Release Notes for CrossCore Embedded Studio 2.3.0

Release Notes for CrossCore Embedded Studio 2.3.0
June 2016

http://www.analog.com
processor.tools.support@analog.com

Contents

1 Introduction

This document describes the changes for CrossCore Embedded Studio (CCES) 2.3.0. You can find the release notes for older releases in the `docs` sub-directory of your CCES installation.

1.1 Supported Operating Systems

This release of CCES is supported on the following operating systems:

- Windows 7 Professional, Enterprise, or Ultimate (32 and 64-bit)
- Windows 8.1 Pro or Enterprise (32 and 64-bit)
- Windows 10 Pro or Enterprise (32 and 64-bit)

> **Note**
>
> Users may experience User Access Control (UAC) related errors if the software is installed into a protected location, such as `Program Files` or `Program Files (x86)`. We recommend installing the software in a non-UAC-protected location.

1.2 System Requirements

Verify that your PC has these minimum requirements for the CCES installation:

- 2 GHz single core processor; 3.3GHz dual core or better recommended
- 1 GB RAM; 4GB or more recommended
- 2 GB available disk space
- One open USB port

> **Note**
>
> - A faster disk drive or SSD decreases the build time, especially for a large amount of source files. 4GB of RAM or more will substantially increase the performance of the IDE.
> - For proper viewing of documentation under Windows, Internet Explorer 9 or greater is recommended.

1.3 Obtaining Technical Support

You can reach Analog Devices software and tools technical support in the following ways:

- Post your questions in the software and development tools support community at EngineerZone®
- E-mail your questions about software and development tools directly from CrossCore Embedded Studio by choosing Help > Email Support or directly to processor.tools.support@analog.com
- E-mail your questions about processors and processor applications to processor.support@analog.com
- Submit your questions to technical support directly via http://www.analog.com/support
- Contact your Analog Devices sales office or authorized distributor

2 New and Noteworthy

2.1 ADSP-SC57x and ADSP-2157x Support

CCES 2.3.0 provides support for the new ADSP-SC57x and ADSP-2157x products from the SHARC family of DSPs.

- ADSP-SC570
- ADSP-SC571
- ADSP-SC572
- ADSP-SC573
- ADSP-21571
- ADSP-21573

Visit http://www.analog.com/adsp-sc57x-2157x-family for more information on this new line of processors.

Please note there is no support in the functional simulator or the cycle-accurate simulator for ADSP-SC57x or ADSP-2157x in this release. If you need to use a cycle-accurate simulator, you should build your project as an ADSP-SC589 project and use the cycle-accurate simulator that is available for ADSP-SC589. As the SHARC+ cores on 57x and 58x families are identical, any performance analysis results will be applicable to both processor families.

2.2 ADSP-SC583 and ADSP-21583 parts support added

CCES 2.3.0 now includes support for the ADSP-SC583 and ADSP-21583 parts.

2.3 SHARC+ L1 parity errors are now enabled for ADSP-SC58x and ADSP-2158x parts

The startup code for ADSP-SC58x and ADSP-2158x parts when building for revision 1.0 now register and enable a handler for the L1 Parity Error (PARI) interrupt. The supplied handler for PARI is called `_adi_parity_error_detected` and is jumped to directly from the PARI interrupt vector code. The `_adi_parity_error_detected` function uses adi_fatal_error support to report the error details in the CCES console and does not return.

This PARI support is not enabled if building for silicon revisions other than 1.0.

2.4 ADSP-BF70x parts revision 1.1 support added

CCES 2.3.0 includes support for new silicon revision 1.1 of the ADSP-BF70x family of parts. This support includes libraries built for revision 1.1 use that no longer include the workaround for silicon anomaly 19000054 *"Dynamic Branch Prediction During Self-Nested ISRs Causes Unpredictable Results"* (as silicon anomaly 19000054 is fixed in revision 1.1). New loader initcode DXE files built for ADSP-BF706 and ADSP-BF707 EZ-Kits that use the new revision 1.1 parts are also provided.

