RISC-V Count Overflow and Mode-Based Filtering Extension

Sscofpmf

Version 0.5.2-01d1df0, 2021-10-13: frozen

Table of Contents

Preamble

Document Source Information

See github.com/riscv/riscv-count-overflow for document source. The document version includes the source control tag or commit hash used to produce this document.

Introduction

The current Privileged specification defines mhpmevent CSRs to select and control event counting by the associated hpmcounter CSRs, but provides no standardization of any fields within these CSRs. For at least Linux-class rich-OS systems it is desirable to standardize certain basic features that are broadly desired (and have come up over the past year plus on RISC-V lists, as well as have been the subject of past proposals). This enables there to be standard upstream software support that eliminates the need for implementations to provide their own custom software support.

This extension serves to accomplish exactly this within the existing mhpmevent CSRs (and correspondingly avoids the unnecessary creation of whole new sets of CSRs - past just one new CSR).

This extension sticks to addressing two basic well-understood needs that have been requested by various people. To make it easy to understand the deltas from the current Priv 1.11/1.12 specs, this is written as the actual exact changes to be made to existing paragraphs of Priv spec text (or additional paragraphs within the existing text).

The extension name is "Sscofpmf" ('Ss' for Privileged arch and Supervisor-level extensions, and 'cofpmf' for Count OverFlow and Privilege Mode Filtering).

Note that the new count overflow interrupt will be treated as a standard local interrupt that is assigned to bit 13 in the mip/mie/sip/sie registers.

Chapter 1. Machine Level Additions

1.1. Hardware Performance Monitor

This extension expands the hardware performance monitor description and extends the mhpmevent registers to 64 bits (in RV32) as follows:

The hardware performance monitor includes 29 additional 64-bit event counters and 29 associated 64-bit event selector registers - the mhpmcounter3–mhpmcounter31 and mhpmevent3–mhpmevent31 CSRs.

The mhpmcounters are WARL registers that support up to 64 bits of precision on RV32 and RV64.

The mhpmevent*n* registers are WARL registers that control which event causes the corresponding counter to increment and what happens when the corresponding count overflows. Currently just a few bits are defined here. Past this, the actual selection and meaning of events is defined by the platform, but (mhpmevent == 0) is defined to mean “no event" and that the corresponding counter will never be incremented. Typically the lower bits of mhpmevent will be used for event selection purposes.

On RV32 only, accesses to the mcycle, minstret, mhpmcounter*n*, and mhpmevent*n* CSRs access the low 32 bits, while accesses to the mcycleh, minstreth, mhpmcounter*n*h, and mhpmevent*n*h CSRs access bits 63–32 of the corresponding counter or event selector. The proposed CSR numbers for mhpmevent*n*h are 0x723 - 0x73F.

The following bits are added to mhpmevent:

bit [63] OF - Overflow status and interrupt disable bit that is set when counter overflows

bit [62] MINH - If set, then counting of events in M-mode is inhibited

bit [61] SINH - If set, then counting of events in S/HS-mode is inhibited

bit [60] UINH - If set, then counting of events in U-mode is inhibited

bit [59] VSINH - If set, then counting of events in VS-mode is inhibited

bit [58] VUINH - If set, then counting of events in VU-mode is inhibited

bit [57] 0 - Reserved for possible future modes

bit [56] 0 - Reserved for possible future modes

Each of the five 'x'INH bits, when set, inhibit counting of events while in privilege mode 'x'. All-zeroes for these bits results in counting of events in all modes.

The OF bit is set when the corresponding hpmcounter overflows, and remains set until written by software. Since hpmcounter values are unsigned values, overflow is defined as unsigned overflow of the implemented counter bits. Note that there is no loss of information after an overflow since the counter wraps around and keeps counting while the sticky OF bit remains set.

If supervisor mode is implemented, the 32-bit scountovf register contains read-only shadow copies of the OF bits in all 32 mhpmevent registers.

If an hpmcounter overflows while the associated OF bit is zero, then a "count overflow interrupt request" is generated. If the OF bit is one, then no interrupt request is generated. Consequently the OF bit also functions as a count overflow interrupt disable for the associated hpmcounter.

Count overflow never results from writes to the mhpmcounter*n* or mhpmevent*n* registers, only from hardware increments of counter registers.

