

PMINTENCLR_EL1, Performance Monitors Interrupt Enable Clear Register

The PMINTENCLR_EL1 characteristics are:

Purpose

Disables the generation of interrupt requests on overflows from the Cycle Count Register, PMU.PMCCNTR_EL0, and the event counters PMU.PMEVCNTR<n>_EL0. Reading the register shows which overflow interrupt requests are enabled.

Configuration

External register PMINTENCLR_EL1 bits [31:0] are architecturally mapped to AArch64 System register [PMINTENCLR_EL1\[31:0\]](#) when FEAT_PMUv3_EXT32 is implemented, FEAT_PMUv3p9 is not implemented and FEAT_PMUv3_ICNTR is not implemented.

External register PMINTENCLR_EL1 bits [31:0] are architecturally mapped to AArch64 System register [PMINTENSET_EL1\[31:0\]](#) when FEAT_PMUv3_EXT32 is implemented, FEAT_PMUv3p9 is not implemented and FEAT_PMUv3_ICNTR is not implemented.

External register PMINTENCLR_EL1 bits [63:0] are architecturally mapped to AArch64 System register [PMINTENCLR_EL1\[63:0\]](#) when FEAT_PMUv3_EXT64 is implemented, or FEAT_PMUv3p9 is implemented or FEAT_PMUv3_ICNTR is implemented.

External register PMINTENCLR_EL1 bits [63:0] are architecturally mapped to AArch64 System register [PMINTENSET_EL1\[63:0\]](#) when FEAT_PMUv3_EXT64 is implemented, or FEAT_PMUv3p9 is implemented or FEAT_PMUv3_ICNTR is implemented.

External register PMINTENCLR_EL1 bits [31:0] are architecturally mapped to AArch32 System register [PMINTENCLR\[31:0\]](#).

External register PMINTENCLR_EL1 bits [31:0] are architecturally mapped to AArch32 System register [PMINTENSET\[31:0\]](#).

This register is present only when FEAT_PMUv3_EXT is implemented. Otherwise, direct accesses to PMINTENCLR_EL1 are res0.

PMINTENCLR_EL1 is in the Core power domain.

Attributes

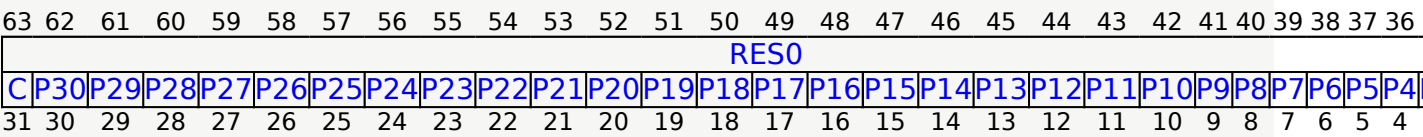
PMINTENCLR_EL1 is a:

- 64-bit register when FEAT_PMUv3_EXT64 is implemented, or FEAT_PMUv3p9 is implemented or FEAT_PMUv3_ICNTR is implemented
- 32-bit register otherwise

This register is part of the [PMU](#) block.

Field descriptions

When FEAT_PMUv3_EXT64 is implemented, or FEAT_PMUv3p9 is implemented or FEAT_PMUv3_ICNTR is implemented:



Bits [63:33]

Reserved, res0.

F0, bit [32]

When FEAT_PMUv3_ICNTR is implemented:

Interrupt request on unsigned overflow of PMU.PMICNTR_EL0 disable. On writes, allows software to disable the interrupt request on unsigned overflow of PMU.PMICNTR_EL0. On reads, returns the interrupt request on unsigned overflow of PMU.PMICNTR_EL0 enable status.

F0	Meaning
0b0	Interrupt request on unsigned overflow of PMU.PMICNTR_EL0 disabled.
0b1	Interrupt request on unsigned overflow of PMU.PMICNTR_EL0 enabled.

The reset behavior of this field is:

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

Accessing this field has the following behavior:

- When SoftwareLockStatus(), access to this field is **RO**.
- Otherwise, access to this field is **W1C**.

Otherwise:

Reserved, res0.

C, bit [31]

Interrupt request or PMU exception on unsigned overflow of PMU.PMCCNTR_EL0 disable. On writes, allows software to disable the interrupt request or PMU exception on unsigned overflow of PMU.PMCCNTR_EL0. On reads, returns the interrupt request or PMU exception on unsigned overflow of PMU.PMCCNTR_EL0 enable status.

C	Meaning
0b0	Interrupt request or PMU exception on unsigned overflow of PMU.PMCCNTR_EL0 disabled.
0b1	Interrupt request or PMU exception on unsigned overflow of PMU.PMCCNTR_EL0 enabled.

