CrossCore® Development Tools by Analog Devices

Release Notes for CrossCore® Embedded Studio 3.0.3

1 Table of Contents

2 Introduction

This document describes the changes for CrossCore Embedded Studio (CCES) 3.0.3. 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.

For information on Linux please refer to the general Linux documentation Linux for ADSP-SC5xx Processors [Analog Devices Wiki]. Other useful links for CrossCore and Linux Development are:

- http://www.analog.com/cces-quickstart
- EngineerZone > Processors and DSP > Software and Development Tools > CrossCore Embedded Studio and Add-ins
- EngineerZone > Processors and DSP > Software and Development Tools > CrossCore Embedded Studio and Add-ins > tags > CCES
- EngineerZone > Processors and DSP > Software and Development Tools > CrossCore Embedded Studio and Add-ins > tags > CCES 3.0.3
- How to debug SHARC cores in CCES while running Linux
- Configuring System Memory for the ADSP-SC5xx When Using Linux and SHARC Applications

3 New and Noteworthy

3.1 Improvements to Debug Configurations for Emulation

CCES 3.0.3 has has the following improvements to debug configuration support:

- CrossCore Group Session Wizard has added support to allow you to configure OpenOCD remote target parameters.
- CrossCore debug configuration session tab now has settings to allow for resetting the target board previously only available while debugging on menu "Target >Debug > Reset target".
- CrossCore debug configuration session tab can now specify the JTAG maximum speed supported by the ICE being used.

3.2 Initcode and preload updates

CCES 3.0.3 has has the following improvements to preloads for debugging and initcodes for booting:

- Improvements to the 4-byte SPI addressing ADSP-SC83x/ADSP-2183x initcode support enabled by setting `CONFIG_ENABLE_4_BYTE_ADDRESSING` to allow access to the complete range of flash.
- Added support to allow for a build of ADSP-21569 family parts initcodes and preloads for Consumer and Industrial that use silicon revision 0.0 and require the DCLK for SDRAM to be set to 500 MHz.
- Added an adi_pwr configurations to provide 600 MHz CCLK to support new ADSP-21560/W parts speed grades.
- Maximum DDR frequency for ADSP-SC83x/ADSP-2183x changed to 800 MHz and write levelling disabled for improved reliability.
- ADSP-SC58x/ADSP-2158x support changed to avoid unnecessary CGU1 setup when configured for 500 MHz CCLK but not using SDRAM.
- Added support for revision C of EV-21568-SOM via new `EV_21568_SOM_REVC` macro to use a different 25 MHz CLKIN.
- For ADSP-21568, Preload Support is added for NOR flash at 166MHz and HyperRAM initialization at 70MHz.
- For ADSP-SC83x, Preload Support is added for Hyperflash initialization along with Phy training at 125MHz.

3.3 System Services and Device Drivers

CCES 3.0.3 has has the following improvements for SSLDD :

- ADSP-SC83x family of processors : Static Analysis has been done and updated SSLDD sources/libraries have been provided with the CCES-3.0.3 release.
- SPI Driver : New API's adi_spi_CoreSoftAssertWrite() and adi_spi_CoreSoftAssertRead() are provided as part of the SPI driver which do not disable SPI during a transfer. This facilitates Software/GPIO based assertion and de-assertion of chip select.
- ADSP-21568 xSPI driver : On EV-21568-SOM board, the HyperRAM supports up to 70 MHz, with both read/write operations.

3.4 Updated ADSP-SC83x and ADSP-2183x Toolchain for SHARC-FX

The Xtensa toolchain for ADSP-SC83x and ADSP-2183x parts support is updated in CCES 3.0.3, features in this update are:

- Xtensa Software Tools for Linux systems are compiled using GNU GCC version 9.3 in this release.
- In this release, GNU binary tools for Xtensa Tools have been updated to GNU Binutils version 2.39.
- ISS/XTMP/XTSC simulation libraries compiled with GNU GCC versions 6.3 and 12.4 for Linux systems.
- ISS/XTMP/XTSC simulation libraries compiled with MSVC versions 2015, 2017, 2019, and 2022 for Windows.
- GCC: XTSC libraries compiled with GCC versions 4.8 or 11.2.

Bugfixes:

- A compiler crash when attempting to build the TFLM library in Release configuration with optimization **has been fixed**.
- The xt-as assembler could fail with a message "Fatal error: cannot find suitable trampoline" when trying to assemble very large input files **has been fixed.**
- The xt-clang compiler could crash with an assertion failure `Expression: EltTy && "Can't get a pointer to <null> type!"" **has been fixed.**
- The xt-clang compiler could crash with a message *** Bad machine code: Virtual register defs don't dominate all uses *** **has been fixed.**
- A bug in xos_thread_abort() caused an internal fatal error if called on a thread that was sleeping or executing a timed wait. **This has been fixed.**
- The XT-CLANG compiler might remove memory accesses performed through pointers to two different TIE ctypes when one of the pointers points to a struct field of type X and the other points to a different struct that also includes a field of type X. **This has been fixed.**
- A bug in the xos_system_check() and xos_thread_check() functions resulted in an error being reported on a thread waiting on an event, when no error condition existed. **This has been fixed.**
- A bug in xthal_core_restore() could result in an incorrect stack pointer on XEA3 systems. **This has been fixed**
- Xtensa NX Performance Counter Inaccuracy for Instruction Cache Misses **has been fixed.**
- Xtensa NX Write-After-Write Hazard In Branch Target Buffer Test Instruction **has been fixed.**
- Xtensa NX L1V Tag Refills Store Incorrect Check Bit Data **has been fixed.**

