XMC4000

XMC4500 Family

CCU4 and Dithering

Signal Quantification & Quality Enhancement by Dithering

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 Signal Quantification & Quality Enhancement by Dithering 5
1.1 A simple case of value adding by Dithering PWM 5
1.1.1 Dithering by SW 5
1.1.2 Dithering by HW 5
1.2 A simple case of value adding by Dithering Micro Ticks 5

2 The Dithering Implementation 6
2.1.1 The Dithering Function 6
2.1.2 The Dithering Range 6
2.1.3 The Dithering Algorithm 7
2.1.4 The Dithering Distribution 7

3 Getting Started with CCU4 and Dithering 11
3.1 Dithering Setup 11
3.2 Setting an Averege Compare Level by Dithering 11

4 Runtime Handling of CCU4 and Dithering 12
4.1 Reload of Dithering PWM parameters 12
4.1.1 Shadow Transfer of Compare and Dither Register values 12
4.2 Top-Level Control of Event Requests to/from a Timer in Dithering Mode 13

5 Example Application for CCU4 and Dithering 13
5.1 Demo - Using Dithering in Micro Ticks Generation 13
5.2 Demo cont:d - Enhancing Micro Ticks Resolution 15

List of Figures

Figure 1 Buck Converter with high precision requirement 5
Figure 2 Optimized distribution of Micro Ticks within a certain time-slot by Dithering 6
Figure 3 Demonstration of the Dithering Function for Edge or Center Aligned Timer Periods 7
Figure 4 Linear Relationship: Average Bit-Time vs. Quotient of Periods with Delayed Compare Events 8
Figure 5 Dithering of Period Width in Edge Aligned Mode 9
Figure 6 Dithering of Pulse Width by Compare in Edge Aligned Mode 10
Figure 7 Getting Started Demo - Setting an Averege Compare Level of 28.9H by Dithering 11
Figure 8 Setup and Reload of Dithering PWM 12
Figure 9 The calculation process for Micro Ticks distribution over period time by Dithering 14
Figure 10 Futher enhancement of resolution over period time with Micro Ticks by Dithering 15

1 Signal Quantification & Quality Enhancement by Dithering

Dithering is a technique actually used to reduce or remove the correlation between noise and signal in audio or graphic signal processing techniques – also used to enhance EMC quality by distributing portions of the signal quantification error in time. This effect can be utilized to enhance PWM DC-level precision or Micro Ticks tuning.

1.1 A simple case of value adding by Dithering PWM

See Figure 1.

Consider a Buck Converter, from V_{in} to V_{out}, such as shown here. Let the PWM frequency be 800 kHz and the accuracy (precision) requirement 0.1%. If one assumes a timer clock frequency of 20 MHz, then the timer period would only be able to perform a 6-bit counter length, which implies that just 1.6% precision would be achievable.

1.1.1 Dithering by SW

Theoretically, it would be possible to control the average value, in-the-long-run, for a 0.1% precision with a 6-bit counter by having a loop control that modifies the PWM from one timer period to another in some weighted way (and signal smoothing by an LC-filter) – *but* practically not by SW though, due to timing restrains and CPU load.

1.1.2 Dithering by HW

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.

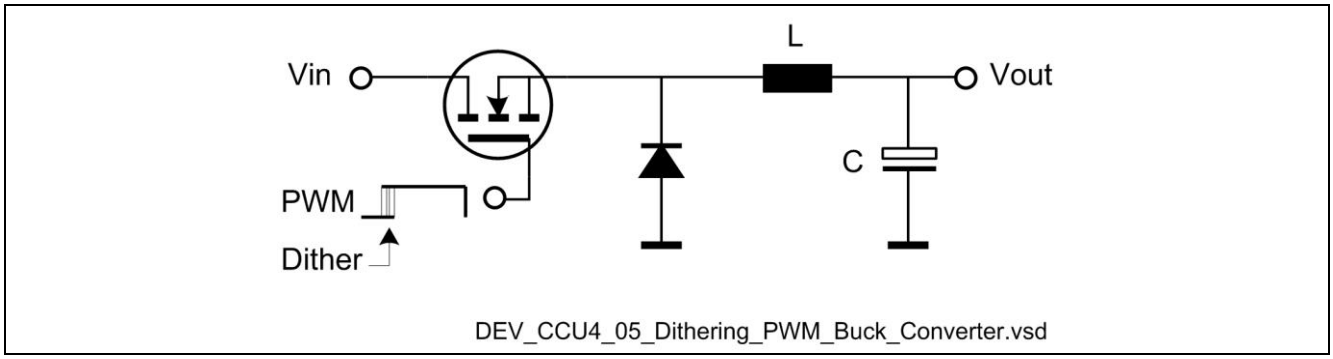

Figure 1 Buck Converter with high precision requirement

1.2 A simple case of value adding by Dithering Micro Ticks

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.

