XMC4000

XMC4500 Family

Capture Compare Unit - CCU4

CCU Basics

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 CCU Basics 6
1.1 Basic Timer Functions 7
1.1.1 The Compound CAPCOM4 System CCU40 – CCU43 7
1.2 CCU Use Cases 9
1.3 Typical Use Case Features of the CAPCOM4/8 9

2 Additional CCU Features 10
2.1 Single Shot 10
2.2 Timer Concatenation 10
2.3 Dithering PWM 10
2.4 Dithering Period Time 10
2.5 Floating Prescaler 10
2.6 External Modulation 10
2.7 Output State Override 10
2.8 Multi Channel Control 10
2.8.1 Multi Channel Use Case 11
2.9 Timer Load Function 11
2.10 Trap Function 11
2.11 Asymmetric Compare 11

3 CCU4 Input Control 11
3.1 Synchronized Control of CAPCOM Units on External Events 11
3.2 External Control Basics 11
3.3 External Events Control 11
3.4 External Events Sources 12
3.5 External Events Input Functions 12
3.5.1 Extended Input Functions 12

4 Capture Basics 12

5 CCU4 Output Control 13
5.1 External Control by Timer Events 13
5.2 Top-Level Control of Event Requests to/from a Timer Slice 14

6 Compare Basics 14
6.1.1 CCU4 Shadow Transfers 14
6.1.2 Shadow Transfer of Compare Register values 14
6.2 Asymmetric Compare Events 15
6.3 Shadow Transfers in General - Compound Shadow Transfers 16
6.4 CCU4 Output State and Output Pin PASSIVE / ACTIVE Level Control 16

7 How to Start a Timer 16
7.1 CAPCOM4 Ultimate Initialization Sequence 16
7.2 Ultimate Start-Up Enable 16
7.3 Start Timer Running 16
7.4 Global Start of CAPCOM Units 17
7.5 Global Start of the CAPCOM Units CCU4 and CCU8 17

8 Getting Started with Capture Compare Unit - CCU4 18
8.1 How to Start Timer CC4y (y=0-3) 18
8.1.1 Example 1: Start Timer by SW (by Instruction) 18
8.2 Example 2: Start Timer by HW (by External Event) 18
8.2.1 External Events Control Setup Registers 18
8.2.2 Using the External Start Control Register Bitfields 18
8.2.3 Example 3: Start Timers Synchronized (by Global Command) 18

9 Runtime Handling of Capture Compare Unit – CCU4 19
9.1 Example: Handling Shadow Transfers 19

List of Figures

Figure 1 The Timer Slice Block Diagram....6
Figure 2 Basic Functions of each Timer Slice....7
Figure 3 The four Capture/Compare units CCU40-CCU43 basic system of CAPCOM4....8
Figure 5 Some Features and Use Cases (1 – 9) characterizing an CAPCOM Unit (CCU) Features....9
Figure 8 External Control of Timer Input Functions on Events by any External Units....12
Figure 9 Timer Slice four Capture Registers....13
Figure 10 Basic Capture Mechanism – setup in two possible scheme alternatives....13
Figure 11 Timer Slice Compare Registers and PWM related Blocks....14
Figure 12 Basic Shadow Transfer Mechanism for Compare Register Values....15
Figure 13 Asymmetric Compare by Shadow Transfers on both Period Match and One Match....16
Figure 15 External Event Control with Global Start Command....17
Figure 16 Compound Shadow Transfer Mechanism with Coherent Update of PWM....19

1 CCU Basics

A CAPCOM Unit (CCU) is a multi purpose timer unit for many use cases. It is designed with repetitive structures with multiple timer slices, which have all the same base functionalities. This allows the development of re-usable software low level drivers. The specific features can be combined among each other and even with other peripherals such as ADC. With this it is possible to generate any kind of pulses, trigger events or time captures without CPU interaction. The main function blocks of 1 of the 4 equal timer slices on a CCU4x is shown here.

See Figure 1.

