PMCR_EL0, Performance Monitors Control Register

The PMCR_EL0 characteristics are:

Purpose

Provides details of the Performance Monitors implementation, including the number of counters implemented, and configures and controls the counters.

Configuration

External register PMCR_EL0 bits [7:0] are architecturally mapped to AArch64 System register PMCR_EL0[7:0].

External register PMCR_EL0 bits [7:0] are architecturally mapped to AArch32 System register PMCR[7:0].

This register is present only when FEAT_PMUv3_EXT is implemented. Otherwise, direct accesses to PMCR_EL0 are res0.

PMCR_EL0 is in the Core power domain.

This register is only partially mapped to the internal PMCR System register. An external agent must use other means to discover the information held in PMCR[31:11], such as accessing PMU.PMCFGR and the ID registers.

Attributes

PMCR_EL0 is a:

- 64-bit register when FEAT_PMUv3_EXT64 is implemented
- 32-bit register otherwise

This register is part of the PMU block.

Field descriptions

When FEAT_PMUv3_EXT64 is implemented:

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																														FZS	
RAZ/WI																					RES0		FZO	RES0		LP		LC	DP	X	D	C	P	E
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 [63:33]

Reserved, res0.

FZS, bit [32]
When FEAT_SPEv1p2 is implemented:

Freeze-on-SPE event. Stop counters when PMBLIMITR_EL1. {PMFZ,E} == {1,1} and PMBSR_EL1.S == 1.

In the description of this field:

- If EL2 is implemented and is using AArch32, PMN is HDCR.HPMN.
- If EL2 is implemented and is using AArch64, PMN is MDCR_EL2.HPMN.
- If EL2 is not implemented, PMN is PMCR_EL0.N.

FZS	Meaning
0b0	Do not freeze on Statistical Profiling Buffer Management event.
0b1	Event counter PMEVCNTR<n>_EL0 does not count following a Statistical Profiling Buffer Management event if n is in the range of affected event counters.

If PMN is not 0, this field affects the operation of event counters in the range [0 .. (PMN-1)].

This field does not affect the operation of other event counters and PMCCNTR_EL0.

The operation of this field applies even when EL2 is disabled in the current Security state.

The reset behavior of this field is:

- On a Warm reset:
  - When AArch32 is supported, this field resets to 0.
  - When the implementation only supports execution in AArch64 state, this field resets to an architecturally unknown value.

Otherwise:

Reserved, res0.

Bits [31:11]

Reserved, RAZ/WI.

Hardware must implement this field as RAZ/WI. Software must not rely on the register reading as zero, and must use a read-modify-write sequence to write to the register.

Bit [10]

Reserved, res0.

FZO, bit [9]
When FEAT_PMUv3p7 is implemented:

Freeze-on-overflow. Stop event counters on overflow.

In the description of this field:

- If EL2 is implemented and is using AArch32, PMN is HDCR.HPMN.
- If EL2 is implemented and is using AArch64, PMN is MDCR_EL2.HPMN.
- If EL2 is not implemented, PMN is PMCR_EL0.N.

FZO	Meaning
0b0	Do not freeze on overflow.
0b1	Event counter PMU.PMEVCNTR<n>_EL0 does not count when PMOVSCLR_EL0[(PMN-1):0] is nonzero and n is in the range of affected event counters.

If PMN is not 0, this field affects the operation of event counters in the range [0 .. (PMN-1)].

This field does not affect the operation of other event counters and PMU.PMCCNTR_EL0.

The operation of this field applies even when EL2 is disabled in the current Security state.

The reset behavior of this field is:

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

Otherwise:

Reserved, res0.

Bit [8]

Reserved, res0.

LP, bit [7]
When FEAT_PMUv3p5 is implemented:

Long event counter enable. Determines when unsigned overflow is recorded by an event counter overflow bit.

In the description of this field:

- If EL2 is implemented and is using AArch32, PMN is HDCR.HPMN.
- If EL2 is implemented and is using AArch64, PMN is MDCR_EL2.HPMN.
- If EL2 is not implemented, PMN is PMCR_EL0.N.

LP	Meaning
0b0	Event counter overflow on increment that causes unsigned overflow of PMU.PMEVCNTR<n>_EL0[31:0].
0b1	Event counter overflow on increment that causes unsigned overflow of PMU.PMEVCNTR<n>_EL0[63:0].

If PMN is not 0, this bit affects the operation of event counters in the range [0 .. (PMN-1)].

The field does not affect the operation of other event counters and PMU.PMCCNTR_EL0.

The operation of this field applies even when EL2 is disabled in the current Security state.

