

VisualDSP++ 4.5 (Updated February 2007) Release Note

The following release note concerns the February 2007 Update to the VisualDSP++ 4.5 release. This release is inclusive of previous Updates. The contents of future Updates will be inclusive of all previous Updates. The release notes for past Updates are appended to the end of this release note.

Identifying Which Update Is Currently Installed on Your System

The Update level is identified in three places:

1. The Add/Remove Programs Control Panel entry for VisualDSP++ 4.5.
2. The VisualDSP++ GUI's About box, located at "Help" > "About VisualDSP++".
3. In the file ...\\System\\VisualDSP.ini, in the ProductName key.

Installing an Update

The procedure for installing an Update to VisualDSP++ is described below. Note that with VisualDSP++'s support for installing multiple instances of itself, it is possible to "trial" an Update in a new directory before switching over your "golden" tools installation to the Update.

1. Use the Start Menu to navigate to VisualDSP++'s "Maintain this Installation" item.
2. Select "Go to the Analog Devices website" and click "Next". This will launch your web browser and navigate it to the proper URL to download Updates from.
3. Download the VisualDSP++ Update file (.VDU) of interest to your hard disk. Note that these files have a .VDU file extension and cannot be executed directly.
4. Navigate to "Maintain this Installation" again. If you have multiple installations of VisualDSP++ on your computer, be doubly sure you are navigating to the installation you wish to Update.
5. Select "Apply a downloaded Update" and click "Next". Click the "..." browser button and navigate to the .VDU file that you downloaded in step 3. Click "OK", then "Next".
6. Follow the on-screen prompts to complete the installation of the Update.

Significant Additions

The primary purpose of VisualDSP++ Updates is to address problems and stabilize the release. Significant new functionality is not expected to be introduced in an Update. However, incremental support (i.e., emulation, example programs, header files, default LDF, errata accommodations, EZ-KIT Lite software, etc.) for new semiconductor products will be added as these products become available and gain support within the VisualDSP++ tools.

In this release:

1. Emulation support for the ADSP-BF54x family has been added.
2. Support for ADSP-BF538/9 revision 0.4 has been added.
3. The Blackfin compiler, runtime, VDK, and SSL libraries include new workarounds for hardware anomalies 05-00-0311 and 05-00-0312. The C/C++ compiler will automatically enable these workarounds when building for parts and silicon revisions that require them. Alternatively they can be enabled using the `-workaround` switch. The Blackfin assembler has a new warning to indicate potential instances of anomaly 05-00-0312.

05-00-0311 –

The anomaly is seen when an access of a System MMR Flag register is followed by an access of a specific MMR. The result of the anomaly can be that flag pins configured as outputs that are "set"

can erroneously transition to "clear". The anomaly impacts all revisions of ADSP-BF53[123] and ADSP-BF561 parts.

The compiler works around the anomaly:

1. By adding a load of the CHIPID MMR at the start of code generated for C defined event handlers. System Services' handlers and VDK's interrupt templates make similar accommodations. The related workaround for 05-00-0283 will be used if it is enabled at the same time as this workaround.
2. By identifying accesses, or potential accesses, of any of the various System MMR Flag registers associated with the errata and generating a load of CHIPID after each such access. The compiler relies on use of literal addressing of MMRs to identify these accesses, such as using the various `p<MMR>` macros defined in the `cdef<PART>.h` include files. Any loads and stores that do not use literal addresses and the type of the access is defined qualified as `volatile`, will be assumed to be flag MMR accesses, unless the `-no-assume-vols-are-mmrs` switch is used. If the `-no-assume-vols-are-mmrs` compiler switch is used the compiler will apply the workaround for suitable literal address accesses only and will make no assumptions for non-literal loads and stores even if `volatile`.

To enable this compiler workaround manually the `-workaround flags-mmr-anom-311` switch can be used. When the workaround is enabled the macro `__WORKAROUND_FLAGS_MMR_ANOM_311` is defined at compile, assemble and link stages.

05-00-0312 -

The anomaly is seen when SSYNC, CSYNC instructions or loads of registers LCx, LTx, and LBx are interrupted. The anomaly impacts all Blackfin parts and revisions except ADSP-BF535.

