XMC4000

XMC4000 Family

POSIF in Triple-Hall Mode

Triple-Hall Commutation Control for BLDC Motors

User Guide

Device Guide

<Revision 1.0>, <201-08-06>

Microcontroller

Edition <201-08-06>

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>, <201-08-06>	

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 Triple-Hall Commutation Control for BLDC Motors ..... Fehler! Textmarke nicht definiert.

2 Triple-Hall Input Pattern 7
2.1 Delayed Hall Input Sampling 7
2.1.1 Shadow Update of Expected Patterns 8
2.2 Verification of Hall Event Input Patterns 8
2.2.1 Hall Event Verification Outputs 8
2.3 Administration (Shadow Transfer) of Pattern Compare Values 9
2.4 Output Pattern Control 9

3 Triple-Hall Output Pattern for BLDC Motor Commutation Control 9
3.1 Output Pattern Control by CAPCOM Slices in Multi-Channel Mode 9
3.2 Output Pattern Control by CAPCOM Slices in External Modulation Mode 10

4 POSIF Interrupt Registers 12
4.1 POSIF interrupts/event requests in Hall Sensor Mode 12
4.2 POSIF Interrupts/Event requests in Multi-Channel Mode 12

5 Getting Started with POSIF in Triple-Hall Mode 13
5.1 POSIF Global Registers 13
5.2 POSIF Setup when using Hall Sensor Mode 13
5.3 POSIF Setup when using Multi-Channel features in Hall Sensor Mode 13
5.4 POSIF Input Selections in Hall Sensor Mode 13
5.5 POSIF Operation Mode Associated Registers 13
5.5.1 POSIF Associated Configurations of CCU4x/-8x Timers 13
5.5.2 POSIF Synchronized Start of Timers by HW Initialization 13
5.5.3 POSIF Optional Setup Selections in Hall Mode 13
5.6 Timer Setup 14
5.6.1 Ultimate Initialization Sequence 14
5.6.2 Ultimate Start-Up Enable 14
5.6.3 Setting the Timer Counting Modes 14
5.6.4 Setting Compare Mode Control 14
5.6.5 Setting the Output Pins PASSIVE / ACTIVE Level Control 14
5.6.6 External Events Control Setup Registers 14
5.6.7 Multi-Channel Mode Enable 14
5.6.8 Multi-Channel Synchronization to Shadow Transfers 15
5.7 Using the POSIF Global Registers for Hall Sensor Mode Setup 15
5.8 Using the POSIF Global and Multi-Channel (MC) Mode Setup Registers 15
5.9 Using the Timer Control Registers for COMPARE Mode 15
5.10 Using the Timer Control Registers for MULTI-CHANNEL Mode 18
5.11 Using the Timer Control Registers for EXTERNAL MODULATION Mode 18
5.11.1 Using the Register Bitfields for EXTERNAL EVENTS CONTROL 18
5.11.2 Using the External Events Control Registers for START 18

List of Figures

Figure 1 BLDC Motor Control with Triple-Hall Commutation 6
Figure 2 The Triple-Hall Input Samples stage for a BLDC Motor Commutation Control 7
Figure 3 Triple-Hall Input Samples Edge Detection 7
Figure 4 Triple-Hall Input Samples Verification and Administration of Expected Samples 8
Figure 5 Triple-Hall BLDC Motor Commutation Control using CAPCOM units in Multi-Channel Mode 10
Figure 6 Triple-Hall BLDC Motor Commutation Control by CAPCOM:s in External Modulation Mode 11

1 Triple-Hall Communication Control for BLDC Motors

The POSIF module 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 POSIF *Hall Sensors Decoder* is one of its main subunits, dedicated for BLDC motor control.

