

CNTHV_CVAL_EL2, Counter-timer Virtual Timer CompareValue register (EL2)

The CNTHV_CVAL_EL2 characteristics are:

Purpose

Holds the compare value for the EL2 virtual timer.

Configuration

AArch64 System register CNTHV_CVAL_EL2 bits [63:0] are architecturally mapped to AArch32 System register [CNTHV_CVAL\[63:0\]](#).

This register is present only when FEAT_VHE is implemented and (EL3 is implemented or (EL3 is not implemented and FEAT_SEL2 is not implemented)). Otherwise, direct accesses to CNTHV_CVAL_EL2 are undefined.

If EL2 is not implemented, this register is res0 from EL3.

Attributes

CNTHV_CVAL_EL2 is a 64-bit register.

Field descriptions

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
CompareValue																															
CompareValue																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

CompareValue, bits [63:0]

Holds the EL2 virtual timer CompareValue.

When [CNTHV_CTL_EL2](#).ENABLE is 1, the timer condition is met when ([CNTVCT_ELO](#) - CompareValue) is greater than or equal to zero. This means that CompareValue acts like a 64-bit upcounter timer. When the timer condition is met:

- [CNTHV_CTL_EL2](#).ISTATUS is set to 1.
- If [CNTHV_CTL_EL2](#).IMASK is 0, an interrupt is generated.

When [CNTHV_CTL_EL2](#).ENABLE is 0, the timer condition is not met, but [CNTVCT_ELO](#) continues to count.

If the Generic counter is implemented at a size less than 64 bits, then this field is permitted to be implemented at the same width as the counter, and the upper bits are res0.

The value of this field is treated as zero-extended in all counter calculations.

The reset behavior of this field is:

- On a Warm reset, this field resets to an architecturally unknown value.

Accessing CNTHV_CVAL_EL2

When [HCR_EL2.E2H](#) is 1, without explicit synchronization, access from EL2 using the mnemonic CNTHV_CVAL_EL2 or CNTV_CVAL_EL0 are not guaranteed to be ordered with respect to accesses using the other mnemonic.

Accesses to this register use the following encodings in the System register encoding space:

MRS <Xt>, CNTHV_CVAL_EL2

op0	op1	CRn	CRm	op2
0b11	0b100	0b1110	0b0011	0b010

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.NV == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        UNDEFINED;
elsif PSTATE.EL == EL2 then
    X[t, 64] = CNTHV_CVAL_EL2;
elsif PSTATE.EL == EL3 then
    X[t, 64] = CNTHV_CVAL_EL2;
```

MSR CNTHV_CVAL_EL2, <Xt>

op0	op1	CRn	CRm	op2
0b11	0b100	0b1110	0b0011	0b010

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.NV == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        UNDEFINED;
elsif PSTATE.EL == EL2 then
    CNTHV_CVAL_EL2 = X[t, 64];
elsif PSTATE.EL == EL3 then
    CNTHV_CVAL_EL2 = X[t, 64];

