

Factsheet



CompCert C Compiler

Release 24.04i, b15309220

April 25, 2024



CompCert is an optimizing C compiler which is formally verified, using machine-assisted mathematical proofs, to guarantee the absence of compiler bugs.

Its intended use is compiling safety-critical and mission-critical software written in C and meeting high levels of assurance. The code it produces is proved to behave exactly as specified by the semantics of the source C program.

The formal proof covers all transformations from the abstract syntax tree to the generated assembly code.

Key benefits

- The correctness proof of CompCert guarantees that all safety properties verified on the source code (e.g. by static analyzers or model checking) automatically hold for the generated code as well.
- On typical embedded processors, code generated by CompCert offers performance comparable to code generated by *GCC* at optimization level 1.

Optimizations

CompCert implements the following optimizations, all formally verified:

- Register allocation using graph coloring and iterated register coalescing.
- Instruction selection with strength reduction, to take advantage of combined instructions provided by the target architectures.
- Constant propagation, for integer and floating point types.
- Common subexpression elimination.
- Dead and redundant code elimination.
- Function inlining.
- Tail call elimination.
- If-conversion.

Supported C dialect

CompCert supports nearly all of the ISO C 2011, ISO C 1999, and ANSI C language standards including:

- The `_Alignof` operator and the `_Alignas` attribute from ISO C 2011.
- Anonymous structures and unions from ISO C 2011.
- Static assertions via the `_Static_assert` keyword from ISO C 2011.
- Unicode string literals and character constants.
- Pragmas and attributes to control alignment and section placement of global variables.

Supported targets

- PowerPC 32-bit
- PowerPC VLE 32-bit (Signal Processing Extension *SPE* and Variable Length Encoding *VLE*)
- PowerPC 32-/64-bit hybrid (32-bit pointers, 64-bit integer computations)
- ARMv6 ISA with VFPv2 coprocessor (big or little endian)
- ARMv7 ISA with VFPv3-D16 coprocessor (big or little endian)
- AArch64 (ARMv8 ISA, 64-bit, little endian)
- ia32 (x86 32-bit, SSE2 extension required)
- AMD64 (x86 64-bit)
- RISC-V (Base instruction sets *RV32I* and *RV64I*; extensions *M*, *F*, and *D*)

Supported target systems are *Linux* with the GNU toolchain for all architectures. For ARM 32-bit *EABI* is supported with the GNU toolchain. Additionally, for PowerPC 32-bit *EABI* is supported with Diab and GNU toolchains, for PowerPC VLE 32-bit *EABI* is supported with the NXP-GCC toolchain. For RISC-V *ELF psABI* with *ILP32D* or *LP64D* calling convention is supported.



Supported tool chains

To preprocess and produce object and executable files, an external C preprocessor, assembler and linker have to be provided. CompCert is currently tested for compatibility with:

- GNU Compiler Collection version 3 or newer, version 4 or 5 is recommended
- Wind River Diab Compiler version 5 for 32-bit PowerPC target architectures

Availability

- The source code and documentation of CompCert, including all proofs, can be downloaded from compcert.inria.fr or github.com/AbsInt/CompCert. For research purposes, usage of CompCert is free of charge.
- AbsInt offers commercial licenses, provides industrial-strength support and maintenance, and contributes to the advancement of the tool.

More information

- Visit our website: www.absint.com
- Speak with a product specialist:
call +49 681 383 600

About AbsInt

AbsInt provides advanced development tools for embedded systems, and tools for analysis, optimization and verification of safety-critical software. Our customers are located in more than 40 countries worldwide. We have distribution agreements with major software distributors in Asia, North America, Middle East, and throughout Europe.

Our headquarters

Science Park 1
66123 Saarbrücken, Germany
Phone: +49 681 383 600
Fax: +49 681 383 60 20
Email: info@absint.com
Web: www.absint.com