Clang command line argument reference

Introduction

This page lists the command line arguments currently supported by the GCC-compatible `clang` and `clang++` drivers.

-B<prefix>, **--prefix** <arg>, **--prefix**=<arg>

Search $prefix$file for executables, libraries, and data files. If $prefix is a directory, search $prefix/$file

-F<arg>

Add directory to framework include search path

-K

-ObjC

Treat source input files as Objective-C inputs

-ObjC++

Treat source input files as Objective-C++ inputs

-Qn, **-fno-ident**

Do not emit metadata containing compiler name and version

-Qunused-arguments

Don’t emit warning for unused driver arguments

-Qy, **-fident**

Emit metadata containing compiler name and version

-Wa,<arg>,<arg2>...

Pass the comma separated arguments in <arg> to the assembler

-Wlarge-by-value-copy=<arg>

-Xarch_<arg1> <arg2>

-Xarch_device <arg>

Pass <arg> to the CUDA/HIP device compilation

-Xarch_host <arg>

Pass <arg> to the CUDA/HIP host compilation

-Xcuda-fatbinary <arg>

Pass <arg> to fatbinary invocation

-Xcuda-ptxas <arg>

Pass <arg> to the ptxas assembler

-alias_list <arg>

-all_load

-allowable_client <arg>

--analyze

Run the static analyzer

--analyzer-no-default-checks

--analyzer-output<arg>

Static analyzer report output format (html|plist|plist-multi-file|plist-html|sarif|sarif-html|text).

-arch <arg>

-arch_errors_fatal

-arch_only <arg>

-arcmt-migrate-emit-errors

Emit ARC errors even if the migrator can fix them

-arcmt-migrate-report-output <arg>

Output path for the plist report

--autocomplete=<arg>

-bind_at_load

-bundle

-bundle_loader <arg>

-client_name<arg>

-compatibility_version<arg>

--config=<file>, **--config** <arg>

Specify configuration file

--constant-cfstrings

-current_version<arg>

-darwin-target-variant <arg>

Generate code for an additional runtime variant of the deployment target

-darwin-target-variant-triple <arg>

Specify the darwin target variant triple

-dead_strip

-dependency-dot <arg>

Filename to write DOT-formatted header dependencies to

-dependency-file <arg>

Filename (or -) to write dependency output to

-dsym-dir<dir>

Directory to output dSYM’s (if any) to

-dumpdir <dumppfx>

Use <dumpfpx> as a prefix to form auxiliary and dump file names

-dumpmachine

Display the compiler’s target processor

-dumpversion

Display the version of the compiler

--dyld-prefix=<arg>, **--dyld-prefix** <arg>

-dylib_file <arg>

-dylinker

-dylinker_install_name<arg>

-dynamic

-dynamiclib

-emit-ast

Emit Clang AST files for source inputs

--emit-extension-symbol-graphs

Generate additional symbol graphs for extended modules.

--emit-static-lib

Enable linker job to emit a static library.

-emit-symbol-graph

Generate Extract API information as a side effect of compilation.

--end-no-unused-arguments

Start emitting warnings for unused driver arguments

-exported_symbols_list <arg>

--extract-api-ignores=<arg1>,<arg2>...

Comma separated list of files containing a new line separated list of API symbols to ignore when extracting API information.

-faligned-new=<arg>

-fautomatic

-fcheck-new, **-fno-check-new**

Do not assume C++ operator new may not return NULL

-fcx-fortran-rules, **-fno-cx-fortran-rules**

Range reduction is enabled for complex arithmetic operations.

-fcx-limited-range, **-fno-cx-limited-range**

Basic algebraic expansions of complex arithmetic operations involving are enabled.

-fheinous-gnu-extensions

-flat_namespace

-fmodule-output

Save intermediate module file results when compiling a standard C++ module unit.

-fmodule-output=<arg>

Save intermediate module file results when compiling a standard C++ module unit.

-fopenmp-targets=<arg1>,<arg2>...

Specify comma-separated list of triples OpenMP offloading targets to be supported

`-force_cpusubtype_ALL`

`-force_flat_namespace`

`-force_load` <arg>

`-fplugin-arg-<name>-<arg>`

Pass <arg> to plugin <name>

`-framework` <arg>

`-frtlib-add-rpath`, `-fno-rtlib-add-rpath`, `--no-offload-add-rpath`, `--offload-add-rpath`

Add -rpath with architecture-specific resource directory to the linker flags. When –hip-link is specified, also add -rpath with HIP runtime library directory to the linker flags

`-fsanitize-system-ignorelist`=<arg>

Path to system ignorelist file for sanitizers

`-fshow-skipped-includes`

#include files may be “skipped” due to include guard optimization

> or #pragma once. This flag makes -H show also such includes.

`-fsystem-module`

Build this module as a system module. Only used with -emit-module

`--gcc-install-dir`=<arg>

Use GCC installation in the specified directory. The directory ends with path components like ‘lib{,32,64}/gcc{,-cross}/$triple/$version’. Note: executables (e.g. ld) used by the compiler are not overridden by the selected GCC installation

`--gcc-toolchain`=<arg>

Specify a directory where Clang can find ‘include’ and ‘lib{,32,64}/gcc{,-cross}/$triple/$version’. Clang will use the GCC installation with the largest version

`--gcc-triple`=<arg>

Search for the GCC installation with the specified triple.

`-gcodeview`

Generate CodeView debug information

`-gcodeview-command-line`, `-gno-codeview-command-line`

Emit compiler path and command line into CodeView debug information

`-gcodeview-ghash`, `-gno-codeview-ghash`

Emit type record hashes in a .debug$H section

`-gen-reproducer`=<arg>, `-fno-crash-diagnostics` (equivalent to -gen-reproducer=off)

Emit reproducer on (option: off, crash (default), error, always)

`-gpulibc`

Link the LLVM C Library for GPUs

-headerpad_max_install_names<arg>

-help, **--help**

Display available options

--help-hidden

Display help for hidden options

-image_base <arg>

-index-header-map

Make the next included directory (-I or -F) an indexer header map

-init <arg>

-install_name <arg>

-interface-stub-version=<arg>

-keep_private_externs

-lazy_framework <arg>

-lazy_library <arg>

--migrate

Run the migrator

-mllvm <arg>, **-mllvm**=<arg>

Additional arguments to forward to LLVM’s option processing

-mmlir <arg>

Additional arguments to forward to MLIR’s option processing

-module-dependency-dir <arg>

Directory to dump module dependencies to

-mtvos-simulator-version-min=<arg>, **-mappletvsimulator-version-min**=<arg>

-multi_module

-multiply_defined <arg>

-multiply_defined_unused <arg>

-mzos-hlq-clang=<ClangHLQ>

High level qualifier for z/OS C++RT side deck datasets

-mzos-hlq-csslib=<CsslibHLQ>

High level qualifier for z/OS CSSLIB dataset

-mzos-hlq-le=<LeHLQ>

High level qualifier for z/OS Language Environment datasets

-mzos-sys-include=<SysInclude>

Path to system headers on z/OS

--no-default-config

Disable loading default configuration files

-no-integrated-cpp, **--no-integrated-cpp**

-no_dead_strip_inits_and_terms

-nodefaultlibs

-nodriverkitlib

-nofixprebinding

-nogpulib, **-nocudalib**

Do not link device library for CUDA/HIP device compilation

-nogpulibc

-nolibc

-nomultidefs

-nopie

-noprebind

-noprofilelib

-noseglinkedit

-nostdlib++

-o<file>, **--output** <arg>, **--output**=<arg>

Write output to <file>

-objcmt-allowlist-dir-path=<arg>, **-objcmt-white-list-dir-path**=<arg>,
-objcmt-whitelist-dir-path=<arg>

Only modify files with a filename contained in the provided directory path

-objcmt-atomic-property

Make migration to ‘atomic’ properties

-objcmt-migrate-all

Enable migration to modern ObjC

-objcmt-migrate-annotation

Enable migration to property and method annotations

-objcmt-migrate-designated-init

Enable migration to infer NS_DESIGNATED_INITIALIZER for initializer methods

-objcmt-migrate-instancetype

Enable migration to infer instancetype for method result type

-objcmt-migrate-literals

Enable migration to modern ObjC literals

-objcmt-migrate-ns-macros

Enable migration to NS_ENUM/NS_OPTIONS macros

-objcmt-migrate-property

Enable migration to modern ObjC property

-objcmt-migrate-property-dot-syntax

Enable migration of setter/getter messages to property-dot syntax

-objcmt-migrate-protocol-conformance

Enable migration to add protocol conformance on classes

-objcmt-migrate-readonly-property

Enable migration to modern ObjC readonly property

-objcmt-migrate-readwrite-property

Enable migration to modern ObjC readwrite property

-objcmt-migrate-subscripting

Enable migration to modern ObjC subscripting

-objcmt-ns-nonatomic-iosonly

Enable migration to use NS_NONATOMIC_IOSONLY macro for setting property's 'atomic' attribute

-objcmt-returns-innerpointer-property

Enable migration to annotate property with NS_RETURNS_INNER_POINTER

-object

-object-file-name=<file>, **-object-file-name** <arg>

Set the output <file> for debug infos

--offload=<arg1>,<arg2>...

Specify comma-separated list of offloading target triples (CUDA and HIP only)

-p, **--profile**

Enable mcount instrumentation with prof

-pagezero_size<arg>

-pg

Enable mcount instrumentation

-pipe, **--pipe**

Use pipes between commands, when possible

-prebind

-prebind_all_twolevel_modules

-preload

--pretty-sgf

Emit pretty printed symbol graphs

--print-diagnostic-categories

-print-diagnostic-options, **--print-diagnostic-options**

Print all of Clang’s warning options

-print-effective-triple, **--print-effective-triple**

Print the effective target triple

-print-file-name=<file>, **--print-file-name**=<file>, **--print-file-name** <arg>

Print the full library path of <file>

-print-ivar-layout

Enable Objective-C Ivar layout bitmap print trace

-print-libgcc-file-name, **--print-libgcc-file-name**

Print the library path for the currently used compiler runtime library (“libgcc.a” or “libclang_rt.builtins.*.a”)

-print-library-module-manifest-path, **--print-library-module-manifest-path**

Print the path for the C++ Standard library module manifest

-print-multi-directory, **--print-multi-directory**

-print-multi-flags-experimental, **--print-multi-flags-experimental**

Print the flags used for selecting multilibs (experimental)

-print-multi-lib, **--print-multi-lib**

-print-prog-name=<name>, **--print-prog-name**=<name>, **--print-prog-name** <arg>

Print the full program path of <name>

-print-resource-dir, **--print-resource-dir**

Print the resource directory pathname

-print-rocm-search-dirs, **--print-rocm-search-dirs**

Print the paths used for finding ROCm installation

-print-runtime-dir, **--print-runtime-dir**

Print the directory pathname containing Clang’s runtime libraries

-print-search-dirs, **--print-search-dirs**

Print the paths used for finding libraries and programs

-print-supported-extensions, **--print-supported-extensions**

Print supported -march extensions (RISC-V, AArch64 and ARM only)

-print-target-triple, **--print-target-triple**

Print the normalized target triple

-print-targets, **--print-targets**

Print the registered targets

-private_bundle

--product-name=<arg>

-pthread, **-no-pthread**

Support POSIX threads in generated code

-pthreads

-read_only_relocs <arg>

-reexport-l<arg>

-reexport_framework <arg>

-reexport_library<arg>

-relocatable-pch, **--relocatable-pch**

Whether to build a relocatable precompiled header

-remap

-rewrite-legacy-objc

Rewrite Legacy Objective-C source to C++

-rtlib=<arg>, **--rtlib**=<arg>, **--rtlib** <arg>

Compiler runtime library to use

-save-stats=<arg>, **--save-stats**=<arg>, **-save-stats** (equivalent to -save-stats=cwd), **--save-stats** (equivalent to -save-stats=cwd)

Save llvm statistics.

-save-temps=<arg>, **--save-temps**=<arg>, **-save-temps** (equivalent to -save-temps=cwd), **--save-temps** (equivalent to -save-temps=cwd)

Save intermediate compilation results. <arg> can be set to ‘cwd’ for current working directory, or ‘obj’ which will save temporary files in the same directory as the final output file

-sectalign <arg1> <arg2> <arg3>

-sectcreate <arg1> <arg2> <arg3>

-sectobjectsymbols <arg1> <arg2>

-sectorder <arg1> <arg2> <arg3>

-seg1addr<arg>

-seg_addr_table <arg>

-seg_addr_table_filename <arg>

-segaddr <arg1> <arg2>

-segcreate <arg1> <arg2> <arg3>

-seglinkedit

-segprot <arg1> <arg2> <arg3>

-segs_read_<arg>

-segs_read_only_addr <arg>

-segs_read_write_addr <arg>

-serialize-diagnostics <arg>, **--serialize-diagnostics** <arg>

Serialize compiler diagnostics to a file

-shared-libgcc

-shared-libsan, **-shared-libasan**

Dynamically link the sanitizer runtime

-single_module

--start-no-unused-arguments

Don’t emit warnings about unused arguments for the following arguments

-static-libgcc

-static-libsan

Statically link the sanitizer runtime (Not supported for ASan, TSan or UBSan on darwin)

-static-libstdc++

-static-openmp

Use the static host OpenMP runtime while linking.

-std-default=<arg>

-stdlib=<arg>, **--stdlib**=<arg>, **--stdlib** <arg>

C++ standard library to use. <arg> must be ‘libc++’, ‘libstdc++’ or ‘platform’.

-sub_library<arg>

-sub_umbrella<arg>

--symbol-graph-dir=<arg>

Directory in which to emit symbol graphs.

--sysroot=<arg>, **--sysroot** <arg>

--target-help

--target=<arg>, **-target** <arg>

Generate code for the given target

-time

Time individual commands

-traditional, **--traditional**

-traditional-cpp, **--traditional-cpp**

Enable some traditional CPP emulation

-twolevel_namespace

-twolevel_namespace_hints

-umbrella <arg>

-unexported_symbols_list <arg>

-unwindlib=<arg>, **--unwindlib**=<arg>

Unwind library to use. <arg> must be ‘libgcc’, ‘unwindlib’ or ‘platform’.

-v, **--verbose**

Show commands to run and use verbose output

--verify-debug-info

Verify the binary representation of debug output

--version

Print version information

-vfsoverlay<arg>, **--vfsoverlay<arg>**

Overlay the virtual filesystem described by file over the real file system. Additionally, pass this overlay file to the linker if it supports it

-w, **--no-warnings**

Suppress all warnings

-weak-l<arg>

-weak_framework <arg>

-weak_library <arg>

-weak_reference_mismatches <arg>

-whatsloaded

-why_load, **-whyload**

-working-directory <arg>, **-working-directory**=<arg>

Resolve file paths relative to the specified directory

-x<language>, **--language** <arg>, **--language**=<arg>

Treat subsequent input files as having type <language>

-y<arg>

Actions

The action to perform on the input.

-E, **--preprocess**

Only run the preprocessor

-S, **--assemble**

Only run preprocess and compilation steps

-c, **--compile**

Only run preprocess, compile, and assemble steps

-emit-interface-stubs

Generate Interface Stub Files.

-emit-llvm

Use the LLVM representation for assembler and object files

-emit-merged-ifs

Generate Interface Stub Files, emit merged text not binary.

-extract-api

Extract API information

-fdriver-only

Only run the driver.

-fsyntax-only

Run the preprocessor, parser and semantic analysis stages

-module-file-info

Provide information about a particular module file

--precompile

Only precompile the input

-rewrite-objc

Rewrite Objective-C source to C++

-verify-pch

Load and verify that a pre-compiled header file is not stale

Compilation options

Flags controlling the behavior of Clang during compilation. These flags have no effect during actions that do not perform compilation.

-Xassembler <arg>

Pass <arg> to the assembler

-Xclang <arg>, **-Xclang**=<arg>

Pass <arg> to clang -cc1

-Xopenmp-target <arg>

Pass <arg> to the target offloading toolchain.

-Xopenmp-target=<triple> <arg>

Pass <arg> to the target offloading toolchain identified by <triple>.

