XMC4000

XMC4500 Family

CCU4 and Single Shot Mode

Event Trigger Delay by Single Shot

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 Event Trigger Delay by Single Shot 5
1.1 Use Cases 5
1.2 Timer Setup in Single Shot Mode 5

2 Using Timer Single Shot Delay for Noise Rejection 6

3 Timer-Start in Single Shot Mode by External Event Control 7
3.1 Timer Inputs for Start or Stop Facilities 7
3.2 External Control by Single-Shot Events 7
3.3 Top-Level Control of Event Requests to/from a Timer in Single-Shot Mode 7

4 Getting Started with CCU4 and Single Shot Mode 8
4.1 Ultimate Initialization Sequence 8
4.1.1 Ultimate Start-Up Enable 8
4.2 Setting Compare Mode Control 8
4.2.1 Setting the Output Pin PASSIVE / ACTIVE Level Control 8
4.2.2 Initialize and Start the Timer 8
4.3 Using the Compare Mode / PWM Control Registers 8
4.4 External Events Control Setup Registers 8
4.4.1 Using the External Start Control Register Bitfields 8
4.4.2 Using the External Stop Control Register Bitfields 9
4.4.3 Pseudo Code for the External Start/Stop Control Setup 9

List of Figures

Figure 1 Basic Timer Slice for Single Shot 5
Figure 2 Timer in Single Shot Mode for Edge Aligned or Center Aligned Mode respectively 6
Figure 3 Using Timer Single Shot Delay for Noise Rejection in Shunt-Current Measurement 6
Figure 5 Single Shot Triggering on External Events 7

1 Event Trigger Delay by Single Shot

Each CAPCOM4 (CCU40 to -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 software low level drivers. The main function blocks of 1 of 4 timer slices on a CCU4x is shown here. See Figure 1.

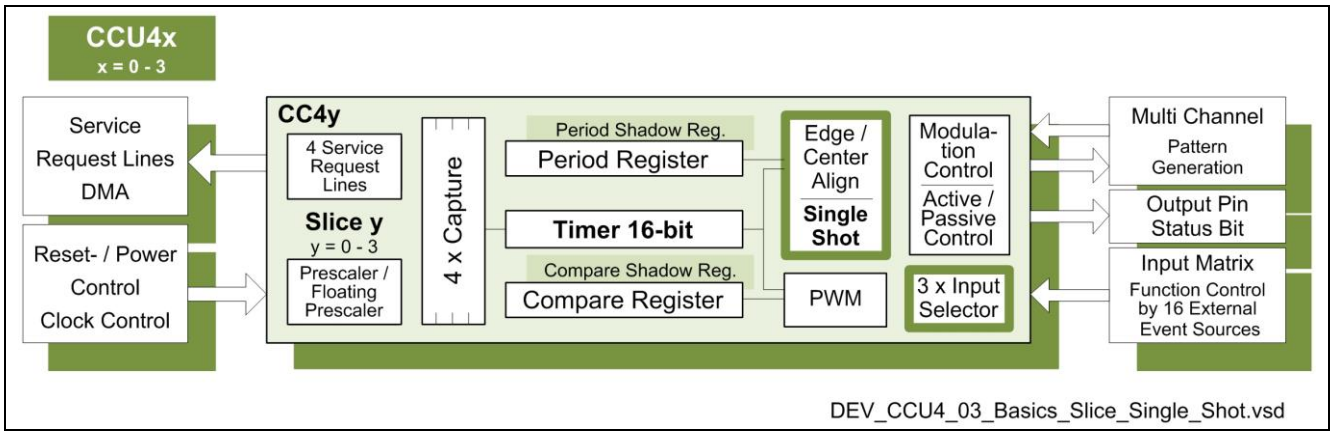

Figure 1 Basic Timer Slice for Single Shot

1.1 Use Cases

A timer in Single Shot mode has a given role for applications where a certain delay has to be invoked between trigger events and operations that should be triggered, e.g. noise rejection in shunt-current signal measurement. Any timer of a CAPCOM4 may be setup in in this mode and cooperate with other timers, ADC or other modules.

