New issue

Jump to bottom

[Bug Report] Wrong exception priority during access memory #971

Open · Phantom1003 opened this issue on Apr 11 · 4 comments

Phantom1003 commented on Apr 11 · Contributor

Let's take the load instruction as an example:

riscv-isa-sim/riscv/mmu.h
Lines 99 to 125 in `0f15aa0`

```
 99     #define load_func(type, prefix, xlate_flags) \
100       inline type##_t prefix##_##type(reg_t addr, bool require_alignment = false) { \
101         if (unlikely(addr & (sizeof(type##_t)-1))) { \
102           if (require_alignment) load_reserved_address_misaligned(addr); \
103           else return misaligned_load(addr, sizeof(type##_t), xlate_flags); \
104         } \
105         reg_t vpn = addr >> PGSHIFT; \
106         size_t size = sizeof(type##_t); \
107         if ((xlate_flags) == 0 && likely(tlb_load_tag[vpn % TLB_ENTRIES] == vpn)) { \
108           if (proc) READ_MEM(addr, size); \
```

At line 101, load will first check if it is aligned, then at line 122 it will try to access the address in the load_slow_path function.

riscv-isa-sim/riscv/mmu.cc
Lines 142 to 162 in `0f15aa0`

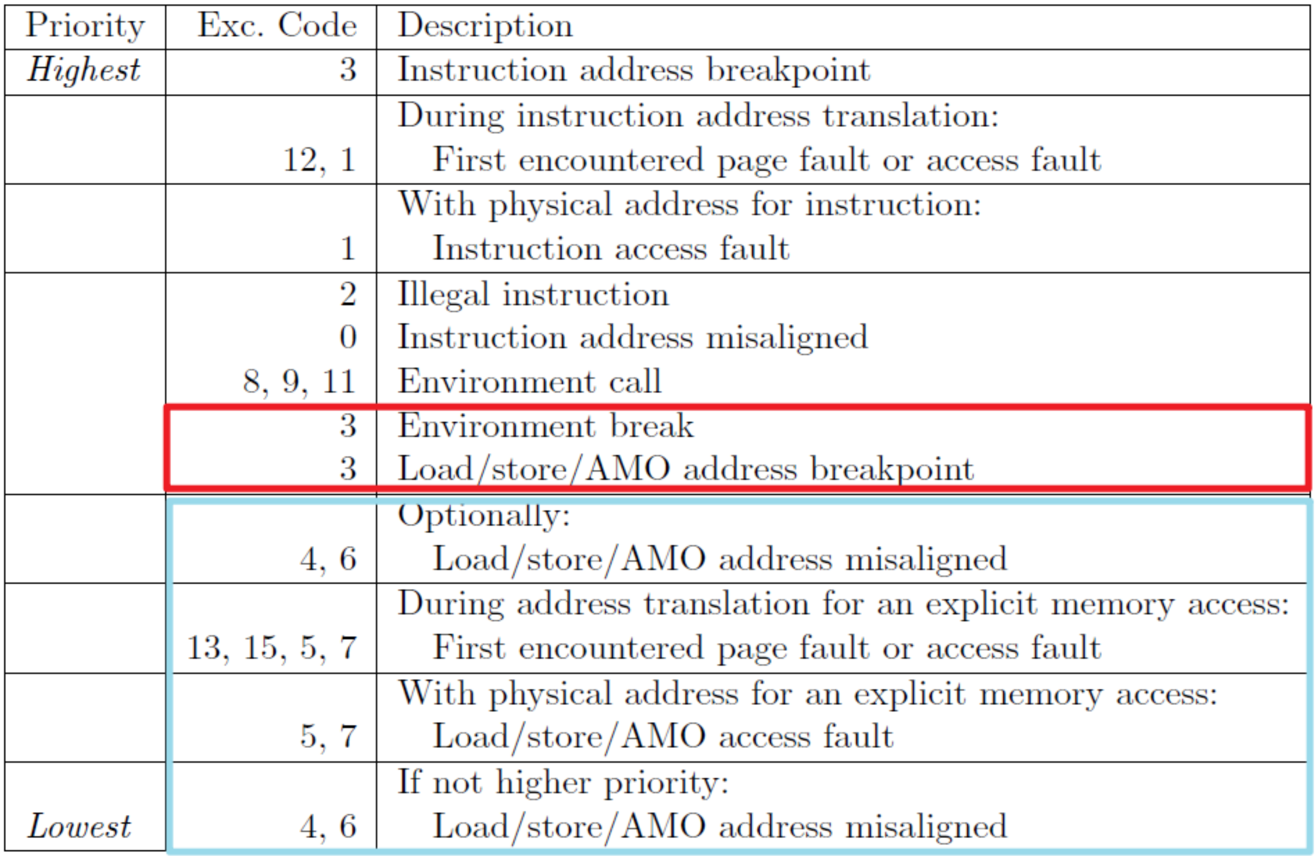
```
142     void mmu_t::load_slow_path(reg_t addr, reg_t len, uint8_t* bytes, uint32_t xlate_flags
143     {
144       reg_t paddr = translate(addr, len, LOAD, xlate_flags);
145
146       if (auto host_addr = sim->addr_to_mem(paddr)) {
147         memcpy(bytes, host_addr, len);
148         if (tracer.interested_in_range(paddr, paddr + PGSIZE, LOAD))
149           tracer.trace(paddr, len, LOAD);
150         else if (xlate_flags == 0)
```

```
151          refill_tlb(addr, paddr, host_addr, LOAD);
```

