```
Usage: pic32-gcc [options] file...
Options:
  -pass-exit-codes         Exit with highest error code from a phase
  --help                   Display this information
  --target-help            Display target specific command line options
  --help={target|optimizers|warnings|params|[^]{joined|separate|undocumented}}[,...]
                           Display specific types of command line options
  (Use '-v --help' to display command line options of sub-processes)
  --version                Display compiler version information
  -dumpspecs               Display all of the built in spec strings
  -dumpversion             Display the version of the compiler
  -dumpmachine             Display the compiler's target processor
  -print-search-dirs       Display the directories in the compiler's search path
  -print-libgcc-file-name  Display the name of the compiler's companion library
  -print-file-name=<lib>   Display the full path to library <lib>
  -print-prog-name=<prog>  Display the full path to compiler component <prog>
  -print-multi-directory   Display the root directory for versions of libgcc
  -legacy-libc             Use legacy (pre v1.12) lib C routines
  -print-multi-lib         Display the mapping between command line options and
                           multiple library search directories
  -print-multi-os-directory Display the relative path to OS libraries
  -print-sysroot           Display the target libraries directory
  -print-sysroot-headers-suffix Display the sysroot suffix used to find headers
  -Wa,<options>            Pass comma-separated <options> on to the assembler
  -Wp,<options>            Pass comma-separated <options> on to the preprocessor
  -Wl,<options>            Pass comma-separated <options> on to the linker
  -Xassembler <arg>        Pass <arg> on to the assembler
  -Xpreprocessor <arg>     Pass <arg> on to the preprocessor
  -Xlinker <arg>           Pass <arg> on to the linker
  -combine                 Pass multiple source files to compiler at once
  -save-temps              Do not delete intermediate files
  -save-temps=<arg>        Do not delete intermediate files
  -no-canonical-prefixes   Do not canonicalize paths when building relative
                           prefixes to other gcc components
  -pipe                    Use pipes rather than intermediate files
  -time                    Time the execution of each subprocess
  -specs=<file>            Override built-in specs with the contents of <file>
  -std=<standard>          Assume that the input sources are for <standard>
  --sysroot=<directory>    Use <directory> as the root directory for headers
                           and libraries
  -B <directory>           Add <directory> to the compiler's search paths
  -b <machine>             Run gcc for target <machine>, if installed
  -V <version>             Run gcc version number <version>, if installed
  -v                       Display the programs invoked by the compiler
  -###                     Like -v but options quoted and commands not executed
  -E                       Preprocess only; do not compile, assemble or link
  -S                       Compile only; do not assemble or link
  -c                       Compile and assemble, but do not link
  -o <file>                Place the output into <file>
  -x <language>            Specify the language of the following input files
                           Permissible languages include: c c++ assembler none
                           'none' means revert to the default behavior of
                           guessing the language based on the file's extension

Options starting with -g, -f, -m, -O, -W, or --param are automatically
 passed on to the various sub-processes invoked by pic32-gcc.  In order to pass
 other options on to these processes the -W<letter> options must be used.

For bug reporting instructions, please see:
<http://www.microchip.com/support>.
```

The following options are target specific:

-falign-arrays	Set the minimum alignment for array variables to be the largest power of two less than or equal to their total storage size, or the biggest alignment used on the machine, whichever is smaller.
-mappio-debug	Enable APPIO debug-support library functions
-mbranch-cost=COST	Set the cost of branches to roughly COST instructions
-mbranch-likely	Use Branch Likely instructions, overriding the architecture default
-mcheck-zero-division	Trap on integer divide by zero
-mdebugger	Allocate space for debugger executive
-mdivide-breaks	Use branch-and-break sequences to check for integer divide by zero
-mdivide-traps	Use trap instructions to check for integer divide by zero
-membedded-data	Use ROM instead of RAM
-mextern-sdata	Use -G for data that is not defined by the current object
-mflip-mips16	Switch on/off MIPS16 ASE on alternating functions for compiler testing
-mflush-func=FUNC	Use FUNC to flush the cache before calling stack trampolines
-mgpopt	Use GP-relative addressing to access small data
-minterlink-mips16	Generate code that can be safely linked with MIPS16 code.
