Using Identify with Libero SoC v11.7

TU0071 Tutorial

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA

Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
E-mail: sales.support@microsemi.com

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for communications, defense & security, aerospace and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; Enterprise Storage and Communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif, and has approximately 4,800 employees globally. Learn more at **www.microsemi.com**.

50200071-2/04.16

1 Revision History

The following table shows the important changes made in this document for each revision.

Revision	Changes
Revision 2 (April 2016)	Updated for Libero SoC tool flow.
Revision 1	Initial release.

Contents

1 Revision History . . . 3

2 Preface . . . 6
- 2.1 Purpose . . . 6
- 2.2 Intended Audience . . . 6

3 Using Identify with Libero SoC v11.7 (Classic Constraint Flow) . . . 7
- 3.1 Introduction . . . 7
- 3.2 Identify Flow . . . 7
- 3.3 How to Use Identify with Libero SoC v11.7 . . . 9
  - 3.3.1 Step One: Create a Libero SoC Project . . . 9
  - 3.3.2 Step Two: Create an Identify Implementation . . . 9
  - 3.3.3 Step Three: Launch Identify Instrumentor . . . 12
  - 3.3.4 Step Four: Instrument the Design . . . 13
  - 3.3.5 Step Five: Change the Implementation View . . . 14
  - 3.3.6 Step Six: Launch and View the New Implementation . . . 14
  - 3.3.7 Step Seven: Import the EDIF Implementation File into Libero SoC . . . 15
  - 3.3.8 Step Eight: Launch Identify Debugger . . . 17

4 Product Support . . . 19
- 4.1 Customer Service . . . 19
- 4.2 Customer Technical Support Center . . . 19
- 4.3 Technical Support . . . 19
- 4.4 Website . . . 19
- 4.5 Contacting the Customer Technical Support Center . . . 19
  - 4.5.1 Email . . . 19
  - 4.5.2 My Cases . . . 19
  - 4.5.3 Outside the U.S. . . . 20
- 4.6 ITAR Technical Support . . . 20

Figures

Figure 1. Identify Flow Diagram . . . 8
Figure 2. Libero SoC Project . . . 9
Figure 3. Configure Identify Launch in Synplify Pro . . . 10
Figure 4. Identify Instrumentor set to Legacy mode . . . 10
Figure 5. Select Identify Instrumentor . . . 11
Figure 6. Create New Identify Implementation . . . 11
Figure 7. Identify Implementation Location . . . 12
Figure 8. Launching Identify Instrumentor . . . 12
Figure 9. Instrumentation with Identify Instrumentor . . . 13
Figure 10. Running Final synthesis on Instrumented design . . . 14
Figure 11. Re-Launching Identify in Synplify Pro . . . 14
Figure 12. Importing an EDN File in Libero SoC . . . 15
Figure 13. Libero Compile,PR, Generate Programming and Run Program Action . . . 16
Figure 14. Set Identify Debugger Profile within Libero . . . 17
Figure 15. Set the Cable Type . . . 18
Figure 16. Identify Debugger . . . 18

2 Preface

2.1 Purpose

This tutorial provides instructions on how to use the Identify RTL Debugger with the Libero®System-on-Chip (SoC) v11.7 software.

2.2 Intended Audience

This tutorial is intended for:

- FPGA designers
- System-level designers

3 Using Identify with Libero SoC v11.7 (Classic Constraint Flow)

3.1 Introduction

Identify is an RTL Debugger that allows you to probe internal signals of the design and view the signals directly from the programmed FPGA in the original RTL code or in a waveform viewer.

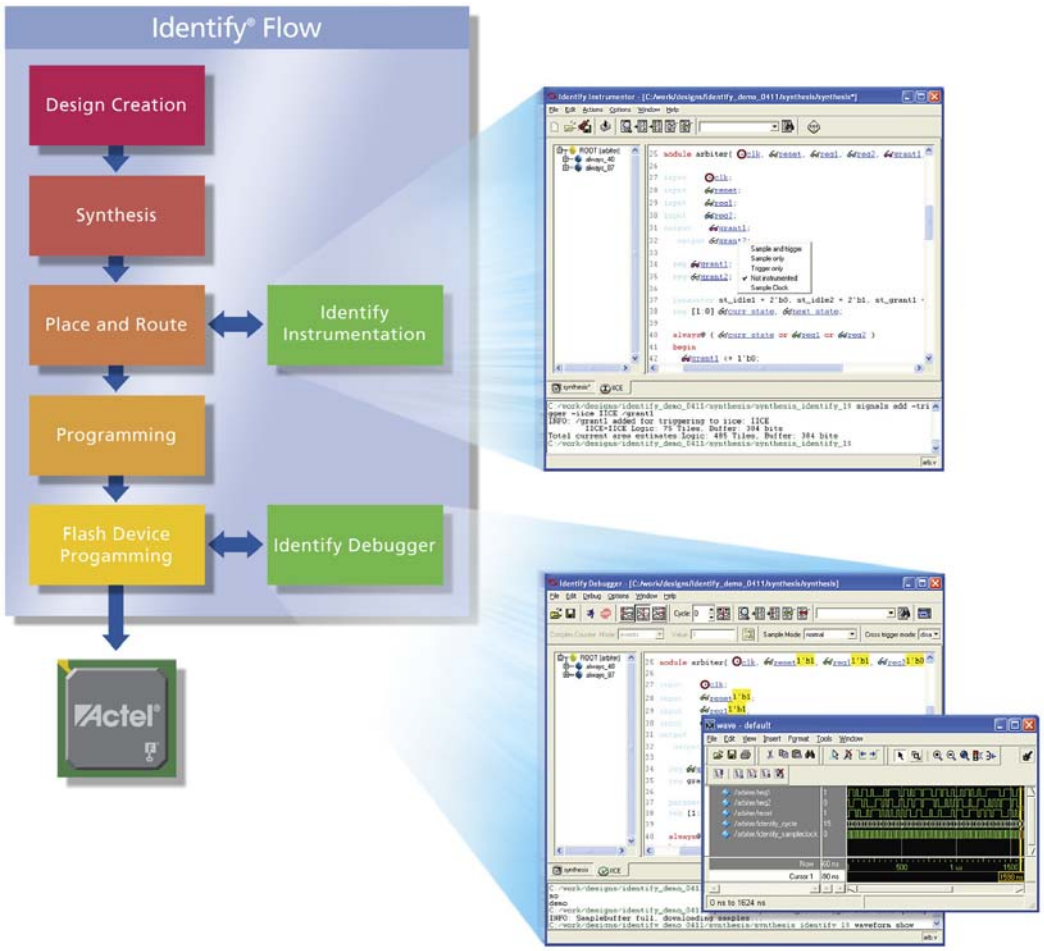
Identify consists of two separate tools—Instrumentor and Debugger. The Instrumentor compiles and inserts the logic into the original RTL to create an instrumented design that allows probing of internal signals. The Debugger communicates to the FlashPro, FlashPro Lite, and FlashPro3/4/5 devices via JTAG to get the values for the probed signals directly from the programmed FPGA and displays those signals.