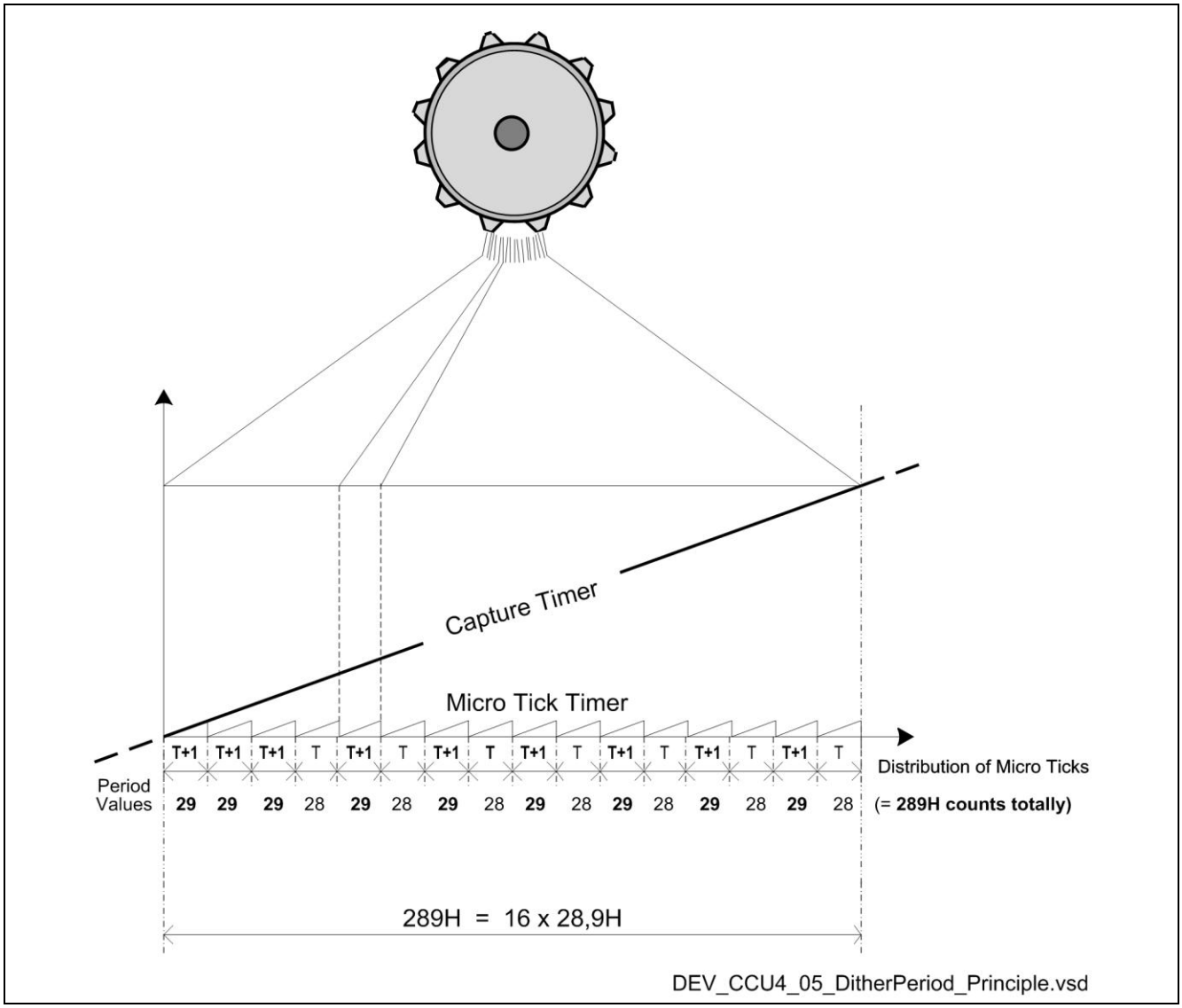

Figure 2 Optimized distribution of Micro Ticks within a certain time-slot by Dithering

2 The Dithering Implementation

For each timer there is an extra counter, a so called **Dither Counter**, which can dither the timer PWM edges. The precision is gained by the same magnitude as the count length of the counter. In the current implementation there is 4-bit Dither Counter for each CAPCOM4 slice timer, thus offering a precision leverage by 16 times.

2.1.1 The Dithering Function

The dithering function can be explained in the following way. Dithering is accomplished by increasing the PWM timer period with 1 bit-time or by delaying the compare event for the pulse width with 1 bit-time (or both these operations) at certain periods of the PWM timer, controlled by the dither logic and the algorithms shown below.

See Figure 3.

2.1.2 The Dithering Range

For example, if the compare event is delayed with 1 bit-time in 9 of 16 periods repeatedly, in a cyclic manner, then the average PWM value would be reduced with 9/16 of a bit-time – i.e. there is a trivial way of fine tuning the precision of a PWM signal average value within a dynamic range of 1 bit-time length.

2.1.3 The Dithering Algorithm

By utilizing the linear relationship between the average bit-time and the quotient of periods using delayed compare events within a specific cycle length of timer periods – then one will have the mechanism behind the precision enhancement by dithering. And the precision leverage is proportional to that specific cycle length:

- A 4-bit binary Dither Counter seems to give the PWM timer a 4-bit precision extention, from 16 to 20 bits! See Figure 4.

2.1.4 The Dithering Distribution

Now, a Dither Counter with a binary representation would not give an optimal distribution of dithered periods, with regard to how the quanticification errors would be spread in time (for low pass filtering or EMC as well) - so the Dither Counter values will be split up according to a *bit-reverse representation*, as demonstrated here.

See Figure 3.