-mips16	Generate MIPS16 code
-mit=TRANSPORT	Enable instrumented trace for MPLAB REAL ICE
-legacy-libc	Use legacy lib/include directory
-mllsc	Use ll, sc and sync instructions
-mlocal-sdata	Use -G for object-local data
-mlong-calls	Use indirect calls
-mmcount-ra-address	Pass the address of the ra save location to _mcount in $12
-mmemcpy	Don't optimize block moves
-mno-float	Prevent the use of all floating-point operations
-mno-flush-func	Do not use a cache-flushing function before calling stack trampolines
-mno-mips16	Generate normal-mode code
-mno-mpdebug-lib	Do not link MPLAB REAL ICE and MPLAB ICD 3 debug-support library
-mno-peripheral-libs	Do not link peripheral libraries
-mplt	When generating -mabicalls code, allow executables to use PLTs and copy relocations
-mprocessor=PROCESSOR	Specify the target PIC32 PROCESSOR
-mrelax-pic-calls	Try to allow the linker to turn PIC calls into direct calls
-mshared	When generating -mabicalls code, make the code suitable for use in shared libraries
-msmart-io	Enable smart-IO library call forwarding level 1
-msmart-io=LEVEL	Enable smart-IO library call forwarding level LEVEL [0,2]
-msplit-addresses	Optimize lui/addiu address loads
-mtext=NAME	Name the text section NAME (default .text)
-muninit-const-in-rodata	Put uninitialized constants in ROM (needs -membedded-data)

The following options control optimizations:

```
  -O<number>                  Set optimization level to <number>
  -Os                         Optimize for space rather than speed
  -falign-functions           Align the start of functions
  -falign-jumps               Align labels which are only reached by jumping
  -falign-labels              Align all labels
  -falign-loops               Align the start of loops
  -fargument-alias            Specify that arguments may alias each other and
                              globals
  -fargument-noalias          Assume arguments may alias globals but not each
                              other
  -fargument-noalias-anything Assume arguments alias no other storage
  -fargument-noalias-global   Assume arguments alias neither each other nor
                              globals
  -fasynchronous-unwind-tables Generate unwind tables that are exact at each
                              instruction boundary
  -fbranch-count-reg          Replace add, compare, branch with branch on count
                              register
  -fbranch-probabilities      Use profiling information for branch probabilities
  -fbranch-target-load-optimize Perform branch target load optimization before
                              prologue / epilogue threading
  -fbranch-target-load-optimize2 Perform branch target load optimization after
                              prologue / epilogue threading
  -fbtr-bb-exclusive          Restrict target load migration not to re-use
                              registers in any basic block
  -fcaller-saves              Save registers around function calls
  -fcommon                    Do not put uninitialized globals in the common
                              section
  -fconserve-stack            Do not perform optimizations increasing
                              noticeably stack usage
  -fcprop-registers           Perform a register copy-propagation optimization
                              pass
  -fcrossjumping              Perform cross-jumping optimization
  -fcse-follow-jumps          When running CSE, follow jumps to their targets
  -fcx-fortran-rules          Complex multiplication and division follow
                              Fortran rules
  -fcx-limited-range          Omit range reduction step when performing complex
                              division
  -fdata-sections             Place data items into their own section
  -fdce                       Use the RTL dead code elimination pass
  -fdefer-pop                 Defer popping functions args from stack until
                              later
  -fdelayed-branch            Attempt to fill delay slots of branch instructions
  -fdelete-null-pointer-checks Delete useless null pointer checks
  -fdse                       Use the RTL dead store elimination pass
  -fearly-inlining            Perform early inlining
  -fexceptions                Enable exception handling
  -fexpensive-optimizations   Perform a number of minor, expensive optimizations
  -ffinite-math-only          Assume no NaNs or infinities are generated
  -ffloat-store               Don't allocate floats and doubles in extended-
                              precision registers
  -fforward-propagate         Perform a forward propagation pass on RTL
  -fgcse                      Perform global common subexpression elimination
  -fgcse-after-reload         Perform global common subexpression elimination
                              after register allocation has finished
  -fgcse-las                  Perform redundant load after store elimination in
                              global common subexpression elimination
  -fgcse-lm                   Perform enhanced load motion during global common
                              subexpression elimination
  -fgcse-sm                   Perform store motion after global common
                              subexpression elimination
  -fgraphite-identity         Enable Graphite Identity transformation
  -fguess-branch-probability  Enable guessing of branch probabilities
```

```
-fif-conversion             Perform conversion of conditional jumps to
                            branchless equivalents
-fif-conversion2            Perform conversion of conditional jumps to
                            conditional execution
-finline-functions          Integrate simple functions into their callers
-finline-functions-called-once Integrate functions called once into their
                            callers
-finline-small-functions    Integrate simple functions into their callers
                            when code size is known to not growth
-fipa-cp                    Perform Interprocedural constant propagation
-fipa-cp-clone              Perform cloning to make Interprocedural constant
                            propagation stronger
-fipa-matrix-reorg          Perform matrix layout flattening and transposing
                            based on profiling information.
-fipa-pta                   Perform interprocedural points-to analysis
-fipa-pure-const            Discover pure and const functions
-fipa-reference             Discover readonly and non addressable static
                            variables
-fipa-sra                   Perform interprocedural reduction of aggregates
-fipa-type-escape           Type based escape and alias analysis
-fivopts                    Optimize induction variables on trees
-fjump-tables               Use jump tables for sufficiently large switch
                            statements
-floop-block                Enable Loop Blocking transformation
-floop-interchange          Enable Loop Interchange transformation
-floop-parallelize-all      Mark all loops as parallel
-floop-strip-mine           Enable Loop Strip Mining transformation
-flto-report                Report various link-time optimization statistics
-fltrans                    Run the link-time optimizer in local
                            transformation (LTRANS) mode.
