CrossCore® Development Tools by Analog Devices

Release Notes for CrossCore Embedded Studio 2.10.0

1 Table of Contents

2 Introduction

This document describes the changes for CrossCore Embedded Studio (CCES) 2.10.0. You can find the release notes for older releases in the `docs` sub-directory of your CCES installation as well as an Installation Guide which will help you install this release.

3 New and noteworthy

3.1 ADSP-2159x/ADSP-SC59x Processors Support

CCES 2.10.0 provides support for the new family of ADSP-2159x/ADSP-SC59x processors. These newly supported parts are:

Non-Automotive Models	Automotive models
ADSP-21591 ADSP-21593 ADSP-21594 ADSP-SC592 ADSP-SC594	ADSP-21591W ADSP-21593W ADSP-21594W ADSP-SC591W ADSP-SC592W ADSP-SC594W

The support provided includes:

- CCES project build and debug capabilities.
- Boards supported
    - EV-21593-SOM
    - EV-SC594-SOM
    - EV-SOMCRR-EZKIT
- System services (SSL) and device drivers (DD) 3.0
- Functional and cycle-accurate simulators.

The following Device Drivers and System Services modules are supported in this release for ADSP-SC59x:

Device Drivers

- ASRC
- CAN-FD
- CRC
- Crypto (PKA, PKTE ,TRNG)
- EMAC
- FIR
- HADC
- IIR
- Link Port
- MLB
- OSPI
- EPPI
- Rotary Counter
- SPDIF Rx
- SPDIF Tx
- SPI
- SPORT
- TMU
- TWI
- UART

System Services

- DAI
- DMA (MDMA, EMDMA)
- GPIO
- MEPU
- PCG
- PDM
- PWR
- RCU
- SMPU
- SPU
- STDIO
- SWU
- TMR
- TRU
- WD

SSL/DD support is now provided for newly-added perhipherals CAN-FD, PDM and EMAC on ADSP-SC59x.

3.2 Eclipse Upgraded to Version 4.9

CrossCore Embedded Studio's Eclipse Platform has been upgraded to Photon (4.9) and its C/C++ Development Tooling (CDT) has been upgraded to 9.5, from versions 4.7 and 9.3 respectively.

New and Noteworthy in Eclipse 4.8

New and Noteworthy in Eclipse 4.9

New and Noteworthy in CDT 9.4

New and Noteworthy in CDT 9.5

3.3 GNU toolchain for the Cortex A5 cores on ADSP-SC5xx parts updated to 10.2.0.

The GCC toolchain used for Cortex A5 cores on the ADSP-SC5xx parts has been changed from 7.2.1 to 10.2.0.

The following pages outline the changes to the toolchain in this upgrade:

- https://gcc.gnu.org/gcc-8/changes.html
- https://gcc.gnu.org/gcc-9/changes.html
- https://gcc.gnu.org/gcc-10/changes.html

In addition, the following pages give advice on any changes needed to your source code for the new release:

- https://gcc.gnu.org/gcc-8/porting_to.html
- https://gcc.gnu.org/gcc-9/porting_to.html
- https://gcc.gnu.org/gcc-10/porting_to.html

3.4 New adi_signtool utility

A new `adi_signtool` utility has been added to CCES. Use `adi_signtool` to sign and encrypt boot streams needed for secure booting on parts such as the ADSP-SC59x and ADSP-2159x. See "CrossCore® Embedded Studio 2.10.0 > Blackfin® Development Tools Documentation > Loader and Utilities Manual > Utilities > adi_signtool - Sign and Encrypt Boot Streams for Secure Booting" in the CCES Online Help for further details.

3.5 DMC PHY Calibration issue workaround

The workaround for DMC PHY Calibration issue 20000117 is incorporated into the initcodes and preloads for ADSP-SC59x/ADSP-2159x parts.

3.6 -no-path-in-file-macros compiler switch

The `cc21k` and `ccblkfn` compilers support a new command-line switch, `-no-path-in-file-macros`, which causes the `__FILE__` and `__BASE_FILE__` macros to expand to the source filename without its path.

3.7 New data formats supported in Memory Browser and Plot View

For SHARC+ parts, new data integer and fractional formats are supported with an explicit bit-size in the Memory Browser and Plot View. These are:

- `Signed Integer 8 bit`
- `Signed Integer 16 bit`
- `Signed Integer 32 bit`
- `Unsigned Integer 8 bit`
- `Unsigned Integer 16 bit`
- `Unsigned Integer 32 bit`
- `Signed Fractional 8 bit`

- `Signed Fractional 16 bit`
- `Signed Fractional 32 bit`
- `Unsigned Fractional 8 bit`
- `Unsigned Fractional 16 bit`
- `Unsigned Fractional 32 bit`

These allow a value to be displayed or plotted using the given format, regardless of the memory space alias of the address used to reference the value.