The compiler workarounds are as follows:

1. When the workaround is enabled the compiler builtin functions `__builtin_ssync()` and `__builtin_csync()` have been modified to ensure that interrupts are disabled before the `sync/csync` instruction and enabled after.
2. New `ssync` and `csync` builtins have been provided that do not disable interrupts. These can be used in place of the existing builtins for code that has been manually verified as safe against the anomaly. These new builtins are called `__builtin_ssync_int()` and `__builtin_csync_int()`.
3. When the workaround is enabled the compiler will ensure that any loads of the LBx, LTx and LCx registers are executed with interrupts disabled. It does this by inserting a CLI instruction before such loads and an STI instruction after. The workaround is not required for hardware loop LSETUP instructions.
4. The compiler will arrange to save and restore the loop registers (LBx, LTx and LCx) while interrupts are disabled for C/C++ nested interrupt handlers defined using the `sys/exception.h EX_REENTRANT_HANDLER` macro.

The assembler has been modified to include a warning which when enabled will identify potential causes of the anomaly. This warning can be enabled using the `-anomaly-warn 05-00-0312` switch. The warning can be suppressed in assembly code ranges which have manually been determined to be safe against the anomaly using the assembler's `.MESSAGE` directive. The warning id to use when this is required is 5515.

To enable this compiler workaround manually the `-workaround sync-loop-anom-312` switch can be used. When the workaround is enabled the macro `__WORKAROUND_SYNC_LOOP_ANOM_312` is defined at compile, assemble and link stages.

For more information on these anomaly please see the appropriate errata sheets which can be downloaded from <http://www.analog.com/processors/blackfin/support/ICanomalies.html>.

Changes to Existing Behaviors, Projects, and Source Code

When addressing problems, we attempt to make any changes backward compatible with existing projects. However, depending on the nature of a problem, compatibility issues are sometimes unavoidable. This section highlights any changes in the Update that may require the modification of “working” projects or otherwise influence existing behavior.

In this release, no changes have been identified.

Problems Addressed

The following table is a list of the problems addressed in this Update. Details on any particular problem can be found on the Tools Anomaly web page. The URL is:

<http://www.analog.com/processors/technicalSupport/toolsAnomalies.html>

Processor Family	Problem Number	Tool	Description
All	17307	Utilities	MEMINIT implementation makes part of .map file useless
All	27177	Compiler	Adding start up code to project breaks build
All	29624	Compiler	-pch -eh crashes compiler - unoptimised and optimised
All	29838	VDK	Destroying a periodic semaphore can damage the time queue
All	30048	VDK	UserHistoryLog called incorrectly during context switch
All	30181	Compiler	Register locked too many times
All	30351	Compiler	assoc_source_line_modif: bad address
Blackfin	29221	Run Time Libraries	multicore runtime libraries always link in I/O library
Blackfin	29552	Emulator	Remove setting PF12 on BF533 USB example
Blackfin	29717	Run Time Libraries	rms_fr16 can return a negative result
Blackfin	29833	Loader	Page needs to support SPI Slave for 561 Rev 0.5
Blackfin	29834	Compiler	Internal error "Cannot locate Reg to be Released" at call
Blackfin	29841	CRTGen	BF52x proj with start up code & Ldf do not build
Blackfin	29917	VDK	VDK exception handler could corrupt P1
Blackfin	29955	Run Time Libraries	Link error with multi-threaded C library and setlocale
Blackfin	30049	Run Time Libraries	C++ streams broken for multicore applications
Blackfin	30112	LDFGen	I/O from both cores of multi-core app fails with cache enabled
Blackfin	30118	Emulator	cache not flushed when emulator performs a write of 1
Blackfin	30164	Emulator	External memory read error at address x200FFFFFF - x20100000
Blackfin	30220	Linker	Linker miscalculates program size for EZ-KIT restriction
SHARC	29236	IDDE	Server License expires 'n' days after validation when compiling
SHARC	29726	VDK	Sharc thread level context switches not interrupt safe
SHARC	29742	Emulator	Usage of Parallel port interrupt restricted
SHARC	29761	Simulator	2136[7,8,9] sysctl incorrect
SHARC	29819	LDF	LDFs contain unnecessary macros to deal with EZ-kit monitors
SHARC	29828	Run Time Libraries	ADSP-2136x CRT need support for new LX3 interrupts
SHARC	29829	Compiler	circindex returns wrong index when constant 0 used fails -Og
SHARC	29856	Run Time Libraries	213xx\lib\2136x_any\libcppeh37x.dlb should be libcpp37xeh.dlb
SHARC	29898	Run Time Libraries	The ifftf() function can generate wrong results
SHARC	29916	Assembler	Hex init < 15 digits for LW space is left-justified with no warn
SHARC	30123	Loader	Loader does not convert PROM non-zero address from Hex to S3
SHARC	30126	Loader	Loader fails to add 0x00000000 to an odd SPI 48-bit block
SHARC	30410	Assembler	Check for anomaly 08000002 absent for 21368/9
SHARC	30411	Assembler	Check for anomaly 07000010 absent for 21362, 21366
SHARC	30415	Assembler	Assembler not checking for anomaly 08000001 on 21367/8/9
TigerSHARC	29792	Run Time	strtod may convert floating-point hex constants incorrectly