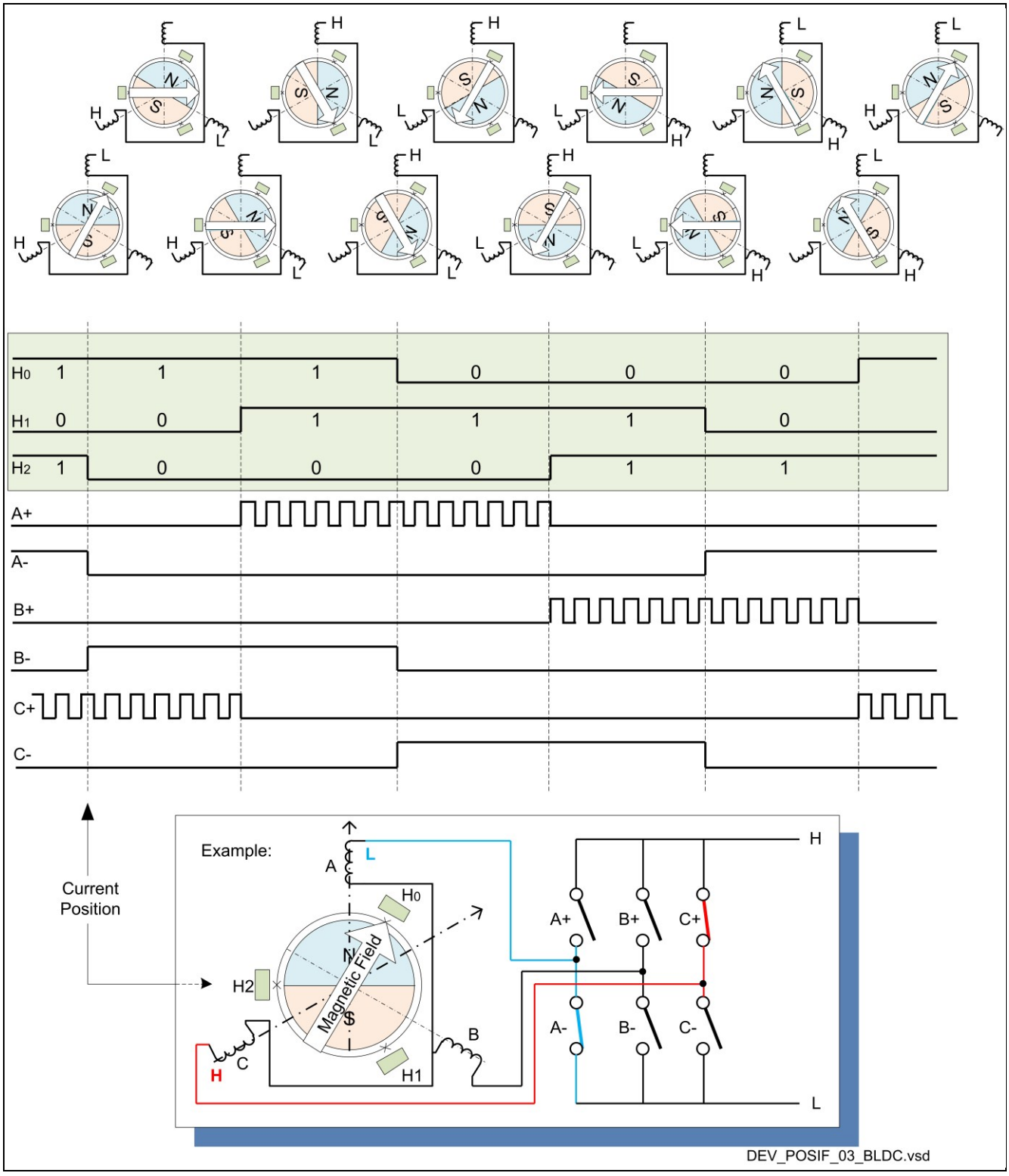

Figure 1 BLDC Motor Control with Triple-Hall Commutation

2 Triple-Hall Input Pattern

The Hall Sensor Decoder mode operates in four succesiev stages: 1) The *Hall Input Samples* stage for the commutation signals, 2) the *Detection & Delay* stage for the inputs sampling after any edge transition, 3) the *Verification & Administration* stage of expected *Hall Patterns* and 4) the *Output Pattern Control* & HW protocol.

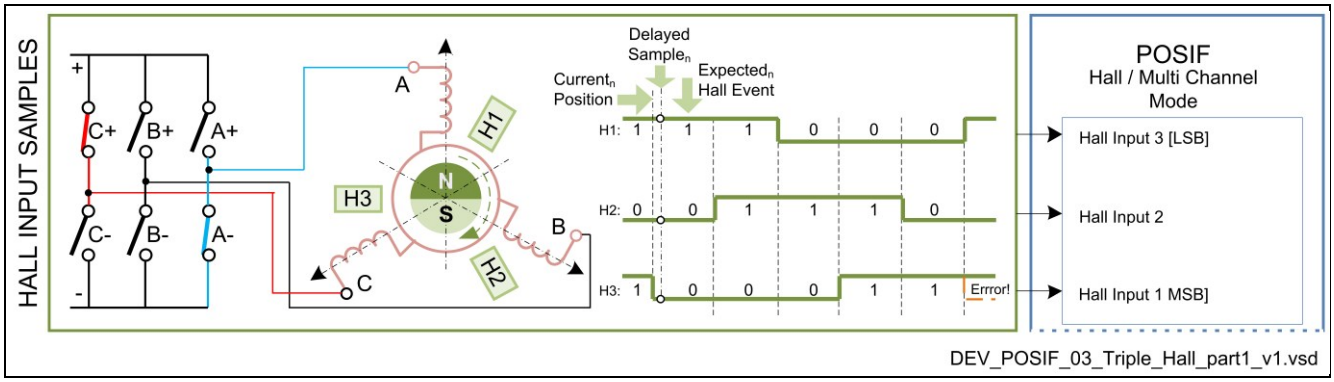

Figure 2 The Triple-Hall Input Samples stage for a BLDC Motor Commutation Control

2.1 Delayed Hall Input Sampling

The sampling of each Hall Input pattern can be delayed a certain time after an input transition edge is detected, in order to reject noise that might appear at these positions. The Input Edge Detection output POSIFx.**OUT0** has to start a CCU4/8 timer that after a certain time triggers the sampling via a selected input: POSIFx.**HSD[B...A]**.

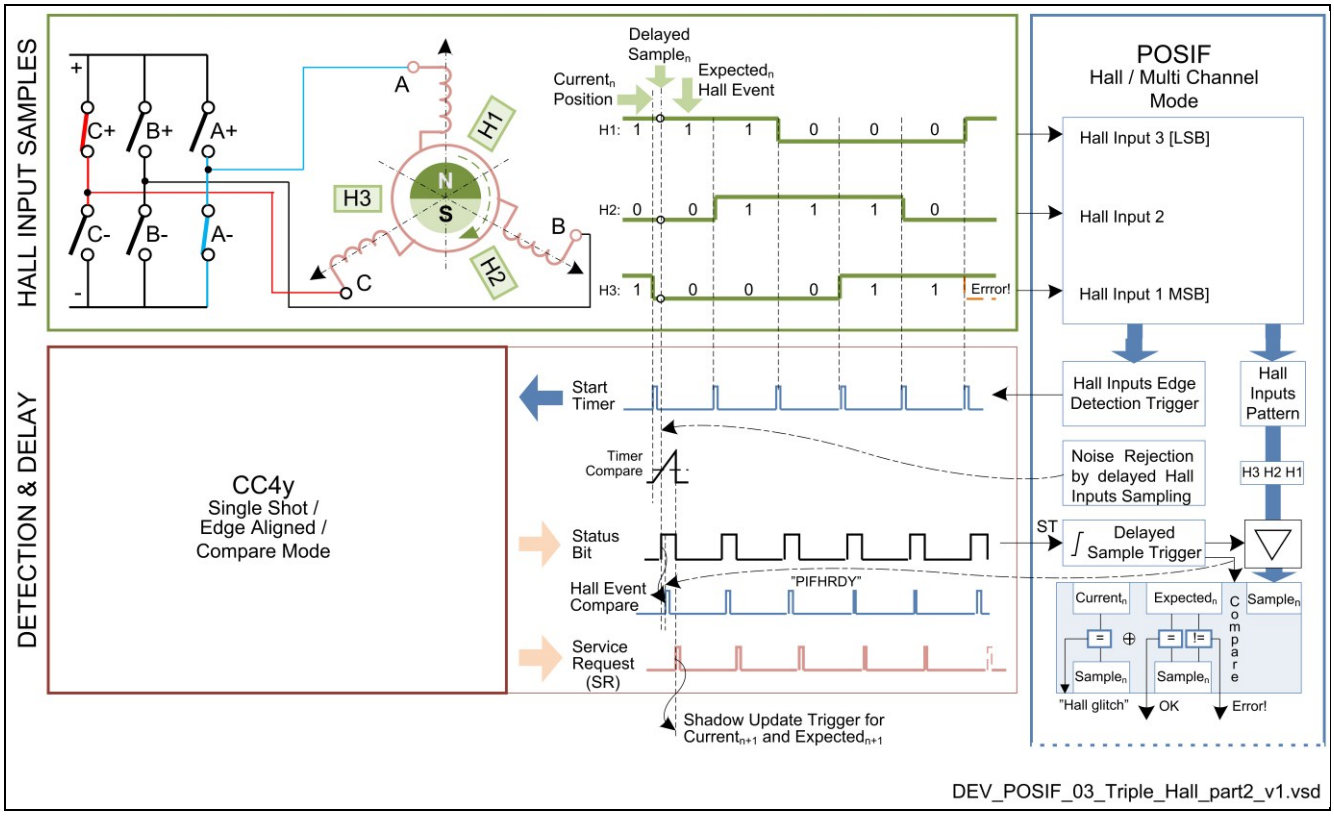

Figure 3 Triple-Hall Input Samples Edge Detection

2.1.1 Shadow Update of Expected Patterns

After each occurrence and sampling of a new pattern (*Sample$_n$*) by the delayed Sample Trigger (*ST*) and an event verification by a Hall Event Compare, there should be a Service Request (*SR*) to Shadow Update the next (*Expected$_{n+1}$*) pattern. An update path could be a DMA transfer, on the falling edge of the Sample Trigger (*ST*).

2.2 Verification of Hall Event Input Patterns

See Figure 4.

Each Hall Input pattern (*Sample$_n$*) is compared to the current pattern (*Current$_n$*) and also to the expected pattern (*Expected$_n$*) - by the PIFHRDY trigger. The comparison results will be interpretated: “Hall *Glitch*” (==*Current$_n$*), “*Correct Hall Event* (CHE)” (==*Expected$_n$*) or “*Wrong Hall Event* (WHE)”, if (*Sample$_n$*) is equal to *none* of those.