This tutorial shows how Identify Instrumentor and Debugger can be used within the Libero SoC flow. It also explains how Identify Instrumentor works with Synplify Pro. It gives you step-by- step instructions with snapshots of the tools.

3.2 Identify Flow

The flow diagram in Figure 1 on page 8 shows the debug flow of Identify. Before synthesis, the RTL design is instrumented using Identify ME Instrumentor. Instrumentation is a process of creating instrumented HDL sources that can be used to debug the design in Identify ME Debugger. This allows you to set the sample signals, trigger points for conditional sampling, and watchpoints for allowing specific event-driven sampling of the signals within the design. After this is done, Identify ME Debugger communicates with the instrumented source and captures the operation of the device via the Flash device programmer. The sampled signal values can be viewed directly in the RTL or on a waveform viewer.

Figure 1 • **Identify Flow Diagram**



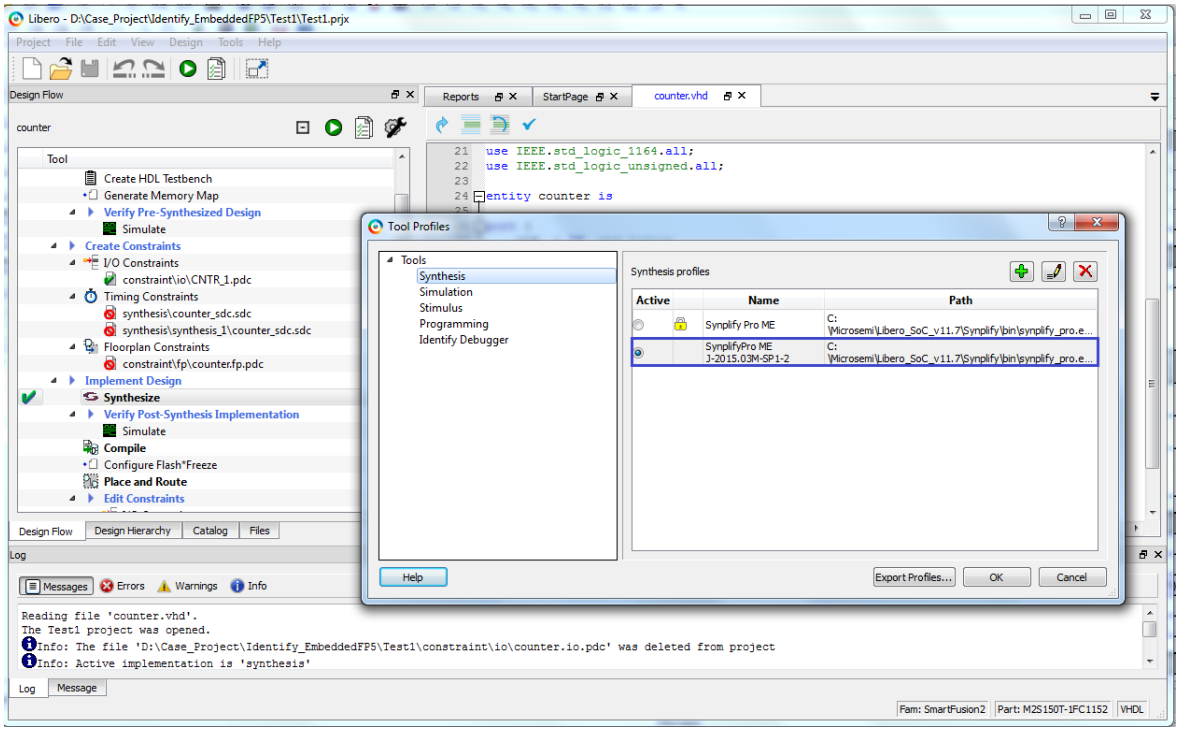
3.3 How to Use Identify with Libero SoC v11.7

The following steps describe how to use the Identify Instrumentor and Debugger tools with Libero SoC v11.7:

3.3.1 Step One: Create a Libero SoC Project

Create a new project in Libero SoC and select Synplify Pro ME (for Libero SoC Platinum customers) as the synthesis tool in the Tool Profiles window, as shown in Figure 2.

