XMC4000

XMC4500 Family

CCU4 and Output State Override

Output State Override on External Events

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 Output State Override on External Events 5
1.1 Use Cases 5

2 External Events Control of a Timer Output State 7
2.1 External Events Control of Timer Input Functions in general 7
2.2 Selection of External Events Control Sources and Input Functions 7

3 Output State Override 8

4 Getting Started with CCU4 and Output State Override 9
4.1 Ultimate Initialization Sequence 9
4.1.1 Ultimate Start-Up Enable 9
4.2 External Events Control Setup Registers 9
4.2.1 Using the Output State Override Control Register Bitfields 9
4.2.2 Pseudo Code for the Output State Override Setup 9

5 Example Application for CCU4 and Output State Override 11
5.1 Principle of Peak&Hold with Current Limitation by Output State Override 11

List of Figures

Figure 1 Use case - Current Control and Protection by Monitoring using Output State Override 6
Figure 2 External Events Control of a Timer Slice Input Functions 7
Figure 3 Output State Override illustration 8
Figure 4 Setting Output State Override Parameters (Control Registers) 10
Figure 5 Principle of Peak&Hold with Current Limitation by Output State Override 11

1 Output State Override on External Events

It is possible to override a slice's output state with the *value of an external source* (or its *complement value*) on the edge event of *another external source*. The *source properties* should be mapped to the **CC4yINS** register and the *override input function* by the **CC4yCMC** register. A use case is e.g. output timing control by monitoring.

1.1 Use Cases

Output State Override is useful for emergency protection (HW-TRAP) or hysteretic control (simple on/off via external overcurrent comparator for primitive control loop). Note that comparator can be the ADC boundary flag. Example: PWM timing may be adjusted by Output State Overrride with regard to different current boundaries.

See Figure 1.