		Libraries	
TigerSHARC	29801	Run Time Libraries	Link Buffer Registers macros defined incorrectly in defts201.h
TigerSHARC	29830	Run Time Libraries	Floating point comparison routine clobbers an unexpected reg
TigerSHARC	30029	Run Time Libraries	The conv2d function can generate wrong results

VisualDSP++ 4.5 (Updated November 2006) Release Note

The following release note concerns the November 2006 Update to the VisualDSP++ 4.5 release. This release is inclusive of previous Updates. The contents of future Updates will be inclusive of all previous Updates. The release notes for past Updates are appended to the end of this release note.

Identifying Which Update Is Currently Installed on Your System

The Update level is identified in three places:

1. The Add/Remove Programs Control Panel entry for VisualDSP++ 4.5.
2. The VisualDSP++ GUI's About box, located at "Help" > "About VisualDSP++".
3. In the file ...\\System\\VisualDSP.ini, in the ProductName key.

Installing an Update

The procedure for installing an Update to VisualDSP++ is described below. Note that with VisualDSP++'s support for installing multiple instances of itself, it is possible to "trial" an Update in a new directory before switching over your "golden" tools installation to the Update.

1. Use the Start Menu to navigate to VisualDSP++'s "Maintain this Installation" item.
2. Select "Go to the Analog Devices website" and click "Next". This will launch your web browser and navigate it to the proper URL to download Updates from.
3. Download the VisualDSP++ Update file (.VDU) of interest to your hard disk. Note that these files have a .VDU file extension and cannot be executed directly.
4. Navigate to "Maintain this Installation" again. If you have multiple installations of VisualDSP++ on your computer, be doubly sure you are navigating to the installation you wish to Update.
5. Select "Apply a downloaded Update" and click "Next". Click the "... " browser button and navigate to the .VDU file that you downloaded in step 3. Click "OK", then "Next".
6. Follow the on-screen prompts to complete the installation of the Update.

Significant Additions

The primary purpose of VisualDSP++ Updates is to address problems and stabilize the release. Significant new functionality is not expected to be introduced in an Update. However, incremental support (i.e., emulation, example programs, header files, default LDF, errata accommodations, EZ-KIT Lite software, etc.) for new semiconductor products will be added as these products become available and gain support within the VisualDSP++ tools.

In this release:

1. Compiler, assembler, and linker support for a new Blackfin processor family, the ADSP-BF54x, has been added. The loader and System Services are not yet available and will be added in a future release.
2. This Update has been broadly tested under Windows Vista Beta 2 (5308) and is believed to be fully operational. While the base release of VisualDSP++ 4.5 installs successfully, changes were required to avoid run-time operational problems. Versions of VisualDSP++ prior to this Update will generally not be operational under Vista and are not supported. It is possible that the final "gold" release of Window Vista will change and require further (minor) modification of VisualDSP++. Future Update release notes will discuss this topic as required.

Note that, at this time, the emulator and EZ-KIT Lite Windows device drivers are not digitally signed. Windows will pop up a warning message for each driver installed. Confirm the installation of each driver during base and/or Update installation.

Changes to Existing Behaviors, Projects, and Source Code

When addressing problems, we attempt to make any changes backward compatible with existing projects. However, depending on the nature of a problem, compatibility issues are sometimes unavoidable. This section highlights any changes in the Update that may require the modification of “working” projects or otherwise influence existing behavior.

