VisualDSP++ 3.5 for 16-bit Processors (Updated October 2006) Release Note

The following release note concerns the October 2006 Update to the VisualDSP++ 3.5 for 16-bit Processors 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++ 3.5.
2. The VisualDSP++ GUI's About box, located at "Help" > "About VisualDSP++".
3. In the file …`\System\VisualDSP.ini`, in the `ProductName` key.

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.

In this release:

1. The emulation software has been updated to support the RoHS-compliant versions of the USB-ICE and HPUSB-ICE products. This Update will be required to support ICE hardware distributed after July 2006. **This software update is beneficial only to users of the ADSP-219x family**. Users of Blackfin, SHARC, and TigerSHARC must use VisualDSP++ 4.0 (July 2006 Update) or later to connect to RoHS-compliant ICE products (No changes were required in the ADSP-218x emulation software).

   The procedure to update the emulator's device driver is as follows. This procedure must be followed to ensure correct upgrade of the device driver:

   1. When executing the EXE that is part of this Update, the installer will ask where files should be extracted to (with the default location being your TEMP directory). Make note of this location.
   2. After installation of the Update, cycle the power on your ICE and you will be prompted by Window's Hardware Wizard. Select "`Install from a list or specific location (Advanced)`" and navigate the Hardware Wizard to the location where the files were extracted to in Step 1. The Hardware Wizard will successfully install the device drivers from this location.

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. 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

23369	Linker	Linker 219x external memory check incorrectly fails
24247	Compiler	error cc1431 and assertion brilops.c:96 initializing far data
26214	Run Time Libraries	cdef218x.h has a couple of errors
26387	Run Time Libraries	fftN and ifftN functions modify I5 in autobuffer mode
26388	Compiler	floating-point division by zero does not return +INF
28556	Compiler	complicated operand to long int operation can cause bad code
29022	Emulator	Emulators must continue to work with version 3.5

VisualDSP++ 3.5 for 16-bit Processors (Updated April 2005) Release Note

The following release note concerns the April 2005 Update to the VisualDSP++ 3.5 for 16-bit Processors 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:

4. The Add/Remove Programs Control Panel entry for VisualDSP++ 3.5.
5. The VisualDSP++ GUI's About box, located at "Help" > "About VisualDSP++".
6. In the file …`\System\VisualDSP.ini`, in the `ProductName` key.

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 no additions have been identified.

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. 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

17204	Loader	meminit does not issue an error message with an illegal switch
18691	Simulator	Interrupt 7 and above for Signals API do not work
20144	Loader	elfloader doesn't inform that ADSP-2153x is no longer supported
20958	Loader	Data missing in loader file
20960	Splitter	splitter file not produce extended linear address record
21627	Compiler	compiler assertion (driver.c:2207) with asm() func ptr ops
22109	Splitter	Splitter produces misaligned DM image
22165	Loader	elfspl21 error: Unable to read string in dynamic section
22218	Assembler	Nested .IF conditional assembly incorrect (Blackfin only)
22275	Utilities	Elf2Aexe-Internal: Internal error found in Symbol table size
22365	Compiler	pragma interrupt may cause write to TX0 and TX1
22737	Compiler	long OR of SHIFT result generated incorrectly
22832	Debug Agent	After connecting to EZKit Lite with 4.0, 3.5 fails (BF561)
22884	Compiler	off by one when issuing error (cc0095) for oversize arrays
22977	Run Time Libraries	fft and ifft functions duplicate data

VisualDSP++ 3.5 for 16-bit Processors (Updated November 2004) Release Note

The following release note concerns the November 2004 Update to the VisualDSP++ 3.5 for 16-bit Processors 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:

7. The Add/Remove Programs Control Panel entry for VisualDSP++ 3.5.
8. The VisualDSP++ GUI's About box, located at "Help" > "About VisualDSP++".
9. In the file …\System\VisualDSP.ini, in the ProductName key.

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. New work-arounds for a Blackfin hardware anomaly has been introduced: The compiler driver now supports avoidance of the following anomaly issue:

   ADSP-BF53{3|2|1}: `-si-revision {0.1-0.3}`
   ADSP-BF561: `-si-revision {0.1-0.3}`
   `-workaround dreg-comp-latency` or `-workaround all`

   This workaround supports the following anomalies:

   ADSP-BF533 anomaly #80
   ADSP-BF531 anomaly #79
   ADSP-BF532 anomaly #79
   ADSP-BF561 anomaly #62 - Speed-Path in computational unit affects certain instructions

   When enabled the compiler should insert a NOP instruction between two instructions where the first instruction assigns a value to a DREG, and the second instruction uses the DREG as a parameter to a SIGNBITS, EXTRACT, DEPOSIT or EXPADJ instruction.

   The compiler also defines the macros `__WORKAROUND_DREG_COMP_LATENCY` and `__WORKAROUNDS_ENABLED` at the source, assembly and link build stages when this workaround is enabled.