-fmath-errno                Set errno after built-in math functions
-fmerge-all-constants       Attempt to merge identical constants and constant
                            variables
-fmerge-constants           Attempt to merge identical constants across
                            compilation units
-fmodulo-sched              Perform SMS based modulo scheduling before the
                            first scheduling pass
-fmove-loop-invariants      Move loop invariant computations out of loops
-fnon-call-exceptions       Support synchronous non-call exceptions
-fomit-frame-pointer        When possible do not generate stack frames
-foptimize-register-move    Do the full register move optimization pass
-foptimize-sibling-calls    Optimize sibling and tail recursive calls
-fpack-struct               Pack structure members together without holes
-fpack-struct=<number>      Set initial maximum structure member alignment
-fpeel-loops                Perform loop peeling
-fpeephole                  Enable machine specific peephole optimizations
-fpeephole2                 Enable an RTL peephole pass before sched2
-fpredictive-commoning      Run predictive commoning optimization.
-fprefetch-loop-arrays      Generate prefetch instructions, if available, for
                            arrays in loops
-fpromote-loop-indices      Promote loop indices to word-sized indices when
                            safe
-freg-struct-return         Return small aggregates in registers
-fregmove                   Enables a register move optimization
-fremove-local-statics      Convert function-local static variables to
                            automatic variables when it is safe to do so
-frename-registers          Perform a register renaming optimization pass
-freorder-blocks            Reorder basic blocks to improve code placement
-freorder-blocks-and-partition Reorder basic blocks and partition into hot
                            and cold sections
-freorder-functions         Reorder functions to improve code placement
-frerun-cse-after-loop      Add a common subexpression elimination pass after
                            loop optimizations
```

```
-freschedule-modulo-scheduled-loops Enable/Disable the traditional scheduling
                            in loops that already passed modulo scheduling
-frounding-math             Disable optimizations that assume default FP
                            rounding behavior
-frtti                      Generate run time type descriptor information
-fsched-critical-path-heuristic Enable the critical path heuristic in the
                            scheduler
-fsched-dep-count-heuristic Enable the dependent count heuristic in the
                            scheduler
-fsched-group-heuristic     Enable the group heuristic in the scheduler
-fsched-interblock          Enable scheduling across basic blocks
-fsched-last-insn-heuristic Enable the last instruction heuristic in the
                            scheduler
-fsched-pressure            Enable register pressure sensitive insn scheduling
-fsched-rank-heuristic      Enable the rank heuristic in the scheduler
-fsched-spec                Allow speculative motion of non-loads
-fsched-spec-insn-heuristic Enable the speculative instruction heuristic in
                            the scheduler
-fsched-spec-load           Allow speculative motion of some loads
-fsched-spec-load-dangerous Allow speculative motion of more loads
-fsched-stalled-insns       Allow premature scheduling of queued insns
-fsched-stalled-insns-dep   Set dependence distance checking in premature
                            scheduling of queued insns
-fsched2-use-superblocks    If scheduling post reload, do superblock
                            scheduling
-fschedule-insns            Reschedule instructions before register allocation
-fschedule-insns2           Reschedule instructions after register allocation
-fsection-anchors           Access data in the same section from shared
                            anchor points
-fsel-sched-pipelining      Perform software pipelining of inner loops during
                            selective scheduling
-fsel-sched-pipelining-outer-loops Perform software pipelining of outer loops
                            during selective scheduling
-fsel-sched-reschedule-pipelined Reschedule pipelined regions without
                            pipelining
-fselective-scheduling      Schedule instructions using selective scheduling
                            algorithm
-fselective-scheduling2     Run selective scheduling after reload
-fshort-double              Use the same size for double as for float
-fshort-enums               Use the narrowest integer type possible for
                            enumeration types
-fshort-wchar               Force the underlying type for "wchar_t" to be
                            "unsigned short"
-fsignaling-nans            Disable optimizations observable by IEEE
                            signaling NaNs
-fsigned-zeros              Disable floating point optimizations that ignore
                            the IEEE signedness of zero
-fsingle-precision-constant Convert floating point constants to single
                            precision constants
-fsplit-ivs-in-unroller     Split lifetimes of induction variables when loops
                            are unrolled
-fsplit-wide-types          Split wide types into independent registers
-fstrict-aliasing           Assume strict aliasing rules apply
-fthread-jumps              Perform jump threading optimizations
-fno-threadsafe-statics     Do not generate thread-safe code for initializing
                            local statics
-ftoplevel-reorder          Reorder top level functions, variables, and asms
-ftrapping-math             Assume floating-point operations can trap
-ftrapv                     Trap for signed overflow in addition, subtraction
                            and multiplication
-ftree-builtin-call-dce     Enable conditional dead code elimination for
                            builtin calls
-ftree-ccp                  Enable SSA-CCP optimization on trees
```

```
-ftree-ch                     Enable loop header copying on trees
-ftree-copy-prop              Enable copy propagation on trees
-ftree-copyrename             Replace SSA temporaries with better names in
                              copies
-ftree-cselim                 Transform condition stores into unconditional ones
-ftree-dce                    Enable SSA dead code elimination optimization on
                              trees
-ftree-dominator-opts         Enable dominator optimizations
-ftree-dse                    Enable dead store elimination
-ftree-forwprop               Enable forward propagation on trees
-ftree-fre                    Enable Full Redundancy Elimination (FRE) on trees
-ftree-loop-distribution      Enable loop distribution on trees
-ftree-loop-im                Enable loop invariant motion on trees
-ftree-loop-ivcanon           Create canonical induction variables in loops
-ftree-loop-linear            Enable linear loop transforms on trees
-ftree-loop-optimize          Enable loop optimizations on tree level
-ftree-lrs                    Perform live range splitting during the SSA-
                              >normal pass
-ftree-phiprop                Enable hoisting loads from conditional pointers.
