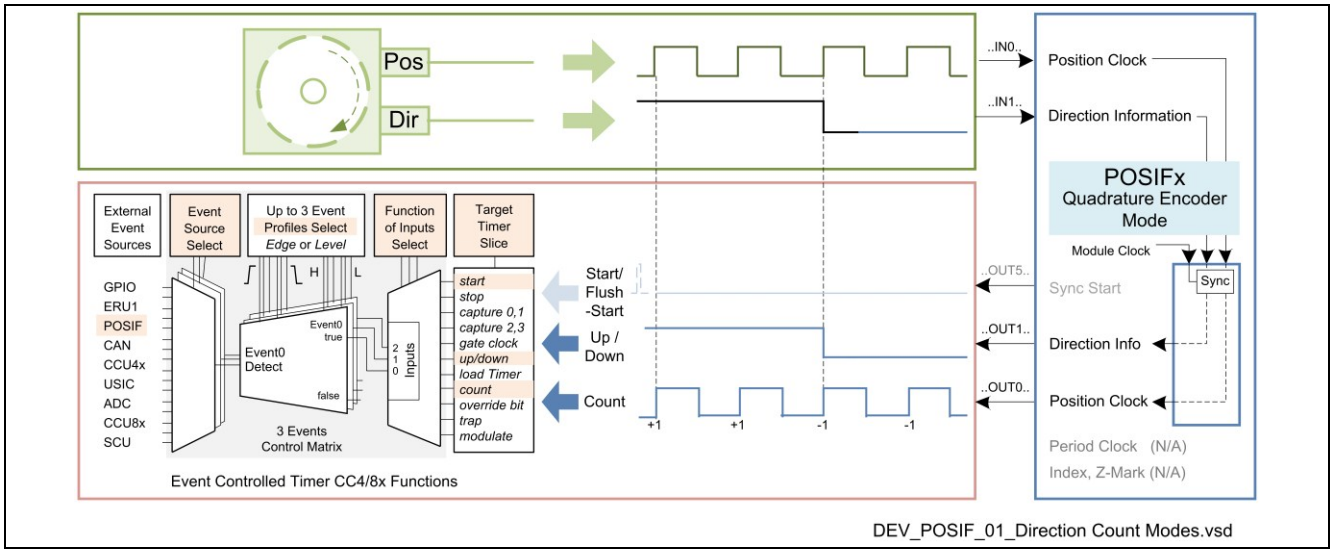

Figure 4 POSIF in Direction Count Mode

4 POSIF Interrupt Registers

PFLG	POSIF Interrrupt Flags		
PFLGE	POSIF Interrrupts Enable		
SPFLG	Set POSIF Interrrupt Flag	**RPFLG**	Reset POSIF Interrrupt Flag

4.1 POSIF Interrupts/Event requests in Quadrature Decoder Mode

Five interrupts/event request flags (+ two flags, on a *Multi-Channe Mode* combination) can be enabled by the **PFLGE** register bits: *Index event detection* (**INDXS**), *phase detection error* (**ERRS**), *quadrature clock generation* (**CNTS**), *period clock generation* (**PCLKS**), *direction change* (**DIRS**). Links to the SR1- / SR2-line are selectable.

4.2 POSIF Interrupts/Event requests in Multi-Channel Mode

The MC Mode interrupt/event request flag (**MSTS**) in register **PFLG** can be enabled by the **PFLGE** register bit *MC Shadow Transfer Enable* (**EMST**). This flag (**MSTS**) status should be mapped to the POSIFx.SR0 /-SR1 *Service Request Line* by the **PFLGE** register bit *MC Pattern Update Event Service Request Selector* (**MSTSEL**).

5 Getting Started with POSIF Quadrature Decoder

5.1 POSIF Global Registers

PCONF	POSIF Global **Control**	All modes HW and Protocol Controlling Register
PSUS	Suspend Configuration	Suspend configuration register for *stop* / *sync:d stop*
PRUNS	POSIF run bit **Set**	Via this register it is possible to set the module run bit
PRUNC	POSIF run bit **Clear**	Via this register the run bit or POSIF status can be reset
.		
PRUN	POSIF run bit Status	Run bit status can just be read: 0_B=*IDLE* / 1_B=*running*
MIDR	Module ID register	Module Identification register: *Revision*, *Type*, *Number*

5.2 POSIF Setup when using Quadrature Decoder Mode

After activation of a POSIF unit by the system start-up procedures (after hard reset), the POSIF control register **PCONF** should be written to for the Function Selection, i.e. the bitfield **PCONF**.FSEL = 01_B for *Quadrature Decoder Mode* enable (or a combined *Quadrature Decoder & Stand Alone Multi-Channe Mode* enable by 11_B).

