

I INDEX

Symbols

__builtin_quad [2-57](#)

__emuclock (get simulator cycle count)
function [3-84](#)

__regclass Construct [2-63](#)

μ -Law compression [3-123](#)

μ -Law expansion [3-124](#)

Numerics

16-Bit data types [2-70](#)

2-D convolution [3-66](#)

32-Bit data types [2-81](#)

A

-A (assert) compiler switch [2-20](#)

a_compress (A-law compression)
function [3-32](#)

a_expand (A-law expansion) function
[3-34](#)

Abridged C++ Library [3-21](#)

abs (absolute value) function [3-30](#)

acos (arc cosine) function [3-31](#)

addbitrev (bit-reversed adder) function
[3-33](#)

-alttok (alternative tokens) C++ mode
compiler switch [2-40](#)

-analog (Analog C compilation)
compiler switch [2-18](#)

-ansi (ANSI standard C compilation)
compiler switch [2-18](#)

ANSI standard compiler [2-1](#), [2-2](#), [2-3](#),
[2-9](#), [2-18](#), [2-19](#), [2-23](#)

arg (get phase of a complex number)
function [3-35](#)

Arguments

handling [2-103](#)

incoming [2-100](#)

number [2-106](#)

outgoing [2-101](#)

passage [2-103](#)

Arithmetic

mixed mode [2-90](#)

Arithmetic operators

fractional arithmetic [2-89](#)

fractional values [2-88](#)

int4x16 values [2-73](#), [2-79](#)

Arrays

variable length [2-66](#)

asin (arc sine) function [3-36](#)

Assembler (easmts) [2-121](#)

Assignments [2-60](#)

atan (arc tangent) function [3-37](#)

INDEX

atan2 (arc tangent of quotient)
function [3-38](#)
atof (string to double conversion)
function [3-39](#)
atoi (string to integer conversion)
function [3-40](#)
atol (string to long conversion)
function [3-41](#)
atold (string to long double
conversion) function [3-42](#)
autocoh (autocoherence) function
[3-43](#)
autocorr (autocorrelation) function
[3-44](#)
Automatic variables [2-58](#)
avg (returns the mean of two values)
function [3-45](#)

B

Binary search [3-46](#)
Bit FIFO Temporaries register
[2-115](#)
Bitwise operators [2-74](#)
-bool (accept boolean) C++ mode
compiler switch [2-40](#)
bool (See Boolean type support
keywords (bool, true, false))
Boolean type support keywords
(bool, true, false) [2-54](#), [2-65](#)
bsearch (binary search) function
[3-46](#)
-build-lib (build library) compiler
switch [2-20](#)
Built-in functions [2-70](#), [3-29](#)

C

C

callable assembly subroutines,
writing [2-116](#)
-C (comments) compiler switch
[2-20](#)
-c (compile only) compiler switch
[2-21](#)
C language extensions
C++-style comments [2-55](#)
warning generation [2-55](#)
-c++ (C++ mode) compiler switch
[2-19](#)
C++ mode option set [2-40](#)
C++ style comments [2-87](#)
C/C++
callable routines from the
assembler [2-120](#)
C/C++ and assembly interface
[2-111](#)
C/C++ compiler
common switches [2-19](#)
glossary [2-121](#)
C/C++ language extensions [2-54](#)
__regclass(unit) keyword [2-54](#)
bool keyword [2-54](#)
false keyword [2-54](#)
indexed initializers [2-55](#)
long long [int] [2-57](#)
non-constant initializers [2-55](#)
section keyword [2-54](#)
segment keyword [2-54](#)
summary [2-53](#)

- true keyword [2-54](#)
 - variable length arrays [2-55](#)
- C/C++ mode selection switches [2-18](#)
- C/C++ run-time
 - environment [2-121](#)
 - library guide [3-3–3-28](#)
 - model [2-97](#)
- cabs (complex absolute value)
 - function [3-47](#)
- cadd (complex addition) function [3-48](#)
- Calling
 - library functions [3-3](#)
- cdiv (complex division) function [3-49](#)
- ceil (ceiling) function [3-50](#)
- cexp (complex exponential)
 - function [3-51](#)
- cfft (N point 2-d complex input FFT) function [3-52](#)
- cfft2d (NxN point 2-d complex input FFT) function [3-54](#)
- clip (clip) function [3-56](#)
- cmatmadd (complex matrix + matrix addition) function [3-57](#)
- cmatmmlt (complex matrix * matrix multiplication) function [3-63](#)
- cmatmsub (complex matrix - matrix subtraction) function [3-59](#)
- cmatsadd (complex matrix + scalar addition) function [3-60](#)
- cmatmlt (complex matrix * scalar multiplication) function [3-61](#)