-ftree-pre                    Enable SSA-PRE optimization on trees
-ftree-pre-partial-partial    In SSA-PRE optimization on trees, enable partial-
                              partial redundancy elimination.
-ftree-pre-partial-partial-obliviously In SSA-PRE optimization on trees,
                              enable partial-partial redundancy elimination
                              without regard for the cost of the inserted phi
                              nodes.
-ftree-pta                    Perform function-local points-to analysis on
                              trees.
-ftree-reassoc                Enable reassociation on tree level
-ftree-scev-cprop             Enable copy propagation of scalar-evolution
                              information.
-ftree-sink                   Enable SSA code sinking on trees
-ftree-slp-vectorize          Enable basic block vectorization (SLP) on trees
-ftree-sra                    Perform scalar replacement of aggregates
-ftree-switch-conversion      Perform conversions of switch initializations.
-ftree-ter                    Replace temporary expressions in the SSA->normal
                              pass
-ftree-vect-loop-version      Enable loop versioning when doing loop
                              vectorization on trees
-ftree-vectorize              Enable loop vectorization on trees
-ftree-vrp                    Perform Value Range Propagation on trees
-funit-at-a-time              Compile whole compilation unit at a time
-funroll-all-loops            Perform loop unrolling for all loops
-funroll-loops                Perform loop unrolling when iteration count is
                              known
-funsafe-loop-optimizations   Allow loop optimizations to assume that the loops
                              behave in normal way
-funsafe-math-optimizations   Allow math optimizations that may violate IEEE or
                              ISO standards
-funswitch-loops              Perform loop unswitching
-funwind-tables               Just generate unwind tables for exception handling
-fvar-tracking                Perform variable tracking
-fvar-tracking-assignments    Perform variable tracking by annotating
                              assignments
-fvar-tracking-assignments-toggle Toggle -fvar-tracking-assignments
-fvar-tracking-uninit         Perform variable tracking and also tag variables
                              that are uninitialized
-fvariable-expansion-in-unroller Apply variable expansion when loops are
                              unrolled
-fvect-cost-model             Enable use of cost model in vectorization
-fvpt                         Use expression value profiles in optimizations
-fweb                         Construct webs and split unrelated uses of single
                              variable
```

-fwhole-program	Perform whole program optimizations
-fwpa	Run the link-time optimizer in whole program analysis (WPA) mode.
-fwrapv	Assume signed arithmetic overflow wraps around

The following options control compiler warning messages:

```
  -W                          This switch is deprecated; use -Wextra instead
  -Wabi                       Warn about things that will change when compiling
                              with an ABI-compliant compiler
  -Waddress                   Warn about suspicious uses of memory addresses
  -Waggregate-return          Warn about returning structures, unions or arrays
  -Waliasing                  Warn about possible aliasing of dummy arguments
  -Walign-commons             Warn about alignment of COMMON blocks
  -Wall                       Enable most warning messages
  -Wampersand                 Warn about missing ampersand in continued
                              character constants
  -Warray-bounds              Warn if an array is accessed out of bounds
  -Warray-temporaries         Warn about creation of array temporaries
  -Wassign-intercept          Warn whenever an Objective-C assignment is being
                              intercepted by the garbage collector
  -Wattributes                Warn about inappropriate attribute usage
  -Wbad-function-cast         Warn about casting functions to incompatible types
  -Wbuiltin-macro-redefined   Warn when a built-in preprocessor macro is
                              undefined or redefined
  -Wc++-compat                Warn about C constructs that are not in the
                              common subset of C and C++
  -Wc++0x-compat              Warn about C++ constructs whose meaning differs
                              between ISO C++ 1998 and ISO C++ 200x
  -Wcast-align                Warn about pointer casts which increase alignment
  -Wcast-qual                 Warn about casts which discard qualifiers
  -Wchar-subscripts           Warn about subscripts whose type is "char"
  -Wcharacter-truncation      Warn about truncated character expressions
  -Wclobbered                 Warn about variables that might be changed by
                              "longjmp" or "vfork"
  -Wcomment                   Warn about possibly nested block comments, and
                              C++ comments spanning more than one physical line
  -Wcomments                  Synonym for -Wcomment
  -Wconversion                Warn for implicit type conversions that may
                              change a value
  -Wconversion-null           Warn for converting NULL from/to a non-pointer
                              type
  -Wcoverage-mismatch         Warn instead of error in case profiles in
                              -fprofile-use do not match
  -Wctor-dtor-privacy         Warn when all constructors and destructors are
                              private
  -Wdeclaration-after-statement Warn when a declaration is found after a
                              statement