5.3 POSIF Position Decoder Mode selection

First of all it should be declared if the *Standard Quadrature Mode* or the *Direction Count Mode* should be used. This can be selected by the bitfield **PCONF**.QDCM. Default (=0) implies the *Standard Quadrature Mode*. If the QDCM=1, then the POSIFx.IN**0**[D...A] and POSIFx.IN**1**[D...A]) are *clock* and *direction* signal inputs respectivally.

5.3.1 POSIF Input Selections in Quadrature Mode

The **PCONF.**INSEL0, **PCONF.**INSEL1 and **PCONF.**INSEL2 bitfields map the input pins POSIFx.IN**0**[D...A], POSIFx.IN**1**[D...A]) and POSIFx.IN**2**[D...A] to the *phase A*, *phase B* and *Index*/*Zero-Mark signals* respectively. Additionally, a *Low-Pass Filters* configuration of 0 - 64 clock cycles can be setup by the **PCONF.**LPC bitfield.

5.3.2 POSIF Operation Mode Associated Registers

For *Quadrature Decoder Mode*: use **QDC**

For *Multichannel functions* : use additionally **MCSM/MCM** for data / **MCMF/MCMS/MCMC** for control

5.3.3 POSIF Quadrature Decoder Control (QDC) Modes

- Phase A and B active levels - (Default (=0): "*high*" ,individually selectable for phase A and phase B)
- Swap of Phases A and B as leading phase for Clockwise Rotation - (Default (=0): "*A is leading phase*")
- Index/Z-Mark Signal Occurrence on the POSIFx.OUT3 pin - (Default (=0): "*none*", (01_B): "*1st one*", (10_B): "*all*")

5.4 POSIF Associated Configurations of CCU4x/-8x Timers

These modes are mainly controlled by the **CC4yTC** (**CC8yTC**) registers – and the *External Events Control* of e.g. a **CC4y** can be set by **CC4yCMC** and **CC4yINS** registers for *Start*, *Flush/Start*, *Capture*, *Capture/Clear* etc.

- See ***Timer Setup***.

5.4.1 POSIF Synchronized Start of Timers by HW Initialization

After the Ultimate Timer Start procedures have been executed - and If the synchronous start function of the POSIF is being used inside the associated Capture/Compare units, then one just needs to set the *Run Bit* of the module, by writing a 1_B into the **PRUNS**.SRB bit.

5.4.2 Initialize and Start the Timer by SW

After the Ultimate Start procedures have been executed, the SW may 1) initialize the slice timer **CC4yTIMER** or just clear it with a **CC4yTCCLR**.TCC bit=1 command and 2) start the timer with a Set Timer Run command in the **CC4yTCSET**.TRBS bit. The timer may be stopped at any time by SW with a **CC4yTCCLR**.TRBC command.

5.5 Timer Setup

In the following initialization and setup descriptions below there have been chosen the CCU4x CAPCOM units as associated Capture/Compare units for the preceedings of the Getting Started demonstration with the POSIF.

5.5.1 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*.

5.5.2 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!

5.5.3 Setting the Timer Counting Modes

The Timer Counting Mode is *Edge Aligned* by default, i.e. **CC4yTC.**TCM=0_B. The *Center Aligned* mode will be set by TCM= 1_B. Single Shot Mode is set by the **CC4yTC.**TSSM bit. *Start* (and even *Stop*) may be controlled by External Events, added by extended functionalities e.g. *Start, Extended Start respective Stop, Extended Stop*.

5.6 Using the POSIF Global Registers for Quadrature Decoder Mode Setup

The following Quadrature Decoder Mode Protocol Controlling Global Registers should be setup:

1. **PCONF**.FSEL Select Mode: 01_B for *Quadrature Decoder Mode*, 11_B for *combined MC & QDR Mode*
2. **PCONF**.QDCM Select *Position Decoder Mode* - 0_B for *Quadrature Mode*, 1_B for *Direction Count Mode*
3. **PCONF**.LPC Select *Low Pass Filters Configuration* on the Quadrature Inputs: *0 - 64 clock cycles*

The following Quadrature Decoder Mode Registers should be considered:

4. **QDC**.PALS Select *Phase A Level* - 0_B for Phase A is *active HIGH*, 1_B for Phase A is *active LOW*
5. **QDC**.PBLS Select *Phase B Level* - 0_B for Phase B is *active HIGH*, 1_B for Phase B is *active LOW*
6. **QDC**.PHS Select Phase: *Leading signal for clockwise rotation* - 0_B for Phase A, 1_B for Phase B
7. **QDC**.ICM Select *Index Marker Control* - 00_B = *none*, 01_B = 1st occurrence, 10_B = *all* occurrences
8. **QDC**.DVAL *Read Only*: Current rotation direction - 0_B = *Counterclockwise*, 1_B = *Clockwise* rotation