- cmatssub (complex matrix - scalar subtraction) function [3-62](#)
- cmlt (complex multiply) function [3-63](#)
- Code
 - inline assembly [2-97](#)
 - sequences [2-108](#)
 - source [2-35](#)
- Comparison operators [2-75](#)
- Compiler
 - command-line reference [2-3–2-52](#)
 - language extensions [2-53–2-90](#)
 - options, setting [2-97](#)
 - overview [2-1](#)
- Complex
 - header file (DSP) [3-14](#)
- Complex absolute value function [3-47](#)
- Complex addition function [3-48](#)
- Complex conjugate function [3-64](#)
- Complex division function [3-49](#)
- Complex exponential function [3-51](#)
- Complex functions
 - complex matrix * scalar multiplication [3-61](#)
- Complex matrix - matrix subtraction function [3-59](#)
- Complex matrix - scalar subtraction function [3-62](#)
- Complex matrix * matrix multiplication function [3-58](#)
- Complex matrix + matrix addition function [3-57](#)

INDEX

- Complex matrix + scalar addition
function [3-60](#)
- Complex multiply function [3-63](#)
- Complex subtraction function [3-74](#)
- Complex vector - scalar subtraction
function [3-78](#)
- Complex vector - vector subtraction
function [3-80](#)
- Complex vector * scalar
multiplication function [3-77](#)
- Complex vector * vector
multiplication function [3-81](#)
- Complex vector + scalar addition
function [3-76](#)
- Complex vector + vector addition
function [3-79](#)
- Complex vector dot product
function [3-75](#)
- Compound statements with in
expressions support (compiler)
[2-67](#)
- Compute Block (x & y) General
registers [2-112](#)
- Compute Block ALU Summation
registers [2-114](#)
- Compute Block X MAC registers
[2-114](#)
- Compute Block Y MAC registers
[2-114](#)
- conj (complex conjugate) function
[3-64](#)
- const keyword [2-19](#)
- Construct from polar coordinates
function [3-126](#)
- Constructors [2-72](#), [2-77](#)
- Control
 - of inlining [2-50](#)
 - of optimization [2-48](#)
- conv2d (2-d convolution)) function
[3-66](#)
- Conversions
 - fractional values [2-89](#)
- convolve (convolution) function
[3-65](#)
- copysign (copysign) function [3-67](#)
- cos (cosine) function [3-68](#)
- cosh (hyperbolic cosine) function
[3-69](#)
- cot (cotangent) function [3-70](#)
- Count zero crossing function [3-162](#)
- count_ones (count one bits in word)
function [3-71](#)
- crosscoh (cross-coherence) function
[3-72](#)
- crosscorr (cross-correlation)
function [3-73](#)
- csub (complex subtraction) function
[3-74](#)
- Customer support [1-2](#)
- cvecdot (complex vector dot
product) function [3-75](#)
- cvecsadd (complex vector + scalar
addition) function [3-76](#)
- cvecsmult (complex vector * scalar
multiplication) function [3-77](#)
- cvecssub (complex vector - scalar
subtraction) function [3-78](#)

cvecvadd (complex vector + vector addition) function [3-79](#)
 cvecvmlt (complex vector * vector multiplication) function [3-81](#)
 cvecvsub (complex vector - vector subtraction) function [3-80](#)

D

-D (define macro) compiler switch [2-21](#), [2-36](#)

Data Alignment registers [2-114](#)

data type [2-57](#)

Data types

16-bit [2-70](#)

16-bit support [2-70](#)

32-bit [2-81](#)

32-bit support [2-81](#)

fract [2-55](#)

non-standard [2-83–2-87](#)

sizes [2-45](#)

Debugging information, remove [2-34](#)

-default-linkage (assembler, C, or C++) compiler switch [2-21](#)

div (integer division) function [3-82](#)

dm (See Dual memory support keywords (pm dm))

-dollar (dollar format) compiler switch [2-21](#)

-double-size-32 (IEEE single-precision double) compiler switch [2-21](#)

-double-size-64 (IEEE double-precision double) compiler switch [2-21](#)

-dry (terse -dry-run) compiler switch [2-23](#)

-dryrun (verbose dry-run) compiler switch [2-23](#)

DSP

filters [3-15](#)

header files [3-14–3-21](#)

Transformations [3-15](#)

Dual memory support keywords (pm dm) [2-18](#), [2-54](#), [2-58](#)

E

-E (stop after preprocessing) compiler switch [2-23](#)

-EE (run after preprocessing) compiler switch [2-23](#)

exp (exponential) function [3-83](#)

