

512Mb Serial Flash memory with QUAD SPI for SC589 Mini Board

(Micron MT25QL512)

The ADSP-SC589 processor has three SPI interfaces: SPI0, SPI1, and SPI2. SPI2 is connected to a Micron MT25QL512 512 Mb serial flash memory with dual and quad SPI support. This flash is used for booting. Quad mode is enabled by default.

Driver Modifications:

The following modifications are made in the driver file of SC589 evaluation board to make it support for the Flash in the SC589 Mini board, also called SAM (SHARC Audio Module).

1. Manufacture ID and the Device ID are modified corresponding to its Flash from Micron MT25QL512, 512-MBit serial flash device data sheet.
Note: Failure to modify this may cause “cannot find DPIO Buffer address” error.
2. Since there is no Software-Controlled Switches (SoftConfig), it is disabled.
3. The earlier Flash driver (Winbond W25Q128FV) had two Read and Write Status Registers. But, the new Flash Driver supports only one Read Status Register and a Write Status Register
Note: Failure to modify this may turn on FAULT LED though flash is successful.

The Driver file should be built after the above code modifications to get the latest DXE file of the desired core.

Creating Bootable Applications:

LDR file has to be created to load the bootable application on to the Flash memory. Here LED blink application is used for testing. After creating a valid application, the following settings are made to generate a LDR file.

To create the loader file for an application:

1. In the **Project Explorer** view, right-click on the Core 0 project and select **Properties**
2. In the **Properties** window, go to the **C/C++ Build→Settings** page and select the **Build Artifact** tab. Under **Artifact Type**, select **Loader File**.
3. On the **Tool Settings** tab, go to the **CrossCore ARM Loader→General** page and input the settings as follows:
 - a. Boot mode: **SPI Master**
 - b. Boot format: **Intel Hex**
 - c. Boot Code: **0x01 / 0xC**

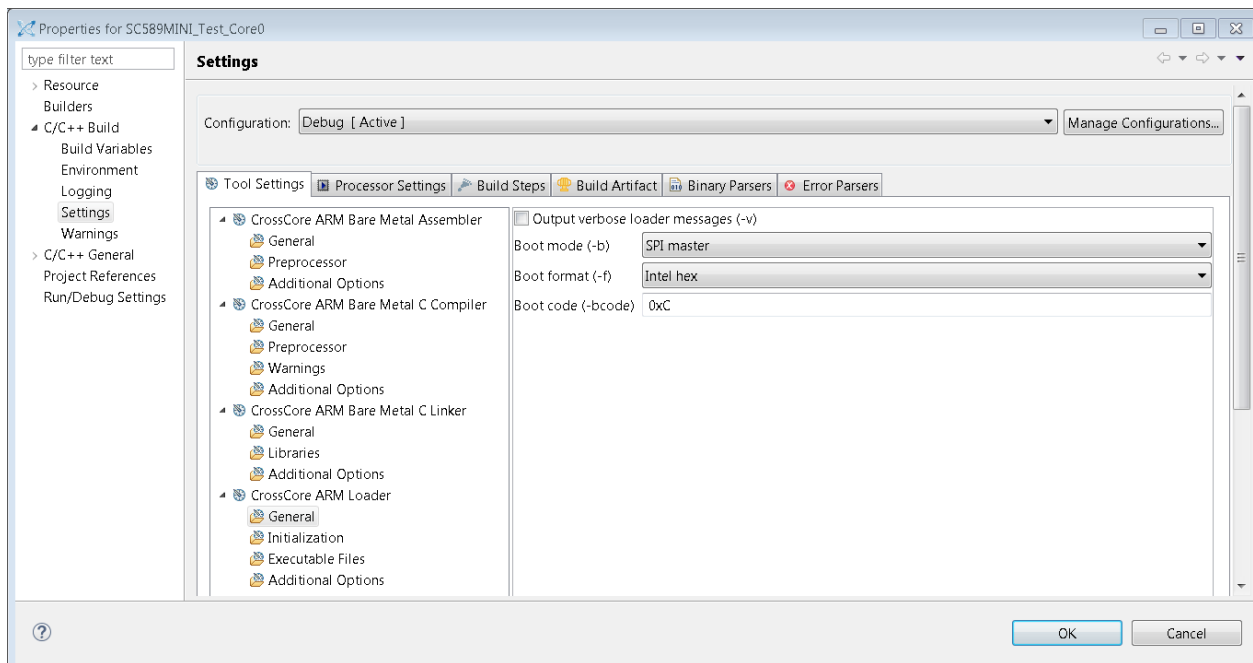


Figure 1. **General** page in Tool Settings in Project Properties

4. Go to the **CrossCore ARM Loader**→**Initialization** page and add the pre-built initialization file by clicking **Browse...** and navigating to
 <CCES Root>\SHARC\ldr\ezkitSC589_initcode_core0_v10