1.2 Timer Setup in Single Shot Mode

A slice can be set into Timer Single Shot Mode (TSSM). Both the Timer and its Run Bit (TRB) is cleared by a Timer Period End that occurs next to when the TSSM bit was set - and thus the timer is stopped. A time frame, e.g. a *Single Shot Delay*, is set by the timer start conditions, selected counting mode and the period (PR) value.

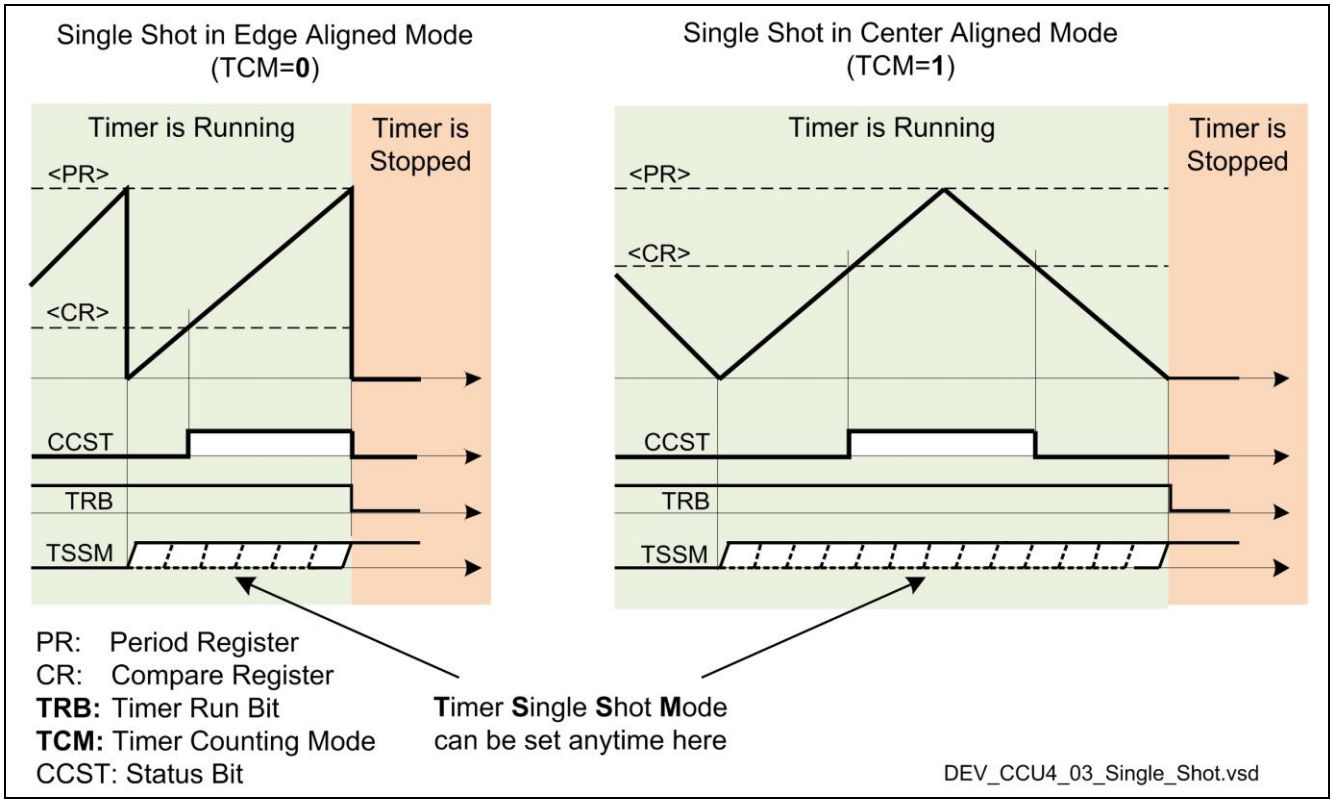

Figure 2 Timer in Single Shot Mode for Edge Aligned or Center Aligned Mode respectively