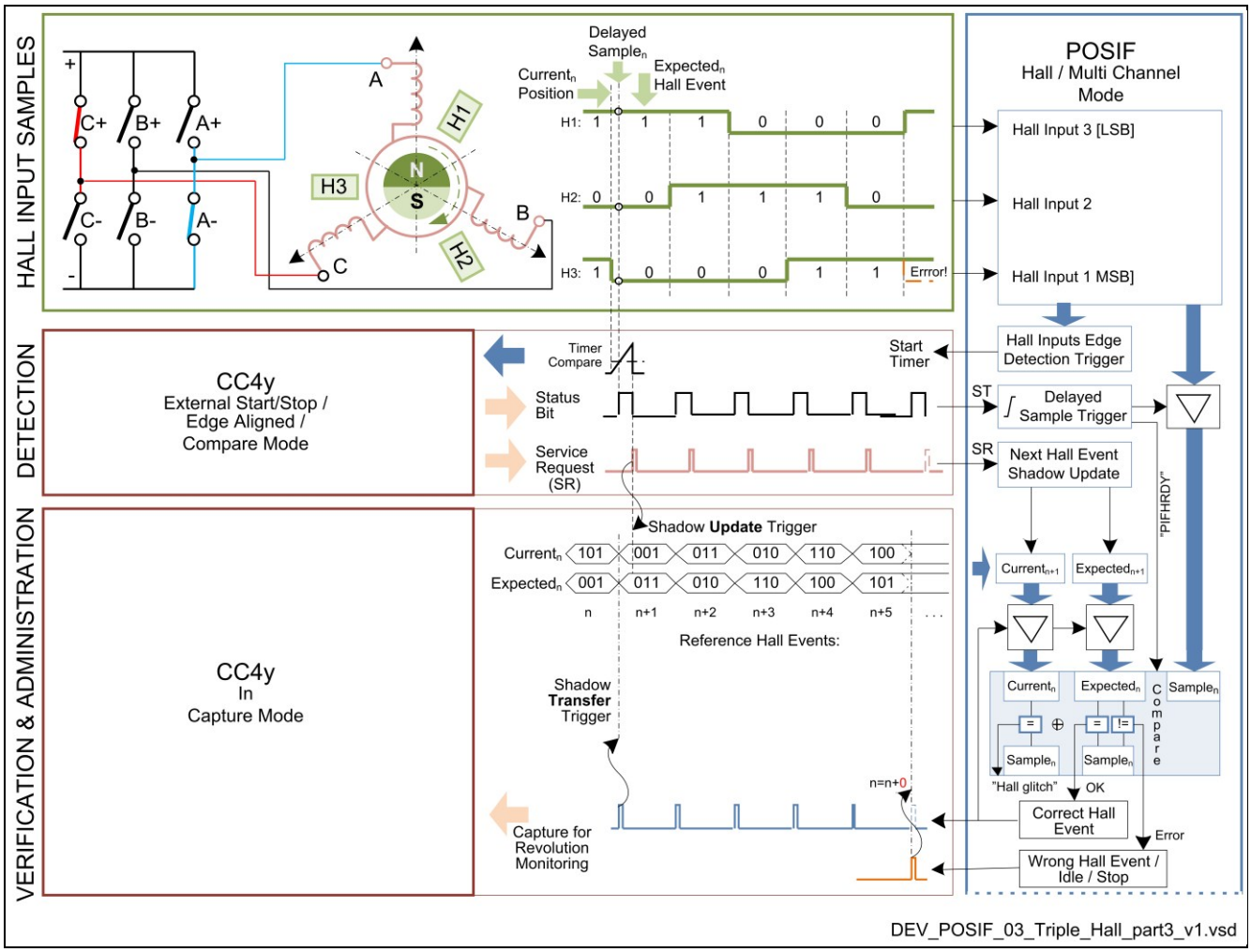

Figure 4 Triple-Hall Input Samples Verification and Administration of Expected Samples

2.2.1 Hall Event Verification Outputs

- *Correct Hall Event* (CHE), linked to POSIFx.**OUT1**, triggers Shadow Transfer of *Current$_{n+1}$* and *Expected$_{n+1}$*.
- *Wrong Hall Event* (WHE), linked to POSIFx.**OUT2**, can be used for IDLE control or STOP Run Bit / PWM.
- *Glitch Hall Event* or *Wrong Hall Event*, both events linked to POSIFx.**OUT3**, will generate a STOP signal.

2.3 Administration (Shadow Transfer) of Pattern Compare Values

On each successful "*Correct Hall Event* (CHE)" there is a *Shadow Transfer* of the *next* pattern compare values: The compare registers "*Current$_n$ Pattern*" (HALP.**HCP**) and "*Expected$_n$ Pattern*" (HALP.**HEP**) are reloaded from the "*Current$_{n+1}$*" / "*Expected$_{n+1}$*" shadow registers (HALP.**HCPS** / HALP.**HEPS**) that are *shadow updated* by SW.

2.4 Output Pattern Control

The Multi-Channel-Mode Pattern register **MCM.MCMP** is linked to the output POSIFx.**MOUT[15:0]**. Its value will be updated from the shadow register **MCSM.MCMPS** by shadow transfer on the Multi-Channel update trigger *PIFMST* - i.e. the *MC Pattern Update Request* POSIFx.**OUT6** & *Pattern Update Sync* POSIFx.**MSYNC[D…A]**. See Figure 5.