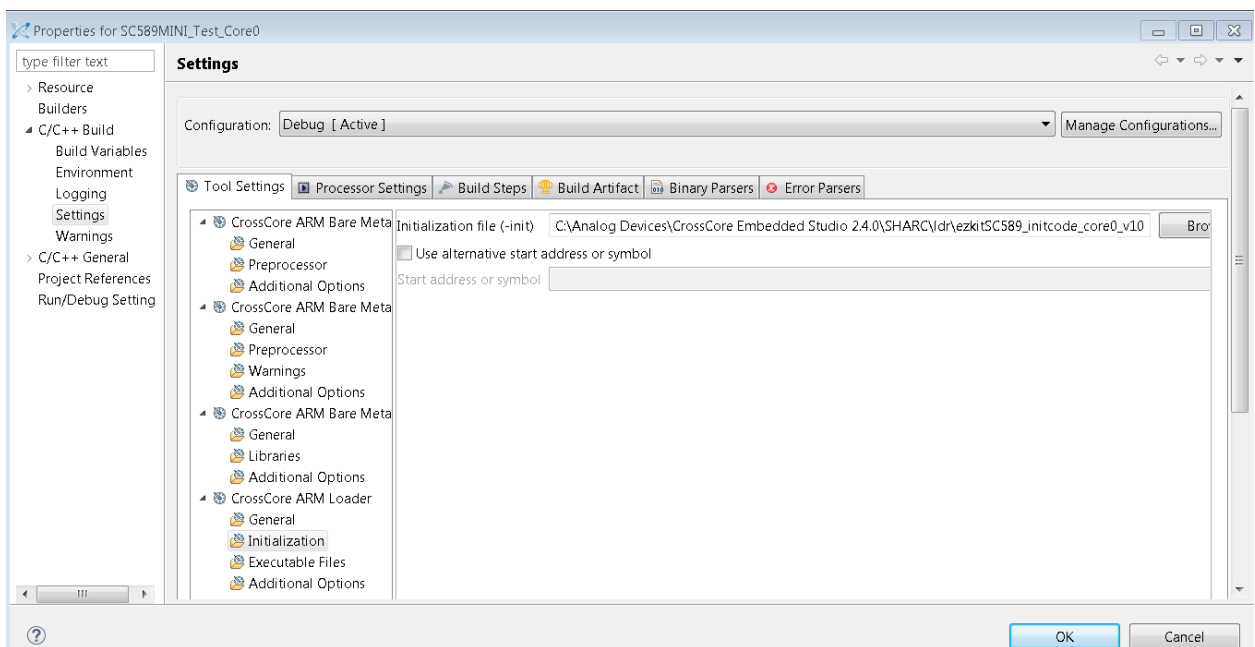


Figure 2. **Initialization** page in Tool Settings in Project Properties

- Since its multi-core processor loader, the following settings are to be configured to properly boot all three cores.

Go to the **CrossCore ARM Loader**→**Executable Files** page. Select **Core 0** as the **Booting Core** and Core1 and Core2 as the other two cores. To set each core's executable file, click **Browse...** and navigate to that core's executable file, which will likely be in the **Debug** folder for that core's project. Note that Core0's executable will not have a DXE suffix, as it is an ARM® core instead of a SHARC® core.

For Core0 and Core1, check the “**Append core input without introducing a FINAL block (-NoFinalTag)**” box. Do not check this box for Core2.