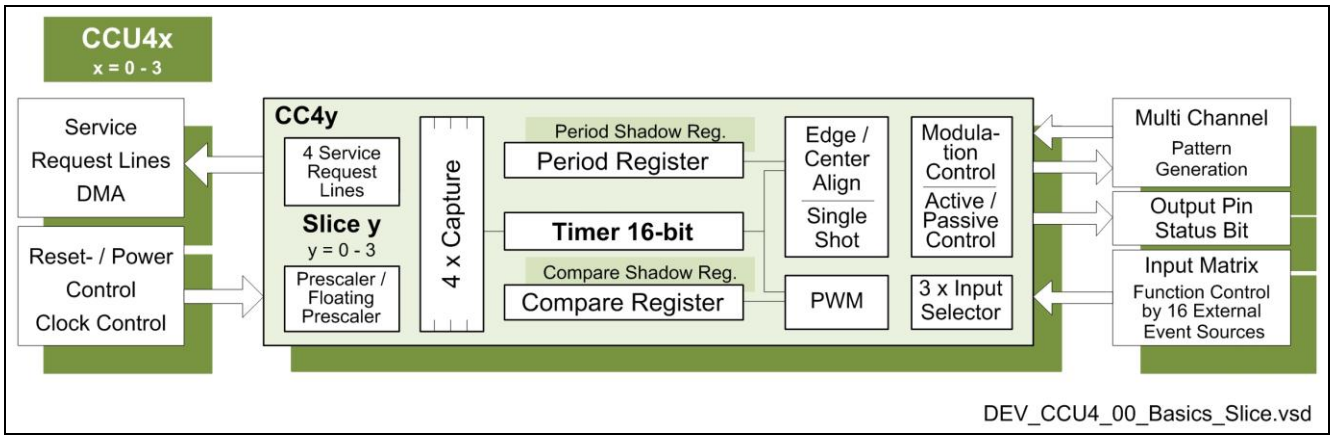

Figure 1 The Timer Slice Block Diagram

1.1 Basic Timer Functions

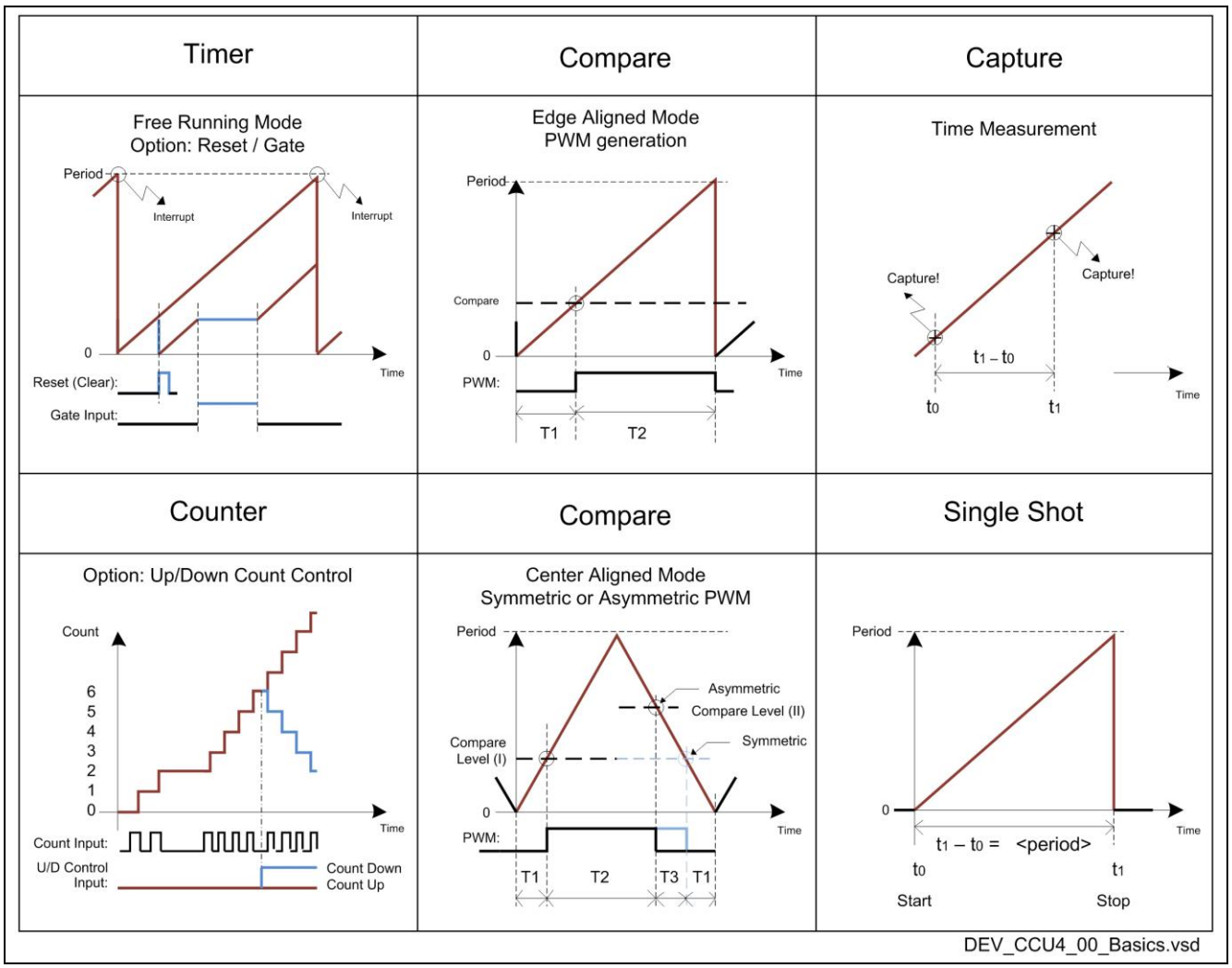

Figure 2 Basic Functions of each Timer Slice

Each timer slice can handle all basic modes and typical options as illustrated here independently of another. See Figure 2.

1.1.1 The Compound CAPCOM4 System CCU40 – CCU43

Each CCU4x (x=3-0), has four 16-bit timer slices CC4y (y=3-0), concatenatable up to 64-bit. A slice has 1 *Timer*, 4 *Capture*-, 1 *Period*- and 1 *Compare* register. The latter two have *shadow registers* for reload. Slices may work independently in different modes - but may also be synchronized, even to other CCU4x slices. They perform multi channel/multi phase pattern generation by update in parallel, on global request flag patterns. See Figure 3.

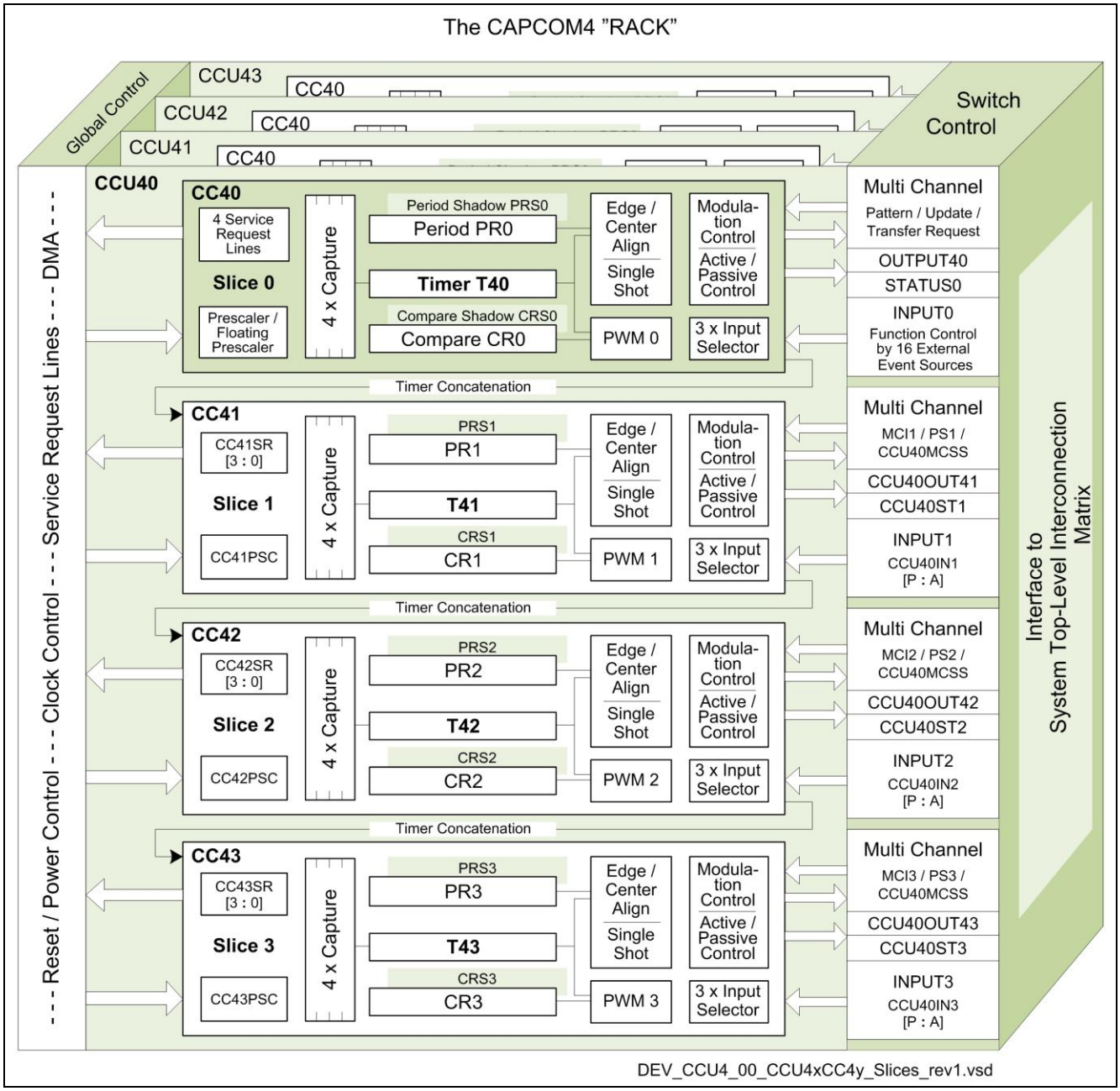

Figure 3 The four Capture/Compare units CCU40-CCU43 basic system of CAPCOM4

1.2 CCU Use Cases

The various capabilities of the CAPCOM timer slices are illustrated here by typical use cases, shortly described. See Figure 4.

