New issue

Jump to bottom

[Bug Report] Incorrect exception type for illegal virtual address access #904

Open **Phantom1003** opened this issue on Jun 7 · 0 comments

Phantom1003 commented on Jun 7 Contributor

Our co-simulation framework found that the exception type when access/load/store an illegal virtual address is incorrect.

In the following test case, we flipped the MSB of a legal virtual address and load with it at 0x00002980.
cva6 throws an access fault, while spike throws a page fault.
Since page fault has been delegated to s mode, they jump to different trap vectors.

```
[spike]  core   0: 0x000000000000297c (0x00200193) li      gp, 2
[cva6]   14681ns    14666 U 000000000000297c 0 00200193 li             gp, 2                gp
:0000000000000002
[cva6]   Exception @   1469600, PC: 0000000000002980, Cause: Load Access Fault,
[cva6]                               tval: 8000000000003000
[spike]  core   0: 0x0000000000002980 (0x00028383) lb      t2, 0(t0)
[spike]  core   0: exception trap_load_page_fault, epc 0x0000000000002980
[spike]  core   0:           tval 0x8000000000003000
[spike]  core   0: 0xffffffffffe00144 (0x14011173) csrrw   sp, sscratch, sp
[error] PC SIM ffffffffffe00144, DUT 0000000080000008
[error] INSN SIM 14011173, DUT 2c40206f
[CJ] Commit Failed
```

According to riscv-privileged specification:

> Instruction fetch addresses and load and store effective addresses, which are 64 bits, must have bits 63–39 all equal to bit 38, or else a page-fault exception will occur.

Both `load`, `store`, and `fetch` can cause this issue.

fetch test case: cva6-7.zip
load test case: cva6-6.zip

> @LuminaDCIX helps reproduce the problem

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

1 participant