  -Wdeprecated                Warn if a deprecated compiler feature, class,
                              method, or field is used
  -Wdeprecated-declarations   Warn about uses of __attribute__((deprecated))
                              declarations
  -Wdisabled-optimization     Warn when an optimization pass is disabled
  -Wdiv-by-zero               Warn about compile-time integer division by zero
  -Wdouble-promotion          Warn about implicit conversions from "float" to
                              "double"
  -Weffc++                    Warn about violations of Effective C++ style rules
  -Wempty-body                Warn about an empty body in an if or else
                              statement
  -Wendif-labels              Warn about stray tokens after #elif and #endif
  -Wenum-compare              Warn about comparison of different enum types
  -Werror-implicit-function-declaration This switch is deprecated; use
                              -Werror=implicit-function-declaration instead
  -Wextra                     Print extra (possibly unwanted) warnings
  -Wfloat-equal               Warn if testing floating point numbers for
                              equality
  -Wformat                    Warn about printf/scanf/strftime/strfmon format
                              string anomalies
  -Wformat-contains-nul       Warn about format strings that contain NUL bytes
```

```
-Wformat-extra-args         Warn if passing too many arguments to a function
                            for its format string
-Wformat-nonliteral         Warn about format strings that are not literals
-Wformat-security           Warn about possible security problems with format
                            functions
-Wformat-y2k                Warn about strftime formats yielding 2-digit years
-Wformat-zero-length        Warn about zero-length formats
-Wignored-qualifiers        Warn whenever type qualifiers are ignored.
-Wimplicit-function-declaration Warn about implicit function declarations
-Wimplicit-int              Warn when a declaration does not specify a type
-Wimplicit-interface        Warn about calls with implicit interface
-Wimplicit-procedure        Warn about called procedures not explicitly
                            declared
-Winit-self                 Warn about variables which are initialized to
                            themselves
-Winline                    Warn when an inlined function cannot be inlined
-Wint-to-pointer-cast       Warn when there is a cast to a pointer from an
                            integer of a different size
-Wintrinsic-shadow          Warn if a user-procedure has the same name as an
                            intrinsic
-Wintrinsics-std            Warn on intrinsics not part of the selected
                            standard
-Winvalid-offsetof          Warn about invalid uses of the "offsetof" macro
-Winvalid-pch               Warn about PCH files that are found but not used
-Wjump-misses-init          Warn when a jump misses a variable initialization
-Wlarger-than=<number>      Warn if an object is larger than <number> bytes
-Wline-truncation           Warn about truncated source lines
-Wlogical-op                Warn when a logical operator is suspiciously
                            always evaluating to true or false
-Wlong-long                 Do not warn about using "long long" when -pedantic
-Wmain                      Warn about suspicious declarations of "main"
-Wmissing-braces            Warn about possibly missing braces around
                            initializers
-Wmissing-declarations      Warn about global functions without previous
                            declarations
-Wmissing-field-initializers Warn about missing fields in struct initializers
-Wmissing-format-attribute  Warn about functions which might be candidates
                            for format attributes
-Wmissing-include-dirs      Warn about user-specified include directories
                            that do not exist
-Wmissing-noreturn          Warn about functions which might be candidates
                            for __attribute__((noreturn))
-Wmissing-parameter-type    Warn about function parameters declared without a
                            type specifier in K&R-style functions
-Wmissing-prototypes        Warn about global functions without prototypes
-Wmudflap                   Warn about constructs not instrumented by
                            -fmudflap
-Wmultichar                 Warn about use of multi-character character
                            constants
-Wnested-externs            Warn about "extern" declarations not at file scope
-Wnon-template-friend       Warn when non-templatized friend functions are
                            declared within a template
-Wnon-virtual-dtor          Warn about non-virtual destructors
-Wnonnull                   Warn about NULL being passed to argument slots
                            marked as requiring non-NULL
-Wnormalized=<id|nfc|nfkc>  Warn about non-normalised Unicode strings
-Wold-style-cast            Warn if a C-style cast is used in a program
-Wold-style-declaration     Warn for obsolescent usage in a declaration
-Wold-style-definition      Warn if an old-style parameter definition is used
-Woverflow                  Warn about overflow in arithmetic expressions
-Woverlength-strings        Warn if a string is longer than the maximum
                            portable length specified by the standard
-Woverloaded-virtual        Warn about overloaded virtual function names
```

```
-Woverride-init             Warn about overriding initializers without side
                            effects
-Wpacked                    Warn when the packed attribute has no effect on
                            struct layout
-Wpacked-bitfield-compat    Warn about packed bit-fields whose offset changed