2 Using Timer Single Shot Delay for Noise Rejection

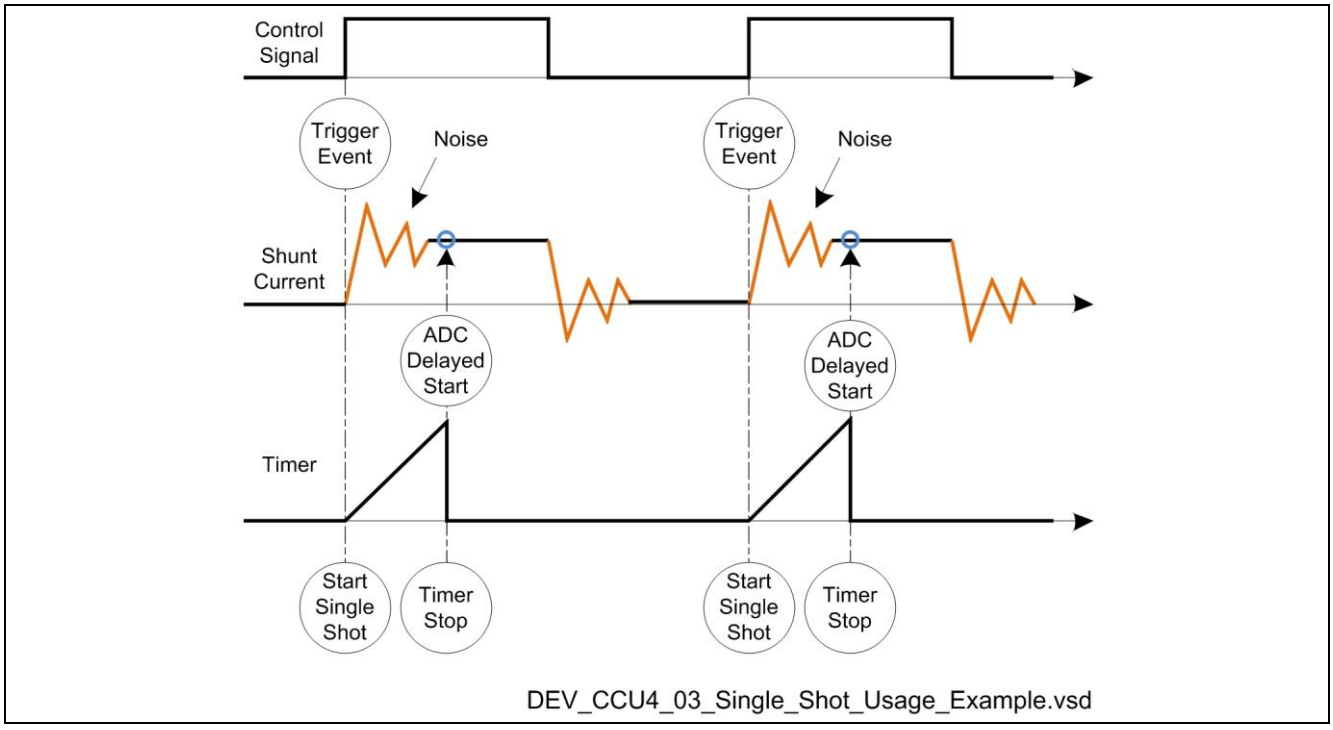

Figure 3 Using Timer Single Shot Delay for Noise Rejection in Shunt-Current Measurement

3 Timer-Start in Single Shot Mode by External Event Control

A *timer-start* in single shot mode can be linked to external triggers – from such as the following sources: *GPIO*, *ERU*, *POSIF*, *CAN*, *CCU4x*, *USIC*, *ADC*, *CCU8x or SCU*. **Pin Connections** are given by the **Top-Level Interconnect** matrix and the **CC4yINS**[P:A] *Input Select* vector - and *Function Select* by the **CC4yCMC** register.

3.1 Timer Inputs for Start or Stop Facilities

A timer has 3 selectable function inputs with configurable *source-event-condition* profiles that are available for **External Events Control** by 16 external sources each, such as for *Timer Start* or *Stop*. The *Extended functions* such as *Flush/Start*, *Flush/Stop* or *Flush-only* may be added too in Single-Shot mode - by the **CC4yTC** register.