In this release:

1. The VDK libraries included in this update now flag an additional value used as a timeout in Pend APIs (PendSemaphore, PendMessage, PendEvent, PendDeviceFlag) as being invalid. Passing the value (0 | kNoTimeoutError) as the timeout will now result in a kInvalidTimeout error. In previous versions of VDK the value was silently accepted. However, specifying that no error should be dispatched in the event of a timeout, but also that the Pend call should never timeout, has never been a useful thing to do. It was also ambiguous as to what the result would be. The issuing of an error in this case will now draw attention to this fact.
2. For Blackfin processors, the assembler keyword LOOP was incorrectly also defined as a macro in Blackfin header files (for UART loopback enablement). This has been corrected in this Update. Any code that uses LOOP in its macro form may now fail to build and must be updated to use LOOP_ENA instead.
1. For Blackfin processors, the user interface management of the -pFlag parameter to the loader has been revised. It is now dynamically managed and varies with processor, silicon revision, boot mode, and width. The loader now generates new warnings for illegal combinations.

After application of this Update, any Blackfin project using -pFlag should be verified to ensure that the correct setting is being used. This setting should also be verified whenever the processor, silicon revision, boot mode, or width is changed in VisualDSP++.

The following tables show the allowed values for -pFlag:

ADSP-BF531/2/3

Revision	0.0-0.2		0.3-0.5	
Width	8	16	8	16
Flash	NONE	NONE	NONE	NONE
SPI	NONE		NONE	
SPI slave			1-15 PF1-15	

ADSP-BF538/9

Revision	0.0-0.3	
Width	8	16
Flash	NONE	NONE
SPI	NONE	
SPI slave	1-15 PF1-15	

ADSP-BF534/6/7

Revision	0.0		0.1-0.2		0.3	
Width	8	16	8	16	8	16
Flash	NONE	NONE	NONE PF0-15 PG0-15 PH0-15	NONE PF0-15 PG0-15 PH0-15	NONE PF0-15 PG0-15 PH0-15	NONE PF0-15 PG0-15 PH0-15
SPI	NONE		NONE PF0-9 PF15 PG0-15 PH0-15		NONE PF0-9 PF15 PG0-15 PH0-15	
SPI slave	1-15 PF1-15		NONE PF0-10 PF15 PG0-15 PH0-15		NONE PF0-10 PF15 PG0-15 PH0-15	
TWI	NONE		NONE PF0-15 PG0-15 PH0-15		NONE PF0-15 PG0-15 PH0-15	
TWI slave	NONE		NONE PF0-15 PG0-15 PH0-15		NONE PF0-15 PG0-15 PH0-15	
UART	2-15 PF2-15		NONE PF2-15 PG0-15 PH0-15		NONE PF2-15 PG0-15 PH0-15	
FIFO						NONE PF0 PF2-15 PG0-15 PH0-15

Notes:

1. Blank fields indicate not supported boot modes.
2. BF533/4/6/7/8/9 always has the RESVECT bit (bit #2 in block header flag word) set.
3. BF531/2 have always RESVECT bit (bit #2 in block header flag word) cleared.
4. VisualDSP++ property page provides a “NONE” option in the pflag pull-down menu. When chosen, no -pflag switch goes to the command line and the PPORT and PFLAG fields in the block header flag word are zero.

Problems Addressed

The following table is a list of the problems addressed in this Update. Details on any particular problem can be found on the Tools Anomaly web page. The URL is:

<http://www.analog.com/processors/technicalSupport/toolsAnomalies.html>

Processor Family	Problem Number	Tool	Description
All	23803	IDDE	Floating License expires warning with permanent license string
All	29233	Run Time Libraries	Wrong file positions may be returned for a text stream
All	29234	Run Time Libraries	scanf does not input data of the form .d{dddd}edd correctly
All	29576	Compiler	circindex returns wrong index when constant 0 used
Blackfin	28080	Linker	False -si-rev incompatibilities when build from command line
Blackfin	29070	Simulator	DMA Descriptor Array uses NXT_DESC not CURR_DESC
Blackfin	29112	Debug Agent	cache flush functions fail on blackfin ez-kits
Blackfin	29163	Config	frequency selection in Ictest doesn't work with HPUSB
Blackfin	29164	Config	Ictest scan test fails on MP targets using HPUSB/USB
Blackfin	29235	Run Time Libraries	signal() will never return SIG_IGN
Blackfin	29258	Compiler	P0 clobbered in -ieee-fp compiler support function
Blackfin	29308	LDFGen	Custom SDRAM setting can make the LDF unusable
Blackfin	29320	System Services	printf before and after adi_pwr_Init causes lockup in Rel on 538
Blackfin	29352	Run Time Libraries	si-revision 'any' builds result in linker warning li2152
Blackfin	29363	Run Time Libraries	defBFXXX.h files #define LOOP, keyword in Blackfin assembly
Blackfin	29434	Flash Program	0 sectors in a flash programmer drivers crashes IDDE
Blackfin	29513	System Services	adi_pwr_SetFreq() disables CLKBUFOE bit in VR_CTL
Blackfin	29524	Assembler	Assembler rejects source file name without .suffix
Blackfin	29648	CRTGen	Customising clock settings can cause the CRT not to compile
Blackfin	29652	Compiler	circptr builtin may fail especially without -O
Blackfin	29662	Run Time Libraries	__var_wa_06000047 undefined when linking for BF561 rev 0.5
Blackfin	29672	Simulator	Signed/Unsigned Integer 8-bit format not properly handled
SHARC	29059	Loader	change in PLL setup in example codes
SHARC	29236	IDDE	Server License expires 'n' days after validation when compiling
SHARC	29267	Compiler	two more cases that can lead to anomaly #4 for ADSP-2126x DSPs
SHARC	29451	Run Time Libraries	The FFTs from filter.h may loop forever
SHARC	29518	Run Time Libraries	Interrupt handler does not restore multiplier regs properly
SHARC	29602	Run Time Libraries	Some interrupt handlers are called with an incorrect parameter
SHARC	29706	Compiler	2nd switch statement not reached
TigerSHARC	29382	VDK	"Run Total Time (cycles)" calculation is incorrect