                            in GCC 4.4
-Wpadded                    Warn when padding is required to align structure
                            members
-Wparentheses               Warn about possibly missing parentheses
-Wpmf-conversions           Warn when converting the type of pointers to
                            member functions
-Wpointer-arith             Warn about function pointer arithmetic
-Wpointer-sign              Warn when a pointer differs in signedness in an
                            assignment
-Wpointer-to-int-cast       Warn when a pointer is cast to an integer of a
                            different size
-Wpoison-system-directories Warn for -I and -L options using system
                            directories if cross compiling
-Wpragmas                   Warn about misuses of pragmas
-Wprotocol                  Warn if inherited methods are unimplemented
-Wredundant-decls           Warn about multiple declarations of the same
                            object
-Wreorder                   Warn when the compiler reorders code
-Wreturn-type               Warn whenever a function's return type defaults
                            to "int" (C), or about inconsistent return types
                            (C++)
-Wselector                  Warn if a selector has multiple methods
-Wsequence-point            Warn about possible violations of sequence point
                            rules
-Wshadow                    Warn when one local variable shadows another
-Wsign-compare              Warn about signed-unsigned comparisons
-Wsign-promo                Warn when overload promotes from unsigned to
                            signed
-Wstack-protector           Warn when not issuing stack smashing protection
                            for some reason
-Wstrict-aliasing           Warn about code which might break strict aliasing
                            rules
-Wstrict-aliasing=          Warn about code which might break strict aliasing
                            rules
-Wstrict-null-sentinel      Warn about uncasted NULL used as sentinel
-Wstrict-overflow           Warn about optimizations that assume that signed
                            overflow is undefined
-Wstrict-overflow=          Warn about optimizations that assume that signed
                            overflow is undefined
-Wstrict-prototypes         Warn about unprototyped function declarations
-Wstrict-selector-match     Warn if type signatures of candidate methods do
                            not match exactly
-Wsurprising                Warn about "suspicious" constructs
-Wswitch                    Warn about enumerated switches, with no default,
                            missing a case
-Wswitch-default            Warn about enumerated switches missing a
                            "default:" statement
-Wswitch-enum               Warn about all enumerated switches missing a
                            specific case
-Wsync-nand                 Warn when __sync_fetch_and_nand and
                            __sync_nand_and_fetch built-in functions are used
-Wsynth                     Deprecated.  This switch has no effect
-Wsystem-headers            Do not suppress warnings from system headers
-Wtabs                      Permit nonconforming uses of the tab character
-Wtraditional               Warn about features not present in traditional C
-Wtraditional-conversion    Warn of prototypes causing type conversions
                            different from what would happen in the absence
                            of prototype
```

```
-Wtrigraphs                  Warn if trigraphs are encountered that might
                             affect the meaning of the program
-Wtype-limits                Warn if a comparison is always true or always
                             false due to the limited range of the data type
-Wundeclared-selector        Warn about @selector()s without previously
                             declared methods
-Wundef                      Warn if an undefined macro is used in an #if
                             directive
-Wunderflow                  Warn about underflow of numerical constant
                             expressions
-Wuninitialized              Warn about uninitialized automatic variables
-Wunknown-pragmas            Warn about unrecognized pragmas
-Wunsafe-loop-optimizations Warn if the loop cannot be optimized due to
                             nontrivial assumptions.
-Wunsuffixed-float-constants Warn about unsuffixed float constants
-Wunused                     Enable all -Wunused- warnings
-Wunused-function            Warn when a function is unused
-Wunused-label               Warn when a label is unused
-Wunused-macros              Warn about macros defined in the main file that
                             are not used
-Wunused-parameter           Warn when a function parameter is unused
-Wunused-result              Warn if a caller of a function, marked with
                             attribute warn_unused_result, does not use its
                             return value
-Wunused-value               Warn when an expression value is unused
-Wunused-variable            Warn when a variable is unused
-Wvariadic-macros            Do not warn about using variadic macros when
                             -pedantic
-Wvla                        Warn if a variable length array is used
-Wvolatile-register-var      Warn when a register variable is declared volatile
-Wwrite-strings              In C++, nonzero means warn about deprecated
                             conversion from string literals to `char *'.  In
                             C, similar warning, except that the conversion is
                             of course not deprecated by the ISO C standard.