-ansi, **--ansi**

-fapinotes, **-fno-apinotes**

Enable external API notes support

-fapinotes-modules, **-fno-apinotes-modules**

Enable module-based external API notes support

-fapinotes-swift-version=<version>

Specify the Swift version to use when filtering API notes

-fc++-abi=<arg>

C++ ABI to use. This will override the target C++ ABI.

-fclang-abi-compat=<version>

Attempt to match the ABI of Clang <version>. <version> must be '<major>.<minor>' or 'latest'.

-fcomment-block-commands=<arg>,<arg2>...

Treat each comma separated argument in <arg> as a documentation comment block command

-fcomplete-member-pointers, **-fno-complete-member-pointers**

Require member pointer base types to be complete if they would be significant under the Microsoft ABI

-fcrash-diagnostics-dir=<dir>

Put crash-report files in <dir>

-fcrash-diagnostics=<arg>, **-fcrash-diagnostics** (equivalent to -fcrash-diagnostics=compiler)

Set level of crash diagnostic reporting, (option: off, compiler, all)

-fdeclspec, **-fno-declspec**

Allow __declspec as a keyword

-fdepfile-entry=<arg>

-fdiagnostics-fixit-info, **-fno-diagnostics-fixit-info**

-fdiagnostics-format=<arg>

-fdiagnostics-parseable-fixits

Print fix-its in machine parseable form

-fdiagnostics-print-source-range-info

Print source range spans in numeric form

-fdiagnostics-show-category=<arg>

-fdiscard-value-names, **-fno-discard-value-names**

Discard value names in LLVM IR

-fexperimental-relative-c++-abi-vtables, **-fno-experimental-relative-c++-abi-vtables**

Use the experimental C++ class ABI for classes with virtual tables

-fexperimental-strict-floating-point

Enables the use of non-default rounding modes and non-default exception handling on targets that are not currently ready.

-ffine-grained-bitfield-accesses, **-fno-fine-grained-bitfield-accesses**

Use separate accesses for consecutive bitfield runs with legal widths and alignments.

-fglobal-isel, **-fexperimental-isel**, **-fno-global-isel**

Enables the global instruction selector

-finline-functions, **-fno-inline-functions**

Inline suitable functions

-finline-hint-functions

Inline functions which are (explicitly or implicitly) marked inline

-fno-sanitize-ignorelist

Don’t use ignorelist file for sanitizers

-fparse-all-comments

-frandomize-layout-seed-file=<file>

File holding the seed used by the randomize structure layout feature

-frandomize-layout-seed=<seed>

The seed used by the randomize structure layout feature

-frecord-command-line, **-fno-record-command-line**, **-frecord-gcc-switches**

Generate a section named “.GCC.command.line” containing the clang driver command-line. After linking, the section may contain multiple command lines, which will be individually terminated by null bytes. Separate arguments within a command line are combined with spaces; spaces and backslashes within an argument are escaped with backslashes. This format differs from the format of the equivalent section produced by GCC with the -frecord-gcc-switches flag. This option is currently only supported on ELF targets.

-fsanitize-address-destructor=<arg>

Set the kind of module destructors emitted by AddressSanitizer instrumentation. These destructors are emitted to unregister instrumented global variables when code is unloaded (e.g. via `dlclose()`). <arg> must be ‘none’ or ‘global’.

-fsanitize-address-field-padding=<arg>

Level of field padding for AddressSanitizer

-fsanitize-address-globals-dead-stripping, **-fno-sanitize-address-globals-dead-stripping**

Enable linker dead stripping of globals in AddressSanitizer

-fsanitize-address-outline-instrumentation, **-fno-sanitize-address-outline-instrumentation**

Always generate function calls for address sanitizer instrumentation

-fsanitize-address-poison-custom-array-cookie, **-fno-sanitize-address-poison-custom-array-cookie**

Enable “poisoning” array cookies when allocating arrays with a custom operator new[] in Address Sanitizer, preventing accesses to the cookies from user code. An array cookie is a small implementation-defined header added to certain array allocations to record metadata such as the length of the array. Accesses to array cookies from user code are technically allowed by the standard but are more likely to be the result of an out-of-bounds array access.

An operator new[] is “custom” if it is not one of the allocation functions provided by the C++ standard library. Array cookies from non-custom allocation functions are always poisoned.

-fsanitize-address-use-after-return=<mode>

Select the mode of detecting stack use-after-return in AddressSanitizer. <mode> must be ‘never’, ‘runtime’ or ‘always’.

-fsanitize-address-use-after-scope, **-fno-sanitize-address-use-after-scope**

Enable use-after-scope detection in AddressSanitizer

-fsanitize-address-use-odr-indicator, **-fno-sanitize-address-use-odr-indicator**

Enable ODR indicator globals to avoid false ODR violation reports in partially sanitized programs at the cost of an increase in binary size

-fsanitize-cfi-canonical-jump-tables, **-fno-sanitize-cfi-canonical-jump-tables**

Make the jump table addresses canonical in the symbol table

-fsanitize-cfi-cross-dso, **-fno-sanitize-cfi-cross-dso**

Enable control flow integrity (CFI) checks for cross-DSO calls.

-fsanitize-cfi-icall-experimental-normalize-integers

Normalize integers in CFI indirect call type signature checks

-fsanitize-cfi-icall-generalize-pointers

Generalize pointers in CFI indirect call type signature checks

-fsanitize-coverage-allowlist=<arg>

Restrict sanitizer coverage instrumentation exclusively to modules and functions that match the provided special case list, except the blocked ones

-fsanitize-coverage-ignorelist=<arg>

Disable sanitizer coverage instrumentation for modules and functions that match the provided special case list, even the allowed ones

-fsanitize-coverage=<arg1>,<arg2>..., **-fno-sanitize-coverage**=<arg1>,<arg2>...

Specify the type of coverage instrumentation for Sanitizers

-fsanitize-hwaddress-abi=<arg>

Select the HWAddressSanitizer ABI to target (interceptor or platform, default interceptor). This option is currently unused.

-fsanitize-hwaddress-experimental-aliasing, **-fno-sanitize-hwaddress-experimental-aliasing**

Enable aliasing mode in HWAddressSanitizer

-fsanitize-ignorelist=<arg>

Path to ignorelist file for sanitizers

-fsanitize-link-c++-runtime, **-fno-sanitize-link-c++-runtime**

-fsanitize-link-runtime, **-fno-sanitize-link-runtime**

-fsanitize-memory-track-origins=<arg>, **-fsanitize-memory-track-origins** (equivalent to -fsanitize-memory-track-origins=2)

Enable origins tracking in MemorySanitizer

-fsanitize-memory-use-after-dtor, **-fno-sanitize-memory-use-after-dtor**

Enable use-after-destroy detection in MemorySanitizer

-fsanitize-memtag-mode=<arg>

Set default MTE mode to ‘sync’ (default) or ‘async’

-fsanitize-minimal-runtime, **-fno-sanitize-minimal-runtime**

-fsanitize-recover=<arg1>,<arg2>..., **-fno-sanitize-recover**=<arg1>,<arg2>..., **-fsanitize-recover** (equivalent to -fsanitize-recover=all)

Enable recovery for specified sanitizers

-fsanitize-stats, **-fno-sanitize-stats**

Enable sanitizer statistics gathering.

-fsanitize-thread-atomics, **-fno-sanitize-thread-atomics**

Enable atomic operations instrumentation in ThreadSanitizer (default)

-fsanitize-thread-func-entry-exit, **-fno-sanitize-thread-func-entry-exit**

Enable function entry/exit instrumentation in ThreadSanitizer (default)

-fsanitize-thread-memory-access, **-fno-sanitize-thread-memory-access**

Enable memory access instrumentation in ThreadSanitizer (default)

-fsanitize-trap=<arg1>,<arg2>..., **-fno-sanitize-trap**=<arg1>,<arg2>..., **-fsanitize-trap** (equivalent to -fsanitize-trap=all), **-fsanitize-undefined-trap-on-error** (equivalent to -fsanitize-trap=undefined)

Enable trapping for specified sanitizers

-fsanitize-undefined-strip-path-components=<number>

Strip (or keep only, if negative) a given number of path components when emitting check metadata.

-fsanitize=<check>,<arg2>..., **-fno-sanitize**=<arg1>,<arg2>...

Turn on runtime checks for various forms of undefined or suspicious behavior. See user manual for available checks

-fverify-intermediate-code, **-fno-verify-intermediate-code**

Enable verification of LLVM IR

-mno-fmv

Disable function multiversioning

-moutline, **-mno-outline**

Enable function outlining (AArch64 only)

-moutline-atomics, **-mno-outline-atomics**

Generate local calls to out-of-line atomic operations

--param <arg>, **--param**=<arg>

-print-supported-cpus, **--print-supported-cpus**, **-mcpu**=help, **-mtune**=help

Print supported cpu models for the given target (if target is not specified, it will print the supported cpus for the default target)

-std=<arg>, **--std**=<arg>, **--std** <arg>

Language standard to compile for

Preprocessor options

Flags controlling the behavior of the Clang preprocessor.

-C, **--comments**

Include comments in preprocessed output

-CC, **--comments-in-macros**

Include comments from within macros in preprocessed output

-D<macro>=<value>, **--define-macro** <arg>, **--define-macro**=<arg>

Define <macro> to <value> (or 1 if <value> omitted)

-H, **--trace-includes**

Show header includes and nesting depth

-P, **--no-line-commands**

Disable linemarker output in -E mode

-U<macro>, **--undefine-macro** <arg>, **--undefine-macro**=<arg>

Undefine macro <macro>

-Wp,<arg>,<arg2>...

Pass the comma separated arguments in <arg> to the preprocessor

-Xpreprocessor <arg>

Pass <arg> to the preprocessor

Include path management

Flags controlling how `#include`s are resolved to files.

-I<dir>, **--include-directory** <arg>, **--include-directory**=<arg>

Add directory to include search path. For C++ inputs, if there are multiple -I options, these directories are searched in the order they are given before the standard system directories are searched. If the same directory is in the SYSTEM include search paths, for example if also specified with -isystem, the -I option will be ignored

-I-, **--include-barrier**

Restrict all prior -I flags to double-quoted inclusion and remove current directory from include path

-cxx-isystem<directory>

Add directory to the C++ SYSTEM include search path

-fbuild-session-file=<file>

Use the last modification time of <file> as the build session timestamp

-fbuild-session-timestamp=<time since Epoch in seconds>

Time when the current build session started

-fmodule-file=\[<name>=\]<file>

Specify the mapping of module name to precompiled module file, or load a module file if name is omitted.

-fmodules-cache-path=<directory>

Specify the module cache path

-fmodules-disable-diagnostic-validation

Disable validation of the diagnostic options when loading the module

-fmodules-prune-after=<seconds>

Specify the interval (in seconds) after which a module file will be considered unused

-fmodules-prune-interval=<seconds>

Specify the interval (in seconds) between attempts to prune the module cache

-fmodules-user-build-path <directory>

Specify the module user build path

-fmodules-validate-once-per-build-session

Don’t verify input files for the modules if the module has been successfully validated or loaded during this build session

-fmodules-validate-system-headers, **-fno-modules-validate-system-headers**

Validate the system headers that a module depends on when loading the module

-fprebuilt-module-path=<directory>

Specify the prebuilt module path

-iapinotes-modules<directory>

Add directory to the API notes search path referenced by module name

-ibuiltininc

Enable builtin #include directories even when -nostdinc is used before or after -ibuiltininc. Using -nobuiltininc after the option disables it

-idirafter<arg>, **--include-directory-after** <arg>, **--include-directory-after**=<arg>

Add directory to AFTER include search path

-iframework<arg>

Add directory to SYSTEM framework search path

-iframeworkwithsysroot<directory>

Add directory to SYSTEM framework search path, absolute paths are relative to -isysroot

-imacros<file>, **--imacros<file>**, **--imacros**=<arg>

Include macros from file before parsing

-include<file>, **--include<file>**, **--include**=<arg>

Include file before parsing

-include-pch <file>

Include precompiled header file

-iprefix<dir>, **--include-prefix** <arg>, **--include-prefix**=<arg>

Set the -iwithprefix/-iwithprefixbefore prefix

-iquote<directory>

Add directory to QUOTE include search path

-isysroot<dir>

Set the system root directory (usually /)

-isystem<directory>

Add directory to SYSTEM include search path

-isystem-after<directory>

Add directory to end of the SYSTEM include search path

-ivfsoverlay<arg>

Overlay the virtual filesystem described by file over the real file system

-iwithprefix<dir>, **--include-with-prefix** <arg>, **--include-with-prefix-after** <arg>, **--include-with-prefix-after**=<arg>, **--include-with-prefix**=<arg>

Set directory to SYSTEM include search path with prefix

-iwithprefixbefore<dir>, **--include-with-prefix-before** <arg>, **--include-with-prefix-before**=<arg>

Set directory to include search path with prefix

-iwithsysroot<directory>

Add directory to SYSTEM include search path, absolute paths are relative to -isysroot

--libomptarget-amdgpu-bc-path=<arg>, **--libomptarget-amdgcn-bc-path**=<arg>

Path to libomptarget-amdgcn bitcode library

--libomptarget-nvptx-bc-path=<arg>

Path to libomptarget-nvptx bitcode library

-nobuiltininc

Disable builtin #include directories

-nogpuinc, **-nocudainc**

Do not add include paths for CUDA/HIP and do not include the default CUDA/HIP wrapper headers

-nohipwrapperinc

Do not include the default HIP wrapper headers and include paths

-nostdinc, **--no-standard-includes**

-nostdinc++

Disable standard #include directories for the C++ standard library

-nostdlibinc

-stdlib++-isystem<directory>

Use directory as the C++ standard library include path

--system-header-prefix=<prefix>, **--no-system-header-prefix**=<prefix>, **--system-header-prefix** <arg>

Treat all #include paths starting with <prefix> as including a system header.

Dependency file generation

Flags controlling generation of a dependency file for `make`-like build systems.

-M, **--dependencies**

Like -MD, but also implies -E and writes to stdout by default

-MD, **--write-dependencies**

Write a depfile containing user and system headers

-MF<file>

Write depfile output from -MMD, -MD, -MM, or -M to <file>

-MG, **--print-missing-file-dependencies**

Add missing headers to depfile

-MJ<arg>

Write a compilation database entry per input

-MM, **--user-dependencies**

Like -MMD, but also implies -E and writes to stdout by default

-MMD, **--write-user-dependencies**

Write a depfile containing user headers

-MP

Create phony target for each dependency (other than main file)

-MQ<arg>

Specify name of main file output to quote in depfile

-MT<arg>

Specify name of main file output in depfile

-MV

Use NMake/Jom format for the depfile

Dumping preprocessor state

Flags allowing the state of the preprocessor to be dumped in various ways.

-d

-d<arg>

-dD

Print macro definitions in -E mode in addition to normal output

-dI

Print include directives in -E mode in addition to normal output

-dM

Print macro definitions in -E mode instead of normal output

Diagnostic options

Flags controlling which warnings, errors, and remarks Clang will generate. See the **full list of warning and remark flags**.

-R<remark>

Enable the specified remark

-Rpass-analysis=<arg>

Report transformation analysis from optimization passes whose name matches the given POSIX regular expression

-Rpass-missed=<arg>

Report missed transformations by optimization passes whose name matches the given POSIX regular expression

-Rpass=<arg>

Report transformations performed by optimization passes whose name matches the given POSIX regular expression

-W<warning>, **--extra-warnings**, **--warn-<arg>**, **--warn-**=<arg>

Enable the specified warning

-Wdeprecated, **-Wno-deprecated**

Enable warnings for deprecated constructs and define __DEPRECATED

-Wframe-larger-than=<arg>, **-Wframe-larger-than**

-Wnonportable-cfstrings<arg>, **-Wno-nonportable-cfstrings<arg>**

Target-independent compilation options

-fPIC, **-fno-PIC**

-fPIE, **-fno-PIE**

-faccess-control, **-fno-access-control**

-faddrsig, **-fno-addrsig**

Emit an address-significance table

-falign-functions, **-fno-align-functions**

-falign-functions=<arg>

-falign-loops=<N>

N must be a power of two. Align loops to the boundary

-faligned-allocation, **-faligned-new**, **-fno-aligned-allocation**

Enable C++17 aligned allocation functions

-fallow-editor-placeholders, **-fno-allow-editor-placeholders**

Treat editor placeholders as valid source code

-fallow-unsupported

-faltivec, **-fno-altivec**

-faltivec-src-compat=<arg>

Source-level compatibility for Altivec vectors (for PowerPC targets). This includes results of vector comparison (scalar for ‘xl’, vector for ‘gcc’) as well as behavior when initializing with a scalar (splatting for ‘xl’, element zero only for ‘gcc’). For ‘mixed’, the compatibility is as ‘gcc’ for ‘vector bool/vector pixel’ and as ‘xl’ for other types. Current default is ‘mixed’. <arg> must be ‘mixed’, ‘gcc’ or ‘xl’.