3 Triple-Hall Output Pattern for BLDC Motor Commutation Control

The POSIF module in conjunction with CAPCOM units in *multi-channel mode* (or in *external modulation mode*) enables complete synchronicity between the output state update and the application of a new pattern. This can be used in Hall Sensor Mode for the direct drive of brushless DC motors with the wanted motor control patterns.

3.1 Output Pattern Control by CAPCOM Slices in Multi-Channel Mode

See Figure 5.

A **CCU4/8OUT**-pin output level of any slice that has been set in *multichannel mode* (**CC4/8yTC.**MCME = 1_B), can be **&**-*controlled in parallel* by an external level that is present on its Multi-Channel Input **MCI**-pin. To get it all synchronized, a Set Shadow Transfer Enable **MCSS** and a Pattern Set **PSy** request complete the HW protocol.

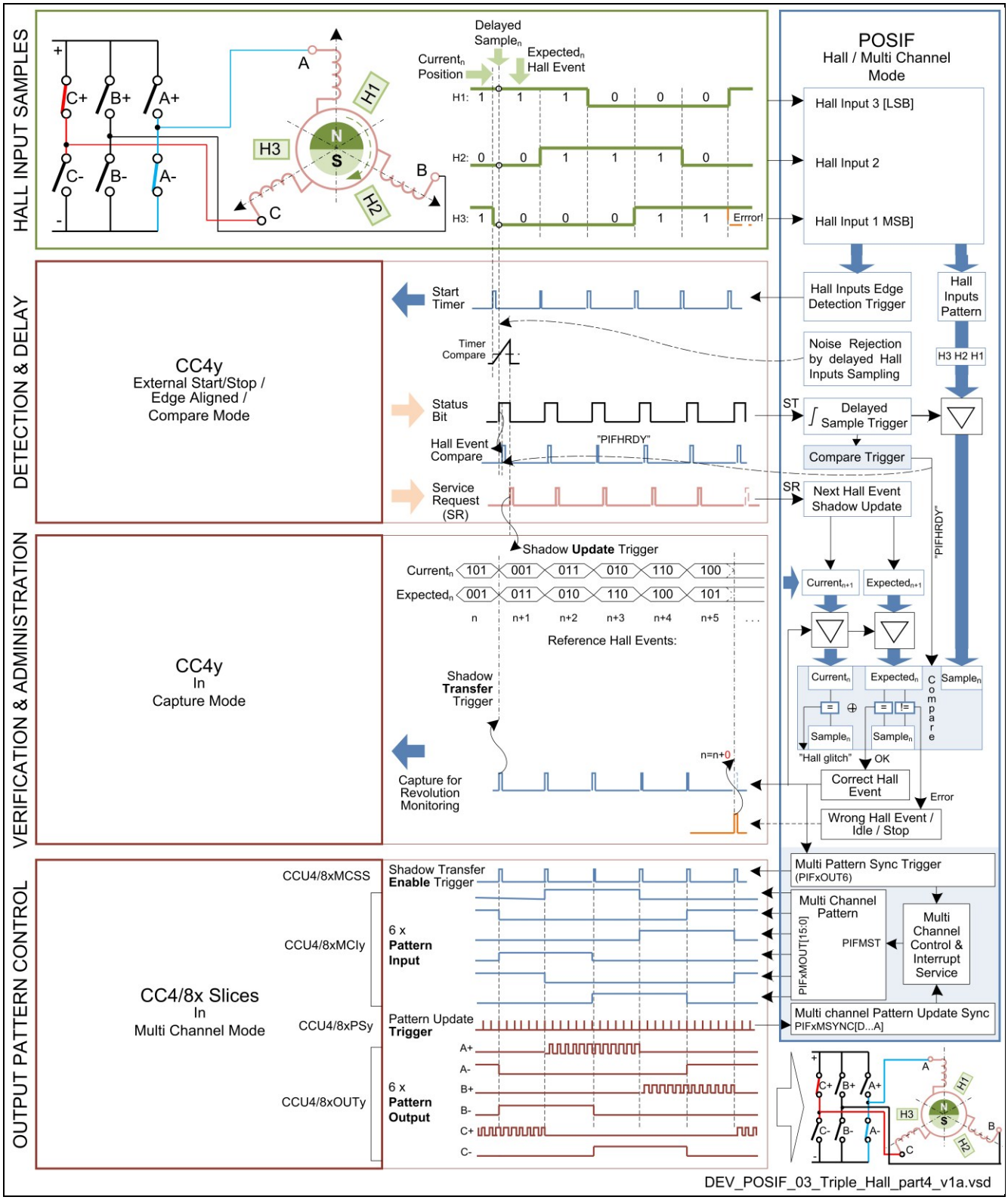

Figure 5 Triple-Hall BLDC Motor Commutation Control using CAPCOM units in Multi-Channel Mode

3.2 Output Pattern Control by CAPCOM Slices in External Modulation Mode

See Figure 6.