In load_slow_path, it will first check if it is legal address at line 153, and the watch point will be checked at the end of the function.

Briefly, the order of priority is as follows: trap_load_address_misaligned > trap_load_access_fault > trap_breakpoint

However, in the specification, trap_breakpoint has a higher priority than the others:



Priority	Exc. Code	Description
Highest	3	Instruction address breakpoint
	12, 1	During instruction address translation: First encountered page fault or access fault
	1	With physical address for instruction: Instruction access fault
	2	Illegal instruction
	0	Instruction address misaligned
	8, 9, 11	Environment call
	3	Environment break
	3	Load/store/AMO address breakpoint
	4, 6	Optionally: Load/store/AMO address misaligned
	13, 15, 5, 7	During address translation for an explicit memory access: First encountered page fault or access fault
	5, 7	With physical address for an explicit memory access: Load/store/AMO access fault
Lowest	4, 6	If not higher priority: Load/store/AMO address misaligned

We also co-simulate with rocket to check this point, rocket threw a breakpoint exception, while spike threw an error misaligned exception.
The test point is at 0x800001c0 where loading a misaligned illegal address 0x100004001:

```
3 0x00800001bc (0x7a261073)
core   0: 0x00000000800001bc (0x7a261073) csrw    tdata2, a2
3 0x0080000004 (0x34302f73) x30 0x0000000100004001
core   0: 0x00000000800001c0 (0x00062603) lw      a2, 0(a2)
core   0: exception trap_load_address_misaligned, epc 0x00000000800001c0
core   0:           tval 0x0000000100004001
core   0: 0x0000000080000004 (0x34302f73) csrr    t5, mtval
3 0x0080000008 (0x34202f73) x30 0x0000000000000003
core   0: 0x0000000080000008 (0x34202f73) csrr    t5, mcause
[error] WDATA SIM 0000000000000004, DUT 0000000000000003
```

```
  [error] check board clear 30 error
  [CJ] integer register Judge Failed
```

spike-0.zip

scottj97 commented on Apr 11 Collaborator

I believe this is a duplicate of #538

Phantom1003 commented on Apr 11 Contributor Author

Yes, but the priority of the misaligned exception is also wrong, which is outside of the load_slow_path function.

scottj97 commented on Apr 11 Collaborator

@timsifive perhaps this is of interest to you, since you're doing some trigger work now.

timsifive commented on Apr 11 Collaborator

I agree that this is wrong, and a (exceptionally clearly described) problem. It's been a long time since I added that trigger code. I'm not sure how to raise the trigger priority while minimizing the performance impact.

I might have time to look at this in a few weeks, when I'm back from vacation.

👍 1

timsifive added a commit that referenced this issue on May 25

`Don't check alignment before load_slow_path()` … 37fe50b

timsifive added a commit that referenced this issue on May 25

`Fix trigger store priority.` … ✓ d392e69

timsifive mentioned this issue on May 25

Check triggers before checking alignment #1013

Closed

timsifive added a commit that referenced this issue on May 26

`Don't check alignment before load_slow_path()` … a315c61

timsifive added a commit that referenced this issue on May 26

`Fix trigger store priority.` … ✓ 3ce7a51

timsifive added a commit that referenced this issue on May 27

`Check for alignment in load_slow_path().` … fb8eb2a

timsifive added a commit that referenced this issue on May 27

`Fix trigger store priority.` … 9cb3499

timsifive added a commit that referenced this issue on Jun 10

`Check for alignment in load_slow_path().` … bb65ba4

timsifive added a commit that referenced this issue on Jun 10

`Fix trigger store priority.` … 60b0caa

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Successfully merging a pull request may close this issue.

Check triggers before checking alignment

riscv-software-src/riscv-isa-sim

3 participants