VisualDSP++ 4.5 (Updated September 2006) Release Note

The following release note concerns the September 2006 Update to the VisualDSP++ 4.5 release. This release is inclusive of previous Updates. The contents of future Updates will be inclusive of all previous Updates. The release notes for past Updates are appended to the end of this release note.

Identifying Which Update Is Currently Installed on Your System

The Update level is identified in three places:

1. The Add/Remove Programs Control Panel entry for VisualDSP++ 4.5.
2. The VisualDSP++ GUI's About box, located at "Help" > "About VisualDSP++".
3. In the file ...\\System\\VisualDSP.ini, in the ProductName key.

Installing an Update

The procedure for installing an Update to VisualDSP++ is described below. Note that with VisualDSP++'s support for installing multiple instances of itself, it is possible to "trial" an Update in a new directory before switching over your "golden" tools installation to the Update.

1. Use the Start Menu to navigate to VisualDSP++'s "Maintain this Installation" item.
2. Select "Go to the Analog Devices website" and click "Next". This will launch your web browser and navigate it to the proper URL to download Updates from.
3. Download the VisualDSP++ Update file (.VDU) of interest to your hard disk. Note that these files have a .VDU file extension and cannot be executed directly.
4. Navigate to "Maintain this Installation" again. If you have multiple installations of VisualDSP++ on your computer, be doubly sure you are navigating to the installation you wish to Update.
5. Select "Apply a downloaded Update" and click "Next". Click the "... " browser button and navigate to the .VDU file that you downloaded in step 3. Click "OK", then "Next".
6. Follow the on-screen prompts to complete the installation of the Update.

Significant Additions

The primary purpose of VisualDSP++ Updates is to address problems and stabilize the release. Significant new functionality is not expected to be introduced in an Update. However, incremental support (i.e., emulation, example programs, header files, default LDF, errata accommodations, EZ-KIT Lite software, etc.) for new semiconductor products will be added as these products become available and gain support within the VisualDSP++ tools.

In this release:

1. Software support for the ADSP-BF538F EZ-KIT Lite has been added. This includes debug connectivity via the on-board USB debug agent, the flash programmer (both GUI support and underlying drivers), and LwIP (Ethernet) drivers. Flash drivers are provided for both the on-chip flash found on the ADSP-BF538F and the off-chip flash device on the EZ-KIT Lite. The on-line help system has been updated to include this product.
2. The System Service Library (SSL) has been enhanced to support ADSP-BF538 Blackfin processor. Included in this Update is support for the EBIU, Dynamic Power Management, DMA, Interrupt, Deferred Callback, Timer, Flag and Port Control system services for the ADSP-BF538 processor. The default LDFs have been updated to link against SSL.
3. Blackfin device drivers have been updated. The adi_ad1836a_ii and adi_ad1938_ii codec drivers now support automatic SPORT configuration. PPI, UART, SPI, TWI and SPORT device drivers for the ADSP-BF538 processor have been introduced.

4. Software support for the ADSP-21375 EZ-KIT Lite has been added. This includes debug connectivity via the on-board USB debug agent, and well as flash programmer GUI support and underlying drivers. Execution from external memory is now supported in simulator, emulator, and EZ-KIT Lite debugging sessions. The on-line help system has been updated to include this product.