The reset behavior of this field is:

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

Otherwise:

Reserved, res0.

LC, bit [6]
When AArch32 is supported:

Long cycle counter enable. Determines when unsigned overflow is recorded by the cycle counter overflow bit.

LC	Meaning
0b0	Cycle counter overflow on increment that causes unsigned overflow of PMU.PMCCNTR_EL0[31:0].
0b1	Cycle counter overflow on increment that causes unsigned overflow of PMU.PMCCNTR_EL0[63:0].

Arm deprecates use of PMU.PMCR_EL0.LC = 0.

The reset behavior of this field is:

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

Otherwise:

Reserved, res1.

DP, bit [5]
When EL3 is implemented or (FEAT_PMUv3p1 is implemented and EL2 is implemented):

Disable cycle counter when event counting is prohibited. The possible values of this bit are:

DP	Meaning
0b0	Cycle counting by PMU.PMCCNTR_EL0 is not affected by this mechanism.

0b1	Cycle counting by PMU.PMCCNTR_EL0 is disabled in prohibited regions and when event counting is frozen: • If FEAT_PMUv3p1 is implemented, EL2 is implemented, and MDCR_EL2.HPMD is 1, then cycle counting by PMU.PMCCNTR_EL0 is disabled at EL2. • If FEAT_PMUv3p7 is implemented, EL3 is implemented and using AArch64, and MDCR_EL3.MPMX is 1, then cycle counting by PMU.PMCCNTR_EL0 is disabled at EL3. • If FEAT_PMUv3p7 is implemented and event counting is frozen by PMCR_EL0.FZO, then cycle counting by PMU.PMCCNTR_EL0 is disabled. • If EL3 is implemented, MDCR_EL3.SPME is 0, and either FEAT_PMUv3p7 is not implemented or MDCR_EL3.MPMX is 0, then cycle counting by PMU.PMCCNTR_EL0 is disabled at EL3 and in Secure state. If MDCR_EL2.HPMN is not 0, this is when event counting by event counters in the range [0..(MDCR_EL2.HPMN-1)] is prohibited or frozen.

For more information, see 'Prohibiting event and cycle counting'.

The reset behavior of this field is:

- On a Warm reset:
  - When the implementation only supports execution in AArch32 state, this field resets to `0`.
  - Otherwise, this field resets to an architecturally unknown value.

Otherwise:

Reserved, res0.

X, bit [4]
When the implementation includes a PMU event export bus:

Enable export of events in an implementation defined PMU event export bus.

X	Meaning
0b0	Do not export events.
0b1	Export events where not prohibited.

This field enables the exporting of events over an implementation defined PMU event export bus to another device, for example to an optional trace unit.

No events are exported when counting is prohibited.

This field does not affect the generation of Performance Monitors overflow interrupt requests or signaling to a cross-trigger interface (CTI) that can be implemented as signals exported from the PE.

When this register has an architecturally-defined reset value, if this field is implemented as an RW field it resets to:

- A value that is architecturally unknown if the reset is into an Exception level that is using AArch64.
- 0 if the reset is into an Exception level that is using AArch32.

Otherwise:

Reserved, RAZ/WI.

D, bit [3]
When AArch32 is supported:

Clock divider.

D	Meaning
0b0	When enabled, PMU.PMCCNTR_EL0 counts every clock cycle.
0b1	When enabled, PMU.PMCCNTR_EL0 counts once every 64 clock cycles.

If PMCR_EL0.LC == 1, this bit is ignored and the cycle counter counts every clock cycle.

Arm deprecates use of PMCR_EL0.D = 1.

When this register has an architecturally-defined reset value, if this field is implemented as an RW field it resets to:

- A value that is architecturally unknown if the reset is into an Exception level that is using AArch64.
- 0 if the reset is into an Exception level that is using AArch32.

Otherwise:

Reserved, res0.

C, bit [2]

Cycle counter reset. The effects of writing to this bit are:

C	Meaning
0b0	No action.
0b1	Reset PMU.PMCCNTR_EL0 to zero.

Note

Resetting PMU.PMCCNTR_EL0 does not change the cycle counter overflow bit. If FEAT_PMUv3p5 is implemented, the value of PMCR_EL0.LC is ignored, and bits [63:0] of the cycle counter are reset.

Access to this field is **WO/RAZ**.

P, bit [1]

Event counter reset. The effects of writing to this bit are:

P	Meaning
0b0	No action.
0b1	Reset all event counters, not including PMU.PMCCNTR_EL0, to zero.

Note

Resetting the event counters does not change the event counter overflow bits. If FEAT_PMUv3p5 is implemented, the value of MDCR_EL2.HLP, or PMCR_EL0.LP is

ignored and bits [63:0] of all affected event counters are reset.