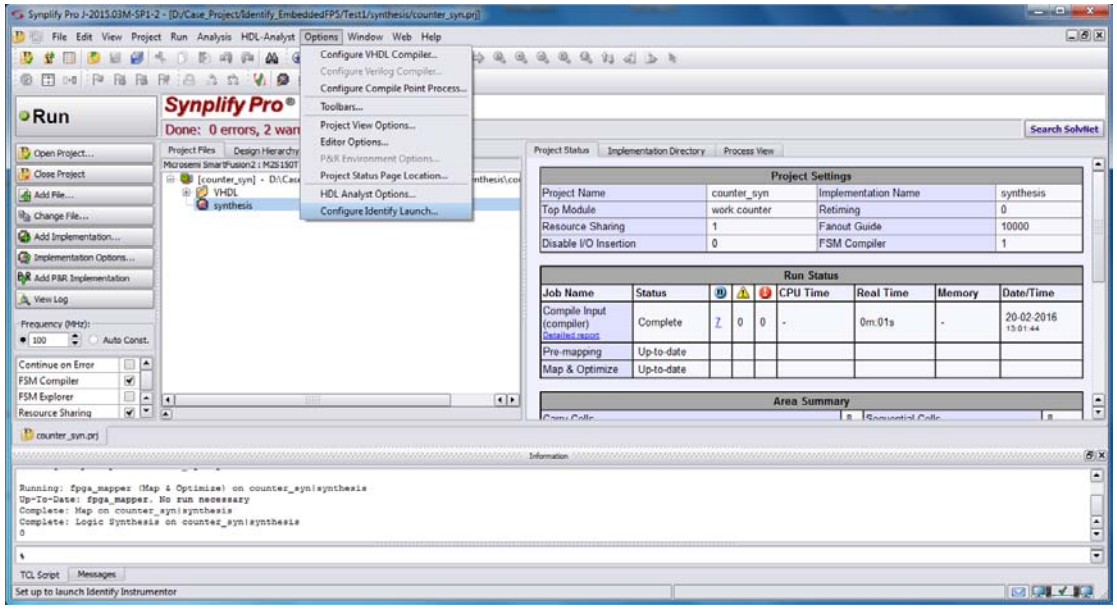
Figure 2 • **Libero SoC Project**



3.3.2 Step Two: Create an Identify Implementation

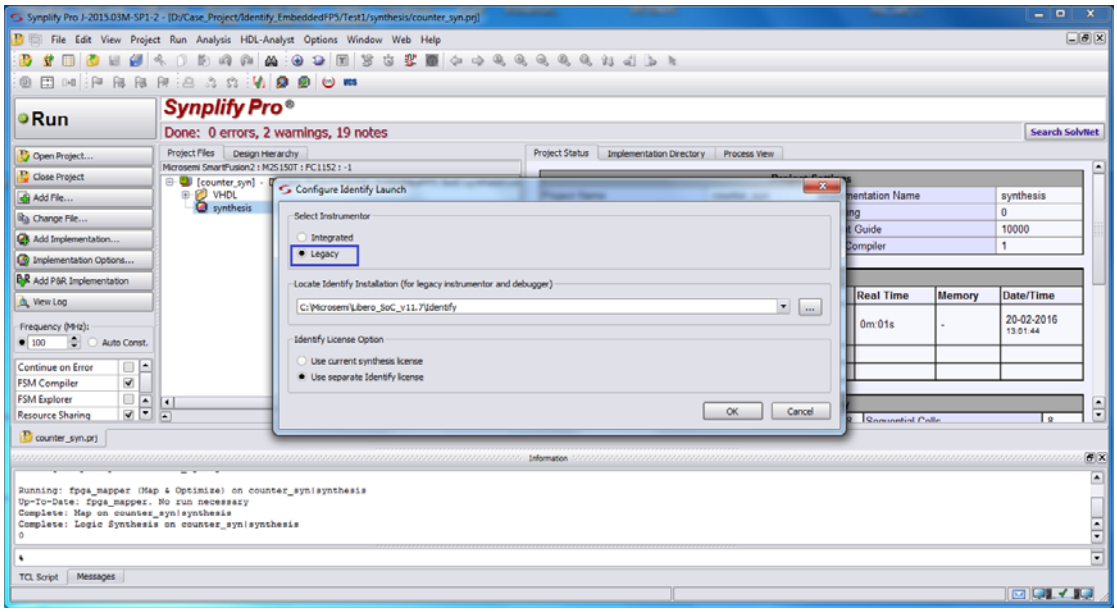
1. After running synthesis (interactively) with the non-instrumented design in Synplify Pro, create an Identify implementation.
2. In SynplifyPro, go to the **Options** menu and click **Configure Identify Launch**, as shown in Figure 3 on page 10.

Figure 3 • **Configure Identify Launch in Synplify Pro**



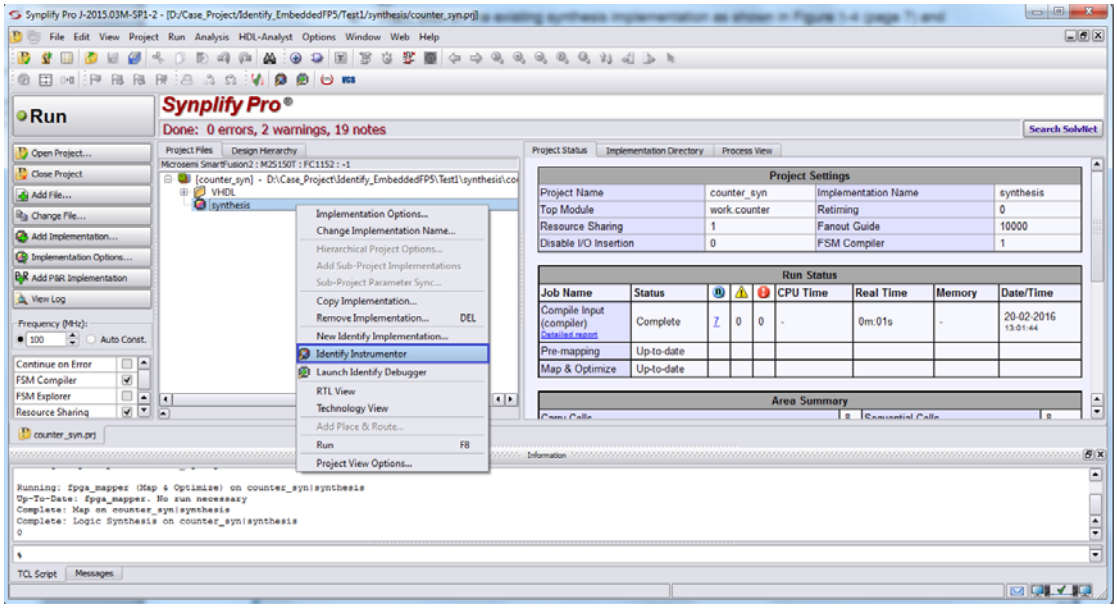
3. In the **Configure Identify Launch** window, select Instrumentor as **Legacy**, as shown in Figure 4.