This "count overflow interrupt request" signal is treated as a standard local interrupt that corresponds to bit 13 in the mip/mie/sip/sie registers. The mip/sip LCOFIP and mie/sie LCOFIE bits are respectively the interrupt-pending and interrupt-enable bits for this interrupt. ('LCOFI' represents 'Local Count Overflow Interrupt'.)

Generation of a "count overflow interrupt request" by an hpmcounter sets the LCOFIP bit in the mip/sip registers and sets the associated OF bit. The mideleg register controls the delegation of this interrupt to S-mode versus M-mode. The LCOFIP bit is cleared by software before servicing the count overflow interrupt resulting from one or more count overflows.

Non-normative

There are not separate overflow status and overflow interrupt enable bits. In practice, enabling overflow interrupt generation (by clearing the OF bit) is done in conjunction with initializing the counter to a starting value. Once a counter has overflowed, it and the OF bit must be reinitialized before another overflow interrupt can be generated.

Non-normative

Software can distinguish newly overflowed counters (yet to be serviced by an overflow interrupt handler) from overflowed counters that have already been serviced or that are configured to not generate an interrupt on overflow, by maintaining a bit mask reflecting which counters are active and due to eventually overflow.

1.2. Machine Interrupt Registers (mip and mie)

This extension adds the description of the LCOFIP/LCOFIE bits in these registers (and modifies related text) as follows:

LCOFIP is added to mip in Figure 3.14 as bit 13. LCOFIP is added to mie in Figure 3.15 as bit 13.

If the Sscofpmf extension is implemented, bits mip.LCOFIP and mie.LCOFIE are the interrupt-pending and interrupt-enable bits for local count overflow interrupts. LCOFIP is read-write in mip and reflects the occurrence of a local count overflow interrupt request resulting from any of the mhpmevent*n*.OF bits being set. If the Sscofpmf extension is not implemented, these LCOFIP and LCOFIE bits are hardwired to zeros.

Multiple simultaneous interrupts destined for different privilege modes are handled in decreasing order of destined privilege mode. Multiple simultaneous interrupts destined for the same privilege mode are handled in the following decreasing priority order: MEI, MSI, MTI, SEI, SSI, STI, LCOFI.

Chapter 2. Supervisor Level Additions

2.1. Supervisor Interrupt Registers (sip and sie)

This extension adds the description of the LCOFIP/LCOFIE bits in these registers (and modifies related text) as follows:

LCOFIP is added to sip in Figure 4.6 as bit 13. LCOFIP is added to sie in Figure 4.7 as bit 13.

If the Sscofpmf extension is implemented, bits sip.LCOFIP and sie.LCOFIE are the interrupt-pending and interrupt-enable bits for local count overflow interrupts. LCOFIP is read-write in sip and reflects the occurrence of a local count overflow interrupt request resulting from any of the mhpmevent*n*.OF bits being set. If the Sscofpmf extension is not implemented, these LCOFIP and LCOFIE bits are hardwired to zeros.

Each standard interrupt type (LCOFI, SEI, STI, or SSI) may not be implemented, in which case the corresponding interrupt-pending and interrupt-enable bits are hardwired to zeros. All bits in sip and sie are WARL fields.

Multiple simultaneous interrupts destined for supervisor mode are handled in the following decreasing priority order: SEI, SSI, STI, LCOFI.

2.2. Supervisor Count Overflow (scountovf)

This extension adds this new CSR.

The scountovf CSR is a 32-bit read-only register that contains shadow copies of the OF bits in the 29 mhpmevent CSRs (mhpmevent*3* - mhpmevent*31*) - where scountovf bit *X* corresponds to mhpmevent*X*. The proposed CSR number is 0xDA0.

This register enables supervisor-level overflow interrupt handler software to quickly and easily determine which counter(s) have overflowed (without needing to make an execution environment call or series of calls ultimately up to M-mode).

Read access to bit *X* is subject to the same mcounteren (or mcounteren and hcounteren) CSRs that mediate access to the hpmcounter CSRs by S-mode (or VS-mode). In M and S modes, scountovf bit *X* is readable when mcounteren bit *X* is set, and otherwise reads as zero. Similarly, in VS mode, scountovf bit *X* is readable when mcounteren bit *X* and hcounteren bit *X* are both set, and otherwise reads as zero.

Index

Bibliography


RISC-V