

[New issue](#)[Jump to bottom](#)

Unclear semantics of CSR instruction #863

Open lorddoskias opened this issue on Jun 30 · 1 comment

lorddoskias commented on Jun 30 • edited ▾

Current CSR instruction go along the lines of (I've emboldened the confusing part):

The CSRRW (Atomic Read/Write CSR) instruction atomically swaps values in the CSRs and integer registers. CSRRW reads the old value of the CSR, zero-extends the value to XLEN bits, then writes it to integer register rd. ****The initial value**** in rs1 is written to the CSR. If rd=x0, then the instruction shall not read the CSR and shall not cause any of the side-effects that might occur on a CSR read.

According to this definition executing something like: CSRRW t0, csr, t0 shall be noop in the sense that the csr is first read and it's value is written to rd (t0) in the example, subsequently t0's value is written to the csr, but because t0 contains the current value of the csr we write the same value. Clearly this is not the case, instead after the instruction is executed the values of rs1 and csr are swapped if rs1==rd. This is kind of alluded to in the description by saying "**The initial value in rs1**" is written, so if rs1 == rd then behind the scenes the initial value is saved somewhere, then it's overwritten by the read and finally it's moved into the csr register. Can perhaps the wording be changed to make this semantic more explicit?

allenjbaum commented on Jul 1

The first sentence makes it clear: the instruction atomically swaps the two values. There is no "subsequently"; both are logically read and written at the same time. How it actually gets performed by HW under the covers is not programmatically visible, so it appears simultaneous. HW may stage the initial value of the CSR and rs1 into a pipeline register or not - programs don't care, as long as there is no way for a program to tell that it has (and you shouldn't be able to).

The interesting part is if the rd is x0, then it's an rs1-> CSR transfer only, so there is no simultaneous CSR->rd transfer,



Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants

