

VISUALDSP++TM 2.0

C/C++ Compiler and Library

Manual for ADSP-21xxx DSPs

Third Revision, September, 2001

Part Number
82-001963-01

Analog Devices, Inc.
Digital Signal Processor Division
One Technology Way
Norwood, Mass. 02062-9106



Copyright Information

©1996–2001 Analog Devices, Inc., ALL RIGHTS RESERVED. This document may not be reproduced in any form without prior, express written consent from Analog Devices, Inc.

Printed in the USA.

Disclaimer

Analog Devices, Inc. reserves the right to change this product without prior notice. Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use; nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under the patent rights of Analog Devices, Inc.

Trademark and Service Mark Notice

The Analog Devices logo, VisualDSP, the VisualDSP logo, SHARC, and the SHARC logo are registered trademarks; VisualDSP++, the VisualDSP++ logo, and EZ-KIT Lite are trademarks of Analog Devices, Inc.

Adobe and Acrobat are registered trademarks of Adobe Corporation.

Intel is registered trademark of Intel Corporation.

Microsoft and Windows are registered trademarks and Windows NT is a trademark of Microsoft Corporation.

All other brand and product names are trademarks or service marks of their respective owners.

CONTENTS

INTRODUCTION

Supported Processors	1-2
For Additional Product Information	1-2
For Technical or Customer Support	1-3
Purpose of This Manual	1-3
Intended Audience	1-4
Manual Contents Description	1-4
What's New in This Manual	1-5
Related Documents	1-5
Conventions	1-7

COMPILER

Overview	2-1
Compiler Command-Line Interface	2-3
Running the Compiler	2-4
C/C++ Compiler Switches	2-8
C/C++ Compiler Switch Summaries	2-8
C/C++ Mode Selection Switch Descriptions.....	2-18

CONTENTS

C/C++ Compiler Common Switch Descriptions	2-19
C++ Mode Compiler Switch Descriptions	2-43
Data Type Sizes	2-47
Integer	2-48
Floating Point.....	2-48
Optimization Control	2-49
Inlining Control	2-51
Interprocedural Analysis	2-51
C/C++ Compiler Language Extensions	2-54
Inline Function Support Keyword (inline)	2-57
Inline Assembly Language Support Keyword (asm)	2-58
Assembly Construct Template	2-59
Assembly Construct Operand Description	2-62
Assembly Constructs With Multiple Instructions.....	2-64
Assembly Construct Reordering and Optimization	2-65
Restrictions on the Use of the asm Construct	2-65
Assembly Constructs with Input and Output Operands	2-66
Assembly Constructs and Macros	2-67
Dual Memory Support Keywords (pm dm)	2-68
Memory Keywords and Assignments/Type Conversions	2-70
Memory Keywords and Function Declarations/Pointers	2-71
Memory Keywords and Function Arguments	2-72
Memory Keywords and Macros	2-73
Placement Support Keyword (section)	2-73

Boolean Type Support Keywords (bool, true, false)	2-74
Pointer Class Support Keyword (restrict)	2-74
Variable-Length Array Support	2-75
Non-Constant Initializer Support	2-77
Indexed Initializer Support	2-77
Aggregate Constructor Expression Support	2-79
Preprocessor Generated Warnings	2-80
C++ Style Comments	2-80
Compiler intrinsic Functions	2-81
Access to System Registers	2-82
C++ Fractional Type Support	2-84
Format of Fractional Literals	2-85
Conversions Involving Fractional Values	2-85
Fractional Arithmetic Operations	2-86
Mixed Mode Operations	2-87
Saturated Arithmetic	2-87
SIMD Support Annotation (#pragma SIMD_for)	2-88
Using SIMD Mode with Multichannel Data	2-89
Using SIMD Mode with Single Channel Data	2-90
Pitfalls in Using SIMD C/C++	2-92
SIMD_for Syntax.....	2-93
Constraints on Using SIMD C/C++	2-94
Impact of Anomaly #40 on SIMD	2-95
Examples Using SIMD C (Problem Cases—Data Increments)	2-96

