

## AMEVCNTR0<n>\_EL0, Activity Monitors Event Counter Registers 0, n = 0 - 3

The AMEVCNTR0<n>\_EL0 characteristics are:

### Purpose

Provides access to the architected activity monitor event counters.

### Configuration

AArch64 System register AMEVCNTR0<n>\_EL0 bits [63:0] are architecturally mapped to AArch32 System register [AMEVCNTR0<n>\[63:0\]](#).

AArch64 System register AMEVCNTR0<n>\_EL0 bits [63:0] are architecturally mapped to External register [AMEVCNTR0<n>\[63:0\]](#).

This register is present only when FEAT\_AMUv1 is implemented. Otherwise, direct accesses to AMEVCNTR0<n>\_EL0 are undefined.

### Attributes

AMEVCNTR0<n>\_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
ACNT																															
ACNT																															
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

### ACNT, bits [63:0]

Architected activity monitor event counter n.

Value of architected activity monitor event counter n, where n is the number of this register and is a number from 0 to 3.

If FEAT\_AMUv1p1 is implemented, [HCR\\_EL2](#).AMVOFFEN is 1, [SCR\\_EL3](#).AMVOFFEN is 1, [HCR\\_EL2](#).{E2H, TGE} is not {1,1}, and EL2 is implemented in the current Security state, access to these registers at EL0 or EL1 return (PCount<63:0> - [AMEVCNTVOFF0<n>\\_EL2](#)<63:0>).

PCount is the physical count returned when AMEVCNTR0<n>\_EL0 is read from EL2 or EL3.

If the counter is enabled, writes to this register have unpredictable results.

The reset behavior of this field is:

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

## Accessing AMEVCNTR0<n>\_EL0

If <n> is greater than or equal to the number of architected activity monitor event counters, reads and writes of AMEVCNTR0<n>\_EL0 are undefined.

---

### Note

[AMCGCR\\_EL0.CG0NC](#) identifies the number of architected activity monitor event counters.

---

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

**MRS <Xt>, AMEVCNTR0<m>\_EL0 ; Where m = 0-3**

op0	op1	CRn	CRm	op2
0b11	0b011	0b1101	0b010:m[3]	m[2:0]

```
integer m = UInt(CRm<0>:op2<2:0>);

if m >= 4 then
    UNDEFINED;
elsif 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;
    elsif 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.AMEVCNTR0<m>_EL0 == '1' then
    AArch64.SystemAccessTrap(EL2, 0x18);
elseif HaveEL(EL3) && CPTR_EL3.TAM == '1' then
    if Halted() && EDSCR.SDD == '1' then
        UNDEFINED;
    else
        AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = AMEVCNTR0_EL0[m];
elseif 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;
    elseif EL2Enabled() && CPTR_EL2.TAM == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() &&
IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3) ||
SCR_EL3.FGTEn == '1') &&
HAFGRTR_EL2.AMEVCNTR0<m>_EL0 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif HaveEL(EL3) && CPTR_EL3.TAM == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = AMEVCNTR0_EL0[m];
elseif 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;
    elseif HaveEL(EL3) && CPTR_EL3.TAM == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = AMEVCNTR0_EL0[m];
elseif PSTATE.EL == EL3 then
    X[t, 64] = AMEVCNTR0_EL0[m];

```

**MSR AMEVCNTR0<m>\_EL0, <Xt> ; Where m = 0-3**

op0	op1	CRn	CRm	op2
0b11	0b011	0b1101	0b010:m[3]	m[2:0]

```
integer m = UInt(CRm<0>:op2<2:0>);
```

```
if m >= 4 then
    UNDEFINED;
elsif IsHighestEL(PSTATE.EL) then
    AMEVCNTR0_EL0[m] = 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.