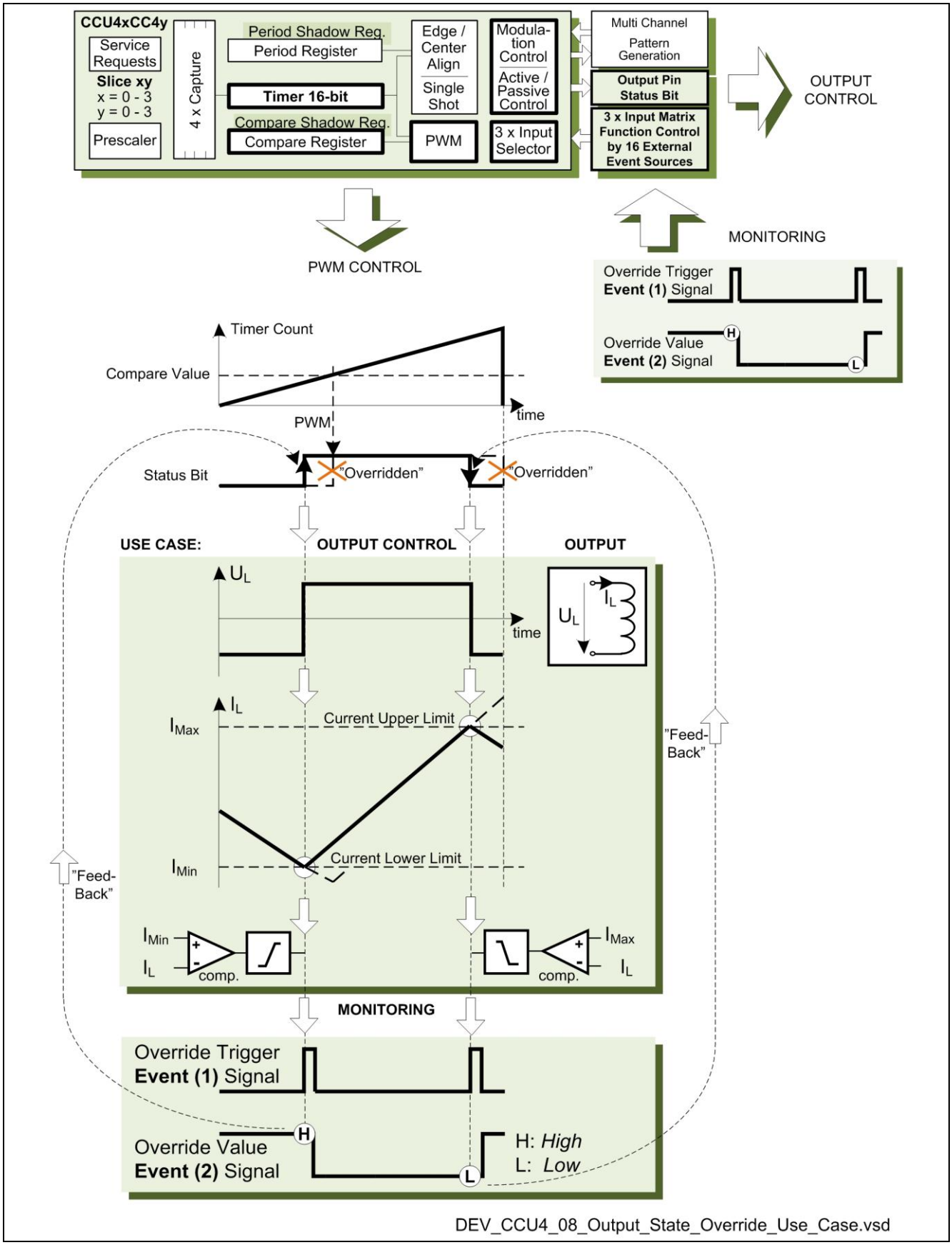

Figure 1 Use case - Current Control and Protection by Monitoring using Output State Override

2 External Events Control of a Timer Output State

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 profile* can be Output State Overriden remotely-like this way!

2.1 External Events Control of Timer Input Functions in general

There are 11 Timer Input Functions (as e.g. *Bit Overrride*), 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, Status Bit Override, Trap & Modulate Output* control.

See Figure 2.

2.2 Selection of External Events Control Sources and Input Functions

The *Output State Override* function can be linked to external trigger requests – 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.