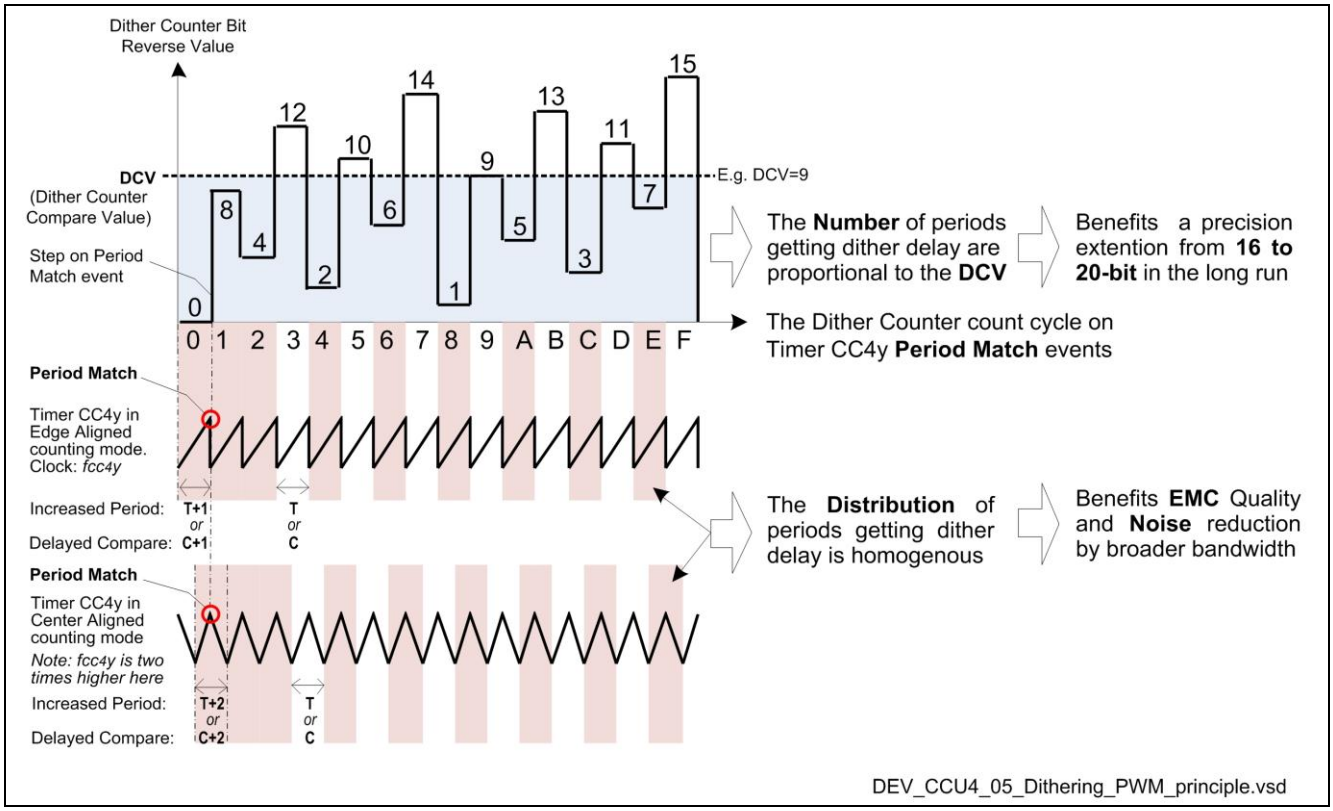

Figure 3 Demonstration of the Dithering Function for Edge or Center Aligned Timer Periods