-explicit (explicit specifier) C++ mode compiler switch [2-41](#)

Extractors

2x16 from a 4x16 value [2-78](#)

and expanders

int2x16 values [2-73](#)

int2x32 values [2-82](#)

-extra-keywords (not quite -analog) compiler switch [2-23](#)

INDEX

F

false (See Boolean Type Support
Keywords (bool, true, false))

File

extension [2-7](#)

Files

extensions [2-7](#), [2-19](#)

intermediate [2-34](#)

searches [2-6](#)

temporary [2-34](#)

Filters

DSP [3-15](#)

fir (finite impulse response filter)

function [3-85](#)

fir_decima (FIR decimation filter)

function [3-87](#)

fir_interp (FIR interpolation filter)

function [3-88](#)

-flags (command line input)

compiler switch [2-24](#)

Floating point [2-46](#)

float.h header file [3-10](#)

floor (floor) function [3-89](#)

fmod (floating point remainder)

function [3-90](#)

Formats

fractional literals [2-88](#)

Fract data type (c++ mode only)

[2-55](#)

Fractional

arithmetic [2-89](#)

type [2-88](#)

values [2-88](#)

conversions [2-89](#)

Frame

stack [2-97–2-110](#)

Free space [2-101](#)

frexp (get mantissa and exponent)

function [3-91](#)

Functions

builtin [2-70](#), [3-29](#)

I/O [3-12](#)

linking library [3-4](#)

math.h header file [3-10](#)

matrix header file [3-17](#)

primitive I/O [3-12](#)

run-time library [3-29–3-162](#)

statistical [3-19](#)

vector [3-19](#)

G

-g (generate debug information)

compiler switch [2-24](#)

gen_bartlett (generate bartlett

window) function [3-92](#)

gen_blackman (generate blackman

window) function [3-94](#)

gen_gaussian (generate gaussian

window) function [3-95](#)

gen_hamming (generate hamming

window) function [3-96](#)

gen_hanning (generate hanning

window) function [3-97](#)

gen_harris (generate harris window)

function [3-98](#)

gen_kaiser (generate kaiser window)

function [3-99](#)

[gen_rectangular \(generate rectangular window\) function 3-100](#)
[gen_triangle \(generate triangle window\) function 3-101](#)
[gen_vonhann \(generate von hann window\) function 3-103](#)
[Generate FFT twiddle factors function 3-152](#)
[Get fraction and integer function 3-122](#)
[Get mantissa and exponent function 3-91](#)
[Glossary C/C++ compiler 2-121](#)

H

[-H \(headers\) compiler switch 2-24](#)
[Header files 2-94, 3-7](#)
 [DSP 3-14–3-21](#)
 [search 2-29](#)
 [standard C library 3-8–3-14](#)
 [user 2-94](#)
 [working with library 3-7](#)
[Header files \(C++ for C facilities\)](#)
 [<cassert> 3-25](#)
 [<cctype> 3-25](#)
 [<cerrno> 3-25](#)
 [<cfloat> 3-25](#)
 [<climits> 3-25](#)
 [<locale> 3-25](#)
 [<cmath> 3-25](#)
 [<csetjmp> 3-25](#)
 [<csignal> 3-25](#)

[<cstdarg> 3-25](#)
[<cstddef> 3-25](#)
[<stdio> 3-25](#)
[<stdlib> 3-25](#)
[<cstring> 3-25](#)

Header files (DSP) 3-14–3-21

[complex \(basic complex arithmetic functions\) 3-14](#)
[filter \(filters and transformations\) 3-15](#)
[libsim \(simulator services\) 3-16](#)
[matrix \(matrix functions\) 3-17](#)
[stats \(statistical functions\) 3-19](#)
[vector \(vector functions\) 3-19](#)
[window \(window generators\) 3-21](#)

Header files (embedded C++)

[<fract> 3-22](#)
[<fstream> 3-22](#)
[<iomanip> 3-22](#)
[<ios> 3-23](#)
[<iosfwd> 3-23](#)
[<iostream> 3-23](#)
[<istream> 3-23](#)
[<new> 3-23](#)
[<ostream> 3-23](#)
[complex.h 3-22](#)
[exception.h 3-22](#)
[sstream.h 3-23](#)
[stdexcept.h 3-23](#)
[streambuf.h 3-24](#)
[string.h 3-24](#)
[strstream.h 3-24](#)