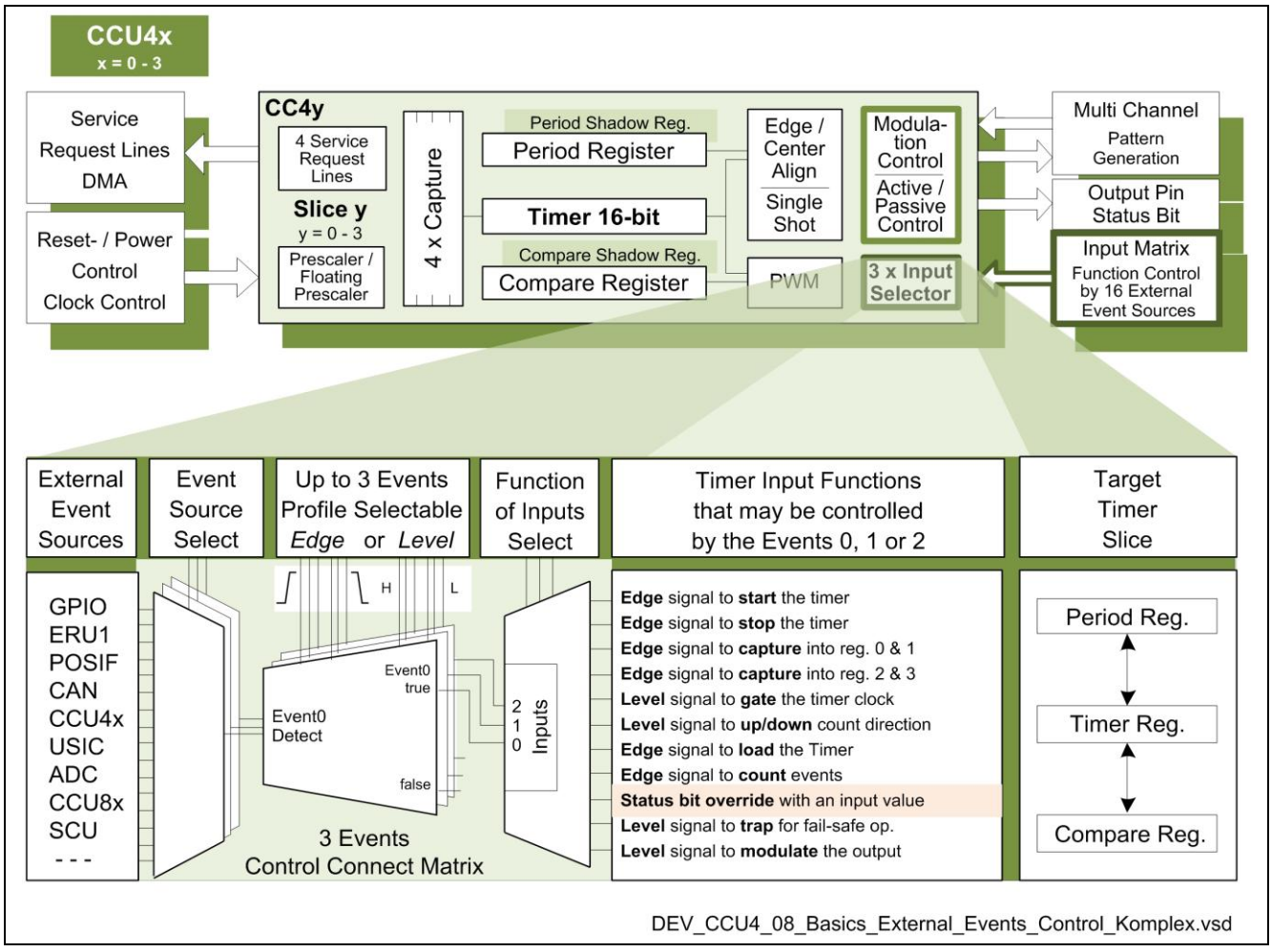

Figure 2 External Events Control of a Timer Slice Input Functions

3 Output State Override

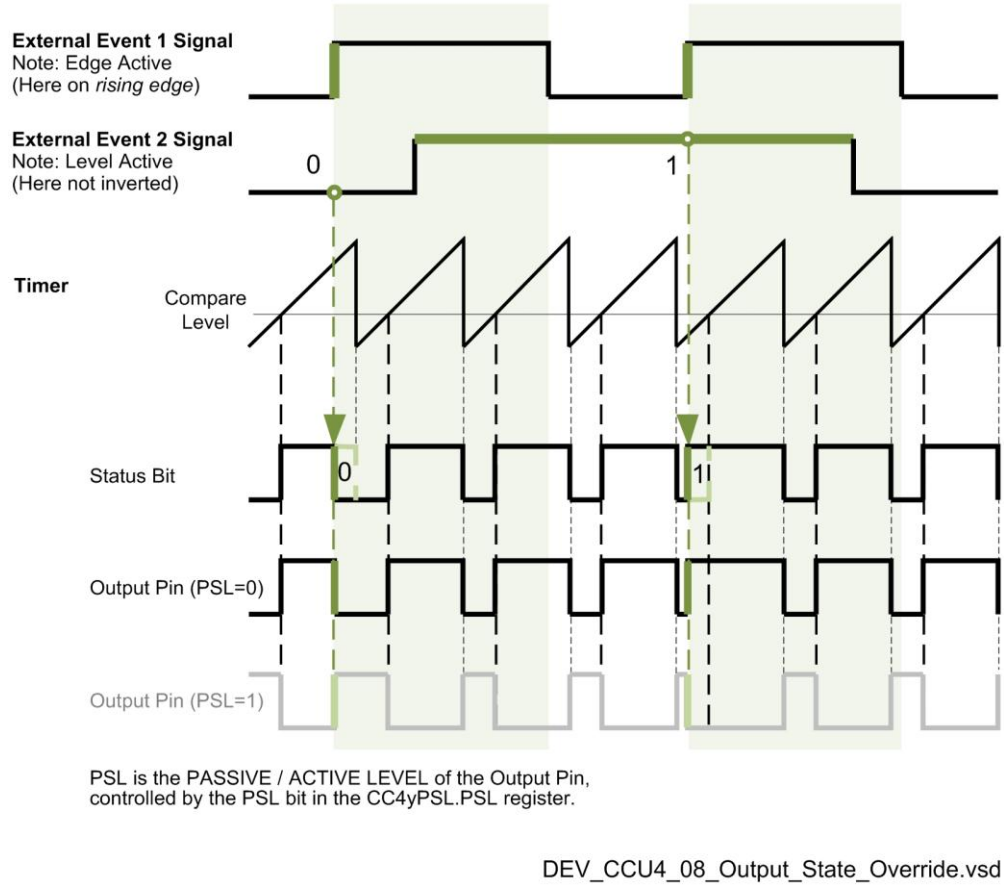

Figure 3 Output State Override illustration

4 Getting Started with CCU4 and Output State Override

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!

All these topics are covered in detail by the Reference Manual: <<< Link to 23.6.1 Initialization Sequence >>>

4.2 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*.

4.2.1 Using the Output State Override Control Register Bitfields

1. **CC4yCMC.**OFS — *Input Function* Selection bit to enable the Output State Override Function
2. **CC4yINS.**<Event1 bitfields> — *Event1 Source* EV1IS (**CC4yINS**[P:A]) and *Edge* Mode EV1EM bitfields
3. **CC4yINS.**<Event2 bitfields> — *Event2 Source* EV2IS (**CC4yINS**[P:A]) and *Level* Mode EV2LM bitfields
4. **CC4yTC.**<Mode bitfields> — Timer Mode selections e.g. *Modes/Extended modes, Counting Schemes etc*

Note: The Output State Override Function function is Edge Sensitive.