Access to this field is **WO/RAZ**.

E, bit [0]

Enable.

In the description of this field:

- If EL2 is implemented and is using AArch32, PMN is HDCR.HPMN.
- If EL2 is implemented and is using AArch64, PMN is MDCR_EL2.HPMN.
- If EL2 is not implemented, PMN is PMCR_EL0.N.

E	Meaning
0b0	PMU.PMCCNTR_EL0 is disabled and event counters PMU.PMEVCNTR<n>_EL0, where n is in the range of affected event counters, are disabled.
0b1	PMU.PMCCNTR_EL0 and event counters PMU.PMEVCNTR<n>_EL0, where n is in the range of affected event counters, are enabled by PMU.PMCNTENSET_EL0.

If PMN is not 0, this field affects the operation of event counters in the range [0 .. (PMN-1)].

This field does not affect the operation of other event counters.

The operation of this field applies even when EL2 is disabled in the current Security state.

The reset behavior of this field is:

- On a Warm reset, this field resets to 0.

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	3	2	1	0
RAZ/WI																					RES0	FZO	RES0	LP	LC	DP	X	D	C	P	E

Bits [31:11]

Reserved, RAZ/WI.

Hardware must implement this field as RAZ/WI. Software must not rely on the register reading as zero, and must use a read-modify-write sequence to write to the register.

Bit [10]

Reserved, res0.

FZO, bit [9]
When FEAT_PMUv3p7 is implemented:

Freeze-on-overflow. Stop event counters on overflow.

In the description of this field:

- If EL2 is implemented and is using AArch32, PMN is HDCR.HPMN.
- If EL2 is implemented and is using AArch64, PMN is MDCR_EL2.HPMN.
- If EL2 is not implemented, PMN is PMCR_EL0.N.

FZO	Meaning
0b0	Do not freeze on overflow.
0b1	Event counter PMU.PMEVCNTR<n>_EL0 does not count when PMOVSCLR_EL0[(PMN-1):0] is nonzero and n is in the range of affected event counters.

If PMN is not 0, this field affects the operation of event counters in the range [0 .. (PMN-1)].

This field does not affect the operation of other event counters and PMU.PMCCNTR_EL0.

The operation of this field applies even when EL2 is disabled in the current Security state.

The reset behavior of this field is:

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

Otherwise:

Reserved, res0.

Bit [8]

Reserved, res0.

LP, bit [7]
When FEAT_PMUv3p5 is implemented:

Long event counter enable. Determines when unsigned overflow is recorded by an event counter overflow bit.

In the description of this field:

- If EL2 is implemented and is using AArch32, PMN is HDCR.HPMN.
- If EL2 is implemented and is using AArch64, PMN is MDCR_EL2.HPMN.
- If EL2 is not implemented, PMN is PMCR_EL0.N.

LP	Meaning
0b0	Event counter overflow on increment that causes unsigned overflow of PMU.PMEVCNTR<n>_EL0[31:0].
0b1	Event counter overflow on increment that causes unsigned overflow of PMU.PMEVCNTR<n>_EL0[63:0].

If PMN is not 0, this bit affects the operation of event counters in the range [0 .. (PMN-1)].

The field does not affect the operation of other event counters and PMU.PMCCNTR_EL0.

The operation of this field applies even when EL2 is disabled in the current Security state.

The reset behavior of this field is:

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

Otherwise:

Reserved, res0.

LC, bit [6]
When AArch32 is supported:

Long cycle counter enable. Determines when unsigned overflow is recorded by the cycle counter overflow bit.

LC	Meaning
0b0	Cycle counter overflow on increment that causes unsigned overflow of PMU.PMCCNTR_EL0[31:0].
0b1	Cycle counter overflow on increment that causes unsigned overflow of PMU.PMCCNTR_EL0[63:0].

Arm deprecates use of PMU.PMCR_EL0.LC = 0.

The reset behavior of this field is:

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

Otherwise:

Reserved, res1.

DP, bit [5]
When EL3 is implemented or (FEAT_PMUv3p1 is implemented and EL2 is implemented):

Disable cycle counter when event counting is prohibited. The possible values of this bit are:

DP	Meaning
0b0	Cycle counting by PMU.PMCCNTR_EL0 is not affected by this mechanism.