Changes to Existing Behaviors, Projects, and Source Code

When addressing problems, we attempt to make any changes backward compatible with existing projects. However, depending on the nature of a problem, compatibility issues are sometimes unavoidable. This section highlights any changes in the Update that may require the modification of “working” projects or otherwise influence existing behavior.

In this release:

1. Within the Blackfin DMA Manager’s include file, `adi_dma.h`, an additional field named `CallbackFlag` has been added to the data structures that describe large and small model descriptors, `ADI_DMA_DESCRIPTOR_LARGE` and `ADI_DMA_DESCRIPTOR_SMALL` respectively. This field should be set to `TRUE`, if a callback is requested after the descriptor has been processed or `FALSE` if no callback is requested after the descriptor has been processed. Previously, the `DI_EN` bit within the configuration register of the descriptor was used to trigger a callback.

This change affects only user code that explicitly calls the `adi_dma_Queue()` function.

2. Source code files that make calls into the System Services (code that includes the file “`services.h`”) should be rebuilt after installation of this update.

Problems Addressed

The following table is a list of the problems addressed in this Update. Details on any particular problem can be found on the Tools Anomaly web page. Note that after the Issues headings in the top half of the Tools Anomaly web page, problems are detailed in numeric order. The URL is:

<http://www.analog.com/processors/technicalSupport/toolsAnomalies.html>

Processor Family	Problem Number	Tool	Description
All	28661	Install	FYI Norton Internet Security prevents installation
All	28799	Run Time Libraries	fread may signal EOF prematurely
All	28849	Compiler	Compiler driver+LDF requires dummy.c file
All	28895	Run Time Libraries	C++ runtime support for alternative heaps fails to link (li1021)
All	28948	Compiler	loop pragmas don't work on C++ loops with embedded declarations
All	29012	VDK	VDK kThreadSwitched history events don't call UserHistoryLog
All	29090	IDDE	-g is added when convert project from 4.0 to 4.5 with -g>0 in th
All	29378	VDK	Incorrect behaviour following VDK timequeue wraparound
Blackfin	25362	Emulator	Emulator returns unknown family when targeting BF534 rev 0.2
Blackfin	26646	CRTGen	Rev 1.7 BF533 EZ-Kit not properly supported in generated CRT
Blackfin	28335	Simulator	C++ exceptions cannot be used on single core BF561 simulator
Blackfin	28339	Simulator	Simulator not updating registers correctly
Blackfin	28595	IDDE	Additional include directories not always added to command line
Blackfin	28764	System Services	1836A_ii codec device driver fails in TDM mode
Blackfin	28819	IDDE	Adding file to project with user defined config cause error
Blackfin	28823	Compiler	3-byte structs returned from functions incorrectly when -O used
Blackfin	28839	IDDE	impossible sclk value in Project wizard
Blackfin	28875	Run Time Libraries	Typo in cdefBF532.h - voidl instead of void
Blackfin	28967	Configurator	configurator doesn't handle more than 11 devices properly
Blackfin	29025	VDK	The placement of TMK and VDK libraries can cause link errors
Blackfin	29072	IDDE	Blackfin processors not listed in EL Global Properties dialog
Blackfin	29177	Run Time Libraries	SIC_IMASK set with bad bit before DMA transfer (meminit)
Blackfin	29231	Loader	BF533 rev 0.5 loader files add a zero byte for each data byte
SHARC	28857	Emulator	primes for 21061 does not run
SHARC	28882	Run Time Libraries	SIMD libdsp function vecdotf() might fail
TigerSHARC	28720	Compiler	TigerSHARC wrapper generation/regs_clobbered not saving K conds
TigerSHARC	28736	Splitter	The splitter counts one more byte per word in .stk format
TigerSHARC	28752	Run Time Libraries	TigerSHARC fread can incorrectly return 0 in byte-address mode
TigerSHARC	28907	VDK	TIMER0H register not set when Timer interrupt is set to None

TigerSHARC	29032	IDDE	Can't load Annotations
TigerSHARC	29043	Assembler	invalid warning about mult instruction option
TigerSHARC	29227	VDK	VDK API level check can cause false positive Kernel Panics

VisualDSP++ 4.5 (Updated July 2006) Release Note

The following release note concerns the July 2006 Update to the VisualDSP++ 4.5 release. This is the first in what is anticipated to be a series of Updates. The contents of future product Updates will be inclusive of all previous Updates. At that time, the release notes for past Updates will be appended to the end of the current release note.