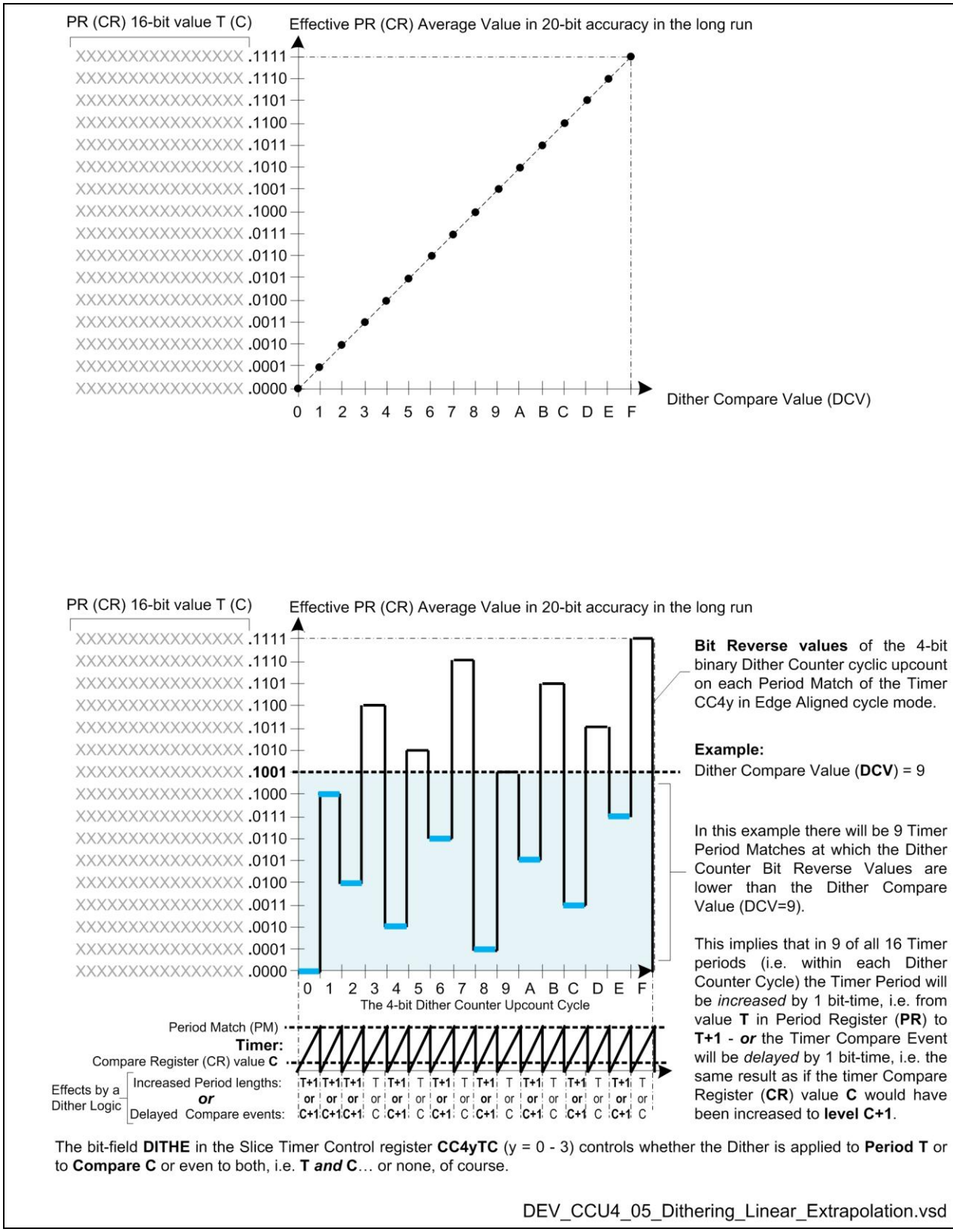

Figure 4 Linear Relationship: Average Bit-Time vs. Quotient of Periods with Delayed Compare Events