0b1	Cycle counting by PMU.PMCCNTR_EL0 is disabled in prohibited regions and when event counting is frozen: • If FEAT_PMUv3p1 is implemented, EL2 is implemented, and MDCR_EL2.HPMD is 1, then cycle counting by PMU.PMCCNTR_EL0 is disabled at EL2. • If FEAT_PMUv3p7 is implemented, EL3 is implemented and using AArch64, and MDCR_EL3.MPMX is 1, then cycle counting by PMU.PMCCNTR_EL0 is disabled at EL3. • If FEAT_PMUv3p7 is implemented and event counting is frozen by PMCR_EL0.FZO, then cycle counting by PMU.PMCCNTR_EL0 is disabled. • If EL3 is implemented, MDCR_EL3.SPME is 0, and either FEAT_PMUv3p7 is not implemented or MDCR_EL3.MPMX is 0, then cycle counting by PMU.PMCCNTR_EL0 is disabled at EL3 and in Secure state. If MDCR_EL2.HPMN is not 0, this is when event counting by event counters in the range [0..(MDCR_EL2.HPMN-1)] is prohibited or frozen.

For more information, see 'Prohibiting event and cycle counting'.

The reset behavior of this field is:

- On a Warm reset:
  - When the implementation only supports execution in AArch32 state, this field resets to `0`.
  - Otherwise, this field resets to an architecturally unknown value.

Otherwise:

Reserved, res0.

X, bit [4]
When the implementation includes a PMU event export bus:

Enable export of events in an implementation defined PMU event export bus.

X	Meaning
0b0	Do not export events.
0b1	Export events where not prohibited.

This field enables the exporting of events over an implementation defined PMU event export bus to another device, for example to an optional trace unit.

No events are exported when counting is prohibited.

This field does not affect the generation of Performance Monitors overflow interrupt requests or signaling to a cross-trigger interface (CTI) that can be implemented as signals exported from the PE.

When this register has an architecturally-defined reset value, if this field is implemented as an RW field it resets to:

- A value that is architecturally unknown if the reset is into an Exception level that is using AArch64.
- 0 if the reset is into an Exception level that is using AArch32.

Otherwise:

Reserved, RAZ/WI.

D, bit [3]
When AArch32 is supported:

Clock divider.

D	Meaning
0b0	When enabled, PMU.PMCCNTR_EL0 counts every clock cycle.
0b1	When enabled, PMU.PMCCNTR_EL0 counts once every 64 clock cycles.

If PMCR_EL0.LC == 1, this bit is ignored and the cycle counter counts every clock cycle.

Arm deprecates use of PMCR_EL0.D = 1.

When this register has an architecturally-defined reset value, if this field is implemented as an RW field it resets to:

- A value that is architecturally unknown if the reset is into an Exception level that is using AArch64.
- 0 if the reset is into an Exception level that is using AArch32.

Otherwise:

Reserved, res0.

C, bit [2]

Cycle counter reset. The effects of writing to this bit are:

C	Meaning
0b0	No action.
0b1	Reset PMU.PMCCNTR_EL0 to zero.

Note

Resetting PMU.PMCCNTR_EL0 does not change the cycle counter overflow bit. If FEAT_PMUv3p5 is implemented, the value of PMCR_EL0.LC is ignored, and bits [63:0] of the cycle counter are reset.

Access to this field is **WO/RAZ**.

P, bit [1]

Event counter reset. The effects of writing to this bit are:

P	Meaning
0b0	No action.
0b1	Reset all event counters, not including PMU.PMCCNTR_EL0, to zero.

Note

Resetting the event counters does not change the event counter overflow bits. If FEAT_PMUv3p5 is implemented, the value of MDCR_EL2.HLP, or PMCR_EL0.LP is

ignored and bits [63:0] of all affected event counters are reset.

Access to this field is **WO/RAZ**.

E, bit [0]

Enable.

In the description of this field:

- If EL2 is implemented and is using AArch32, PMN is HDCR.HPMN.
- If EL2 is implemented and is using AArch64, PMN is MDCR_EL2.HPMN.
- If EL2 is not implemented, PMN is PMCR_EL0.N.

E	Meaning
0b0	PMU.PMCCNTR_EL0 is disabled and event counters PMU.PMEVCNTR<n>_EL0, where n is in the range of affected event counters, are disabled.
0b1	PMU.PMCCNTR_EL0 and event counters PMU.PMEVCNTR<n>_EL0, where n is in the range of affected event counters, are enabled by PMU.PMCNTENSET_EL0.

If PMN is not 0, this field affects the operation of event counters in the range [0 .. (PMN-1)].

This field does not affect the operation of other event counters.

The operation of this field applies even when EL2 is disabled in the current Security state.

The reset behavior of this field is:

- On a Warm reset, this field resets to 0.

Accessing PMCR_EL0

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

Accessible at offset 0xE04 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 0xE10 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**.

28/03/2023 16:01; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94