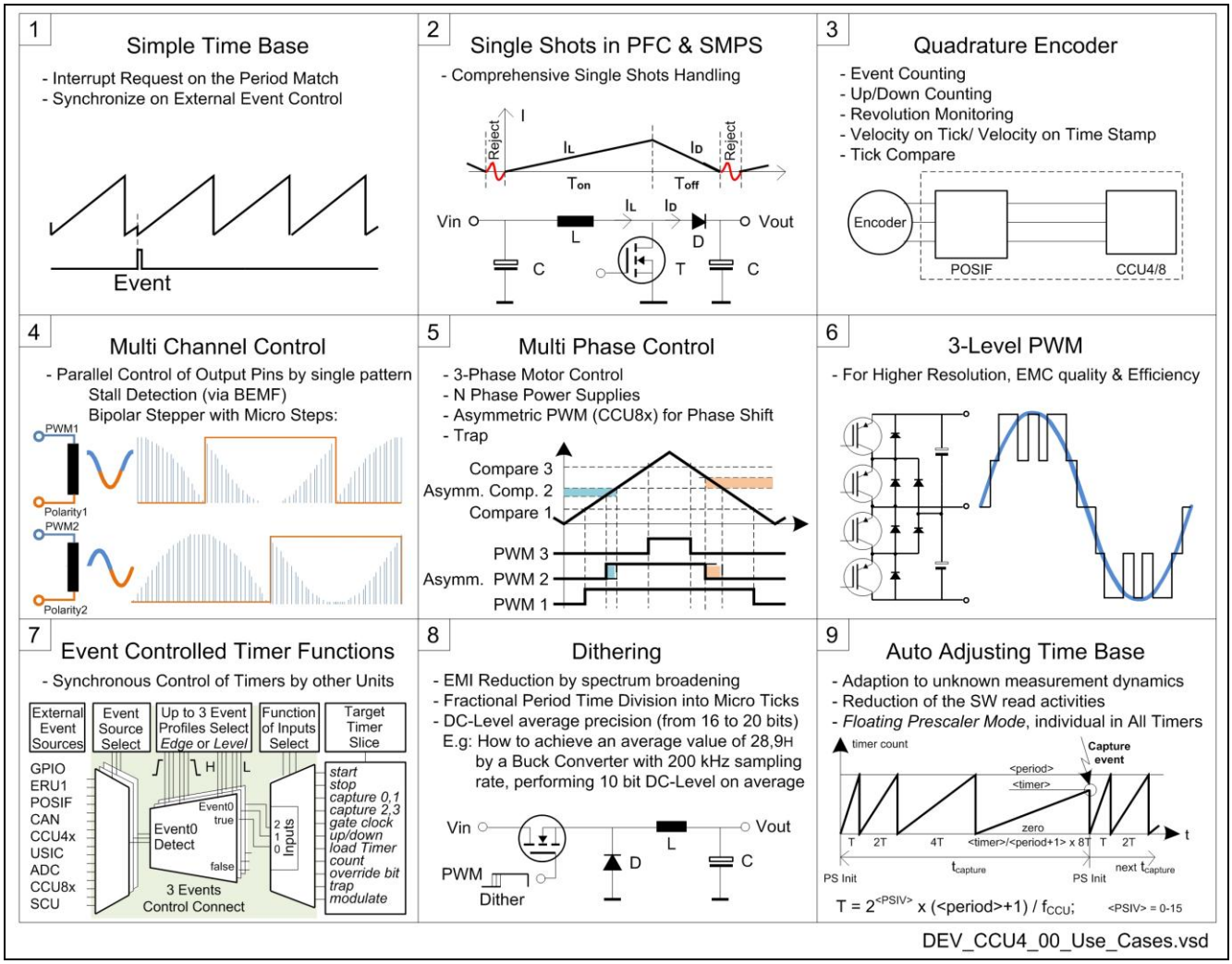

Figure 5 Some Features and Use Cases (1 – 9) characterizing an CAPCOM Unit (CCU) Features

1.3 Typical Use Case Features of the CAPCOM4/8

1. **Simple Time Base** with synchronization option by external events control
2. **Power Conversion** Systems (PFC, SMPS) using Single Shot Mode
3. **Event Monitoring** and Revolution by Capture, Count and Position Interface facilities (POSIF)
4. **Multi Signal Pattern** on Output Pins, created by parallel Multi Channel Control
5. **Drive & Motor Control** with Multi Phase Systems, Phase Adjustment and Trap Handling
6. **3-Level PWM** for Inverters and Direct Torque Control (DTC) of AC Motors and High Precision Sync. Motors
7. **External Events Control** of Timer Input Functions by requests from external system units in interaction
8. **Dithering PWM or Period** for DC-Level precision, Reduced EMI, Fractional Split of Periods into Micro Steps

9. **Auto Adjusting Time Base** by Floating Prescaler for adaption of time measurement to unknown dynamics

2 Additional CCU Features

2.1 Single Shot

A slice may be set in Timer Single Shot Mode (TSSM). Both the Timer and its Run Bit (TRB) is cleared by the Period/One match that occurs next to when the TSSM bit was set - and thus the timer is stopped. A time frame, e.g. a certain delay, is determined by the timer preset conditions and may start on any internal or external event.

2.2 Timer Concatenation

Any timer may be concatenated with an adjacent slice timer as easy as just by setting the respective TCE bits. Thus you may get one or two 32-bit timers, one of 2 possible 48-bit timers or a whole 64-bit timer this way. The first slice's configuration is propagated throughout the rest and so a complete synchronization from LSB to MSB.

2.3 Dithering PWM

Dithering is an HW option in all the CAPCOM4 timers and may leverage PWM DC-level precision upto 16 times. This special HW part may create a kind of superposed jitter upon a timer PWM output signal – to such an extent that the affected PWM pattern DC-level (V_{out}) will converge to an average value with the targeted precision level.

2.4 Dithering Period Time

Micro Ticks can be used in the interpolation between e.g. the sensor pulses from a cog wheel, to achieve position monitoring with higher precision. However, since the divisions of the sensor pulse period intervals seldom become exact multiples of micro tick periods, dithering can optimize the distribution of ticks over time.

2.5 Floating Prescaler

By successive halving of the timer clock frequency periodically, as long there is no capture (or compare) event, the dynamic range may be autonomously adapted to any time length, by using the slice's floating prescaler. In compare mode an arbitrary count range can be setup by using the floating prescaler compare value register too.

2.6 External Modulation

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 **CC4yCMC** and **CC4yINS** registers. Modulation alternatives can be selected in **CC4yTC** for either *Status Bit Clear* or just *Output Gating* on events.

