NAME

AS - the portable GNU assembler.

SYNOPSIS

as [**-a**[**cdghlns**][=*file*]] [**--alternate**] [**-D**] [**--compress-debug-sections**] [**--nocompress-debug-sections**] [**--debug-prefix-map** *old=new*] [**--defsym** *sym=val*] [**-f**] [**-g**] [**--gstabs**] [**--gstabs+**] [**--gdwarf-2**] [**--gdwarf-sections**] [**--help**] [**-I** *dir*] [**-J**] [**-K**] [**-L**] [**--listing-lhs-width**=*NUM*] [**--listing-lhs-width2**=*NUM*] [**--listing-rhs-width**=*NUM*] [**--listing-cont-lines**=*NUM*] [**--keep-locals**] [**-o** *objfile*] [**-R**] [**--reduce-memory-overheads**] [**--statistics**] [**-v**] [**-version**] [**--version**] [**-W**] [**--warn**] [**--fatal-warnings**] [**-w**] [**-x**] [**-Z**] [**@***FILE*] [**--size-check=[error|warning]**] [**--target-help**] [*target-options*] [**--**|*files* ...]

Target AArch64 options: [**-EB**|**-EL**] [**-mabi**=*ABI*]

Target Alpha options: [**-m***cpu*] [**-mdebug** | **-no-mdebug**] [**-replace** | **-noreplace**] [**-relax**] [**-g**] [**-G***size*] [**-F**] [**-32addr**]

Target ARC options: [**-marc[5|6|7|8]**] [**-EB**|**-EL**]

Target ARM options: [**-mcpu**=*processor*[*+extension...*]] [**-march**=*architecture*[*+extension...*]] [**-mfpu**=*floating-point-format*] [**-mfloat-abi**=*abi*] [**-meabi**=*ver*] [**-mthumb**] [**-EB**|**-EL**] [**-mapcs-32**|**-mapcs-26**|**-mapcs-float**| **-mapcs-reentrant**] [**-mthumb-interwork**] [**-k**]

Target Blackfin options: [**-mcpu**=*processor*[*-sirevision*]] [**-mfdpic**] [**-mno-fdpic**] [**-mnopic**]

Target CRIS options: [**--underscore** | **--no-underscore**] [**--pic**] [**-N**] [**--emulation=criself** | **--emulation=crisaout**] [**--march=v0_v10** | **--march=v10** | **--march=v32** | **--march=common_v10_v32**]

Target D10V options: [**-O**]

Target D30V options: [**-O**|**-n**|**-N**]

Target EPIPHANY options: [**-mepiphany**|**-mepiphany16**]

Target H8/300 options: [-h-tick-hex]

Target i386 options: [**--32**|**--x32**|**--64**] [**-n**] [**-march**=*CPU*[*+EXTENSION...*]] [**-mtune**=*CPU*]

Target i960 options: [**-ACA**|**-ACA_A**|**-ACB**|**-ACC**|**-AKA**|**-AKB**| **-AKC**|**-AMC**] [**-b**] [**-no-relax**]

Target IA-64 options: [**-mconstant-gp**|**-mauto-pic**] [**-milp32**|**-milp64**|**-mlp64**|**-mp64**] [**-mle**|**mbe**] [**-mtune=itanium1**|**-mtune=itanium2**] [**-munwind-check=warning**|**-munwind-check=error**] [**-mhint.b=ok**|**-mhint.b=warning**|**-mhint.b=error**] [**-x**|**-xexplicit**] [**-xauto**] [**-xdebug**]

Target IP2K options: [**-mip2022**|**-mip2022ext**]

Target M32C options: [**-m32c**|**-m16c**] [-relax] [-h-tick-hex]

Target M32R options: [**--m32rx**|**--[no-]warn-explicit-parallel-conflicts**| **--W[n]p**]

Target M680X0 options: [**-l**] [**-m68000**|**-m68010**|**-m68020**|...]

Target M68HC11 options: [**-m68hc11**|**-m68hc12**|**-m68hcs12**|**-mm9s12x**|**-mm9s12xg**] [**-mshort**|**-mlong**] [**-mshort-double**|**-mlong-double**] [**--force-long-branches**] [**--short-branches**] [**--strict-direct-mode**] [**--print-insn-syntax**] [**--print-opcodes**] [**--generate-example**]

Target MCORE options: [**-jsri2bsr**] [**-sifilter**] [**-relax**] [**-mcpu=[210|340]**]

Target Meta options: [**-mcpu=***cpu*] [**-mfpu=***cpu*] [**-mdsp=***cpu*] *Target MICROBLAZE options:*

Target MIPS options: [**-nocpp**] [**-EL**] [**-EB**] [**-O**[*optimization level*]] [**-g**[*debug level*]] [**-G** *num*] [**-KPIC**] [**-call_shared**] [**-non_shared**] [**-xgot** [**-mvxworks-pic**] [**-mabi**=*ABI*] [**-32**] [**-n32**] [**-64**] [**-mfp32**] [**-mgp32**] [**-mfp64**] [**-mgp64**] [**-mfpxx**] [**-modd-spreg**] [**-mno-odd-spreg**] [**-march**=*CPU*] [**-mtune**=*CPU*] [**-mips1**] [**-mips2**] [**-mips3**] [**-mips4**] [**-mips5**] [**-mips32**] [**-mips32r2**] [**-mips32r3**] [**-mips32r5**] [**-mips32r6**] [**-mips64**] [**-mips64r2**] [**-mips64r3**] [**-mips64r5**] [**-mips64r6**] [**-construct-floats**] [**-no-construct-floats**] [**-mnan=***encoding*] [**-trap**] [**-no-break**] [**-break**] [**-no-trap**] [**-mips16**] [**-no-mips16**] [**-mmicromips**] [**-mno-micromips**] [**-msmartmips**] [**-mno-smartmips**] [**-mips3d**] [**-no-mips3d**] [**-mdmx**] [**-no-mdmx**] [**-mdsp**] [**-mno-dsp**] [**-mdspr2**] [**-mno-dspr2**] [**-mmsa**] [**-mno-msa**] [**-mxpa**] [**-mno-xpa**] [**-mmt**] [**-mno-mt**] [**-mmcu**] [**-mno-mcu**] [**-minsn32**] [**-mno-insn32**] [**-mfix7000**] [**-mno-fix7000**] [**-mfix-rm7000**]

[**-mno-fix-rm7000**] [**-mfix-vr4120**] [**-mno-fix-vr4120**] [**-mfix-vr4130**] [**-mno-fix-vr4130**] [**-mdebug**] [**-no-mdebug**] [**-mpdr**] [**-mno-pdr**]

Target MMIX options: [**--fixed-special-register-names**] [**--globalize-symbols**] [**--gnu-syntax**] [**--relax**] [**--no-predefined-symbols**] [**--no-expand**] [**--no-merge-gregs**] [**-x**] [**--linker-allocated-gregs**]

Target Nios II options: [**-relax-all**] [**-relax-section**] [**-no-relax**] [**-EB**] [**-EL**]

Target NDS32 options: [**-EL**] [**-EB**] [**-O**] [**-Os**] [**-mcpu=***cpu*] [**-misa=***isa*] [**-mabi=***abi*] [**-mall-ext**] [**-m[no-]16-bit**] [**-m[no-]perf-ext**] [**-m[no-]perf2-ext**] [**-m[no-]string-ext**] [**-m[no-]dsp-ext**] [**-m[no-]mac**] [**-m[no-]div**] [**-m[no-]audio-isa-ext**] [**-m[no-]fpu-sp-ext**] [**-m[no-]fpu-dp-ext**] [**-m[no-]fpu-fma**] [**-mfpu-freg=***FREG*] [**-mreduced-regs**] [**-mfull-regs**] [**-m[no-]dx-regs**] [**-mpic**] [**-mno-relax**] [**-mb2bb**]

Target PDP11 options: [**-mpic**|**-mno-pic**] [**-mall**] [**-mno-extensions**] [**-m***extension*|**-mno-***extension*] [**-m***cpu*] [**-m***machine*]

Target picoJava options: [**-mb**|**-me**]

Target PowerPC options: [**-a32**|**-a64**] [**-mpwrx**|**-mpwr2**|**-mpwr**|**-m601**|**-mppc**|**-mppc32**|**-m603**|**-m604**|**-m403**|**-m405**| **-m440**|**-m464**|**-m476**|**-m7400**|**-m7410**|**-m7450**|**-m7455**|**-m750cl**|**-mppc64**| **-m620**|**-me500**|**-e500x2**|**-me500mc**|**-me500mc64**|**-me5500**|**-me6500**|**-mppc64bridge**| **-mbooke**|**-mpower4**|**-mpwr4**|**-mpower5**|**-mpwr5**|**-mpwr5x**|**-mpower6**|**-mpwr6**| **-mpower7**|**-mpwr7**|**-mpower8**|**-mpwr8**|**-ma2**|**-mcell**|**-mspe**|**-mtitan**|**-me300**|**-mcom**] [**-many**] [**-maltivec**|**-mvsx**|**-mhtm**|**-mvle**] [**-mregnames**|**-mno-regnames**] [**-mrelocatable**|**-mrelocatable-lib**|**-KPIC**] [**-memb**] [**-mlittle**|**-mlittle-endian**|**-le**|**-mbig**|**-mbig-endian**|**-be**] [**-msolaris**|**-mno-solaris**] [**-nops=***count*]

Target RL78 options: [**-mg10**] [**-m32bit-doubles**|**-m64bit-doubles**]

Target RX options: [**-mlittle-endian**|**-mbig-endian**] [**-m32bit-doubles**|**-m64bit-doubles**] [**-muse-conventional-section-names**] [**-msmall-data-limit**] [**-mpid**] [**-mrelax**] [**-mint-register=***number*] [**-mgcc-abi**|**-mrx-abi**]

Target s390 options: [**-m31**|**-m64**] [**-mesa**|**-mzarch**] [**-march=***CPU*] [**-mregnames**|**-mno-regnames**] [**-mwarn-areg-zero**]

Target SCORE options: [**-EB**][**-EL**][**-FIXDD**][**-NWARN**] [**-SCORE5**][**-SCORE5U**][**-SCORE7**][**-SCORE3**] [**-march=score7**][**-march=score3**] [**-USE_R1**][**-KPIC**][**-O0**][**-G** *num*][**-V**]

Target SPARC options: [**-Av6**|**-Av7**|**-Av8**|**-Asparclet**|**-Asparclite** **-Av8plus**|**-Av8plusa**|**-Av9**|**-Av9a**] [**-xarch=v8plus**|**-xarch=v8plusa**] [**-bump**] [**-32**|**-64**]

Target TIC54X options: [**-mcpu=54[123589]**|**-mcpu=54[56]lp**] [**-mfar-mode**|**-mf**] [**-merrors-to-file** <*filename*>|**-me** <*filename*>]

Target TIC6X options: [**-march=***arch*] [**-mbig-endian**|**-mlittle-endian**] [**-mdsbt**|**-mno-dsbt**] [**-mpid=no**|**-mpid=near**|**-mpid=far**] [**-mpic**|**-mno-pic**]

Target TILE-Gx options: [**-m32**|**-m64**][**-EB**][**-EL**]

Target Xtensa options: [**--[no-]text-section-literals**] [**--[no-]absolute-literals**] [**--[no-]target-align**] [**--[no-]longcalls**] [**--[no-]transform**] [**--rename-section** *oldname=newname*] [**--[no-]trampolines**]

Target Z80 options: [**-z80**] [**-r800**] [**-ignore-undocumented-instructions**] [**-Wnud**] [**-ignore-unportable-instructions**] [**-Wnup**] [**-warn-undocumented-instructions**] [**-Wud**] [**-warn-unportable-instructions**] [**-Wup**] [**-forbid-undocumented-instructions**] [**-Fud**] [**-forbid-unportable-instructions**] [**-Fup**]

DESCRIPTION

GNU **as** is really a family of assemblers. If you use (or have used) the GNU assembler on one architecture, you should find a fairly similar environment when you use it on another architecture. Each version has much in common with the others, including object file formats, most assembler directives (often called *pseudo-ops*) and assembler syntax.

as is primarily intended to assemble the output of the GNU C compiler `gcc` for use by the linker `ld`. Nevertheless, we've tried to make **as** assemble correctly everything that other assemblers for the same machine would assemble. Any exceptions are documented explicitly. This doesn't mean **as** always uses the same syntax as another assembler for the same architecture; for example, we know of several incompatible versions of 680x0 assembly language syntax.

Each time you run **as** it assembles exactly one source program. The source program is made up of one or more files. (The standard input is also a file.)

You give **as** a command line that has zero or more input file names. The input files are read (from left file name to right). A command line argument (in any position) that has no special meaning is taken to be an input file name.

If you give **as** no file names it attempts to read one input file from the **as** standard input, which is normally your terminal. You may have to type **ctl-D** to tell **as** there is no more program to assemble.

Use **--** if you need to explicitly name the standard input file in your command line.

If the source is empty, **as** produces a small, empty object file.

as may write warnings and error messages to the standard error file (usually your terminal). This should not happen when a compiler runs **as** automatically. Warnings report an assumption made so that **as** could keep assembling a flawed program; errors report a grave problem that stops the assembly.

If you are invoking **as** via the GNU C compiler, you can use the **-Wa** option to pass arguments through to the assembler. The assembler arguments must be separated from each other (and the **-Wa**) by commas. For example:

```
gcc -c -g -O -Wa,-alh,-L file.c
```

This passes two options to the assembler: **-alh** (emit a listing to standard output with high-level and assembly source) and **-L** (retain local symbols in the symbol table).

Usually you do not need to use this **-Wa** mechanism, since many compiler command-line options are automatically passed to the assembler by the compiler. (You can call the GNU compiler driver with the **-v** option to see precisely what options it passes to each compilation pass, including the assembler.)

OPTIONS

@file

Read command-line options from *file*. The options read are inserted in place of the original *@file* option. If *file* does not exist, or cannot be read, then the option will be treated literally, and not removed.

Options in *file* are separated by whitespace. A whitespace character may be included in an option by surrounding the entire option in either single or double quotes. Any character (including a backslash) may be included by prefixing the character to be included with a backslash. The *file* may itself contain additional *@file* options; any such options will be processed recursively.

-a[cdghlmns]

Turn on listings, in any of a variety of ways:

-ac omit false conditionals

-ad omit debugging directives

-ag include general information, like as version and options passed

-ah include high-level source

-al include assembly

-am
include macro expansions

-an omit forms processing

-as include symbols

=file
set the name of the listing file

You may combine these options; for example, use **-aln** for assembly listing without forms processing. The **=file** option, if used, must be the last one. By itself, **-a** defaults to **-ahls**.

--alternate
Begin in alternate macro mode.

--compress-debug-sections
Compress DWARF debug sections using zlib. The debug sections are renamed to begin with **.zdebug**, and the resulting object file may not be compatible with older linkers and object file utilities.

--nocompress-debug-sections
Do not compress DWARF debug sections. This is the default.

-D Ignored. This option is accepted for script compatibility with calls to other assemblers.

--debug-prefix-map *old=new*
When assembling files in directory *old*, record debugging information describing them as in *new* instead.

--defsym *sym=value*
Define the symbol *sym* to be *value* before assembling the input file. *value* must be an integer constant. As in C, a leading **0x** indicates a hexadecimal value, and a leading **0** indicates an octal value. The value of the symbol can be overridden inside a source file via the use of a `.set` pseudo-op.

-f ''fast''---skip whitespace and comment preprocessing (assume source is compiler output).

-g
--gen-debug
Generate debugging information for each assembler source line using whichever debug format is preferred by the target. This currently means either STABS, ECOFF or DWARF2.

--gstabs
Generate stabs debugging information for each assembler line. This may help debugging assembler code, if the debugger can handle it.

--gstabs+
Generate stabs debugging information for each assembler line, with GNU extensions that probably only gdb can handle, and that could make other debuggers crash or refuse to read your program. This may help debugging assembler code. Currently the only GNU extension is the location of the current working directory at assembling time.

--gdwarf-2
Generate DWARF2 debugging information for each assembler line. This may help debugging assembler code, if the debugger can handle it. Note---this option is only supported by some targets, not all of them.

--gdwarf-sections
Instead of creating a .debug_line section, create a series of .debug_line.*foo* sections where *foo* is the name of the corresponding code section. For example a code section called *.text.func* will have its dwarf line number information placed into a section called *.debug_line.text.func*. If the code section is just called *.text* then debug line section will still be called just *.debug_line* without any suffix.

--size-check=error
--size-check=warning
Issue an error or warning for invalid ELF .size directive.

--help
Print a summary of the command line options and exit.

--target-help
Print a summary of all target specific options and exit.

-I *dir*
Add directory *dir* to the search list for `.include` directives.

-J Don't warn about signed overflow.

-K Issue warnings when difference tables altered for long displacements.

-L
--keep-locals
Keep (in the symbol table) local symbols. These symbols start with system-specific local label prefixes, typically **.L** for ELF systems or **L** for traditional a.out systems.

--listing-lhs-width=*number*
Set the maximum width, in words, of the output data column for an assembler listing to *number*.

--listing-lhs-width2=*number*
Set the maximum width, in words, of the output data column for continuation lines in an assembler listing to *number*.

--listing-rhs-width=*number*
Set the maximum width of an input source line, as displayed in a listing, to *number* bytes.

--listing-cont-lines=*number*
Set the maximum number of lines printed in a listing for a single line of input to *number* + 1.

-o *objfile*
Name the object-file output from **as** *objfile*.

-R Fold the data section into the text section.

Set the default size of GAS's hash tables to a prime number close to *number*. Increasing this value can reduce the length of time it takes the assembler to perform its tasks, at the expense of increasing the assembler's memory requirements. Similarly reducing this value can reduce the memory requirements at the expense of speed.

--reduce-memory-overheads
This option reduces GAS's memory requirements, at the expense of making the assembly processes slower. Currently this switch is a synonym for **--hash-size=4051**, but in the future it may have other effects as well.

--statistics
Print the maximum space (in bytes) and total time (in seconds) used by assembly.

--strip-local-absolute
Remove local absolute symbols from the outgoing symbol table.

-v
-version
Print the **as** version.

--version
Print the **as** version and exit.

-W
--no-warn
Suppress warning messages.

--fatal-warnings
Treat warnings as errors.

--warn
Don't suppress warning messages or treat them as errors.

-w Ignored.

-x Ignored.

-Z Generate an object file even after errors.

-- | *files* **...**
Standard input, or source files to assemble.

The following options are available when as is configured for the 64-bit mode of the ARM Architecture (AArch64).

-EB
This option specifies that the output generated by the assembler should be marked as being encoded for a big-endian processor.

-EL
This option specifies that the output generated by the assembler should be marked as being encoded for a little-endian processor.

-mabi=*abi*
Specify which ABI the source code uses. The recognized arguments are: `ilp32` and `lp64`, which decides the generated object file in ELF32 and ELF64 format respectively. The default is `lp64`.

-mcpu=*processor***[+***extension***...]**
This option specifies the target processor. The assembler will issue an error message if an attempt is made to assemble an instruction which will not execute on the target processor. The following processor names are recognized: `cortex-a53`, `cortex-a57`, `cortex-a72`, `exynos-m1`, `thunderx`, `xgene1`, and `xgene2`. The special name`all` may be used to allo w the assembler to accept instructions valid for any supported processor, including all optional extensions.

In addition to the basic instruction set, the assembler can be told to accept, or restrict, various extension mnemonics that extend the processor.

If some implementations of a particular processor can have an extension, then then those extensions are automatically enabled. Consequently, you will not normally have to specify any additional extensions.

-march=*architecture***[+***extension***...]**
This option specifies the target architecture. The assembler will issue an error message if an attempt is made to assemble an instruction which will not execute on the target architecture. The only value for *architecture* is `armv8-a`.

If both **-mcpu** and **-march** are specified, the assembler will use the setting for **-mcpu**. If neither are specified, the assembler will default to **-mcpu=all**.

The architecture option can be extended with the same instruction set extension options as the **-mcpu** option. Unlike **-mcpu**, extensions are not always enabled by default,

-mverbose-error
This option enables verbose error messages for AArch64 gas. This option is enabled by default.

-mno-verbose-error
This option disables verbose error messages in AArch64 gas.

The following options are available when as is configured for an Alpha processor.

-m*cpu*
This option specifies the target processor. If an attempt is made to assemble an instruction which will not execute on the target processor, the assembler may either expand the instruction as a macro or issue an error message. This option is equivalent to the `.arch` directive.

The following processor names are recognized: `21064`, `21064a`, `21066`, `21068`, `21164`, `21164a`, `21164pc`, `21264`, `21264a`, `21264b`, `ev4`, `ev5`, `lca45`, `ev5`, `ev56`, `pca56`, `ev6`, `ev67`, `ev68`. The special name `all` may be used to allow the assembler to accept instructions valid for any Alpha

processor.

In order to support existing practice in OSF/1 with respect to `.arch`, and existing practice within **MILO** (the Linux ARC bootloader), the numbered processor names (e.g. 21064) enable the processor-specific PALcode instructions, while the ''electro-vlasic'' names (e.g. `ev4`) do not.

-mdebug
-no-mdebug
Enables or disables the generation of `.mdebug` encapsulation for stabs directives and procedure descriptors. The default is to automatically enable `.mdebug` when the first stabs directive is seen.

-relax
This option forces all relocations to be put into the object file, instead of saving space and resolving some relocations at assembly time. Note that this option does not propagate all symbol arithmetic into the object file, because not all symbol arithmetic can be represented. However, the option can still be useful in specific applications.

-replace
-noreplace
Enables or disables the optimization of procedure calls, both at assemblage and at link time. These options are only available for VMS targets and `-replace` is the default. See section 1.4.1 of the OpenVMS Linker Utility Manual.

-g This option is used when the compiler generates debug information. When **gcc** is using **mips-tfile** to generate debug information for ECOFF, local labels must be passed through to the object file. Otherwise this option has no effect.

-G*size*
A local common symbol larger than *size* is placed in `.bss`, while smaller symbols are placed in `.sbss`.

-F
-32addr
These options are ignored for backward compatibility.

The following options are available when as is configured for an ARC processor.

-marc[5|6|7|8]
This option selects the core processor variant.

-EB | -EL
Select either big-endian (-EB) or little-endian (-EL) output.

The following options are available when as is configured for the ARM processor family.

-mcpu=*processor***[+***extension***...]**
Specify which ARM processor variant is the target.

-march=*architecture***[+***extension***...]**
Specify which ARM architecture variant is used by the target.

-mfpu=*floating-point-format*
Select which Floating Point architecture is the target.

-mfloat-abi=*abi*
Select which floating point ABI is in use.

-mthumb
Enable Thumb only instruction decoding.

-mapcs-32 | -mapcs-26 | -mapcs-float | -mapcs-reentrant
Select which procedure calling convention is in use.

-EB | -EL
Select either big-endian (-EB) or little-endian (-EL) output.

-mthumb-interwork
Specify that the code has been generated with interworking between Thumb and ARM code in mind.

-mccs
Turns on CodeComposer Studio assembly syntax compatibility mode.

-k Specify that PIC code has been generated.

The following options are available when as is configured for the Blackfin processor family.

-mcpu=*processor*[**-***sirevision*]
This option specifies the target processor. The optional *sirevision* is not used in assembler. It's here such that GCC can easily pass down its `-mcpu=` option. The assembler will issue an error message if an attempt is made to assemble an instruction which will not execute on the target processor. The following processor names are recognized: `bf504`, `bf506`, `bf512`, `bf514`, `bf516`, `bf518`, `bf522`, `bf523`, `bf524`, `bf525`, `bf526`, `bf527`, `bf531`, `bf532`, `bf533`, `bf534`, `bf535` (not implemented yet), `bf536`, `bf537`, `bf538`, `bf539`, `bf542`, `bf542m`, `bf544`, `bf544m`, `bf547`, `bf547m`, `bf548`, `bf548m`, `bf549`, `bf549m`, `bf561`, and `bf592`.

-mfdpic
Assemble for the FDPIC ABI.

-mno-fdpic
-mnopic
Disable -mfdpic.

See the info pages for documentation of the CRIS-specific options.

The following options are available when as is configured for a D10V processor.

-O Optimize output by parallelizing instructions.

The following options are available when as is configured for a D30V processor.

-O Optimize output by parallelizing instructions.

-n Warn when nops are generated.

-N Warn when a nop after a 32-bit multiply instruction is generated.

The following options are available when as is configured for an Epiphany processor.

-mepiphany
Specifies that the both 32 and 16 bit instructions are allowed. This is the default behavior.

-mepiphany16
Restricts the permitted instructions to just the 16 bit set.

The following options are available when as is configured for an H8/300 processor. `@chapter` H8/300 Dependent Features

Options
The Renesas H8/300 version of `as` has one machine-dependent option:

-h-tick-hex
Support H'00 style hex constants in addition to 0x00 style.

The following options are available when as is configured for an i386 processor.

--32 | --x32 | --64
Select the word size, either 32 bits or 64 bits. **--32** implies Intel i386 architecture, while **--x32** and **--64** imply AMD x86-64 architecture with 32-bit or 64-bit word-size respectively.

These options are only available with the ELF object file format, and require that the necessary BFD support has been included (on a 32-bit platform you have to add --enable-64-bit-bfd to configure

enable 64-bit usage and use x86-64 as target platform).

-n By default, x86 GAS replaces multiple nop instructions used for alignment within code sections with multi-byte nop instructions such as leal 0(%esi,1),%esi. This switch disables the optimization.

--divide
On SVR4-derived platforms, the character / is treated as a comment character, which means that it cannot be used in expressions. The **--divide** option turns / into a normal character. This does not disable / at the beginning of a line starting a comment, or affect using # for starting a comment.

-march=*CPU***[+***EXTENSION***...]**
This option specifies the target processor. The assembler will issue an error message if an attempt is made to assemble an instruction which will not execute on the target processor. The following processor names are recognized: `i8086`, `i186`, `i286`, `i386`, `i486`, `i586`, `i686`, `pentium`, `pentiumpro`, `pentiumii`, `pentiumiii`, `pentium4`, `prescott`, `nocona`, `core`, `core2`, `corei7`, `l1om`, `k1om`, `k6`, `k6_2`, `athlon`, `opteron`, `k8`, `amdfam10`, `bdver1`, `bdver2`, `bdver3`, `bdver4`, `btver1`, `btver2`, `generic32` and `generic64`.

In addition to the basic instruction set, the assembler can be told to accept various extension mnemonics. For example, `-march=i686+sse4+vmx` extends *i686* with *sse4* and *vmx*. The following extensions are currently supported: `8087`, `287`, `387`, `no87`, `mmx`, `nommx`, `sse`, `sse2`, `sse3`, `ssse3`, `sse4.1`, `sse4.2`, `sse4`, `nosse`, `avx`, `avx2`, `adx`, `rdseed`, `prfchw`, `smap`, `mpx`, `sha`, `prefetchwt1`, `clflushopt`, `se1`, `clwb`, `pcommit`, `avx512f`, `avx512cd`, `avx512er`, `avx512pf`, `avx512vl`, `avx512bw`, `avx512dq`, `avx512ifma`, `avx512vbmi`, `noavx`, `vmx`, `vmfunc`, `smx`, `xsave`, `xsaveopt`, `xsavec`, `xsaves`, `aes`, `pclmul`, `fsgsbase`, `rdrnd`, `f16c`, `bmi2`, `fma`, `movbe`, `ept`, `lzcnt`, `hle`, `rtm`, `invpcid`, `clflush`, `lwp`, `fma4`, `xop`, `cx16`, `syscall`, `rdtscp`, `3dnow`, `3dnowa`, `sse4a`, `sse5`, `svme`, `abm` and `padlock`. Note that rather than extending a basic instruction set, the extension mnemonics starting with `no` revoke the respective functionality.

When the `.arch` directive is used with **-march**, the `.arch` directive will take precedent.

-mtune=*CPU*
This option specifies a processor to optimize for. When used in conjunction with the **-march** option, only instructions of the processor specified by the **-march** option will be generated.

Valid *CPU* values are identical to the processor list of **-march=***CPU*.

-msse2avx
This option specifies that the assembler should encode SSE instructions with VEX prefix.

-msse-check=*none*
-msse-check=*warning*
-msse-check=*error*
These options control if the assembler should check SSE instructions. **-msse-check=***none* will mak e the assembler not to check SSE instructions, which is the default. **-msse-check=***warning* will make the assembler issue a warning for any SSE instruction. **-msse-check=***err or* will make the assembler issue an error for any SSE instruction.

-mavxscalar=*128*
-mavxscalar=*256*
These options control how the assembler should encode scalar AVX instructions. **-mavxscalar=***128* will encode scalar AVX instructions with 128bit vector length, which is the default. **-mavxscalar=***256* will encode scalar AVX instructions with 256bit vector length.

-mevexlig=*128*
-mevexlig=*256*
-mevexlig=*512*
These options control how the assembler should encode length-ignored (LIG) EVEX instructions. **-mevexlig=***128* will encode LIG EVEX instructions with 128bit vector length, which is the default. **-mevexlig=***256* and **-mevexlig=***512* will encode LIG EVEX instructions with 256bit and 512bit vector

length, respectively.

-mevexwig=*0*
-mevexwig=*1*
These options control how the assembler should encode w-ignored (WIG) EVEX instructions. **-mevexwig=***0* will encode WIG EVEX instructions with evex.w = 0, which is the default. **-mevexwig=***1* will encode WIG EVEX instructions with evex.w = 1.

-mmnemonic=*att*
-mmnemonic=*intel*
This option specifies instruction mnemonic for matching instructions. The `.att_mnemonic` and `.intel_mnemonic` directives will take precedent.

-msyntax=*att*
-msyntax=*intel*
This option specifies instruction syntax when processing instructions. The `.att_syntax` and `.intel_syntax` directives will take precedent.

-mnaked-reg
This opetion specifies that registers don't require a `%` prefix. The `.att_syntax` and `.intel_syntax` directives will take precedent.

-madd-bnd-prefix
This option forces the assembler to add BND prefix to all branches, even if such prefix was not explicitly specified in the source code.

-mbig-obj
On x86-64 PE/COFF target this option forces the use of big object file format, which allows more than 32768 sections.

-momit-lock-prefix=*no*
-momit-lock-prefix=*yes*
These options control how the assembler should encode lock prefix. This option is intended as a workaround for processors, that fail on lock prefix. This option can only be safely used with single-core, single-thread computers **-momit-lock-prefix=***yes* will omit all lock prefixes. **-momit-lock-prefix=***no* will encode lock prefix as usual, which is the default.

-mevexrcig=*rne*
-mevexrcig=*rd*
-mevexrcig=*ru*
-mevexrcig=*rz*
These options control how the assembler should encode SAE-only EVEX instructions. **-mevexrcig=***rne* will encode RC bits of EVEX instruction with 00, which is the default. **-mevexrcig=***rd*, **-mevexrcig=***ru* and **-mevexrcig=***rz* will encode SAE-only EVEX instructions with 01, 10 and 11 RC bits, respectively.

The following options are available when as is configured for the Intel 80960 processor.

-ACA | -ACA_A | -ACB | -ACC | -AKA | -AKB | -AKC | -AMC
Specify which variant of the 960 architecture is the target.

-b Add code to collect statistics about branches taken.

-no-relax
Do not alter compare-and-branch instructions for long displacements; error if necessary.

The following options are available when as is configured for the Ubicom IP2K series.

-mip2022ext
Specifies that the extended IP2022 instructions are allowed.

-mip2022
Restores the default behaviour, which restricts the permitted instructions to just the basic IP2022 ones.

The following options are available when as is configured for the Renesas M32C and M16C processors.

-m32c
Assemble M32C instructions.

-m16c
Assemble M16C instructions (the default).

-relax
Enable support for link-time relaxations.

-h-tick-hex
Support H'00 style hex constants in addition to 0x00 style.

The following options are available when as is configured for the Renesas M32R (formerly Mitsubishi M32R) series.

--m32rx
Specify which processor in the M32R family is the target. The default is normally the M32R, but this option changes it to the M32RX.

--warn-explicit-parallel-conflicts or --Wp
Produce warning messages when questionable parallel constructs are encountered.

--no-warn-explicit-parallel-conflicts or --Wnp
Do not produce warning messages when questionable parallel constructs are encountered.

The following options are available when as is configured for the Motorola 68000 series.

-l Shorten references to undefined symbols, to one word instead of two.

-m68000 | -m68008 | -m68010 | -m68020 | -m68030
| -m68040 | -m68060 | -m68302 | -m68331 | -m68332
| -m68333 | -m68340 | -mcpu32 | -m5200
Specify what processor in the 68000 family is the target. The default is normally the 68020, but this can be changed at configuration time.

-m68881 | -m68882 | -mno-68881 | -mno-68882
The target machine does (or does not) have a floating-point coprocessor. The default is to assume a coprocessor for 68020, 68030, and cpu32. Although the basic 68000 is not compatible with the 68881, a combination of the two can be specified, since it's possible to do emulation of the coprocessor instructions with the main processor.

-m68851 | -mno-68851
The target machine does (or does not) have a memory-management unit coprocessor. The default is to assume an MMU for 68020 and up.

The following options are available when as is configured for an Altera Nios II processor.

-relax-section
Replace identified out-of-range branches with PC-relative `jmp` sequences when possible. The generated code sequences are suitable for use in position-independent code, but there is a practical limit on the extended branch range because of the length of the sequences. This option is the default.

-relax-all
Replace branch instructions not determinable to be in range and all call instructions with `jmp` and `callr` sequences (respectively). This option generates absolute relocations against the target symbols and is not appropriate for position-independent code.

-no-relax
Do not replace any branches or calls.

-EB
Generate big-endian output.

-EL
Generate little-endian output. This is the default.

The following options are available when as is configured for a Meta processor.

`-mcpu=metac11`
Generate code for Meta 1.1.

`-mcpu=metac12`
Generate code for Meta 1.2.

`-mcpu=metac21`
Generate code for Meta 2.1.

`-mfpu=metac21`
Allow code to use FPU hardware of Meta 2.1.

See the info pages for documentation of the MMIX-specific options.

The following options are available when as is configured for a NDS32 processor.

`-O1`
Optimize for performance.

`-Os`
Optimize for space.

`-EL`
Produce little endian data output.

`-EB`
Produce little endian data output.

`-mpic`
Generate PIC.

`-mno-fp-as-gp-relax`
Suppress fp-as-gp relaxation for this file.

`-mb2bb-relax`
Back-to-back branch optimization.

`-mno-all-relax`
Suppress all relaxation for this file.

`-march=<arch name>`
Assemble for architecture <arch name> which could be v3, v3j, v3m, v3f, v3s, v2, v2j, v2f, v2s.

`-mbaseline=<baseline>`
Assemble for baseline <baseline> which could be v2, v3, v3m.

`-mfpu-freg=`*`FREG`*
Specify a FPU configuration.

```
0 8 SP / 4 DP registers
1 16 SP / 8 DP registers
2 32 SP / 16 DP registers
3 32 SP / 32 DP registers
```

`-mabi=`*`abi`*
Specify a abi version <abi> could be v1, v2, v2fp, v2fpp.

`-m[no-]mac`
Enable/Disable Multiply instructions support.

`-m[no-]div`
Enable/Disable Divide instructions support.

`-m[no-]16bit-ext`
Enable/Disable 16-bit extension

`-m[no-]dx-regs`
Enable/Disable d0/d1 registers

`-m[no-]perf-ext`
Enable/Disable Performance extension

`-m[no-]perf2-ext`
Enable/Disable Performance extension 2

`-m[no-]string-ext`
Enable/Disable String extension

`-m[no-]reduced-regs`
Enable/Disable Reduced Register configuration (GPR16) option

`-m[no-]audio-isa-ext`
Enable/Disable AUDIO ISA extension

`-m[no-]fpu-sp-ext`
Enable/Disable FPU SP extension

`-m[no-]fpu-dp-ext`
Enable/Disable FPU DP extension

`-m[no-]fpu-fma`
Enable/Disable FPU fused-multiply-add instructions

`-mall-ext`
Turn on all extensions and instructions support

The following options are available when as is configured for a PowerPC processor.

-a32
Generate ELF32 or XCOFF32.

-a64
Generate ELF64 or XCOFF64.

-K PIC
Set EF_PPC_RELOCATABLE_LIB in ELF flags.

-mpwrx | -mpwr2
Generate code for POWER/2 (RIOS2).

-mpwr
Generate code for POWER (RIOS1)

-m601
Generate code for PowerPC 601.

-mppc, -mppc32, -m603, -m604
Generate code for PowerPC 603/604.

-m403, -m405
Generate code for PowerPC 403/405.

-m440
Generate code for PowerPC 440. BookE and some 405 instructions.

-m464
Generate code for PowerPC 464.

-m476
Generate code for PowerPC 476.

-m7400, -m7410, -m7450, -m7455
Generate code for PowerPC 7400/7410/7450/7455.

-m750cl
Generate code for PowerPC 750CL.

-mppc64, -m620
Generate code for PowerPC 620/625/630.

-me500, -me500x2
Generate code for Motorola e500 core complex.

-me500mc
Generate code for Freescale e500mc core complex.

-me500mc64
Generate code for Freescale e500mc64 core complex.

-me5500
Generate code for Freescale e5500 core complex.

-me6500
Generate code for Freescale e6500 core complex.

-mspe
Generate code for Motorola SPE instructions.

-mtitan
Generate code for AppliedMicro Titan core complex.

-mppc64bridge
Generate code for PowerPC 64, including bridge insns.

-mbooke
Generate code for 32-bit BookE.

-ma2
Generate code for A2 architecture.

-me300
Generate code for PowerPC e300 family.

-maltivec
Generate code for processors with AltiVec instructions.

-mvle
Generate code for Freescale PowerPC VLE instructions.

-mvsx
Generate code for processors with Vector-Scalar (VSX) instructions.

-mhtm
Generate code for processors with Hardware Transactional Memory instructions.

-mpower4, -mpwr4
Generate code for Power4 architecture.

-mpower5, -mpwr5, -mpwr5x
Generate code for Power5 architecture.

-mpower6, -mpwr6
Generate code for Power6 architecture.

-mpower7, -mpwr7
Generate code for Power7 architecture.

-mpower8, -mpwr8
Generate code for Power8 architecture.

-mcell
-mcell
Generate code for Cell Broadband Engine architecture.

-mcom
Generate code Power/PowerPC common instructions.

-many
Generate code for any architecture (PWR/PWRX/PPC).

-mregnames
Allow symbolic names for registers.

-mno-regnames
Do not allow symbolic names for registers.

-mrelocatable
Support for GCC's -mrelocatable option.

-mrelocatable-lib
Support for GCC's -mrelocatable-lib option.

-memb
Set PPC_EMB bit in ELF flags.

-mlittle, -mlittle-endian, -le
Generate code for a little endian machine.

