Release Notes for CrossCore Embedded Studio 3.0.1

Release Notes for CrossCore Embedded Studio 3.0.1
April 2024

http://www.analog.com

1 Table of Contents

2 Introduction

This document describes the changes for CrossCore Embedded Studio (CCES) 3.0.1. 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 3.0.1
- 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 ADSP-21568 Family Processors Support

CCES 3.0.1 provides support for a new family of SHARC+ Single Core DSP parts. These newly supported parts are:

Non-Automotive Models	Automotive models
ADSP-21560 ADSP-21561 ADSP-21564 ADSP-21568	ADSP-21560W ADSP-21561W ADSP-21564W ADSP-21568W

CrossCore Embedded Studio includes usual support for the new parts, such as:

- CCES project build and debug capabilities.
- Initcodes and SHARC preloads for initialization.
- System services (SSL) and device drivers (DD) 3.0
- Functional and cycle-accurate simulators.

3.1.1 System Services and Device Drivers for the new parts

The following Device Drivers and System Services modules are supported in this release for the ADSP-21568 family of parts.

Device drivers	System Services
• ASRC • CRC • Crypto (PKA, PKTE ,TRNG) • FIR • IIR • MLB (automotive parts only) • Rotary Counter • SPDIF Rx • SPDIF Tx • SPI • SPORT • TWI • UART • xSPI	• DAI • DMA (MDMA, EMDMA) • GPIO • MEPU • PCG • PWR • RCU • SMPU • SPU • STDIO • TMR • TRU • WDT

For more information regarding device drivers and system services see the following CCES help topics:

- CrossCore® Embedded Studio 3.0.1 > System Run-Time Documentation > System Services and Device Drivers User Guide

- CrossCore® Embedded Studio 3.0.1 > System Run-Time Documentation > System Services and Device Drivers
- CrossCore® Embedded Studio 3.0.1 > System Run-Time Documentation > System Services and Device Drivers > ADSP-21568 Family (SHARC+ Core) API Reference

3.1.2 Evaluation Boards

The following evaluation boards are available for support of the new parts.

- EV-21568-SOM
- EV-SOMCRR-EZLITE

The examples BSP for these evaluation boards will be available soon.

3.2 Programming xSPI Hyperflash on ADSP-SC/2183x SOM

Flash commands via xSPI are now supported for SHARC-FX processors in cldp for S71KS512SC0 HyperFlash™ on the ADSP-SC/2183x SOMs. The main specific condition that must be met is the device specifier must be "hyperflash".

3.3 ICE-1500 support

Support for a new Emulator ICE-1500 target is enabled in CCES 3.0.1.

3.4 OTP service support extended

The prebuilt libssl system services libraries for ADSP-SC594 and ADSP-21569 processor families now contain the OTP (One Time Programmable memory) service. See the API references for these parts in the CCES help to get further details of the available adi_otp APIs.

3.5 New SHARC Compiler Warning for Potential Instance of Anomaly 20000129

Calls of the `adi_rom_otp_pgm()` and `adi_rom_otp_pgm_data()` ROM APIs may now trigger a compiler warning indicating a possible issue related to silicon anomaly 20000129 (OTP Program API fails to program OTP memory correctly in presence of a leaky bit). An example of the warning is shown below. The suggestion to address the issue is use of the OTP services replacing use of these two ROM APIs. Alternatively you could choose to suppress the warning either in the project settings compiler additional options using "-Wsuppress 3840" or by adding `#pragma diag(suppress:3840)` in the source. See the latest "Silicon Anomaly Sheet" document for your part available from www.analog.com for more information on silicon anomaly 20000129 .

```
"ss.c", line 4: cc3840: {D} warning: Potential instance of anomaly 20000129 in
          call to "adi_rom_otp_pgm" function - OTP Program API fails to
          program OTP memory correctly in presence of a leaky bit. Please use
          driver code instead to program the OTP.
```

3.6 Workaround for ADSP-2183x/ADSP-SC83x Anomaly 25000013

A workaround for silicon anomaly 25000013 is added to the default exception handler provided in CCES 3.0.1. This workaround handles incorrectly raised memory errors for correctable errors on L1 instruction RAM.

