Cranelift codegen backend for rust

The goal of this project is to create an alternative codegen backend for the rust compiler based on Cranelift. This has the potential to improve compilation times in debug mode. If your project doesn't use any of the things listed under "Not yet supported", it should work fine. If not please open an issue.

Building and testing

```
$ git clone https://github.com/bjorn3/rustc_codegen_cranelift.git
$ cd rustc_codegen_cranelift
$ ./y.rs prepare # download and patch sysroot src and install hyperfine for benchmarking
$ ./y.rs build
```

To run the test suite replace the last command with:

```
$ ./test.sh
```

This will implicitly build cg_clif too. Both `y.rs build` and `test.sh` accept a `--debug` argument to build in debug mode.

Alternatively you can download a pre built version from GHA. It is listed in the artifacts section of workflow runs. Unfortunately due to GHA restrictions you need to be logged in to access it.

Usage

rustc_codegen_cranelift can be used as a near-drop-in replacement for `cargo build` or `cargo run` for existing projects.

Assuming `$cg_clif_dir` is the directory you cloned this repo into and you followed the instructions (`y.rs prepare` and `y.rs build` or `test.sh`).

In the directory with your project (where you can do the usual `cargo build`), run:

```
$ $cg_clif_dir/build/cargo-clif build
```

This will build your project with rustc_codegen_cranelift instead of the usual LLVM backend.

For additional ways to use rustc_codegen_cranelift like the JIT mode see usage.md.

Configuration

See the documentation on the `BackendConfig` struct in config.rs for all configuration options.

Not yet supported

- Inline assembly (no cranelift support)
  - On Linux there is support for invoking an external assembler for `global_asm!` and `asm!` . `llvm_asm!` will remain unimplemented forever. `asm!` doesn't yet support reg classes. You

have to specify specific registers instead.

- SIMD (tracked here, some basic things work)

License

Licensed under either of

- Apache License, Version 2.0 (LICENSE-APACHE or http://www.apache.org/licenses/LICENSE-2.0)
- MIT license (LICENSE-MIT or http://opensource.org/licenses/MIT)

at your option.

Contribution

Unless you explicitly state otherwise, any contribution intentionally submitted for inclusion in the work by you shall be dual licensed as above, without any additional terms or conditions.