Identifying Which Update Is Currently Installed on Your System

The Update level is identified in three places:

1. The Add/Remove Programs Control Panel entry for VisualDSP++ 4.5.
2. The VisualDSP++ GUI's About box, located at "Help" > "About VisualDSP++".
3. In the file ...\\System\\VisualDSP.ini, in the ProductName key.

Installing an Update

The procedure for installing an Update to VisualDSP++ is described below. Note that with VisualDSP++'s support for installing multiple instances of itself, it is possible to "trial" an Update in a new directory before switching over your "golden" tools installation to the Update.

1. Use the Start Menu to navigate to VisualDSP++'s "Maintain this Installation" item.
2. Select "Go to the Analog Devices website" and click "Next". This will launch your web browser and navigate it to the proper URL to download Updates from.
3. Download the VisualDSP++ Update file (.VDU) of interest to your hard disk. Note that these files have a .VDU file extension and cannot be executed directly.
4. Navigate to "Maintain this Installation" again. If you have multiple installations of VisualDSP++ on your computer, be doubly sure you are navigating to the installation you wish to Update.
5. Select "Apply a downloaded Update" and click "Next". Click the "... " browser button and navigate to the .VDU file that you downloaded in step 3. Click "OK", then "Next".
6. Follow the on-screen prompts to complete the installation of the Update.

Significant Additions

The primary purpose of VisualDSP++ Updates is to address problems and stabilize the release. Significant new functionality is not expected to be introduced in an Update. However, incremental support (i.e., emulation, example programs, header files, default LDF, errata accommodations, EZ-KIT Lite software, etc.) for new semiconductor products will be added as these products become available and gain support within the VisualDSP++ tools.

In this release:

1. The Blackfin compiler, runtime, VDK, and SSL libraries include new workarounds for hardware anomalies 05-00-0189 and 05-00-0283. The compiler will automatically enable these workarounds when building for parts and silicon revisions that require them. Alternatively they can be enabled using the `-workaround` switch.

05-00-0283 –

One part of the workaround is to include a code sequence in all event handlers. The sequence makes a mispredicted jump over a dummy MMR read. This must be done before any `SSYNCS` in the handler. This sequence is generated by the compiler for C/C++ based event handlers that use `#pragma interrupt` or `sys/exception.h` defined macros such as `EX_INTERRUPT_HANDLER`. The two handlers affected in the

runtime libraries are `_cplb_hdr` and the interrupt dispatcher `_despint` which have been modified to include the workaround.

The second part of the workaround is to avoid system MMR writes in the two instructions after a not-predicted conditional jump. The compiler will insert `nop` instructions to avoid this when it identifies the problem sequence.

These workarounds can be enabled using the `-workaround stalled-mmr-write-283` switch. When the workaround is enabled the macro `__WORKAROUND_STALLED_MMR_WRITE_283` is defined at compile, assemble and link stages

05-00-0198 –

A workaround for this anomaly was already available in the compiler. However the conditions which cause anomaly have changed to include a new code sequence. The compiler has been modified to identify this new sequence. The anomaly may occur where MMR reads or writes occur immediately after a stalled memory read. The compiler will avoid such code being generated for C/C++ compiled code. The runtime libraries are safe against this anomaly.

This workaround can be enabled using the `-workaround sdram-mmr-read` switch. When the workaround is enabled the macro `__WORKAROUND_SDRAM_MMR_READ` is defined at compile, assemble and link stages.

2. There has been a change of compiler behavior relating to MMR (Memory Mapped Register) accesses and volatile variables. The new switch `-no-assume-vols-are-mmrs` has been added.

There are various MMR related hardware errata that the compiler supports workarounds for; 05-00-0122, 05-00-0157, 05-00-0198, 05-00-0283. Previously the compiler would only implement these workaround for accesses that it could absolutely determine were to MMRs. This in practice meant that only literal MMR addresses accesses could be determined accurately. More complex accesses, for example using addresses stored in variables, might not be identifiable as MMR accesses and could therefore result in the various anomalies being hit.

The compiler has been modified to try and avoid missing these more complex MMR accesses. If there is an access to a variable that is defined as `volatile`, and the compiler cannot determine that the access is not to an MMR, the compiler will now assume it is an access to an MMR unless the new switch `-no-assume-vols-are-mmrs` is used.