```

The --param option recognizes the following as parameters:

struct-reorg-cold-struct-ratio	The threshold ratio between current and hottest structure counts
predictable-branch-outcome	Maximal estimated outcome of branch considered predictable
max-inline-insns-single	The maximum number of instructions in a single function eligible for inlining
max-inline-insns-auto	The maximum number of instructions when automatically inlining
max-inline-insns-recursive	The maximum number of instructions inline function can grow to via recursive inlining
max-inline-insns-recursive-auto	The maximum number of instructions non-inline function can grow to via recursive inlining
max-inline-recursive-depth	The maximum depth of recursive inlining for inline functions
max-inline-recursive-depth-auto	The maximum depth of recursive inlining for non-inline functions
min-inline-recursive-probability	Inline recursively only when the probability of call being executed exceeds the parameter
max-early-inliner-iterations	The maximum number of nested indirect inlining performed by early inliner
max-variable-expansions-in-unroller	If -fvariable-expansion-in-unroller is used, the maximum number of times that an individual variable will be expanded during loop unrolling
min-vect-loop-bound	If -ftree-vectorize is used, the minimal loop bound of a loop to be considered for vectorization
max-delay-slot-insn-search	The maximum number of instructions to consider to fill a delay slot
max-delay-slot-live-search	The maximum number of instructions to consider to find accurate live register information
max-pending-list-length	The maximum length of scheduling's pending operations list
large-function-insns	The size of function body to be considered large
large-function-growth	Maximal growth due to inlining of large function (in percent)
large-unit-insns	The size of translation unit to be considered large
inline-unit-growth	How much can given compilation unit grow because of the inlining (in percent)
ipcp-unit-growth	How much can given compilation unit grow because of the interprocedural constant propagation (in percent)
early-inlining-insns	Maximal estimated growth of function body caused by early inlining of single call
large-stack-frame	The size of stack frame to be considered large
large-stack-frame-growth	Maximal stack frame growth due to inlining (in percent)
max-gcse-memory	The maximum amount of memory to be allocated by GCSE
gcse-after-reload-partial-fraction	The threshold ratio for performing partial redundancy elimination after reload
gcse-after-reload-critical-fraction	The threshold ratio of critical edges execution count that permit performing redundancy elimination after reload
gcse-cost-distance-ratio	Scaling factor in calculation of maximum distance an expression can be moved by GCSE optimizations
gcse-unrestricted-cost	Cost at which GCSE optimizations will not constraint the distance an expression can travel
max-hoist-depth	Maximum depth of search in the dominator tree for expressions to hoist
max-unrolled-insns	The maximum number of instructions to consider to unroll in a loop

max-average-unrolled-insns The maximum number of instructions to consider to unroll in a loop on average
max-unroll-times The maximum number of unrollings of a single loop
max-peeled-insns The maximum number of insns of a peeled loop
max-peel-times The maximum number of peelings of a single loop
max-completely-peeled-insns The maximum number of insns of a completely peeled loop
max-completely-peel-times The maximum number of peelings of a single loop that is peeled completely
max-once-peeled-insns The maximum number of insns of a peeled loop that rolls only once
max-completely-peel-loop-nest-depth The maximum depth of a loop nest we completely peel
max-unswitch-insns The maximum number of insns of an unswitched loop
max-unswitch-level The maximum number of unswitchings in a single loop
max-iterations-to-track Bound on the number of iterations the brute force # of iterations analysis algorithm evaluates
max-iterations-computation-cost Bound on the cost of an expression to compute the number of iterations
sms-max-ii-factor A factor for tuning the upper bound that swing modulo scheduler uses for scheduling a loop
sms-dfa-history The number of cycles the swing modulo scheduler considers when checking conflicts using DFA
sms-loop-average-count-threshold A threshold on the average loop count considered by the swing modulo scheduler
hot-bb-count-fraction Select fraction of the maximal count of repetitions of basic block in program given basic block needs to have to be considered hot
hot-bb-frequency-fraction Select fraction of the maximal frequency of executions of basic block in function given basic block needs to have to be considered hot
align-threshold Select fraction of the maximal frequency of executions of basic block in function given basic block get alignment
align-loop-iterations Loops iterating at least selected number of iterations will get loop alignement.
max-predicted-iterations The maximum number of loop iterations we predict statically
tracer-dynamic-coverage-feedback The percentage of function, weighted by execution frequency, that must be covered by trace formation. Used when profile feedback is available
tracer-dynamic-coverage The percentage of function, weighted by execution frequency, that must be covered by trace formation. Used when profile feedback is not available
tracer-max-code-growth Maximal code growth caused by tail duplication (in percent)
tracer-min-branch-ratio Stop reverse growth if the reverse probability of best edge is less than this threshold (in percent)
tracer-min-branch-probability-feedback Stop forward growth if the probability of best edge is less than this threshold (in percent). Used when profile feedback is available
tracer-min-branch-probability Stop forward growth if the probability of best edge is less than this threshold (in percent). Used when profile feedback is not available
max-crossjump-edges The maximum number of incoming edges to consider for crossjumping
min-crossjump-insns The minimum number of matching instructions to consider for crossjumping
max-grow-copy-bb-insns The maximum expansion factor when copying basic blocks

```
max-goto-duplication-insns  The maximum number of insns to duplicate when
                            unfactoring computed gotos
max-cse-path-length         The maximum length of path considered in cse
max-cse-insns               The maximum instructions CSE process before
                            flushing
lim-expensive               The minimum cost of an expensive expression in
                            the loop invariant motion
iv-consider-all-candidates-bound Bound on number of candidates below that all
                            candidates are considered in iv optimizations
iv-max-considered-uses      Bound on number of iv uses in loop optimized in
                            iv optimizations
iv-always-prune-cand-set-bound If number of candidates in the set is smaller,
