IOPMP Task Group Meeting

August 31, 2023

Video link

Meeting Notes

- New fields to add:
  - Global enable
  - Version number
- 32-bit control port:
  - Remove LC-extension
  - No atomicity support on 64-bit register, except MD_STALL
  - The location of lock-bit and exempt-bit: move to LSB
- Previous items and ask feedback:
  - Not to support the per-entry interrupt-enable bits: no objection so far
  - To support the parallel rule match: no objection so far
  - Address encoding, the leading bits can be hardwired: no objection so far
  - Error reporting on multi-fault: will refine the proposal in next meeting

New Fields

- Global enable, GE:
  - It can be hardwired to 1 or sticky to 1 (init GE = 0)
  - To enable the checks (when GE=1)
- Version number
  - RO, to indicate the spec version

All registers are defined as 32-bit

- Remove LC-extension :
  - Because it makes some registers have different offsets and makes SW fragmented.
- The original 64-bit registers will be spilt into two 32-bit registers.
- Unused registers (holding higher 32 bits): WARZ
  - Still occupy a space.
  - On a 32-bit system: ENTRY_ADDRH(i)
  - On a system with <32 MDs: MD_STALLH, SRCMDH, MDMSKH, …
- Atomical writes to paired regs should be ensured by programmers:
  - Any form of 64-bit atomical write is implementation-dependent, except MD_STALL(H)
- MD_STALL(H) (if both implemented) should support 64-bit atomical write.
- Lock bits and exempt bits:
  - Move them from MSB (bit 63) to LSB (bit 0) to better support the systems with < 32 MDs.
  - SRCMD(H), MDMSK(H), MD_STALL(H)

Atomical write on MD_STALL and MD_STALLH

- The original MD_STALL is 64-bit:
  - Split it into MD_STALL (lower 32 bits) and MD_STALLH (higher 32 bits)
- If MD_STALLH is implemented (#MDs >31), the MD_STALL and MD_STALLH should work synchronously. Missing anyone will interpret incorrectly.
- Write MD_STALLH ➔ just hold the value, no action is taken.
- Write MD_STALL ➔ perform the stall action according to the current MD_STALLH and new updated MD_STALL.
- The programming sequence of MD_STALL and MD_STALLH should be:
  - Step1: write MD_STALLH // the higher 32 bits are held only
  - Step2: write MD_STALL // take effects on both registers