Using UART with a SmartFusion cSoC

Libero SoC and IAR Embedded Workbench Flow Tutorial

Microsemi

Table of Contents

Introduction

This tutorial describes the use of Microsemi tools to configure the SmartFusion® customizable system-on-chip (cSoC) microcontroller subsystem (MSS)-based hardware design on SmartFusion cSoC devices and how to use Embedded Workbench® for ARM (EWARM) from IAR® to develop applications to run on a SmartFusion cSoC.

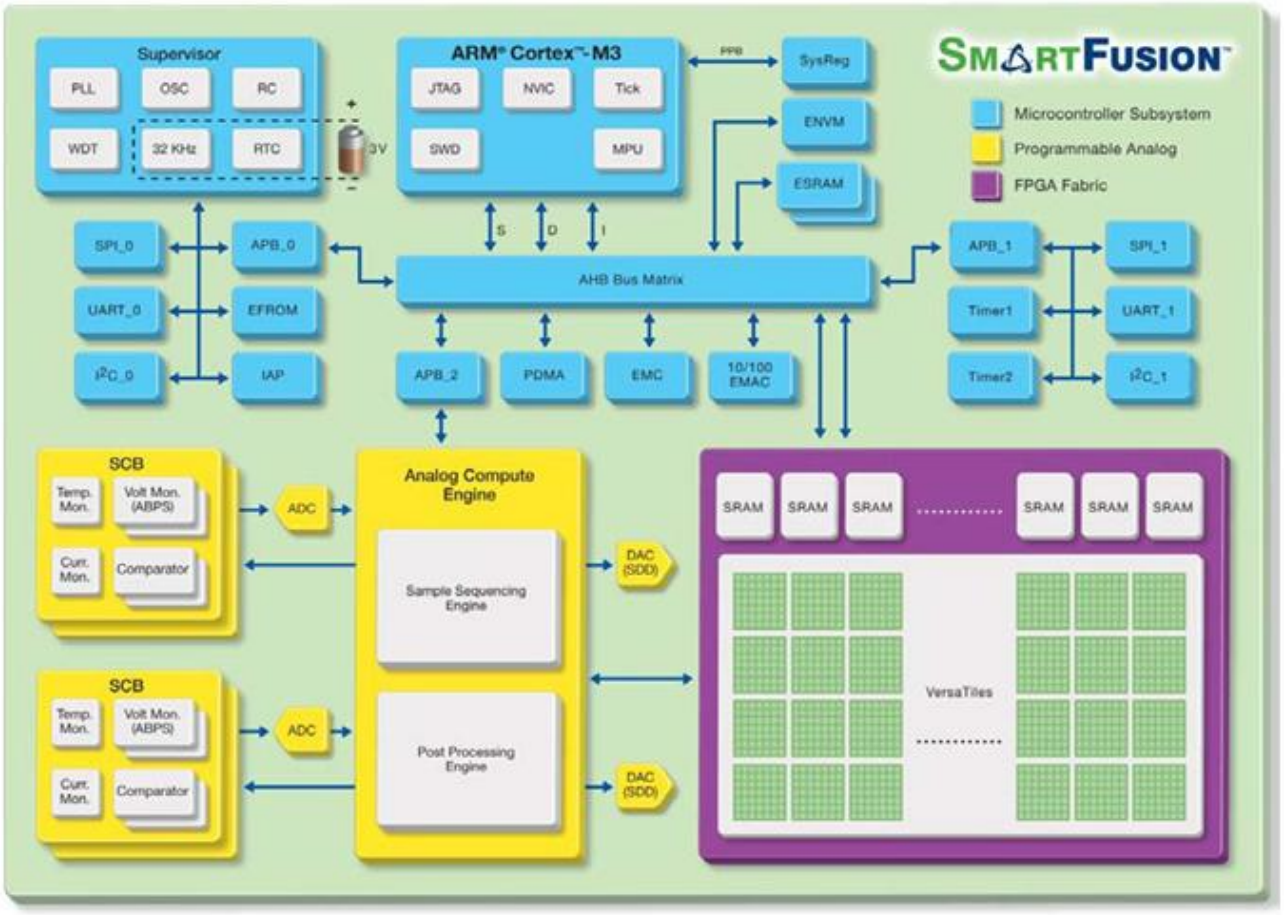

Figure 1 · SmartFusion cSoC Block Diagram

After completing this tutorial, you will be familiar with the following:

- Creating and setting up a Libero® system-on-chip (SoC) project for a SmartFusion cSoC device
- Using SmartDesign MSS configurator to configure the peripherals and generate an embedded memory configuration file (EFC)
- Generating the programming file to program the SmartFusion cSoC device
- Opening the project in EWARM IDE from Libero SoC and writing application code
- Compiling application code
- Creating and launching a debug session

Tutorial Requirements

Software Requirements

This tutorial requires the following software installed on your PC.

- EWARM v6.30.7 or later
- Libero SoC v10.0 or later, which can be downloaded from www.microsemi.com/soc/download/software/libero/default.aspx

Hardware Requirements

This tutorial requires the following hardware:

- Two USB cables (programming and communication) - one for connecting the programmer to your PC and the other to connect the universal asynchronous receiver/transmitter (UART) interface on the board to the PC.
- The Microsemi kit contains a low cost programming stick (LCPS) or FlashPro4 programmer that can be used to program the SmartFusion cSoC device with the PDB. However, to debug using EWARM, IAR supplied J-Link debugger hardware (not supplied with the SmartFusion Evaluation Kit Board or SmartFusion Development Kit Board) is required.

Associated Project Files for IAR Workbench

You can download the associated solution project files for this tutorial from Microsemi website:

www.microsemi.com/soc/download/rsc/?f=SmartFusion_UART_HW_flow_LiberoIAR_tutorial_DF.

The IAR_Debugger_files folder contains the **SmartFusion_dss.ddf** file. DDF files enable you to see the MSS register map for all peripherals.

Note: Extract the Design Files to root directory.

You can download the programming file (*.stp) for this tutorial from the Microsemi website:
www.microsemi.com/soc/download/rsc/?f=SmartFusion_UART_HW_flow_LiberoIAR_tutorial_PF.

MSS Components Used

- ARM® Cortex™-M3 processor
- Communications matrix
- Clock conditioning circuit (CCC)
- UART0

Target Board

SmartFusion Evaluation Kit Board (A2F-EVAL-KIT) or SmartFusion Development Kit Board (A2F-DEV-KIT).

Objective

The objective of this tutorial is to instruct how to configure the SmartFusion UART hard (on-chip) peripherals and access them through the ARM Cortex-M3 processor.

Design Steps

Following are the major steps to be executed for this tutorial:

1. Create a Libero SoC project for SmartFusion cSoC.
2. Configure the SmartFusion cSoC microcontroller subsystem (MSS) peripherals.
3. Generate the MSS component.
4. Perform synthesis and layout and generate a programming file to program the SmartFusion cSoC device.

5. Program the SmartFusion A2F200M3F or A2F500M3F cSoC device.
6. Open the software project in EWARM and write the application code.
7. Run the application to communicate with HyperTerminal through UART0 on the SmartFusion Evaluation Kit Board or Development Kit Board.

Working with Libero SoC and IAR

This section describes how to create a new Libero SoC project, configure the microcontroller subsystem (MSS), program the design on the SmartFusion cSoC board, and run an application program in the IAR Workbench.

Step 1 – Creating a Libero SoC Project

1. Launch Libero SoC v10.0 or later.
2. From the **Project** menu, select **New Project**. Enter the information shown below in the Libero **New Project** dialog box.
   - Name: SmartFusion_UART
   - Location: <..> (For example, *C:\Microsemiprj\UART_Libero_IAR*)
   - Family: SmartFusion
   - Die: If you are using SmartFusion Evaluation Kit Board, enter A2F200M3F; if you are using SmartFusion Development Kit Board, enter A2F500M3F.
   - Package: 484 FBGA
   - Speed: STD

   Leave others as default.

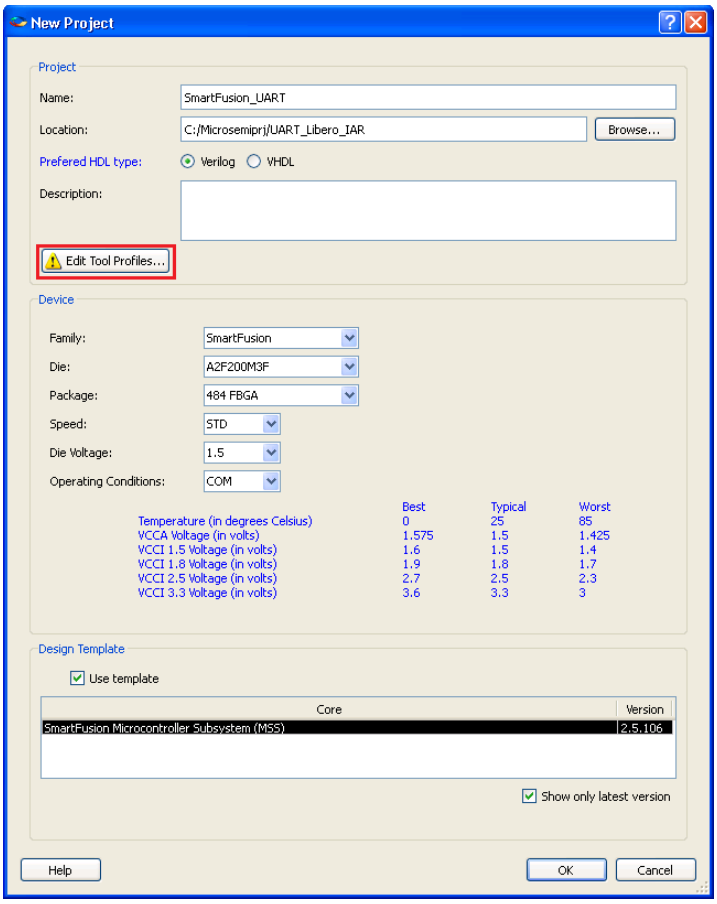

Figure 2 · New Project Dialog Box

3. Click **Edit Tool Profiles** and add IAR by clicking **Software IDE**, as shown in Figure 3 · .

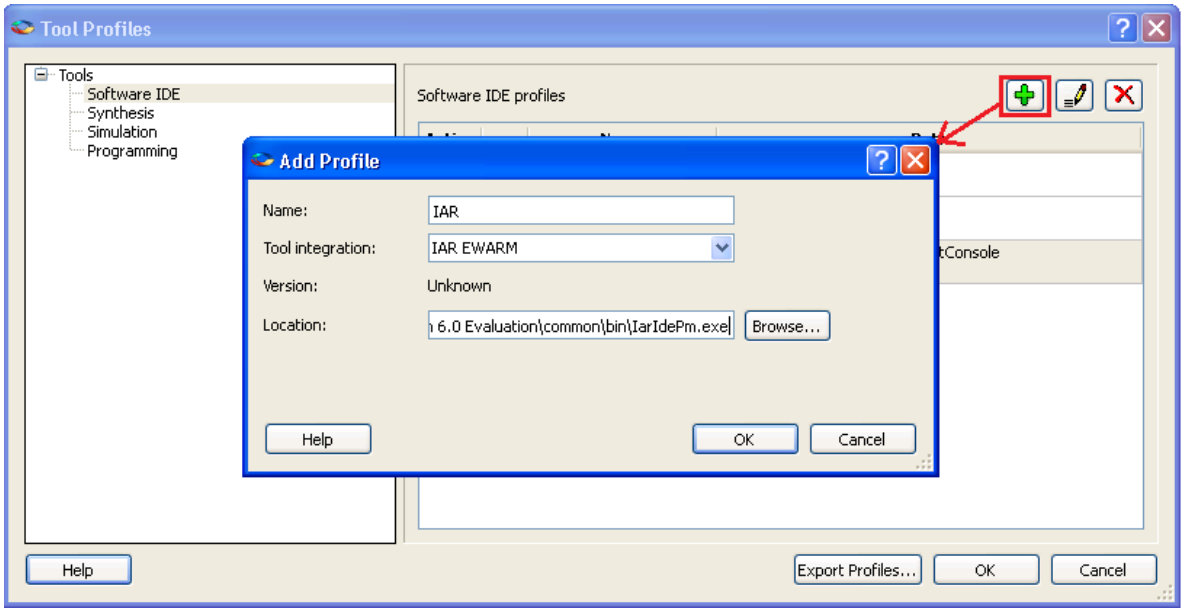

Figure 3 · Selecting IAR as Software IDE

4. After adding the Profile, click **OK** to close the Add Profile dialog window.

   Repeat the steps (3 and 4) above for Synthesis, Simulation, and Programming and then click **OK** to close the Tool Profiles dialog window.

5. Select the MSS core in the **New Project** dialog box and then click **OK**.

   Note: If SmartFusion cSoC MSS is not displayed in the list, refer to Appendix A – Libero SoC Catalog Settings to find out how to set the repositories. If the vault does not have the MSS core, download the core by double-clicking the core name in the **Design** template in the **New Project** dialog box.

6. The project is created and the Libero SoC window appears, as shown in Figure 4 · . The SmartDesign "SmartFusion_UART" is created with the instantiation of the MSS component.

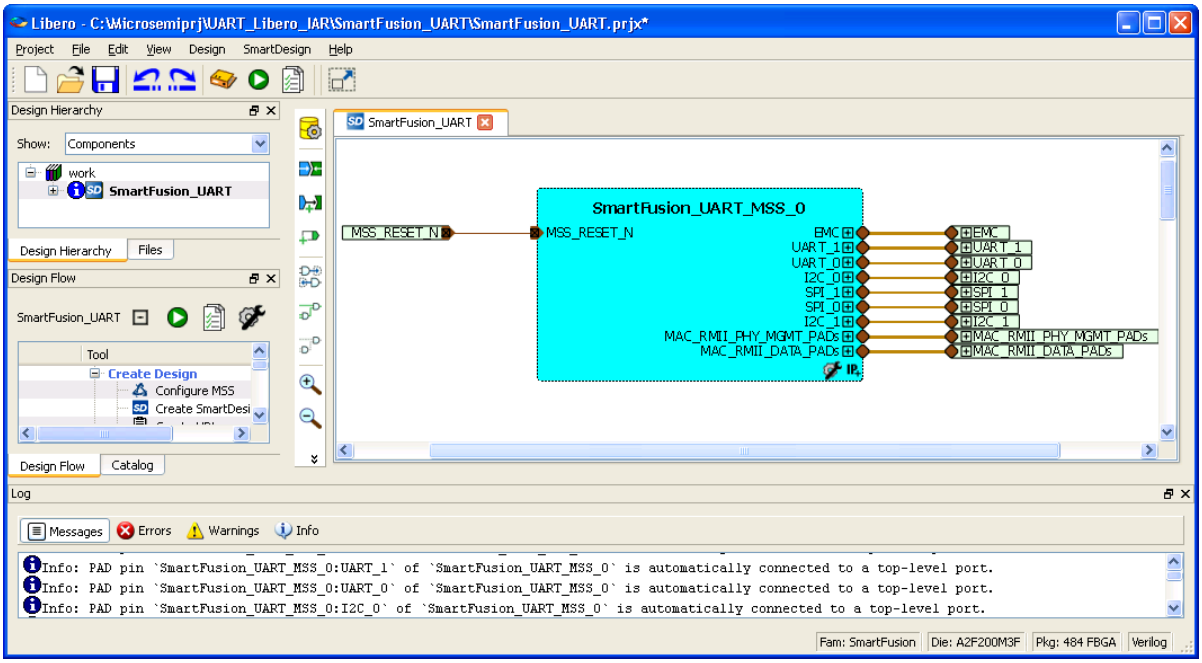

Figure 4 · The Libero SoC Window After Completing New Project Wizard

Step 2 – Configuring MSS Peripherals

1. Double-click the **SmartFusion_UART_MSS_0** component to configure the MSS. The MSS is displayed in the SmartDesign Canvas in a new tab, as shown in Figure 5 · .

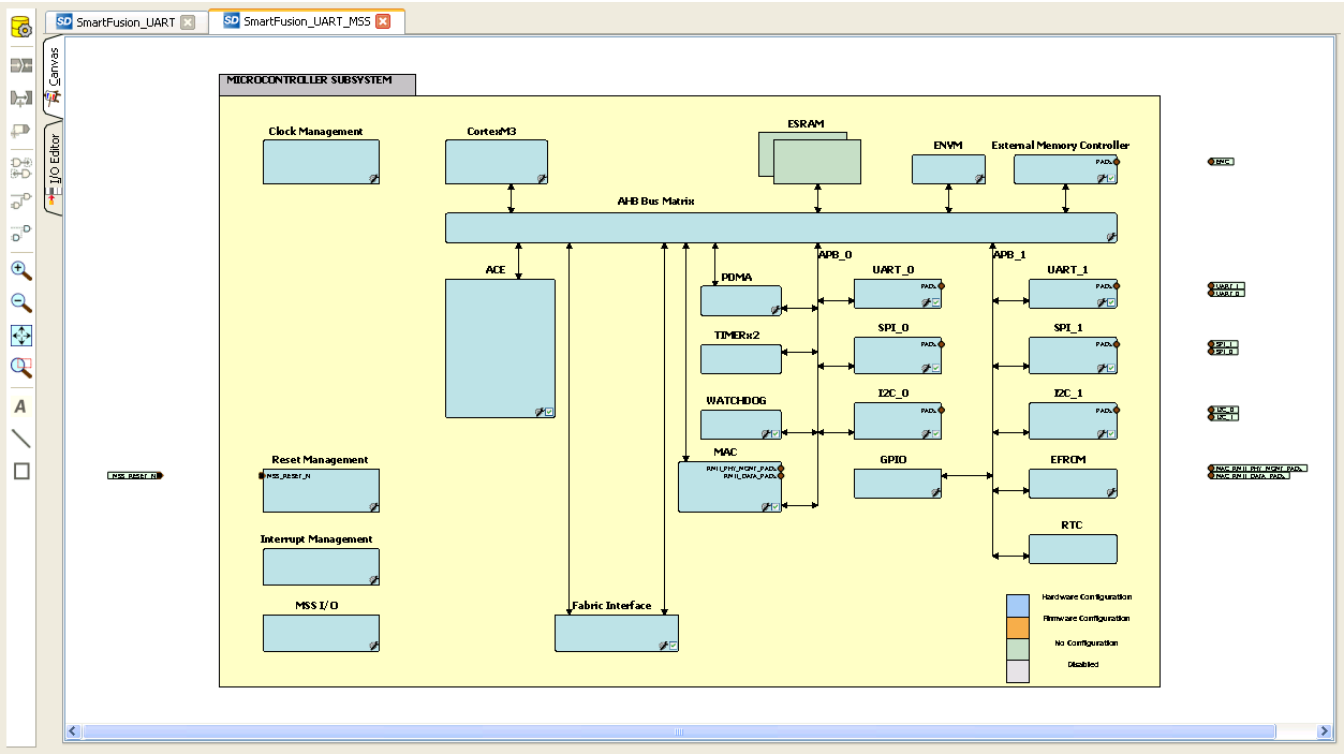

Figure 5 · MSS in the SmartDesign Canvas

The enabled MSS peripherals are highlighted in blue, and can be configured in the hardware. The disabled MSS peripherals are shown in gray.

To disable a peripheral that is not required, select the peripheral, right-click and clear the **Enabled** check box; or clear the check box in the lower right corner of the **peripheral** box. The box turns grey to indicate a peripheral has been disabled. Disabled peripherals can be enabled by repeating the procedure.

An enabled peripheral is shown in Figure 6 · .

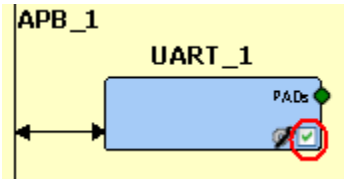

Figure 6 · Enabling a Peripheral

2. The following peripherals must be disabled because they are not used in this tutorial: analog compute engine (ACE), MAC, fabric interface, WATCHDOG, SPI0, SPI1, I2C0, I2C1, UART1, and EMC.

3. After disabling the peripherals, the SmartDesign canvas looks similar to Figure 7 · .

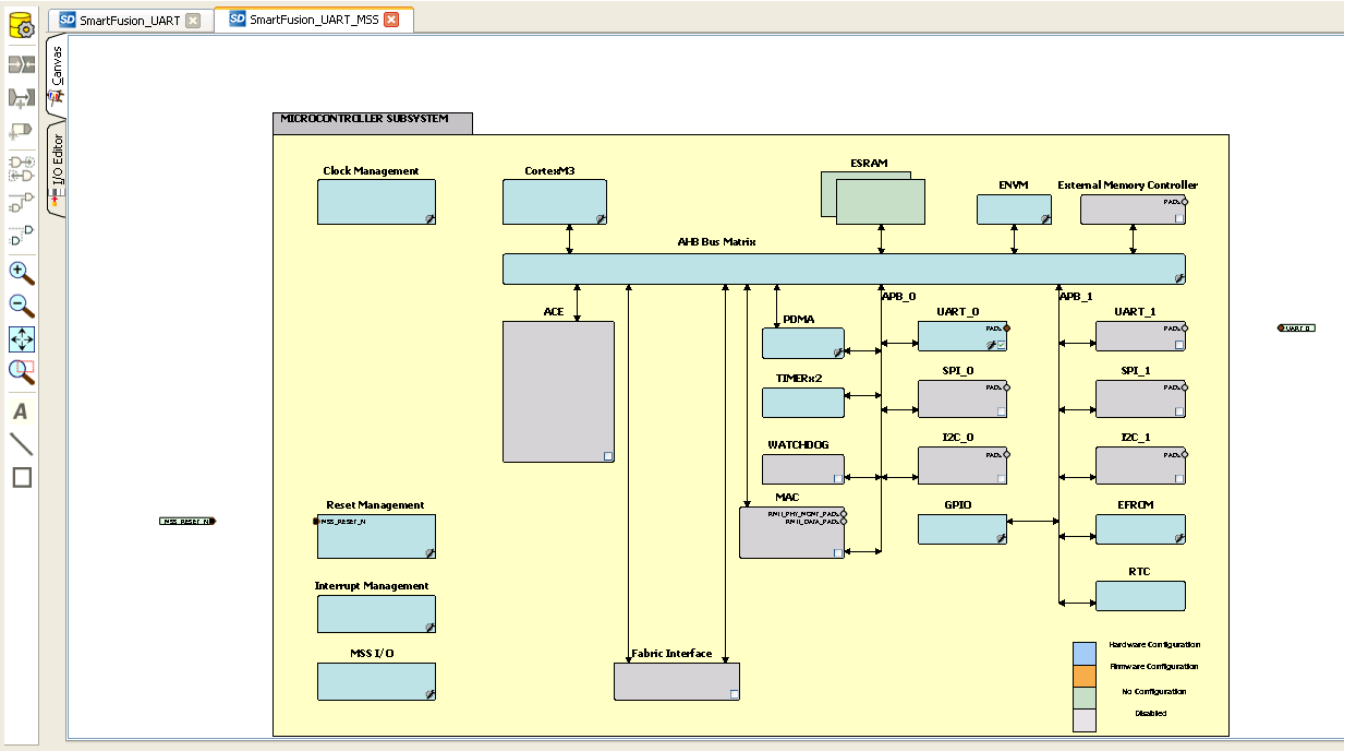

Figure 7 · Used MSS Peripherals

4. Double-click the **Clock Management** block and configure as shown below:
   - CLKA: On-Chip RC Oscillator
   - MSS clock source: PLL output
   - MSS clock frequency: 80 MHz

   Use default settings for all the other fields.

   After completing the configuration, click **OK**.

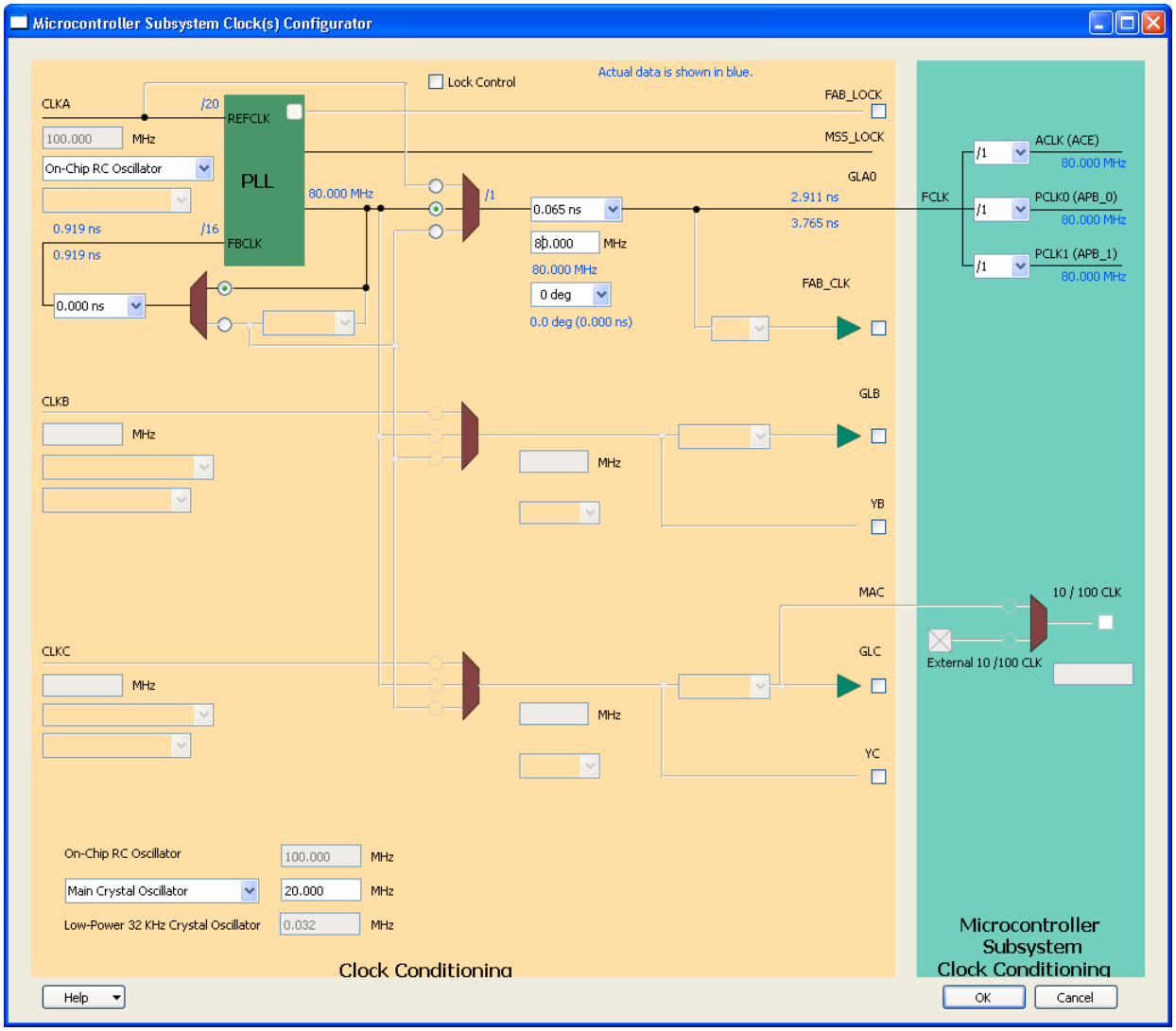

Figure 8 · MSS Clock Configuration

5. Click **File** > **Save** to save the SmartFusion_UART_MSS.

Step 3 – Generating the MSS Component

1. Right-click the **SmartFusion_UART_MSS_0** component on the **SmartFusion_UART_MSS** tab and select **Update Instance(s) with Latest Component…**, as shown in Figure 9 · .