Silicon revision 1.1 shares the same L2 Utility ROM and CCES support as provided for revision 1.0 silicon.

The default silicon revision for new ADSP-BF70x CCES projects or for the command-line tools if no silicon revision is specified is unchanged and remains revision 1.0.

2.5 Updates to Breakpoint Support

2.5.1 Enable "Toggle Hardware Breakpoint" for ADSP-SC5xx SHARC+ Cores

Hardware Breakpoints can be set by right-clicking on the left margin beside a line of code and choosing "Toggle Hardware Breakpoint" on ADSP-SC5xx SHARC+ Cores.

2.5.2 Restore ADSP-SC5xx Hardware Breakpoints and Watchpoints

Hardware breakpoints on ADSP-SC5xx SHARC Cores are restored after a `Run Menu->Terminate` and relaunch of a Debug session. On ADSP-SC5xx Cortex-A5 Cores, Hardware Watchpoints are now restored as well.

2.5.3 CCES Debugger Emulator disallows setting software breakpoints in delayed slots for SHARC+ cores of ADSP-SC5xx/ADSP-215xx

Setting a software breakpoint in delayed slots (ie. the first two instructions after a (db) instruction) will now result in the debugger emitting a message that the breakpoint cannot be set.

2.5.4 Setting Hardware breakpoint on automatic breakpoint is disallowed

Setting a hardware breakpoint on source line or assembly instruction where there is already an automatic breakpoint set is now disallowed.

2.6 Updates to Target Option Support

2.6.1 Target Options apply to individual ADSP-SC5xx SHARC Cores

Target Options changes are only applied to the specified core. Previously, changing a Target Option on one of the ADSP-SC5xx SHARC Cores would also enforce that change on the other SHARC core.

2.6.2 New Target Option for setting the Reset Address on ADSP-SC5xx SHARC Cores

The user can set the address to which the processor is set after a reset on ADSP-SC5xx SHARC Cores. Enter the reset address by selecting the '`Target`' menu, '`Settings`', '`Target Options`…', and using the '`Reset Address`' text box.

2.7 Support for block FIR filter using FFTA accelerator

Support has been added to the `libfftacc` library to allow implementation of high-performance block FIR filters. This support comprises the following functions:

- `accel_fft_fir_pipe`
- `accel_fft_fir_set_tcbs`

Details of how to use these functions is given in their reference pages in the DSP library documentation, and in the section "Performing a FIR Filter using the FFTA accelerator" in the Online Help.

2.8 Reset functionality on the ADSP-SC5xx/ADSP-215xx

System reset has been added for the following processors:

- ADSP-SC58x silicon revision 1.0 or greater
- ADSP-2158x silicon revision 1.0 or greater
- All ADSP-SC57x and ADSP-2157x

System reset will only occur when specifically loading applications to **all cores** for the specific processor that is under debug. Otherwise a core reset will be performed on all cores that have an application being loaded. All cores that do not have an application being loaded will be untouched and will continue running. Using reset from the menu options in CCES will perform a core reset on the core that is currently in focus.

2.9 Unlocking a secure ADSP-SC5xx processor with OpenOCD

The ADSP-SC5xx processors can be locked in order to protect customer IP. Once a processor is locked, it cannot be accessed externally by an emulator unless a special key is provided, which must also have been programmed into the processor's OTP memory *before* the part is locked. The CrossCore Debugger uses custom board support files to make the access key available to the emulator.

It is now also possible to make the access key available to the OpenOCD emulator. For more details, see the CCES Online Help under:

```
ARM Development Tools Documentation > OpenOCD User's Guide > Supported Targets
```

2.10 Peripheral activity is halted when the emulator halts (ADSP-BF70x)

Peripherals such as the watchdog timer and general-purpose timer will now halt when the emulator halts the processor. Previously the peripherals would continue running even when the emulator was halted.