A slice may have its output pin modulated by *external events*. This function (and the *active level* of the event source) should be mapped to one of the 3 inputs of the slice in the **CC4/8yCMC** and **CC4/8yINS** registers. Modulation alternatives should be mapped in **CC4/8yTC -** for *Status Bit Clear* or just *Output Gating* on events.

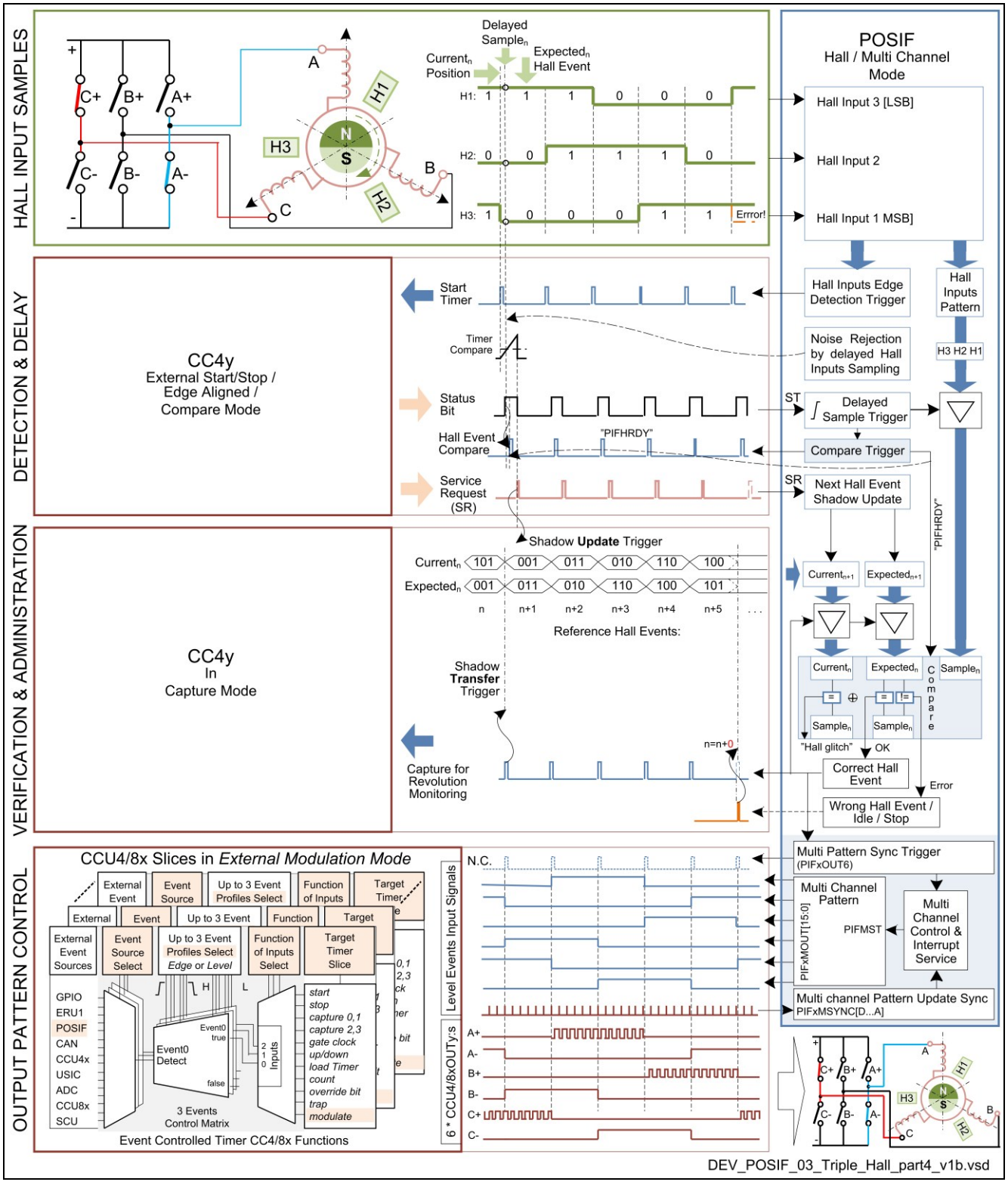

Figure 6 Triple-Hall BLDC Motor Commutation Control by CAPCOM:s in External Modulation Mode

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 Hall Sensor Mode

Four interrupts/event request flags can be enabled by the **PFLGE** register bits: *Transition at the Hall Inputs* (**HIES**), *Occurrence of a correct hall event* (**CHES**), *Occurrence of a wrong hall event* (**WHES**), *Shadow transfer of the Multi-Channel pattern* (**MSTS**) - which is triggered when the Multi-Channel pattern is updated (*PIFMST*).