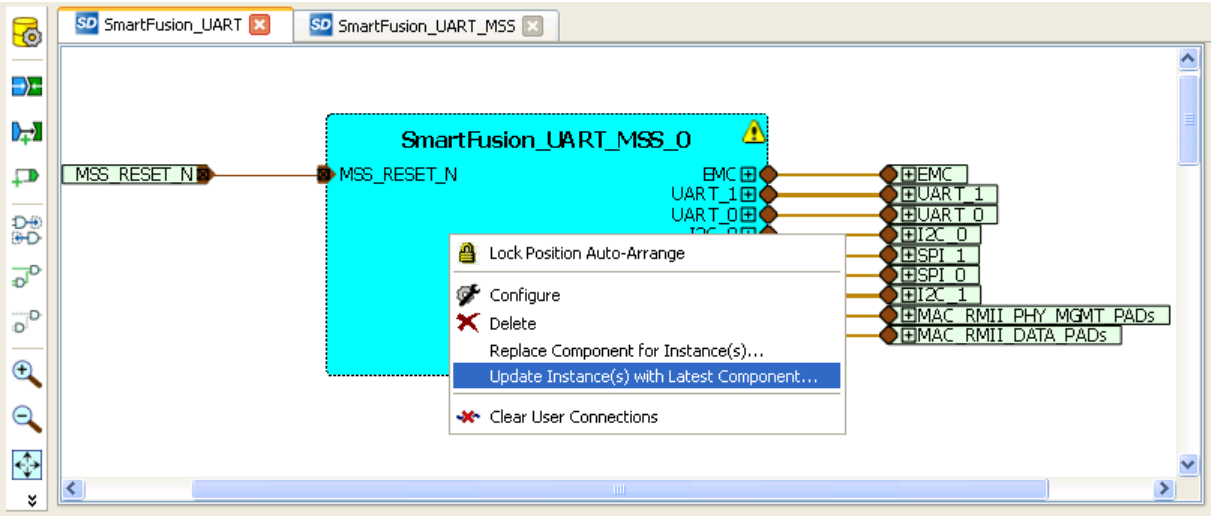

Figure 9 · Updating the MSS

2. Click **Design > Configure Firmware**, as shown in Figure 10 · .

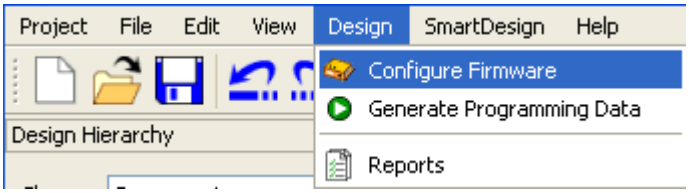

Figure 10 · Opening Design_Firmware

3. On the **DESIGN_FIRMWARE** tab, clear the **Generate** check boxes for all the peripherals for which you do not need to generate the firmware. Click **Configuration** on the SmartFusion_CMSIS_PAL_0 instance and select **IAR Embedded Workbench** as the configuration.

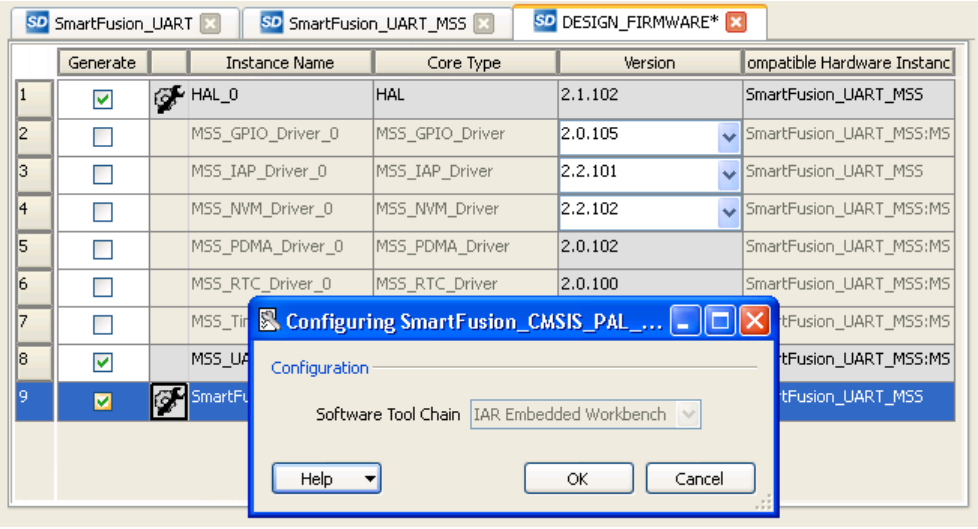

Figure 11 · Configuring SmartFusion_CMSIS_PAL_0

Note: Check whether or not you are able to see the latest version of the drivers without any warning or error indicating that firmware is missing from the Vault. If missing, refer to Appendix B – Firmware Catalog Settings.

4. Click **File > Save** to save the **DESIGN_FIRMWARE**.
5. **Save** the design and generate the component by clicking **Generate Component** or click **SmartDesign > Generate Component**.

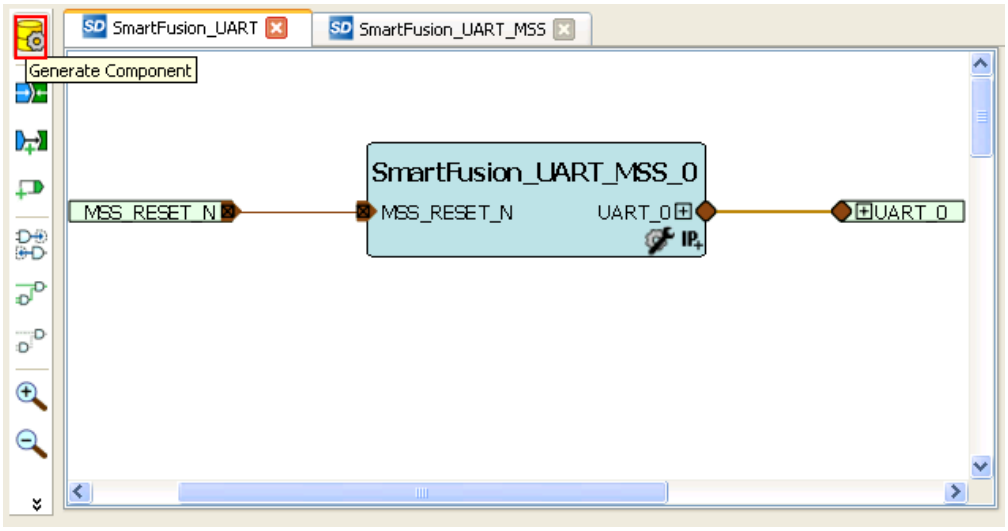

Figure 12 · Generating the Component

6. After successful generation of the MSS component, the log window displays the message "**Info: 'SmartFusion_UART' was successfully generated. Open datasheet for details**". The datasheet contains the project information such as generated files, used I/Os, and memory map.
7. Ensure that the IAR folder is created with the folders and files, as shown in Figure 13 · .

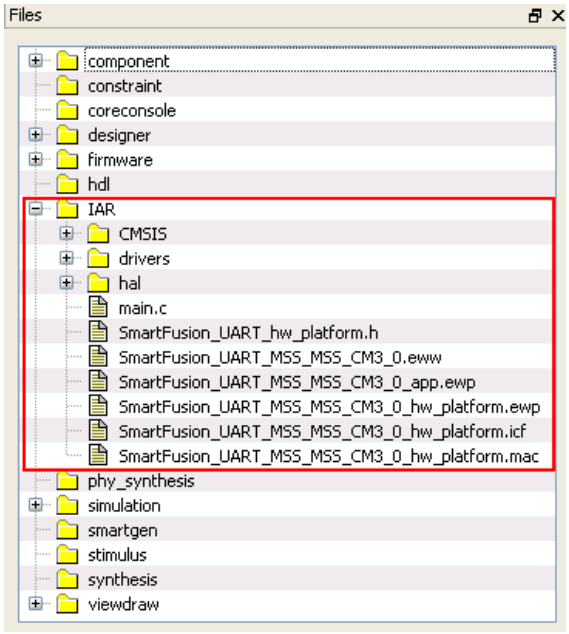

Figure 13 · Files Window After Generating the Project

Step 4 – Generating the Program File

Libero SoC provides the push-button flow for generating the programming data of a project in a single step. By clicking **Generate Programming Data**, you can complete the synthesis, place-and-route, verify the timing, and generate the programming file. You can also complete the flow by running the synthesis and place-and-route tools in interactive mode step-by-step. For more information, refer to the *Libero SoC Quick Start Guide*.

Push-button Design Flow

Click **Generate Programming Data**, as shown in Figure 14 · to complete the synthesis, place-and-route, verify timing, and generate the programming file. This completes the *.fdb file generation.

Figure 14 · Generating Programming Data

The **Design Flow** window is displayed as shown in Figure 15 · .

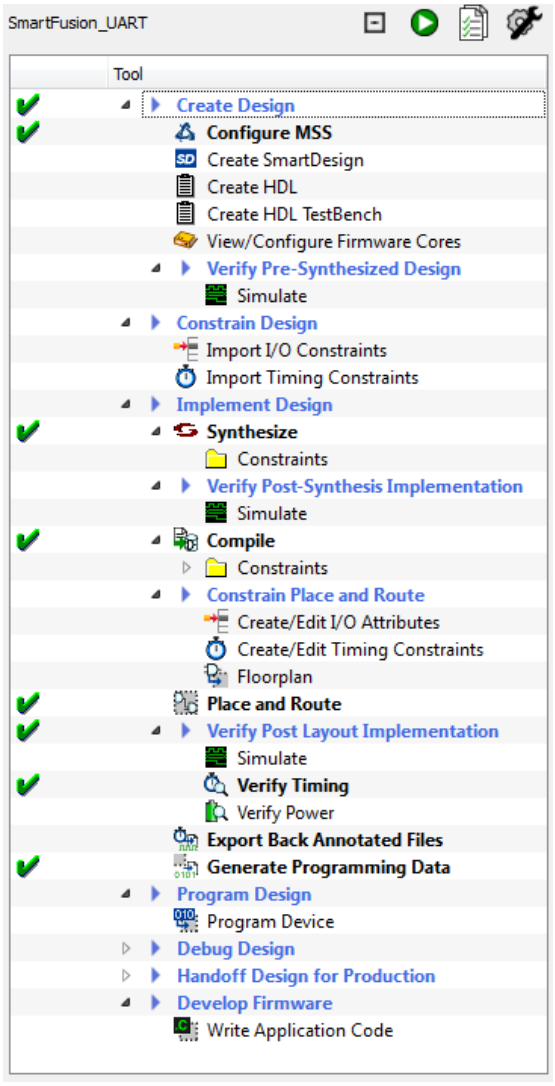

Figure 15 · Design Flow Window After Building the Project

Step 5 – Programming SmartFusion Board Using FlashPro

Before you proceed with programming the device, ensure that the low cost programming stick (LCPS) or FlashPro4 is properly connected to the board. Use the following details to ensure the correct jumper settings. Refer to the *SmartFusion Evaluation Kit User's Guide* and the *SmartFusion Development Kit User's Guide* for additional information.