Changes to Existing Behaviors, Projects, and Source Code

When addressing problems, we attempt to make any changes backward compatible with existing projects. However, depending on the nature of a problem, compatibility issues are sometimes unavoidable. This section highlights any changes in the Update that may require the modification of “working” projects or otherwise influence existing behavior.

In this release no changes have been identified.

Problems Addressed

The following table is a list of the problems addressed in this Update. Details on any particular problem can be found on the Tools Anomaly web page. The URL is:

<http://www.analog.com/processors/cda/epTASearch>

Processor Family	Reference Number	Tool	Description
All	28180	Compiler	modena test c0527101 fails byte addressing when compiled -eh
All	28225	Compiler	C++ exceptions thrown from inline virtual functions may fail
All	28244	VDK	Issue with dynamically created VDK components at startup
All	28271	Run Time Libraries	Increase in code size for printf
All	28341	Run Time Libraries	attributes missing in libx dojs
All	28399	Compiler	static C++ classes can cause bad debug
All	28876	Assembler	Cannot perform source-level debug of assembly source files
All	28929	Emulator	USB-ICE inoperable when updating to 4.5 while connected
All	28935	VDK	User's timer interrupt settings can be overwritten
All	29140	Emulator	RoHS USB-ICE does not work with base 4.5 install
Blackfin	28043	Loader	Loader supports different default Rev #s from what it should.
Blackfin	28229	IDDE	Annotations left in source pane after they are turned off
Blackfin	28287	Loader	Zero padding to booting stream
Blackfin	28297	Compiler	Compiler internal error (macdefs.c:1162) with -O
Blackfin	28305	Run Time Libraries	ftell() with -full-io in text mode can return incorrect position
Blackfin	28309	Compiler	Non-interrupt safe prologue code generated for BF535
Blackfin	28338	Compiler	INTERNAL COMPILER ERROR: No switch note found
Blackfin	28383	Assembler	.inc/binary produces corrupted doj
Blackfin	28410	Run Time Libraries	Cache flushing on BF535 and wireless parts doesn't work
Blackfin	28445	IDDE	If I add cplbtable and then disable cache project does not link
Blackfin	28450	IDDE	configurator screen not coming to front
Blackfin	28467	IDDE	errors and other issues removing configurations
Blackfin	28472	Run Time Libraries	Possibility of erroneous result computed by fir_decima_fr16()
Blackfin	28487	Run Time Libraries	Wrong comment in the source of the radix2 FFT library functions
Blackfin	28497	Run Time Libraries	Incorrect macro in defBF534.h and defBF538.h
Blackfin	28517	LDFGen	Possible link error with generated BF561 LDF and mem init
Blackfin	28521	LDF	OTHERCORE not implemented correctly in default multi-core LDFs
Blackfin	28588	Compiler	bad compare of unsigned short and unsigned literals -O
Blackfin	28600	Loader	Loader does not work with Rev 0.3 for 539.
Blackfin	28679	Loader	Remove the ignore block from loader files.
Blackfin	28686	LDFGen	Single core generated LDF uses \$OBJECTS before definition
Blackfin	28688	Run Time Libraries	Instance of speed path anomaly 05-00-0209 in cache flush func
Blackfin	28689	LDFGen	LDFGen does not always use the correct CPLB table
Blackfin	28710	Loader	Loader need to support Rev. 0.5 for 531/2/3
Blackfin	28765	IDDE	project not restored after starting connection-less IDDE
Blackfin	28779	Run Time	defbf534.h has incorrect PFDE_UART macro definitions

		Libraries	
Blackfin	28994	VDK	Potential for excessive stack usage on Blackfin processors
SHARC	28283	LDF	The section "seg_int_code" has grown unnecessarily
SHARC	28569	VDK	Sharc li2152 link warnings when using earlier Si Revision
SHARC	28692	Run Time Libraries	0.0 2126x libraries built with an inappropriate silicon revision
SHARC	28761	Run Time Libraries	No SRU header files
TigerSHARC	28263	Compiler	long long to double conversion fails in byte-addressing mode
TigerSHARC	28267	Compiler	Assertion failure: bril/zp/macdefs.c:2747 with -O -never-inline
TigerSHARC	28295	VDK	Cannot view system stack usage in the expert linker
TigerSHARC	28490	VDK	CCNTx register is read in the wrong order
TigerSHARC	28880	VDK	TS20x Idle thread prevents scheduling of user threads