G4.4.12 PMOVSR, Performance Monitors Overflow Flag Status Register

The PMOVSR characteristics are:

Purpose

Contains the state of the overflow bit for the Cycle Count Register, PMCCNTR, and each of the implemented event counters PMEVCNTR<x>. Writing to this register clears these bits.

This register is part of the Performance Monitors registers functional group.

Usage constraints

This register is accessible as shown below:

EL0 (NS)	EL0 (S)	EL1 (NS)	EL1 (S)	EL2	EL3 (SCR.NS=1)	EL3 (SCR.NS=0)
Config-RW	Config-RW	RW	RW	RW	RW	RW

This register is accessible at EL0 when PMUSERENR.EN is set to 1.

If EL2 is implemented, in Non-secure EL1 and EL0 modes, the value of HDCR.HPMN can change the behavior of accesses to PMOVSR. See the description of the Px bit.

Configurations

PMOVSR is architecturally mapped to AArch64 register PMOVSCLR_EL0.

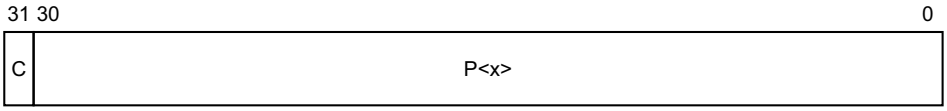
PMOVSR is architecturally mapped to external register PMOVSCLR_EL0.

There is one instance of this register that is used in both Secure and Non-secure states.

Attributes

PMOVSR is a 32-bit register.

The PMOVSR bit assignments are:



C, bit [31]

PMCCNTR overflow bit. Possible values are:

`0` When read, means the cycle counter has not overflowed. When written, has no effect.

`1` When read, means the cycle counter has overflowed. When written, clears the overflow bit to 0.

PMCR.LC is used to control from which bit of PMCCNTR (bit 31 or bit 63) an overflow is detected.

On Warm reset, the field reset value is architecturally UNKNOWN.

P<x>, bit [x] for x = 0 to (N - 1)

Event counter overflow clear bit for PMEVCNTR<x>.

When EL2 is implemented, in Non-secure EL1 and EL0, N is the value in HDCR.HPMN. Otherwise, N is the value in PMCR.N.

Bits [30:N] are RAZ/WI.

Possible values of each bit are:

`0` When read, means that PMEVCNTR<x> has not overflowed. When written, has no effect.

1 When read, means that PMEVCNTR<x> has overflowed. When written, clears the PMEVCNTR<x> overflow bit to 0.

On Warm reset, the field reset value is architecturally UNKNOWN.

Accessing the PMOVSR:

To access the PMOVSR:

```
MRC p15,0,<Rt>,c9,c12,3 ; Read PMOVSR into Rt
MCR p15,0,<Rt>,c9,c12,3 ; Write Rt to PMOVSR
```

Register access is encoded as follows:

coproc	opc1	CRn	CRm	opc2
1111	000	1001	1100	011