-fandroid-pad-segment, **-fno-android-pad-segment**

-fansi-escape-codes

Use ANSI escape codes for diagnostics

-fapple-kext, **-findirect-virtual-calls**, **-fterminated-vtables**

Use Apple’s kernel extensions ABI

-fapple-link-rtlib

Force linking the clang builtins runtime library

-fapple-pragma-pack, **-fno-apple-pragma-pack**

Enable Apple gcc-compatible #pragma pack handling

-fapplication-extension, **-fno-application-extension**

Restrict code to those available for App Extensions

-fapprox-func, **-fno-approx-func**

Allow certain math function calls to be replaced with an approximately equivalent calculation

-fasm, **-fno-asm**

-fasm-blocks, **-fno-asm-blocks**

-fassociative-math, **-fno-associative-math**

-fassume-nothrow-exception-dtor, **-fno-assume-nothrow-exception-dtor**

Assume that exception objects' destructors are non-throwing

-fassume-sane-operator-new, **-fno-assume-sane-operator-new**

-fassume-unique-vtables, **-fno-assume-unique-vtables**

-fassumptions, **-fno-assumptions**

-fast

-fastcp

-fastf

-fasync-exceptions, **-fno-async-exceptions**

Enable EH Asynchronous exceptions

-fasynchronous-unwind-tables, **-fno-asynchronous-unwind-tables**

-fauto-import, **-fno-auto-import**

MinGW specific. Enable code generation support for automatic dllimport, and enable support for it in the linker. Enabled by default.

-fautolink, **-fno-autolink**

-fbasic-block-address-map, **-fno-basic-block-address-map**

Emit the basic block address map section.

-fbasic-block-sections=<arg>

Generate labels for each basic block or place each basic block or a subset of basic blocks in its own section. <arg> must be 'all', 'labels', 'none' or 'list='.

-fbinutils-version=<major.minor>

Produced object files can use all ELF features supported by this binutils version and newer. If -fno-integrated-as is specified, the generated assembly will consider GNU as support. 'none' means that all ELF features can be used, regardless of binutils support. Defaults to 2.26.

-fblocks, **-fno-blocks**

Enable the 'blocks' language feature

-fbootclasspath=<arg>, **--bootclasspath** <arg>, **--bootclasspath**=<arg>

-fborland-extensions, **-fno-borland-extensions**

Accept non-standard constructs supported by the Borland compiler

-fbracket-depth=<arg>

-fbuiltin, **-fno-builtin**

-fbuiltin-module-map

Load the clang builtins module map file.

-fc++-static-destructors, **-fno-c++-static-destructors**

-fcaret-diagnostics, **-fno-caret-diagnostics**

-fcaret-diagnostics-max-lines=<arg>

Set the maximum number of source lines to show in a caret diagnostic (0 = no limit).

-fcf-protection=<arg>, **-fcf-protection** (equivalent to -fcf-protection=full)

Instrument control-flow architecture protection. <arg> must be 'return', 'branch', 'full' or 'none'.

-fcf-runtime-abi=<arg>
<arg> must be 'unspecified', 'standalone', 'objc', 'swift', 'swift-5.0', 'swift-4.2' or 'swift-4.1'.

-fchar8_t, **-fno-char8_t**

Enable C++ builtin type char8_t

-fclangir, **-fno-clangir**

Use the ClangIR pipeline to compile

-fclasspath=<arg>, **--CLASSPATH** <arg>, **--CLASSPATH**=<arg>, **--classpath** <arg>, **--classpath**=<arg>

-fcolor-diagnostics, **-fdiagnostics-color**, **-fno-color-diagnostics**

Enable colors in diagnostics

-fcommon, **-fno-common**

Place definitions of variables with no storage class and no initializer (tentative definitions) in a common block, instead of generating individual zero-initialized definitions (default -fno-common).

-fcompile-resource=<arg>, **--resource** <arg>, **--resource**=<arg>

-fcomplex-arithmetic=<arg>
<arg> must be 'full', 'improved', 'promoted' or 'basic'.

-fconstant-cfstrings, **-fno-constant-cfstrings**

-fconstant-string-class=<arg>

-fconstexpr-backtrace-limit=<arg>

Set the maximum number of entries to print in a constexpr evaluation backtrace (0 = no limit)

-fconstexpr-depth=<arg>

Set the maximum depth of recursive constexpr function calls

-fconstexpr-steps=<arg>

Set the maximum number of steps in constexpr function evaluation

-fconvergent-functions, **-fno-convergent-functions**

-fcoro-aligned-allocation, **-fno-coro-aligned-allocation**

Prefer aligned allocation for C++ Coroutines

-fcoroutines, **-fno-coroutines**

Enable support for the C++ Coroutines

-fcoverage-compilation-dir=<arg>

The compilation directory to embed in the coverage mapping.

-fcoverage-mapping, **-fno-coverage-mapping**

Generate coverage mapping to enable code coverage analysis

-fcoverage-mcdc, **-fno-coverage-mcdc**

Enable MC/DC criteria when generating code coverage

-fcoverage-prefix-map=<old>=<new>

remap file source paths <old> to <new> in coverage mapping. If there are multiple options, prefix replacement is applied in reverse order starting from the last one

-fcreate-profile

-fcs-profile-generate

Generate instrumented code to collect context sensitive execution counts into default.profraw (overridden by LLVM_PROFILE_FILE env var)

-fcs-profile-generate=<directory>

Generate instrumented code to collect context sensitive execution counts into <directory>/default.profraw (overridden by LLVM_PROFILE_FILE env var)

-fcxx-exceptions, **-fno-cxx-exceptions**

Enable C++ exceptions

-fcxx-modules, **-fno-cxx-modules**

Enable modules for C++

-fdata-sections, **-fno-data-sections**

Place each data in its own section

-fdebug-compilation-dir=<arg>, **-fdebug-compilation-dir** <arg>

The compilation directory to embed in the debug info

-fdebug-default-version=<arg>

Default DWARF version to use, if a -g option caused DWARF debug info to be produced

-fdebug-info-for-profiling, **-fno-debug-info-for-profiling**

Emit extra debug info to make sample profile more accurate

-fdebug-macro, **-fno-debug-macro**

Emit macro debug information

-fdebug-pass-arguments

-fdebug-pass-structure

-fdebug-prefix-map=<old>=<new>

For paths in debug info, remap directory <old> to <new>. If multiple options match a path, the last option wins

-fdebug-ranges-base-address, **-fno-debug-ranges-base-address**

Use DWARF base address selection entries in .debug_ranges

-fdebug-types-section, **-fno-debug-types-section**

Place debug types in their own section (ELF Only)

-fdefine-target-os-macros, **-fno-define-target-os-macros**

Enable predefined target OS macros

-fdelayed-template-parsing, **-fno-delayed-template-parsing**

Parse templated function definitions at the end of the translation unit

-fdelete-null-pointer-checks, **-fno-delete-null-pointer-checks**

When enabled, treat null pointer dereference, creation of a reference to null, or passing a null pointer to a function parameter annotated with the “nonnull” attribute as undefined behavior. (And, thus the optimizer may assume that any pointer used in such a way must not have been null and optimize away the branches accordingly.) On by default.

-fdenormal-fp-math=<arg>

-fdiagnostics-absolute-paths

Print absolute paths in diagnostics

-fdiagnostics-color=<arg>

-fdiagnostics-hotness-threshold=<value>

Prevent optimization remarks from being output if they do not have at least this profile count. Use ‘auto’ to apply the threshold from profile summary

-fdiagnostics-misexpect-tolerance=<value>

Prevent misexpect diagnostics from being output if the profile counts are within N% of the expected.

-fdiagnostics-show-hotness, **-fno-diagnostics-show-hotness**

Enable profile hotness information in diagnostic line

-fdiagnostics-show-line-numbers, **-fno-diagnostics-show-line-numbers**

-fdiagnostics-show-note-include-stack, **-fno-diagnostics-show-note-include-stack**

Display include stacks for diagnostic notes

-fdiagnostics-show-option, **-fno-diagnostics-show-option**

Print option name with mappable diagnostics

-fdiagnostics-show-template-tree

Print a template comparison tree for differing templates

-fdigraphs, **-fno-digraphs**

Enable alternative token representations ‘<:’, ‘:>’, ‘<%’, ‘%>’, ‘%:’, ‘%:%:’ (default)

-fdirect-access-external-data, **-fno-direct-access-external-data**

Don’t use GOT indirection to reference external data symbols

-fdirectives-only, **-fno-directives-only**

-fdollars-in-identifiers, **-fno-dollars-in-identifiers**

Allow ‘$’ in identifiers

-fdouble-square-bracket-attributes, **-fno-double-square-bracket-attributes**

-fdwarf-directory-asm, **-fno-dwarf-directory-asm**

-fdwarf-exceptions

Use DWARF style exceptions

-felide-constructors, **-fno-elide-constructors**

-feliminate-unused-debug-symbols, **-fno-eliminate-unused-debug-symbols**

-feliminate-unused-debug-types, **-fno-eliminate-unused-debug-types**

Do not emit debug info for defined but unused types

-fembed-bitcode=<option>, **-fembed-bitcode** (equivalent to -fembed-bitcode=all), **-fembed-bitcode-marker** (equivalent to -fembed-bitcode=marker)

Embed LLVM bitcode. <option> must be ‘off’, ‘all’, ‘bitcode’ or ‘marker’.

-fembed-offload-object=<arg>

Embed Offloading device-side binary into host object file as a section.

-femit-all-decls

Emit all declarations, even if unused

-femit-compact-unwind-non-canonical, **-fno-emit-compact-unwind-non-canonical**

Try emitting Compact-Unwind for non-canonical entries. Maybe overriden by other constraints

-femit-dwarf-unwind=<arg>

When to emit DWARF unwind (EH frame) info. <arg> must be ‘always’, ‘no-compact-unwind’ or ‘default’.

-femulated-tls, **-fno-emulated-tls**

Use emutls functions to access thread_local variables

-fenable-matrix

Enable matrix data type and related builtin functions

-fencoding=<arg>, **--encoding** <arg>, **--encoding**=<arg>

-ferror-limit=<arg>

-fescaping-block-tail-calls, **-fno-escaping-block-tail-calls**

-fexceptions, **-fno-exceptions**

Enable support for exception handling

-fexcess-precision=<arg>

Allows control over excess precision on targets where native support for the precision types is not available. By default, excess precision is used to calculate intermediate results following the rules specified in ISO C99. <arg> must be ‘standard’, ‘fast’ or ‘none’.

-fexec-charset=<arg>

-fexperimental-late-parse-attributes, **-fno-experimental-late-parse-attributes**

Enable experimental late parsing of attributes

-fexperimental-library, **-fno-experimental-library**

Control whether unstable and experimental library features are enabled. This option enables various library features that are either experimental (also known as TSes), or have been but are not stable yet in the selected Standard Library implementation. It is not recommended to use this option in production code, since neither ABI nor API stability are guaranteed. This is intended to provide a preview of features that will ship in the future for experimentation purposes

-fexperimental-modules-reduced-bmi

Generate the reduced BMI

-fexperimental-new-constant-interpreter

Enable the experimental new constant interpreter

-fexperimental-openacc-macro-override <arg>, **-fexperimental-openacc-macro-override**=<arg>

Overrides the _OPENACC macro value for experimental testing during OpenACC support development

-fexperimental-sanitize-metadata-ignorelist=<arg>

Disable sanitizer metadata for modules and functions that match the provided special case list

-fexperimental-sanitize-metadata=<arg1>,<arg2>...,
-fno-experimental-sanitize-metadata=<arg1>,<arg2>...

Specify the type of metadata to emit for binary analysis sanitizers

-fextdirs=<arg>, **--extdirs** <arg>, **--extdirs**=<arg>

-fextend-arguments=<arg>

Controls how scalar integer arguments are extended in calls to unprototyped and varargs functions. <arg> must be ‘32’ or ‘64’.

-ffast-math, **-fno-fast-math**

Allow aggressive, lossy floating-point optimizations

-ffat-lto-objects, **-fno-fat-lto-objects**

Enable fat LTO object support

-ffile-compilation-dir=<arg>

The compilation directory to embed in the debug info and coverage mapping.

-ffile-prefix-map=<arg>

remap file source paths in debug info, predefined preprocessor macros and __builtin_FILE(). Implies -ffile-reproducible.

-ffile-reproducible, **-fno-file-reproducible**

Use the target’s platform-specific path separator character when expanding the __FILE__ macro

-ffinite-loops, **-fno-finite-loops**

Assume all non-trivial loops are finite.

-ffinite-math-only, **-fno-finite-math-only**

Allow floating-point optimizations that assume arguments and results are not NaNs or +-inf. This defines the __FINITE_MATH_ONLY__ preprocessor macro.

-ffixed-point, **-fno-fixed-point**

Enable fixed point types

-ffixed-r19

Reserve register r19 (Hexagon only)

-ffor-scope, **-fno-for-scope**

-fforce-check-cxx20-modules-input-files

Check the input source files from C++20 modules explicitly

-fforce-dwarf-frame, **-fno-force-dwarf-frame**

Always emit a debug frame section

-fforce-emit-vtables, **-fno-force-emit-vtables**

In order to improve devirtualization, forces emitting of vtables even in modules where it isn’t necessary. It causes more inline virtual functions to be emitted.

-fforce-enable-int128, **-fno-force-enable-int128**

Enable support for int128_t type

-ffp-contract=<arg>

Form fused FP ops (e.g. FMAs): fast (fuses across statements disregarding pragmas) | on (only fuses in the same statement unless dictated by pragmas) | off (never fuses) | fast-honor-pragmas (fuses across statements unless dictated by pragmas). Default is ‘fast’ for CUDA, ‘fast-honor-pragmas’ for HIP, and ‘on’ otherwise. <arg> must be ‘fast’, ‘on’, ‘off’ or ‘fast-honor-pragmas’.

-ffp-eval-method=<arg>

Specifies the evaluation method to use for floating-point arithmetic. <arg> must be ‘source’, ‘double’ or ‘extended’.

-ffp-exception-behavior=<arg>

Specifies the exception behavior of floating-point operations. <arg> must be ‘ignore’, ‘maytrap’ or ‘strict’.

-ffp-model=<arg>

Controls the semantics of floating-point calculations.

-ffreestanding

Assert that the compilation takes place in a freestanding environment

-ffunction-sections, **-fno-function-sections**

Place each function in its own section

-fgnu-inline-asm, **-fno-gnu-inline-asm**

-fgnu-keywords, **-fno-gnu-keywords**

Allow GNU-extension keywords regardless of language standard

-fgnu-runtime

Generate output compatible with the standard GNU Objective-C runtime

-fgnu89-inline, **-fno-gnu89-inline**

Use the gnu89 inline semantics

-fgnuc-version=<arg>

Sets various macros to claim compatibility with the given GCC version (default is 4.2.1)

-fgpu-approx-transcendentals, **-fcuda-approx-transcendentals**, **-fno-gpu-approx-transcendentals**

Use approximate transcendental functions

-fhonor-infinities, **-fhonor-infinites**, **-fno-honor-infinities**

Specify that floating-point optimizations are not allowed that assume arguments and results are not +-inf.