2.7 Output State Override

An input signal value (or its complement value) from *one* external source may override a slice's status bit value on an edge event by *another* external source. Both input sources are selected by the **CC4yINS** register - and the specific function by the matrix control register **CC4yCMC**. Use cases are output timing control by monitoring.

2.8 Multi Channel Control

The output pin state of slices that are set in *multi channel mode* (by the **CC4yTC.MCME** bit) can be controlled in parallel by a single pattern, according to the connected external signal on each **CCU4xMCIy** input. This task is requested by the **CCU4xMCSS** input and synchronized with the update of the status bit and shadow transfers.

2.8.1 Multi Channel Use Case

The Position Interface module POSIF in conjunction with a CCU4x unit in multi channel mode enables complete synchronicity between the output state update and the application of a new pattern. With POSIF in Hall Sensor Mode, this can be used for direct control of brushless DC motors, to output the wanted motor control pattern.

2.9 Timer Load Function

An external source may trigger a Timer Reload transfer from *either* the compare register (if counting up) *or* from the Period register (if counting down). The count direction bit **CC4yTCST.**CDIR can be set by SW or by *another* external source. The *input source(s)* and *function(s)* should be mapped to the **CC4yINS** & **CC4yCMC** registers.

2.10 Trap Function

The trap function forces a PWM output into a predefined state, preset in the active/passive bit PSL. A power device can be switched off safely this way. The input selection, trap and output functionality should be mapped to the slice's **CC4yINS**, **CC4yCMC** and **CC4yPSL** registers. Outputs are Trap-Enabled by the **CC4yTC** register.

2.11 Asymmetric Compare

Shadow transfers on both Period and One Match in center aligned mode enable asymmetric compare events. (However, CAPCOM8 slices offer *two* compare registers per timer plus *two* corresponding shadow registers – which circumstances simplify for example applications that require PWM Phase Shift Control facilities etc.)

3 CCU4 Input Control

3.1 Synchronized Control of CAPCOM Units on External Events

The manner of External Events Control distribution to CCUs (i.e. including CCU8 as well) opens for advanced applications with synchronized timer control - e.g. in Drive & Power Control, such as 3-Level Inverters requiering 12 synchronized PWM. Just the fantacy may limit the realizable topography or timing pattern complexity range.

3.2 External Control Basics

A slice may have its input functions controlled by *external* sources. The *external source(s)*, *active mode(s)* and *input function(s)* should be mapped to the 3 inputs of the slice in the **CC4yINS** and **CC4yCMC** registers. Function mode extending alternatives can be added by selections in the **CC4yTC** Timer Slice Control register.

3.3 External Events Control

An External Event Control request is a *considered edge* or *level event* signal from a peripheral unit or a GPIO. It can be linked to CCU4xCC4y slices input selection stages via a comprehensive matrix. A slice that by any of its 3 events setups detects a *considered source-event-input profile* can be function controlled "remotely" this way!

See Figure 5.

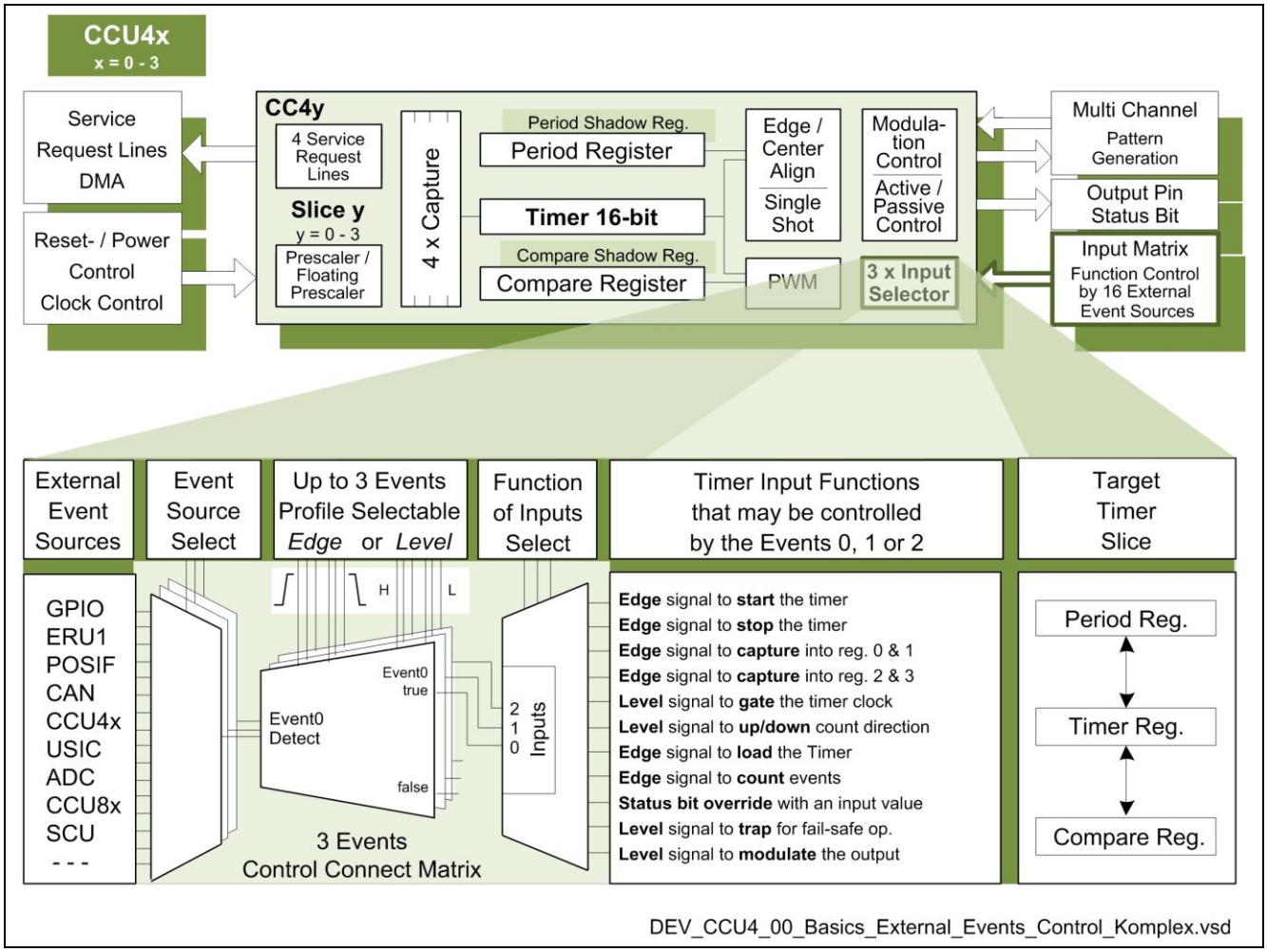

Figure 6 External Control of Timer Input Functions on Events by any External Units

3.4 External Events Sources

CCU4xCC4y Input Functions can be linked to external trigger requests – 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.