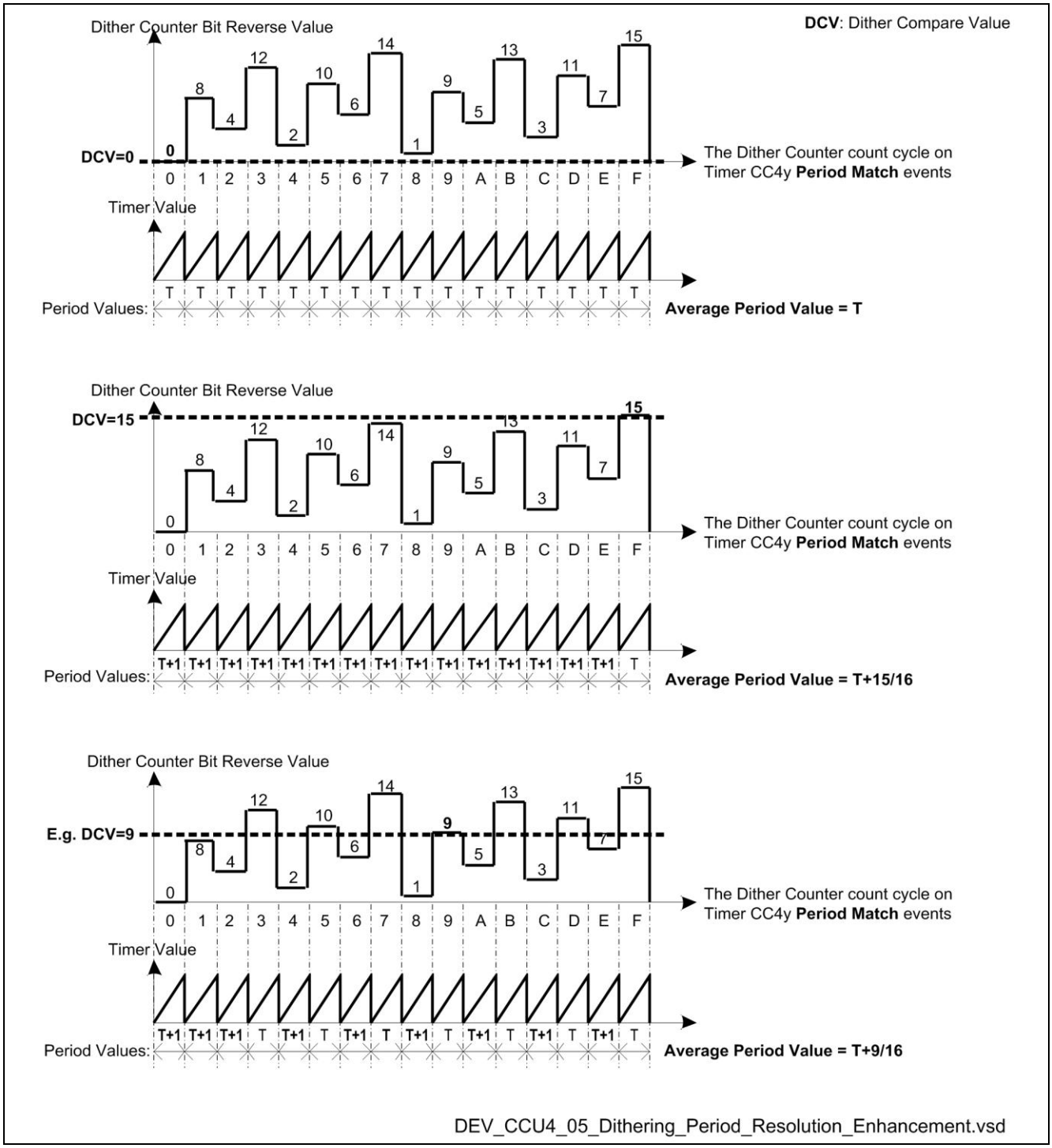

Figure 5 Dithering of Period Width in Edge Aligned Mode

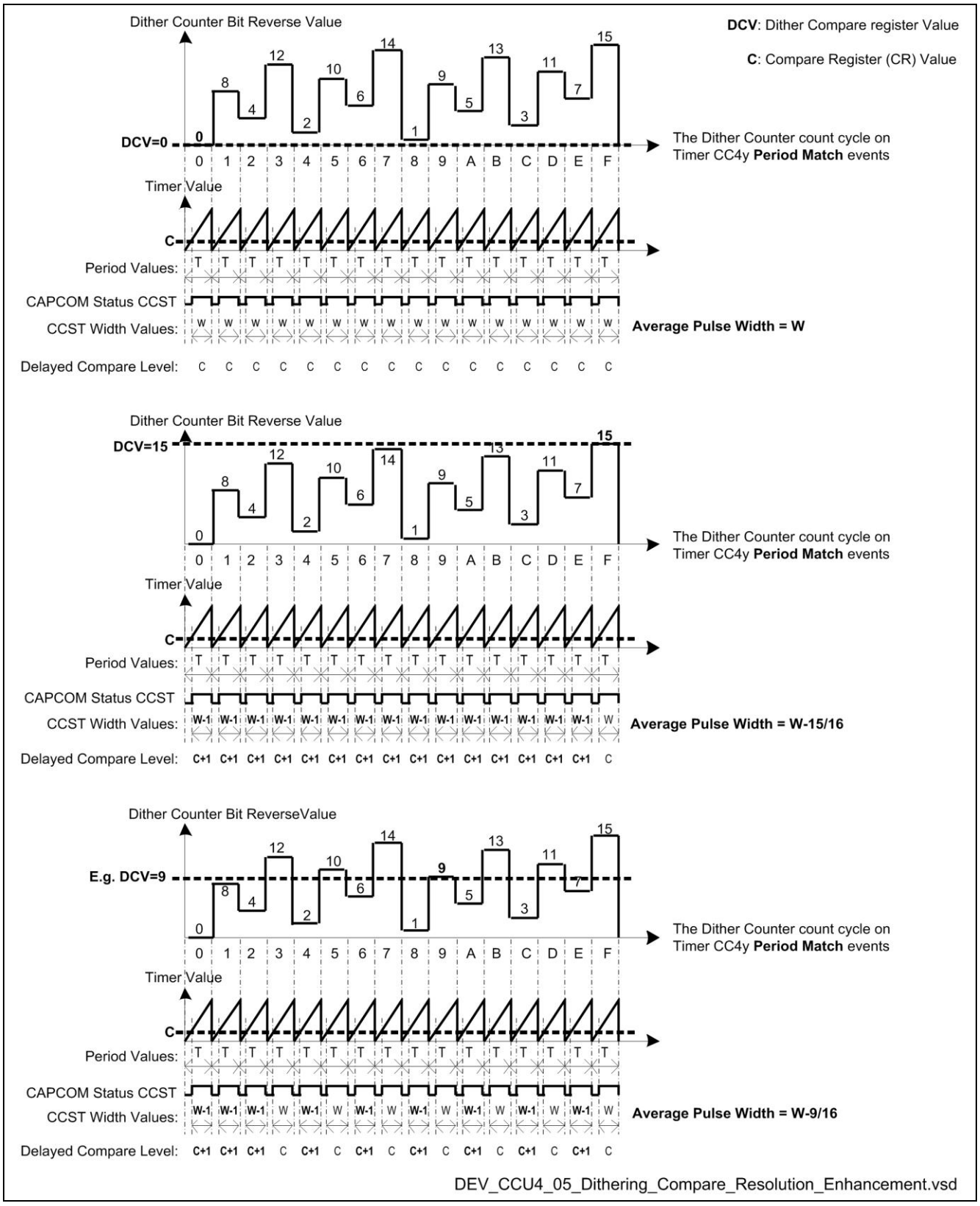

Figure 6 Dithering of Pulse Width by Compare in Edge Aligned Mode

3 Getting Started with CCU4 and Dithering

3.1 Dithering Setup

Dither mode and initialization should be setup separately or together with other parameters represented in a Slice y (y=3.0) Timer Control register **CC4yTC**.

3.2 Setting an Averege Compare Level by Dithering

See Figure 7.

Assume that an Averege Compare Level of 28.9H by Dithering should be set:

After setting the Dither Compare Match Enable (**DITHE**) the compare values that are targeted for reload operations should be written into the **CC4yCRS** respective **CC4yDIT** shadow registers. However, no transfer from those shadow registers will be done until the **Transfer Set Enable** bits, **SySE**, **SyDSE** in **GCSS** are preset.