Header files (standard C) 3-8–3-14

INDEX

- assert.h [3-10](#)
- ctype.h [3-10](#)
- errno.h [3-10](#)
- float.h [3-10](#)
- limits.h [3-10](#)
- locale.h [3-10](#)
- math.h [3-10](#)
- setjmp.h [3-11](#)
- signal.h [3-11](#)
- stdarg.h [3-12](#)
- stddef.h [3-12](#)
- stdio.h [3-12](#)
- stdlib.h [3-13](#)
- string.h [3-14](#)
- time.h [3-14](#)
- Header files (template library) [3-26](#)
 - <algorithm> [3-26](#)
 - <deque> [3-26](#)
 - <functional> [3-26](#)
 - <hash_map> [3-26](#)
 - <hash_set> [3-26](#)
 - <iterator> [3-26](#)
 - <list> [3-26](#)
 - <map> [3-27](#)
 - <memory> [3-27](#)
 - <numeric> [3-27](#)
 - <queue> [3-27](#)
 - <set> [3-27](#)
 - <stack> [3-27](#)
 - <utility> [3-27](#)
 - <vector> [3-27](#)
- iomanip.h [3-28](#)
- iostream.h [3-28](#)
- new.h [3-28](#)
- stream.h [3-28](#)
- help (command-line help)
 - compiler switch [2-25](#)
- HH (list *.h and compile) compiler switch [2-25](#)
- histogram (histogram) function [3-104](#)
- Hyperbolic cosine function [3-69](#)
- Hyperbolic sine function [3-139](#)
- Hyperbolic tangent function [3-151](#)
- I
 - I (include search directory)
 - compiler switch [2-25](#), [2-29](#)
 - I/O functions [3-12](#)
 - iALU (j & k) Special registers:
 - Circular Buffering –B (base) & L (length) [2-113](#)
 - iALU (j and k) General registers [2-111](#)
 - ifft (N point inverse FFT) function [3-105](#)
 - ifft2d (NxN point 2-d inverse input FFT) function [3-107](#)
 - iir (infinite impulse response filter) function [3-109](#)
 - include (include file) compiler switch [2-25](#)
 - Incoming arguments
 - stacks [2-100](#)
 - Indexed initializer support (compiler) [2-68](#)
 - Inline assembly code [2-97](#)

Inline function support keyword
(inline) [2-18](#), [2-19](#), [2-54](#), [2-56](#),
[2-58](#)

-instant (instantiate all or used) C++
mode compiler switch [2-41](#)

instantiate templates [2-41](#)

-instantused (instantiate used) C++
mode compiler switch [2-41](#)

C/C++ language extensions

long long [2-54](#)

int2x32 [2-81](#)

Integer [2-46](#)

Integer division function [3-82](#)

Integer function [2-53](#)

Interfacing C/C++ & assembly (See
Mixed C/C++/assembly
programming)

Intermediate files [2-34](#)

Intrinsic (built-in) functions [2-70](#),
[3-29](#)

-ipa (interprocedural analysis)
compiler switch [2-25](#)

K

Keywords

(See also Memory keywords)

placement support keyword
(section) [2-64](#)

pointer class support keyword
(restrict) [2-54](#), [2-65](#)

signed [2-19](#)

volatile [2-19](#)

Keywords (compiler) (See also
Compiler, language extensions)

L

-l (link library) compiler switch [2-26](#)

-L (search library) compiler switch
[2-29](#)

Language extensions (compiler) (See
Compiler C extensions)

Leaf

procedure [2-107](#)

routines [2-121](#)

Library

calling functions [3-3](#)

header files [3-7](#)

source code, working with [3-6](#)

Linkage

information [2-100](#)

outgoing [2-101](#)

Linker [2-121](#)

Linking

library functions [3-4](#)

Local

temporaries [2-100](#)

variables [2-100](#)

variables/temporaries [2-100](#)

log (natural logarithm) function
[3-111](#)

log10 (Logarithm base 10) function
[3-112](#)

long long integer type support [2-54](#)

Loop counters [2-114](#)

INDEX

M

-M (make only) compiler switch
2-26

Macros

and the C preprocessor 2-121
writing 2-95

Macros (predefined) 2-91

__ADSPTS__ 2-91
__ADSPTS001__ 2-92
__ADSPTS101__ 2-92
__ANALOG_EXTENSIONS__
2-92
__CPLUSPLUS__ 2-92
__DATE__ 2-92
__DOUBLES_ARE_FLOATS
2-92
__ECC__ 2-92
__EDG__ 2-93
__EDG_VERSION__ 2-93
__FILE__ 2-93
__LINE__ 2-93
__NO_BUILTIN 2-93
__SIGNED_CHARS__ 2-93
__STDC__ 2-93
__STDC_VERSION__ 2-94
__TIME__ 2-94