3.5 xSPI as SRAM Support for ADSP-21568 Family Parts

The Startup/LDF LDF settings now have an option to include memory segments and input sections for xSPI memory device enabled for SRAM in the generated app.ldf file. The same support is added to the non-generated LDFs vis the following macros which can be defined when linking:

** USE_XSPI_SRAM
** Makes xSPI SRAM available as addition external memory.
** USE_L2_HEAP, USE_XSPI_SRAM_HEAP
** Use L2/xSPI SRAM for dynamic heap allocation rather than L1 default. Should

** result in a much larger heap being available but can have a significant
** performance impact.
** USE_L2_STACK, USE_XSPI_SRAM_STACK
** Use L2/xSPI SRAM for system stack rather than L1 default. Should allow for
** much larger stack but likely to have a significant performance impact.

CCES defaults to support the 256MB SRAM for the xSPI device on the EV-21568-SOM revision C board starting at 0x60000000.

Applications using this xSPI as SRAM support will rely on the configuration that is done by an initcode or preload built with the CONFIG_XSPI_RAM macro in config.h source for these projects being changed from 0 to 1. The default initcode and preload executables from the CCES installation do not have this support enabled.

This LDF and initcode/preload support is available for ADSP-21560, ADSP-21561, ADSP-21564 and ADSP-21568 parts and their W-part Automotive variants.

3.6 Command Line Device Programmer (CLDP) Updates for ADSP-SC573, ADSP-SC584 and ADSP-SC589

For multicore parts `-core` is used to specify the core[0, 1, 2, … n] on which the device programmer interface application (dpia) runs. For ADSP-SC5xx SHARC+ based parts normally `-core 1` and `-driver <path/filename>` switches are required and the driver available from the BSP install. Alternatively for ADSP-SC573, ADSP-SC584 and ADSP-SC589 parts it is now possible to use `-core 0` with `-driver <path/filename>` to select flash programming to be done using OpenOCD. For this new OpenOCD support the files are included within `<CCES 3.0.3 install folder>/ARM/openocd/share/openocd/scripts/board/flash_algorithms` folder.

Example commands for ADSP-SC573 using this new support are shown below:

```
# erase the ADSP-SC573 EZ-KIT SPI flash via ICE-1000 using OpenOCD
cldp -proc ADSP-SC573 -cmd erase -erase all -core 0 -driver "c:/analog/cces/
3.0.3/ARM/openocd/share/openocd/scripts/board/flash_algorithms/
57x_w25q128fv_spi.out" -emu 1000 -board adspsc573_ezkit.cfg

# program SC573_bin.ldr into the  ADSP-SC573 EZ-KIT SPI flash via ICE-2000 using
OpenOCD
cldp -proc ADSP-SC573 -cmd prog -erase affected -core 0 -driver "c:/analog/cces/
3.0.3/ARM/openocd/share/openocd/scripts/board/flash_algorithms/
57x_w25q128fv_spi.out" -board adspsc573_ezkit.cfg -emu 2000 -format bin -file
SC573_bin.ldr
```

Example commands for ADSP-SC584 using this new support are shown below:

```
# erase the ADSP-SC584 EZ-KIT SPI flash via ICE-1000 using OpenOCD
cldp -proc ADSP-SC584 -cmd erase -erase all -core 0 -driver "c:/analog/cces/
3.0.3/ARM/openocd/share/openocd/scripts/board/flash_algorithms/
58x_w25q128fv_spi.out" -emu 1000 -board adspsc584_ezbrd.cfg

# program SC584_bin.ldr into the ADSP-SC584 EZ-KIT SPI flash via ICE-2000 using
OpenOCD
cldp -proc ADSP-SC584 -cmd prog -erase affected -core 0 -driver "c:/analog/cces/
3.0.3/ARM/openocd/share/openocd/scripts/board/flash_algorithms/
58x_w25q128fv_spi.out" -board adspsc584_ezbrd.cfg -emu 2000 -format bin -file
SC584_bin.ldr
```