-fhonor-nans, **-fno-honor-nans**

Specify that floating-point optimizations are not allowed that assume arguments and results are not NANs.

-fhosted

-fignore-exceptions

Enable support for ignoring exception handling constructs

-fimplicit-module-maps, **-fmodule-maps**, **-fno-implicit-module-maps**

Implicitly search the file system for module map files.

-fimplicit-modules, **-fno-implicit-modules**

-fincremental-extensions

Enable incremental processing extensions such as processingstatements on the global scope.

-finline-max-stacksize=<arg>

Suppress inlining of functions whose stack size exceeds the given value

-finput-charset=<arg>

Specify the default character set for source files

-finstrument-function-entry-bare

Instrument function entry only, after inlining, without arguments to the instrumentation call

-finstrument-functions

Generate calls to instrument function entry and exit

-finstrument-functions-after-inlining

Like -finstrument-functions, but insert the calls after inlining

-fintegrated-as, **-fno-integrated-as**, **-integrated-as**

Enable the integrated assembler

-fintegrated-cc1, **-fno-integrated-cc1**

Run cc1 in-process

-fintegrated-objemitter, **-fno-integrated-objemitter**

Use internal machine object code emitter.

-fjmc, **-fno-jmc**

Enable just-my-code debugging

-fjump-tables, **-fno-jump-tables**

Use jump tables for lowering switches

-fkeep-persistent-storage-variables, **-fno-keep-persistent-storage-variables**

Enable keeping all variables that have a persistent storage duration, including global, static and thread-local variables, to guarantee that they can be directly addressed

-fkeep-static-consts, **-fno-keep-static-consts**

Keep static const variables even if unused

-fkeep-system-includes, **-fno-keep-system-includes**

Instead of expanding system headers when emitting preprocessor output, preserve the #include directive. Useful when producing preprocessed output for test case reduction. May produce incorrect output if preprocessor symbols that control the included content (e.g. _XOPEN_SOURCE) are defined in the including source file. The portability of the resulting source to other compilation environments is not guaranteed.

Only valid with -E.

-flax-vector-conversions=<arg>, **-flax-vector-conversions** (equivalent to -flax-vector-conversions=integer), **-fno-lax-vector-conversions** (equivalent to -flax-vector-conversions=none)

Enable implicit vector bit-casts. <arg> must be ‘none’, ‘integer’ or ‘all’.

-flimited-precision=<arg>

-flto-jobs=<arg>

Controls the backend parallelism of -flto=thin (default of 0 means the number of threads will be derived from the number of CPUs detected)

-flto=<arg>, **-flto** (equivalent to -flto=full), **-flto**=auto (equivalent to -flto=full), **-flto**=jobserver (equivalent to -flto=full)

Set LTO mode. <arg> must be ‘thin’ or ‘full’.

-fmacro-backtrace-limit=<arg>

Set the maximum number of entries to print in a macro expansion backtrace (0 = no limit)

-fmacro-prefix-map=<arg>

remap file source paths in predefined preprocessor macros and __builtin_FILE(). Implies -ffile-reproducible.

-fmath-errno, **-fno-math-errno**

Require math functions to indicate errors by setting errno

-fmax-tokens=<arg>

Max total number of preprocessed tokens for -Wmax-tokens.

-fmax-type-align=<arg>

Specify the maximum alignment to enforce on pointers lacking an explicit alignment

-fmemory-profile, **-fno-memory-profile**

Enable heap memory profiling

-fmemory-profile-use=<pathname>

Use memory profile for profile-guided memory optimization

-fmemory-profile=<directory>

Enable heap memory profiling and dump results into <directory>

-fmerge-all-constants, **-fno-merge-all-constants**

Allow merging of constants

-fmessage-length=<arg>

Format message diagnostics so that they fit within N columns

-fminimize-whitespace, **-fno-minimize-whitespace**

Ignore the whitespace from the input file when emitting preprocessor output. It will only contain whitespace when necessary, e.g. to keep two minus signs from merging into to an increment operator. Useful with the -P option to normalize whitespace such that two files with only formatting changes are equal.

Only valid with -E on C-like inputs and incompatible with -traditional-cpp.

-fmodule-file-deps, **-fno-module-file-deps**

-fmodule-header

Build a C++20 Header Unit from a header

-fmodule-header=<kind>

Build a C++20 Header Unit from a header that should be found in the user (fmodule-header=user) or system (fmodule-header=system) search path.

-fmodule-map-file=<file>

Load this module map file

-fmodule-name=<name>, **-fmodule-implementation-of** <arg>

Specify the name of the module to build

-fmodules, **-fno-modules**

Enable the ‘modules’ language feature

-fmodules-decluse, **-fno-modules-decluse**

Require declaration of modules used within a module

-fmodules-ignore-macro=<arg>

Ignore the definition of the given macro when building and loading modules

-fmodules-search-all, **-fno-modules-search-all**

Search even non-imported modules to resolve references

-fmodules-strict-decluse

Like -fmodules-decluse but requires all headers to be in modules

-fmodules-validate-input-files-content

Validate PCM input files based on content if mtime differs

-fms-compatibility, **-fno-ms-compatibility**

Enable full Microsoft Visual C++ compatibility

-fms-compatibility-version=<arg>

Dot-separated value representing the Microsoft compiler version number to report in _MSC_VER (0 = don’t define it (default))

-fms-extensions, **-fno-ms-extensions**

Accept some non-standard constructs supported by the Microsoft compiler

-fms-hotpatch

Ensure that all functions can be hotpatched at runtime

-fms-memptr-rep=<arg>

<arg> must be ‘single’, ‘multiple’ or ‘virtual’.

-fms-omit-default-lib<arg>

-fms-runtime-lib=<arg>

Specify Visual Studio C runtime library. “static” and “static_dbg” correspond to the cl flags /MT and /MTd which use the multithread, static version. “dll” and “dll_dbg” correspond to the cl flags /MD and /MDd which use the multithread, dll version. <arg> must be ‘static’, ‘static_dbg’, ‘dll’ or ‘dll_dbg’.

-fms-volatile, **-fno-ms-volatile**

Volatile loads and stores have acquire and release semantics

-fmsc-version=<arg>

Microsoft compiler version number to report in _MSC_VER (0 = don’t define it (default))

-fmudflap

-fmudflapth

-fnested-functions

-fnew-alignment=<align>, **-fnew-alignment** <arg>

Specifies the largest alignment guaranteed by ‘::operator new(size_t)’

-fnew-infallible, **-fno-new-infallible**

Enable treating throwing global C++ operator new as always returning valid memory (annotates with __attribute__((returns_nonnull)) and throw()). This is detectable in source.

-fnext-runtime

-fno-builtin-<arg>

Disable implicit builtin knowledge of a specific function

-fno-elide-type

Do not elide types when printing diagnostics

-fno-knr-functions

Disable support for K&R C function declarations

-fno-max-type-align

-fno-modules-check-relocated<arg>

Skip checks for relocated modules when loading PCM files

-fno-modules-validate-textual-header-includes

Do not enforce -fmodules-decluse and private header restrictions for textual headers. This flag will be removed in a future Clang release.

-fno-strict-modules-decluse

-fno-temp-file

Directly create compilation output files. This may lead to incorrect incremental builds if the compiler crashes

-fno-working-directory

-fno_modules-validate-input-files-content

-fno_pch-validate-input-files-content

-fobjc-abi-version=<arg>

-fobjc-arc, **-fno-objc-arc**

Synthesize retain and release calls for Objective-C pointers

-fobjc-arc-exceptions, **-fno-objc-arc-exceptions**

Use EH-safe code when synthesizing retains and releases in -fobjc-arc

-fobjc-avoid-heapify-local-blocks, **-fno-objc-avoid-heapify-local-blocks**

Try to avoid heapifying local blocks

-fobjc-convert-messages-to-runtime-calls, **-fno-objc-convert-messages-to-runtime-calls**

-fobjc-disable-direct-methods-for-testing

Ignore attribute objc_direct so that direct methods can be tested

-fobjc-encode-cxx-class-template-spec, **-fno-objc-encode-cxx-class-template-spec**

Fully encode c++ class template specialization

-fobjc-exceptions, **-fno-objc-exceptions**

Enable Objective-C exceptions

-fobjc-infer-related-result-type, **-fno-objc-infer-related-result-type**

-fobjc-legacy-dispatch, **-fno-objc-legacy-dispatch**

-fobjc-link-runtime

-fobjc-nonfragile-abi, **-fno-objc-nonfragile-abi**

-fobjc-nonfragile-abi-version=<arg>

-fobjc-runtime=<arg>

Specify the target Objective-C runtime kind and version

-fobjc-sender-dependent-dispatch

-fobjc-weak, **-fno-objc-weak**

Enable ARC-style weak references in Objective-C

-foffload-lto=<arg>, **-foffload-lto** (equivalent to -foffload-lto=full)

Set LTO mode for offload compilation. <arg> must be ‘thin’ or ‘full’.

-foffload-uniform-block, **-cl-uniform-work-group-size**, **-fno-offload-uniform-block**

Assume that kernels are launched with uniform block sizes (default true for CUDA/HIP and false otherwise)

-fomit-frame-pointer, **-fno-omit-frame-pointer**

Omit the frame pointer from functions that don’t need it. Some stack unwinding cases, such as profilers and sanitizers, may prefer specifying -fno-omit-frame-pointer. On many targets, -O1 and higher omit the frame pointer by default. -m[no-]omit-leaf-frame-pointer takes precedence for leaf functions

-fopenacc

Enable OpenACC

-fopenmp, **-fno-openmp**

Parse OpenMP pragmas and generate parallel code.

-fopenmp-extensions, **-fno-openmp-extensions**

Enable all Clang extensions for OpenMP directives and clauses

-fopenmp-force-usm

Force behvaior as if the user specified pragma omp requires unified_shared_memory.

-fopenmp-offload-mandatory

Do not create a host fallback if offloading to the device fails.

-fopenmp-simd, **-fno-openmp-simd**

Emit OpenMP code only for SIMD-based constructs.

-fopenmp-target-debug, **-fno-openmp-target-debug**

Enable debugging in the OpenMP offloading device RTL

-fopenmp-target-jit

Emit code that can be JIT compiled for OpenMP offloading. Implies -foffload-lto=full

-fopenmp-version=<arg>

Set OpenMP version (e.g. 45 for OpenMP 4.5, 51 for OpenMP 5.1). Default value is 51 for Clang

-fopenmp=<arg>

-foperator-arrow-depth=<arg>

Maximum number of ‘operator->’s to call for a member access

-foperator-names, **-fno-operator-names**

-foptimization-record-file=<file>

Specify the output name of the file containing the optimization remarks. Implies -fsave-optimization-record. On Darwin platforms, this cannot be used with multiple -arch <arch> options.

-foptimization-record-passes=<regex>

Only include passes which match a specified regular expression in the generated optimization record (by default, include all passes)

-foptimize-sibling-calls, **-fno-optimize-sibling-calls**

-forder-file-instrumentation

Generate instrumented code to collect order file into default.profraw file (overridden by '=' form of option or LLVM_PROFILE_FILE env var)

-foutput-class-dir=<arg>, **--output-class-directory** <arg>, **--output-class-directory**=<arg>

-fpack-struct, **-fno-pack-struct**

-fpack-struct=<arg>

Specify the default maximum struct packing alignment

-fpascal-strings, **-fno-pascal-strings**, **-mpascal-strings**

Recognize and construct Pascal-style string literals

-fpass-plugin=<dsopath>

Load pass plugin from a dynamic shared object file (only with new pass manager).

-fpatchable-function-entry=<N,M>

Generate M NOPs before function entry and N-M NOPs after function entry

-fpcc-struct-return

Override the default ABI to return all structs on the stack

-fpch-codegen, **-fno-pch-codegen**

Generate code for uses of this PCH that assumes an explicit object file will be built for the PCH

-fpch-debuginfo, **-fno-pch-debuginfo**

Generate debug info for types in an object file built from this PCH and do not generate them elsewhere

-fpch-instantiate-templates, **-fno-pch-instantiate-templates**

Instantiate templates already while building a PCH

-fpch-preprocess

-fpch-validate-input-files-content

Validate PCH input files based on content if mtime differs

-fpic, **-fno-pic**

-fpie, **-fno-pie**

-fplt, **-fno-plt**

-fplugin=<dsopath>

Load the named plugin (dynamic shared object)

-fprebuilt-implicit-modules, **-fno-prebuilt-implicit-modules**

Look up implicit modules in the prebuilt module path

-fpreserve-as-comments, **-fno-preserve-as-comments**

-fproc-stat-report<arg>

Print subprocess statistics

-fproc-stat-report=<arg>

Save subprocess statistics to the given file

-fprofile-arcs, **-fno-profile-arcs**

Instrument code to produce gcov data files (*.gcda)

-fprofile-dir=<arg>

-fprofile-exclude-files=<arg>

Instrument only functions from files where names don’t match all the regexes separated by a semi-colon

-fprofile-filter-files=<arg>

Instrument only functions from files where names match any regex separated by a semi-colon

-fprofile-function-groups=<N>

Partition functions into N groups and select only functions in group i to be instrumented using -fprofile-selected-function-group

-fprofile-generate, **-fno-profile-generate**

Generate instrumented code to collect execution counts into default.profraw (overridden by LLVM_PROFILE_FILE env var)

-fprofile-generate=<directory>

Generate instrumented code to collect execution counts into <directory>/default.profraw (overridden by LLVM_PROFILE_FILE env var)

-fprofile-instr-generate, **-fno-profile-instr-generate**

Generate instrumented code to collect execution counts into default.profraw file (overridden by ‘=’ form of option or LLVM_PROFILE_FILE env var)

-fprofile-instr-generate=<file>

Generate instrumented code to collect execution counts into <file> (overridden by LLVM_PROFILE_FILE env var)

-fprofile-instr-use, **-fno-profile-instr-use**, **-fprofile-use**

-fprofile-instr-use=<arg>

Use instrumentation data for profile-guided optimization

-fprofile-list=<arg>

Filename defining the list of functions/files to instrument. The file uses the sanitizer special case list format.

-fprofile-remapping-file=<file>

Use the remappings described in <file> to match the profile data against names in the program

-fprofile-sample-accurate, **-fauto-profile-accurate**, **-fno-profile-sample-accurate**

Specifies that the sample profile is accurate. If the sample

> profile is accurate, callsites without profile samples are marked as cold. Otherwise, treat callsites without profile samples as if we have no profile

-fprofile-sample-use, **-fauto-profile**, **-fno-profile-sample-use**

-fprofile-sample-use=<arg>, **-fauto-profile**=<arg>

Enable sample-based profile guided optimizations

-fprofile-selected-function-group=<i>

Partition functions into N groups using -fprofile-function-groups and select only functions in group i to be instrumented. The valid range is 0 to N-1 inclusive

-fprofile-update=<method>

Set update method of profile counters. <method> must be ‘atomic’, ‘prefer-atomic’ or ‘single’.

-fprofile-use=<pathname>

Use instrumentation data for profile-guided optimization. If pathname is a directory, it reads from <pathname>/default.profdata. Otherwise, it reads from file <pathname>.