-map (generate a memory map)
compiler switch 2-27

Map file 2-27

Math functions

math.h header file 3-10

Math intrinsics 2-82

matmadd (real matrix + matrix
addition) function 3-114

matmmlt (real matrix * matrix
addition) function 3-115

matmsub (real matrix - matrix
subtraction) function 3-116

Matrix functions

matrix header file 3-17

matsadd (real matrix + scalar
addition) function 3-113

matsmult (real matrix * scalar
multiplication) function 3-117

matssub (real matrix - scalar
subtraction) function 3-118

max (maximum) function 3-119

mean (mean) function 3-120

Memory keywords

assignments 2-60

function arguments 2-62

function declarations 2-61

macros 2-63

pointers 2-61

type conversions 2-60

min (minimum) function 3-121

Mixed C/assembly programming
2-97

Mixed mode arithmetic 2-90

modf (get fraction and integer)
function 3-122

mu_compress (μ -law compression)
function 3-123

mu_expand (μ -law expansion)
function 3-124

N

N point complex input FFT
function [3-52](#)

N point inverse FFT function [3-105](#)

N point real input FFT function
[3-131](#)

-namespace (enable namespaces)
C++ mode compiler switch
[2-41](#)

Naming files [2-4](#)

Natural logarithm function [3-111](#)

-newforinit (limit scope) C++ mode
compiler switch [2-41](#)

-newvec (new vector) C++ mode
compiler switch [2-41](#)

-no-alttok (disable tokens) C++
mode compiler switch [2-42](#)

-no-bool (disable boolean) C++
mode compiler switch [2-42](#)

-no-builtin (no builtin functions)
compiler switch [2-27](#)

-no-char (disable wide char type)
C++ mode compiler switch
[2-43](#)

-no-defs (disable definitions)
compiler switch [2-27](#)

-no-demangle (disable demangler)
C++ mode compiler switch
[2-42](#)

-no-dollar (disable dollar format)
compiler switch [2-27](#)

-no-explicit (disable explicit
specifier) C++ mode compiler
switch [2-42](#)

-no-extra-keywords (not quite -ansi)
compiler switch [2-28](#)

-no-inline (disable inline keyword)
compiler switch [2-28](#)

-no-namespace (disable
namespaces) C++ mode
compiler switch [2-42](#), [2-43](#)

Non-Constant aggregate initializer
[2-68](#)

-no-newvec (disallow a new vector)
C++ mode compiler switch
[2-42](#)

Non-leaf routines [2-121](#)

norm (normalization) function
[3-125](#)

-no-std (disable std namespace) C++
mode compiler switch [2-43](#)

-no-std-def (disable standard
definitions) compiler switch
[2-28](#)

-no-std-inc (disable standard
include search) compiler switch
[2-29](#)

-no-std-lib (disable standard library
search) compiler switch [2-29](#)

Notation conventions for this
manual [1-6](#)

-nothreads (disable thread-safe
build) compiler switch [2-29](#)

-notstrict (non-strict compilation)
C++ mode compiler switch
[2-43](#)

NxN point 2-d complex input FFT
function [3-54](#)

INDEX

NxN point 2-d inverse input FFT
function [3-107](#)

NxN point 2-d real input FFT
function [3-133](#)

O

-O (enable standard optimization)
compiler switch [2-29](#)

-o (output) compiler switch [2-30](#)

-OO (disable standard optimization)
compiler switch [2-30](#)

Operators

bitwise [2-74](#)

comparison [2-75](#)

-Os (enable code size optimization)
compiler switch [2-30](#)

P

-P (omit #line and halt) compiler
switch [2-30](#)

-path-install (installation location)
compiler switch [2-31](#)

-path-output (non-temporary files
location) compiler switch [2-31](#)

-path-temp (temporary files
location) compiler switch [2-31](#)

-path-tool (tool location) compiler
switch [2-31](#)

-pch (precompiled header) compiler
switch [2-31](#)

-pchdir (locate PCHRepository)
compiler switch [2-32](#)

-pedantic (ANSI C warnings)
compiler switch [2-32](#)

-pedantic-errors (ANSI C errors)
compiler switch [2-32](#)

Placement support keyword
(section) [2-64](#)

pm (See Dual memory support
keywords (pm dm))

Pointer class support keyword
(restrict) [2-54](#), [2-65](#)

polar (construct from polar
coordinates) function [3-126](#)

pow (power) function [3-127](#)

-PP (omit #line and compile)
compiler switch [2-30](#)

-pplist (preprocessor listing)
compiler switch [2-32](#)

Prelinker [2-41](#), [2-51](#)

Preprocessing
a program [2-91](#)

Preprocessor
features [2-91](#)
generated warnings [2-87](#)