3.5 External Events Input Functions

There are 11 Timer Input Functions (as e.g. *Start the Timer*), controlable by external events via 3 selectable input lines with configurable *source-event profile* conditions to the Timer Slices CC4y (y=0-3) of a CCU4x unit (x=0-3) for *Start, Stop, Capture0-3, Gate, Up/Down, Load, Count, Bit Override, Trap & Modulate Output* control.

3.5.1 Extended Input Functions

There are some Extended Input Functions too, in the register **CC4yTC** for *Extended Start,-Stop* with *Flush/Start, Flush/Stop* or *Flush* – or *Extended Capture Mode* as well, which together with a read access register (**ECRD**) simplifies administration of capture registers and full-flags, when more than one slice is used in Capture mode.

4 Capture Basics

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.

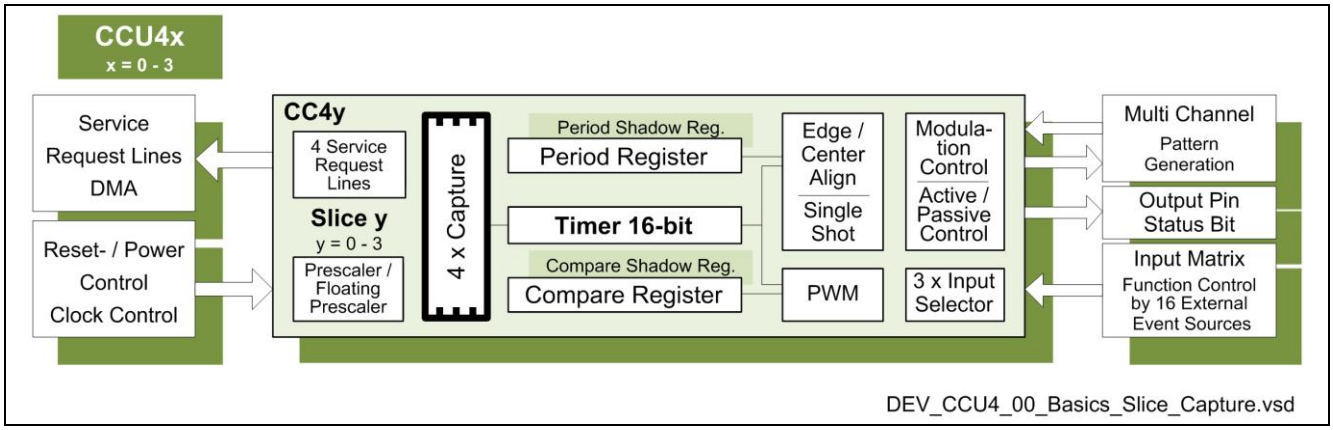

Figure 7 Timer Slice four Capture Registers

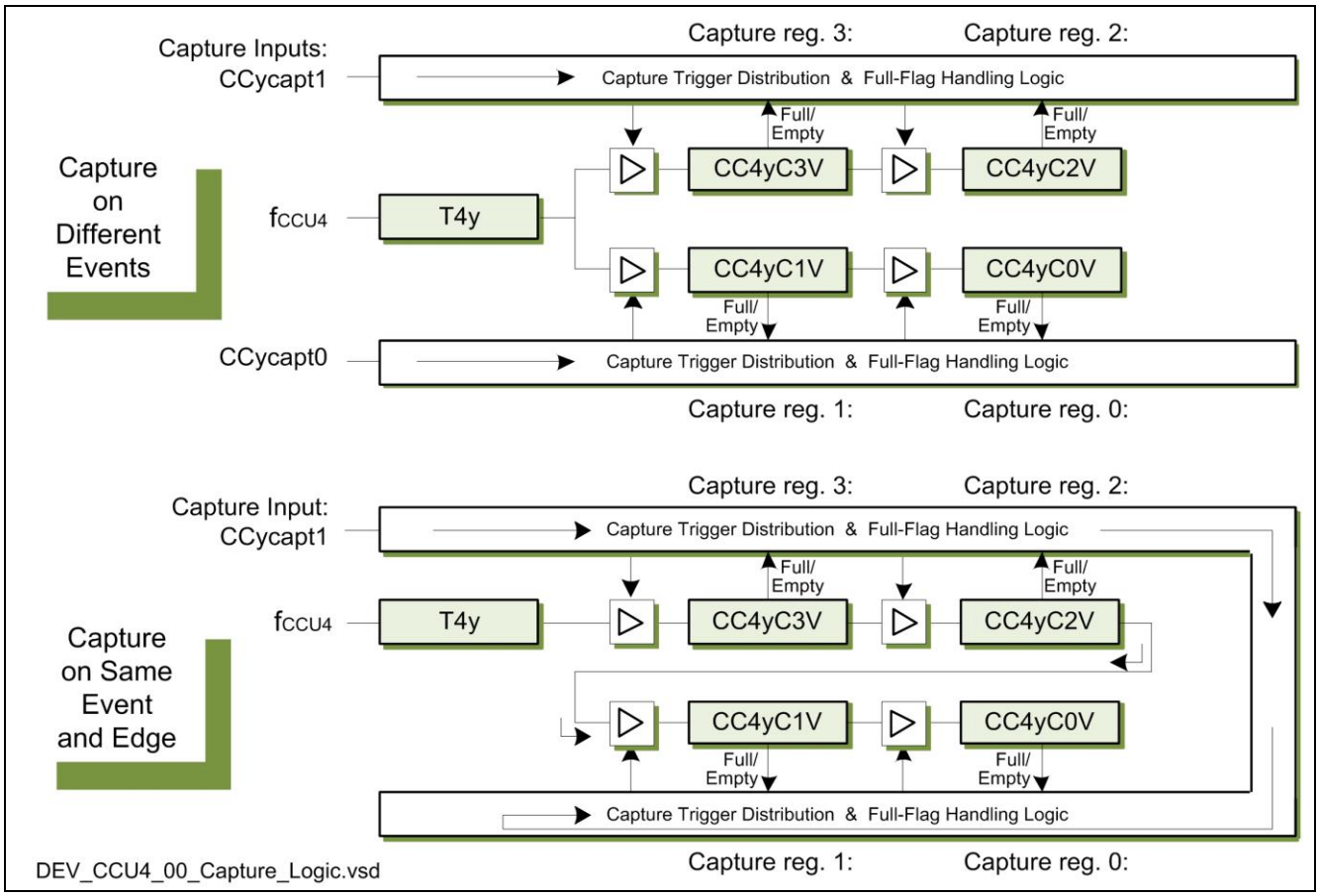

Figure 8 Basic Capture Mechanism – setup in two possible scheme alternatives

5 CCU4 Output Control

5.1 External Control by Timer Events

A timer 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.** I.e. timer slice events may request for direct peripheral actions or an interrupt.

5.2 Top-Level Control of Event Requests to/from a Timer Slice

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. invoke I/O states, Time Windowing etc.