The reset behavior of this field is:

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

Accessing this field has the following behavior:

- When SoftwareLockStatus(), access to this field is **RO**.
- Otherwise, access to this field is **W1C**.

P<m>, bit [m], for m = 30 to 0

Interrupt request or PMU exception on unsigned overflow of [PMEVCNTR<m>_EL0](#) disable. On writes, allows software to disable the interrupt request or PMU exception on unsigned overflow of [PMEVCNTR<m>_EL0](#). On reads, returns the interrupt request or PMU exception on unsigned overflow of [PMEVCNTR<m>_EL0](#) enable status.

P<m>	Meaning
0b0	Interrupt request or PMU exception on unsigned overflow of PMEVCNTR<m>_EL0 disabled.
0b1	Interrupt request or PMU exception on unsigned overflow of PMEVCNTR<m>_EL0 enabled.

The reset behavior of this field is:

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

Accessing this field has the following behavior:

- When $m \geq \text{NUM_PMU_COUNTERS}$, access to this field is **RAZ/WI**.
- When `SoftwareLockStatus()`, access to this field is **RO**.
- Otherwise, access to this field is **W1C**.

Otherwise:

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
C	P30	P29	P28	P27	P26	P25	P24	P23	P22	P21	P20	P19	P18	P17	P16	P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4

C, bit [31]

Interrupt request or PMU exception on unsigned overflow of `PMU.PMCCNTR_EL0` disable. On writes, allows software to disable the interrupt request or PMU exception on unsigned overflow of `PMU.PMCCNTR_EL0`. On reads, returns the interrupt request or PMU exception on unsigned overflow of `PMU.PMCCNTR_EL0` enable status.

C	Meaning
0b0	Interrupt request or PMU exception on unsigned overflow of <code>PMU.PMCCNTR_EL0</code> disabled.
0b1	Interrupt request or PMU exception on unsigned overflow of <code>PMU.PMCCNTR_EL0</code> enabled.

The reset behavior of this field is:

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

Accessing this field has the following behavior:

- When `SoftwareLockStatus()`, access to this field is **RO**.
- Otherwise, access to this field is **W1C**.

P<m>, bit [m], for m = 30 to 0

Interrupt request or PMU exception on unsigned overflow of [`PMEVCNTR<m>_EL0`](#) disable. On writes, allows software to disable the interrupt request or PMU exception on unsigned overflow of [`PMEVCNTR<m>_EL0`](#). On reads, returns the interrupt request or PMU exception on unsigned overflow of [`PMEVCNTR<m>_EL0`](#) enable status.

P<m>	Meaning
0b0	Interrupt request or PMU exception on unsigned overflow of PMEVCNTR<m>_ELO disabled.
0b1	Interrupt request or PMU exception on unsigned overflow of PMEVCNTR<m>_ELO enabled.

The reset behavior of this field is:

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

Accessing this field has the following behavior:

- When $m \geq \text{NUM_PMU_COUNTERS}$, access to this field is **RAZ/WI**.
- When `SoftwareLockStatus()`, access to this field is **RO**.
- Otherwise, access to this field is **W1C**.

Accessing PMINTENCLR_EL1

Note

`SoftwareLockStatus()` depends on the type of access attempted and `AllowExternalPMUAccess()` has a new definition from Armv8.4. Refer to the Pseudocode definitions for more information.

Accesses to this register use the following encodings:

When FEAT_PMUv3_EXT64 is implemented, or FEAT_PMUv3_ICNTR is implemented or FEAT_PMUv3p9 is implemented
[63:0] Accessible at offset 0xC60 from PMU

- When `DoubleLockStatus()`, or `!IsCorePowered()`, or `OSLockStatus()` or `!AllowExternalPMUAccess()`, accesses to this register generate an error response.
- When `FEAT_PMUv3_EXT32` is implemented and `SoftwareLockStatus()`, accesses to this register are **RO**.
- Otherwise, accesses to this register are **RW**.

When FEAT_PMUv3_EXT32 is implemented, FEAT_PMUv3_ICNTR is not implemented and FEAT_PMUv3p9 is not implemented

[31:0] Accessible at offset 0xC60 from PMU

- When DoubleLockStatus(), or !IsCorePowered(), or OSLockStatus() or !AllowExternalPMUAccess(), accesses to this register generate an error response.
- When SoftwareLockStatus(), accesses to this register are **RO**.
- Otherwise, accesses to this register are **RW**.

[AArch32
Registers](#)

[AArch64
Registers](#)

[AArch32
Instructions](#)

[AArch64
Instructions](#)

[Index by
Encoding](#)

[External
Registers](#)

28/03/2023 16:01; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94

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