-fprotect-parens, **-fno-protect-parens**

Determines whether the optimizer honors parentheses when floating-point expressions are evaluated

-fpseudo-probe-for-profiling, **-fno-pseudo-probe-for-profiling**

Emit pseudo probes for sample profiling

-fptrauth-auth-traps, **-fno-ptrauth-auth-traps**

Enable traps on authentication failures

-fptrauth-calls, **-fno-ptrauth-calls**

Enable signing and authentication of all indirect calls

-fptrauth-init-fini, **-fno-ptrauth-init-fini**

Enable signing of function pointers in init/fini arrays

-fptrauth-intrinsics, **-fno-ptrauth-intrinsics**

Enable pointer authentication intrinsics

-fptrauth-returns, **-fno-ptrauth-returns**

Enable signing and authentication of return addresses

-fptrauth-vtable-pointer-address-discrimination, **-fno-ptrauth-vtable-pointer-address-discrimination**

Enable address discrimination of vtable pointers

-fptrauth-vtable-pointer-type-discrimination, **-fno-ptrauth-vtable-pointer-type-discrimination**

Enable type discrimination of vtable pointers

-freciprocal-math, **-fno-reciprocal-math**

Allow division operations to be reassociated

-freg-struct-return

Override the default ABI to return small structs in registers

-fregister-global-dtors-with-atexit, **-fno-register-global-dtors-with-atexit**

Use atexit or __cxa_atexit to register global destructors

-frelaxed-template-template-args, **-fno-relaxed-template-template-args**

Enable C++17 relaxed template template argument matching

-fretain-comments-from-system-headers

-frewrite-imports, **-fno-rewrite-imports**

-frewrite-includes, **-fno-rewrite-includes**

-fropi, **-fno-ropi**

Generate read-only position independent code (ARM only)

-frounding-math, **-fno-rounding-math**

-frtlib-defaultlib, **-fno-rtlib-defaultlib**

On Windows, emit /defaultlib: directives to link compiler-rt libraries (default)

-frtti, **-fno-rtti**

-frtti-data, **-fno-rtti-data**

-frwpi, **-fno-rwpi**

Generate read-write position independent code (ARM only)

-fsafe-buffer-usage-suggestions, **-fno-safe-buffer-usage-suggestions**

Display suggestions to update code associated with -Wunsafe-buffer-usage warnings

-fsample-profile-use-profi

Infer block and edge counts. If the profiles have errors or missing

> blocks caused by sampling, profile inference (profi) can convert basic block counts to branch probabilites to fix them by extended and re-engineered classic MCMF (min-cost max-flow) approach.

-fsanitize-memory-param-retval, **-fno-sanitize-memory-param-retval**

Enable detection of uninitialized parameters and return values

-fsanitize-stable-abi, **-fno-sanitize-stable-abi**

Stable ABI instrumentation for sanitizer runtime. Default: Conventional

-fsave-optimization-record, **-fno-save-optimization-record**

Generate a YAML optimization record file

-fsave-optimization-record=<format>

Generate an optimization record file in a specific format

-fseh-exceptions

Use SEH style exceptions

-fsemantic-interposition, **-fno-semantic-interposition**

Enable semantic interposition. Semantic interposition allows for the interposition of a symbol by another at runtime, thus preventing a range of inter-procedural optimisation.

-fseparate-named-sections, **-fno-separate-named-sections**

Use separate unique sections for named sections (ELF Only)

-fshort-enums, **-fno-short-enums**

Allocate to an enum type only as many bytes as it needs for the declared range of possible values

-fshort-wchar, **-fno-short-wchar**

Force wchar_t to be a short unsigned int

-fshow-column, **-fno-show-column**

-fshow-overloads=<arg>

Which overload candidates to show when overload resolution fails. Defaults to ‘all’. <arg> must be ‘best’ or ‘all’.

-fshow-source-location, **-fno-show-source-location**

-fsignaling-math, **-fno-signaling-math**

-fsigned-bitfields

-fsigned-char, **-fno-signed-char**, **--signed-char**

char is signed

-fsigned-zeros, **-fno-signed-zeros**

-fsized-deallocation, **-fno-sized-deallocation**

Enable C++14 sized global deallocation functions

-fsjlj-exceptions

Use SjLj style exceptions

-fskip-odr-check-in-gmf, **-fno-skip-odr-check-in-gmf**

Skip ODR checks for decls in the global module fragment.

-fslp-vectorize, **-fno-slp-vectorize**, **-ftree-slp-vectorize**

Enable the superword-level parallelism vectorization passes

-fspell-checking, **-fno-spell-checking**

-fspell-checking-limit=<arg>

Set the maximum number of times to perform spell checking on unrecognized identifiers (0 = no limit)

-fsplit-dwarf-inlining, **-fno-split-dwarf-inlining**

Provide minimal debug info in the object/executable to facilitate online symbolication/stack traces in the absence of .dwo/.dwp files when using Split DWARF

-fsplit-lto-unit, **-fno-split-lto-unit**

Enables splitting of the LTO unit

-fsplit-machine-functions, **-fno-split-machine-functions**

Enable late function splitting using profile information (x86 ELF)

-fsplit-stack, **-fno-split-stack**

Use segmented stack

-fstack-clash-protection, **-fno-stack-clash-protection**

Instrument stack allocation to prevent stack clash attacks

-fstack-protector, **-fno-stack-protector**

Enable stack protectors for some functions vulnerable to stack smashing. This uses a loose heuristic which considers functions vulnerable if they contain a char (or 8bit integer) array or constant sized calls to alloca , which are of greater size than ssp-buffer-size (default: 8 bytes). All variable sized calls to alloca are considered vulnerable. A function with a stack protector has a guard value added to the stack frame that is checked on function exit. The guard value must be positioned in the stack frame such that a buffer overflow from a vulnerable variable will overwrite the guard value before overwriting the function's return address. The reference stack guard value is stored in a global variable.

-fstack-protector-all

Enable stack protectors for all functions

-fstack-protector-strong

Enable stack protectors for some functions vulnerable to stack smashing. Compared to -fstack-protector, this uses a stronger heuristic that includes functions containing arrays of any size (and any type), as well as any calls to alloca or the taking of an address from a local variable

-fstack-size-section, **-fno-stack-size-section**

Emit section containing metadata on function stack sizes

-fstack-usage

Emit .su file containing information on function stack sizes

-fstandalone-debug, **-fno-limit-debug-info**, **-fno-standalone-debug**

Emit full debug info for all types used by the program

-fstrict-aliasing, **-fno-strict-aliasing**

Enable optimizations based on strict aliasing rules

-fstrict-enums, **-fno-strict-enums**

Enable optimizations based on the strict definition of an enum's value range

-fstrict-flex-arrays=<n>

Enable optimizations based on the strict definition of flexible arrays. <n> must be '0', '1', '2' or '3'.

-fstrict-float-cast-overflow, **-fno-strict-float-cast-overflow**

Assume that overflowing float-to-int casts are undefined (default)

-fstrict-overflow, **-fno-strict-overflow**

-fstrict-return, **-fno-strict-return**

-fstrict-vtable-pointers, **-fno-strict-vtable-pointers**

Enable optimizations based on the strict rules for overwriting polymorphic C++ objects

-fstruct-path-tbaa, **-fno-struct-path-tbaa**

-fswift-async-fp=<option>

Control emission of Swift async extended frame info. <option> must be ‘auto’, ‘always’ or ‘never’.

-fsymbol-partition=<arg>

-ftabstop=<arg>

-ftemplate-backtrace-limit=<arg>

Set the maximum number of entries to print in a template instantiation backtrace (0 = no limit)

-ftemplate-depth=<arg>, **-ftemplate-depth-<arg>**

Set the maximum depth of recursive template instantiation

-ftest-coverage, **-fno-test-coverage**

Produce gcov notes files (*.gcno)

-fthin-link-bitcode=<arg>

Write minimized bitcode to <file> for the ThinLTO thin link only

-fthinlto-index=<arg>

Perform ThinLTO importing using provided function summary index

-fthreadsafe-statics, **-fno-threadsafe-statics**

-ftime-report

-ftime-report=<arg>

(For new pass manager) ‘per-pass’: one report for each pass; ‘per-pass-run’: one report for each pass invocation. <arg> must be ‘per-pass’ or ‘per-pass-run’.

-ftime-trace

Turn on time profiler. Generates JSON file based on output filename. Results can be analyzed with chrome://tracing or **Speedscope App** for flamegraph visualization.

-ftime-trace-granularity=<arg>

Minimum time granularity (in microseconds) traced by time profiler

-ftime-trace=<arg>

Similar to -ftime-trace. Specify the JSON file or a directory which will contain the JSON file

-ftls-model=<arg>

<arg> must be ‘global-dynamic’, ‘local-dynamic’, ‘initial-exec’ or ‘local-exec’.

-ftrap-function=<arg>

Issue call to specified function rather than a trap instruction

-ftrapping-math, **-fno-trapping-math**

-ftrapv

Trap on integer overflow

-ftrapv-handler <arg>

-ftrapv-handler=<function name>

Specify the function to be called on overflow

-ftrigraphs, **-fno-trigraphs**, **-trigraphs**, **--trigraphs**

Process trigraph sequences

-ftrivial-auto-var-init-max-size=<arg>

Stop initializing trivial automatic stack variables if var size exceeds the specified number of instances (in bytes)

-ftrivial-auto-var-init-stop-after=<arg>

Stop initializing trivial automatic stack variables after the specified number of instances

-ftrivial-auto-var-init=<arg>

Initialize trivial automatic stack variables. Defaults to ‘uninitialized’. <arg> must be ‘uninitialized’, ‘zero’ or ‘pattern’.

-funified-lto, **-fno-unified-lto**

Use the unified LTO pipeline

-funique-basic-block-section-names, **-fno-unique-basic-block-section-names**

Use unique names for basic block sections (ELF Only)

-funique-internal-linkage-names, **-fno-unique-internal-linkage-names**

Uniqueify Internal Linkage Symbol Names by appending the MD5 hash of the module path

-funique-section-names, **-fno-unique-section-names**

-funroll-loops, **-fno-unroll-loops**

Turn on loop unroller

-funsafe-math-optimizations, **-fno-unsafe-math-optimizations**

Allow unsafe floating-point math optimizations which may decrease precision

-funsigned-bitfields

-funsigned-char, **-fno-unsigned-char**, **--unsigned-char**

-funwind-tables, **-fno-unwind-tables**

-fuse-cxa-atexit, **-fno-use-cxa-atexit**

-fuse-init-array, **-fno-use-init-array**

-fuse-ld=<arg>

-fuse-line-directives, **-fno-use-line-directives**

Use #line in preprocessed output

-fvalidate-ast-input-files-content

Compute and store the hash of input files used to build an AST. Files with mismatching mtime's are considered valid if both contents is identical

-fveclib=<arg>

Use the given vector functions library. <arg> must be 'Accelerate', 'libmvec', 'MASSV', 'SVML', 'SLEEF', 'Darwin_libsystem_m', 'ArmPL', 'AMDLIBM' or 'none'.

-fvectorize, **-fno-vectorize**, **-ftree-vectorize**

Enable the loop vectorization passes

-fverbose-asm, **-dA**, **-fno-verbose-asm**

Generate verbose assembly output

-fvirtual-function-elimination, **-fno-virtual-function-elimination**

Enables dead virtual function elimination optimization. Requires -flto=full

-fvisibility-dllexport=<arg>

The visibility for dllexport definitions. If Keep is specified the visibility is not adjusted [-fvisibility-from-dllstorageclass]. <arg> must be 'keep', 'hidden', 'protected' or 'default'.

-fvisibility-externs-dllimport=<arg>

The visibility for dllimport external declarations. If Keep is specified the visibility is not adjusted [-fvisibility-from-dllstorageclass]. <arg> must be 'keep', 'hidden', 'protected' or 'default'.

-fvisibility-externs-nodllstorageclass=<arg>

The visibility for external declarations without an explicit DLL storage class. If Keep is specified the visibility is not adjusted [-fvisibility-from-dllstorageclass]. <arg> must be 'keep', 'hidden', 'protected' or 'default'.

-fvisibility-from-dllstorageclass, **-fno-visibility-from-dllstorageclass**

Override the visibility of globals based on their final DLL storage class.

-fvisibility-global-new-delete-hidden

Give global C++ operator new and delete declarations hidden visibility

-fvisibility-global-new-delete=<arg>

The visibility for global C++ operator new and delete declarations. If 'source' is specified the visibility is not adjusted. <arg> must be 'force-default', 'force-protected', 'force-hidden' or 'source'.

-fvisibility-inlines-hidden, **-fno-visibility-inlines-hidden**

Give inline C++ member functions hidden visibility by default

-fvisibility-inlines-hidden-static-local-var, **-fno-visibility-inlines-hidden-static-local-var**

When -fvisibility-inlines-hidden is enabled, static variables in inline C++ member functions will also be given hidden visibility by default

-fvisibility-ms-compat

Give global types 'default' visibility and global functions and variables 'hidden' visibility by default

-fvisibility-nodllstorageclass=<arg>

The visibility for definitions without an explicit DLL storage class. If Keep is specified the visibility is not adjusted [-fvisibility-from-dllstorageclass]. <arg> must be ‘keep’, ‘hidden’, ‘protected’ or ‘default’.

-fvisibility=<arg>

Set the default symbol visibility for all global definitions. <arg> must be ‘default’, ‘hidden’, ‘internal’ or ‘protected’.

-fwasm-exceptions

Use WebAssembly style exceptions

-fwhole-program-vtables, **-fno-whole-program-vtables**

Enables whole-program vtable optimization. Requires -flto

-fwrapv, **-fno-wrapv**

Treat signed integer overflow as two’s complement

-fwritable-strings

Store string literals as writable data

-fxl-pragma-pack, **-fno-xl-pragma-pack**

Enable IBM XL #pragma pack handling

-fxray-always-emit-customevents, **-fno-xray-always-emit-customevents**

Always emit __xray_customevent(…) calls even if the containing function is not always instrumented

-fxray-always-emit-typedevents, **-fno-xray-always-emit-typedevents**

Always emit __xray_typedevent(…) calls even if the containing function is not always instrumented

-fxray-always-instrument=<arg>

DEPRECATED: Filename defining the whitelist for imbuing the ‘always instrument’ XRay attribute.

-fxray-attr-list=<arg>

Filename defining the list of functions/types for imbuing XRay attributes.

-fxray-function-groups=<arg>

Only instrument 1 of N groups

-fxray-function-index, **-fno-xray-function-index**

-fxray-ignore-loops, **-fno-xray-ignore-loops**

Don’t instrument functions with loops unless they also meet the minimum function size

-fxray-instruction-threshold=<arg>

Sets the minimum function size to instrument with XRay

-fxray-instrument, **-fno-xray-instrument**

Generate XRay instrumentation sleds on function entry and exit

-fxray-instrumentation-bundle=<arg>

Select which XRay instrumentation points to emit. Options: all, none, function-entry, function-exit, function, custom. Default is ‘all’. ‘function’ includes both ‘function-entry’ and ‘function-exit’.

`-fxray-link-deps`, **`-fno-xray-link-deps`**

Link XRay runtime library when -fxray-instrument is specified (default)

`-fxray-modes``=<arg>`

List of modes to link in by default into XRay instrumented binaries.

`-fxray-never-instrument``=<arg>`

DEPRECATED: Filename defining the whitelist for imbuing the ‘never instrument’ XRay attribute.

`-fxray-selected-function-group``=<arg>`

When using -fxray-function-groups, select which group of functions to instrument. Valid range is 0 to fxray-function-groups - 1

`-fzero-call-used-regs``=<arg>`

Clear call-used registers upon function return (AArch64/x86 only). <arg> must be ‘skip’, ‘used-gpr-arg’, ‘used-gpr’, ‘used-arg’, ‘used’, ‘all-gpr-arg’, ‘all-gpr’, ‘all-arg’ or ‘all’.

`-fzero-initialized-in-bss`, **`-fno-zero-initialized-in-bss`**

`-fzvector`, **`-fno-zvector`**, **`-mzvector`**

Enable System z vector language extension

`-pedantic`, **`--pedantic`**, **`-no-pedantic`**, **`--no-pedantic`**

Warn on language extensions

`-pedantic-errors`, **`--pedantic-errors`**

Common Offloading options

`--amdgpu-arch-tool``=<arg>`

Tool used for detecting AMD GPU arch in the system.

`-cuid``=<arg>`

An ID for compilation unit, which should be the same for the same compilation unit but different for different compilation units. It is used to externalize device-side static variables for single source offloading languages CUDA and HIP so that they can be accessed by the host code of the same compilation unit.

`-fgpu-default-stream``=<arg>`

Specify default stream. The default value is ‘legacy’. (CUDA/HIP only). <arg> must be ‘legacy’ or ‘per-thread’.

`-fgpu-defer-diag`, **`-fno-gpu-defer-diag`**

Defer host/device related diagnostic messages for CUDA/HIP

`-fgpu-flush-denormals-to-zero`, **`-fcuda-flush-denormals-to-zero`**, **`-fno-gpu-flush-denormals-to-zero`**

Flush denormal floating point values to zero in CUDA/HIP device mode.