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. 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

15020	Assembler	Assembler syntax error initializing a single element char
19290	Emulator	Hardware watchpoints are not supported via the BF561 emulator
19330	Assembler	type 9a instr with yreg 0 encoded incorrectly
20360	Compiler	Compiler driver and prelinker not using same temporary path,
20375	Compiler	internal error shifting/extending fields in structs
20391	Simulator	ADSP-BF561 Simulator target location is not obvious
20412	Compiler	IPA fails to analyse vararg functions correctly
20470	Run Time Libraries	BF561: CRT loops when reloading when in interrupt handler
20497	Compiler	Compiler optimizes wrongly resulting in loss of sign extension
20521	Run Time Libraries	Twiddle table generated by the function twidfft2d_fr16 is wrong
20523	Simulator	Halting running simulation can result in debugger PC=0.
20540	Emulator	FIO_POLAR register width is 32 bits instead of 16
20544	Compiler	compiler does not generate hardware loop for long type counters
20807	VDK	Custom marshalled message template doesn't compile

VisualDSP++ 3.5 for 16-bit Processors (Updated September 2004) Release Note

The following release note concerns the September 2004 Update to the VisualDSP++ 3.5 for 16-bit Processors 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++ 3.5.
2. The VisualDSP++ GUI's About box, located at "Help" > "About VisualDSP++".
3. In the file …\System\VisualDSP.ini, in the ProductName key.

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. Cache control in the Blackfin start up code has been improved. The DCBS (Data Cache Bank Select) bit in the DMEM_CONTROL register can be set to a user-defined value by the startup code. This is achieved by including the bitmask CPLB_SET_DCBS (defined in cplb.h) in the value of the __cplb_ctrl variable. The default behavior is the same as previous releases -- the bit is cleared. The DCBS bit specifies which bit of a memory address (bit 14 if DCBS=0, bit 23 if DCBS=1) determines the data cache bank (A or B) used to cache the location. Depending on the placement of data within the application memory space, one setting or the other ensures more data is cached at runtime. See the Hardware Reference Manual for further details.

2. New work-arounds for two Blackfin hardware anomalies have been introduced:

   ADSP-BF53{3|2|1}: `-si-revision {0.0-0.2}`
   `-workaround sdram-mmr-read`

   Enables workarounds for anomaly 04-00-0038 "Memory Access Pipeline Bug" which effects ADSP-BF53{3|2|1} silicon revisions 0.0-0.2. The problem is seen in a sequence of SDRAM load directly followed by an MMR load and will result in an incorrect value being loaded from the MMR. When this workaround is enabled. The compiler will insert a NOP between the two loads. It will also define the following macros at the compile, link, and assembly phases:

   ```
   __WORKAROUND_SDRAM_MMR_READ
   __WORKAROUNDS_ENABLED
   ```

ADSP-BF535: `-si-revision {0.0-1.3}`
AD6532: `-si-revision {0.1|0.2|1.0}`
`-workaround astat-rnd_mod`

Enables workaround for anomaly 05-00-0197 "Latency in Writes to RND_MOD bit" which affects ADSP-BF535 and AD6532. Enabling the workaround inserts two `NOP` instructions after any direct write to the ASTAT register. The compiler will only generate a write to the ASTAT register when `__builtin_sysreg_write()` is used to write to the ASTAT register.

When this workaround is enabled, the following macros are defined at the compile, link, and assembly phases:

```
__WORKAROUND_ASTAT_RND_MOD
__WORKAROUNDS_ENABLED
```

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. 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

19956	Simulator	BF561: SICB_SYSCR supplemental interrupts not simulated
20077	IDDE	fopen not working - query regression
20088	Compiler	Compiler doesn't generate dependency information for .asm files
20125	Compiler	long switch types not supported for 218x and 219x
20133	Compiler	Compiler may clobber P0 in function prologue
20136	Run Time Libraries	_l1_memcpy() can corrupt memory when processing C++ exceptions
20146	Compiler	dense switch with range of large case values may fail to link
20195	Run Time Libraries	cdefBF561.h: DMA*_*_*_MODIFY should be volatile short*
20203	Compiler	far pointers not displayed correctly in source level debugging
20209	Run Time Libraries	RTL should allow config of DMEM_CONTROL DCBS bit
20233	Compiler	pm far pointers cause compiler assert: No Register Left To Claim
20243	Run Time Libraries	linker elimination can delete -Os functions entry routine
20334	Compiler	compiler does not restore DMPG1 after an asm clobbering it
20337	Compiler	incorrect far access to 2-D array after integer return function
20344	Compiler	Compiler should provide workaround for anomaly 04-00-0038:

VisualDSP++ 3.5 for 16-bit Processors (Updated August 2004) Release Note

The following release note concerns the August 2004 Update to the VisualDSP++ 3.5 for 16-bit Processors 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++ 3.5.
2. The VisualDSP++ GUI's About box, located at "Help" > "About VisualDSP++".
3. In the file …`\System\VisualDSP.ini`, in the `ProductName` key.

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. Better partitioning of SDRAM for ADSP-BF533 and ADSP-BF561 EZ-Kit Lites are now available in the default LDFs.

   The External Bus Interface Unit's (EBUI) SDRAM Controller (SDC) provides access to the four external banks of SDRAM that can be supported. As well as supporting up to four external banks, the SDC also provides support for accessing four internal banks within each of the external SDRAM banks. The SDC allows for multiple internal banks to remain open in parallel, which can offer improved performance. For example executing instructions from one external bank which accesses data in another internal bank.

   The ADSP-BF533 allows for four internal banks, within a single external bank to remain open at the same time. The ADSP-BF561 allows for four internal banks, across the four external banks to remain open at the same time.

   The default Linker Description Files (LDF) for the ADSP-BF533 and the ADSP-BF561 now contain an optional enhanced setup which provides SDRAM partitioning in the manner above.

   For ADSP-BF533 the configuration is enabled by passing the macros `-MDUSE_CACHE` and `-MDPARTITION_EZKIT_SDRAM` to the linker. The configuration makes use of the first external SDRAM bank, which is the SDRAM configuration for the ADSP-BF533 EZ-Kit Lite. The memory is partitioned as follows:

   E0I0: Heap (8MB)
   E0I1: Data (8MB)
   E0I2: Data/Bsz (8MB)
   E0I3: Program (8MB)

For ADSP-BF561 EZ-KIT Lite the configuration is enabled by passing the macro `-MDUSE_CACHE` to the linker. The configuration makes use of the first and second external SDRAM banks, both of which are populated by 32MB SDRAM for the ADSP-BF561 EZ-Kit Lite. The memory is partitioned as follows:

E0I0: Core A Heap
E0I1: Core A Data
E0I2: Core A Data/Bsz
E0I3: Core A Program
E1I0: Core B Heap
E1I1: Core B Data
E1I2: Core B Data/Bsz
E1I3: Core B Program

Note: For ADSP-BF561 enabling the `USE_CACHE` linker macro moves the heap from L2 memory to L3/SDRAM. The L2 space is now made available for additional program/data.

2. Versions of the Blackfin VDK core libraries built with workarounds enabled are included in this Update. In order to ensure that these libraries are linked in when workaround libraries are specified for existing projects, the relevant changes to the default VDK LDF must be incorporated in the project's LDF.

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. 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

19285	Compiler	Internal error when declaring a var with section("bsz")
19495	Run Time Libraries	SDRAM in LDFs should be partitioned into data and code areas
19938	Run Time Libraries	Radix-2 ffts in Blackfin libdsp fail for N>16384
19947	Compiler	Compiler optimizer may remove sysreg_read/write calls
19993	Run Time Libraries	ETSI library function mult_r doesn't restore RND_MOD bit
19999	Run Time Libraries	float_to_fr32() may return incorrect LSW for 32-bit result
20040	Compiler	Compiler workaround "-workaround wt-dcache" not complete
20051	Compiler	Compiler crashes compiling illegal statement
20052	Compiler	large locals cause compiler assert:ARCHloadlit: literal out of r
20059	Flash Programmer	BF561 Flash programmer driver incorrectly protects memory region

VisualDSP++ 3.5 for 16-bit Processors (Updated July 2004) Release Note

The following release note concerns the July 2004 Update to the VisualDSP++ 3.5 for 16-bit Processors 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++ 3.5.
2. The VisualDSP++ GUI's About box, located at "Help" > "About VisualDSP++".
3. In the file …`\System\VisualDSP.ini`, in the `ProductName` key.

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 no additions has been identified.

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. Only users of the ADSP-BF561 EZ-KIT Lite are impacted by this change. The signals PF7 and PF8 were being driven by the Cypress part on the BF561 EZ-KIT Lite whether connected using the USB cable or connecting to the EZ-KIT Lite using an emulator. This issue can be resolved only through the EZ-KIT Lite's USB connection and not through an ICE. If you have never used the USB connection before, the "Add new hardware wizard" will start when making the USB connection. Follow the on-screen instructions, making sure to point the wizard to tvvhe folder that contains the `WmUSBEz.inf` and `WmUSBEz.sys` files from the Update that is being installed. If the driver installation is successful, the USB_Monitor LED (`LED4`) will illuminate.

   While still connected through USB, open a DOS command prompt and change directories to `...\VisualDSP 3.5 16-Bit\System`. Execute the following command and follow the on-screen prompts exactly to fix this issue:

   ```
   WmEzKitUpdate -e
   ```

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

18326	Debug Agent	Some system MMR's do not read/write when using memory window
19548	Debug Agent	some pins in the firmware are set as outputs and shouldn't be
19554	Run Time Libraries	Overflow/Carry generating ETSI functions incorrect
19597	IDDE	Positive long long value displayed incorrectly
19698	Compiler	Compiler driver removes .obj files from the command line
19721	IDDE	close button in source window is incorrectly disabled
19725	Run Time Libraries	DSP function rms_fr16 fails for -ve data if sample length is 1
19731	Run Time Libraries	meminit_support run before main() when meminit not used
19759	Utilities	meminit produces .meminit section possibly misaligned
19770	Compiler	compiler assertion in driver.c:1101
19782	Compiler	far pointer assertion in compiler: Cannot locate Reg to be Relea
19791	Compiler	io_space_read with non-lit input causes assert:Estack not empty
19822	Compiler	three unnecessary loads of registers AR in pragma interrupt func
19829	Compiler	-csync workaround can insert NOPs after predicted taken branches
19839	Compiler	-workaround all switches on all workaround not just all targets
19844	Compiler	compiler assertion with large automatics: ARCHrlit: invalid ope

VisualDSP++ 3.5 for 16-bit Processors (Updated June 2004) Release Note

The following release note concerns the June 2004 Update to the VisualDSP++ 3.5 for 16-bit Processors 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++ 3.5.
2. The VisualDSP++ GUI's About box, located at "Help" > "About VisualDSP++".
3. In the file …\System\VisualDSP.ini, in the ProductName key.

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. Build tool support for a new Blackfin derivative, the ADSP-BF539, has been added. Similar support for four wireless derivatives (AD6527, AD6528, AD6529, AD6758) has been added.

2. Support to place library code into shared memory of the ADSP-BF561 as been added. A document detailing the configuration of such an environment is available at:

   ftp://ftp.analog.com/pub/tools/patches/doc_updates/
   ADSP-BF561_with_Shared_Resources.pdf

3. Support to work-around two cache anomalies on the ADSP-BF535 and AD6532 has been added.

   ADSP-BF535: -si-revision {0.0-1.3}
   ADSP-AD6532: -si-revision {0.1|0.2|1.0}
   -workaround wb-dcache

   Enables workaround for anomaly 05-00-0165 "Data cache dirty bit set when a load-miss-fill is in progress", which affects ADSP-BF535 and ADSP-AD6532. Enabling the workaround has the following effects:

   - An SSYNC instruction is placed at the beginning of ISRs.
   - Two NOP instructions are placed after RTI, RTX, RTN, and RTE instructions.
   - The assembler is passed the switch -wb_fix, which results in a NOP being inserted between conditional branches and load or store instructions.

   When this workaround is enabled, the following macros are defined at the compile, link, and assembly phases:

   __WORKAROUND_WB_DCACHE
   __WORKAROUNDS_ENABLED

ADSP-BF535: `-si-revision {0.0-1.3}`
ADSP-AD6532: `-si-revision {0.1|0.2}`
`-workaround wt-dcache`

Enables workaround for anomaly 05-00-0164 "Store to Load Forwarding in Write Through Mode" which affects ADSP-BF535 and ADSP-AD6532. This workaround includes the wb-dcache workaround, and in addition:

- An `SSYNC` instruction is placed before the `RTI` instruction in ISRs.
- In conjunction with the assembler, to which the switch `-wb_wt_fix` is passed, an `SSYNC` instruction is inserted between the target of a conditional branch and a the first store in that basic block.

When this workaround is enabled, the following macros are defined at the compile, link, and assembly phases:

```
__WORKAROUND_WT_DCACHE
__WORKAROUNDS_ENABLED
```

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. A new voltage selection capability for HPUSB and USB-ICEs has been added. This requires the user to choose the proper I/O voltage for their target when running `IceTest`. It defaults to 0 and will fail if another valid selection is not made. This is a change to the previous behavior of this utility.

   When running VisualDSP++ it will default to 3.3/5 V for existing platforms. The JTAG Configurator now has another field for HPUSB and USB-ICE platforms which will allow you to choose the proper JTAG I/O voltage for new platforms that you make.

   This capability was added for processors which support and/or require something other than a 3.3/5 V JTAG I/O level.

2. Support for objects file names which use a `.o` suffix will generate a warning in the latest tools. The support for this undocumented suffix will be removed in future releases of the tools. This is because the C++ runtime and compiler support are only designed to work with the correct `.doj` suffix named files.

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

12837	Simulator	memdma config reg is not loaded correctly at end of transfer
14285	L/A Preprocessor	Linker fails with line continuation '\' outside of preproc macro
16769	Run Time Libraries	Cache flush routine should avoid anomaly 05-00-0123
16989	Compiler	LDF split of L1 memory limited sizes unnecessarily
17056	Compiler	Default LDFs don't have .bsz "spill over" into MEM_L1_DATA_B
17180	Simulator	Timer tscale register implemented as 32 bits
17303	Simulator	Superfluous mem_iexc_c1f internal trace message output
17354	Run Time Libraries	defAD6532.h should not simply include defBF535.h
17808	Simulator	Expanded lsetup may cause runtime failure for BF535 simulator
18146	IDDE	R3 in Compiled Simulator under Custom Regs is really R3.B
18689	Compiler	-ipa incompatible with dual-core projects; prelinker loops
19250	IDDE	Memory:Fill writes bad data if count is larger than file values
19302	Compiler	compiler assert-"Trying to lock already paired reg as singleton"
19325	Emulator	IDDE may crash if 218x hardware stack overflows
19357	Linker	Resolve command (linker)
19365	Run Time Libraries	problems in cplbtab files
19368	IDDE	Source Control crashes when checking-in file
19404	Compiler	Calling interrupt with workaround leaves interrupts disabled
19409	Emulator	R0 destroyed when using "Mask interrupts during step" option
19420	IDDE	Change project name in Project Options to include $ fails build
19441	Compiler	Comparisons of unsigned ints can be wrong
19470	Compiler	Empty structs can produce a compiler crash
19474	Compiler	Compiler optimizes switch statement incorrectly
19511	Run Time Libraries	cdefblackfin.h declares incorrect size for event control regs
19513	Compiler	Compiler may re-order a read and a cli/sti when optimised
19525	Installation	Various Blackfin EZ-KIT Examples missing files after Update
19529	Compiler	Compiler assertion -O: Release cannot find register
19533	Compiler	Compiler assertion -O shift lit: Release cannot find register
19545	Compiler	Compiler may generate misaligned pointer in overlapping memcpy
19546	Run Time Libraries	Default LDF for BF561 overwrites 2nd stage loader

VisualDSP++ 3.5 for 16-bit Processors (Updated May 2004) Release Note

The following release note concerns the May 2004 Update to the VisualDSP++ 3.5 for 16-bit Processors 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++ 3.5.
2. The VisualDSP++ GUI's About box, located at "Help" > "About VisualDSP++".
3. In the file …\System\VisualDSP.ini, in the ProductName key.

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. A new switch, -decls-<weak|strong>, has been added for use with the compiler. This switch controls how the compiler interprets uninitialized global variable definitions, such as "int x;". The -decls-strong switch treats this as equivalent to "int x = 0;", in that other declarations of the same variable in other modules will cause a "multiply-defined symbol" error. The -decls-weak switch treats this as equivalent to "extern int x;", i.e. as a declaration of a symbol that is defined in another module. Use of -decls-weak has a risk of a slight performance disadvantage compared to -decls-strong, because it referenced variables by a separate label as opposed to a common label for the definition's output section. The default is -decls-strong. ANSI C behavior is -decls-weak.

2. The compiler has been enhanced to allow users to disable the default GNU compatibility extension which causes the compiler to accept string literals which span several lines without a '\' at the end of each line. The new switches are:

   -multiline
   Enables string literals over multiple lines. This is the default mode.

   -no-multiline
   Disables string literals over multiple lines.

3. A new ADSP-BF533 EZ-KIT Lite example has been provided that demonstrates usage of the RTL's ability to extend I/O support to new devices. In this example, stdout is emitted via the serial port on the EZ-KIT Lite. This example can be found in the …\Blackfin\EZ-KITs\ADSP-BF533\Examples\STDIO UART directory.

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. It has been necessary to introduce the use of a new input code section as part of the solution to a problem found with the runtime C++ exceptions support. The new section, called `noncache_code`, is used to define a small number of functions from the exceptions support library (`libx`) and is required to be mapped to memory which cannot be configured as cache (i.e. not L1 SRAM/Cache memory blocks). The default LDFs have been modified to map this section in the necessary way. Applications which use C++ exceptions and a non-default LDF file will need to make a corresponding change, otherwise the linker will issue an error.

2. The previous versions of the ADSP-BF561's default LDF declared a single heap section that would be shared between both cores. However, as the runtime libraries are not re-entrant, it is not safe to share the heap section between both cores. Users who have an existing ADSP-BF561 LDF with a single heap section in shared memory (and are allocating from the heap on both cores) should update their LDF to follow the new model. Note that the heap space in the new default LDF for each core is approximately half the previous size; the old heap section has been split into two sections, one for each core.

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

16885	Compiler	Global variables are not marked .weak
17666	Utilities	Meminit fails if run on dxe from AD6532
17920	Loader	Loader takes only one section for initial block
18785	Compiler	Inline ASM clobber of RETS not working.
18895	Compiler	assertion with -Os "Unexpected load in loop"
18897	Compiler	Compiler allows illegal #pragma regs_clobbered statements
18931	Compiler	__builtin_external_vol_memory_read use deleted from loop with -O
19000	Compiler	Objects are not destroyed after an exception has been handled
19014	Linker	.LSW2147483644-0x100000 (=-0x7ffe8) is out of range
19016	Run Time Libraries	Unused bsz section in EL Template LDFs
19059	Compiler	Unsigned long long shift causes stack overwriting
19061	Compiler	Compiler may loop when optimizing division of short parameter
19067	Run Time Libraries	realloc does not copy old block memory to new
19069	Compiler	assertion copying far structure types in C
19075	Compiler	ISR can clobber saved value of RETS
19076	Compiler	Compiler generates bad ISR wrapper with HW workaround enabled
19083	Compiler	Compiler crashes when optimising loop with C++ typeid calls
19094	Compiler	Various 219x VDK Examples build with warnings
19099	Compiler	compiler creates invalid interprocedural optimisation file
19103	Compiler	shifts compiled badly -O including ETSI shl operator
19107	Examples	Example no longer links
19128	Run Time Libraries	DMCx_P bit position macros in def header files incorrect
19133	Run Time Libraries	scanf, fscanf and sscanf return incorrect when "%%" or "%n" used
19158	Debug Agent	Install only Blackfin, then March update, BF561 not available
19196	Compiler	VDSP++ 3.5 compiler generates more data than previous versions
19215	Compiler	Compiler doesn't reuse stack when function returns using stack
19220	Compiler	String literals over multiple lines cause preprocessing problems
19254	Run Time Libraries	__l1_memcpy can be put in cacheable L1 memory
19293	Run Time Libraries	ADSP-BF561C/CPP LDF's place heap in shared memory.

VisualDSP++ 3.5 for 16-bit Processors (Updated April 2004) Release Note

The following release note concerns the April 2004 Update to the VisualDSP++ 3.5 for 16-bit Processors 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++ 3.5.
2. The VisualDSP++ GUI's About box, located at "Help" > "About VisualDSP++".
3. In the file …`\System\VisualDSP.ini`, in the `ProductName` key.

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. A new switch, `-instantlocal`, has been added to the `ccblkfn` compiler. This switch should be used when compiling C++ source code for dual-core targets applications (such as for ADSP-BF561). The switch forces the compiler to instantiate all template entities that are used in the local compilation module. The switch also ensures that all templates are given internal linkage.

2. A new C++ compiler pragma, `no_implicit_inclusion`, has been added. With the `-c++` switch, for each included .h file, the compiler will attempt to include the corresponding .c or .cpp file. This is called implicit inclusion. If `#pragma no_implicit_inclusion` is placed in a .h file, the compiler will not implicitly include the corresponding .c or .cpp file with the `-c++` switch. This behavior only affects the .h file with `#pragma no_implicit_inclusion` and the corresponding .c or .cpp file.

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. Interprocedural Analysis (IPA) is automatically disabled when the linking process detects that a multi-core application (those containing two `main()` functions) is being created. This is being done to avoid some problems in the current IPA implementation which may result in incorrect analysis and optimizations of multi-core applications.

2. The compiler has been changed to issue an error when is sees use of both `-threads` and `-p{1|2}`. This is necessary because instrumented profiling is not supported under VDK.

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

17281	Compiler	Compiling code that takes builtin address causes compiler crash
18331	Assembler	assembler does not fault illegal register pair
18346	Assembler	assembler crash on very-long asm("...") instruction in C source
18579	Linker	Objects misaligned after absolute placement or elimination
18594	IDDE	Load MP confirmation dialog shows previous dxe in a MP session
18675	Compiler	Compiler fails when using options -ED and -M
18701	Splitter	splitter crash at processing zero_init section
18762	Run Time Libraries	Core addresses swapped in BF561 Expert Linker templates
18767	Simulator	DMA Descriptor chaining does not work
18792	Compiler	Bad code in REGS_CLOBBERED functions
18827	Compiler	Compiler incorrectly optimises XOR operation on unsigned short
18859	Run Time Libraries	builtin div_s ETSI operator does not saturate results
18860	Compiler	Compiler raises internal error when compiling loop code
18861	Compiler	Compiler should not allow instrumented profiling with -threads
18862	Run Time Libraries	more than one operator errors when using shortfract ops
18881	Run Time Libraries	relatively large errors for sinf of small input values
18895	Compiler	assertion with -Os "Unexpected load in loop
18922	VDK	SP can be invalid if interrupt serviced during reschedule ISR
18931	Compiler	__builtin_external_vol_memory_read use deleted from loop with -O
18956	Run Time Libraries	C++ fract class uses 16-bit multiplication for 32-bit operands
18992	Compiler	Wrong output after optimization
18994	Compiler	assertion error using pragma reg_clobbered sets with no p-regs
18997	Installation	Some files not updated on consecutive Update installs
19000	Compiler	Objects are not destroyed after an exception has been handled
19023	Run Time Libraries	type of lock variable is incorrect for Blackfin chips

VisualDSP++ 3.5 for 16-bit Processors (Updated March 2004) Release Note

The following release note concerns the March 2004 Update to the VisualDSP++ 3.5 for 16-bit Processors 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++ 3.5.
2. The VisualDSP++ GUI's About box, located at "Help" > "About VisualDSP++".
3. In the file …\System\VisualDSP.ini, in the ProductName key.

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. Under Blackfin emulation sessions, a new option, "Use XML reset values" is available in the "Settings"->"Target Options" dialog box. Specific registers to reset and the values to reset them to can be specified in the RegReset section of …\System\ADSP-BF533.xml (Note that the specific filename to use is whatever processor you are working with). This feature can be used to avoid the need to manually reset, for example, external SDRAM after a target reset.

   Note that future Updates may replace the XML files, in which case you may need to reapply any modifications you make this file.

2. Multiprocessor support for the ADSP-BF561 has been enhanced with two new built-in functions:

   `int __builtin_testset(char *)`
   The pointer should point to a location in shared L2 SRAM that will contain the testset byte. The built-in generates code that performs the TESTSET instruction on the location provided. The result of the operation (the success or failure value which is set in CC) is returned by the built-in.

   `void __builtin_untestset(char *)`
   The pointer should point to a location in shared L2 SRAM that will contain the testset byte. The built-in produces the code to zero the value stored in the location pointed to by the parameter. No checking is performed to ensure the core zero'ing the byte has ownership of the byte.

   Both functions produce code that the compiler considers to be a "strong barrier". Code cannot be moved past the barriers, ensuring that code requiring exclusive access to a resource will do so. For more information on the restrictions on parameters to the TESTSET instruction, please refer to the Blackfin DSP Instruction Set Reference Manual, Pages 11-22 - 11-24.

   It is recommended that users make use of the provided inline functions for accessing testset bytes. Three inline functions are provided in the ccblkfn.h header file:

`void adi_acquire_lock(testset_t *)`
A spin lock function that will return once it has gained control of the given lock. The lock pointed to by the parameter to the function must reside in shared L2 SRAM. The function incurs some overhead as it is required to perform a csync in its loop body to ensure it reads the flushed value of the lock.

`int adi_try_lock(testset_t *)`
A try lock function that returns the value after a single attempt to acquire the given lock. If the request is successful a value of 1 is returned, otherwise 0 is returned.

`void adi_release_lock(testset_t *)`
A release lock function. The function zeroes the given lock. No attempt is made to verify that the core releasing the lock is the owner of the lock.

None of the functions disable interrupts during their execution. This is left at the users discretion.

3. The compiler driver now supports avoidance of two new hardware anomalies.

   ADSP-BF561 anomaly #6 - signbits

   This anomaly avoidance is enabled for all ADSP-BF561 silicon revisions currently supported by `-si-revision`. It is also enabled by use of the switches `-workaround all` and `-workaround signbits`.

   When enabled the compiler will avoid generating code where the input register of a signbits instruction is loaded in the immediately preceding instruction. The compiler also defines macros `__WORKAROUND_SIGNBITS` and `__WORKAROUNDS_ENABLED` at the source, assembly and link build stages when this workaround is enabled.

   ADSP-BF561 - anomaly #24
   ADSP-BF531 - anomaly #49
   ADSP-BF532 - anomaly #69
   ADSP-BF533 - anomaly #49 - killed-mmr-write

   This anomaly is enabled for silicon revisions less than 0.3 when using `-si-revision`. It is also enabled by use of the switches `-workaround all` and `-workaround killed-mmr-write`.

   When enabled the compiler should insert a dummy 32-bit system MMR read at the start of ISR code and to avoid 32-bit system MMR write accesses in the 3 slots after a conditional branch which is predicted not taken. The compiler also defines macros `__WORKAROUND_KILLED_MMR_WRITE` and `__WORKAROUNDS_ENABLED` at the source, assembly and link build stages when this workaround is enabled.

4. A new command-line utility has been provided, `xmlmap2html`. It will convert a linker map file from XML to HTML format. HTML-format map files can be viewed on a PC that does not have the same version of VisualDSP++ installed on it.

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. Two existing library routines for claiming/releasing a testset lock have been deprecated. They now exist as inline functions with calls to the new inline functions detailed above.

   `claim_atomic_access` now calls the `adi_acquire_lock` function.
   `release_atomic_access` now calls the `adi_release_lock` function.

   These two functions will be removed at a later date. As in prior releases of VisualDSP, calls to the `PrimIO` routine, which passes I/O requests to the host system, are guarded using a testset lock to ensure that two concurrent calls to the hosts I/O systems are not made.

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

8190	Linker	Linker allows RESOLVE to HW page not from an overlay section
10422	Compiler	Optimiser throws away function call
15114	Run Time Libraries	CPLB Manager clobbers loop registers
17311	Compiler	C++ exceptions not supported in L1 instruction memory
17528	VDK	Threads killed when insufficient space left to satisfy new
18036	Installation	Manually updating/installing 2191 driver on win98 has many bumps
18063	Emulator	Emulator displays warning message with non Blackfin dev in chain
18064	Simulator	ADSP-BF5xx external interrupts no longer simulated in VisualDSP
18134	Compiler	PM pointers to DM not dereferenced correctly
18147	Emulator	cannot create configurator session for 2189 COM EZ-KIT
18156	Compiler	asm statement output clobbered by compiler generated code non-op
18162	Compiler	internal compiler error when optimizing (eval.c:459)
18165	Compiler	external memory builtins can create bad loads with -O
18176	Run Time Libraries	sysreg.h bit macros can work incorrectly
18214	VDK	Killed 32-Bit system MMR write workaround required in VDK
18216	Simulator	PGO execute fails in compiled simulation with no streams
18274	Simulator	compiled sim -cmvs option broken
18299	Run Time Libraries	pTMRS8_STATUS/pTMRS4_STATUS should be long ptrs in cdefBF561.h
18300	IDDE	Stack can be adjusted to reside on non-aligned mem = DSP crash
18383	Compiler	asm accumulator output operand causes compiler assertion failure
18385	VDK	Zero timeout with kNoTimeoutError corrupts VDK internals
18396	VDK	VDK library code does not always use jump.x when required
18401	Emulator	Only external bank 0 defined for emulator on BF561
18402	VDK	Some VDK LDFs do not have all guard symbols for data cache
18407	Emulator	BF535 and AD6532 default to useM3 for context save/restore
18409	Compiler	Compiler can generate write to stack frame before SP set up
18416	Simulator	Blackfin processor dll missing EVT and PLL register windows
18417	Compiler	use of OldAsmCall linkage can cause bad calls to normal funcs
18418	Compiler	Empty GNU statement expression crashes compiler
18419	Compiler	io_space_read/write cause compiler assertion with -O
18422	Run Time Libraries	unsigned int64 to float32 routine failing for some values
18436	Run Time Libraries	Some window functions can produce large negative results
18446	Compiler	Compiler fail with Internal Assertion in driver.c:1101
18459	Compiler	Compiler incorrectly calls destructor on uninitialised object
18494	Compiler	___vtbl__ symbols multiply defined when linking C++ libraries
18496	Simulator	IO functions fseek and rewind do not work
18501	Compiler	symbol literal offsets cause errors
18521	Compiler	#pragma all_aligned not working for array index style references
18522	Simulator	compsim ignores breakpoint on first instruction of lsetup loop

18528	Compiler	out of range link error in ".debug_aranges"
18538	Utilities	profblkfn may not read mon.out file correctly
18559	VDK	Initialize on pool creation always false for boot memory pools
18560	VDK	Boot memory pools always created in system heap
18673	Compiler	Assertion failure optimizing code containing shift by literal
18682	Run Time Libraries	The DSP library radix2 FFT functions may read undefined memory
18754	Compiler	bad param address offsets passed to asm statement address inputs
18755	Compiler	near to far not performed for file scope address symbols

VisualDSP++ 3.5 for 16-bit Processors (Updated February 2004) Release Note

The following release note concerns the February 2004 Update to the VisualDSP++ 3.5 for 16-bit Processors 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++ 3.5.
2. The VisualDSP++ GUI's About box, located at "Help" > "About VisualDSP++".
3. In the file …`\System\VisualDSP.ini`, in the `ProductName` key.

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, no additions have been identified.

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. Due to a 64-byte character string limitation on product names in Windows 98 and Windows NT (see problem 18136 below), "Analog Devices" has been dropped from the product name as reported in the Add/Remove Programs control panel. Since this window is sorted alphabetically, look for VisualDSP++ under "V" and no longer under "A". The product is not otherwise impacted.

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

17458	IDDE	can't leave loader page
17500	IDDE	Stepping doesn't follow already opened source
17529	IDDE	Users can edit the routing thread name in the Routing Table
17619	Emulator	UART registers do not update
17792	Installation	In 533 EZKIT Session Flashprogrammer driver file point to 9x
17964	Assembler	.import keyword for *.asm file needs an additional license
18073	Linker	Linker INCLUDE() command doesn't have a search path
18136	Installation	Add/Remove is lost on Win98 and WinNT after January Update
18144	IDDE	IDDE Help About box does not have enough room for Update string
18164	Emulator	Configurator crashes when using node locked license
18181	Loader	Elfloader creates wrong byte count for Init Blocks
18193	VDK	VDK Reschedule ISR is not robust if self-nesting is enabled
18198	VDK	CPLB_ENABLE_CPLBS bit required to be set to enable cplbs
18218	Installation	APEX opening Emulator Interface fails ICETest
18353	Emulator	Server license with one remaining license fails with emulation

VisualDSP++ 3.5 for 16-bit Processors (Updated January 2004) Release Note

The following release note concerns the January 2004 Update to the VisualDSP++ 3.5 for 16-bit Processors 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++ 3.5.
2. The VisualDSP++ GUI's About box, located at "Help" > "About VisualDSP++".
3. In the file …\System\VisualDSP.ini, in the ProductName key.

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 software and example applications for the ADSP-BF561 EZ-KIT Lite are included in this Update. This product's User's Manual has been added to the on-line help system.

Changes to Existing 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.

In this release:

1. The .edt section produced by the updated compiler may be larger than the base release. This is because of additional required information that was omitted and was necessary to resolve various issues. The increased size of this section will only be seen in code that uses C++ exceptions and may result in a link error which had not been seen in the base release.
2. Problem #17921 has now been corrected in the default VDK LDFs, but if your LDF is based on the versions that was provided in the base release, it may need to be updated accordingly. Only users linking in the ETSI libraries with __WORKAROUNDS_ENABLED defined are impacted by this change.

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

16910	IDDE	Problems with "Load Symbols" on running target
17095	Compiler	Compiler does not default to using BSS sections
17191	Emulator Run Time	Core and System Registers aren't resetted on reload.
17212	Libraries Run Time	Non-ANSI symbols always used in math.h
17256	Libraries	Run time header changes PLL frequency
17275	Emulator	Can't connect two HPPCI Emulators on the same machine
17281	Compiler	Compiling code that takes builtin address causes compiler crash.
17454	Assembler	sometimes SHF_ALLOC bit in section header flag not set
17466	Compiler Run Time	Compiler crashes compiling exceptions code
17479	Libraries	The function fir_fr16 truncates result
17563	Compiler	C++ exceptions code loops at runtime built -O
17644	VIDL Compiler	embedded lightweight components can not be used.
17647	VIDL Compiler	embedded component fails to build
17648	VIDL Compiler	embedded lightweight components cannot be build
17664	VIDL Compiler	Testing of out arrays omits to copy back values
17665	VIDL Compiler	test shell component return VCSE_MRESULT code for array checks
17672	Simulator	hardware loops only work when set up using lsetup
17679	Compiler	Compiler fails to convert High->high move to Low->High
17683	Compiler	Circular buffer only set up in conditional path
17684	VIDL Compiler	in_assert and out_assert test shell attributes do not work
17689	VIDL Compiler	test shell component states attributes do not work
17718	Utilities	DSM configuration files are not correct for BF535 EZKIT
17725	VDK	Compilation error due to VDK_Public.h when built-ins disabled
17749	VDK	VDK NoTimeoutError functionality is broken
17767	Compiler Run Time	Compiler performs illegal re-arrangement of saturating add instr
17777	Libraries	Function rfft_fr16 fails to perform in-place operation
17787	Utilities	meminit generates error for sections with SHT_NOBITS and size>0
17800	VDK	VDK NoTimeoutError functionality is broken
17805	Compiler	Hw Loops with breaking Branches not killed
17817	VDK	VDK Tick Period representation changed from double to float
17818	VDK	VDK Tick Period representation changed from double to float
17833	Compiler	Compiler clobbers register before context save in ISR
17858	Emulator	FLAG Registers FL0, FL1, FL2 interact
17859	Linker	PLT entries are generated for 218x DM overlays
17883	Compiler	Compiler clobbers register before context save in ISR
17886	VDK	VDK startup code uses call instead of call.x
17890	Emulator	BF531/2 Memory map incorrectly defined in ADSP-BF531.xml file

17904	Compiler	218x and 219x compilers assert in colour.c:2369 with -O
17905	IDDE	Loader fails if Initialization file path contains spaces
17908	Compiler	Compiler does not mark objects as compiled with profiling
17915	IDDE	Tcl generated menu item crashed IDDE when clicking the menu item
17921	VDK	Blackfin VDK .ldf files sometimes link incorrect ETSI libraries
17933	Compiler	Unexpected exception handler not executed
17934	Compiler	Incorrect C++ dynamic_cast conversion should return NULL.
17935	Emulator	Can not connect to 2191 + unknown
17949	Flash Programmer	ADSP21992 Flash Loader is Inoperative
17958	Compiler	Internal asssertion with -O "Funny jump for DSFill"
17960	Run Time Libraries	adsp-2199x.h include file compiler warnings
17970	Compiler	Compiler debug information produced linker error.
17982	IDDE	& is not supported in folder name and has problem to convert 3.1
17986	Run Time Libraries	Multithreaded srand() function references undefined symbols
18013	Compiler	Network drive resident projects fail to build
18015	Compiler	INTERNAL COMPILER ERROR: Lost half reg
18018	Compiler	Internal compiler error with long VLA index types
18023	Run Time Libraries	C++ Exceptions Library With Workarounds Is Not Safe
18028	Compiler	Compiler raises assertion at eval.c:459
18029	Installation	Flash programmer not installed in 21xx-only installation
18030	Examples	BF535 EZ-KIT pwr_mgmt ex. won't work for si rev 1.0 and later
18037	IDDE	"Stack Size" field no longer displayed for Thread Types
18039	Run Time Libraries	Cache invalidation leave Write commands active
18043	Run Time Libraries	io_space_read and io_space_write not working for non-literals
18047	Linker	No PLIT entries for h/w pages
18061	Emulator	Unable to load program code to external memory
18062	Run Time Libraries	long division results can be incorrect
18086	Compiler	Pointer to non-const handler catches pointer to const exceptions
18093	Compiler	expressions with conversions from floats to ints can be wrong
18094	Compiler	PGO execution counts not collected for some static functions.
18107	Compiler	hard-register asm inputs listed in clobber list result in assert
18119	VDK	"Rebuild All" does not work correctly with VDK