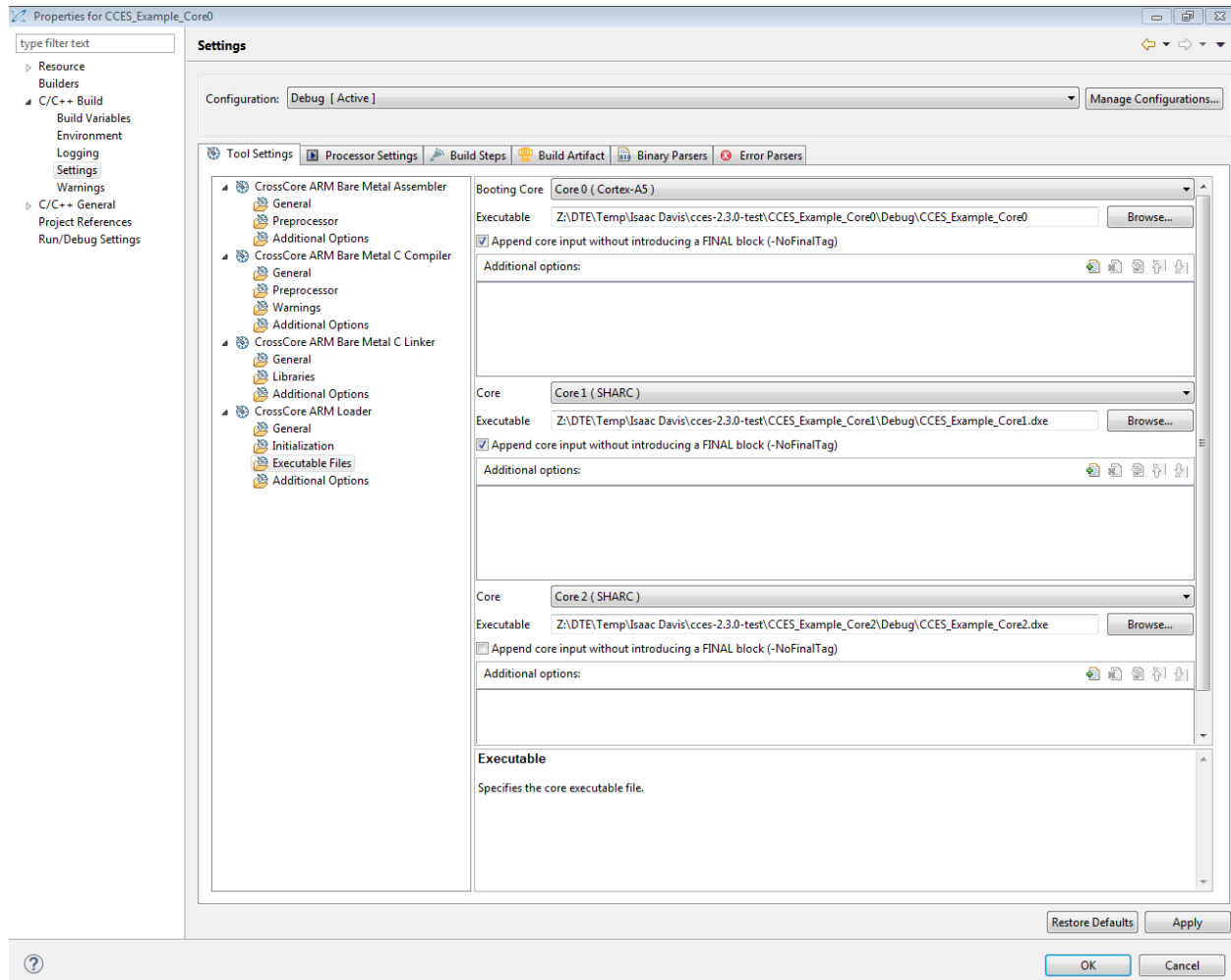


Figure 3. Setting **Executable Files** in the **Tool Settings** tab

- Make sure that the project for Core0 is selected in the **Project Explorer** window. Build the LDR file using the **Project**→**Build Project** pull-down.