```

MRS <Xt>, CNTV_CVAL_EL0

op0	op1	CRn	CRm	op2
0b11	0b011	0b1110	0b0011	0b010

```

if PSTATE.EL == EL0 then
    if !(EL2Enabled() && HCR_EL2.<E2H,TGE> == '11')
    && CNTKCTL_EL1.EL0VTEN == '0' then
        if EL2Enabled() && HCR_EL2.TGE == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        else
            AArch64.SystemAccessTrap(EL1, 0x18);
        elsif EL2Enabled() && HCR_EL2.<E2H,TGE> == '11'
    && CNTHCTL_EL2.EL0VTEN == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
        elsif EL2Enabled() && HCR_EL2.<E2H,TGE> != '11'
    && CNTHCTL_EL2.EL1TVT == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
        elsif EL2Enabled() && HCR_EL2.<E2H,TGE> == '11'
    && SCR_EL3.NS == '0' &&
    IsFeatureImplemented(FEAT_SEL2) then
            X[t, 64] = CNTHVS_CVAL_EL2;
        elsif EL2Enabled() && HCR_EL2.<E2H,TGE> == '11'
    && SCR_EL3.NS == '1' then
            X[t, 64] = CNTHV_CVAL_EL2;
        else
            X[t, 64] = CNTV_CVAL_EL0;
    elsif PSTATE.EL == EL1 then
        if EL2Enabled() && CNTHCTL_EL2.EL1TVT == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif EL2Enabled() && HCR_EL2.<NV2,NV1,NV> ==
    '111' then
            X[t, 64] = NVMem[0x168];
        else
            X[t, 64] = CNTV_CVAL_EL0;
    elsif PSTATE.EL == EL2 then
        if HCR_EL2.E2H == '1' && SCR_EL3.NS == '0' &&
    IsFeatureImplemented(FEAT_SEL2) then
            X[t, 64] = CNTHVS_CVAL_EL2;
        elsif HCR_EL2.E2H == '1' && SCR_EL3.NS == '1'

```

```
then
    X[t, 64] = CNTHV_CVAL_EL2;
else
    X[t, 64] = CNTV_CVAL_EL0;
elsif PSTATE.EL == EL3 then
    X[t, 64] = CNTV_CVAL_EL0;
```

MSR CNTV_CVAL_EL0, <Xt>

op0	op1	CRn	CRm	op2
0b11	0b011	0b1110	0b0011	0b010

```

if PSTATE.EL == EL0 then
    if !(EL2Enabled() && HCR_EL2.<E2H,TGE> == '11')
    && CNTKCTL_EL1.EL0VTEN == '0' then
        if EL2Enabled() && HCR_EL2.TGE == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        else
            AArch64.SystemAccessTrap(EL1, 0x18);
        elsif EL2Enabled() && HCR_EL2.<E2H,TGE> == '11'
    && CNTHCTL_EL2.EL0VTEN == '0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
        elsif EL2Enabled() && HCR_EL2.<E2H,TGE> != '11'
    && CNTHCTL_EL2.EL1TVT == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
        elsif EL2Enabled() && HCR_EL2.<E2H,TGE> == '11'
    && SCR_EL3.NS == '0' &&
    IsFeatureImplemented(FEAT_SEL2) then
        CNTHVS_CVAL_EL2 = X[t, 64];
        elsif EL2Enabled() && HCR_EL2.<E2H,TGE> == '11'
    && SCR_EL3.NS == '1' then
        CNTHV_CVAL_EL2 = X[t, 64];
        else
            CNTV_CVAL_EL0 = X[t, 64];
    elsif PSTATE.EL == EL1 then
        if EL2Enabled() && CNTHCTL_EL2.EL1TVT == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif EL2Enabled() && HCR_EL2.<NV2,NV1,NV> ==
    '111' then
            NVMem[0x168] = X[t, 64];
        else
            CNTV_CVAL_EL0 = X[t, 64];
    elsif PSTATE.EL == EL2 then
        if HCR_EL2.E2H == '1' && SCR_EL3.NS == '0' &&
    IsFeatureImplemented(FEAT_SEL2) then
            CNTHVS_CVAL_EL2 = X[t, 64];
        elsif HCR_EL2.E2H == '1' && SCR_EL3.NS == '1'
    then
            CNTHV_CVAL_EL2 = X[t, 64];
        else
            CNTV_CVAL_EL0 = X[t, 64];
    elsif PSTATE.EL == EL3 then
        CNTV_CVAL_EL0 = X[t, 64];

```

[AArch32
Registers](#)

[AArch64
Registers](#)

[AArch32
Instructions](#)

[AArch64
Instructions](#)

[Index by
Encoding](#)

[External
Registers](#)

28/03/2023 16:02; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94

Copyright Â© 2010-2023 Arm Limited or its affiliates. All rights reserved. This document is Non-Confidential.