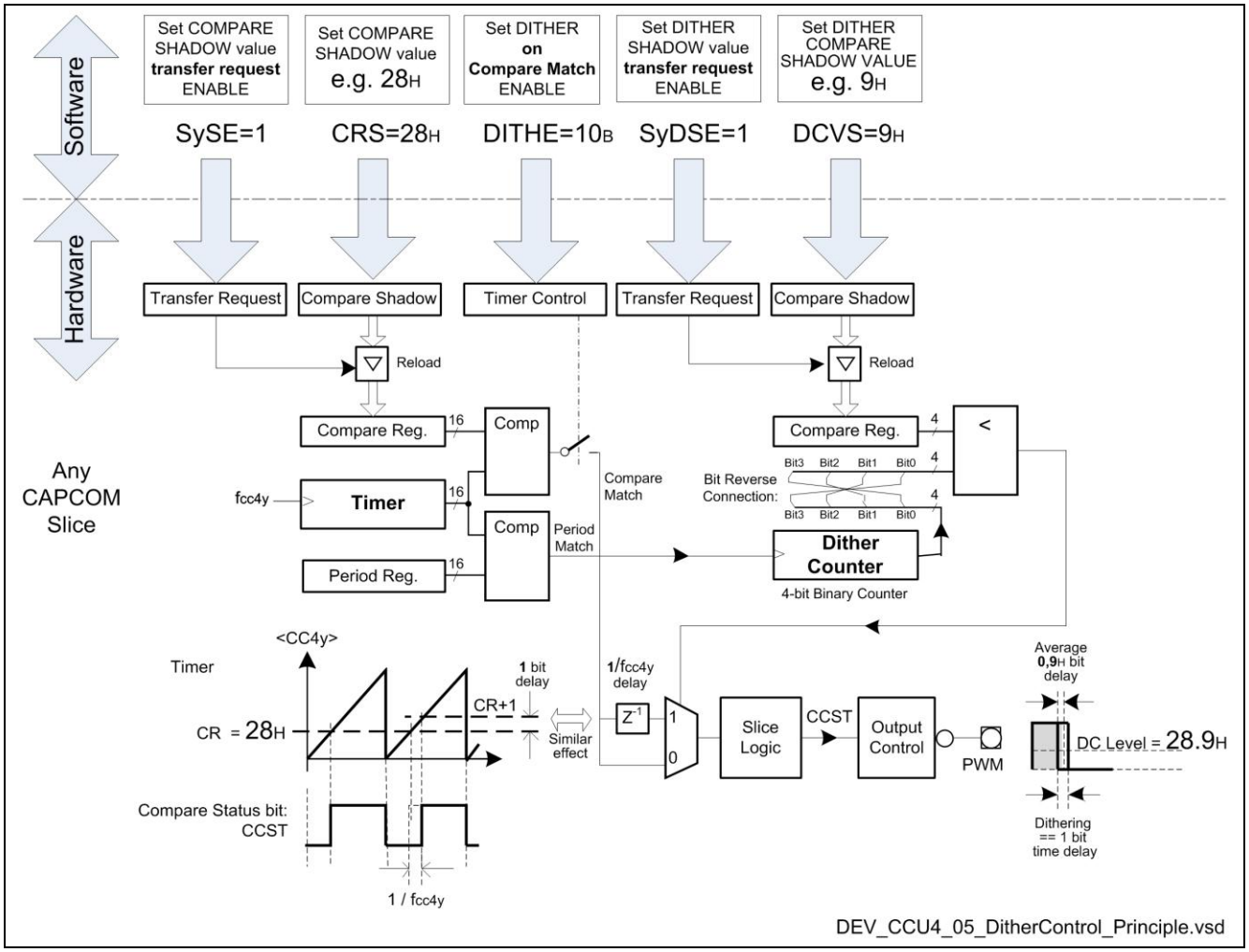

Figure 7 Getting Started Demo - Setting an Averege Compare Level of 28.9H by Dithering

4 Runtime Handling of CCU4 and Dithering

4.1 Reload of Dithering PWM parameters

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 resp. function registers at **Period-** or **One-Match**.

4.1.1 Shadow Transfer of Compare and Dither Register values

The compare values that are targeted for reload operations must be written in the **CC4yCRS** resp. **CC4yDIT** shadow registers - *AND the respective* **Transfer Set Enable** bits, **SySE**, **SyDSE** 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*)

See Figure 8.