Jumper Settings for SmartFusion Evaluation Kit Board

- JP10: Connect pin 1 and 2.
- JP7: Connect pin 1 and 2 for LCPS programming mode.
- J6: Connect pins 1 and 2 with the jumper.
- JP6: Connect pins 2 and 3 with the jumper.
- J13: Connect the USB cable to the J13 connector. When the cable is connected, the FlashPro4 or FlashPro drivers may be installed if they are not already installed.
- J14: Connect one end of USB mini B cable to J14.

Jumper Settings for SmartFusion Development Kit Board

SW9 must be off (JTAGSEL = H) in order to program the SmartFusion device. SW9 remains in the off position for Libero SoC and SoftConsole programming. Make the jumper settings as shown in table:

Table 1 · Jumper Settings for Development Kit Board

Factory Default	Factory Default	Factory Default
JP1: 1–2	JP12: 1–2	JP21: 1–2
JP2: 1–2	JP13: 1–2	JP22: 2–3
JP4: 1–3; 7–9	JP14: 1–2	JP23: 1–2
JP5: 1–2; 3–4	JP15: 1–2	JP24: 1–2
JP6: 2–3	JP16: 2–3	JP27: 1–2
J7: 2–3; 6–7; 10–11; 14–15	JP17: 2–3	JP28: 1-2
JP7: 1–2	JP18: 1–2	J32: 1–2; 3–4; 5–6
JP8: 3-4; 7-8; 11-12; 15-16	JP19: 2–3	–
JP11: 1-2	JP20: 1–2	–

Programming the Device

1. Double-click **Program Device** under **Program Design** in the **Design Flow** window to program the SmartFusion cSoC device as shown in Figure 16 · on page 16.
2. Click **Yes** when it prompts that the I/O and timing constraints are not yet set.

   Note: Do not interrupt the programming sequence; it may damage the device or the programmer. If you face any problems, contact Microsemi SoC Products Group Technical Support at soc_tech@microsemi.com.

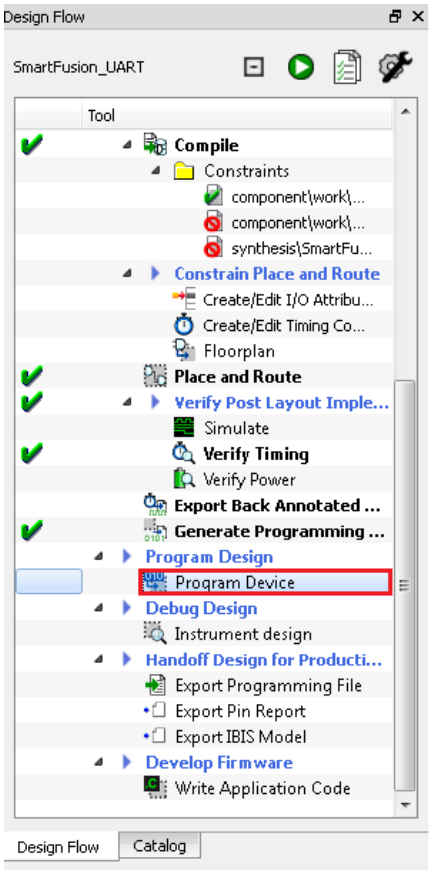

Figure 16 · Design Flow Window

You can use FlashPro interactively by right-clicking **Program Device** in **Design Flow** window and selecting **Open Interactively**. For more information on FlashPro refer to the *FlashPro User's Guide*.

Step 6 – Building the Software Application Through IAR Embedded Workbench

1. From **Libero SoC**, open the IAR project by double- clicking **Write Application Code** under **Develop Firmware** in the **Design Flow** window.

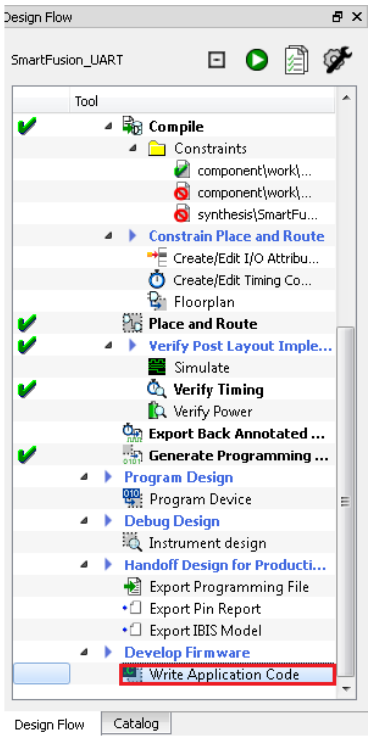

Figure 17 · Invoking IAR from Libero SoC

The IAR workbench perspective looks similar to Figure 18 · .

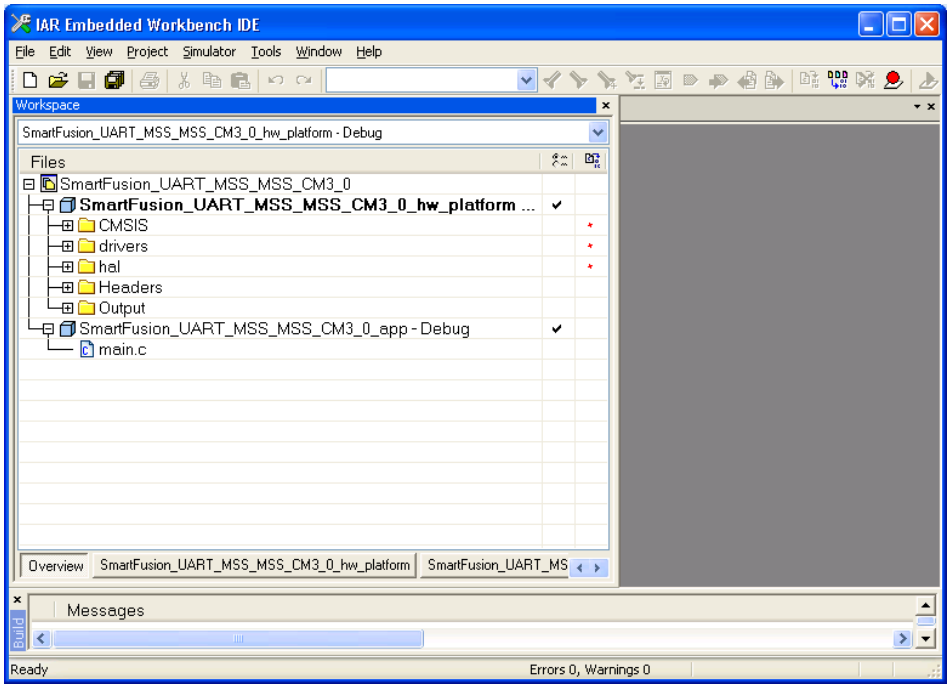

Figure 18 · IAR Workbench

2. Copy the code provided below and paste it in the **main.c** file under the SmartFusion_UART_MSS_MSS_CM3_0_app project in the IAR editor, and then delete the existing code.

```
#include "mss_uart.h"

#define  Microsemi_logo \
"\n\r \
**     **  *******   ******  *****    ****    *****  ******  **     **  ******* \n\r \
* *   * *     *     *        *    *  *    *  *       *       * *   * *     *    \n\r \
*  * *  *     *     *        *****   *    *   ****   ******  *  * *  *     *    \n\r \
*   *   *     *     *        *    *  *    *       *  *       *   *   *     *    \n\r \
*       *  *******   ******  *    *   ****   *****   ******  *       *  ******* "

/* Main function */
void main()
{
  const uint8_t greeting[] = "\n\rWelcome to SmartFusion "
                      "- customizable system-on-chip (cSoC) \n\r";
  size_t rx_size;
  uint8_t rx_buff[1];

  /* Initialize and configure UART0. */
  MSS_UART_init
    (
     &g_mss_uart0,
     MSS_UART_57600_BAUD,
     MSS_UART_DATA_8_BITS | MSS_UART_NO_PARITY | MSS_UART_ONE_STOP_BIT
     );

  /* Send the Microsemi Logo over the UART_0 */
  MSS_UART_polled_tx_string( &g_mss_uart0, (const uint8_t *)Microsemi_logo);

  /* Send greeting message over the UART_0 */
  MSS_UART_polled_tx( &g_mss_uart0, greeting, sizeof(greeting) );

  /* Echo back any characters received. */
  while(1)
  {
    rx_size = MSS_UART_get_rx ( &g_mss_uart0, rx_buff, sizeof(rx_buff) );
    if( rx_size > 0 )
    {
      MSS_UART_polled_tx( &g_mss_uart0, rx_buff, sizeof(rx_buff) );
    }
  }
}
```

3. To configure the options for the project, right-click the project name (*SmartFusion_UART_MSS_MSS_CM30_app*) and click **Options**.

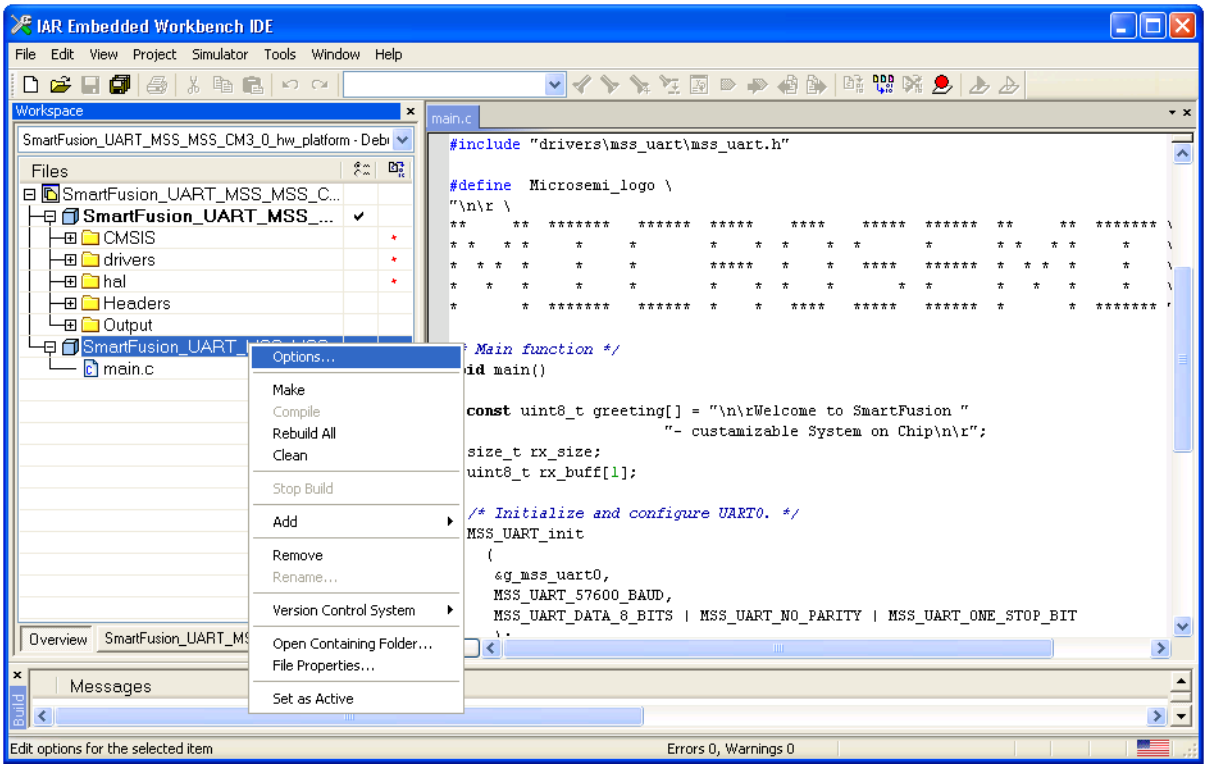

Figure 19 · Selecting Options for the Project

The available options for SmartFusion_UART_MSS_MSS_CM30_app node are displayed in Figure 20 . .