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 in Triple-Hall Mode

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 Hall Sensor Mode

After activation of a POSIF unit by the system start-up procedures (after hard reset), the POSIF control function register bitfield **PCONF**.FSEL should be written 00_B for *Hall Sensor Mode* enable. The **PCONF**.DSEL bit selects which *Delay Pin* to trigger the Hall inputs sampling. The **PCONF**.SPES bit selects a *rising* or *falling* trigger edge.

5.3 POSIF Setup when using Multi-Channel features in Hall Sensor Mode

The POSIF control function register bitfield **PCONF**.FSEL should be written 00_B for *Hall Sensor Mode* enable – **even if** the *Multi-Channel Mode* subunit features are intended to be used for a *Multi-Channel Pattern Update* in the output commutation control flow. The **PCONF**.FSEL should be written within the system start-up procedure.

5.4 POSIF Input Selections in Hall Sensor Mode

The **PCONF.**INSEL0, **PCONF.**INSEL1 and **PCONF.**INSEL2 bitfields map an input pin out of POSIFx.IN**0**[D...A], POSIFx.IN**1**[D...A]) and POSIFx.IN**2**[D...A] to the *Hall Input 1*, *Hall Input 2* and *Hall Input 3 signals* respectively. Additionally, a *Low-Pass Filters* configuration of 0 - 64 clock cycles can be setup by the **PCONF.**LPC bitfield.

5.5 POSIF Operation Mode Associated Registers

For *Hall Sensor Mode*: use **HALPS/HALP**

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

5.5.1 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***.

5.5.2 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.

5.5.3 POSIF Optional Setup Selections in Hall Mode

It is possible to *Disable* the *IDLE* signal from clearing the Multi-Channel Pattern and the Run Bit, by setting the bit **PCONF**.HIDG =1. (Default =0). It is also possible to use SW instead of HW for Multi-Channel pattern update by setting the bit **PCONF**.MCUE =1. (Default =0).

5.6 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.

5.6.1 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*.

5.6.2 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!

5.6.2.1 Initialize and Start the Timer

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.

5.6.3 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*.

5.6.4 Setting Compare Mode Control

Compare Mode is the default state of a CAPCOM timer, i.e. **CC4yTC.**CMOD=0_B. The *compare level* will be set in the compare register **CC4yCR** by the value from the compare *shadow* registers **CC4yCRS** on *shadow transfers - on request within the current period cycle* by the **Slice Transfer Set Enable** bit **GCSS.**SySE.

5.6.5 Setting the Output Pins PASSIVE / ACTIVE Level Control

The PASSIVE/ACTIVE state of a slice *internal* output (i.e. the status bits **CC4yST** is controlled by the compare level and the External Modulation Mode. Whether the *external* output pin **CCU4xOUTy** should be *Passive Low / Active High* or vice versa (e.g. the PWM), is controlled by the **CC4yPSL.**PSL bit.

5.6.6 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*.

5.6.7 Multi-Channel Mode Enable

A slice has to be set in *multichannel mode* (**CC4yTC.**MCME = 1_B), in order to let its output pattern **CC4yOUT** be *controlled in parallel* by an external pattern present on the Multi-Channel Input **MCIy**. Set the Multi-Channel shadow transfer enable bit **GCTRL.**MSEy and configurate the shadow transfers by the **GCTRL.**MSDE bitfield.

5.6.8 Multi-Channel Synchronization to Shadow Transfers

When it is necessary to perform certain compound shadow transfers synchronously with the update of the Multi-Channel pattern, then the **CCU4xMCSS** trigger can synchronize those scenarios. The synchronization will be performed via the **GCTRL.**MSEy bit field respective the shadow transfers configured on the **GCTRL.**MSDE field.

5.7 Using the POSIF Global Registers for Hall Sensor Mode Setup

The following Hall Sensor Mode Protocol Controlling Global Registers should be setup:

1. **PCONF**.FSEL Select Mode: 00_B for Hall (even if Multi-Channel Mode features will be used)
2. **PCONF**.HIDG *Hall Idle Generation Enable*: Disables a clear of the Multi-Channel pattern and run bit
3. **PCONF**.INSEL0 Select *PhaseA/Hall input* pin by selecting an input pin out of the POSIFx.IN0[D...A] pins
4. **PCONF**.INSEL1 Select *PhaseB/Hall input* pin by selecting an input pin out of the POSIFx.IN1[D...A] pins
5. **PCONF**.INSEL2 Select *Index/Hall input* pin by selecting an input pin out of the POSIFx.IN2[D...A] pins
6. **PCONF**.DSEL Select *Delay input* pin out of the POSIFx.HSD [B...A] pins - to trigger the actual sample
7. **PCONF**.SPES Select *Edge for the sampling trigger* - 0_B for *rising* edge / 1_B for *falling* edge
8. **PCONF**.<bitfields> Map the EWIS, EWIE, EWIL bitfields for handling *Wrong Hall Events* (“*WHE*”)
9. **PCONF**.LPC Select *Low Pass Filters Configuration* on the Hall Sensor Inputs: *0 - 64 clock cycles*

The following Hall Sensor Mode Registers should be considered:

10. **HALP**.HCP Hall *Current* Pattern Updated with <HALPS.HCPS> on correct Hall events
11. **HALP**.HEP Hall *Expected* Pattern Updated with <HALPS.HEPS> on correct Hall events
12. **HALPS**. HCPS Hall *Current Shadow* Pattern Contains the *next Hall Current Pattern* updated by SW
13. **HALPS**. HEPS Hall *Expected Shadow* Pattern Contains the *next Hall expected Pattern* updated by SW

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

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

14. **PCONF**.MCUE Select MC Pattern Update Enable control: “by HW” (default) by 0_B or “by SW” by 1_B
15. **PCONF**.MSETS Select the input POSIFx.**MSET[H...A]** signal that is used to enable a MC pattern update
16. **PCONF**.MSYNS Select the PWM synchronization input POSIFx.**MSYNC[D...A]** for a MC pattern update
17. **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:

18. **MCM**.MCMP MC Mode Pattern Contains *MC Pattern Out* applied to POSIFx.**OUT[15:0]**
19. **MCSM**. MCMPS MC Mode Shadow Pattern This is passed into the **MCM**.MCMP on *Transfer* events
20. **MCMF**.MSS MC Mode **Flag** status *Read Only*: *MC Update Trigger* request bit *F/F* status
21. **MCMS**.<bitfields> MC Mode Control **Set** Perform ***Immidiate*** *Pattern Update* - or just set *Enable*
22. **MCMC**.<bitfields> MC Mode Control **Clear** Clears the *MC Pattern Out*, the **MCMF**.MSS bit - or both

5.9 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:

23. **CC4yTC.**CMOD CAPCOM Mode selection bit may be used: 0 → *Compare* / 1 → *Capture* mode
24. **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!

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

Options:

27. **CC4yPSL.**PSL *PASSIVE/ACTIVE* control bit: (*CCU4xOUTy)* = (*CCUxSTy*) xor (*CC4yPSL.PSL*)
28. **CC4yTC.**STRM *Extended Start* on *External Start*. 0_B: *Start (=Set Run Bit)* / 1_B: *Flush* & *Start* Timer
29. **CC4yTC.**ENDM *Extended Stop* on *Ext. Stop*. 00_B: *Stop* / 01_B: *Flush* / 10_B: *Stop* & *Flush* /11_B:reserved

30. **CC4yTIMER** Initialize a timer start value (or *flush* by the **CC4yTCCLR**.TCC bit=1 command)
31. **CC4yTCSET**.TRBS This is the Start Timer by SW command.

5.10 Using the Timer Control Registers for MULTI-CHANNEL Mode

1. **GCTRL.**MSDE **Configurate** transfer for *Compare* / *Compare+Period+Prescaler* / + add even *Dither*

```
for(y=0;y<=3;y++) {
```

2. **CC4yTC.**MCME **Enable** *Multi-Channel Mode* (or not = default)
3. **GCTRL.**MSE**y** **Enable** Shadow Transfer Request

4. **CC4yTC.**CMOD **CAPCOM Mode** selection bit should be =0 for *Compare Mode* (default)
5. **CC4yTC.**TCM Set Timer **Counting Mode** selection bit: 0 → *Edge-Aligned* / 1 → *Center-Aligned*
6. **CC4yCRS.**CRS Write the wanted **Compare Value** into the **CRS** compare *shadow* register
7. **CC4yPR.**PR Write the timer count cycle upper limit, **Period Match**, into the *period* register
8. **CC4yPSL.**PSL Set **PASSIVE/ACTIVE** by **0:** *CCU4xOUTy Passive Low*/ **1:** *CCU4xOUTy Active Low*

Optionally:
9. **CC4yTIMER** Initialize a timer start value (or *flush* by the **CC4yTCCLR**.TCC bit=1 command)
10. **CC4yTC.**CLST Enable a shadow transfer each time a clearing action is done by SW or External Event
11. **GCSS.**SySE Set request at least once for a Shadow Transfer to *Initialize the Compare Registers*

}

12. **Global Start** Map start control to **CCUCON** and *External Events Control* to **CC4yINS** & **CC4yCMC**

5.11 Using the Timer Control Registers for EXTERNAL MODULATION Mode

(See *Using the External Events Control* below.)

```
for(y=0;y<=3;y++) {
```

1. **CC4yCMC.**MOS 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 Level Mode EVxLM bitfields
3. **CC4yTC**.EMT Timer Control bit for Status Bit Clear or Output Gating Only
4. **CC4yTC**.EMS Timer Control bit for Unsynchronized or Synchronized modulation

}

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 MOS Input Function is always x == <MOS>-1 (when <MOS> != 0).

Note: By "Synchronization" should be understood that the Modulated Signal keeps its pulse width when active (i.e. there is no shortening or cut of any active pulses by the Modulation Signal edges).

5.11.1 Using the Register Bitfields for EXTERNAL EVENTS CONTROL

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

5.11.2 Using the External Events Control Registers for START

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, Extended stop, etc

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

Published by Infineon Technologies AG