`-fgpu-rdc`, **`-fcuda-rdc`**, **`-fno-gpu-rdc`**

Generate relocatable device code, also known as separate compilation mode

`-fgpu-sanitize`, **`-fno-gpu-sanitize`**

Enable sanitizer for supported offloading devices

-foffload-implicit-host-device-templates, **-fno-offload-implicit-host-device-templates**

Template functions or specializations without host, device and global attributes have implicit host device attributes (CUDA/HIP only)

-fuse-cuid=<arg>

Method to generate ID’s for compilation units for single source offloading languages CUDA and HIP: ‘hash’ (ID’s generated by hashing file path and command line options) | ‘random’ (ID’s generated as random numbers) | ‘none’ (disabled). Default is ‘hash’. This option will be overridden by option ‘-cuid=[ID]’ if it is specified.

--nvptx-arch-tool=<arg>

Tool used for detecting NVIDIA GPU arch in the system.

--offload-arch=<arg>, **--cuda-gpu-arch**=<arg>, **--no-offload-arch**=<arg>

Specify an offloading device architecture for CUDA, HIP, or OpenMP. (e.g. sm_35). If ‘native’ is used the compiler will detect locally installed architectures. For HIP offloading, the device architecture can be followed by target ID features delimited by a colon (e.g. gfx908:xnack+:sramecc-). May be specified more than once.

--offload-compress, **--no-offload-compress**

Compress offload device binaries (HIP only)

--offload-device-only, **--cuda-device-only**

Only compile for the offloading device.

--offload-host-device, **--cuda-compile-host-device**

Compile for both the offloading host and device (default).

--offload-host-only, **--cuda-host-only**

Only compile for the offloading host.

--offload-new-driver, **--no-offload-new-driver**

Use the new driver for offloading compilation.

OpenCL options

-cl-denorms-are-zero

OpenCL only. Allow denormals to be flushed to zero.

-cl-ext=<arg1>,<arg2>...

OpenCL only. Enable or disable OpenCL extensions/optional features. The argument is a comma-separated sequence of one or more extension names, each prefixed by ‘+’ or ‘-‘.

-cl-fast-relaxed-math

OpenCL only. Sets -cl-finite-math-only and -cl-unsafe-math-optimizations, and defines __FAST_RELAXED_MATH__.

-cl-finite-math-only

OpenCL only. Allow floating-point optimizations that assume arguments and results are not NaNs or +-Inf.

-cl-fp32-correctly-rounded-divide-sqrt

OpenCL only. Specify that single precision floating-point divide and sqrt used in the program source are correctly rounded.

-cl-kernel-arg-info

OpenCL only. Generate kernel argument metadata.

-cl-mad-enable

OpenCL only. Allow use of less precise MAD computations in the generated binary.

-cl-no-signed-zeros

OpenCL only. Allow use of less precise no signed zeros computations in the generated binary.

-cl-no-stdinc

OpenCL only. Disables all standard includes containing non-native compiler types and functions.

-cl-opt-disable

OpenCL only. This option disables all optimizations. By default optimizations are enabled.

-cl-single-precision-constant

OpenCL only. Treat double precision floating-point constant as single precision constant.

-cl-std=<arg>

OpenCL language standard to compile for. <arg> must be ‘cl’, ‘CL’, ‘cl1.0’, ‘CL1.0’, ‘cl1.1’, ‘CL1.1’, ‘cl1.2’, ‘CL1.2’, ‘cl2.0’, ‘CL2.0’, ‘cl3.0’, ‘CL3.0’, ‘clc++’, ‘CLC++’, ‘clc++1.0’, ‘CLC++1.0’, ‘clc++2021’ or ‘CLC++2021’.

-cl-strict-aliasing

OpenCL only. This option is added for compatibility with OpenCL 1.0.

-cl-unsafe-math-optimizations

OpenCL only. Allow unsafe floating-point optimizations. Also implies -cl-no-signed-zeros and -cl-mad-enable.

SYCL options

-fsycl, **-fno-sycl**

Enables SYCL kernels compilation for device

-sycl-std=<arg>

SYCL language standard to compile for. <arg> must be ‘2020’, ‘2017’, ‘121’, ‘1.2.1’ or ‘sycl-1.2.1’.

CUDA options

--cuda-feature=<arg>

Manually specify the CUDA feature to use

--cuda-include-ptx=<arg>, **--no-cuda-include-ptx**=<arg>

Include PTX for the following GPU architecture (e.g. sm_35) or ‘all’. May be specified more than once.

--cuda-noopt-device-debug, **--no-cuda-noopt-device-debug**

Enable device-side debug info generation. Disables ptxas optimizations.

--cuda-path-ignore-env

Ignore environment variables to detect CUDA installation

--cuda-path=<arg>

CUDA installation path

-fcuda-short-ptr, **-fno-cuda-short-ptr**

Use 32-bit pointers for accessing const/local/shared address spaces

--no-cuda-version-check

Don’t error out if the detected version of the CUDA install is too low for the requested CUDA gpu architecture.

--ptxas-path=<arg>

Path to ptxas (used for compiling CUDA code)

HIP options

-fgpu-allow-device-init, **-fno-gpu-allow-device-init**

Allow device side init function in HIP (experimental)

-fhip-emit-relocatable, **-fno-hip-emit-relocatable**

Compile HIP source to relocatable

-fhip-fp32-correctly-rounded-divide-sqrt, **-fno-hip-fp32-correctly-rounded-divide-sqrt**

Specify that single precision floating-point divide and sqrt used in the program source are correctly rounded (HIP device compilation only)

-fhip-kernel-arg-name, **-fno-hip-kernel-arg-name**

Specify that kernel argument names are preserved (HIP only)

-fhip-new-launch-api, **-fno-hip-new-launch-api**

Use new kernel launching API for HIP

--gpu-bundle-output, **--no-gpu-bundle-output**

Bundle output files of HIP device compilation

--gpu-instrument-lib=<arg>

Instrument device library for HIP, which is a LLVM bitcode containing __cyg_profile_func_enter and __cyg_profile_func_exit

--gpu-max-threads-per-block=<arg>

Default max threads per block for kernel launch bounds for HIP

--hip-device-lib=<arg>

HIP device library

--hip-link

Link clang-offload-bundler bundles for HIP

--hip-path=<arg>

HIP runtime installation path, used for finding HIP version and adding HIP include path.

--hip-version=<arg>

HIP version in the format of major.minor.patch

--hipspv-pass-plugin=<dsopath>

path to a pass plugin for HIP to SPIR-V passes.

--hipstdpar

Enable HIP acceleration for standard parallel algorithms

--hipstdpar-interpose-alloc

Replace all memory allocation / deallocation calls with hipManagedMalloc / hipFree equivalents

--hipstdpar-path=<arg>

HIP Standard Parallel Algorithm Acceleration library path, used for finding and implicitly including the library header

--hipstdpar-prim-path=<arg>

rocPrim path, required by the HIP Standard Parallel Algorithm Acceleration library, used to implicitly include the rocPrim library

--hipstdpar-thrust-path=<arg>

rocThrust path, required by the HIP Standard Parallel Algorithm Acceleration library, used to implicitly include the rocThrust library

-no-hip-rt

Do not link against HIP runtime libraries

--rocm-device-lib-path=<arg>, **--hip-device-lib-path**=<arg>

ROCm device library path. Alternative to rocm-path.

--rocm-path=<arg>

ROCm installation path, used for finding and automatically linking required bitcode libraries.

Target-dependent compilation options

-G<size>, **-G**=<arg>, **-msmall-data-limit**=<arg>, **-msmall-data-threshold**=<arg>

Put objects of at most <size> bytes into small data section (MIPS / Hexagon)

-ffixed-x1

Reserve the x1 register (AArch64/RISC-V only)

-ffixed-x10

Reserve the x10 register (AArch64/RISC-V only)

-ffixed-x11

Reserve the x11 register (AArch64/RISC-V only)

-ffixed-x12

Reserve the x12 register (AArch64/RISC-V only)

-ffixed-x13

Reserve the x13 register (AArch64/RISC-V only)

-ffixed-x14

Reserve the x14 register (AArch64/RISC-V only)

-ffixed-x15

Reserve the x15 register (AArch64/RISC-V only)

-ffixed-x16

Reserve the x16 register (AArch64/RISC-V only)

-ffixed-x17

Reserve the x17 register (AArch64/RISC-V only)

-ffixed-x18

Reserve the x18 register (AArch64/RISC-V only)

-ffixed-x19

Reserve the x19 register (AArch64/RISC-V only)

-ffixed-x2

Reserve the x2 register (AArch64/RISC-V only)

-ffixed-x20

Reserve the x20 register (AArch64/RISC-V only)

-ffixed-x21

Reserve the x21 register (AArch64/RISC-V only)

-ffixed-x22

Reserve the x22 register (AArch64/RISC-V only)

-ffixed-x23

Reserve the x23 register (AArch64/RISC-V only)

-ffixed-x24

Reserve the x24 register (AArch64/RISC-V only)

-ffixed-x25

Reserve the x25 register (AArch64/RISC-V only)

-ffixed-x26

Reserve the x26 register (AArch64/RISC-V only)

-ffixed-x27

Reserve the x27 register (AArch64/RISC-V only)

-ffixed-x28

Reserve the x28 register (AArch64/RISC-V only)

-ffixed-x29

Reserve the x29 register (AArch64/RISC-V only)

-ffixed-x3

Reserve the x3 register (AArch64/RISC-V only)

-ffixed-x30

Reserve the x30 register (AArch64/RISC-V only)

-ffixed-x31

Reserve the x31 register (AArch64/RISC-V only)

-ffixed-x4

Reserve the x4 register (AArch64/RISC-V only)

-ffixed-x5

Reserve the x5 register (AArch64/RISC-V only)

-ffixed-x6

Reserve the x6 register (AArch64/RISC-V only)

-ffixed-x7

Reserve the x7 register (AArch64/RISC-V only)

-ffixed-x8

Reserve the x8 register (AArch64/RISC-V only)

-ffixed-x9

Reserve the x9 register (AArch64/RISC-V only)

-ffuchsia-api-level=<arg>

Set Fuchsia API level

-inline-asm=<arg>

<arg> must be ‘att’ or ‘intel’.

-m16

-m32

-m64

-mabi=<arg>

-mabi=quadword-atomics

Enable quadword atomics ABI on AIX (AIX PPC64 only). Uses lqarx/stqcx. instructions.

-maix-struct-return

Override the default ABI for 32-bit targets to return all structs in memory, as in the Power 32-bit ABI for Linux (2011), and on AIX and Darwin.

-maix32

-maix64

-malign-branch-boundary=<arg>

Specify the boundary’s size to align branches

-malign-branch=<arg1>,<arg2>...

Specify types of branches to align

-malign-double

Align doubles to two words in structs (x86 only)

-mamdgpu-ieee, **-mno-amdgpu-ieee**

Sets the IEEE bit in the expected default floating point mode register. Floating point opcodes that support exception flag gathering quiet and propagate signaling NaN inputs per IEEE 754-2008. This option changes the ABI. (AMDGPU only)

-mamdgpu-precise-memory-op, **-mno-amdgpu-precise-memory-op**

Enable precise memory mode (AMDGPU only)

-march=<arg>

For a list of available architectures for the target use ‘-mcpu=help’

-marm64x<arg>

Link as a hybrid ARM64X image

-masm=<arg>

-mbackchain, **-mno-backchain**

Link stack frames through backchain on System Z

-mbig-endian, **-EB**

-mbranch-protection=<arg>

Enforce targets of indirect branches and function returns

-mbranches-within-32B-boundaries

Align selected branches (fused, jcc, jmp) within 32-byte boundary

-mcmodel=<arg>

-mcode-object-version=<arg>

Specify code object ABI version. Defaults to 5. (AMDGPU only). <arg> must be ‘none’, ‘4’, ‘5’ or ‘6’.

-mconsole<arg>

-mconstructor-aliases, **-mno-constructor-aliases**

Enable emitting complete constructors and destructors as aliases when possible

-mcpu=<arg>, **-mv5** (equivalent to -mcpu=hexagonv5), **-mv55** (equivalent to -mcpu=hexagonv55), **-mv60** (equivalent to -mcpu=hexagonv60), **-mv62** (equivalent to -mcpu=hexagonv62), **-mv65** (equivalent to -mcpu=hexagonv65), **-mv66** (equivalent to -mcpu=hexagonv66), **-mv67** (equivalent to -mcpu=hexagonv67), **-mv67t** (equivalent to -mcpu=hexagonv67t), **-mv68** (equivalent to -mcpu=hexagonv68), **-mv69** (equivalent to -mcpu=hexagonv69), **-mv71** (equivalent to -mcpu=hexagonv71), **-mv71t** (equivalent to -mcpu=hexagonv71t), **-mv73** (equivalent to -mcpu=hexagonv73)

For a list of available CPUs for the target use ‘-mcpu=help’

-mcrc, **-mno-crc**

Allow use of CRC instructions (ARM/Mips only)

-mdaz-ftz, **-mno-daz-ftz**

Globally set the denormals-are-zero (DAZ) and flush-to-zero (FTZ) bits in the floating-point control register on program startup

-mdefault-build-attributes<arg>, **-mno-default-build-attributes<arg>**

-mdefault-visibility-export-mapping=<arg>

Mapping between default visibility and export. <arg> must be ‘none’, ‘explicit’ or ‘all’.

-mdll<arg>

-mdouble-float

-mdouble=<n

Force double to be <n> bits. <n must be ‘32’ or ‘64’.

-mdynamic-no-pic<arg>

-meabi <arg>

Set EABI type. Default depends on triple). <arg> must be ‘default’, ‘4’, ‘5’ or ‘gnu’.

-menable-experimental-extensions

Enable use of experimental RISC-V extensions.

-mfentry

Insert calls to fentry at function entry (x86/SystemZ only)

-mfloat-abi=<arg>

<arg> must be ‘soft’, ‘softfp’ or ‘hard’.

-mfpmath=<arg>

-mfpu=<arg>

-mfunction-return=<arg>

Replace returns with jumps to ``__x86_return_thunk`` (x86 only, error otherwise). <arg> must be ‘keep’ or ‘thunk-extern’.

-mgeneral-regs-only

Generate code which only uses the general purpose registers (AArch64/x86 only)

-mglobal-merge, **-mno-global-merge**

Enable merging of globals

-mguard=<arg>

Enable or disable Control Flow Guard checks and guard tables emission. <arg> must be ‘none’, ‘cf’ or ‘cf-nochecks’.

-mhard-float

-mharden-sls=<arg>

Select straight-line speculation hardening scope (ARM/AArch64/X86 only). <arg> must be: all, none, retbr(ARM/AArch64), blr(ARM/AArch64), comdat(ARM/AArch64), nocomdat(ARM/AArch64), return(X86), indirect-jmp(X86)

-mhwdiv=<arg>, **--mhwdiv** <arg>, **--mhwdiv**=<arg>

-mhwmult=<arg>

-miamcu, **-mno-iamcu**

Use Intel MCU ABI

-mignore-xcoff-visibility

Not emit the visibility attribute for asm in AIX OS or give all symbols ‘unspecified’ visibility in XCOFF object file

-mimplicit-float, **-mno-implicit-float**

-mimplicit-it=<arg>

-mincremental-linker-compatible, **-mno-incremental-linker-compatible**

(integrated-as) Emit an object file which can be used with an incremental linker

-mindirect-branch-cs-prefix

Add cs prefix to call and jmp to indirect thunk

-mios-simulator-version-min=<arg>, **-miphonesimulator-version-min**=<arg>

-mios-version-min=<arg>, **-miphoneos-version-min**=<arg>

Set iOS deployment target

-mkernel

-mlarge-data-threshold=<arg>

-mlink-builtin-bitcode-postopt, **-mno-link-builtin-bitcode-postopt**

Link builtin bitcodes after the optimization pipeline

-mlinker-version=<arg>

-mlittle-endian, **-EL**

-mlong-calls, **-mno-long-calls**

Generate branches with extended addressability, usually via indirect jumps.