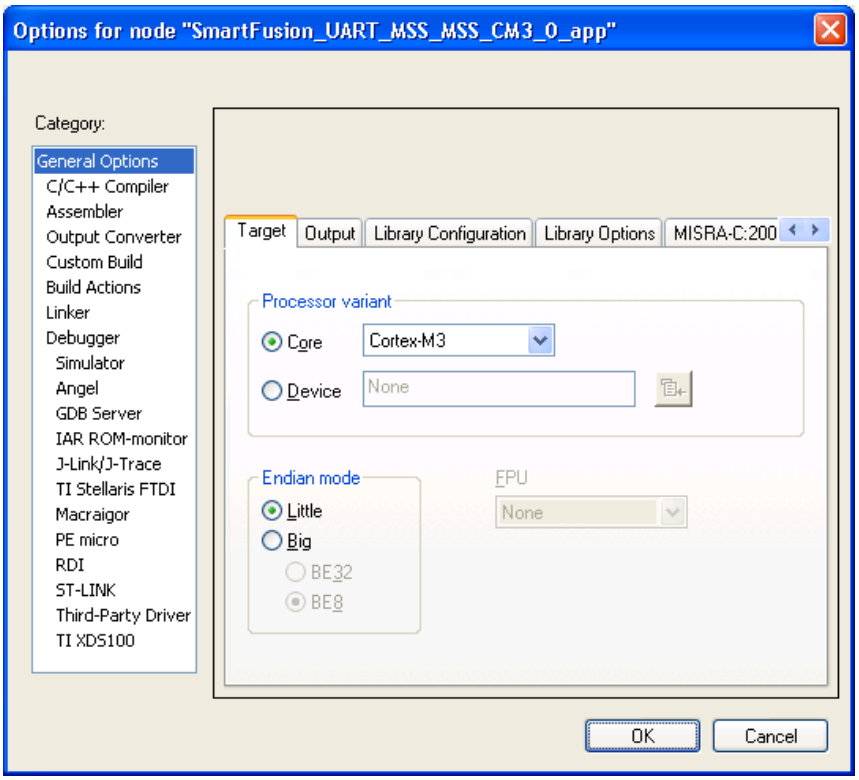

Figure 20 · Accessing the Options for Your Project

4. Click **Linker** under **Category**, as shown in Figure 21 · . The Linker configure file (*SmartFusion_UART_MSS_MSS_CM3_0_hw_platform.icf*). Add the file, which is available at C:\Microsemiprj\UART_Libero_IAR\SmartFusion_UART\IAR folder by selecting the **Override default** check box. The *.icf file is a linker configuration file that defines the segmentation of memory.

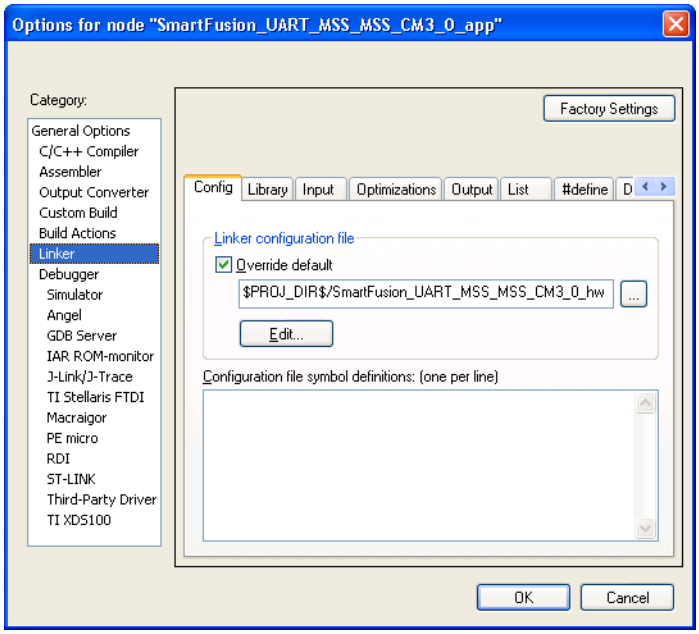

Figure 21 · Adding the Linker Configuration File

5. Click **Edit** to display the **Linker configuration file editor**, as shown in Figure 22 · . This window displays the Vector Table, Memory Regions, and Stack/Heap Sizes tabs.
6. Click the **Memory Regions** tab to view the RAM/ROM regions.

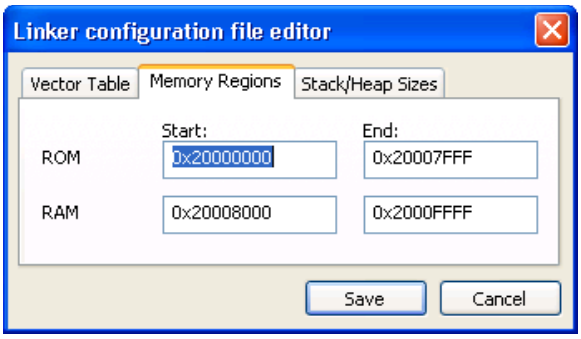

Figure 22 · The Linker Configuration File Editor – Memory Regions Tab

7. Click **Stack/Heap Sizes** tab to view the Stack/Heap sizes. Click **Save.**

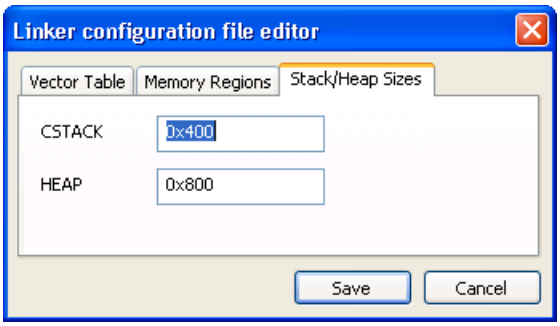

Figure 23 · The Linker Configuration File Editor – Stack/Heap Sizes Tab

8. Click **Debugger** under **Category**. On the **Setup** tab, select the driver **J-Link/J-Trace** from the drop-down list.

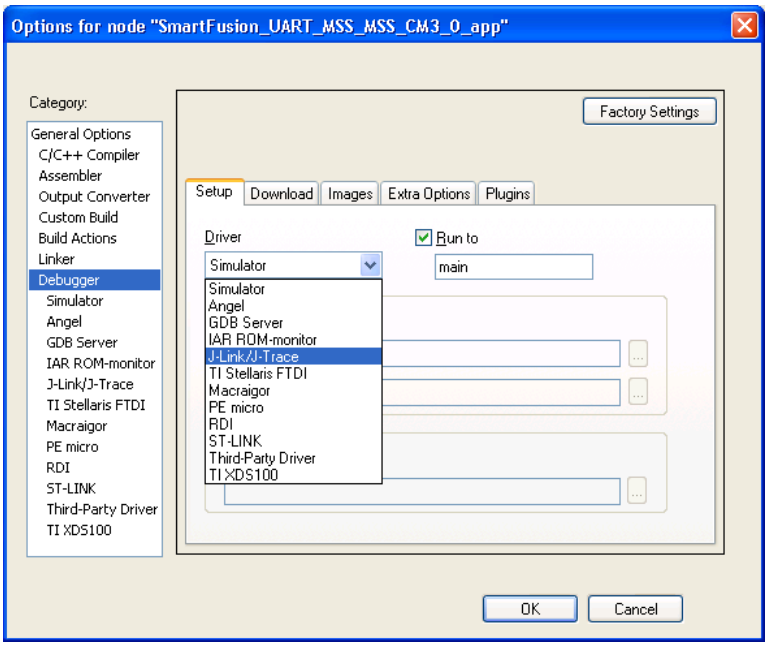

Figure 24 · Selecting the J-Link/J-Trace Driver for the Debugger

9. Select the **Use macro file(s)** option listed under **Setup macros**. Browse to the *SmartFusion_UART_MSS_MSS_CM3_0_hw_platform.mac* file, which is available in the C:\Microsemiprj\UART_Libero_IAR\SmartFusion_UART\IAR folder. This file is used to remap the ESRAM to the 0th location.

10. Select the **Override default** check box under **Device description file** and browse to the *SmartFusion_dss.ddf* file, available in C:\Microsemiprj\UART_Libero_IAR\SmartFusion_UART\IAR. The DDF file enables you to see the MSS Register map for all the peripherals.

    Note: The debugger files provided in the attached zip folder are extracted to the project directory where you will store the EWARM project (for example, C:\Microsemiprj\UART_Libero_IAR\SmartFusion_UART\IAR).

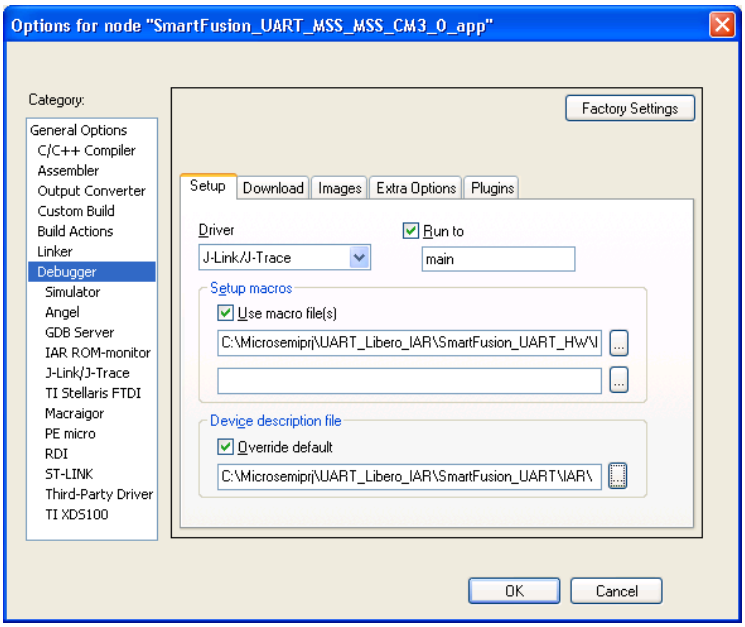

Figure 25 · Setting Up Macros for the Debugger

11. Select the **Verify download** check box on the **Download** tab.

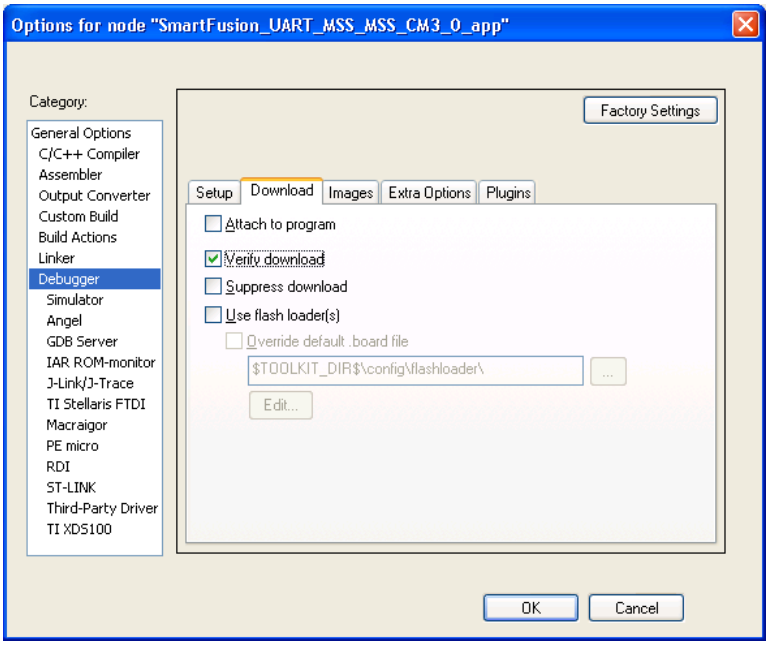

Figure 26 · Download Verification Settings for the Debugger

12. Click **J-Link/J-Trace** under **Category**. On the **Setup** tab, select **Core** from the **Reset** drop-down list, as shown in Figure 27 · .

