Scalar Efficiency SIG Meeting

May 2, 2024

Derek Hower, Qualcomm

Agenda

- Call for Chair/Vice-chair
- Load/store pair
- Instruction database format
- Discuss processor types / metrics / workloads

Call for Chair/Vice-chair

- Nominations must be received by **May 10, 2024**
  - Send name, affiliation, qualifications, and short bio to help@riscv.org
- Further reading:
  - Groups & Chairs policy for more information on the process
  - Chairs Best Practices for more information on chair duties & responsibilities.

Load/store pair

- Recall: ARC authorized Zilsd (RV32 load/store double into sequential registers) fast-track extension
- Apps & Tools HC has requested that flexible load/store pairs (independent dst regs), included in the SE SIG charter, be considered for consistency and to reduce burden on toolchains.
- Two proposals:

	Alibaba T-Head	**Qualcomm**	**LD/SD (RV32)**
Encoding size	32	32	32
Dest Regs	Independently specified	Independently specified	Sequential (even/odd)
Addressing mode	Reg-imm (shifted)	Reg-imm (shifted)	Reg-imm
Src Reg	**Independently specified**	**Implicitly `sp`**	Independently specified
Variants	**w, uw (RV64), d (RV64)**	**b, ub, h, uh, w, uw (RV64), d (RV64)** **Pre-update/Post-update**	d

Comparison

	Alibaba T-Head	Qualcomm	LD/SD (RV32)
Codepoints per variant	2^{17}	2^{15}	2^{22}
Variants	5	33	2
% SROS per variant	0.0163%	0.0041%	0.5208%
% SROS total	0.0815%	0.1353%	1.0416%
Implicit offset shift	2*data size (aligned to pair)	data size (aligned to single)	0
Offset bits	2	5	12
Offset reach (doubleword)	64 bytes	256 bytes	4096 bytes
SPEC 2006, RVA23 clang 16, -O3 static code size reduction		1.98% Avg 5.51% Max	
% Avg Reduction / % SROS		14.63	

Load/store pair semantics

- Ideally, consistent semantics across all pair instructions are consistent
- Exceptions:
    - Precise, atomic (both pairs occur or neither does)
    - *tval written with address causing fault (could be either address)
- Consistency:
    - Each load/store in the pair is independent, can be reordered in global order
    - Non-idempotent memory: implementations have option to trap. If no trap, each load/store must only be performed once (exceptions resolved ahead of time)

Instruction database format

- Presented Google Sheet format last meeting
- Text format suggested to manage concurrent work. See prototype
    - Instruction data specified in YAML files.
    - Vendors can be separate.
    - Script aggregates into Asciidoc table.

Processor classes

- See Draft