Figure 4 • **Identify Instrumentor set to Legacy mode**



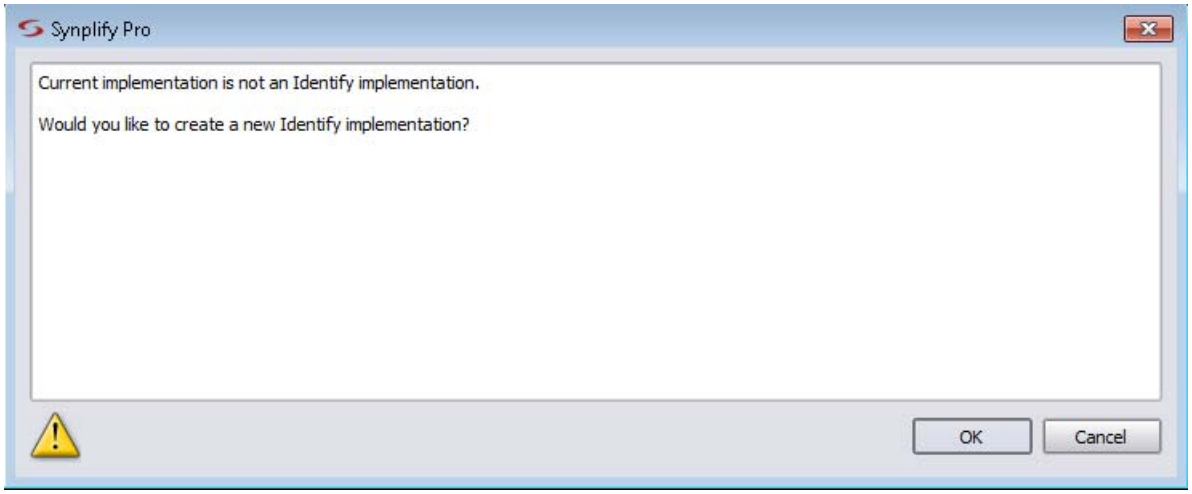
4. Right-click **synthesis** and select **Identify Instrumentor**. Do not select **New Identify Implementation**.

Figure 5 • **Select Identify Instrumentor**



5. Click **OK** on the dialog box, as shown in Figure 6 on page 11.

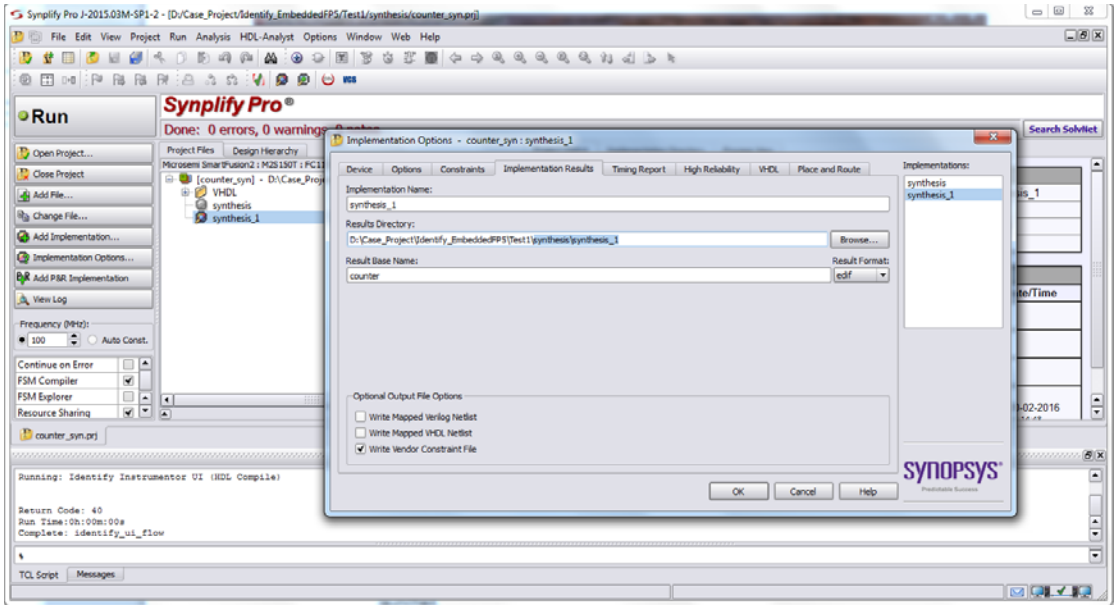
Figure 6 • **Create New Identify Implementation**



SynplifyPro creates a new Identify implementation with the default location at *<Libero_Project_Location>\synthesis\synthesis_1,* as shown in Figure 7 on page 12.

You can edit the implementation name in the **Implementation Results** tab. Do not change the path of the Results Directory.

Figure 7 • **Identify Implementation Location**



Note: The Identify implementation folder must be under the synthesis folder of the Libero project. This is required for Libero to pick the correct Identify-instrumented synthesis netlist.