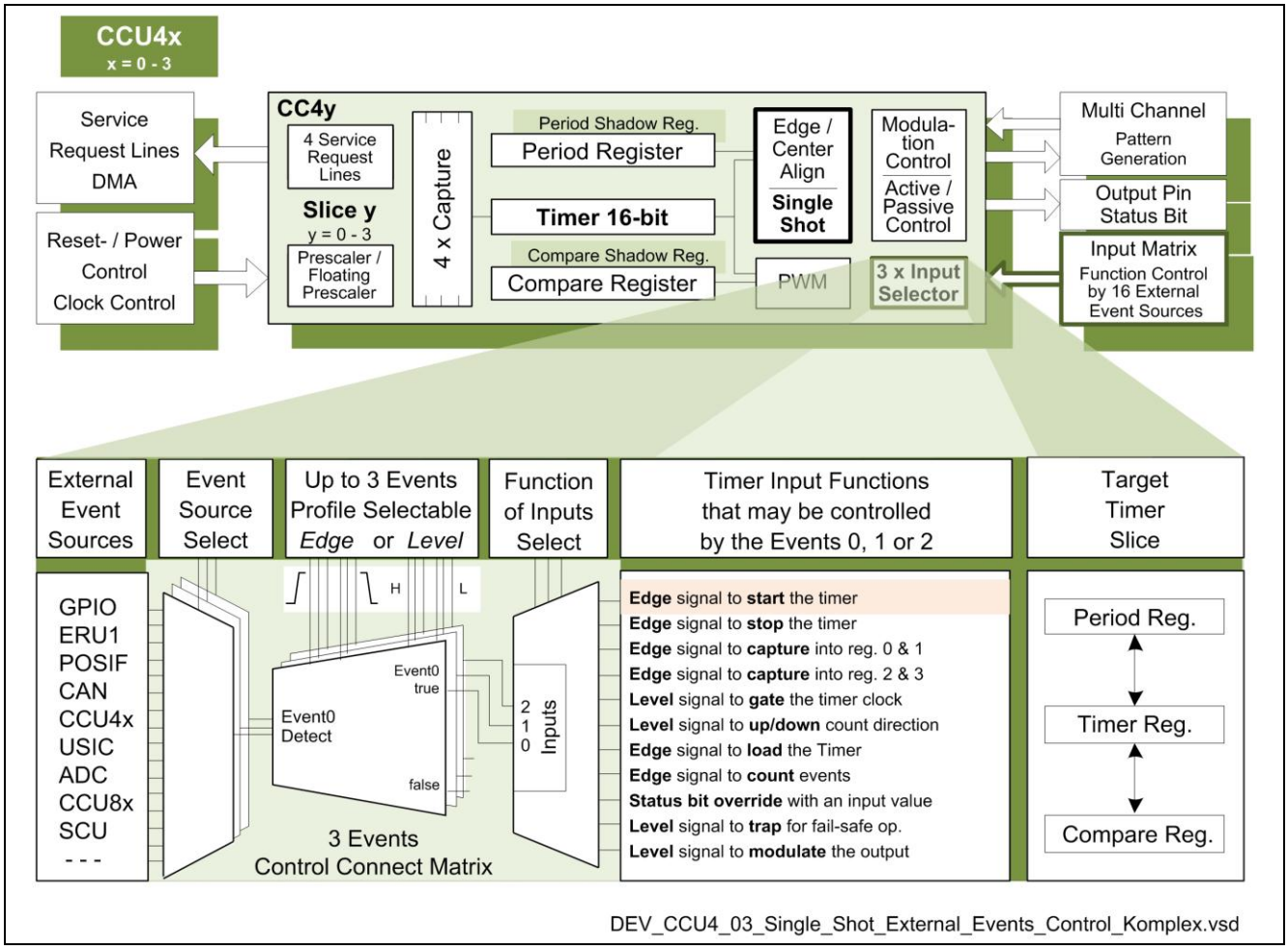

Figure 4 Single Shot Triggering on External Events

3.2 External Control by Single-Shot Events

Single-Shot events may *trigger external actions* via the **Top-Level Interconnect** matrix or *request for interrupts*. Each CAPCOM4 has four **Service Request Lines** and each slice has a dedicated output signal **CC4ySR[3...0]** selectable to a line by **CC4ySRS.** So, single-shots may e.g. act as delayed trigger for ADC actions or interrupts.