6 Compare Basics

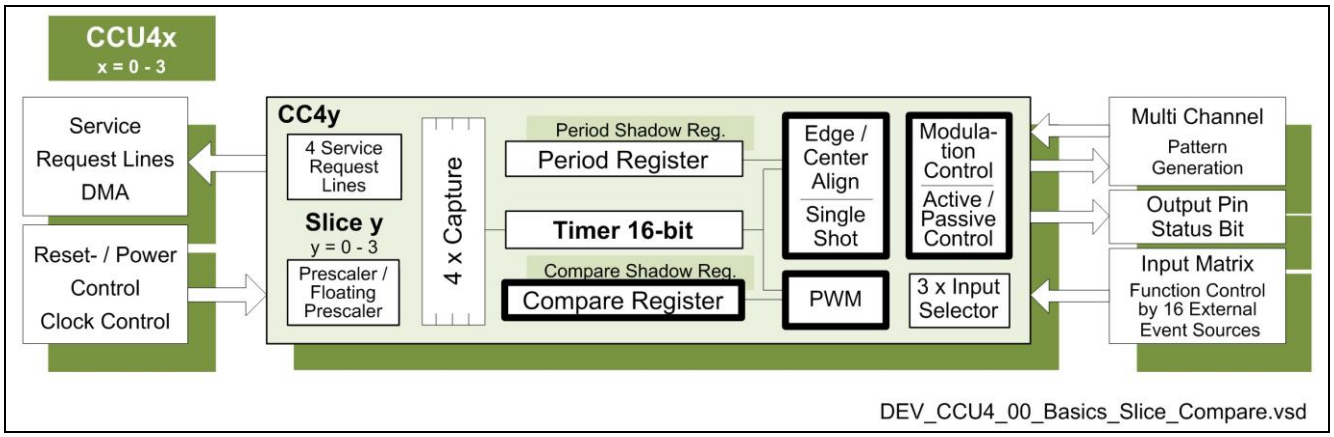

Figure 9 Timer Slice Compare Registers and PWM related Blocks

6.1.1 CCU4 Shadow Transfers

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 all function registers at a **Period-** or **One-Match**.

6.1.2 Shadow Transfer of Compare Register values

The compare values that are targeted for an update operation have to be written into the **CC4yCRS** shadow registers - *AND the corresponding* **Slice Transfer Set Enable** bits, e.g. **SySE** 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*)

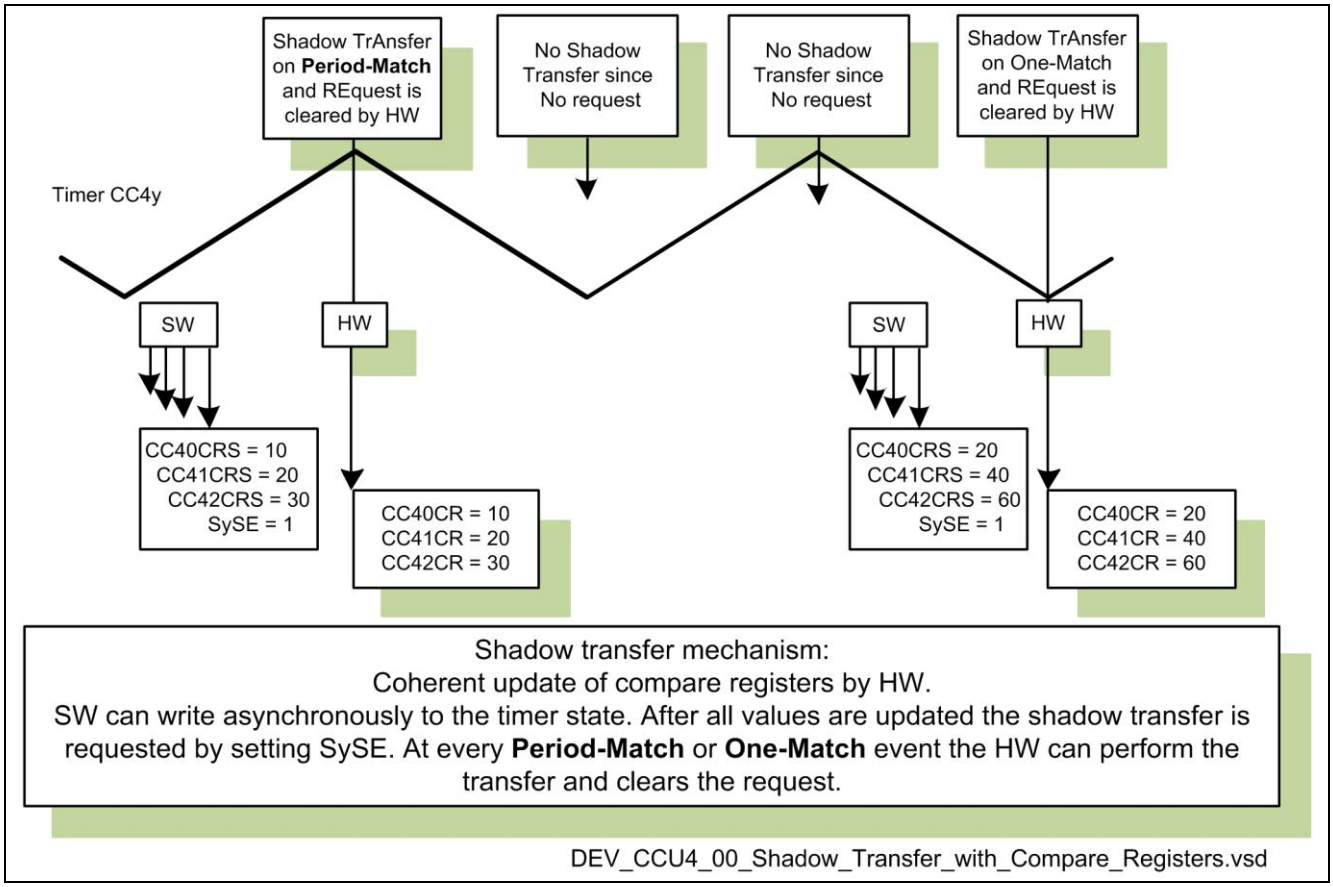

Figure 10 Basic Shadow Transfer Mechanism for Compare Register Values

6.2 Asymmetric Compare Events

The benefit of shadow transfers on both Period Match and One Match, allows asymmetric compare events to be provided in center aligned mode. The real time conditions are similar to handling shadow value updates in edge aligned mode. An even stronger solution for this is offered by the CCU8 units by offering two shadow registers.