2.11 ADSP-SC5xx/ADSP-215xx read side effect registers

Certain registers are known to cause side effects when read through the CCES debugger using the Register Browser and Registers windows. Since there is no way for users to stop these registers from being read, we have compiled a list of registers that the debugger will no longer read from the processor when the register window in which they reside is open. The value shown for these registers instead of reading them from the processor is all E's. If there are any children registers for these read side effect registers, their contents are considered NA and should be ignored. The list of registers compiled thus far is as follows:

GICCPU1_INT_ACK
EMAC0_TM_STMPSTAT, EMAC0_LPI_CTLSTAT, EMAC0_MMC_RXINT,
EMAC0_MMC_TXINT, EMAC0_DMA0_MISS_FRM, EMAC0_DMA2_MISS_FRM
EMAC1_MMC_RXINT, EMAC1_MMC_TXINT, EMAC1_DMA0_MISS_FRM
USB0_INTRTX, USB0_INTRRX, USB0_IRQ, USB0_DMA_IRQ, USB0_LPM_IRQ
USB1_INTRTX, USB1_INTRRX, USB1_IRQ, USB1_DMA_IRQ, USB1_LPM_IRQ
FIR0_MACSTAT
IIR0_DMASTAT
DAI0_IRPTL_H, DAI0_IRPTL_L
DAI1_IRPTL_H, DAI1_IRPTL_L
TWI0_RXDATA8, TWI0_RXDATA16
TWI1_RXDATA8, TWI1_RXDATA16
TWI2_RXDATA8, TWI2_RXDATA16

UART0_RBR
UART1_RBR
UART2_RBR

Note that these registers should not be accessed using the Memory Browser.

2.12 New Warning About SLOWLOOP bit in MODE2 Register for ADSP-SC5xx/ADSP-215xx Processors

In CCES 2.2.0, the SLOWLOOP bit(bit 7) in the MODE2 register was set by default in the CCES debugger for the SHARC+ cores of the ADSP-SC5xx/ADSP-215xx processors. Since this can affect timing of F1 active loops, a new warning dialog was created to make users completely aware that this bit was set by default and allow them to clear the bit. The dialog will show when the first user software breakpoint is placed in their SHARC+ core application. There is a separate dialog for each SHARC+ core that will only appear once per debug configuration provided users decide to suppress the dialog after it first appears.

NOTE: Since a software breakpoint is used when doing a ***Step Over*** or ***Step Return*** in CCES, this warning will appear.

Additional documentation regarding the SLOWLOOP bit and debugging F1 active loops can be found in the Online Help under the following topic:

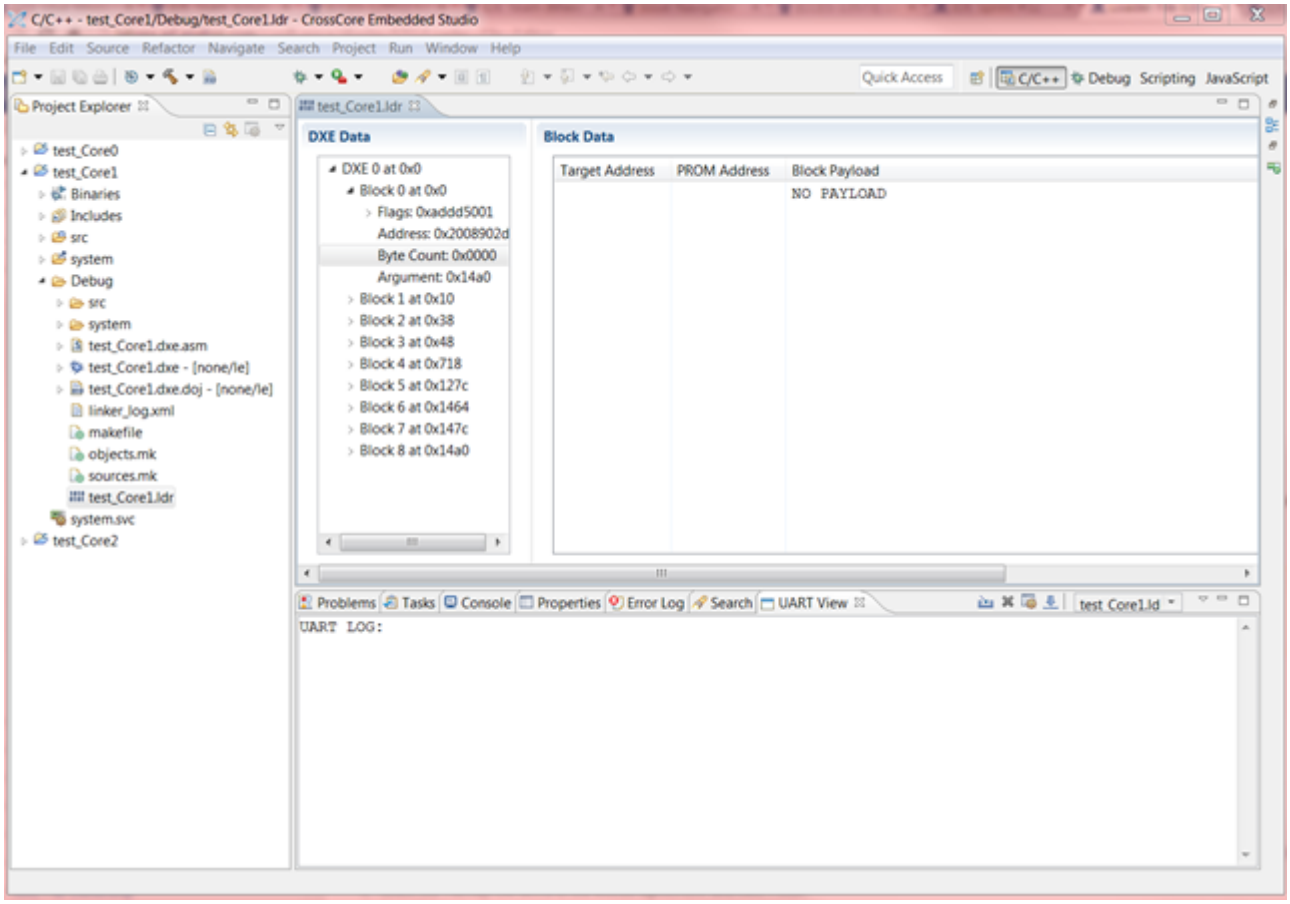
```
Integrated Development Environment -> Debugging Targets -> Debugging SHARC

and SHARC+ targets ->  About Hardware Counter-Based Loops for SHARC+ Cores
```

Additional documentation regarding Hardware Counter-Based Loops can be found at:

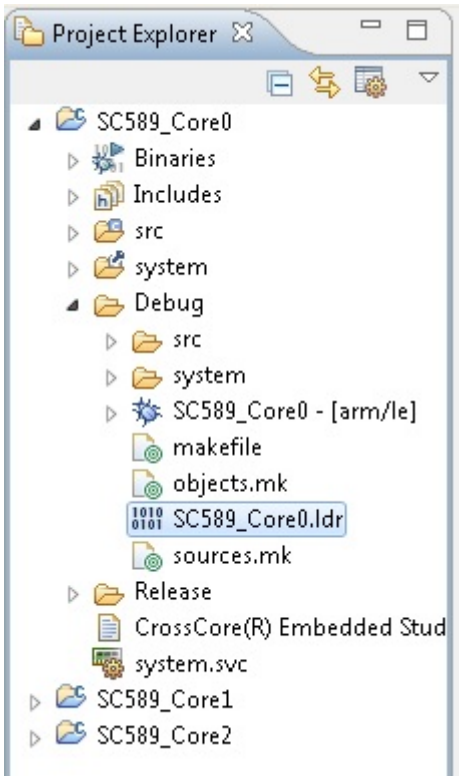
```
Processor Documentation -> SHARC Processor Programming Documentation -> SHARC+ Core

Programming Reference -> Program Sequencer -> Loop Sequencer -> Loop Categories
```

2.13 Loader Editor

The Loader Editor can be used to view the structure and binary contents of a Loader file (.ldr). The Loader Editor comes with a new Console View that can download your Loader file (.ldr) to a target board via UART.