-mlvi-cfi, **-mno-lvi-cfi**

Enable only control-flow mitigations for Load Value Injection (LVI)

-mlvi-hardening, **-mno-lvi-hardening**

Enable all mitigations for Load Value Injection (LVI)

-mmacos-version-min=<arg>, **-mmacosx-version-min**=<arg>

Set macOS deployment target

-mmcu=<arg>

-mms-bitfields, **-mno-ms-bitfields**

Set the default structure layout to be compatible with the Microsoft compiler standard

-mno-gather

Disable generation of gather instructions in auto-vectorization(x86 only)

-mno-scatter

Disable generation of scatter instructions in auto-vectorization(x86 only)

-mnop-mcount

Generate mcount/__fentry__ calls as nops. To activate they need to be patched in.

-momit-leaf-frame-pointer, **-mno-omit-leaf-frame-pointer**

Omit frame pointer setup for leaf functions

-moslib=<arg>

-mpacked-stack, **-mno-packed-stack**

Use packed stack layout (SystemZ only).

-mpad-max-prefix-size=<arg>

Specify maximum number of prefixes to use for padding

-mpic-data-is-text-relative, **-mno-pic-data-is-text-relative**

Assume data segments are relative to text segment

-mprefer-vector-width=<arg>

Specifies preferred vector width for auto-vectorization. Defaults to ‘none’ which allows target specific decisions.

-mprintf-kind=<arg>

Specify the printf lowering scheme (AMDGPU only), allowed values are “hostcall”(printing happens during kernel execution, this scheme relies on hostcalls which require system to support pcie atomics) and “buffered”(printing happens after all kernel threads exit, this uses a printf buffer and does not rely on pcie atomic support). <arg> must be ‘hostcall’ or ‘buffered’.

-mqdsp6-compat

Enable hexagon-qdsp6 backward compatibility

-mrecip

Equivalent to ‘-mrecip=all’

-mrecip=<arg1>,<arg2>...

Control use of approximate reciprocal and reciprocal square root instructions followed by <n> iterations of Newton-Raphson refinement. <value> = ([‘!’] [‘vec-’] (‘rcp’|’sqrt’) [(‘h’|’s’|’d’)] [‘:’<n>]) | ‘all’ | ‘default’ | ‘none’

-mrecord-mcount

Generate a __mcount_loc section entry for each __fentry__ call.

-mred-zone, **-mno-red-zone**

-mregnames, **-mno-regnames**

Use full register names when writing assembly output

-mregparm=<arg>

-mrelax, **-mno-relax**

Enable linker relaxation

`-mrelax-all`, **`-mno-relax-all`**

(integrated-as) Relax all machine instructions

`-mretpoline`, **`-mno-retpoline`**

`-mrtd`, **`-mno-rtd`**

Make StdCall calling convention the default

`-mrvv-vector-bits``=<arg>`

Defaults to the vector length agnostic value of “scalable”. Accepts power of 2 values between 64 and 65536. Also accepts “zvl” to use the value implied by -march/-mcpu. The value will be reflected in __riscv_v_fixed_vlen preprocessor define (RISC-V only)

`-mseses`, **`-mno-seses`**

Enable speculative execution side effect suppression (SESES). Includes LVI control flow integrity mitigations

`-msign-return-address``=<arg>`

Select return address signing scope. <arg> must be ‘none’, ‘all’ or ‘non-leaf’.

`-msim`

`-msingle-float`

`-mskip-rax-setup`, **`-mno-skip-rax-setup`**

Skip setting up RAX register when passing variable arguments (x86 only)

`-msoft-float`, **`-mno-soft-float`**

Use software floating point

`-mspeculative-load-hardening`, **`-mno-speculative-load-hardening`**

`-mstack-alignment``=<arg>`

Set the stack alignment

`-mstack-arg-probe`, **`-mno-stack-arg-probe`**

Enable stack probes

`-mstack-probe-size``=<arg>`

Set the stack probe size

`-mstack-protector-guard-offset``=<arg>`

Use the given offset for addressing the stack-protector guard

`-mstack-protector-guard-reg``=<arg>`

Use the given reg for addressing the stack-protector guard

`-mstack-protector-guard-symbol``=<arg>`

Use the given symbol for addressing the stack-protector guard

`-mstack-protector-guard``=<arg>`

Use the given guard (global, tls) for addressing the stack-protector guard

`-mstackrealign`, **`-mno-stackrealign`**

Force realign the stack at entry to every function

-mstrict-align, **-mno-strict-align**

Force all memory accesses to be aligned (AArch64/LoongArch/RISC-V only)

-msvr4-struct-return

Override the default ABI for 32-bit targets to return small structs in registers, as in the System V ABI (1995).

-mtargetos=<arg>

Set the deployment target to be the specified OS and OS version

-mthread-model <arg>

The thread model to use. Defaults to ‘posix’). <arg> must be ‘posix’ or ‘single’.

-mthreads<arg>

-mthumb, **-mno-thumb**

-mtls-dialect=<arg>

Which thread-local storage dialect to use for dynamic accesses of TLS variables

-mtls-direct-seg-refs, **-mno-tls-direct-seg-refs**

Enable direct TLS access through segment registers (default)

-mtls-size=<arg>

Specify bit size of immediate TLS offsets (AArch64 ELF only): 12 (for 4KB) | 24 (for 16MB, default) | 32 (for 4GB) | 48 (for 256TB, needs -mcmodel=large)

-mtocdata, **-mno-tocdata**

All suitable variables will have the TOC data transformation applied

-mtocdata=<arg1>,<arg2>..., **-mno-tocdata**=<arg1>,<arg2>...

Specifies a list of variables to which the TOC data transformationwill be applied.

-mtune=<arg>

Only supported on AArch64, PowerPC, RISC-V, SPARC, SystemZ, and X86

-mtvos-version-min=<arg>, **-mappletvos-version-min**=<arg>

-munaligned-access, **-mno-unaligned-access**

Allow memory accesses to be unaligned (AArch32/MIPSr6 only)

-munaligned-symbols, **-mno-unaligned-symbols**

Expect external char-aligned symbols to be without ABI alignment (SystemZ only)

-municode<arg>

-munsafe-fp-atomics, **-mno-unsafe-fp-atomics**

Enable generation of unsafe floating point atomic instructions. May generate more efficient code, but may not respect rounding and denormal modes, and may give incorrect results for certain memory destinations. (AMDGPU only)

-mvx, **-mno-vx**

-mwarn-nonportable-cfstrings, **-mno-warn-nonportable-cfstrings**

-mwatchos-simulator-version-min=<arg>, **-mwatchsimulator-version-min**=<arg>

-mwatchos-version-min=<arg>

-mwavefrontsize64, **-mno-wavefrontsize64**

Specify wavefront size 64 mode (AMDGPU only)

-mwindows<arg>

-mx32

-mxcoff-roptr, **-mno-xcoff-roptr**

Place constant objects with relocatable address values in the RO data section and add -bforceimprw to the linker flags (AIX only)

-regcall4

Set __regcall4 as a default calling convention to respect __regcall ABI v.4

AARCH64

-fcall-saved-x10

Make the x10 register call-saved (AArch64 only)

-fcall-saved-x11

Make the x11 register call-saved (AArch64 only)

-fcall-saved-x12

Make the x12 register call-saved (AArch64 only)

-fcall-saved-x13

Make the x13 register call-saved (AArch64 only)

-fcall-saved-x14

Make the x14 register call-saved (AArch64 only)

-fcall-saved-x15

Make the x15 register call-saved (AArch64 only)

-fcall-saved-x18

Make the x18 register call-saved (AArch64 only)

-fcall-saved-x8

Make the x8 register call-saved (AArch64 only)

-fcall-saved-x9

Make the x9 register call-saved (AArch64 only)

-mfix-cortex-a53-835769, **-mno-fix-cortex-a53-835769**

Workaround Cortex-A53 erratum 835769 (AArch64 only)

-mmark-bti-property

Add .note.gnu.property with BTI to assembly files (AArch64 only)

`-msve-vector-bits``=<arg>`

Specify the size in bits of an SVE vector register. Defaults to the vector length agnostic value of “scalable”. (AArch64 only)

AMDGPU

`-mcumode`, **`-mno-cumode`**

Specify CU wavefront execution mode (AMDGPU only)

`-mtgsplit`, **`-mno-tgsplit`**

Enable threadgroup split execution mode (AMDGPU only)

ARM

`-faapcs-bitfield-load`

Follows the AAPCS standard that all volatile bit-field write generates at least one load. (ARM only).

`-faapcs-bitfield-width`, **`-fno-aapcs-bitfield-width`**

Follow the AAPCS standard requirement stating that volatile bit-field width is dictated by the field container type. (ARM only).

`-ffixed-r9`

Reserve the r9 register (ARM only)

`-mcmse`

Allow use of CMSE (Armv8-M Security Extensions)

`-mexecute-only`, **`-mno-execute-only`**, **`-mpure-code`**

Disallow generation of data access to code sections (ARM only)

`-mfix-cmse-cve-2021-35465`, **`-mno-fix-cmse-cve-2021-35465`**

Work around VLLDM erratum CVE-2021-35465 (ARM only)

`-mfix-cortex-a57-aes-1742098`, **`-mfix-cortex-a72-aes-1655431`**, **`-mno-fix-cortex-a57-aes-1742098`**

Work around Cortex-A57 Erratum 1742098 (ARM only)

`-mframe-chain``=<arg>`

Select the frame chain model used to emit frame records (Arm only). <arg> must be ‘none’, ‘aapcs’ or ‘aapcs+leaf’.

`-mno-bti-at-return-twice`

Do not add a BTI instruction after a setjmp or other return-twice construct (Arm/AArch64 only)

`-mno-movt`

Disallow use of movt/movw pairs (ARM only)

`-mno-neg-immediates`

Disallow converting instructions with negative immediates to their negation or inversion.

`-mnocrc`

Disallow use of CRC instructions (ARM only)

`-mrestrict-it`, **`-mno-restrict-it`**

Disallow generation of complex IT blocks. It is off by default.

`-mtp``=<arg>`

Thread pointer access method. For AArch32: ‘soft’ uses a function call, or ‘tpidrurw’, ‘tpidruro’ or ‘tpidrprw’ use the three CP15 registers. ‘cp15’ is an alias for ‘tpidruro’. For AArch64: ‘tpidr_el0’, ‘tpidr_el1’, ‘tpidr_el2’, ‘tpidr_el3’ or ‘tpidrro_el0’ use the five system registers. ‘elN’ is an alias for ‘tpidr_elN’. <arg> must be ‘soft’, ‘cp15’, ‘tpidrurw’, ‘tpidruro’, ‘tpidrprw’, ‘el0’, ‘el1’, ‘el2’, ‘el3’, ‘tpidr_el0’, ‘tpidr_el1’, ‘tpidr_el2’, ‘tpidr_el3’ or ‘tpidrro_el0’.

Hexagon

`-mcabac`

Enable CABAC instructions

`-mhvx-ieee-fp`, **`-mno-hvx-ieee-fp`**

Enable Hexagon HVX IEEE floating-point

`-mieee-rnd-near`

`-mmemops`, **`-mno-memops`**

Enable generation of memop instructions

`-mnvj`, **`-mno-nvj`**

Enable generation of new-value jumps

`-mnvs`, **`-mno-nvs`**

Enable generation of new-value stores

`-mpackets`, **`-mno-packets`**

Enable generation of instruction packets

SPARC

`-ffixed-g1`

Reserve the G1 register (SPARC only)

`-ffixed-g2`

Reserve the G2 register (SPARC only)

`-ffixed-g3`

Reserve the G3 register (SPARC only)

`-ffixed-g4`

Reserve the G4 register (SPARC only)

`-ffixed-g5`

Reserve the G5 register (SPARC only)

`-ffixed-g6`

Reserve the G6 register (SPARC only)

-ffixed-g7

Reserve the G7 register (SPARC only)

-ffixed-i0

Reserve the I0 register (SPARC only)

-ffixed-i1

Reserve the I1 register (SPARC only)

-ffixed-i2

Reserve the I2 register (SPARC only)

-ffixed-i3

Reserve the I3 register (SPARC only)

-ffixed-i4

Reserve the I4 register (SPARC only)

-ffixed-i5

Reserve the I5 register (SPARC only)

-ffixed-l0

Reserve the L0 register (SPARC only)

-ffixed-l1

Reserve the L1 register (SPARC only)

-ffixed-l2

Reserve the L2 register (SPARC only)

-ffixed-l3

Reserve the L3 register (SPARC only)

-ffixed-l4

Reserve the L4 register (SPARC only)

-ffixed-l5

Reserve the L5 register (SPARC only)

-ffixed-l6

Reserve the L6 register (SPARC only)

-ffixed-l7

Reserve the L7 register (SPARC only)

-ffixed-o0

Reserve the O0 register (SPARC only)

-ffixed-o1

Reserve the O1 register (SPARC only)

-ffixed-o2

Reserve the O2 register (SPARC only)

-ffixed-o3

Reserve the O3 register (SPARC only)

-ffixed-o4

Reserve the O4 register (SPARC only)

-ffixed-o5

Reserve the O5 register (SPARC only)

-mfpu, **-mno-fpu**

-mfsmuld, **-mno-fsmuld**

-mhard-quad-float

-mpopc, **-mno-popc**

-msoft-quad-float

-mvis, **-mno-vis**

-mvis2, **-mno-vis2**

-mvis3, **-mno-vis3**

Hexagon

-mhvx, **-mno-hvx**

Enable Hexagon Vector eXtensions

-mhvx-length=<arg>

Set Hexagon Vector Length. <arg> must be ‘64B’ or ‘128B’.

-mhvx-qfloat, **-mno-hvx-qfloat**

Enable Hexagon HVX QFloat instructions

-mhvx=<arg>

Enable Hexagon Vector eXtensions

M68k

-ffixed-a0

Reserve the a0 register (M68k only)

-ffixed-a1

Reserve the a1 register (M68k only)

-ffixed-a2

Reserve the a2 register (M68k only)

-ffixed-a3

Reserve the a3 register (M68k only)

-ffixed-a4

Reserve the a4 register (M68k only)

-ffixed-a5

Reserve the a5 register (M68k only)

-ffixed-a6

Reserve the a6 register (M68k only)

-ffixed-d0

Reserve the d0 register (M68k only)

-ffixed-d1

Reserve the d1 register (M68k only)

-ffixed-d2

Reserve the d2 register (M68k only)

-ffixed-d3

Reserve the d3 register (M68k only)

-ffixed-d4

Reserve the d4 register (M68k only)

-ffixed-d5

Reserve the d5 register (M68k only)

-ffixed-d6

Reserve the d6 register (M68k only)

-ffixed-d7

Reserve the d7 register (M68k only)

-m68000

-m68010

-m68020

-m68030

-m68040

-m68060

-m68881

MIPS

-mabicalls, **-mno-abicalls**

Enable SVR4-style position-independent code (Mips only)

-mabs=<arg>

-mcheck-zero-division, **-mno-check-zero-division**

-mcompact-branches=<arg>

-mdsp, **-mno-dsp**

-mdspr2, **-mno-dspr2**

-membedded-data, **-mno-embedded-data**

Place constants in the .rodata section instead of the .sdata section even if they meet the -G <size> threshold (MIPS)

-mextern-sdata, **-mno-extern-sdata**

Assume that externally defined data is in the small data if it meets the -G <size> threshold (MIPS)

-mfix4300

-mfp32

Use 32-bit floating point registers (MIPS only)

-mfp64

Use 64-bit floating point registers (MIPS only)

-mginv, **-mno-ginv**

-mgpopt, **-mno-gpopt**

Use GP relative accesses for symbols known to be in a small data section (MIPS)

-mindirect-jump=<arg>

Change indirect jump instructions to inhibit speculation

-mips16

-mldc1-sdc1, **-mno-ldc1-sdc1**

-mlocal-sdata, **-mno-local-sdata**

Extend the -G behaviour to object local data (MIPS)

-mmadd4, **-mno-madd4**

Enable the generation of 4-operand madd.s, madd.d and related instructions.