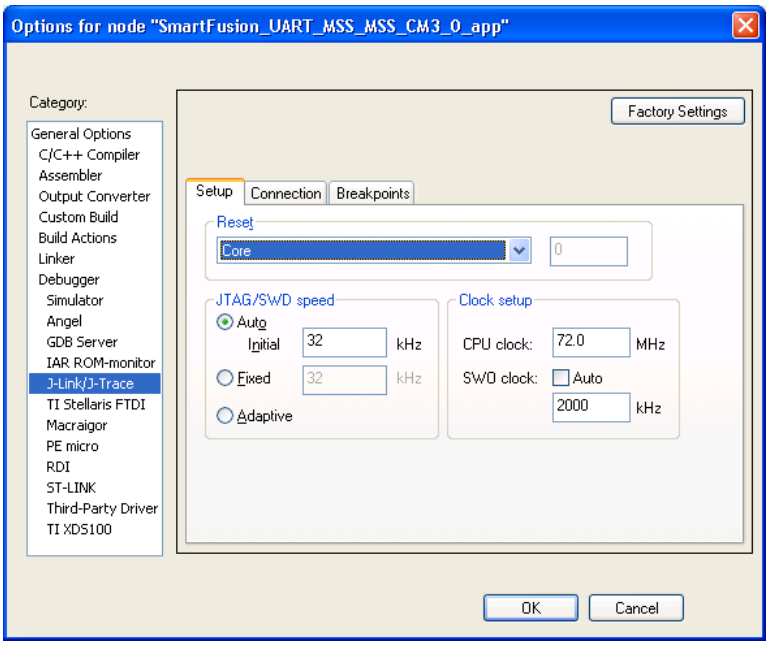

Figure 27 · J-Link/J-Trace Setup

13. Click **OK** to close this window. Now you can build the project.

14. Right-click **SmartFusion_UART_MSS_MSS_CM3_0_hw_platform** and select **Make**.

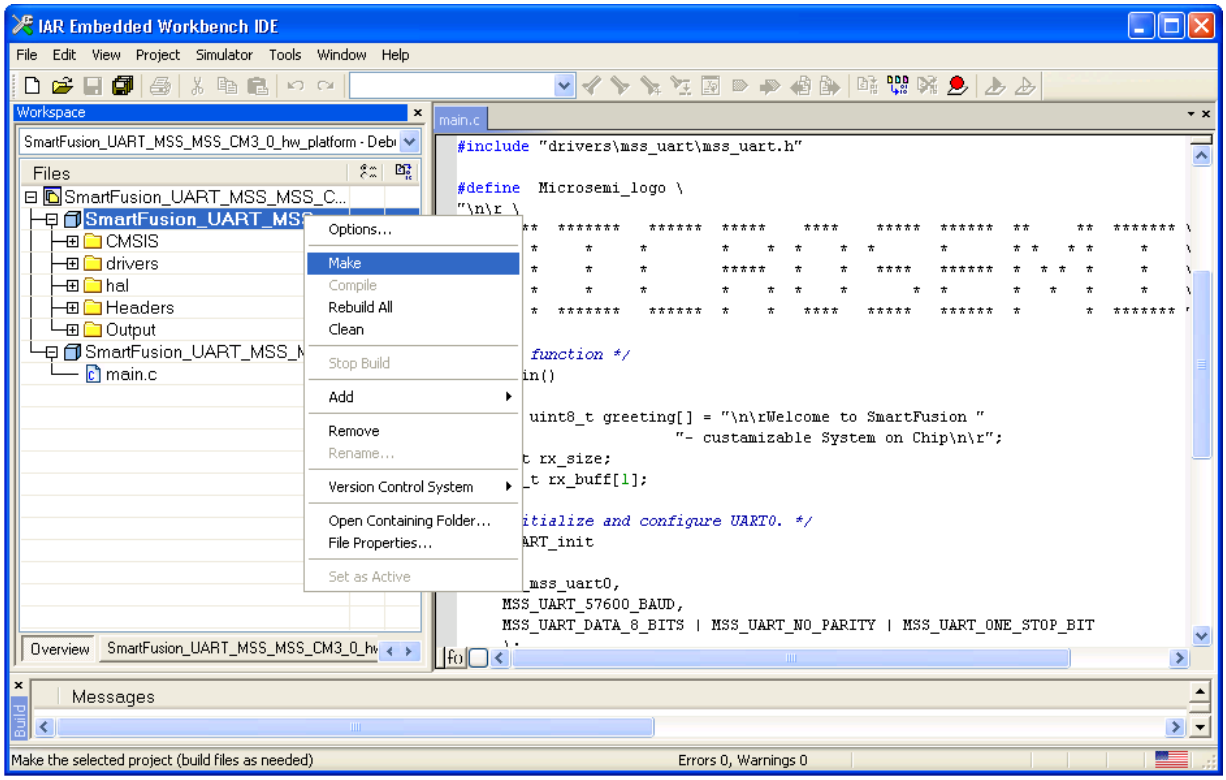

Figure 28 · Make the Hardware Platform

15. Right-click **SmartFusion_UART_MSS_MSS_CM3_0_app** and select **Set as Active**.

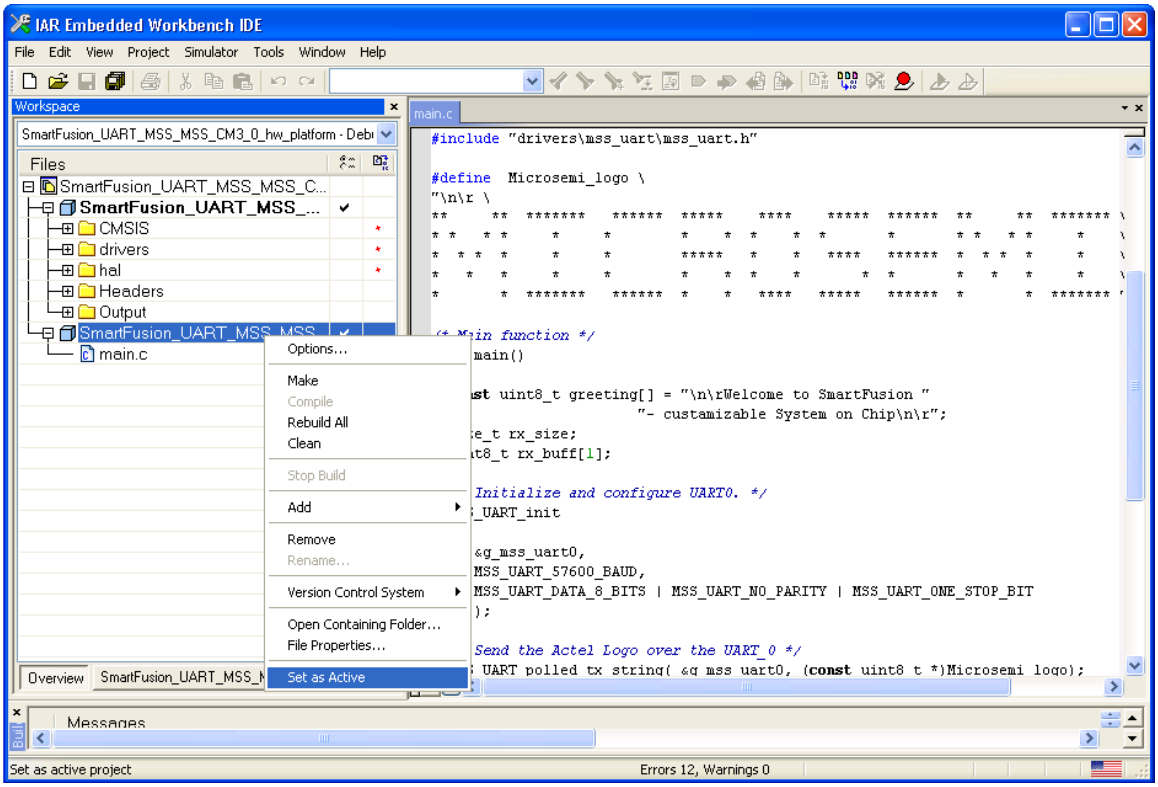

Figure 29 · Select Active Project

16. Right-click **SmartFusion_UART_MSS_MSS_CM3_0_app** and select **Clean**.

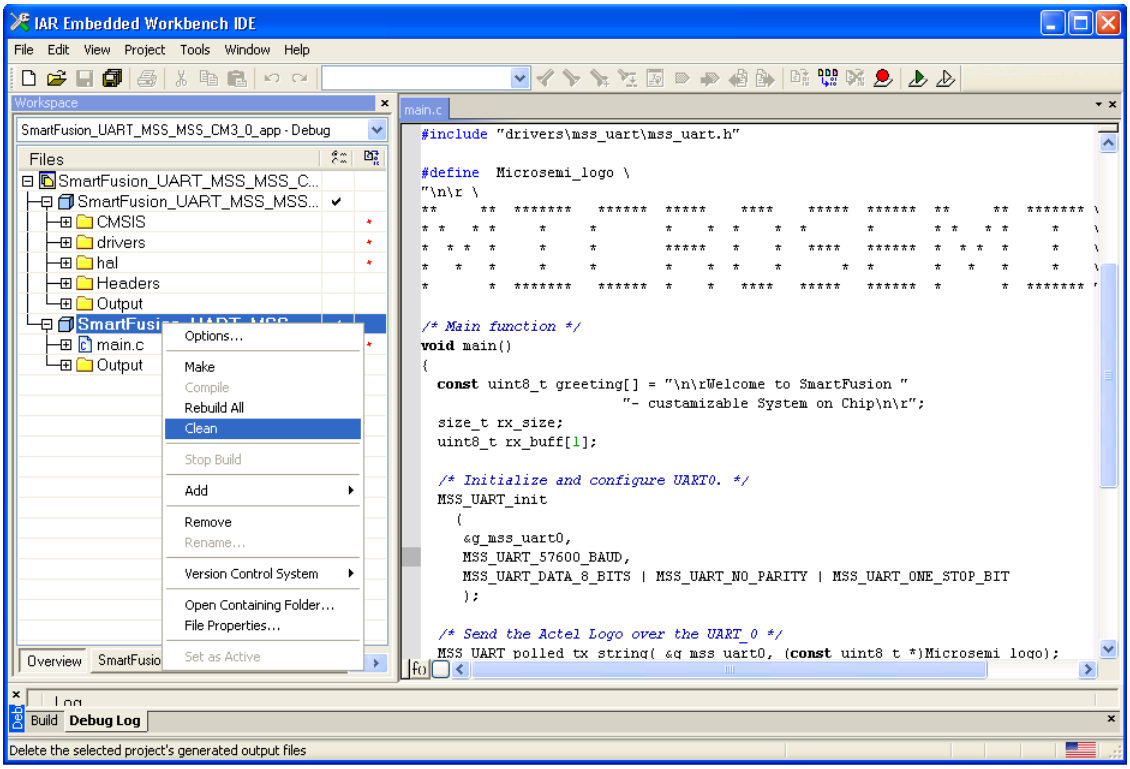

Figure 30 · Building a Clean Project

17. After cleaning the project, a message is displayed in the **Message** log, as shown in Figure 31 · .

Figure 31 · Files Deleted Message

18. Right-click **SmartFusion_UART_MSS_MSS_CM3_0_app** and click **Rebuild All**.
19. Ensure that there are no errors displayed in the **Messages** log. Now follow the next steps to configure HyperTerminal.

Figure 32 · Error-Free Message Log

Step 7 – Configuring the Serial Terminal Emulation Program

Prior to running the application program, you need to configure the terminal emulator program (HyperTerminal, included with Windows®) on your PC. Follow the steps given below to use the SmartFusion Evaluation Kit Board or the SmartFusion Development Kit Board.

1. Connect a second mini USB cable between the USB connector on the SmartFusion Evaluation Kit Board (or the SmartFusion Development Kit Board) and a USB port of your PC. If Windows prompts you to connect to Windows Update, select **No, not at this time** and click **Next**.
2. If the Silicon Labs CP210x USB to UART Bridge drivers are automatically detected (this can be verified in Device Manager), as shown in Figure 33 · , go to the next step. If not, follow Step 8 – Installing Drivers for the USB to RS232 Bridge to install drivers for the USB to RS232 Bridge.

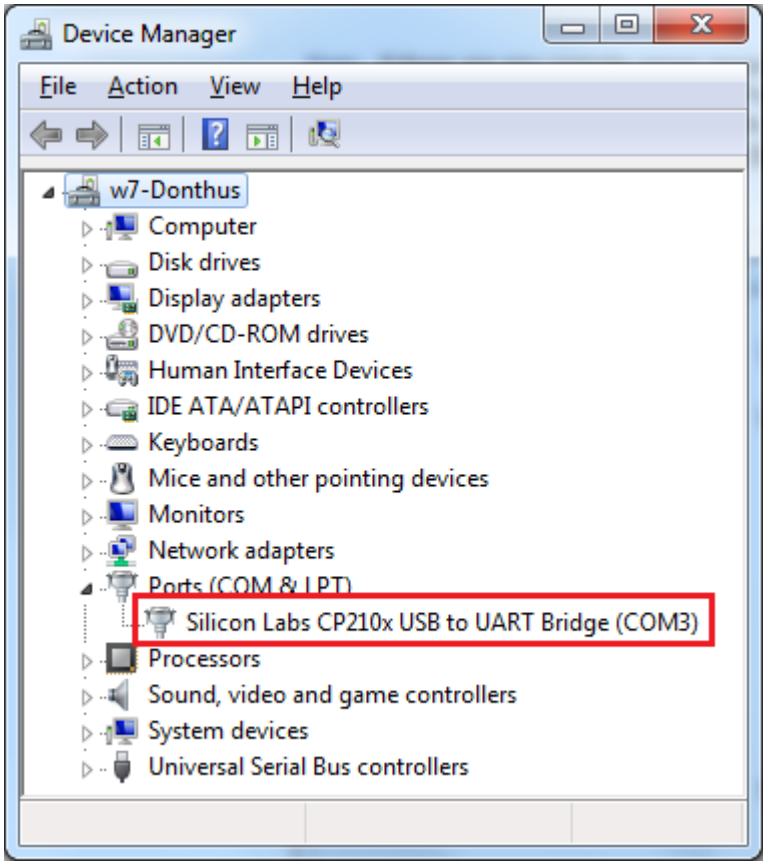

Figure 33 · Device Manager Listing Silicon Labs CP210x USB to UART Bridge Drivers

3. From the Windows **Start** menu, select **Programs > Accessories > Communications > HyperTerminal**. This opens HyperTerminal. If your PC does not have HyperTerminal, use any free serial terminal emulation program such as PuTTY or Tera Term. Refer to the *Configuring Serial Terminal Emulation Programs Tutorial* for configuring HyperTerminal, Tera Term, and PuTTY.
4. Enter **Hyperterminal** in the **Name** field in the **Connection Description** dialog box and click **OK**.

Figure 34 · New Connection

5. Select the appropriate COM port (to which USB-RS232 drivers are pointed) from the **Connect using** drop-down list and click **OK**.