CONTENTS

Examples Using SIMD C (Problem Cases—Data Alignment)	2-98
Performance When Using SIMD C/C++	2-100
Preprocessing a Program	2-102
Predefined Macros	2-104
Header Files	2-109
Writing Macros	2-109
Support for Multiple Heaps	2-111
Heap Identifiers	2-112
Using Alternate Heaps with the Standard Interface	2-113
Using the Alternate Heap Interface	2-113
Re-initializing Heaps	2-114
Creating Heap Descriptor Records	2-115
Allocating Heap Storage Areas	2-116
Initializing Heaps	2-117
Example C program	2-118
C/C++ Run-Time Model	2-121
C/C++ Run-Time Environment	2-122
Memory Usage	2-123
Compiler Registers	2-129
User Registers	2-130
Call Preserved Registers	2-131
Scratch Registers.....	2-132
Stack Registers.....	2-133
Alternate Registers	2-134

Managing the Stack.....	2-135
Transferring Function Arguments and Return Value.....	2-140
Using Data Storage Formats	2-143
Using the Run-Time Header	2-147
C/C++ and Assembly Interface	2-147
Calling Assembly Language Subroutines from C/C++ Programs	2-148
Calling C/C++ Functions from Assembly Language Programs	2-150
Using Mixed C/C++ and Assembly Support Macros	2-153
Interface Support Macros, Defined	2-153
Using Mixed C/C++ and Assembly Naming Conventions..	2-157
Implementing C++ Member Functions in Assembly	2-159
Writing C/C++ Callable SIMD Subroutines.....	2-161
C++ Programming Examples	2-162
Using Fract Support	2-163
Using Complex Support	2-163
Mixed C/C++/Assembly Programming Examples	2-165
Using Inline Assembly (Add)	2-166
Using Macros to Manage the Stack	2-167
Using Scratch Registers (Dot Product)	2-168
Using Void Functions (Delay)	2-170
Using the Stack for Arguments and Return (Add 5)	2-171
Using Registers for Arguments and Return (Add 2)	2-172

CONTENTS

Using Non-leaf Routines That Make Calls (RMS)	2-173
Using Call Preserved Registers (Pass Array)	2-175
C/C++ Compiler Glossary	2-177
C/C++ RUN-TIME LIBRARY	
Overview	3-1
C and C++ Run-Time Libraries Guide	3-3
Calling Library Functions	3-3
Linking Library Functions	3-4
Working with Library Header Files	3-6
Standard C Library Header Files	3-7
Standard C Library Header File Descriptions.....	3-8
Using the Compiler's Built-In C library Functions	3-19
Abridged C++ Library Support	3-21
Embedded C++ Library Header Files.....	3-22
C++ Header Files for C Library Facilities.....	3-25
Embedded Standard Template Library Header Files	3-26
C Run-Time Library Reference	3-29
abort	3-30
abs	3-31
acos, acosf	3-32
asin, asinf	3-33
atan, atanf	3-34
atan2, atan2f	3-35
atexit.....	3-36

atof.....	3-37
atoi.....	3-38
atol.....	3-39
avg	3-40
bsearch	3-41
calloc.....	3-43
ceil, ceilf.....	3-45
clear_interrupt.....	3-46
clip.....	3-51
cos, cosf.....	3-52
cosh, coshf.....	3-53
div.....	3-54
exit.....	3-55
exp, expf.....	3-56
fabs, fabsf	3-57
floor, floorf.....	3-58
fmod, fmodf	3-59
free.....	3-60
frexp, frexpf.....	3-61
getenv.....	3-62
heap_calloc.....	3-63
heap_free	3-65
heap_init	3-67
heap_lookup	3-69

CONTENTS

heap_malloc	3-71
heap_realloc	3-73
heap_switch	3-76
interrupt	3-79
isalnum	3-81
isalpha.....	3-82
isctrl.....	3-83
isdigit.....	3-84
isgraph	3-85
islower.....	3-86
isprint	3-87
ispunct	3-88
isspace.....	3-89
isupper	3-90
isxdigit	3-91
labs	3-92
lavg	3-93
lclip	3-94
ldexp, ldexpf.....	3-95
ldiv	3-96
lmax.....	3-97
lmin	3-98
localeconv	3-99
log, logf.....	3-103

log10, log10f	3-104
longjmp	3-105
malloc	3-107
max	3-108
memchr	3-109
memcmp	3-110
memcpy	3-111
memmove	3-112
memset	3-113
min	3-114
modf, modff	3-115
pow, powf	3-116
qsort	3-117
raise	3-119
rand	3-120
realloc	3-121
setjmp	3-123
setlocale	3-124
signal	3-125
sin, sinf	3-127
sinh, sinh	3-128
sqrt, sqrtf	3-129
srand	3-130
strcat	3-131