If you want to override the CCES exception handler, you will need to add a workaround for this anomaly in your exception handler. The following code gives the basis of a handler that has the required workaround

```
#include <stdbool.h>
#include <xtensa/hal.h>
#include <xtensa/xtruntime.h>
#include <xtensa/tie/xt_memerr.h>

void mem_error_handler_wa_25000013(ExcFrame* ef)
{
  static bool reread_in_progress = false;
  /* Test for loop on hard error exception (0x646). This may be indicative of the anomaly. */
  if ((ef->exccause & EXCCAUSE_FULLTYPE_MASK) == 0x646 && !reread_in_progress) {
    /* Find the address of the erroneous read, retry that read with an instruction
    ** sequence that avoids the erratum.  Deal with the possibility that a hard
    ** error will cause another exception.
    */
    uint32_t addr = XT_RSR_MEVADDR(); /* Obtain the failing address */
    addr &= 0xfffffffc;               /* Align to a 32-bit aligned address */
    reread_in_progress = true;        /* Set reread_in_progress */
    /* Reread the address, under circumstances that cannot cause the anomaly.
    ** This will correct the errors if they can be corrected.
    */
    __asm__ __volatile__("l32i %0,%0,0\n\tnop\n\tnop\n\t" :: "a"(addr));
    reread_in_progress = false;
    return;
  }
  reread_in_progress = false;
  /* From here, a real error has been detected - insert your own code */
}

xtos_set_exception_handler(EXCCAUSE_HARDWARE, (xtos_handler) mem_error_handler_wa_25000013, 0);
```

4 Changes That Might Impact Backwards Compatibility

4.1 Possible Out of Memory Errors Linking ADSP-SC83x ARM Cortex-M33 Core 2 Projects

Using the CCES 3.0.0 provided linker (.ld) scripts for ADSP-SC83x ARM Cortex-M33 core 2 projects can result in successfully linking what is a bad executable because the stack and heap are defined so they overlap data or stretch outside available memory. These .ld scripts have been updated for CCES 3.0.1 to correct this problem and there is a possibility that projects that build successfully using CCES 3.0.0 may fail to link now with out of memory errors. The correction for these errors is to adjust the linker script used to make more memory available or to make the application smaller so that it fits in the available memory.

5 Known Issues

5.1 Export of All Registers Fails for ADSP-21568 Family Parts

The register browser support to export the values of all registers is failing when used for the new ADSP-21568 family parts.

5.2 ADSP-SC83x_DFP Version Compatibility Update Needed

The built-in version of the ADSP-SC83x_DFP has been updated to 1.10.1. As a result, dependencies on the include directory might be affected when importing a project created from CrossCore Embedded Studio 3.0.0 that uses ADSP-SC83x_DFP 1.10.0. Ensure to update the following dependencies accordingly.

- Project Properties > Settings > CrossCore GCC ARM Embedded Assembler > Preprocessor > Additional include directories (-I)
- Project Properties > Settings > CrossCore GCC ARM Embedded C Compiler > Preprocessor > Additional include directories (-I)
- Project Properties > Settings > CrossCore GCC ARM Embedded C Linker > Preprocessor > Additional include directories (-I)

5.3 Register Browser Display for GDB with OpenOCD

When using GDB with OpenOCD to debug ARM or SHARC-FX core applications the values of registers may be blank in the Register Browser window. This issue can be worked around by one of the following actions:

- close and reopen the register browser view
- terminate and relaunch the application once the register browser is open
- by adding the register(s) you wish to view to the expressions view as a new expression with $<register name>, for example to view the pc register enter `$pc` as a new expression
- by using the debugger console to print the register(s) you wish to view using command `print $<register name>` , for example `print $pc` .

5.4 Help View on Ubuntu 22.04

Using CCES on Ubuntu 22.04 CCES search help links and the help contents may not open as they should in the default browser on some machines. A possible workaround is to change the default "external browser" selection setting under Window > Preferences >Help to use a specific browser such as Chrome.

5.5 Use of `creal`, `cimag` and other C99 complex math functions with `-pre-fx-compatibility`

The C99 complex functions `creal`, `cimag` and others do not compile when the -pre-fx-compatibility switch is used.