The existing unsized types, `Signed Integer`, `Unsigned Integer`, `Signed Fractional`, `Unsigned Fractional` and `Hexadecimal` continue to display the value of the size of one addressable unit in memory (so, for example, when showing a value at a word-address, a 32-bit value is displayed, while when showing a value at a byte address, an 8-bit value is displayed).

These new formats are also available when dumping or filling memory using the Memory Browser.

3.8 Handling of stride in Plot View and Fill/Dump Memory

The Online Help has been updated to correctly reflect the meaning of the *stride* setting in the Plot View and in the Memory Browser's Fill and Dump Memory functionality. The stride is specified in number of data values, rather than addressable units.

In addition, an issue has been fixed whereby using a non-unit stride could corrupt the last two values printed out when using Dump Memory.

3.9 New library inflection point for ADSP-21467 and ADSP-21469, silicon revision 0.2

A new version of the runtime libraries is now being used for ADSP-21467 and ADSP-21469 parts with silicon revision 0.2. This library does not include software workarounds for silicon anomalies which do not apply to this silicon revision. This library will therefore give better performance for these parts.

3.10 FFTA Accelerator no longer terminates with error on underflow or overflow

Prior to this release, a computational underflow or overflow that occurred during execution of an FFT operation on the FFTA accelerator would cause the accelerator APIs to give fatal error and terminate the FFT operation. This is no longer the case, and the FFT operation will now complete even when an underflow or overflow occurs. Subsequent to the completion of the FFTA operation, the `adi_fft_GetHWErrorStatus` API can be used to find out if an underflow or overflow occurred.

3.11 LDFs for ADSP-214xx parts now make use of first 4MB of SDRAM

The LDFs provided with CrossCore Embedded Studio did not previously make use of the first 4MB of SDRAM memory. This has now been fixed.

4 Changes That Might Impact Backwards Compatibility

4.1 Elfloader -ghc (Global Header Cookie) switch removed

The elfloader's `-ghc` switch did not function correctly and could result in undefined behavior. It has been removed. Please do not use the switch; the elfloader will now always use the correct default value in bits 31-28 of the global header.

4.2 Silicon revision 'none' behaves as default silicon revision

The silicon revision 'none' option, which prevents workarounds for silicon errata being applied, is no longer supported. The consequence of these changes are that silicon anomaly workarounds necessary for the default silicon revision will be applied, and the code will be linked against libraries built for the default silicon revision. Libraries built for silicon revision 'none' have also been removed from CrossCore Embedded Studio.

- When a project that specifies silicon revision 'none' is opened in CCES 2.10.0, it will automatically be updated to target the default silicon revision for the target part. A warning that this has been done will appear in the CCES Problems View.
- If the command-line tools `-si-revision none` switch is used, a warning will be emitted and the default silicon revision for the target processor will be used instead.

If you wish to build a source file without workarounds enabled, please use the `-no-workaround all` compiler switch. If you wish to enable only specific workarounds, add these explicitly to the command line using the `-workaround` switch. For example

```
cc21k -proc ADSP-SC589 -no-workaround all -workaround 20000069 test.c
```

will only work around the 20000069 anomaly.

4.3 Cortex-A GCC defaults to -fno-common

As of version 10, GCC defaults to enabling option `-fno-common`. Previously, variable declarations without extern qualifier or initializer at file scope (for example `int x;`) resulted in tentative definitions. This meant that there could be multiple such definitions of the same variable in different compilation units, for example when a header containing such a definition was included in multiple source files. With this change, such definitions are no longer marked as tentative and hence will result in multiply-defined symbol errors when linking.

Affected variable declarations should be corrected by adding the `extern` qualifier. Alternatively, switch `-fcommon` can be used on the compiler command line or in the compiler additional options of a CCES project's tool settings to restore the old behavior.

5 Known Issues

5.1 PKTE operations in AUTO and TCM mode fail on ARM (Core0).

The PKTE crypto driver for ADSP-SC59x and ADSP-2159x operates correctly on SHARC cores but fails for the ARM core 0 on ADSP-SC59x in AUTO and TCM modes.

5.2 OpenOCD does not support the EV-SOMCRR-EZKIT On-board Debug Agent

Currently OpenOCD only supports use of an ICE-1000 or ICE-2000 for debugging.

5.3 Separately installed Add-ins are not loaded correctly by CCES

Add-ins that are installed separately from CrossCore Embedded Studio are not properly loaded ("recognized"). Add-ins that are bundled with CCES (Internal Add-ins) such as Startup Code/LDF and those Add-ins that are installed via Analog Devices' External Update Site, such as the SSL/DD configuration UI, are properly loaded ("recognized") by CCES.

5.4 Debugger support for static const members

The IDDE does not display accurate debug information for `static const` class members. The values that are displayed in the expressions window are incorrect, and they may appear to change as the values of other struct members are modified.

5.5 Accessing SPI flash Memory Space Could Cause Hang

For the ADSP-2156x, ADSP-SC59x, and ADSP-2159x processors, accessing SPI flash memory (0x60000000 - 0x80000000) before properly configuring this region of memory can cause a hang in CCES.