Note: The Carries of the Override State Value is Level Sensitive.

Note: ***Selection of Event Numbers are Default!!!*** I.e. *Event 1 is Trigger Edge signal, so the Output State Override time point is controlled by Source 1. Event 2 is Carrier of Level signal, so the Output State Override value is controlled by the level of Source 2.*

4.2.2 Pseudo Code for the Output State Override 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 |= OFS_FUNCTION_AND_EVENTS_BITFIELDS // Enable the Output State Override
// step 2
CC4yINS &= OFS_SOURCE1_AND_EVENT1_MODES_BITFIELDS_MASK; //Focus on Source1 events
CC4yINS |= OFS_SOURCE1_AND_EVENT1_MODES_BITFIELDS_PATTERN;  //Paste setup pattern
// step 3
CC4yINS &= OFS_SOURCE2_AND_EVENT2_MODES_BITFIELDS_MASK; //Focus on Source1 events
CC4yINS |= OFS_SOURCE2_AND_EVENT2_MODES_BITFIELDS_PATTERN;  //Paste setup pattern

// step 2 & 3 could naturally be merged into one operation sequence, as follows:
CC4yINS &= OFS_SOURCES_AND_EVENTS_MODES_BITFIELDS_MASK; //Focus on all bitfielfs
CC4yINS |= OFS_SOURCES_AND_EVENTS_MODES_BITFIELDS_PATTERN; //Paste both setups!
```