3.3 Top-Level Control of Event Requests to/from a Timer in Single-Shot 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. ADC triggering limited by time windows.

4 Getting Started with CCU4 and Single Shot Mode

4.1 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*.

4.1.1 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!

4.2 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 read-only compare register **CC4yCR** by the value from the compare *shadow* register **CC4yCRS** on each *shadow transfer, if requested within the current period cycle* by the **Slice Transfer Set Enable** bit **GCSS.**SySE.

4.2.1 Setting the Output Pin PASSIVE / ACTIVE Level Control

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

4.2.2 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.

4.3 Using the Compare Mode / PWM Control Registers

1. **CC4yTC.**TSSM — Set **Timer Single Shot Mode** =1. (Default=0)

2. **CC4yTC.**CMOD — CAPCOM Mode selection bit should be set =0 (which is the default value)
3. **CC4yTC.**TCM — Timer **Counting Mode selection** bit: 0 → *Edge-Aligned* / 1 → *Center-Aligned*
4. **CC4yCRS.**CRS — Write the wanted **Compare Level** value into the compare *shadow* register
5. **CC4yPR.**PR — Write the timer count cycle upper limit, **Period Match**, into the *period* register

Options:

6. **CC4yPSL.**PSL — **PASSIVE/ACTIVE** control bit: (*CCU4xOUTy)* = (*CCUxSTy*) xor (*CC4yPSL.PSL*)
7. **CC4yTIMER** — Initialize a timer start value (or *flush* by the **CC4yTCCLR**.TCC bit=1 command)
8. **GCSS.**SySE — Request at least once a Shadow Transfer to *Initialize the Compare Register*
9. **CC4yTCSET**.TRBS — Start Timer by SW

4.4 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 *Timer Control* register **CC4yTC** enables different *Mode Alternatives*.

4.4.1 Using the External Start Control Register Bitfields

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

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!

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

4.4.2 Using the External Stop Control Register Bitfields

Besides the Timer Single Shot Mode (by the **CC4yTC.**TSSM bit set), a timer may be stopped by External Events Control as well - even by its own events! The following Control Register Bitfields should be used then:

13. **CC4yCMC.**ENDS Input Function Selection and Event Number x=0-2 Input Line Selection bitfield
14. **CC4yINS.**<Event-*x* bitfields> Event-x Source EVxIS (**CC4yINS**[P:A]) and Event Edge Mode EVxEM bitfields
15. **CC4yTC.**<bitfields> Timer Mode selections for e.g. *Stop, Extended Stop*

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

4.4.3 Pseudo Code for the External Start/Stop Control 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 a MASK OPERATION first for bitfield focusing – if not first time – else:
CC4yCMC |= STRTS_FUNCTION_AND_EVENT_NUMBER_BITFIELD //Initialize External Start
// step 2
CC4yINS &= STRTS_SOURCE_AND_EVENT_MODE_BITFIELDS_MASK; //Focus on STRTS events
CC4yINS |= STRTS_SOURCE_AND_EVENT_MODE_BITFIELDS_PATTERN; //Paste setup pattern
// step 3
CC4yTC &= STRTS_MODES_AND_EXTENDED_MODE_BITFIELDS_MASK; //Focus on STRTS modes
CC4yTC |= STRTS_MODES_AND_EXTENDED_MODE_BITFIELDS_PATTERN; //Paste setup pattern


// step4
// Make a MASK OPERATION first for bitfield focusing – if not first time – else:
CC4yCMC |= ENDS_FUNCTION_AND_EVENT_NUMBER_BITFIELD //Initialize External Start
// step 5
CC4yINS &= ENDS_SOURCE_AND_EVENT_MODE_BITFIELDS_MASK; //Focus on STRTS events
CC4yINS |= ENDS_SOURCE_AND_EVENT_MODE_BITFIELDS_PATTERN; //Paste setup pattern
// step 6
CC4yTC &= ENDS_MODES_AND_EXTENDED_MODE_BITFIELDS_MASK; //Focus on STRTS modes
CC4yTC |= ENDS_MODES_AND_EXTENDED_MODE_BITFIELDS_PATTERN; //Paste setup pattern
```

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

Published by Infineon Technologies AG