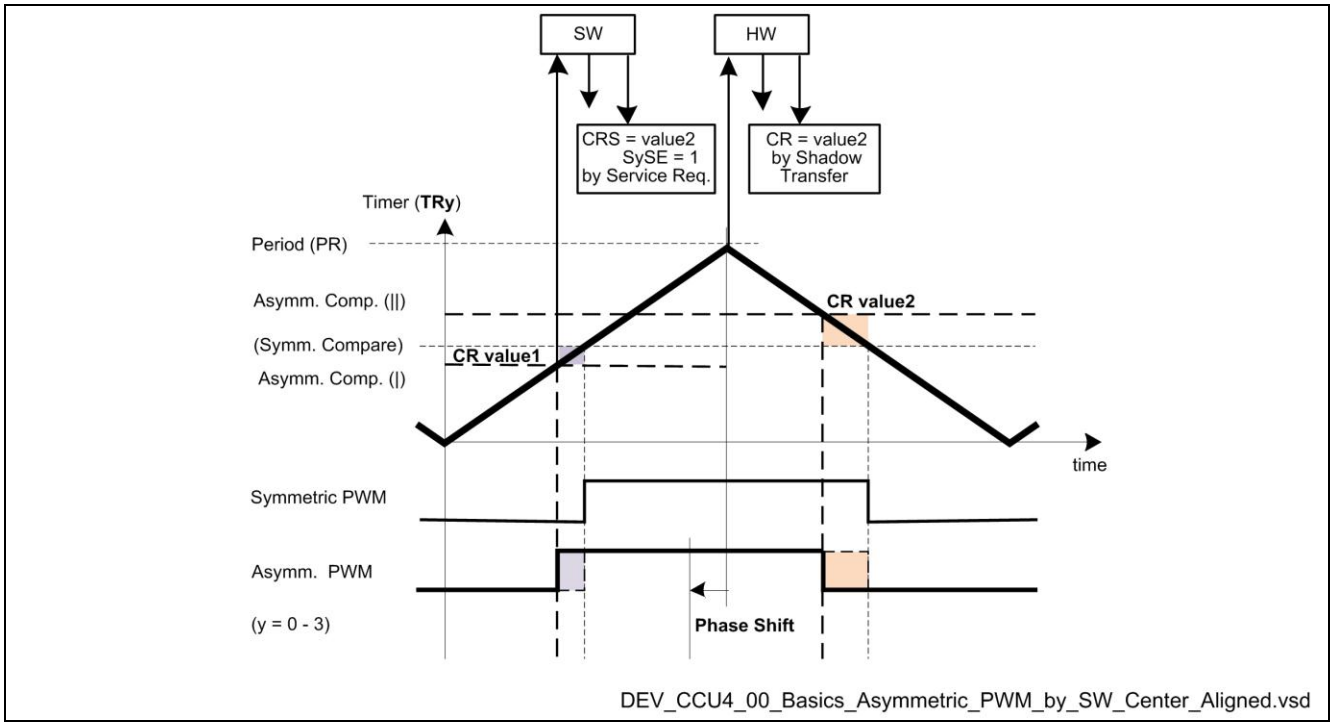

Figure 11 Asymmetric Compare by Shadow Transfers on both Period Match and One Match

6.3 Shadow Transfers in General - Compound Shadow Transfers

Beside the *Compare* (CR) values, there are also the the timer *Period* register (PR) and the PWM *Active/Passive control* bit (PSL) that may be updated simultaneously on the **SySE** flag. *Dithering* or *Floating Prescaler* values (not shown here though) are able to get simultaneous update too by the **SyDSE** & **SyPSE** request flags.

6.4 CCU4 Output State and 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. Wether 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**.

7 How to Start a Timer

7.1 CAPCOM4 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*.

7.2 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!

7.3 Start Timer Running

There are two ways to start a timer, either *directly* by SW setting the Timer Run Bit Set (**TRBS**) - or *indirectly* by HW when a considered event occurs in an external unit that may be selected for this, according to the **Top-Level Connection Matrix** for **External Events Control** – by *CAN, ADC, USIC, IO, CCU4/8, ERU1, POSIF* etc.

7.4 Global Start of CAPCOM Units

There is a way to get a synchronized start of CAPCOM Units, both CCU40/-41/-42,/-43 and CCU80/-81:

- **Global Start by SW**, by the CCUx Global Start Control bits in the Global Start Control register **CCUCON**
- **Global Start by HW**, indirectly, by **External Events Control** using the registers **CC4yINS** and **CC4yCMC**

7.5 Global Start of the CAPCOM Units CCU4 and CCU8

Global start of nearly any number of timers, independently of CAPCOM unit belonging, can be performed as shown here. By this manner a synchronization of the timers will be achieved and all timings can be controlled in parallel – such a output pattern generation of many different kinds.

See Figure 15.