```
// Manipulation of the Output Pin PASSIVE/ACTIVE LEVEL control bit (PSL) if needed
// CC4yPSL = ACTUAL_PASSIVE_LEVEL // Use for LOW: 0x00000000 or HIGH: 0x00000001
// (There are no other bits but the PSL bit (CC4yPSL.PSL) in this register!)
```

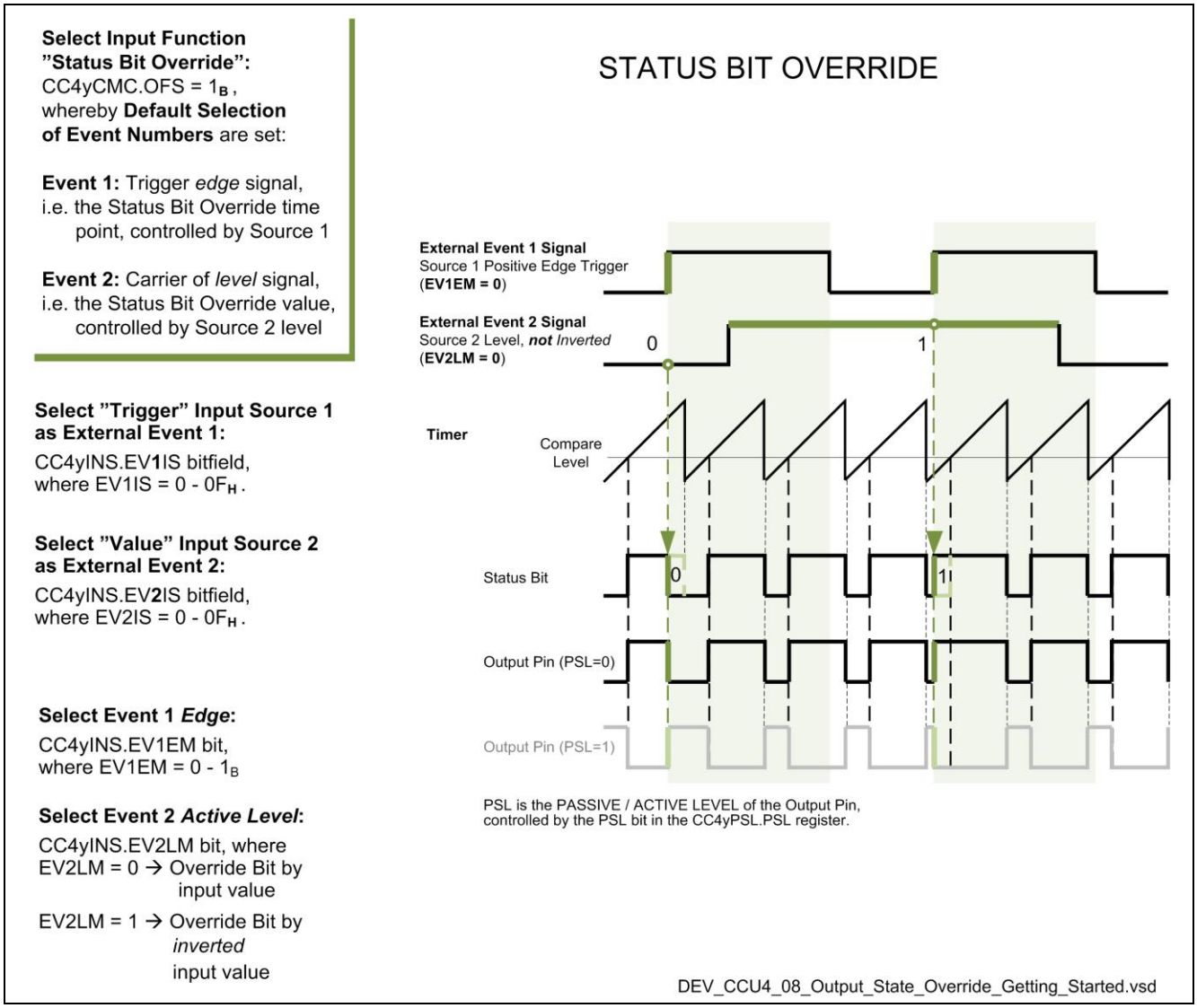

Figure 4 Setting Output State Override Parameters (Control Registers)