To open the Editor, simply double-click on a Loader file (.ldr) in your Project Explorer or use "Open File..." from the File menu.



2.14 New EZK License Type

A new EZK license type is available. An EZ-KIT license behaves like a full license except that it only supports development with the EZ-KIT board for which it is licensed.

An EZK license restricts building applications to the processor that the EZK license supports (e.g. ADSP-SC589). An EZK license also restricts running and debugging your application to an emulator debug session. The Simulator is not available when using an EZK license.

The currently supported EZ-KIT boards are:

- ADSP-BF706
- ADSP-BF707
- ADSP-SC571
- ADSP-SC573
- ADSP-SC584
- ADSP-SC589

3 Changes That Might Impact Backwards Compatibility

3.1 New ADSP-SC58x Core 0 CRT adi_mmu_Init and adi_osal_Init error checks

Updates have been made to crt-sc58x C runtime startup codes to check if the `adi_mmu_Init` and `adi_osal_Init` calls return an error. If an error is returned from these calls the code branches to `adi_fatal_error,` details are printed in the CCES console when debugging and execution terminates.

3.2 adi_sec_Init changes to ensure the SHARC0 completes SEC initialization before SHARC1 uses the SEC

The `adi_initComponents()` call of `adi_SEC_init()` done in adi_initialize.c for the ADSP-SC5xx and ADSP-215xx parts SHARC cores behaves differently in CCES 2.3.0. The changes ensure that the lowest SHARC core, SHARC0, completes the initialization done in `adi_SEC_init()` before SHARC1 can use the SEC. This means the SHARC1 call of `adi_SEC_init()` will not return until adi_SEC_init() called from SHARC0 has completed.

3.3 Additional linker el2011 errors may be seen for invalid DMC1 uses in SHARC+ LDFs

The CCES SHARC+ parts linker previously did not issue an error for LDFs that attempted to use DMC1/DDR-B memory for the ADSP-SC582, ADSP-SC584 and ADSP-21584 parts that do not have that second DMC. This issue has been corrected in CCES 2.3.0 so it is possible that you may see this error for the first time. A sample of error output is copied below. Correct the problem by deleting uses of the DMC1/DDR-B memory in the LDFs that trigger the error.

```
[Error el2011]  Invalid memory range and/or width for memory 'mem_DMC1_S
Block outside defined memory regions
```

3.4 autocohf and crosscohf functions on SHARC+ parts use SIMD mode

The `autocohf` and `crosscohf` functions on ADSP-SC5xx and ADSP-215xx processors now use SIMD mode by default, which significantly increases their performance. Due to the fact that the floating-point operations are re-ordered, the functions no longer return results that are bit-exact with the old implementations. If necessary, the old behavior can be obtained by building code that calls these functions with the `-no-simd` compiler switch.

3.5 Target Options apply to individual ADSP-SC5xx SHARC Cores

When using a Debug Session created with a previous version of CCES, it will be necessary to re-enter the Target Options for each ADSP-SC5xx SHARC Core.

3.6 Default behavior when connecting to a core with no application being loaded

Previously CCES would halt any core when connecting but not loading an application to it. There has been a change starting in this release that leaves any core running that does not have an application being loaded to it.

3.7 SHARC+ SYSCTL register has been moved from SHDBG to CMMR

The SHARC+ `SYSCTL` register was previously part of the SHDBG registers grouping and has been moved to the CMMR group for CCES 2.3.0. This means that the def header macros for accessing `SYSCTL` have been renamed as shown in the table below. Macro definitions of the previous names are included for backwards compatibility when using the `defSC58*.h` headers or `sys/platform.h` so applications should be unaffected by this change, however we would recommend changing uses of the old names to the new names.

Also note that the register browser location for viewing the SHARC+ `SYSCTL` register has also changed : view `SYSCTL` using the CMMR group in CCES 2.3.0.

