Contributing to `std::simd`

Simple version:

1. Fork it and `git clone` it
2. Create your feature branch: `git checkout -b my-branch`
3. Write your changes.
4. Test it: `cargo test` . Remember to enable whatever SIMD features you intend to test by setting `RUSTFLAGS` .
5. Commit your changes: `git commit add ./path/to/changes && git commit -m 'Fix some bug'`
6. Push the branch: `git push --set-upstream origin my-branch`
7. Submit a pull request!

Taking on an Issue

SIMD can be quite complex, and even a "simple" issue can be huge. If an issue is organized like a tracking issue, with an itemized list of items that don't necessarily have to be done in a specific order, please take the issue one item at a time. This will help by letting work proceed apace on the rest of the issue. If it's a (relatively) small issue, feel free to announce your intention to solve it on the issue tracker and take it in one go!

CI

We currently use GitHub Actions which will automatically build and test your change in order to verify that `std::simd` 's portable API is, in fact, portable. If your change builds locally, but does not build in CI, this is likely due to a platform-specific concern that your code has not addressed. Please consult the build logs and address the error, or ask for help if you need it.

Beyond stdsimd

A large amount of the core SIMD implementation is found in the rustc_codegen_* crates in the [main rustc repo](#). In addition, actual platform-specific functions are implemented in [stdarch](#). Not all changes to `std::simd` require interacting with either of these, but if you're wondering where something is and it doesn't seem to be in this repository, those might be where to start looking.

Questions? Concerns? Need Help?

Please feel free to ask in the [#project-portable-simd](#) stream on the [rust-lang Zulip](#) for help with making changes to `std::simd` ! If your changes include directly modifying the compiler, it might also be useful to ask in [#t-compiler/help](#).