5 Example Application for CCU4 and Output State Override

5.1 Principle of Peak&Hold with Current Limitation by Output State Override

See Figure 5.

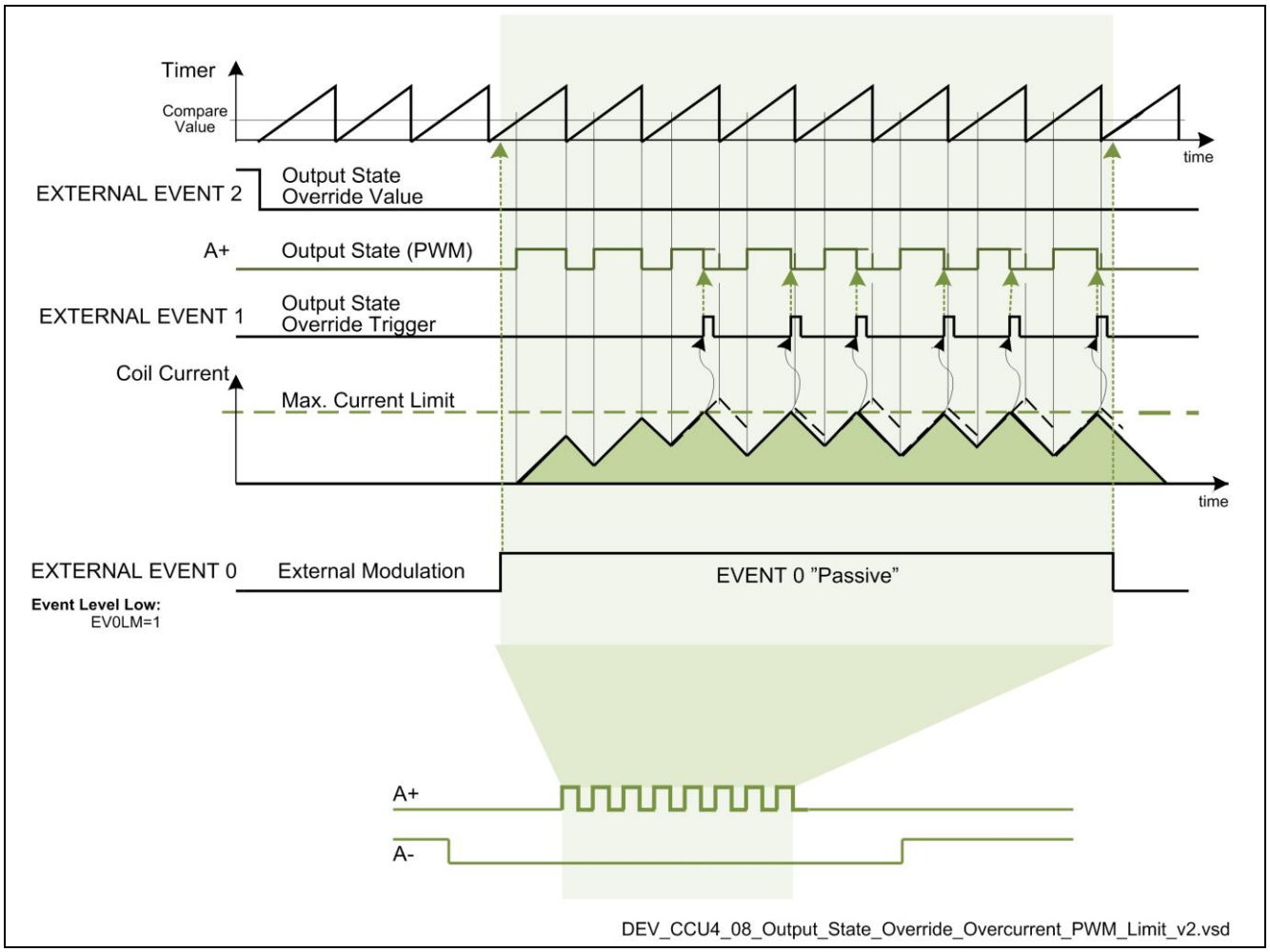

Figure 5 Principle of Peak&Hold with Current Limitation by Output State Override

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

Published by Infineon Technologies AG