Example

Andrew Clark

June 26, 2022

1 Table of Contents

Contents

2 General Information

- The assembly language for `volt32_cpu` is intended to be that of a memory-to-memory machine without the awkwardness of instruction LARs. Data LARs machines approximate memory-to-memory machines, and GCC (and probably LLVM) supports arbitrary addressing modes. The problem with attempting to use GCC for a DLARs machine is that multiple DLARs may update at the same time if they point to the same address, assuming the compiler doesn't know that they point to the same address.
- As a side note, there's an LLVM backend targeting the 6502, which basically uses the first 256 bytes of RAM as extra registers. Since the first 256 bytes of memory can be pointed to, this means that LLVM could probably deal with multiple DLARs that point to the same address updating at the same time.

3 Instruction LARs

- `fetch`
  - At some performance cost, it's pretty cheap to be excessive and always insert after branches and on alignment boundaries, or slightly clever and run a pessimistic heuristic cache policy in the assembler, since spurious loads (fetches) are almost free unless you evict something you needed.