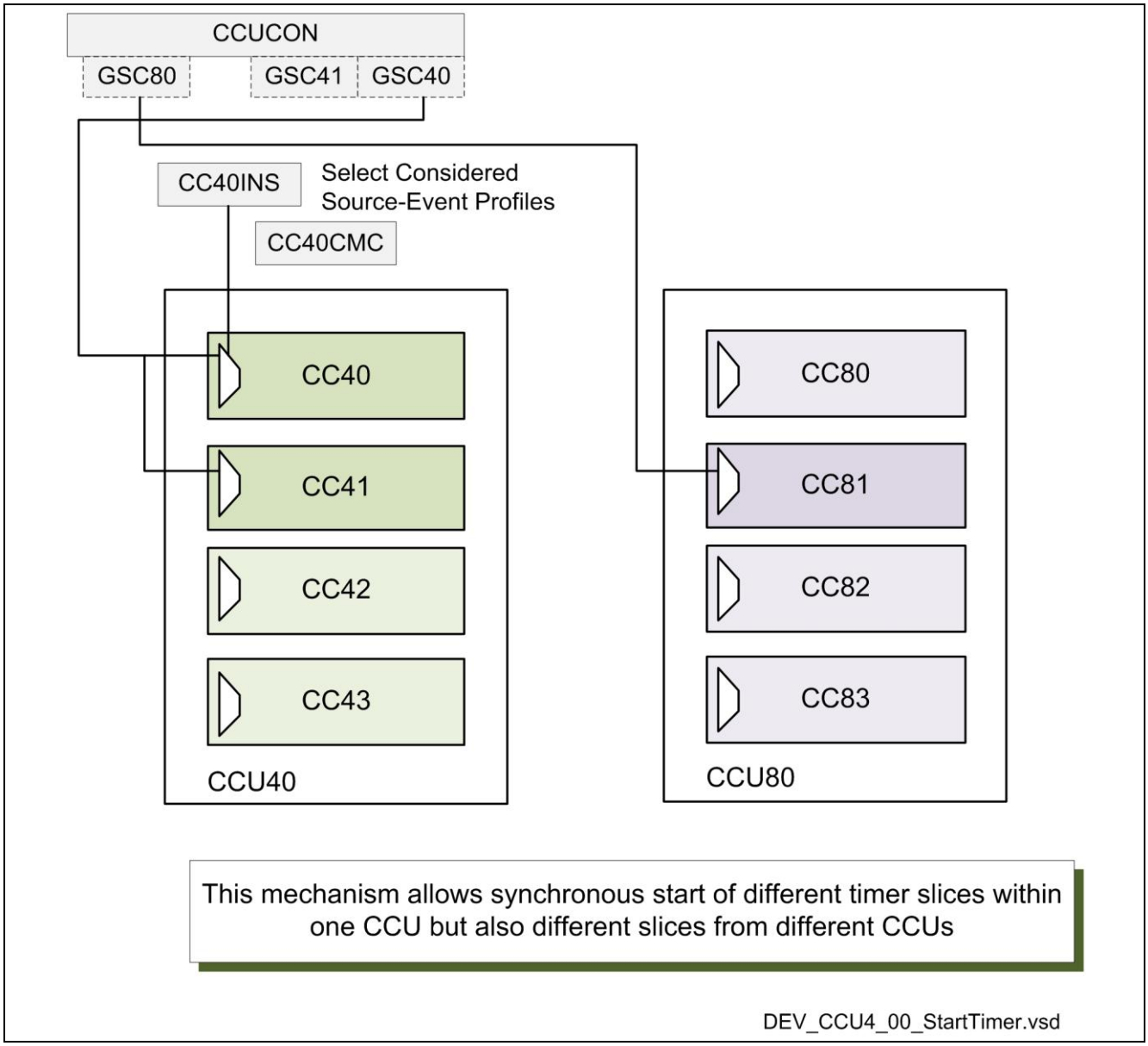

Figure 12 External Event Control with Global Start Command

8 Getting Started with Capture Compare Unit - CCU4

8.1 How to Start Timer CC4y (y=0-3)

In the following examples it is assumed that the CAPCOM4 Ultimate Initialization Sequence start is done before.

8.1.1 Example 1: Start Timer by SW (by Instruction)

```
CC40TCSET |= 0x00000001; // Start Timer CC40 directly by setting the TRBS bit = 1
```

8.2 Example 2: Start Timer by HW (by External Event)

8.2.1 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*.

8.2.2 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 *Source* EVxIS (**CC4yINS**[P:A]), Event *Mode* EVxEM or EVxLM bitfields
12. **CC4yTC.**<bitfields> Timer Mode selections e.g. Modes/*Extended modes, Counting Schemes etc*

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).

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!

```
// Example 2:
// Assume a falling edge transition on GPIO Pin P1.3 as a "considered" external
// event control signal to start Timer CC40 in CAPCOM Unit CCU40.
// Define this event as "Event2". (The alternatives Event0..1 would do as well!):
CC40CMC |= 0x00000003; // Start of Timer CC40 indirectly by External Event 2.

// Setup Timer CC40 External Event Control Input Selection register (CC40INS)
// to activate GPIO Pin P1.3 as selected source for Event 2 in the bitfield EV2S:
EV2S = 0; // where "0" is the input number (from 0-15) for the input "CCU40IN0[A]"
          // and represents Pin P1.3 according to the Top-level Connection Matrix.

EV2EM = 2; // set signal active on falling edge on Event 2, as "considered".
// See reference: Table 26-10 Pin Definition and I/O Functions
//
```

8.2.3 Example 3: Start Timers Synchronized (by Global Command)

```
// Start on Negative Edge on Port Pin P1.3 for CCU40IN0[A]
// See reference: Table 26-10 Pin Definition and I/O Functions
```

9 Runtime Handling of Capture Compare Unit – CCU4

9.1 Example: Handling Shadow Transfers

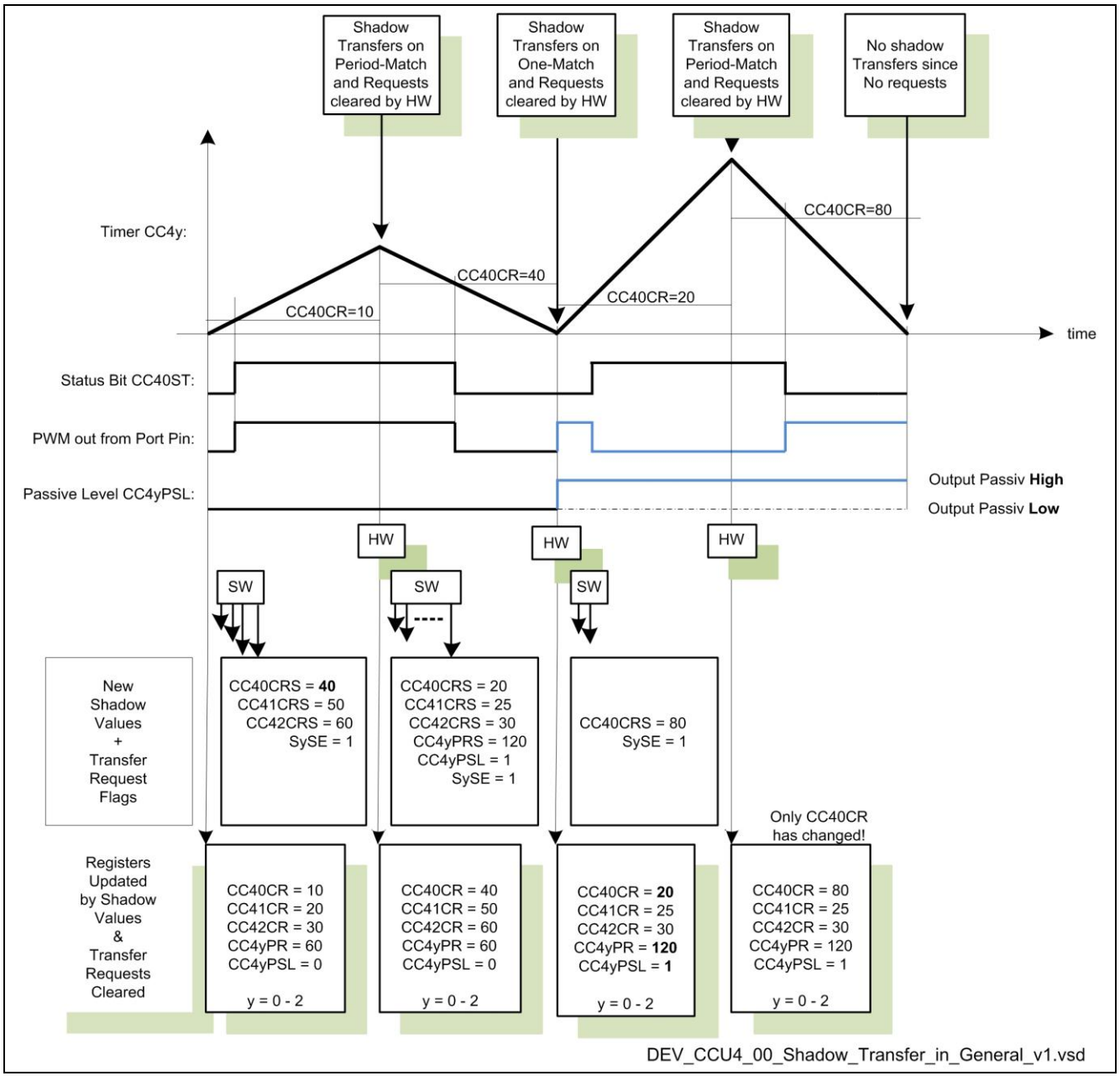

Figure 13 Compound Shadow Transfer Mechanism with Coherent Update of PWM

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

Published by Infineon Technologies AG