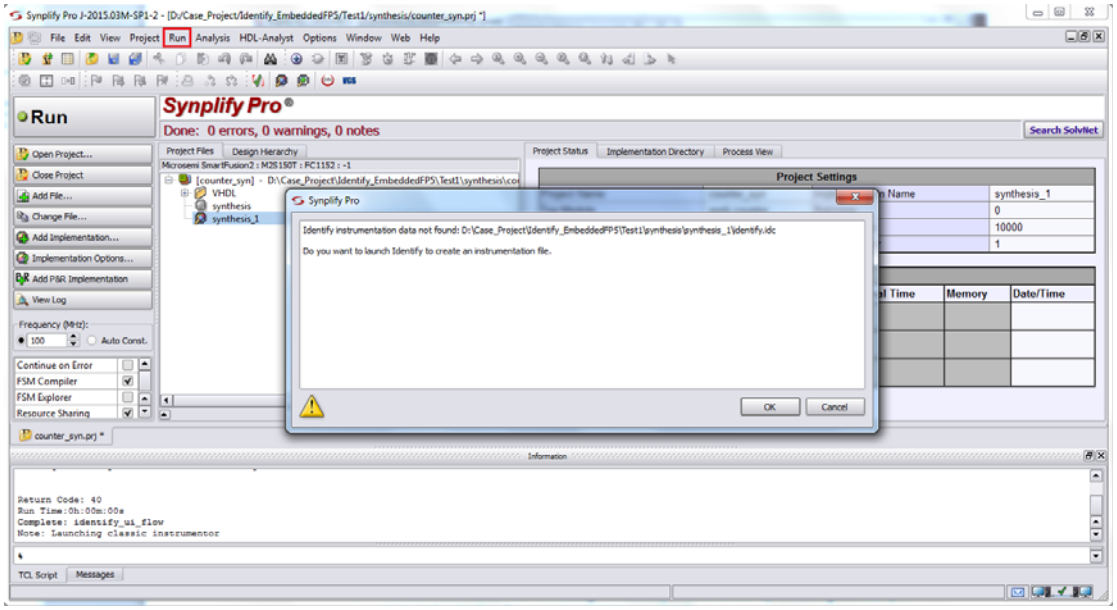
3.3.3 Step Three: Launch Identify Instrumentor

After creating a new Identify implementation, Identify Instrumentor is automatically launched.

Another way of launching Identify Instrument is by clicking **Run** in Synplify Pro. The **Identify Instrumentation data not found** message is displayed, as shown in Figure 8.

This message is shown because the design is not yet instrumented with Identify Instrumentor, and the instrumented database (*instr.db*) cannot be located.The instrumented database is created by Identify Instrumentor. Click **OK** to launch Identify Instrumentor.

Figure 8 • **Launching Identify Instrumentor**

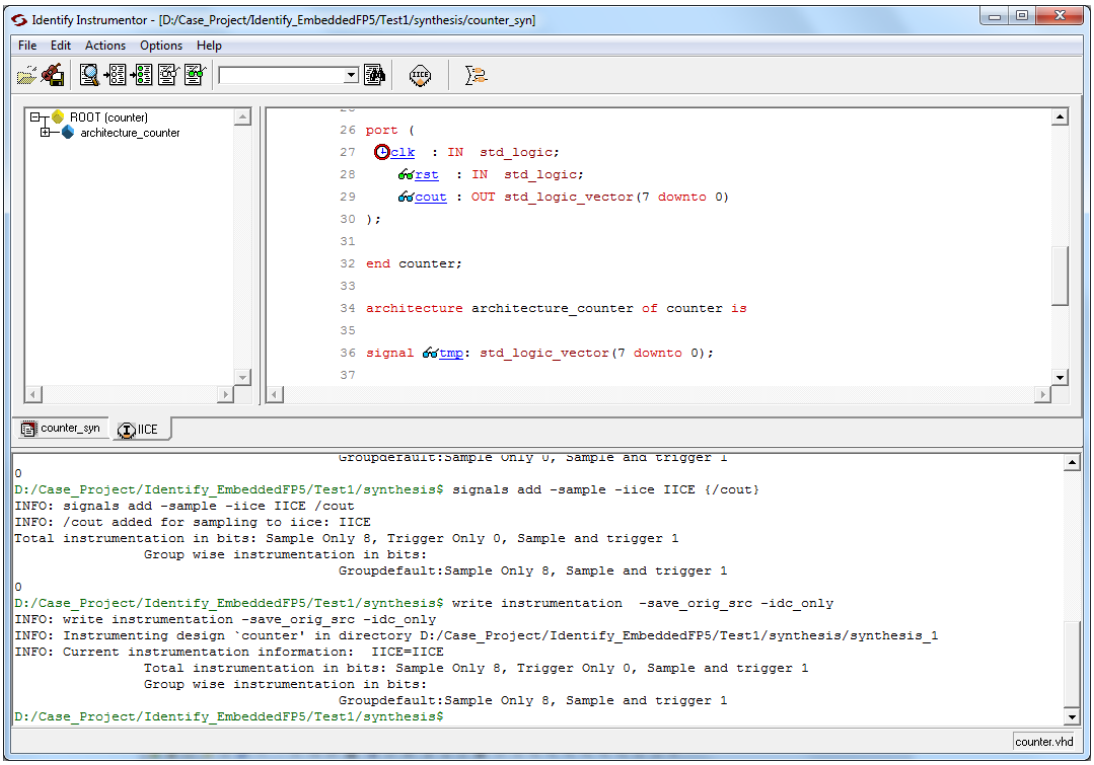


3.3.4 Step Four: Instrument the Design

To instrument the design in Identify Instrumentor, select the sample clock and the sample and trigger signals.

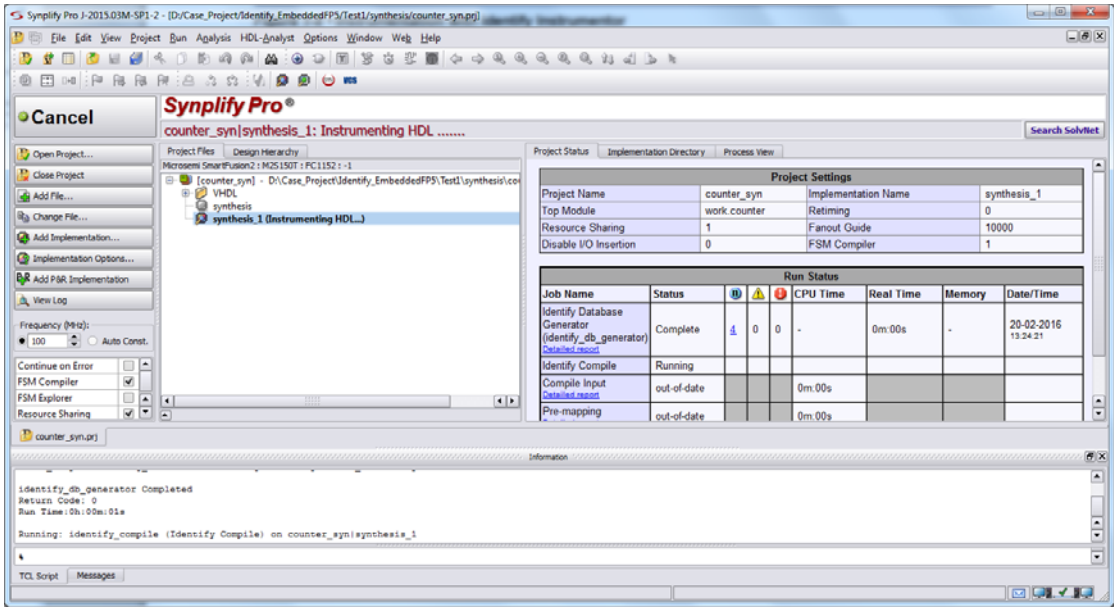
In the **Identify Instrumentor** window, the signals available for instrumentation have icons in front of them, as shown in Figure 9.

Figure 9 • **Instrumentation with Identify Instrumentor**



1. To select the sample clock, right-click the signal name and select **sample clock**. In the same way, select **sample and trigger signals**.
2. Click **File > Save Project Instrumentation** to save and instrument the design.
3. Close Instrumentor and run the final synthesis in Synplify Pro, as shown in Figure 10. Synplify Pro then maps the design into logic gates.

Figure 10 • **Running Final synthesis on Instrumented design**



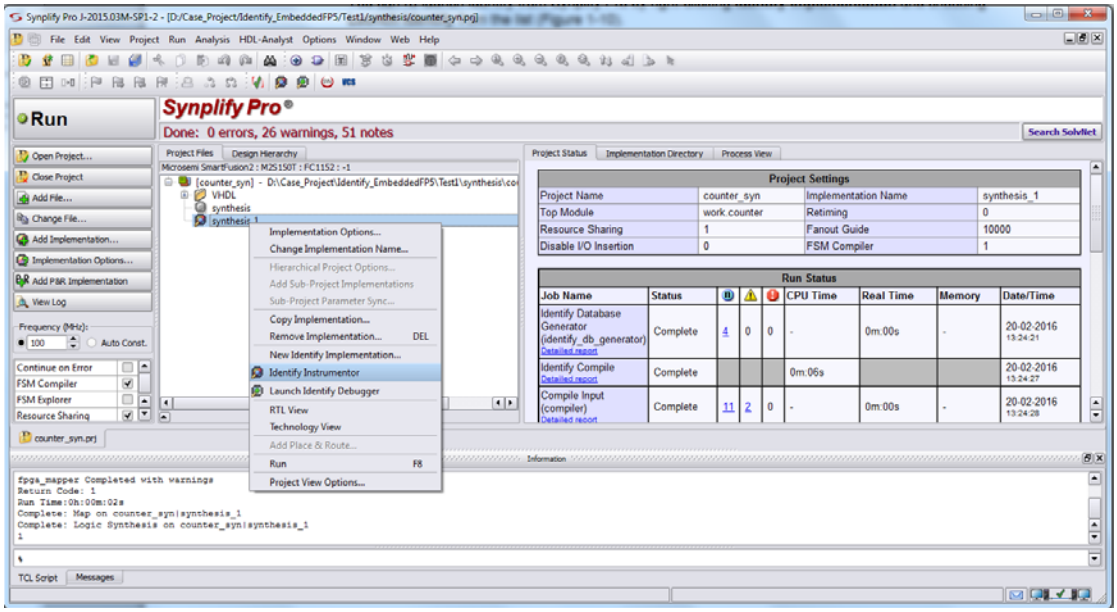
3.3.5 Step Five: Change the Implementation View

To change the implementation view in Synplify Pro, select the implementation. Synplify Pro allows multiple implementation views at the same time.

3.3.6 Step Six: Launch and View the New Implementation

To re-launch Identify from Synplify Pro, right-click the Identify implementation and select **Launch Identify**, as shown in Figure 11 on page 14.

Figure 11 • **Re-Launching Identify in Synplify Pro**

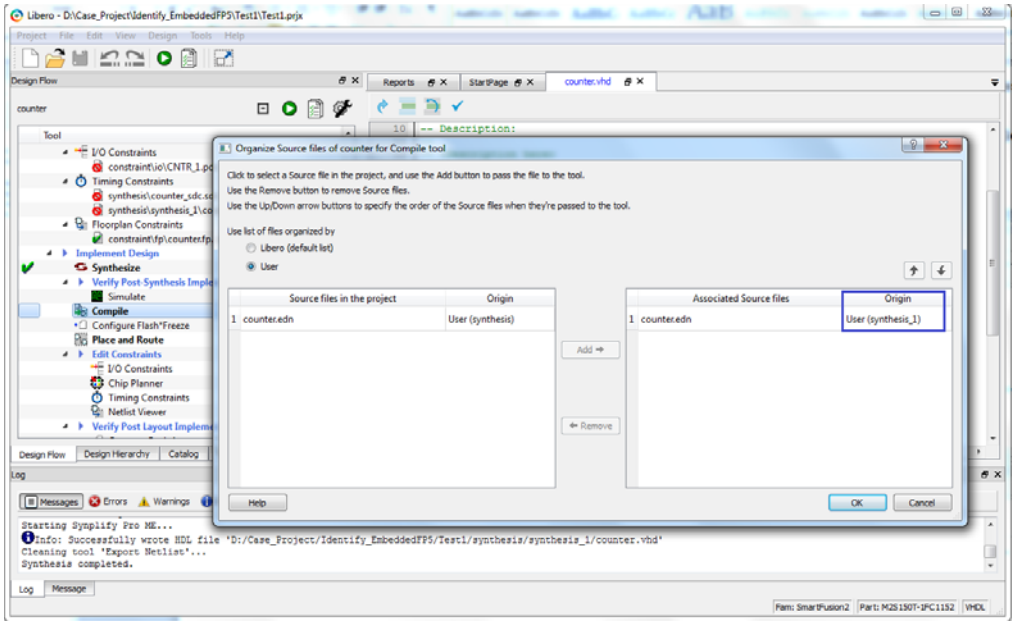


3.3.7 Step Seven: Import the EDIF Implementation File into Libero SoC

Libero SoC automatically selects the `*.edn` netlist from the Identify implementation folder (synthesis_1).

