openhwgroup / **cva6** Public

Code   Issues 201   Pull requests 7   Discussions   Actions   Projects   ···

New issue   Jump to bottom

[Bug Report] Incorrect *tval for ecall/ebreak #898

Open **Phantom1003** opened this issue on Jun 3 · 2 comments

Phantom1003 commented on Jun 3 • edited   Contributor

Our co-simulation framework found that the `*tval` of `ecall/ebreak` is incorrect.
In cva6, after `ecall/ebreak`, *tval will set to the machine code of the `ecall/ebreak` instruction.

In the following test case, after calling `ebreak` in s-mode, the value of `mtval` register is written to `0x100073`, which is the machine code of `ebreak` instruction.

```
[spike] core   0: 0x0000000080000174 (0x00100193) li      gp, 1
[cva6]       664ns      649 S 0000000080000174 0 00100193 li             gp, 1
[cva6]  Exception @     66500, PC: 0000000080000178, Cause: Breakpoint, tval: 0000000000100073
[spike] core   0: 0x0000000080000178 (0x00100073) ebreak
[spike] core   0: exception trap_breakpoint, epc 0x0000000080000178
[spike] core   0:           tval 0x0000000080000178
... /* in handler */
[spike] core   0: 0x0000000080000190 (0x343022f3) csrr    t0, mtval
[error] WDATA SIM 0000000080000178, DUT 0000000000100073
[error] check board clear 5 error
[CJ]  integer register Judge Failed
```

riscv-priviledged P41 : If `mtval` is written with a nonzero value when a breakpoint, address-misaligned, access-fault, or page-fault exception occurs on an instruction fetch, load, or store, then mtval will contain the faulting virtual address.
According to specifications, `mtval` should be the faulting address (or zero).

Issue 448 tests the value in `stval` of `ecall` from user mode, our verification framework further discovered that `ebreak` also has the same bug, and both of them could be triggered under **any privilege modes**.

ebreak testcase: cva6-1.zip
ecall testcase: cva6-2.zip

> @LuminaDCIX helps reproduce the problem

zarubaf commented on Jun 7 Contributor

Indeed, the instruction bits are the default case for every instruction. Confirming that we are not complying.

`ebreak` / `ecall` should be able to be triggered from any privilege level, no? How would a syscall/debug call work otherwise from user space?

Phantom1003 commented on Jun 7 Contributor Author

Thanks, the point we wanted to confirm was the mismatched *tval.
And sorry for the confusion in my description, we wanted to point out that the ecall/ebreak triggered in any privileged mode will produce a mismatched value, not just the case in user mode as mentioned in 448.

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants