PMCCFILTR_EL0, Performance Monitors Cycle Counter Filter Register

The PMCCFILTR_EL0 characteristics are:

Purpose

Determines the modes in which the Cycle Counter, PMU.PMCCNTR_EL0, increments.

Configuration

External register PMCCFILTR_EL0 bits [31:0] are architecturally mapped to AArch64 System register PMCCFILTR_EL0[31:0] when FEAT_PMUv3_EXT32 is implemented.

External register PMCCFILTR_EL0 bits [63:0] are architecturally mapped to AArch64 System register PMCCFILTR_EL0[63:0] when FEAT_PMUv3_EXT64 is implemented.

External register PMCCFILTR_EL0 bits [31:0] are architecturally mapped to AArch32 System register PMCCFILTR[31:0].

This register is present only when FEAT_PMUv3_EXT is implemented. Otherwise, direct accesses to PMCCFILTR_EL0 are res0.

PMCCFILTR_EL0 is in the Core power domain.

On a Warm or Cold reset, RW fields in this register reset to:

- Architecturally unknown values if the reset is to an Exception level that is using AArch64.
- 0 if the reset is to an Exception level that is using AArch32.

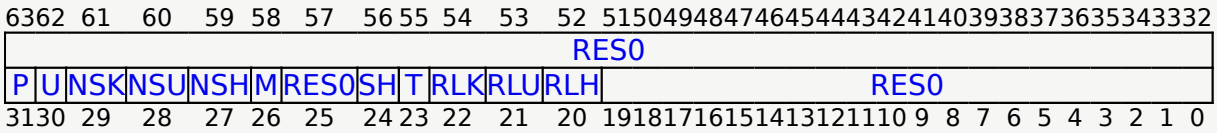
The register is not affected by an External debug reset.

Attributes

PMCCFILTR_EL0 is a 64-bit register.

This register is part of the PMU block.

Field descriptions



Bits [63:32]

Reserved, res0.

P, bit [31]

EL1 filtering. Controls counting cycles in EL1.

P	Meaning
0b0	This field has no effect on filtering of cycles.
0b1	Cycles in EL1 are not counted.

If Secure and Non-secure states are implemented, then counting cycles in Non-secure EL1 is further controlled by PMCCFILTR_EL0.NSK.

If FEAT_RME is implemented, then counting cycles in Realm EL1 is further controlled by PMCCFILTR_EL0.RLK.

If EL3 is implemented, then counting cycles in EL3 is further controlled by PMCCFILTR_EL0.M.

The reset behavior of this field is:

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

U, bit [30]

EL0 filtering. Controls counting cycles in EL0.

U	Meaning
0b0	This field has no effect on filtering of cycles.
0b1	Cycles in EL0 are not counted.

If Secure and Non-secure states are implemented, then counting cycles in Non-secure EL0 is further controlled by PMCCFILTR_EL0.NSU.

If FEAT_RME is implemented, then counting cycles in Realm EL0 is further controlled by PMCCFILTR_EL0.RLU.

The reset behavior of this field is:

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

NSK, bit [29]
When EL3 is implemented:

Non-secure EL1 filtering. Controls counting cycles in Non-secure EL1. If PMCCFILTR_EL0.NSK is not equal to PMCCFILTR_EL0.P, then cycles in Non-secure EL1 are not counted. Otherwise, PMCCFILTR_EL0.NSK has no effect on filtering of cycles in Non-secure EL1.

NSK	Meaning
0b0	When PMCCFILTR_EL0.P == 0, this field has no effect on filtering of cycles. When PMCCFILTR_EL0.P == 1, cycles in Non-secure EL1 are not counted.
0b1	When PMCCFILTR_EL0.P == 0, cycles in Non-secure EL1 are not counted. When PMCCFILTR_EL0.P == 1, this field has no effect on filtering of cycles.

The reset behavior of this field is:

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

Otherwise:

Reserved, res0.

NSU, bit [28]
When EL3 is implemented:

Non-secure EL0 filtering. Controls counting cycles in Non-secure EL0. If PMCCFILTR_EL0.NSU is not equal to PMCCFILTR_EL0.U, then cycles in Non-secure EL0 are not counted. Otherwise, PMCCFILTR_EL0.NSU has no effect on filtering of cycles in Non-secure EL0.

NSU	Meaning

0b0	When PMCCFILTR_EL0.U == 0, this field has no effect on filtering of cycles. When PMCCFILTR_EL0.U == 1, cycles in Non-secure EL0 are not counted.
0b1	When PMCCFILTR_EL0.U == 0, cycles in Non-secure EL0 are not counted. When PMCCFILTR_EL0.U == 1, this field has no effect on filtering of cycles.

The reset behavior of this field is:

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

Otherwise:

Reserved, res0.

NSH, bit [27]
When EL2 is implemented:

EL2 filtering. Controls counting cycles in EL2.

NSH	Meaning
0b0	Cycles in EL2 are not counted.
0b1	This field has no effect on filtering of cycles.

If EL3 is implemented and FEAT_SEL2 is implemented, then counting cycles in Secure EL2 is further controlled by PMCCFILTR_EL0.SH.

If FEAT_RME is implemented, then counting cycles in Realm EL2 is further controlled by PMCCFILTR_EL0.RLH.

The reset behavior of this field is:

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

Otherwise:

Reserved, res0.

M, bit [26]
When EL3 is implemented:

EL3 filtering. Controls counting cycles in EL3. If PMCCFILTR_EL0.M is not equal to PMCCFILTR_EL0.P, then cycles in EL3 are not counted. Otherwise, PMCCFILTR_EL0.M has no effect on filtering of cycles in EL3.

M	Meaning
0b0	When PMCCFILTR_EL0.P == 0, this field has no effect on filtering of cycles. When PMCCFILTR_EL0.P == 1, cycles in EL3 are not counted.
0b1	When PMCCFILTR_EL0.P == 0, cycles in EL3 are not counted. When PMCCFILTR_EL0.P == 1, this field has no effect on filtering of cycles.

The reset behavior of this field is:

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

Otherwise:

Reserved, res0.

Bit [25]

Reserved, res0.

SH, bit [24]
When EL3 is implemented and FEAT_SEL2 is implemented:

Secure EL2 filtering. Controls counting cycles in Secure EL2. If PMCCFILTR_EL0.SH is equal to PMCCFILTR_EL0.NSH, then cycles in Secure EL2 are not counted. Otherwise, PMCCFILTR_EL0.SH has no effect on filtering of cycles in Secure EL2.

SH	Meaning
0b0	When PMCCFILTR_EL0.NSH == 0, cycles in Secure EL2 are not counted. When PMCCFILTR_EL0.NSH == 1, this field has no effect on filtering of cycles.

0b1	When PMCCFILTR_EL0.NSH == 0, this field has no effect on filtering of cycles. When PMCCFILTR_EL0.NSH == 1, cycles in Secure EL2 are not counted.

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 !IsSecureEL2Enabled(), access to this field is **RES0**.
- Otherwise, access to this field is **RW**.

Otherwise:

Reserved, res0.

T, bit [23]
When FEAT_TME is implemented:

Transactional state filtering bit. Controls counting of Attributable events in Non-transactional state.

T	Meaning
0b0	This bit has no effect on the filtering of events.
0b1	Do not count Attributable events in Non-transactional state.

For each Unattributable event, it is implementation defined whether the filtering applies.

The reset behavior of this field is:

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

Otherwise:

Reserved, res0.

RLK, bit [22]
When FEAT_RME is implemented:

Realm EL1 (kernel) filtering bit. Controls counting in Realm EL1.

If the value of this bit is equal to the value of the PMCCFILTR_EL0.P bit, cycles in Realm EL1 are counted.

Otherwise, cycles in Realm EL1 are not counted.

The reset behavior of this field is:

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

Otherwise:

Reserved, res0.

RLU, bit [21]
When FEAT_RME is implemented:

Realm EL0 (unprivileged) filtering bit. Controls counting in Realm EL0.

If the value of this bit is equal to the value of the PMCCFILTR_EL0.U bit, cycles in Realm EL0 are counted.

Otherwise, cycles in Realm EL0 are not counted.

The reset behavior of this field is:

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

Otherwise:

Reserved, res0.

RLH, bit [20]
When FEAT_RME is implemented:

Realm EL2 filtering bit. Controls counting in Realm EL2.

If the value of this bit is not equal to the value of the PMCCFILTR_EL0.NSH bit, cycles in Realm EL2 are counted.

Otherwise, cycles in Realm EL2 are not counted.

The reset behavior of this field is:

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

Otherwise:

Reserved, res0.

Bits [19:0]

Reserved, res0.

Accessing PMCCFILTR_EL0

If FEAT_PMUv3_EXT32 is implemented, and at least one of FEAT_PMUv3_TH or FEAT_PMUv3p8 is implemented, then bits [63:32] of this register are accessible at offset `0xA7C`. Otherwise accesses at this offset are implementation defined.

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_EXT32 is implemented

[31:0] Accessible at offset 0x47C 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**.

When FEAT_PMUv3_EXT64 is implemented

Accessible at offset 0x4F8 from PMU

- When DoubleLockStatus(), or !IsCorePowered(), or OSLockStatus() or !AllowExternalPMUAccess(), accesses to this register generate an error response.

- Otherwise, accesses to this register are **RW**.

When FEAT_PMUv3_EXT32 is implemented and (FEAT_PMUv3_TH is implemented or FEAT_PMUv3p8 is implemented)

[63:32] Accessible at offset 0xA7C 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**.

28/03/2023 16:01; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94