5.7 Using the POSIF Global and Multi-Channel (MC) Mode Setup Registers

The following Multi-Channel (MC) Protocol Controlling Global Registers may be setup:

9. **PCONF**.MCUE Select MC Pattern Update Enable control: “by HW” (default) by 0_B or “by SW” by 1_B
10. **PCONF**.MSETS Select the input POSIFx.**MSET[H...A]** signal that is used to enable a MC pattern update
11. **PCONF**.MSYNS Select the PWM synchronization input POSIFx.**MSYNC[D...A]** for a MC pattern update
12. **PCONF**.MSES Select the PWM synchronization eignal trigger edge: 0_B for *rising* / 1_B for *falling* edge

The following Multi-Channel (MC) Mode Registers may be considered:

13. **MCM**.MCMP MC Mode Pattern Contains *MC Pattern Out* applied to POSIFx.**OUT[15:0]**
14. **MCSM**. MCMPS MC Mode Shadow Pattern This is passed into the **MCM**.MCMP on *Transfer* events
15. **MCMF**.MSS MC Mode **Flag** status *Read Only*: *MC Update Trigger* request bit *F/F* status

16. **MCMS**.<bitfields> MC Mode Control **Set** Perform ***Immidiate*** *Pattern Update* - or just set *Enable*
17. **MCMC**.<bitfields> MC Mode Control **Clear** Clears the *MC Pattern Out*, the **MCMF**.MSS bit - or both

5.8 Using the Timer Control Registers for COMPARE Mode

There is practically just one and only register that is concerned here – namely the **CC4yTC** register bitfields:

18. **CC4yTC.**CMOD CAPCOM Mode selection bit may be used: 0 → *Compare* / 1 → *Capture* mode
19. **CC4yTC.**TCM Timer **Counting Mode selection** bit: 0 → *Edge-Aligned* / 1 → *Center-Aligned*

Note: If External Events Control of Up/Down Counting is used – then the counting scheme is always edge aligned and the Timer Counting Mode bit (TCM) is read-only!

20. **CC4yPR.**PR Write the timer count cycle upper limit into the *period* register
21. **CC4yCRS.**CRS Write the wanted *Compare Level* value into the compare *shadow* register

Options:

22. **CC4yPSL.**PSL *PASSIVE/ACTIVE* control bit: (*CCU4xOUTy)* = (*CCUxSTy*) xor (*CC4yPSL.PSL*)
23. **CC4yTC.**STRM *Extended Start* on *External Start*. 0_B: *Start (=Set Run Bit)* / 1_B: *Flush* & *Start* Timer
24. **CC4yTC.**ENDM *Extended Stop* on *Ext. Stop*. 00_B: *Stop* / 01_B: *Flush* / 10_B: *Stop* & *Flush* /11_B:reserved
25. **CC4yTIMER** Initialize a timer start value (or *flush* by the **CC4yTCCLR**.TCC bit=1 command)
26. **CC4yTCSET**.TRBS This is the Start Timer by SW command.

5.9 Using the Timer Control Registers for CAPTURE Mode

There is practically just one register that is concerned here as well – namely the **CC4yTC** register bitfields:

1. **CC4yTC.**CMOD *CAPCOM Mode* bit should be set =1. (CMOD=0 is Compare Mode by default)
2. **CC4yTC.**TCM Timer **Counting Mode selection** bit: 0 → *Edge-Aligned* / 1 → *Center-Aligned*
3. **CC4yTC.**CAPC *Clear* Timer on *Capture* event: 0: *never* / 1: on *Cap0* / 2: on *Cap1* / 3: *always*
4. **CC4yTC.**SCE 0: *different* events to *Capt0* and *Capt1* / 1: *same* event cascaded via Capt1.

Note: When (SCE =1) & (CAPC !=0) then the Timer is always cleared in a capture event!

Note: If External Events Control of Up/Down Counting is used – then the counting scheme is always edge aligned and the Timer Counting Mode bit (TCM) is read-only!

5. **CC4yPR.**PR Write the timer count cycle upper limit into the *period* register

Options:

6. **CC4yTC.**CCS 0: capture rules are linked with the *Full Flags* / 1: *Continuous Capture* enabled
7. **CC4yTC.**ECM *Extended Capture Mode* enable bit (=0 by default → no extended read mode)
8. **CC4yTC.**STRM *Extended Start* on *External Start*. 0_B: *Start (=Set Run Bit)* / 1_B: *Flush* & *Start* Timer
9. **CC4yTC.**ENDM *Extended Stop* on *Ext. Stop*. 00_B: *Stop* / 01_B: *Flush* / 10_B: *Stop* & *Flush* /11_B:reserved
10. **CC4yTIMER** Initialize a timer start value (or *flush* by the **CC4yTCCLR**.TCC bit=1 command)

5.10 Using the 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 the three *Event[2:0] Input Lines* to any of the 11 *Input Functions*. The *Timer Control* register **CC4yTC** enables different *Mode Alternatives*.