-mmicromips, **-mno-micromips**

-mmsa, **-mno-msa**

Enable MSA ASE (MIPS only)

-mmt, **-mno-mt**

Enable MT ASE (MIPS only)

-mnan=<arg>

-mno-mips16

-mvirt, **-mno-virt**

-mxgot, **-mno-xgot**

PowerPC

-maix-shared-lib-tls-model-opt

For shared library loaded with the main program, change local-dynamic access(es) to initial-exec access(es) at the function level (AIX 64-bit only).

-maix-small-local-dynamic-tls

Produce a faster access sequence for local-dynamic TLS variables where the offset from the TLS base is encoded as an immediate operand (AIX 64-bit only). This access sequence is not used for variables larger than 32KB.

-maix-small-local-exec-tls

Produce a faster access sequence for local-exec TLS variables where the offset from the TLS base is encoded as an immediate operand (AIX 64-bit only). This access sequence is not used for variables larger than 32KB.

-maltivec, **-mno-altivec**

Enable AltiVec vector initializer syntax

-mcmpb, **-mno-cmpb**

-mcrbits, **-mno-crbits**

Control the CR-bit tracking feature on PowerPC. ``-mcrbits`` (the enablement of CR-bit tracking support) is the default for POWER8 and above, as well as for all other CPUs when optimization is applied (-O2 and above).

-mcrypto, **-mno-crypto**

-mdirect-move, **-mno-direct-move**

-mefpu2

-mfloat128, **-mno-float128**

-mfprnd, **-mno-fprnd**

-mhtm, **-mno-htm**

-minvariant-function-descriptors, **-mno-invariant-function-descriptors**

-misel, **-mno-isel**

-mlongcall, **-mno-longcall**

-mmfocrf, **-mmfcrf**, **-mno-mfocrf**

-mmma, **-mno-mma**

-mpaired-vector-memops, **-mno-paired-vector-memops**

-mpcrel, **-mno-pcrel**

-mpopcntd, **-mno-popcntd**

-mpower10-vector, **-mno-power10-vector**

-mpower8-vector, **-mno-power8-vector**

-mpower9-vector, **-mno-power9-vector**

-mprefixed, **-mno-prefixed**

-mprivileged

-mrop-protect

-msecure-plt

-mspe, **-mno-spe**

-mvsx, **-mno-vsx**

WebAssembly

-matomics, **-mno-atomics**

-mbulk-memory, **-mno-bulk-memory**

-mexception-handling, **-mno-exception-handling**

-mextended-const, **-mno-extended-const**

-mhalf-precision, **-mno-half-precision**

-mmultimemory, **-mno-multimemory**

-mmultivalue, **-mno-multivalue**

-mmutable-globals, **-mno-mutable-globals**

-mnontrapping-fptoint, **-mno-nontrapping-fptoint**

-mreference-types, **-mno-reference-types**

-mrelaxed-simd, **-mno-relaxed-simd**

-msign-ext, **-mno-sign-ext**

-msimd128, **-mno-simd128**

-mtail-call, **-mno-tail-call**

WebAssembly Driver

-mexec-model=<arg>

Select between “command” and “reactor” executable models. Commands have a main-function which scopes the lifetime of the program. Reactors are activated and remain active until explicitly terminated. <arg> must be ‘command’ or ‘reactor’.

X86

-m3dnow, **-mno-3dnow**

-m3dnowa, **-mno-3dnowa**

-madx, **-mno-adx**

-maes, **-mno-aes**

-mamx-bf16, **-mno-amx-bf16**

-mamx-complex, **-mno-amx-complex**

-mamx-fp16, **-mno-amx-fp16**

-mamx-int8, **-mno-amx-int8**

-mamx-tile, **-mno-amx-tile**

-mapx-features=<arg1>,<arg2>..., **-mapxf** (equivalent to -mapx-features=egpr,push2pop2,ppx,ndd), **-mno-apx-features**=<arg1>,<arg2>...

Enable features of APX. <arg> must be ‘egpr’, ‘push2pop2’, ‘ppx’, ‘ndd’, ‘ccmp’, ‘nf’ or ‘cf’.

-mavx, **-mno-avx**

-mavx2, **-mno-avx2**

-mavx512bf16, **-mno-avx512bf16**

-mavx512bitalg, **-mno-avx512bitalg**

-mavx512bw, **-mno-avx512bw**

-mavx512cd, **-mno-avx512cd**

-mavx512dq, **-mno-avx512dq**

-mavx512er, **-mno-avx512er**

-mavx512f, **-mno-avx512f**

-mavx512fp16, **-mno-avx512fp16**

-mavx512ifma, **-mno-avx512ifma**

-mavx512pf, **-mno-avx512pf**

-mavx512vbmi, **-mno-avx512vbmi**

-mavx512vbmi2, **-mno-avx512vbmi2**

-mavx512vl, **-mno-avx512vl**

-mavx512vnni, **-mno-avx512vnni**

-mavx512vp2intersect, **-mno-avx512vp2intersect**

-mavx512vpopcntdq, **-mno-avx512vpopcntdq**

-mavxifma, **-mno-avxifma**

-mavxneconvert, **-mno-avxneconvert**

-mavxvnni, **-mno-avxvnni**

-mavxvnniint16, **-mno-avxvnniint16**

-mavxvnniint8, **-mno-avxvnniint8**

-mbmi, **-mno-bmi**

-mbmi2, **-mno-bmi2**

-mcldemote, **-mno-cldemote**

-mclflushopt, **-mno-clflushopt**

-mclwb, **-mno-clwb**

-mclzero, **-mno-clzero**

-mcmpccxadd, **-mno-cmpccxadd**

-mcrc32, **-mno-crc32**

-mcx16, **-mno-cx16**

-menqcmd, **-mno-enqcmd**

-mevex512, **-mno-evex512**

-mf16c, **-mno-f16c**

-mfma, **-mno-fma**

-mfma4, **-mno-fma4**

-mfsgsbase, **-mno-fsgsbase**

-mfxsr, **-mno-fxsr**

-mgfni, **-mno-gfni**

-mhreset, **-mno-hreset**

-minvpcid, **-mno-invpcid**

-mkl, **-mno-kl**

-mlwp, **-mno-lwp**

-mlzcnt, **-mno-lzcnt**

-mmmx, **-mno-mmx**

-mmovbe, **-mno-movbe**

-mmovdir64b, **-mno-movdir64b**

-mmovdiri, **-mno-movdiri**

-mmwaitx, **-mno-mwaitx**

-mpclmul, **-mno-pclmul**

-mpconfig, **-mno-pconfig**

-mpku, **-mno-pku**

-mpopcnt, **-mno-popcnt**

-mprefetchi, **-mno-prefetchi**

-mprefetchwt1, **-mno-prefetchwt1**

-mprfchw, **-mno-prfchw**

-mptwrite, **-mno-ptwrite**

-mraoint, **-mno-raoint**

-mrdpid, **-mno-rdpid**

-mrdpru, **-mno-rdpru**

-mrdrnd, **-mno-rdrnd**

-mrdseed, **-mno-rdseed**

-mretpoline-external-thunk, **-mno-retpoline-external-thunk**

-mrtm, **-mno-rtm**

-msahf, **-mno-sahf**

-mserialize, **-mno-serialize**

-msgx, **-mno-sgx**

-msha, **-mno-sha**

-msha512, **-mno-sha512**

-mshstk, **-mno-shstk**

-msm3, **-mno-sm3**

-msm4, **-mno-sm4**

-msse, **-mno-sse**

-msse2, **-mno-sse2**

-msse3, **-mno-sse3**

-msse4.1, **-mno-sse4.1**

-msse4.2, **-mno-sse4.2**, **-msse4**

-msse4a, **-mno-sse4a**

-mssse3, **-mno-ssse3**

-mtbm, **-mno-tbm**

-mtsxldtrk, **-mno-tsxldtrk**

-muintr, **-mno-uintr**

-musermsr, **-mno-usermsr**

-mvaes, **-mno-vaes**

-mvpclmulqdq, **-mno-vpclmulqdq**

-mvzeroupper, **-mno-vzeroupper**

-mwaitpkg, **-mno-waitpkg**

-mwbnoinvd, **-mno-wbnoinvd**

-mwidekl, **-mno-widekl**

-mx87, **-m80387**, **-mno-x87**

-mxop, **-mno-xop**

-mxsave, **-mno-xsave**

-mxsavec, **-mno-xsavec**

-mxsaveopt, **-mno-xsaveopt**

-mxsaves, **-mno-xsaves**

X86 AVX10

-mavx10.1-256, **-mavx10.1**, **-mno-avx10.1-256**

-mavx10.1-512, **-mno-avx10.1-512**

RISC-V

-mforced-sw-shadow-stack, **-mno-forced-sw-shadow-stack**

Force using software shadow stack when shadow-stack enabled

-msave-restore, **-mno-save-restore**

Enable using library calls for save and restore

VE

-mvevpu, **-mno-vevpu**

Emit VPU instructions for VE

LoongArch

-mlasx, **-mno-lasx**

Enable Loongson Advanced SIMD Extension (LASX).

-mlsx, **-mno-lsx**

Enable Loongson SIMD Extension (LSX).

Long double options

Selects the long double implementation

-mlong-double-128

Force long double to be 128 bits

-mlong-double-64

Force long double to be 64 bits

-mlong-double-80

Force long double to be 80 bits, padded to 128 bits for storage

Optimization level

Flags controlling how much optimization should be performed.

-O<arg>, **-O** (equivalent to -O1), **--optimize**, **--optimize**=<arg>

-Ofast<arg>

Debug information generation

Flags controlling how much and what kind of debug information should be generated.

Kind and level of debug information

-g, **--debug**, **--debug**=<arg>

Generate source-level debug information

-gdwarf

Generate source-level debug information with the default dwarf version

-gdwarf-2

Generate source-level debug information with dwarf version 2

-gdwarf-3

Generate source-level debug information with dwarf version 3

-gdwarf-4

Generate source-level debug information with dwarf version 4

-gdwarf-5

Generate source-level debug information with dwarf version 5

-gdwarf32

Enables DWARF32 format for ELF binaries, if debug information emission is enabled.

-gdwarf64

Enables DWARF64 format for ELF binaries, if debug information emission is enabled.

-gfull

-ginline-line-tables, **-gno-inline-line-tables**

-gused

Debug level

-g0

-g2

-g3

-ggdb0

-ggdb1

-ggdb2

-ggdb3

-gline-directives-only

Emit debug line info directives only

-gline-tables-only, **-g1**, **-gmlt**

Emit debug line number tables only

-gmodules, **-gno-modules**

Generate debug info with external references to clang modules or precompiled headers

Debugger to tune debug information for

-gdbx

-ggdb

-glldb

-gsce

Debug information options

-gcolumn-info, **-gno-column-info**

-gdwarf-aranges

-gembed-source, **-gno-embed-source**

Embed source text in DWARF debug sections

-ggnu-pubnames, **-gno-gnu-pubnames**

-gpubnames, **-gno-pubnames**

-grecord-command-line, **-gno-record-command-line**, **-grecord-gcc-switches**

-gsimple-template-names, **-gno-simple-template-names**

-gsplit-dwarf, **-gno-split-dwarf**

-gsplit-dwarf=<arg>

Set DWARF fission mode. <arg> must be ‘split’ or ‘single’.

-gstrict-dwarf, **-gno-strict-dwarf**

Restrict DWARF features to those defined in the specified version, avoiding features from later versions.

-gtemplate-alias, **-gno-template-alias**

-gz=<arg>, **-gz** (equivalent to -gz=zlib)

DWARF debug sections compression type

Static analyzer options

Flags controlling the behavior of the Clang Static Analyzer.

-Xanalyzer <arg>

Pass <arg> to the static analyzer

Fortran compilation options

Flags that will be passed onto the `gfortran` compiler when Clang is given a Fortran input.

-A<arg>, **--assert** <arg>, **--assert**=<arg>

-A-<arg>

-faggressive-function-elimination, **-fno-aggressive-function-elimination**

-falign-commons, **-fno-align-commons**

-fall-intrinsics, **-fno-all-intrinsics**

-fbacktrace, **-fno-backtrace**

-fblas-matmul-limit=<arg>

-fbounds-check, **-fno-bounds-check**

-fcheck-array-temporaries, **-fno-check-array-temporaries**

-fcheck=<arg>

-fcoarray=<arg>

-fcray-pointer, **-fno-cray-pointer**

-fd-lines-as-code, **-fno-d-lines-as-code**

-fd-lines-as-comments, **-fno-d-lines-as-comments**

-fdollar-ok, **-fno-dollar-ok**

-fdump-fortran-optimized, **-fno-dump-fortran-optimized**

-fdump-fortran-original, **-fno-dump-fortran-original**

-fdump-parse-tree, **-fno-dump-parse-tree**

-fexternal-blas, **-fno-external-blas**

-ff2c, **-fno-f2c**

-ffpe-trap=<arg>

-ffree-line-length-<arg>

-ffrontend-optimize, **-fno-frontend-optimize**

-finit-character=<arg>

-finit-integer=<arg>

-finit-local-zero, **-fno-init-local-zero**

-finit-logical=<arg>

-finit-real=<arg>

-finteger-4-integer-8, **-fno-integer-4-integer-8**

-fmax-array-constructor=<arg>

-fmax-errors=<arg>

-fmax-identifier-length, **-fno-max-identifier-length**

-fmax-stack-var-size=<arg>

-fmax-subrecord-length=<arg>

-fmodule-private, **-fno-module-private**

-fpack-derived, **-fno-pack-derived**

-frange-check, **-fno-range-check**

-freal-4-real-10, **-fno-real-4-real-10**

-freal-4-real-16, **-fno-real-4-real-16**

-freal-4-real-8, **-fno-real-4-real-8**

-freal-8-real-10, **-fno-real-8-real-10**

-freal-8-real-16, **-fno-real-8-real-16**

-freal-8-real-4, **-fno-real-8-real-4**

-frealloc-lhs, **-fno-realloc-lhs**

-frecord-marker=<arg>

-frecursive, **-fno-recursive**

-frepack-arrays, **-fno-repack-arrays**

-fsecond-underscore, **-fno-second-underscore**

-fsign-zero, **-fno-sign-zero**

-fwhole-file, **-fno-whole-file**

-imultilib <arg>

-static-libgfortran

Linker options

Flags that are passed on to the linker

-L<dir>, **--library-directory** <arg>, **--library-directory**=<arg>

Add directory to library search path

-Mach

-T<script>

Specify <script> as linker script

-Wl,<arg>,<arg2>...

Pass the comma separated arguments in <arg> to the linker

-X

-Xlinker <arg>, **--for-linker** <arg>, **--for-linker**=<arg>

Pass <arg> to the linker

-Xoffload-linker<triple> <arg>

Pass <arg> to the offload linkers or the ones idenfied by -<triple>

-Z

-b<arg>

Pass -b <arg> to the linker on AIX

-coverage, **--coverage**

-e <arg>, **--entry**

-filelist <arg>

-l<arg>

--ld-path=<arg>

-mxcoff-build-id=<0xHEXSTRING>

On AIX, request creation of a build-id string, “0xHEXSTRING”, in the string table of the loader section inside the linked binary

-nostartfiles

-nostdlib, **--no-standard-libraries**

--offload-link

Use the new offloading linker to perform the link job.

-pie, **-no-pie**

-r

-rdynamic

-rpath <arg>

-s

-shared, **--shared**

-specs=<arg>, **--specs**=<arg>

-static, **--static**

-static-pie

`-t`

`-u<arg>`, **`--force-link`** `<arg>`, **`--force-link`**`=<arg>`

`-undef`

undef all system defines

`-undefined<arg>`, **`--no-undefined`**

`-z` `<arg>`

Pass -z <arg> to the linker

clang-dxc options

dxc compatibility options.

`--dxv-path``=<arg>`

DXIL validator installation path

`-fspv-target-env``=<arg>`

Specify the target environment. <arg> must be ‘vulkan1.2’ or ‘ vulkan1.3’.

`-hlsl-entry` `<arg>`

Entry point name for hlsl