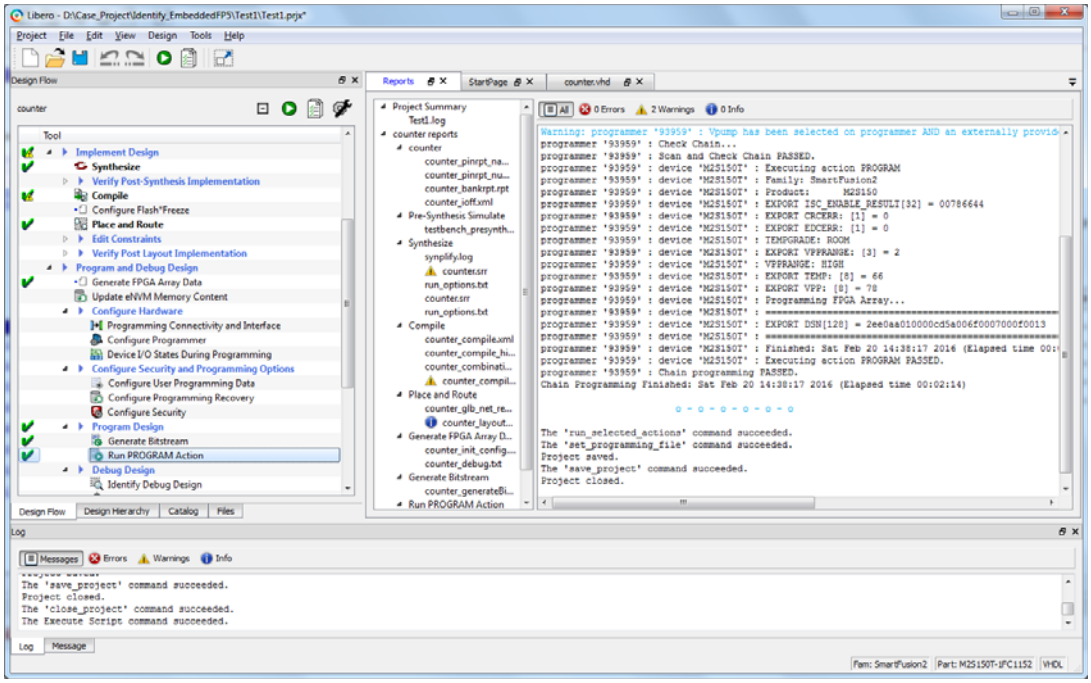
1. Run **Compile** and **Place and Route**.
2. Alternately, in the Libero Design Flow window, right-click **Compile**, select **Organize input files** followed by **Organize Source files**.

Figure 12 • **Importing an EDN File in Libero SoC**



3. Add an instrumented `*.edn` file.
4. After adding the input file, generate the programming file by clicking **Generate programming file**.
5. Program the part by double-clicking **Run PROGRAM Action** as shown in Figure 13.

Figure 13 • **Libero Compile,PR, Generate Programming and Run Program Action**

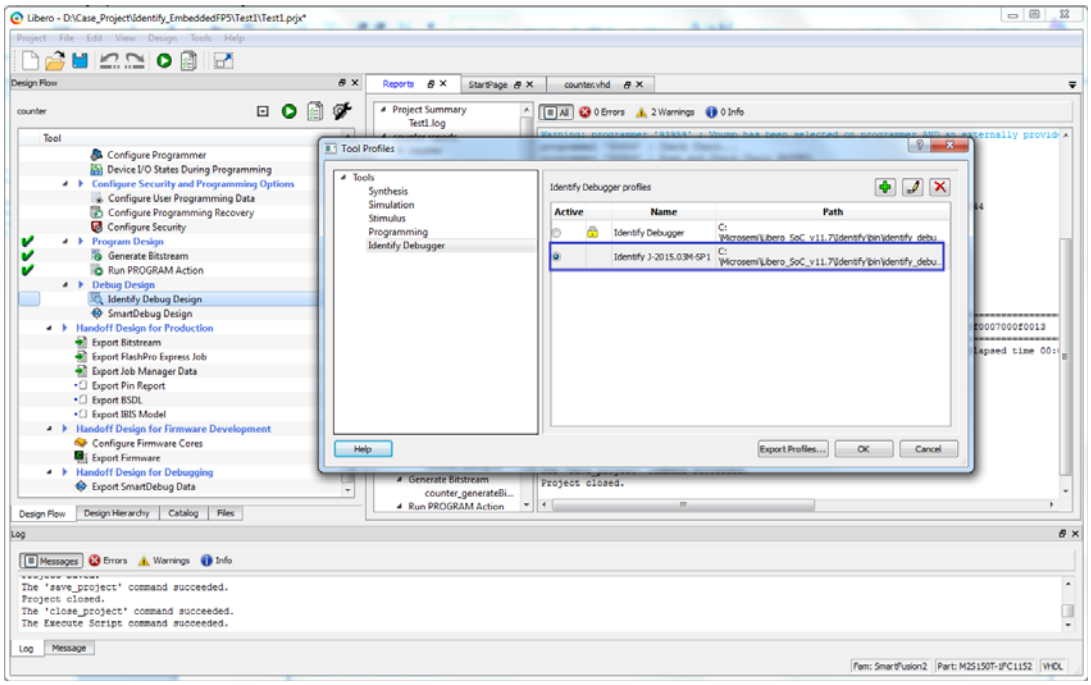


3.3.8 Step Eight: Launch Identify Debugger

After programming the device using the FlashPro software, open the Identify Debugger through the Libero Design Flow.