Example commands for ADSP-SC589 using this new support are shown below:

```
# erase the ADSP-SC589 EZ-KIT SPI flash via ICE-1000 using OpenOCD
cldp -proc ADSP-SC589 -cmd erase -erase all -core 0 -driver "c:/analog/cces/
3.0.3/ARM/openocd/share/openocd/scripts/board/flash_algorithms/
58x_w25q128fv_spi.out" -emu 1000 -board adspsc589_ezbrd.cfg

# program SC589_bin.ldr into the ADSP-SC589 EZ-KIT SPI flash via ICE-2000 using
OpenOCD
cldp -proc ADSP-SC589 -cmd prog -erase affected -core 0 -driver "c:/analog/cces/
3.0.3/ARM/openocd/share/openocd/scripts/board/flash_algorithms/
58x_w25q128fv_spi.out" -board adspsc589_ezbrd.cfg -emu 2000 -format bin -file
SC589_bin.ldr
```

4 Known Issues

4.1 Programming Flash for ADSP-SC83x/ADSP-2183x Family

When programming flash for any of the ADSP-SC83x or ADSP-2183x family of processors use `-board` with either adsp**21**835w_ev_**som**.cfg or adsp**21**835w_ev_**ezkit**.cfg depending on if only a SOM is being used or a SOM connected to the carrier board. Do not use either of the adsp**sc**835w_ev_**som**.cfg or adsp**sc**835w_ev_**ezkit**.cfg files when programming flash.

In the CrossCore SHARC-FX Device Programmer settings of a CCES project to build a loader file default CLDP switches will be included if not user specified. Replacements to the default switches shown should be added as an additional option setting.

Default settings :

```
 -driver ${CCES}ARM/openocd/share/openocd/scripts/board/flash_algorithms/
2183x_flash.dxe -board adsp21835w_ev_som.cfg
```

4.2 Backward Compatibility for ADSP-SC83x/ADSP-2183x/ ADSP-21568 xSPI Projects

xSPI driver has been updated with certain Enums/Structures/Macros/API for ADSP-SC83x/ADSP-2183x/ ADSP-21568 family of processors. The xSPI driver static configuration "adi_xspi_config_SC8xx.h/ adi_xspi_config_2156x.h" files have been updated.

Any ADSP-SC83x/ADSP-21568 family projects with the xSPI driver added using CCES-3.0.0/CCES-3.0.1/ CCES-3.0.2 may get certain warnings/errors while importing these examples projects in CCES-3.0.3. In case of any warnings, please refer to updated adi_xspi_config_SC8xx.h & adi_xspi_config_2156x.h files and update the application accordingly.

Steps to update your "adi_xspi_config_SC8xx.h/adi_xspi_config_2156x.h" are:

- Take a backup of xSPI static configuration "adi_xspi_config_SC8xx.h" file at "$project/ system/drivers/xspi"
- Uninstall the xSPI addin from system.svc
- Re-install the xSPI add from system.svc
- Update the macros in the "adi_xspi_config_SC8xx.h/adi_xspi_config_2156x.h" file as per the backed up file.

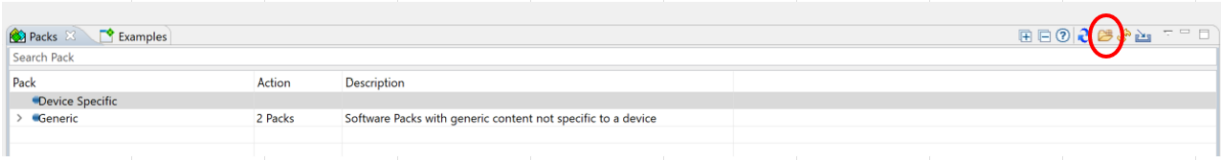
4.3 ADSP-SC83x/ADSP-2183x xSPI boot fails booting when the initcode is used

xSPI boot will fail if the initcode with xSPI PHY training is used in the CrossCore SHARC-FX loader settings.

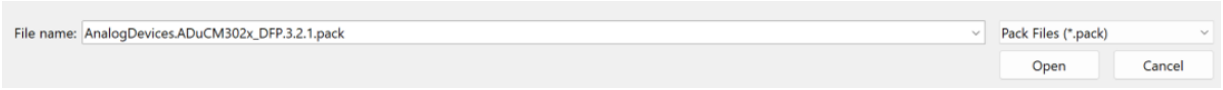
4.4 Unable to install Analog Devices Device Family Pack (DFP) using command-line interface and "Check for Updates on Web" button

Users may encounter an issue where the Device Family Pack (DFP) fails to install or is inaccessible through the CrossCore Embedded Studio - CMSIS Pack Manager or command-line interface (CLI).

Workaround:
An alternative method is available:
• Download the required DFP manually from the Keil website.
• Import the Device Family Pack (DFP) into the CrossCore Embedded Studio - CMSIS Pack Manager manually thru "Import Existing Packs" button.



• Open the Device Family Pack (DFP) and proceed to install.



• Refresh to check the changes.

4.5 ICE-2000 Maximum JTAG Frequency

The maximum JTAG frequency for the ICE-2000 is 46 MHz but it may not work for all processors. Most ADI processors will work at a JTAG frequency up to ~30 MHz.

4.6 ICE-2000 Incorrect Default JTAG Frequency

The ICE-2000 default JTAG frequency should be 9 MHz but is incorrectly defaulting to 5 MHz. This can easily be updated when creating the debug configuration.