5.10.1 External Event Control Register Bitfields

1. **CC4yCMC.**<function> Input *function* selection and Event Number (x=0-2) Input Line Selection bitfield
2. **CC4yINS.**<Event-*x* bitfields> Event-x *source* EVxIS (**CC4yINS**[P:A]), *edge* EVxEM or *level* EVxLM *mode*
3. **CC4yTC.**<Mode bitfields> Timer Mode selections e.g. Modes/*Extended modes, Counting Schemes etc*

5.10.2 Using the External Start Control Register Bitfields

1. **CC4yCMC.**STRTS Input Function Selection and Event Number x=0-2 Input Line Selection bitfield
2. **CC4yINS.**<Event-*x* bitfields> Event-x Source EVxIS (**CC4yINS**[P:A]) and Event Edge Mode EVxEM bitfields
3. **CC4yTC.**<bitfields> Timer Mode selections for e.g. *Start, Extended start, Edge or Center Align, etc*

5.10.3 Using the External Capture Event 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

Options:

5. **CC4yTC.**STRM *Extended Start* on *External Start*. 0_B: *Start (=Set Run Bit)* / 1_B: *Flush&Start*
6. **CC4yTC.**ENDM *Extended Stop* on *Ext. Stop*. 00_B: *Stop* / 01_B: *Flush* / 10_B: *Stop&Flush* /11_B: res.

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).

5.11 Getting Started Example

5.11.1 Timer Input Connections

To select an external control of an input function, map one of the 3 inputs to the wanted event source signal by selecting the source number in the **CC4yINS.EVxIS** field and indicate the *active edge* in the **CC4yINS.EVxEM** (or *active level* in **.EVxLM**) field. The "**x**" defines the *event number* (i.e. slice input 0, 1 or 2) that should be used.

- The External Events Control Source Number **EVxIS** (15-0) corresponds to the CCU4xINy[P:A] vector in the **Top-Level Interconnect** Matrix, where "**4x**" stands for CAPCOM Unit (40-43) and "**y**" is 0-3 for timer CC4y.

5.11.2 Timer Input Functions

To select which input function(s) that should be controlled by which event(s) of event 0, 1 or 2, the **CC4yCMC** fields should be programmed accordingly:

- **CC4yCMC.CNTS** for the COUNT function
- **CC4yCMC.UDS** for the UP/DOWN function

5.11.3 Setup Strategy

- Check which timers that can be controlled by the POSIF0 or POSIF1 for Direction and Position Count in Quadrature Encoder Mode.
- Secondly, knowing that control signals are transferred as event requests (and that a slice has 3 event associated function control inputs) one should:
    - Setup one event for the Position Count Control
    - Setup another event for the Direction Control.
    - The third/fourth event(s) may, optionally, be used for Index/Zero-Mark signals.

5.11.3.1 Available Timers for Quadrature Clock and Direction Control

According to the **Top-Level Interconnect** Matrix: Any CC40/-41 timer in any of the CAPCOM Units CCU40/-41 can perform the requested tasks.

5.11.3.2 Define the Event Profiles for Input Function Control Detection

According to the different nature of events:

- The Count Control event should be considered as a *Trigger Edge* (here active on a *falling edge*).
- The Direction Control event should be considered as *not inverted Level* (by active on *high level*).

5.12 Timer Count and Direction Control Setup (example)

As a proposal for this demo the following resources will be setup: POSIF0, CCU40, CC41.

POSIF:	**POSIF0**			CCU4**x**IN**y**(..):
	Quadrature Clock:	**POSIF0.OUT0**	connected to	**CCU40.IN1(E)**
	Shaft Direction:	**POSIF0.OUT1**	connected to	**CCU40.IN1(F)**
CAPCOM4:	**CCU40**	**i.e. x = 0**	in CCU4**x**CC4y or	CCU4**x**IN**y**(..)
Timer Slice:	**CC41**	i.e. **y = 1**	in CCU4**x**CC4y or	CCU4**x**IN**y**(..)

Input Connections:			CC4yINS.field:
	Quadrature Clock:	**Event 0**	**CC41INS.EV0IS** = 0100B
	Shaft Direction:	**Event 1**	**CC41INS.EV1IS** = 0101B
		(Event 2 not used)	

Considered Active Events:	
External Count Function triggered on falling edge:	**CC4yINS.EV0EM** = 10B
External Up/Down Function counts up on high level:	**CC4yINS.EV1LM** = 0B

Input Functions:	CC4yCMC.field:
External Count Function triggered by Event 0:	**CC41CMC.CNTS** = 01B
External Up/Down Function triggered by Event 1:	**CC41CMC.UDS** = 10B

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

Published by Infineon Technologies AG