

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

Compressed Extension, immediates, offsets, etc. #830



saahm opened this issue on Mar 18 · 0 comments

saahm commented on Mar 18

Hey,

I'm currently working with the ISA for the C-Extension and noticed there is a lot of different scrambled immediates and offsets for some instructions.

As someone who implements parts of the ISA I found the RV32I way of describing immediates very helpful. The diagram in figure 2.4 in the RV32I section was insightful for what would else be described in text only. Now for the C-Extension I run into this issue, where its not entirely clear how an immediate/offset/etc. encoded value is extended (especially with the formulations alone the idea of "scaled by 8" etc.). Generally it feels like the C-Extension ISA is written very concise (not to say compressed). While only focusing on the relevant ideas it also shouldn't let the be confused about the formulation (that seems to be a bit different from lets say RV32I or M in the specification). I found other issues that mention similar points (for example [#389](#) [#509](#) [#583](#)).

I do understand that due to the nature of the C-Extension there is a big variety in the immediate values. Yet that should not cause the reader of the specification to spend time deciphering how to build certain immediate values. Tho having a table similar to the RV32I figure 2.4 with a clear description on how the immediate/offset values are built, extended and ordered from the instruction would help to add a lot of clarity to the specification.

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

1 participant

