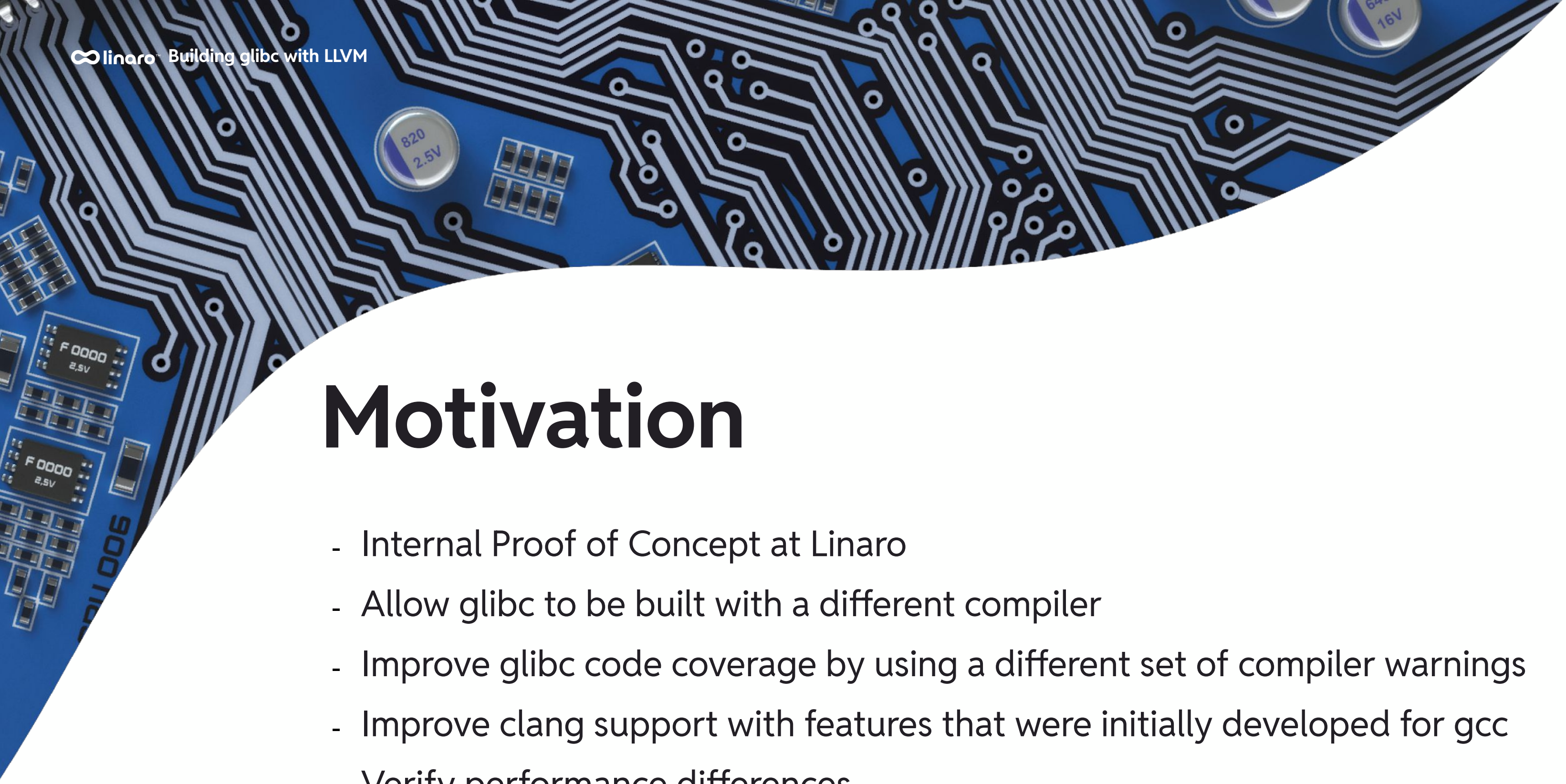
Building glibc with LLVM

linaro™

Carlos Seo

Arm Solutions at Lightspeed



Motivation

- Internal Proof of Concept at Linaro
- Allow glibc to be built with a different compiler
- Improve glibc code coverage by using a different set of compiler warnings
- Improve clang support with features that were initially developed for gcc
- Verify performance differences
- Open possibility to build a full Linux distribution with LLVM in the future

Challenges

- glibc uses a GNU C standard — relies on gcc extensions
- glibc assumes binutils as the assembler and static linker
- Some GNU extensions will not be implemented by LLVM
- Other extensions are not fully compatible between compilers

Status

- Removed the usage of GNU extensions that LLVM will not implement
- Adapted glibc to use a different static linker
- Refactored code to avoid binutils and gcc-specific extensions