Preserved registers [2-110](#)

Primitive I/O functions [3-12](#)

Procedure call [2-102](#)
and return [2-106](#)

Prologue [2-109](#)

Q

qsort (quick sort) function [3-128](#)

Quad word [2-55](#), [2-57](#)

R

-R (search for source files) compiler switch [2-33](#)

rand (random number generator) function [3-130](#)

Real matrix - matrix subtraction function [3-116](#)

Real matrix - scalar subtraction function [3-118](#)

Real matrix * matrix multiplication function [3-115](#)

Real matrix * scalar multiplication function [3-117](#)

Real matrix + matrix addition function [3-114](#)

Real matrix + scalar addition function [3-113](#)

Real vector - scalar subtraction function [3-158](#)

Real vector - vector subtraction function [3-161](#)

Real vector * scalar multiplication function [3-157](#)

Real vector * vector multiplication function [3-160](#)

Real vector + scalar addition function [3-156](#)

Real vector + vector addition function [3-159](#)

Real vector dot product function [3-155](#)

Registers

ADSP-TS001 and ADSP-TS101 [2-110–2-116](#)

Bit FIFO Temporaries [2-115](#)

classification [2-110](#)

Compute Block (x & y) General [2-112](#)

Compute Block ALU Summation [2-114](#)

Compute Block X MAC [2-114](#)

Compute Block Y MAC [2-114](#)

Data Alignment [2-114](#)

iALU (j & k) Special registers:

Circular Buffering –B (base) & L (length) [2-113](#)

iALU (j and k) General [2-111](#)

preserved [2-110](#)

qualifier, __regclass [2-63](#)

Return Address [2-115](#)

save (scratch) [2-110](#)

Static Condition Flags [2-115](#)

Registers, usage (See Mixed

C/assembly programming)

-restrict (restrict) compiler switch [2-33](#)

restrict (See Pointer Class Support Keyword (restrict))

Return

from a procedure [2-108](#)

Return Address register [2-115](#)

Return values [2-105](#)

rfft (N point real input FFT) function [3-131](#)

rfft2d (NxN point 2-d real input FFT) function [3-133](#)

rms (root mean square) function [3-135](#)

INDEX

rsqrt (reciprocal square root function) function [3-136](#)

Running the Compiler [2-4](#)

Run-time environment (see Mixed C/assembly programming)

S

-S (stop after compilation) compiler switch [2-34](#)

-s (strip debugging information) compiler switch [2-34](#)

Save (scratch) registers [2-110](#)

-save-temps (save intermediate files) compiler switch [2-34](#)

Search

binary [3-46](#)

Search path

for include files [2-25](#)

for library files [2-26](#)

section (See Placement support keyword (section))

Set seed for random number generator function [3-141](#)

-show (display command line) compiler switch [2-34](#)

Sideways sum functions

(int2x16 values) [2-75](#)

(int4x16 values) [2-80](#)

Sign (sign) function [3-137](#)

signed keyword [2-19](#)

-signed-char (make char signed) compiler switch [2-34](#)

Simulator services [3-16](#)

sin (sine) function [3-138](#)

sinh (hyperbolic sine) function [3-139](#)

Sizes

data type [2-45](#)

Source code [2-35](#)

sourcefile compiler switch [2-19](#), [2-20](#)

Space

free [2-101](#)

Specifications

stacks [2-102](#)

sqr (square root) function [3-140](#)

srand (set seed for random number generator) function [3-141](#)

Stack

frame [2-97–2-110](#)

Stack frame

incoming arguments [2-100](#)

linkage information [2-100](#)

local variables/temporaries [2-100](#)

overview [2-98](#)

parts of [2-100](#)

specifications [2-102](#)

Standard C library

functions [3-8–3-14](#)

header files [3-8–3-14](#)

Static Condition Flags register [2-115](#)

Statistical functions

stats header file [3-19](#)

-std (std namespace) C++ mode compiler switch [2-43](#)

-strict (strict compilation) C++ mode compiler switch [2-43](#)