CONTENTS

strchr	3-132
strcmp	3-133
strcoll	3-134
strcpy	3-135
strcspn	3-136
strerror	3-137
strlen	3-138
strncat	3-139
strncmp	3-140
strncpy	3-141
strpbrk	3-142
strrchr	3-143
strspn	3-144
strstr	3-145
strtod	3-146
strtok	3-148
strtol	3-150
strtoul	3-152
strxfrm	3-154
system	3-156
tan, tanf	3-157
tanh, tanhf	3-158
tolower	3-159
toupper	3-160

va_arg.....	3-161
va_end.....	3-163
va_start.....	3-164

DSP LIBRARY FOR ADSP-2106X PROCESSORS

Overview	4-1
DSP Run-Time Library Guide	4-2
Linking DSP Library Functions	4-3
Working With Library Source Code	4-3
DSP Header Files	4-4
Built-In DSP Functions	4-11
DSP Run-Time Library Reference	4-13
a_compress	4-14
a_expand	4-15
autocoh	4-16
autocorr.....	4-17
biquad	4-18
cabsf.....	4-21
cexpf.....	4-22
cfftN	4-23
copysign, copysignf.....	4-26
cot, cotf.....	4-27
crosscoh.....	4-28
crosscorr	4-29
favg, favgf.....	4-30

CONTENTS

fclip, fclipf.....	4-31
fir	4-32
fmax, fmaxf	4-34
fmin, fminf	4-35
gen_bartlett.....	4-36
gen_blackman	4-38
gen_gaussian	4-39
gen_hamming	4-40
gen_hanning	4-41
gen_harris	4-42
gen_kaiser	4-43
gen_rectangular	4-45
gen_triangle	4-46
histogram	4-48
idle	4-50
ifftN	4-51
iir.....	4-54
matadd.....	4-57
matmul	4-58
matscaltmult.....	4-60
matsub	4-61
mean	4-62
mu_compress	4-63
mu_expand	4-64

<code>poll_flag_in</code>	4-65
<code>rfftN</code>	4-67
<code>rms</code>	4-70
<code>rsqrt, rsqrtf</code>	4-71
<code>set_flag</code>	4-72
<code>set_semaphore</code>	4-74
<code>timer_off</code>	4-75
<code>timer0_off, timer1_off</code>	4-76
<code>timer_on</code>	4-77
<code>timer0_on, timer1_on</code>	4-78
<code>timer_set</code>	4-79
<code>timer0_set, timer1_set</code>	4-81
<code>var</code>	4-83
<code>zero_cross</code>	4-84

DSP LIBRARY FOR ADSP-2116X PROCESSORS

Overview	5-1
DSP Run-Time Library Guide	5-2
Linking DSP Library Functions	5-3
Working With Library Source Code	5-3
DSP Header Files	5-4
Built-In DSP Functions	5-12
Pitfalls Using SIMD Mode	5-13
DSP Run-Time Library Reference	5-15
a_compress	5-16

CONTENTS

a_expand	5-17
autocoh	5-18
autocorr	5-19
cabsf	5-20
cexpf	5-21
cfftN	5-22
copysign, copysignf	5-25
cot, cotf	5-26
crosscoh	5-27
crosscorr	5-28
cvecdot	5-29
cvecsadd	5-30
cvecsmult	5-31
cvecssub	5-32
cvecvadd	5-33
cvecvmlt	5-34
cvecvsub	5-35
favg, favgf	5-36
fclip, fclipf	5-37
fft_mag	5-38
fir	5-39
fmax, fmaxf	5-41
fmin, fminf	5-42
gen_bartlett	5-43

gen_blackman.....	5-45
gen_gaussian.....	5-46
gen_hamming.....	5-47
gen_hanning.....	5-48
gen_harris.....	5-49
gen_kaiser.....	5-50
gen_rectangular	5-51
gen_triangle.....	5-52
histogram	5-54
idle.....	5-56
ifftN.....	5-57
iir	5-60
matadd	5-62
matmul.....	5-63
matscaltmult	5-64
matsub	5-65
mean	5-66
mu_compress.....	5-67
mu_expand.....	5-68
poll_flag_in	5-69
rfftN.....	5-71
rfft2_N.....	5-74
rms.....	5-77
rsqrt, rsqrtf	5-78

CONTENTS

set_flag.....	5-79
set_semaphore	5-81
timer_off.....	5-82
timer_on	5-83
timer_set	5-84
var	5-85
vecdot	5-86
vecsadd	5-87
vecsmult.....	5-88
vecssub.....	5-89
vecvadd	5-90
vecvmlt	5-91
vecvsub	5-92
zero_cross.....	5-93

INDEX