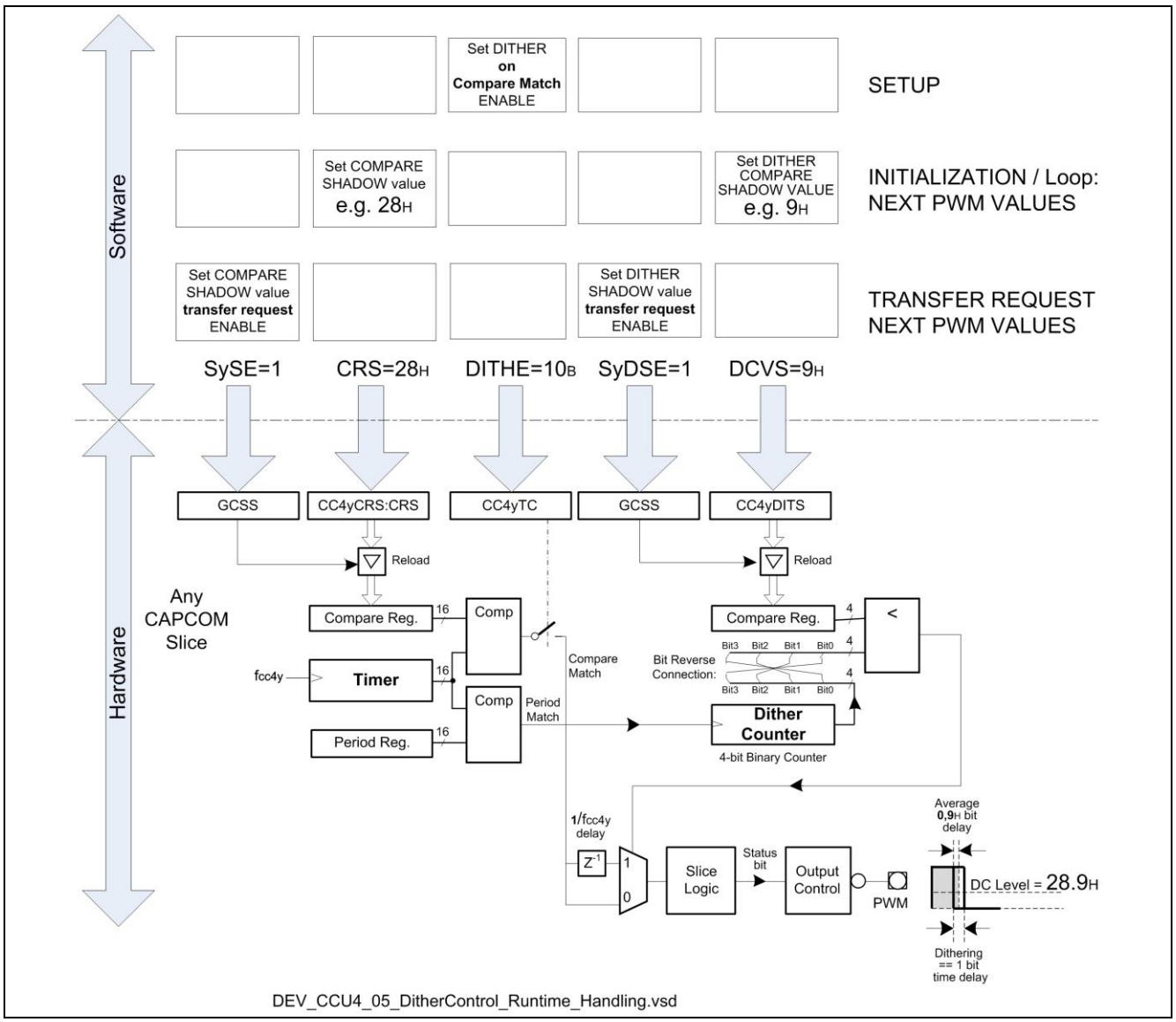

Figure 8 Setup and Reload of Dithering PWM

4.2 Top-Level Control of Event Requests to/from a Timer in Dithering 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. It can be used to supervise Dithering Reload.

5 Example Application for CCU4 and Dithering

5.1 Demo - Using Dithering in Micro Ticks Generation

Micro Ticks are clock pulses that are created and distributed within the time-slots of a pulse train for resolution enhancement purposes – as in *Position Monitoring* of a rotating cog wheel shown here. The time-slots are given by a position sensor and detected by a Capture Timer. The Micro Tick Timer is controlled by specific algorithms.

See Figure 9.