- strictwarn (warn if non-strict) C++ mode compiler switch [2-44](#)
- String to double conversion
 - atof function [3-39](#)
 - strtod function [3-142](#)
- String to integer conversion
 - atoi function [3-40](#)
 - strtoi function [3-144](#)
- String to long double conversion
 - atold function [3-42](#)
 - strtold function [3-148](#)
- String to long integer conversion [3-146](#)
- String to unsigned long integer conversion [3-149](#)
- strtod (string to double conversion) function [3-142](#)
- strtof (string to float conversion) function [3-143](#)
- strtoi (string to integer conversion) function [3-144](#)
- strtol (string to long integer conversion) function [3-146](#)
- strtold (string to long double conversion) function [3-148](#)
- strtoul (string to unsigned long integer conversion) function [3-149](#)
- Support
 - 64-bit integer (long long) [2-57](#)
 - boolean type keywords [2-65](#)
 - C++ fractional type [2-88](#)
 - customer [1-2](#)
 - dual memory keywords (pm, dm) [2-58](#)
 - indexed initializer [2-68](#)
 - inline function keyword [2-56](#)
 - non-constant aggregate initializer [2-68](#)
 - packed 16-bit integer [2-72](#)
 - placement keyword [2-64](#)
 - pointer class keyword (restrict) [2-65](#)
 - quad word [2-57](#)
 - variable length array [2-66](#)
- Switches (C++ mode)
 - alttok (alternative tokens) [2-40](#)
 - bool (accept boolean) [2-40](#)
 - explicit (explicit specifier) [2-41](#)
 - instant (instantiate all or used) [2-41](#)
 - namespace (enable namespaces) [2-41](#)
 - newforinit (limit scope) [2-41](#)
 - newvec (new vector) [2-41](#)
 - no-alttok (disable tokens) [2-42](#)
 - no-bool (disable boolean) [2-42](#)
 - no-demangle (disable demangler) [2-42](#)
 - no-explicit (disable explicit specifier) [2-42](#)
 - no-namespace (disable namespaces) [2-42](#)
 - no-newvec (disallow a new vector) [2-42](#)
 - no-std (disable std namespace) [2-43](#)
 - notstrict (non-strict compilation) [2-43](#)
 - no-wchar (disable wide char type) [2-43](#)
 - std (std namespace) [2-43](#)
 - strict (strict compilation) [2-43](#)
 - strictwarn (warn if non-strict) [2-44](#)

- tpautooff (no automatic templates) [2-44](#)
- trdforinit (traditional initialization) [2-44](#)
- typename (support typename) [2-44](#)
- wchar (support wide char type) [2-44](#)
- Switches (C/C++ common)
 - [2-24, 2-37](#)
 - A (assert) [2-20](#)
 - build-lib (build library) [2-20](#)
 - C (comments) [2-20](#)
 - c (compile only) [2-21](#)
 - D (define macro) [2-21](#)
 - default-linkage (assembler, C, or C++) [2-21](#)
 - dollar (dollar format) [2-21](#)
 - double-size-32 (IEEE single-precision double) [2-21](#)
 - double-size-64 (IEEE double-precision double) [2-21](#)
 - dry (verbose dry-run) [2-23](#)
 - dryrun (terse dry-run) [2-23](#)
 - E (stop after preprocessing) [2-23](#)
 - EE (run after preprocessing) [2-23](#)
 - extra-keywords (not quite -analog) [2-23](#)
 - flags (command line input) [2-24](#)
 - g (generate debug information) [2-24](#)
 - H (list headers) [2-24](#)
 - help (command-line help) [2-25](#)
 - HH (list *.h and compile) [2-25](#)
 - I (include search directory) [2-25](#)
 - include (include file) [2-25](#)
 - ipa (interprocedural analysis) [2-25](#)
 - L (library search directory) [2-26](#)
 - l (link library) [2-26](#)
 - M (make only) [2-26](#)
 - map (memory map) [2-27](#)
 - MM (make and compile) [2-26](#)
 - no-builtin (no built-in functions) [2-27](#)
 - no-defs (disable definitions) [2-27](#)
 - no-dollar (disable dollar format) [2-27](#)
 - no-extra-keywords [2-28](#)
 - no-inline (disable inline keyword) [2-28](#)
 - no-restrict (disable restrict keyword) [2-28](#)
 - no-std-def (disable standard macro definitions) [2-28](#)
 - no-std-inc (disable standard include search) [2-29](#)
 - no-std-lib (disable standard library search) [2-29](#)
 - nothreads (disable thread-safe build) [2-29](#)
 - O (enable optimization) [2-29](#)
 - o filename (output file) [2-30](#)
 - O0 (disable optimization) [2-30](#)
 - Os (enable code size optimizations) [2-30](#)
 - P (omit #line and halt) [2-30](#)
 - path (tool location) [2-31](#)
 - path-install directory (installation location) [2-31](#)
 - path-output (non-temporary files location) [2-31](#)