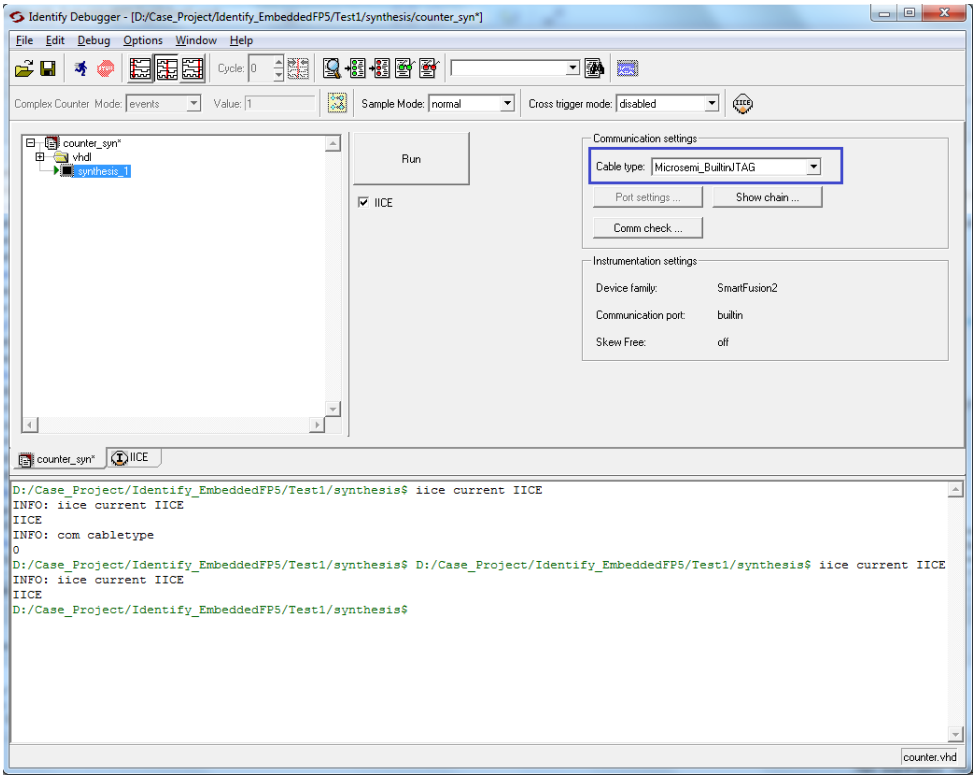
1. Click **Identify Debug Design > Edit Profile** as shown in Figure 14.

Figure 14 • **Set Identify Debugger Profile within Libero**



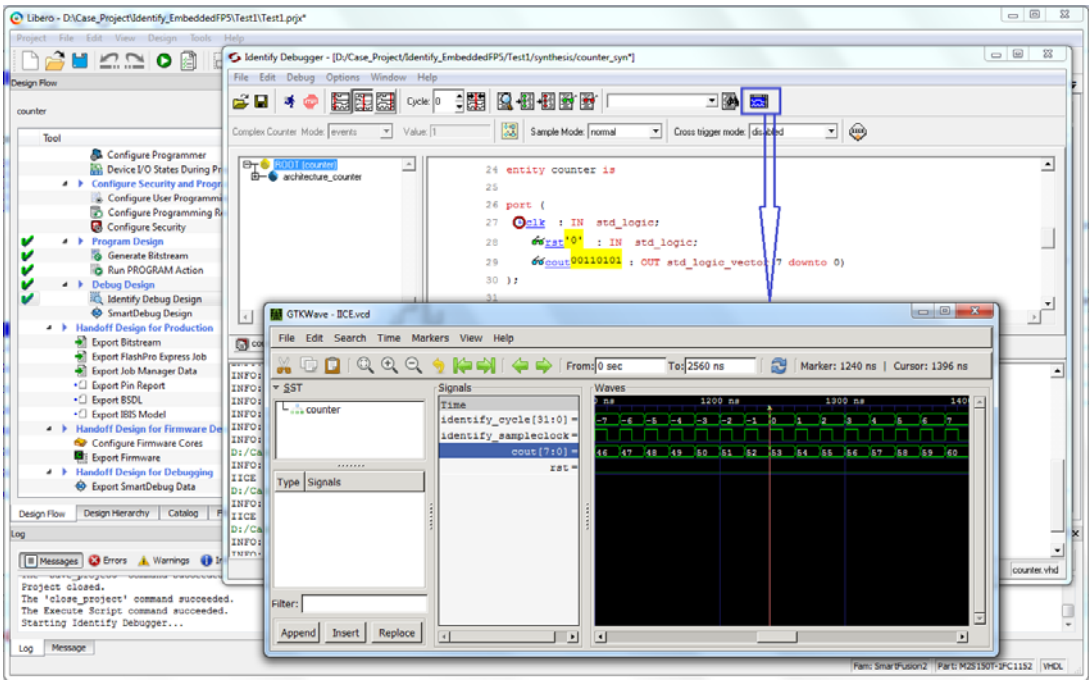
2. Open the `*.prj` file that was created earlier by SynplifyPro/Identify Instrumentor in Identify Debugger.
3. Set the cable type to **Microsemi_BuiltinJTAG**, as shown in Figure 15 on page 18.

Figure 15 • **Set the Cable Type**



4. Click **Run**. The Identify Debugger stops when a trigger condition occurs and displays the signal values in the **GTKWave** viewer, as shown in Figure 16.

Figure 16 • **Identify Debugger**



Note: For using different features and options within the Identify Instrumentor or Debugger tool, see the Identify Tutorial from Synopsys on the Microsemi website.

4 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.

4.1 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

4.2 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.

4.3 Technical Support

For Microsemi SoC Products Support, visit
http://www.microsemi.com/products/fpga-soc/design-support/fpga-soc-support.

4.4 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/products/fpga-soc/fpga-and-soc.

4.5 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.

4.5.1 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.

4.5.2 My Cases

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

4.5.3 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. Visit About Us for sales office listings and corporate contacts.

4.6 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@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.