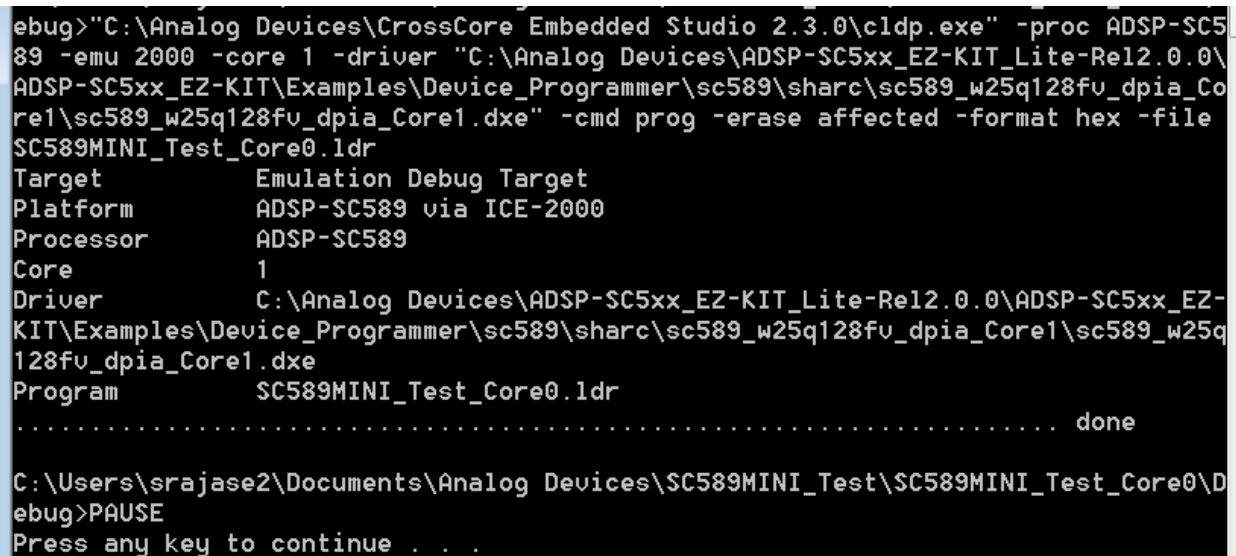
Command Line Device Programmer (CLDP):

CLDP utility helps to load applications into different forms of boot memory. Here the LED blink application is loaded into the Flash memory on the ASDP-589 Mini board. CLDP utility used the LDR file from the application and the DXE file of the Driver.

The following example command loads the application into Flash memory:

```
cldp -proc ADSP-SC589 -emu 2000 -core 1 -driver "<SC589 root directory>\ADSP-SC5xx_EZ-KIT_Lite-Rel2.0.0\ADSP-SC5xx_EZ-KIT\Examples\Device_Programmer\sc589\sharc\sc589_w25q128fv_dpia_Core1\sc589_w25q128fv_dpia_Core1.dxe" -cmd prog -erase affected -format hex -file "<workspace>\SC589MINI_Test_Core0.ldr"
```

The successful **done** message in the command prompt view indicates that the device programming is complete, and the application is now present in flash memory.



```
ebug>"C:\Analog Devices\CrossCore Embedded Studio 2.3.0\cldp.exe" -proc ADSP-SC589 -emu 2000 -core 1 -driver "C:\Analog Devices\ADSP-SC5xx_EZ-KIT_Lite-Rel2.0.0\ADSP-SC5xx_EZ-KIT\Examples\Device_Programmer\sc589\sharc\sc589_w25q128fv_dpia_Core1\sc589_w25q128fv_dpia_Core1.dxe" -cmd prog -erase affected -format hex -file SC589MINI_Test_Core0.ldr
Target          Emulation Debug Target
Platform        ADSP-SC589 via ICE-2000
Processor       ADSP-SC589
Core            1
Driver          C:\Analog Devices\ADSP-SC5xx_EZ-KIT_Lite-Rel2.0.0\ADSP-SC5xx_EZ-KIT\Examples\Device_Programmer\sc589\sharc\sc589_w25q128fv_dpia_Core1\sc589_w25q128fv_dpia_Core1.dxe
Program        SC589MINI_Test_Core0.ldr
. . . . . done

C:\Users\srajase2\Documents\Analog Devices\SC589MINI_Test\SC589MINI_Test_Core0\Debug>PAUSE
Press any key to continue . . .
```

Figure 4. Executing CLDP

Note: Make sure that the boot mode of SC589-Mini board is set using the Jumper JP1 in such a way that it is connected to pin 1-2 and reset the device to get the output of an application.