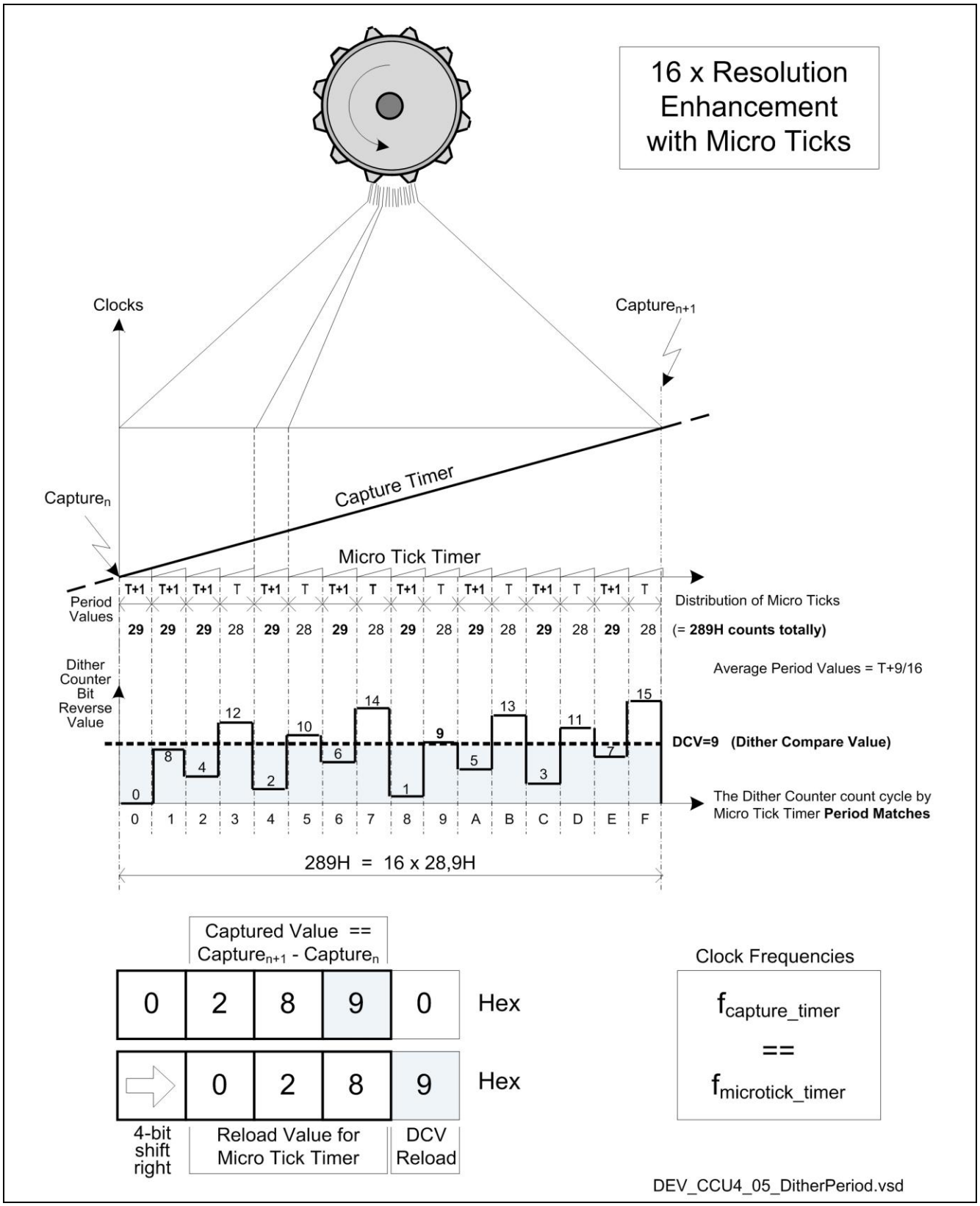

Figure 9 The calculation process for Micro Ticks distribution over period time by Dithering

The controlling algorithms may utilize *dithering* to *distribute* the micro ticks optimally. The clock rate quotient between the Micro-Tick and Capture timers could be simplified by a 4-bit shift as shown here. However, the algorithms have to regard time-slot variations too, since the micro ticks are by nature calculated retrospectively.

5.2 Demo cont:d - Enhancing Micro Ticks Resolution

By magnifying the clock frequency quotient difference between the capture timer clock and the micro ticks timer clock by factors of 2, the resolution in micro ticks will be enhanced accordingly - with sustained optimization in distribution without changing the algorithm. Thus the position measurement precision can be magnified this way.

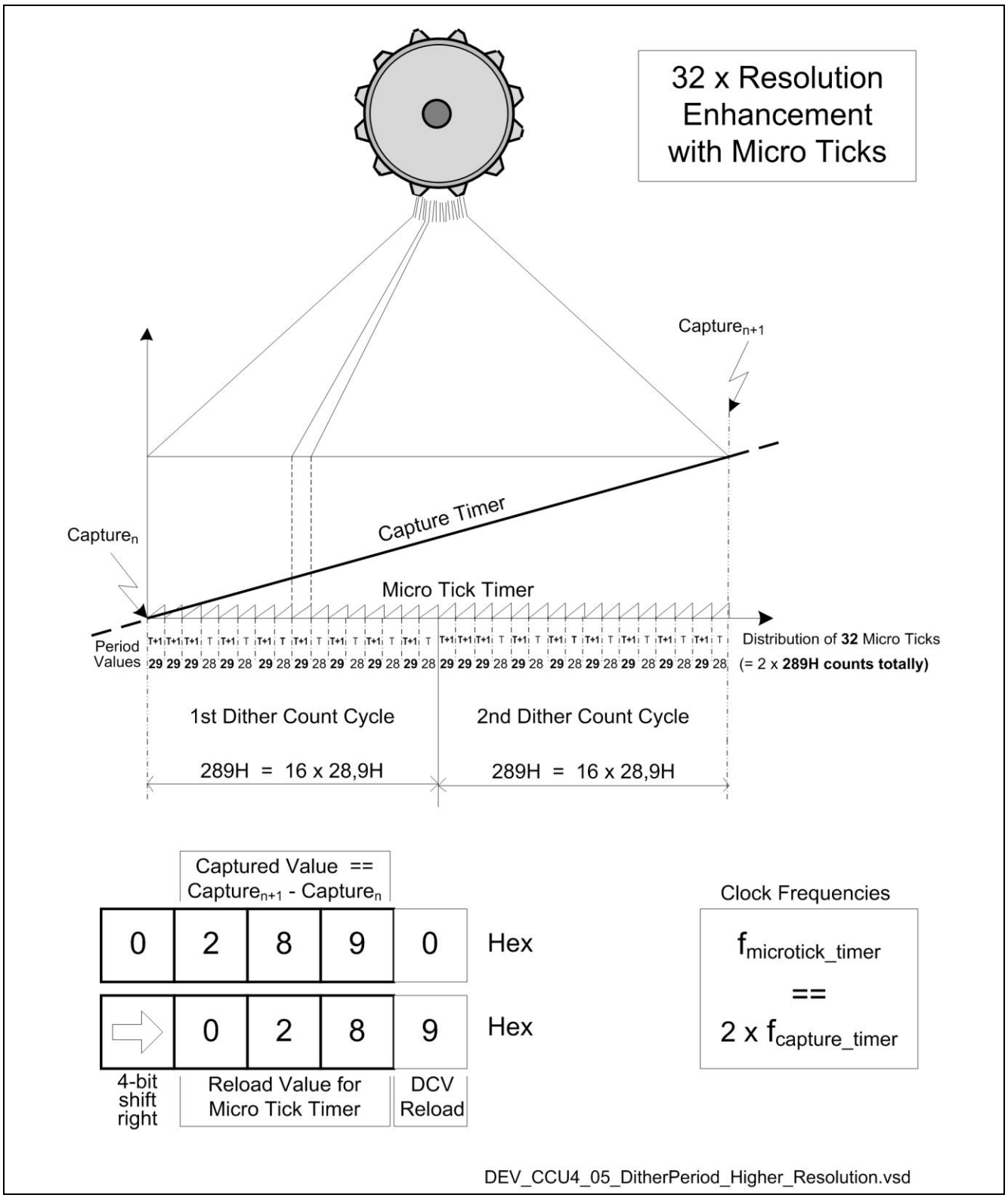

Figure 10 Futher enhancement of resolution over period time with Micro Ticks by Dithering

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

Published by Infineon Technologies AG