[Bug Report] Incorrect exception type of PMP violation during address translation #906

Closed **Phantom1003** opened this issue on Jun 8 · 5 comments · Fixed by #908

Phantom1003 commented on Jun 8 • edited Contributor

Our co-simulation framework found that the exception type of address translation PMP violation is incorrect.

In the following test case, we let a non-leaf (level 2) PTE out of the PMP range.
Next, we will perform a store operation with a special virtual address, during translation processor will try to access the PTE that does not match any PMP.
cva6 throws a load access fault, while spike throws a store access fault.

```
[cva6]  Exception @    119000, PC: 000000008000036c, Cause: Load Access Fault,
[cva6]                                 tval: 0000000080102008
[spike] core   0: 0x000000008000036c (0x00b52023) sw      a1, 0(a0)
[spike] core   0: exception trap_store_access_fault, epc 0x000000008000036c
[spike] core   0:           tval 0x0000000040201010
[spike] core   0: 0x0000000080000004 (0x34302f73) csrr    t5, mtval
[error] WDATA SIM 0000000040201010, DUT 0000000080102008
[error] check board clear 30 error
```

According to riscv-privileged specification:

> If accessing pte violates a PMA or PMP check, raise an access-fault exception corresponding to the original access type.

Hence, cva6 should throw a store access fault.

cva6-9.zip

> **@ProjectDimlight** helps reproduce the problem

Moschn added a commit to Moschn/ariane that referenced this issue on Jun 9

`Fix exception type on PMP check during PTW` … 078225d

✓

Moschn mentioned this issue on Jun 9

Fix exception type on PMP check during PTW #908

Merged

Moschn commented on Jun 9 Contributor

Thanks for reporting this.

I proposed a fix in #908. Can you try if this solves your issues?

zarubaf commented on Jun 9 Contributor

Thanks **@Moschn**.
@Phantom1003 It would be really great if we could test the patch of **@Moschn**.

Phantom1003 commented on Jun 9 • edited Contributor Author

Thanks, cva6 throws a correct Store Access Fault exception, but the *tval seems filled with the physical address instead of the virtual address:

```
Exception @    119000, PC: 000000008000036c, Cause: Store Access Fault,
                                tval: 0000000080102008
core   0: 0x000000008000036c (0x00b52023) sw      a1, 0(a0)
core   0: exception trap_store_access_fault, epc 0x000000008000036c
core   0:           tval 0x0000000040201010
core   0: 0x0000000080000004 (0x34302f73) csrr    t5, mtval
[error] WDATA SIM 0000000040201010, DUT 0000000080102008
[error] check board clear 30 error
[CJ]           0 integer register Judge Failed
    1204ns     1189 M 0000000080000004 0 34302f73 csrr           t5, mtval
```

The related verse in specification is:

> When page-based virtual memory is enabled, mtval is written with the faulting virtual address, even for physical-memory access-fault exceptions.

Moschn added a commit to Moschn/ariane that referenced this issue on Jun 9

`Fix exception type on PMP check during PTW` … ✓ 904acc8

Moschn commented on Jun 9 Contributor

I just read this section of the spec and it seems to be a bit ambiguous. Yes `mtval` should contain the faulting virtual address, but which address is this exactly? Is this the virtual address that is being translated? or the address of the PTW that failed? I guess all accesses during a page table walk are to physical memory and thus the faulting address should be the initial virtual address.

I force pushed the fix to #908. Can you check if it passes your tests now?

Phantom1003 commented on Jun 9 Contributor Author

Yes, I agree with your understanding.
And the test case passed, cva6 and spike have the same behavior.
Thank you for your quick response, I will close this issue after the commit has been merged.

Phantom1003 closed this as completed on Jun 9

JeanRochCoulon pushed a commit that referenced this issue on Jul 8

`Fix exception type on PMP check during PTW (#908)` … ✗ `011edf4`

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Successfully merging a pull request may close this issue.

Fix exception type on PMP check during PTW
Moschn/ariane

3 participants