Previous macros	New macros

Previous macros	New macros
REG_SHDBG0_SYSCTL	REG_CMMR0_SYSCTL
BITP_SHDBG_SYSCTL_SMCFIFO	BITP_CMMR_SYSCTL_SMCFIFO
BITP_SHDBG_SYSCTL_IMDWBLK3	BITP_CMMR_SYSCTL_IMDWBLK3
BITP_SHDBG_SYSCTL_IMDWBLK2	BITP_CMMR_SYSCTL_IMDWBLK2
BITP_SHDBG_SYSCTL_IMDWBLK1	BITP_CMMR_SYSCTL_IMDWBLK1
BITP_SHDBG_SYSCTL_IMDWBLK0	BITP_CMMR_SYSCTL_IMDWBLK0
BITP_SHDBG_SYSCTL_EIVT	BITP_CMMR_SYSCTL_EIVT
BITP_SHDBG_SYSCTL_IIVT	BITP_CMMR_SYSCTL_IIVT
BITP_SHDBG_SYSCTL_SRST	BITP_CMMR_SYSCTL_SRST
BITM_SHDBG_SYSCTL_SMCFIFO	BITM_CMMR_SYSCTL_SMCFIFO
BITM_SHDBG_SYSCTL_IMDWBLK3	BITM_CMMR_SYSCTL_IMDWBLK3
BITM_SHDBG_SYSCTL_IMDWBLK2	BITM_CMMR_SYSCTL_IMDWBLK2
BITM_SHDBG_SYSCTL_IMDWBLK1	BITM_CMMR_SYSCTL_IMDWBLK1
BITM_SHDBG_SYSCTL_IMDWBLK0	BITM_CMMR_SYSCTL_IMDWBLK0
BITM_SHDBG_SYSCTL_EIVT	BITM_CMMR_SYSCTL_EIVT
BITM_SHDBG_SYSCTL_IIVT	BITM_CMMR_SYSCTL_IIVT
BITM_SHDBG_SYSCTL_SRST	BITM_CMMR_SYSCTL_SRST

3.8 Assembler section qualifier warning ea1114 has been deleted

The ea1114 Blackfin and SHARC assembler warning has been deemed to be unnecessary so it has been deleted in CCES 2.3.0. Existing CCES projects may have been setup to suppress ea1114 and this will cause the following warning message when built using CCES 2.3.0:

```
[Warning ea1195] "test.asm":1 This number does not represent a known message id : 1
```

Fix the problem by removing the suppression of ea1114.

3.9 Avoiding a link error due to an overly long command line

To avoid problems linking a SHARC/ADSP-SCxxx/Core 0 (ARM) project that contains a lot of source files and produces a very long linker command line, the IDE now writes the list of object files to a input file. The IDE specifies the input file on the linker command line rather than the list of object files. This change could result in the order of the object files that were created from source files in your project being different than before and it has the potential to expose a problem with projects that happened to work because of the order the object files were linked.

4 Known Issues

4.1 Hidden SLOWLOOP window may cause CCES to appear hung when stepping, launching, or setting breakpoints with Emulator (CCES-15438)

When user breakpoints are set, or when launching or executing after user breakpoints have been set, or when executing a 'Step Over/Return', a 'User Breakpoint Enabled' window is displayed to alert the user the SLOWLOOP bit is being set and allow the user to opt out. However, sometimes the window will not be visible because it is behind CCES. CCES will appear unresponsive until the window is dismissed. Click on the Windows taskbar at the bottom of your screen on the CCES icon to bring up the ‘User Breakpoint Enabled’ window and click ‘OK’ in order to continue with CCES. Click on 'Do not display this box again' to avoid this.

4.2 Hardware Breakpoints must be disabled prior to Restart for Multi-Core Processors (CCES-15525)

Sometimes hardware breakpoints are not restored properly after a Restart operation. The breakpoint is set multiple times. The full number of hardware breakpoints then cannot be set, and the existing breakpoint cannot be cleared properly.

If this is seen, Hardware Breakpoints should be disabled prior to a Restart operation and then re-enabled after the Restart. Alternatively, the Relaunch operation can be used.

For the latest anomalies please consult our Software and Tools Anomalies Search page.