

## AMCNTENCLR0\_EL0, Activity Monitors Count Enable Clear Register 0

The AMCNTENCLR0\_EL0 characteristics are:

### Purpose

Disable control bits for the architected activity monitors event counters, [AMEVCNTR0<n>\\_EL0](#).

### Configuration

AArch64 System register AMCNTENCLR0\_EL0 bits [31:0] are architecturally mapped to AArch32 System register [AMCNTENCLR0\[31:0\]](#).

AArch64 System register AMCNTENCLR0\_EL0 bits [31:0] are architecturally mapped to External register [AMCNTENCLR0\[31:0\]](#).

This register is present only when FEAT\_AMUv1 is implemented. Otherwise, direct accesses to AMCNTENCLR0\_EL0 are undefined.

### Attributes

AMCNTENCLR0\_EL0 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
RES0																															
RES0																RAZ/WI												P3	P2	P1	P0
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

#### Bits [63:16]

Reserved, res0.

#### Bits [15:4]

Reserved, RAZ/WI.

This field is reserved for additional architected activity monitor event counters, which Arm might define in a future version of the Activity Monitors architecture.

## P<n>, bit [n], for n = 3 to 0

Activity monitor event counter disable bit for [AMEVCNTR0<n>\\_EL0](#).

### Note

[AMCGCR\\_EL0.CG0NC](#) identifies the number of architected activity monitor event counters. In an implementation that includes FEAT\_AMUv1, the number of architected activity monitor event counters is 4.

Possible values of each bit are:

P<n>	Meaning
0b0	When read, means that <a href="#">AMEVCNTR0&lt;n&gt;_EL0</a> is disabled. When written, has no effect.
0b1	When read, means that <a href="#">AMEVCNTR0&lt;n&gt;_EL0</a> is enabled. When written, disables <a href="#">AMEVCNTR0&lt;n&gt;_EL0</a> .

The reset behavior of this field is:

- On an AMU reset, this field resets to 0.

## Accessing AMCNTENCLR0\_EL0

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

### MRS <Xt>, AMCNTENCLR0\_EL0

op0	op1	CRn	CRm	op2
0b11	0b011	0b1101	0b0010	0b100

```
if PSTATE.EL == EL0 then
    if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
    && boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1'" && CPTR_EL3.TAM == '1' then
    UNDEFINED;
elseif AMUSERENR_EL0.EN == '0' then
    if EL2Enabled() && HCR_EL2.TGE == '1' then
```

```

        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        AArch64.SystemAccessTrap(EL1, 0x18);
    elsif EL2Enabled() && CPTR_EL2.TAM == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && HCR_EL2.<E2H,TGE> != '11'
&& IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3)
|| SCR_EL3.FGTEn == '1') && HAFGRTR_EL2.AMCNTEN0 ==
'1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif HaveEL(EL3) && CPTR_EL3.TAM == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = AMCNTENCLR0_EL0;
    elsif PSTATE.EL == EL1 then
        if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
&& boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1'" && CPTR_EL3.TAM == '1' then
            UNDEFINED;
        elsif EL2Enabled() && CPTR_EL2.TAM == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif EL2Enabled() &&
IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3) ||
SCR_EL3.FGTEn == '1') && HAFGRTR_EL2.AMCNTEN0 == '1'
then
            AArch64.SystemAccessTrap(EL2, 0x18);
        elsif HaveEL(EL3) && CPTR_EL3.TAM == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = AMCNTENCLR0_EL0;
    elsif PSTATE.EL == EL2 then
        if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
&& boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1'" && CPTR_EL3.TAM == '1' then
            UNDEFINED;
        elsif HaveEL(EL3) && CPTR_EL3.TAM == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                X[t, 64] = AMCNTENCLR0_EL0;
    elsif PSTATE.EL == EL3 then
        X[t, 64] = AMCNTENCLR0_EL0;

```

## MSR AMCNTENCLR0\_EL0, <Xt>

op0	op1	CRn	CRm	op2
-----	-----	-----	-----	-----

0b11	0b011	0b1101	0b0010	0b100
------	-------	--------	--------	-------

```

if IsHighestEL(PSTATE.EL) then
    AMCNTENCLR0_EL0 = X[t, 64];
else
    UNDEFINED;

```

[AArch32  
Registers](#)

[AArch64  
Registers](#)

[AArch32  
Instructions](#)

[AArch64  
Instructions](#)

[Index by  
Encoding](#)

[External  
Registers](#)

28/03/2023 16:02; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94

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