- path-temp directory (temporary files location) [2-31](#)
- pch (precompiled header) [2-31](#)
- pchdir (locate PCHRepository) [2-32](#)
- pedantic (C warnings) [2-32](#)
- pedantic-errors (ANSI C errors) [2-32](#)
- PP (omit #line and compile) [2-30](#)
- pplist (preprocessor listing) [2-32](#)
- R (search for source files) [2-33](#)
- restrict (restrict) [2-33](#)
- S (stop after compilation) [2-34](#)
- s (strip debugging information) [2-34](#)
- save-temps (save intermediate files) [2-34](#)
- show (display command line) [2-34](#)
- signed-char (make char signed) [2-34](#)
- sourcefile [2-19](#), [2-20](#)
- syntax-only (check syntax only) [2-35](#)
- sysdef (system definitions) [2-35](#)
- T filename (linker description file) [2-35](#)
- threads (enable thread-safe build) [2-36](#)
- time (tell time) switch [2-36](#)
- TS001 (ADSP-TS001) [2-36](#)
- TS101 (ADSP-TS101) [2-36](#)
- U (undefine macro) [2-36](#)
- unsigned-char (make char unsigned) [2-37](#)
- v (version & verbose) [2-37](#)
- verbose (command line) [2-37](#)
- version (display version) [2-37](#)
- w (disable all warnings) [2-38](#)
- Wdriver-limit (maximum process errors) [2-38](#)
- Werror-limit (maximum compiler errors) [2-38](#)
- Werror-limit (maximum compiler errors) [2-38](#)
- Wremarks (enable diagnostic warnings) [2-38](#)
- write-files (enable driver I/O redirection) [2-39](#)
- Wterse (terse warnings) [2-38](#)
- xref (cross-reference list) [2-39](#)
- Switches (mode selection)
 - analog (Analog C compilation) [2-18](#)
 - ansi (ANSI standard C compilation) [2-18](#)
 - c++ (C++ mode) [2-19](#)
 - traditional (traditional C compilation) [2-19](#)
- syntax-only (just check syntax) compiler switch [2-35](#)
- sysdef (system definitions) compiler switch [2-35](#)
- System header files [2-94](#)
- T
 - T (linker description file) compiler switch [2-35](#)
- tan (tangent) function [3-150](#)
- tanh (hyperbolic tangent) function [3-151](#)
- Technical support [1-2](#)
- Templates
 - instantiate [2-41](#)

Temporary files [2-34](#)

-threads (enable thread-safe build)

 compiler switch [2-36](#)

-time (tell time) switch [2-36](#)

-tpautooff (no automatic templates)

 C++ mode compiler switch [2-44](#)

-traditional (traditional C compilation)

 compiler switch [2-19](#)

-trdforinit (traditional initialization)

 C++ mode compiler switch [2-44](#)

true (See Boolean Type Support

 Keywords (bool, true, false))

-TS001 (ADSP-TS001) compiler

 switch [2-36](#)

-TS101 (ADSP-TS101) compiler

 switch [2-36](#)

twidfft (generate FFT twiddle factors)

 function [3-152](#)

Two dimensional FFT [3-54](#)

-typename (support typename) C++

 mode compiler switch [2-44](#)

U

-U (undefine macro) compiler switch

[2-21](#), [2-36](#)

-unsigned-char (make char unsigned)

 compiler switch [2-37](#)

V

-v (version & verbose) compiler switch

[2-37](#)

var (variance) function [3-154](#)

Variable length array [2-66](#)

Variables

 automatic [2-58](#)

vecdot (real vector dot product)

 function [3-155](#)

vecsadd (real vector + scalar addition)

 function [3-156](#)

vecsmult (real vector * scalar

 multiplication) function [3-157](#)

vecsub (real vector - scalar subtraction)

 function [3-158](#)

Vector functions [3-19](#)

vecvadd (real vector + vector addition)

 function [3-159](#)

vecvmult (real vector * vector

 multiplication) function [3-160](#)

vecvsub (real vector - vector subtraction)

 function [3-161](#)

-verbose (display command line)

 compiler switch [2-37](#)

-version (display version) compiler

 switch [2-37](#)

VisualDSP++ compiler

 extensions [2-54](#)

 overview [2-1](#)

volatile keyword [2-19](#)

W

-w (disable all warnings) compiler

 switch [2-38](#)

-W (override error) compiler switch

[2-37](#)

-wchar (support wide char type) C++

- Wdriver-limit (maximum process errors) compiler switch [2-38](#)
- Werror-limit (maximum compiler errors) compiler switch [2-38](#)
- Wremarks (enable diagnostic warnings) compiler switch [2-38](#)
- write-files (enable driver I/O pipe) compiler switch [2-39](#)
- Wterse (enable terse warnings) compiler switch [2-38](#)

X

- xref (cross-reference list) compiler switch [2-39](#)

Z

- zero_cross (count zero crossing) function [3-162](#)