-mbig, -mbig-endian, -be
Generate code for a big endian machine.

-msolaris
Generate code for Solaris.

-mno-solaris
Do not generate code for Solaris.

-nops=*count*
If an alignment directive inserts more than *count* nops, put a branch at the beginning to skip execution of the nops.

See the info pages for documentation of the RX-specific options.

The following options are available when as is configured for the s390 processor family.

-m31
-m64
Select the word size, either 31/32 bits or 64 bits.

-mesa
-mzarch
Select the architecture mode, either the Enterprise System Architecture (esa) or the z/Architecture mode (zarch).

-march=*processor*
Specify which s390 processor variant is the target, **g6**, **g6**, **z900**, **z990**, **z9-109**, **z9-ec**, **z10**, **z196**, or **zEC12**.

-mregnames
-mno-regnames
Allow or disallow symbolic names for registers.

-mwarn-areg-zero
Warn whenever the operand for a base or index register has been specified but evaluates to zero.

The following options are available when as is configured for a TMS320C6000 processor.

-march=*arch*
Enable (only) instructions from architecture *arch*. By default, all instructions are permitted.

The following values of *arch* are accepted: `c62x`, `c64x`, `c64x+`, `c67x`, `c67x+`, `c674x`.

-mdsbt
-mno-dsbt
The **-mdsbt** option causes the assembler to generate the `Tag_ABI_DSBT` attribute with a value of 1, indicating that the code is using DSBT addressing. The **-mno-dsbt** option, the default, causes the tag to have a value of 0, indicating that the code does not use DSBT addressing. The linker will emit a warning if objects of different type (DSBT and non-DSBT) are linked together.

-mpid=no
-mpid=near
-mpid=far
The **-mpid=** option causes the assembler to generate the `Tag_ABI_PID` attribute with a value indicating the form of data addressing used by the code. **-mpid=no**, the default, indicates position-dependent data addressing, **-mpid=near** indicates position-independent addressing with GOT accesses using near DP addressing, and **-mpid=far** indicates position-independent addressing with GOT accesses using far DP addressing. The linker will emit a warning if objects built with different settings of this option are linked together.

-mpic
-mno-pic
The **-mpic** option causes the assembler to generate the `Tag_ABI_PIC` attribute with a value of 1, indicating that the code is using position-independent code addressing, The `-mno-pic` option, the default, causes the tag to have a value of 0, indicating position-dependent code addressing. The linker will emit a warning if objects of different type (position-dependent and position-independent) are linked together.

-mbig-endian
-mlittle-endian
Generate code for the specified endianness. The default is little-endian.

The following options are available when as is configured for a TILE-Gx processor.

-m32 | -m64
Select the word size, either 32 bits or 64 bits.

-EB | -EL
Select the endianness, either big-endian (-EB) or little-endian (-EL).

The following options are available when as is configured for an Xtensa processor.

--text-section-literals | --no-text-section-literals
Control the treatment of literal pools. The default is **--no-text-section-literals**, which places literals in separate sections in the output file. This allows the literal pool to be placed in a data RAM/ROM. With **--text-section-literals**, the literals are interspersed in the text section in order to keep them as close as possible to their references. This may be necessary for large assembly files, where the literals would otherwise be out of range of the`L32R` instructions in the te xt section. These options only affect literals referenced via PC-relative `L32R` instructions; literals for absolute mode `L32R` instructions are handled separately.

--absolute-literals | --no-absolute-literals
Indicate to the assembler whether `L32R` instructions use absolute or PC-relative addressing. If the processor includes the absolute addressing option, the default is to use absolute `L32R` relocations. Otherwise, only the PC-relative `L32R` relocations can be used.

--target-align | --no-target-align
Enable or disable automatic alignment to reduce branch penalties at some expense in code size. This optimization is enabled by default. Note that the assembler will always align instructions like `LOOP` that have fixed alignment requirements.

--longcalls | --no-longcalls
Enable or disable transformation of call instructions to allow calls across a greater range of addresses. This option should be used when call targets can potentially be out of range. It may degrade both code size and performance, but the linker can generally optimize away the unnecessary overhead when a call ends up within range. The default is **--no-longcalls**.

--transform | --no-transform
Enable or disable all assembler transformations of Xtensa instructions, including both relaxation and optimization. The default is **--transform**; **--no-transform** should only be used in the rare cases when the instructions must be exactly as specified in the assembly source. Using **--no-transform** causes out of range instruction operands to be errors.

--rename-section *oldname*=*newname*
Rename the *oldname* section to *newname*. This option can be used multiple times to rename multiple sections.

--trampolines | --no-trampolines
Enable or disable transformation of jump instructions to allow jumps across a greater range of addresses. This option should be used when jump targets can potentially be out of range. In the absence of such jumps this option does not affect code size or performance. The default is **--trampolines**.

The following options are available when as is configured for a Z80 family processor.

-z80
Assemble for Z80 processor.

-r800
Assemble for R800 processor.

-ignore-undocumented-instructions
-Wnud
Assemble undocumented Z80 instructions that also work on R800 without warning.

-ignore-unportable-instructions
-Wnup
Assemble all undocumented Z80 instructions without warning.

-warn-undocumented-instructions
-Wud
Issue a warning for undocumented Z80 instructions that also work on R800.

-warn-unportable-instructions
-Wup
Issue a warning for undocumented Z80 instructions that do not work on R800.

-forbid-undocumented-instructions
-Fud
Treat all undocumented instructions as errors.

-forbid-unportable-instructions

-Fup
Treat undocumented Z80 instructions that do not work on R800 as errors.

SEE ALSO

gcc(1), *ld(1)*, and the Info entries for *binutils* and *ld*.

COPYRIGHT