                            we always try to remove unused ivs during its
                            optimization
scev-max-expr-size          Bound on size of expressions used in the scalar
                            evolutions analyzer
omega-max-vars              Bound on the number of variables in Omega
                            constraint systems
omega-max-geqs              Bound on the number of inequalities in Omega
                            constraint systems
omega-max-eqs               Bound on the number of equalities in Omega
                            constraint systems
omega-max-wild-cards        Bound on the number of wild cards in Omega
                            constraint systems
omega-hash-table-size       Bound on the size of the hash table in Omega
                            constraint systems
omega-max-keys              Bound on the number of keys in Omega constraint
                            systems
omega-eliminate-redundant-constraints When set to 1, use expensive methods to
                            eliminate all redundant constraints
vect-max-version-for-alignment-checks Bound on number of runtime checks
                            inserted by the vectorizer's loop versioning for
                            alignment check
vect-max-version-for-alias-checks Bound on number of runtime checks inserted
                            by the vectorizer's loop versioning for alias
                            check
max-cselib-memory-locations The maximum memory locations recorded by cselib
ggc-min-expand              Minimum heap expansion to trigger garbage
                            collection, as a percentage of the total size of
                            the heap
ggc-min-heapsize            Minimum heap size before we start collecting
                            garbage, in kilobytes
max-reload-search-insns     The maximum number of instructions to search
                            backward when looking for equivalent reload
max-sched-region-blocks     The maximum number of blocks in a region to be
                            considered for interblock scheduling
max-sched-region-insns      The maximum number of insns in a region to be
                            considered for interblock scheduling
max-pipeline-region-blocks  The maximum number of blocks in a region to be
                            considered for interblock scheduling
max-pipeline-region-insns   The maximum number of insns in a region to be
                            considered for interblock scheduling
min-spec-prob               The minimum probability of reaching a source
                            block for interblock speculative scheduling
max-sched-extend-regions-iters The maximum number of iterations through CFG
                            to extend regions
max-sched-insn-conflict-delay The maximum conflict delay for an insn to be
                            considered for speculative motion
sched-spec-prob-cutoff      The minimal probability of speculation success
                            (in percents), so that speculative insn will be
                            scheduled.
selsched-max-lookahead      The maximum size of the lookahead window of
                            selective scheduling
```

```
selsched-max-sched-times    Maximum number of times that an insn could be
                            scheduled
selsched-insns-to-rename    Maximum number of instructions in the ready list
                            that are considered eligible for renaming
sched-mem-true-dep-cost     Minimal distance between possibly conflicting
                            store and load
max-last-value-rtl          The maximum number of RTL nodes that can be
                            recorded as combiner's last value
integer-share-limit         The upper bound for sharing integer constants
min-virtual-mappings        Minimum number of virtual mappings to consider
                            switching to full virtual renames
virtual-mappings-ratio      Ratio between virtual mappings and virtual
                            symbols to do full virtual renames
ssp-buffer-size             The lower bound for a buffer to be considered for
                            stack smashing protection
max-jump-thread-duplication-stmts Maximum number of statements allowed in a
                            block that needs to be duplicated when threading
                            jumps
max-fields-for-field-sensitive Maximum number of fields in a structure before
                            pointer analysis treats the structure as a single
                            variable
max-sched-ready-insns       The maximum number of instructions ready to be
                            issued to be considered by the scheduler during
                            the first scheduling pass
prefetch-latency            The number of insns executed before prefetch is
                            completed
simultaneous-prefetches     The number of prefetches that can run at the same
                            time
l1-cache-size               The size of L1 cache
l1-cache-line-size          The size of L1 cache line
l2-cache-size               The size of L2 cache
use-canonical-types         Whether to use canonical types
max-partial-antic-length    Maximum length of partial antic set when
                            performing tree pre optimization
sccvn-max-scc-size          Maximum size of a SCC before SCCVN stops
                            processing a function
ira-max-loops-num           Max loops number for regional RA
ira-max-conflict-table-size Max size of conflict table in MB
ira-loop-reserved-regs      The number of registers in each class kept unused
                            by loop invariant motion
switch-conversion-max-branch-ratio The maximum ratio between array size and
                            switch branches for a switch conversion to take
                            place
loop-block-tile-size        size of tiles for loop blocking
graphite-max-nb-scop-params maximum number of parameters in a SCoP
graphite-max-bbs-per-function maximum number of basic blocks per function to
                            be analyzed by Graphite
loop-invariant-max-bbs-in-loop Max basic blocks number in loop for loop
                            invariant motion
slp-max-insns-in-bb         Maximum number of instructions in basic block to
                            be considered for SLP vectorization
min-insn-to-prefetch-ratio  Min. ratio of insns to prefetches to enable
                            prefetching for a loop with an unknown trip count
prefetch-min-insn-to-mem-ratio Min. ratio of insns to mem ops to enable
                            prefetching in a loop
max-vartrack-size           Max. size of var tracking hash tables
min-nondebug-insn-uid       The minimum UID to be used for a nondebug insn
ipa-sra-ptr-growth-factor   Maximum allowed growth of size of new parameters
                            ipa-sra replaces a pointer to an aggregate with
```