PoC available in an out-of-tree branch at sourceware.org (`azanella/clang` glibc branch)

- ⅓ of mapped work to build for `x86_64` and `aarch64`
- ⅔ of mapped work to build all test cases
- Latest testing using LLVM 18 and gcc 11 runtime

Limitations

- IFUNC support not complete
- gmon on `aarch64` not working
- Codegen issues in some math tests, mainly for `long double` and `_Float128`
- No support for the LLVM runtime

To Do

- Add extra support in compiler-rt/libunwind
  - Missing some symbols
  - Missing floating point rounding modes
  - All symbols are public

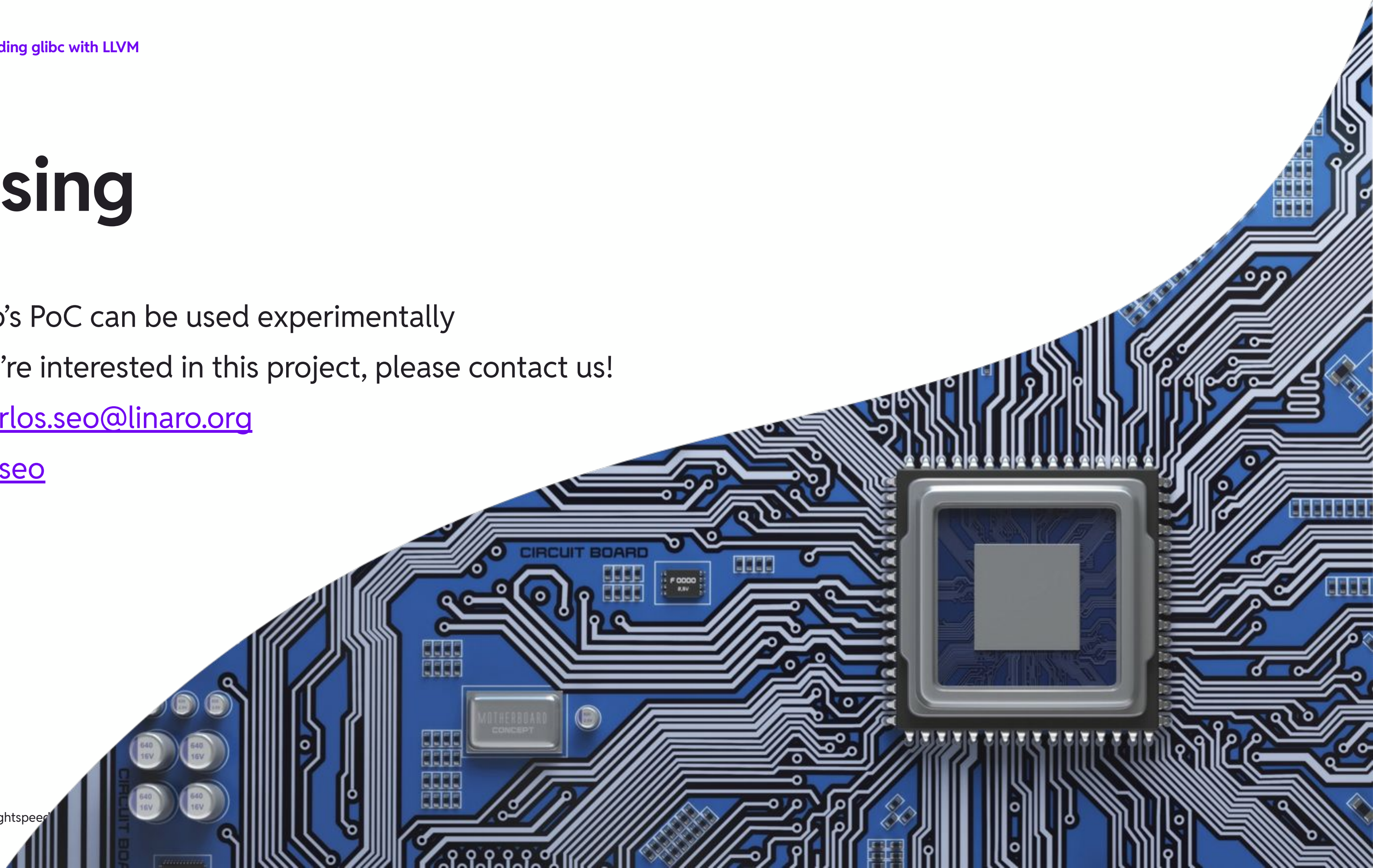
- Solve the remaining limitations on the glibc side

Closing

- Linaro’s PoC can be used experimentally
- If you’re interested in this project, please contact us!

✉ carlos.seo@linaro.org

ceseo



Thank you!