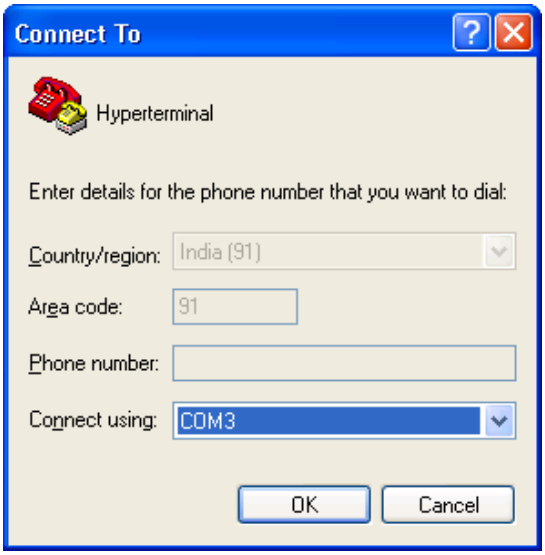

Figure 35 · Selecting the COM Port

6. Set the following in the **COM Properties** window and click **OK**.
   - Bits per second: 57600
   - Data bits: 8
   - Parity: None
   - Stop Bits: 1
   - Flow control: None

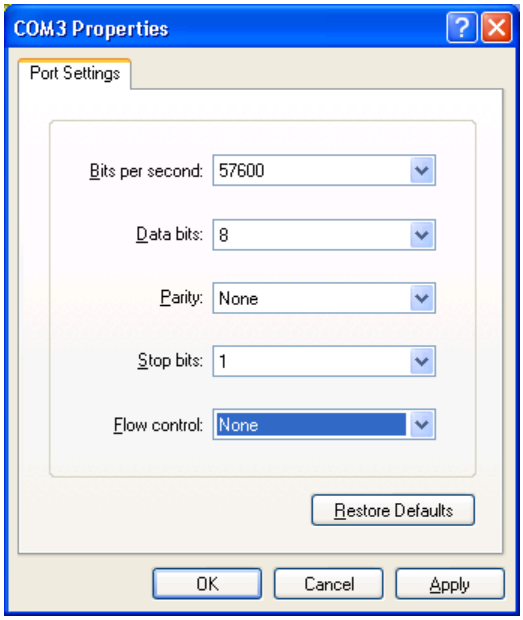

Figure 36 · Setting the COM Properties

7. Click **OK** to close the UART_Hyperterminal Properties dialog box.

   You can directly open HyperTerminal next time (without configuring) by selecting **Programs > Accessories > Communications > HyperTerminal > Hyperterminal**.

Step 8 – Installing Drivers for the USB to RS232 Bridge

Note: To install the USB-RS232 drivers, you should have administrative privileges for your PC.

Use the following steps to install drivers for the USB to RS232 Bridge:

1. Download the USB to RS232 bridge drivers from www.microsemi.com/soc/documents/CP2102_driver.zip.
2. Unzip the CP2102_driver.zip file.
3. Double-click (Run) the CP210x_VCP_Win_XP_S2K3_Vista_7.exe file.
4. Accept the default installation location and click Install.
5. Click Continue Anyway if prompted.
6. When the installation is complete, click OK. The Ports (COM & LPT) section of the Device Manager lists Silicon Labs CP210x USB to UART Bridge under the Ports section of Device Manager.

Step 9 – Debugging the Application Project Using IAR

Follow the steps given below to debug the application project using IAR:

1. Connect J-link Box to Board and PC using the following connection tips:

Connections:

- Connect the J-Link ARM to the RVI-Header of the SmartFusion Evaluation Kit Board or the SmartFusion Development Kit Board.
- JP7: Connect pin 2 and 3 for IAR debugging mode.
- JP10: Connect pin 2 and 3 for SmartFusion Evaluation Kit Board. SW9 must be in ON position for SmartFusion Development Kit Board.
- J6: Connect pin 1 and 2.

2. In the IAR Workbench, click **Download and Debug**.

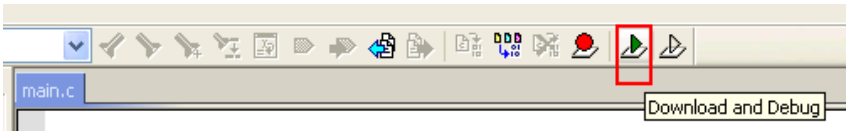

Figure 37 · Download and Debug the Project

3. The Workbench is displayed, as shown in Figure 38 · .

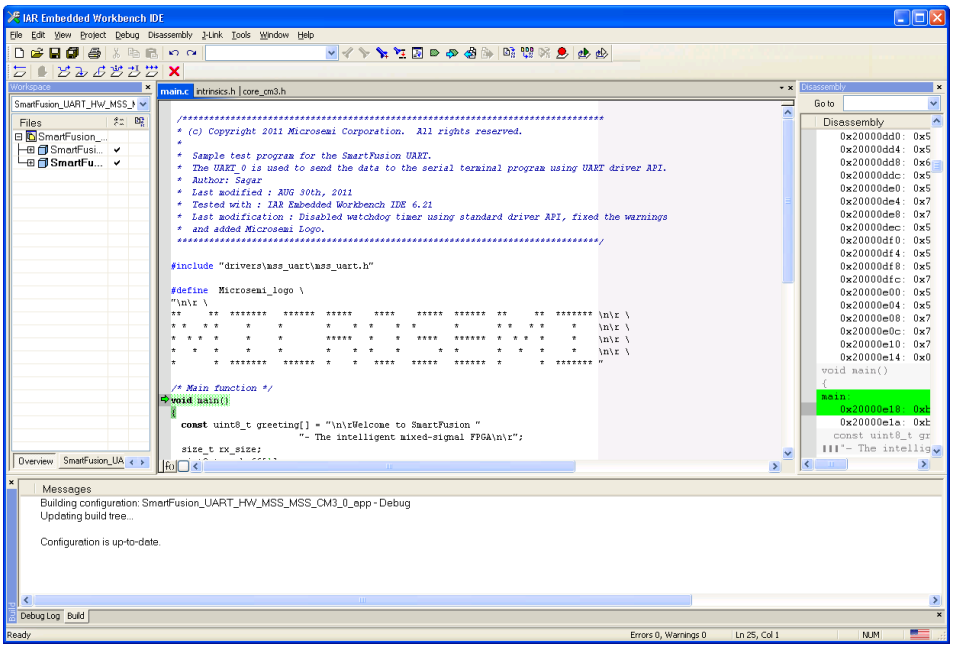

Figure 38 · IAR Workbench After Downloading the Project

4. Click **Go**.

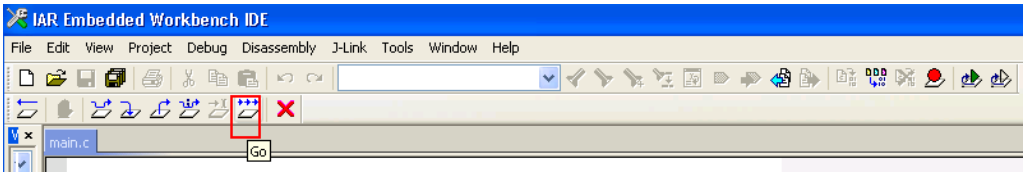

Figure 39 · Run the Project

5. Observe the HyperTerminal window. It should display the greeting message with the Microsemi name, as shown in Figure 40 · .

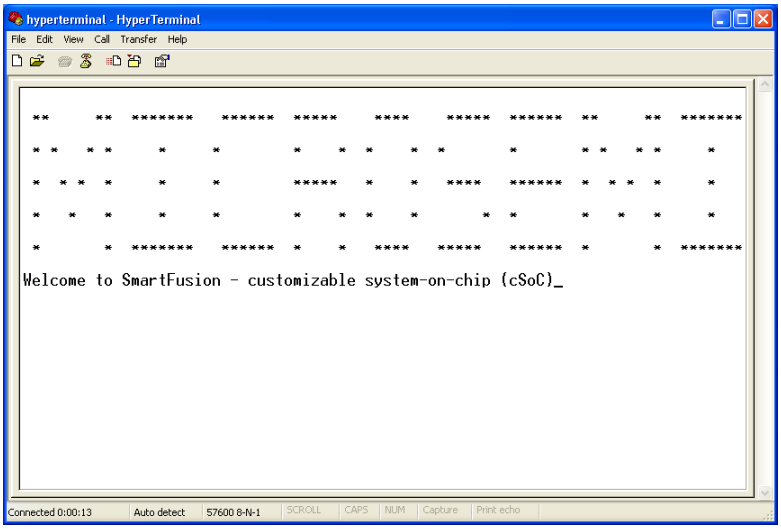

Figure 40 · UART HyperTerminal Window

Note: If you try typing some alphabetic characters, you will find they are echoed in the terminal window.

6. To check the status of the UART0 registers, click **View > Register**. In the Register pane, select **UART_0** from the drop-down list.

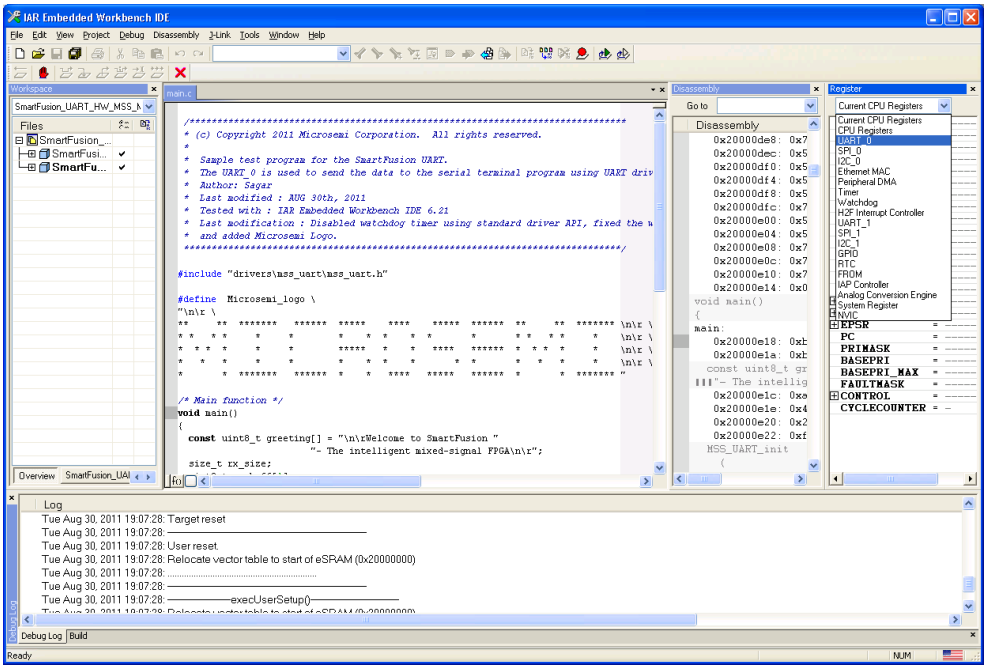

Figure 41 · Viewing Registers

7. The UART registers are displayed, as shown in Figure 42 · .

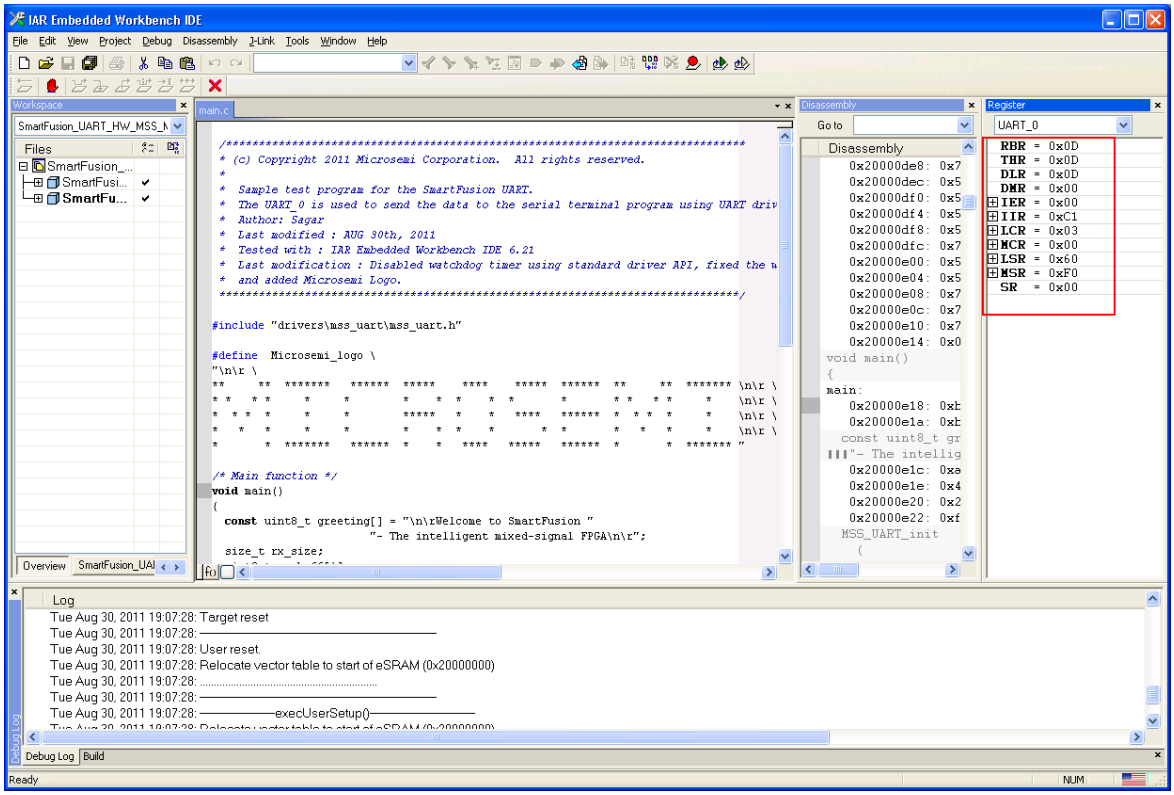

Figure 42 · Viewing UART_0 in the Registers Pane

8. To stop debugging, click **Stop Debugging**.

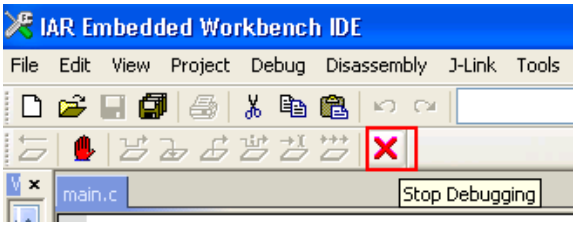

Figure 43 · Stop the Debugging

This concludes the tutorial.

Appendix A – Libero SoC Catalog Settings

Listed below are the steps for configuring the vault location and setting up the repositories in Libero SoC.

1. On the **Catalog** window, click **Options**.

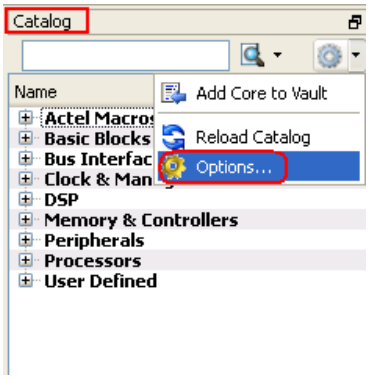

Figure 44 · Catalog – Options

2. The **Options** window is displayed. Click **Repositories** under **Vault/Repositories Settings** and add the following details in the address field.
   - www.actel-ip.com/repositories/SgCore
   - www.actel-ip.com/repositories/DirectCore
   - www.actel-ip.com/repositories/Firmware

   Note: Click **Add** after entering each path.

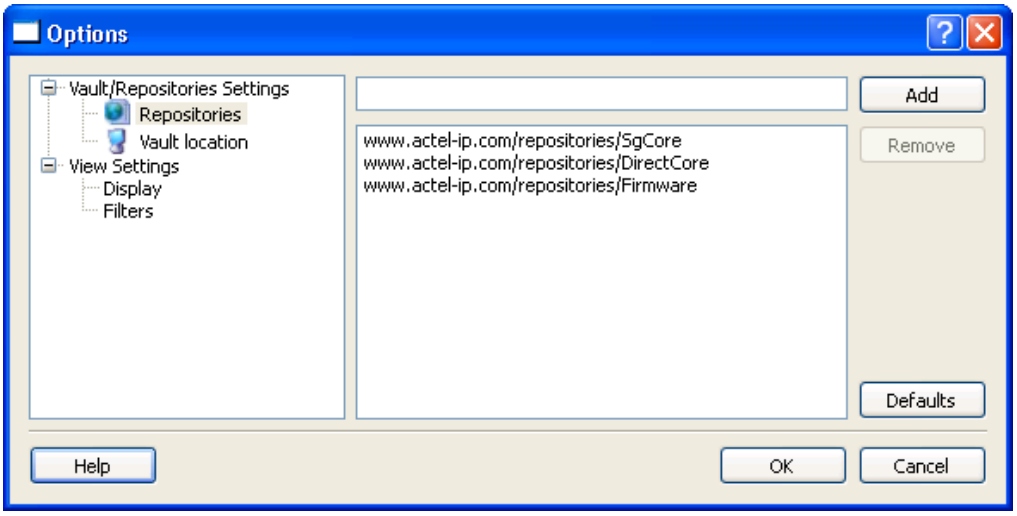

Figure 45 · Setting Repositories

3. Click **Vault location** under **Vault/Repositories Settings** in the **Options** window. Browse to a location on your PC to set the vault location where the IPs can be downloaded from the repositories.

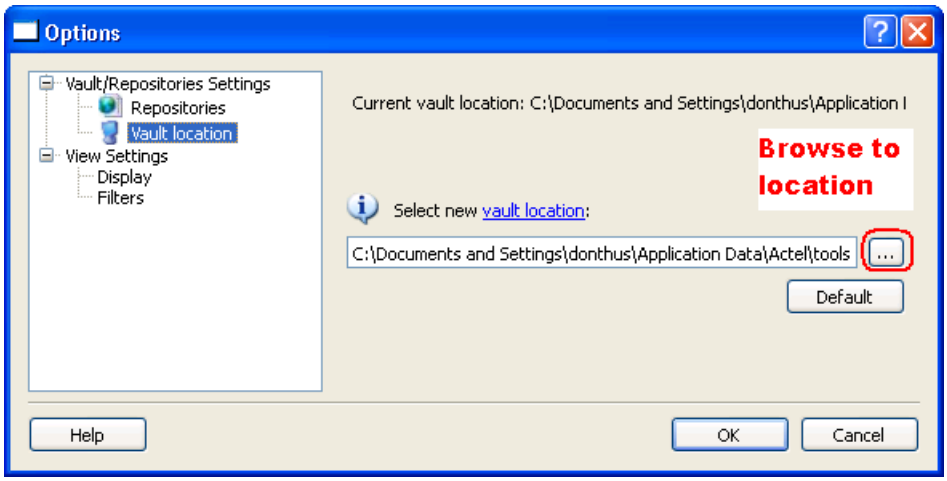

Figure 46 · Setting the Vault Location

4. Click **OK**.

Appendix B – Firmware Catalog Settings

1. Open the **<Libero Installation directory>\Designer\bin\catalog.exe**.
2. Select **Tools > Vault/Repositories Settings**, from the **Firmware Catalog** window.

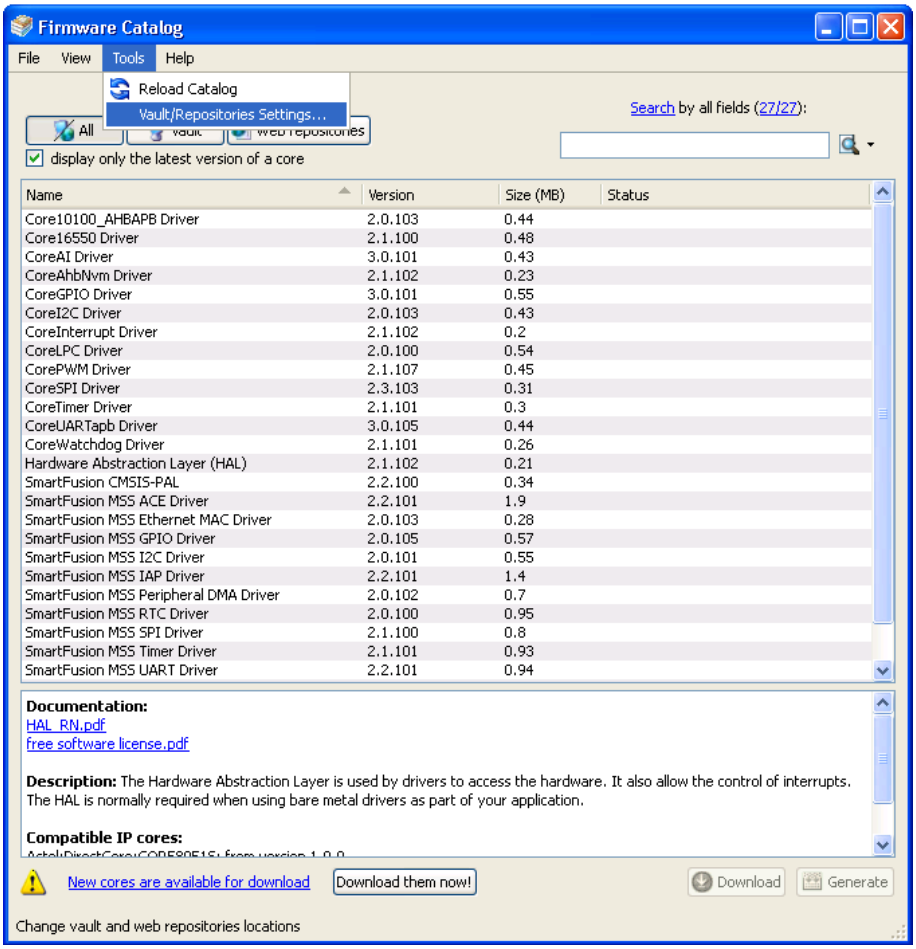

Figure 47 · Firmware Catalog Settings

3. Select **Repositories** under **Vault/Repositories Settings** in the **Options** dialog box.
4. Confirm that the following repositories are displayed (add them, if required):
   - www.actel-ip.com/repositories/SgCore
   - www.actel-ip.com/repositories/DirectCore
   - www.actel-ip.com/repositories/Firmware
5. Add the above paths in the address field, if required, by selecting the repository and clicking **Add**.

   If new cores are available for download, click **Download them now!** to download the new cores to the vault.

List of Changes

Revision	Changes	Page
Revision 6 (May 2012)	Modified Software Requirements under Tutorial Requirements section (SAR 38346)	4
	Modified Step 2 – Configuring MSS Peripherals (SAR 38346)	9
	Replaced Figure 15 · (SAR 38346)	14
	Replaced Figure 16 · (SAR 38346)	16
	Modified Step 6 – Building the Software Application Through IAR Embedded Workbench (SAR 38346)	17
	Replaced Figure 17 · (SAR 38346)	17
	Modified Step 9 – Debugging the Application Project Using IAR (SAR 38346)	28
Revision 5 (February 2012)	Modified Associated Project Files for IAR Workbench section (SAR 36894).	4
	Modified Step 7 – Configuring the Serial Terminal Emulation Program section (SAR 36894).	25
	Modified Figure 33 · (SAR 36894).	25
	Modified Step 8 – Installing Drivers for the USB to RS232 Bridge section (SAR 36894).	28
	Modified Step 9 – Debugging the Application Project Using IAR section (SAR 36894).	28
Revision 4 (November 2011)	Updated the document for Libero SoC v10.0 (SAR 35047).	N/A

Note: The revision number is located in the part number after the hyphen. The part number is displayed at the bottom of the last page of the document. The digits following the slash indicate the month and year of publication.

Product Support

Microsemi SoC Products Group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, electronic mail, and worldwide sales offices. This appendix contains information about contacting Microsemi SoC Products Group and using these support services.

Customer Service

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

From North America, call **800.262.1060**
From the rest of the world, call **650.318.4460**
Fax, from anywhere in the world **408.643.6913**

Customer Technical Support Center

Microsemi SoC Products Group staffs its Customer Technical Support Center with highly skilled engineers who can help answer your hardware, software, and design questions about Microsemi SoC Products. The Customer Technical Support Center spends a great deal of time creating application notes, answers to common design cycle questions, documentation of known issues and various FAQs. So, before you contact us, please visit our online resources. It is very likely we have already answered your questions.

Technical Support

Visit the Microsemi SoC Products Group Customer Support website for more information and support (http://www.microsemi.com/soc/support/search/default.aspx). Many answers available on the searchable web resource include diagrams, illustrations, and links to other resources on website.

Website

You can browse a variety of technical and non-technical information on the Microsemi SoC Products Group home page, at http://www.microsemi.com/soc/.

Contacting the Customer Technical Support Center

Highly skilled engineers staff the Technical Support Center. The Technical Support Center can be contacted by email or through the Microsemi SoC Products Group website.

Email

You can communicate your technical questions to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. We constantly monitor the email account throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.

The technical support email address is soc_tech@microsemi.com.

My Cases

Microsemi SoC Products Group customers may submit and track technical cases online by going to My Cases.

Outside the U.S.

Customers needing assistance outside the US time zones can either contact technical support via email (soc_tech@microsemi.com) or contact a local sales office. Sales office listings can be found at www.microsemi.com/soc/company/contact/default.aspx.

ITAR Technical Support

For technical support on RH and RT FPGAs that are regulated by International Traffic in Arms Regulations (ITAR), contact us via soc_tech_itar@microsemi.com. Alternatively, within My Cases, select **Yes** in the ITAR drop-down list. For a complete list of ITAR-regulated Microsemi FPGAs, visit the ITAR web page.

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo CA 92656 USA
Within the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996

Microsemi Corporation (NASDAQ: MSCC) offers a comprehensive portfolio of semiconductor solutions for: aerospace, defense and security; enterprise and communications; and industrial and alternative energy markets. Products include high-performance, high-reliability analog and RF devices, mixed signal and RF integrated circuits, customizable SoCs, FPGAs, and complete subsystems. Microsemi is headquartered in Aliso Viejo, Calif. Learn more at **www.microsemi.com**.

50200214-6/05.12