Libero SoC v11.7 SP2

Release Notes

10/2016

Microsemi
Power Matters.™

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Fax: +1 (949) 215-4996
Email: **sales.support@microsemi.com**
www.microsemi.com

About Microsemi
Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center 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, California, and has approximately 4,800 employees globally. Learn more at **www.microsemi.com**.

51300161-0/10.16

Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

Revision 1.1

Paragraph 1.4.2 – Additional options to disable CoreABC

Revision 1.0

Revision 1.0 is the first publication of this document.

Contents

1 Libero SoC v11.7 SP2 Release Notes

Microsemi Libero® System-on-Chip chip (SoC) design suite offers high productivity with its comprehensive, easy to learn, easy to adopt development tools for designing with Microsemi's power efficient flash FPGAs, SoC FPGAs, and Rad-Tolerant FPGAs. The suite integrates industry standard Synopsys Synplify Pro® synthesis and Mentor Graphics ModelSim ME® simulation with best-in-class constraints management, debug capabilities, and secure production programming support.

Use Libero SoC v11.7 SP2 for designing with Microsemi RTG4 Rad-Tolerant FPGAs, SmartFusion2 and SmartFusion® SoC FPGAs,IGLOO®2, IGLOO, ProASIC3®, and Fusion® FPGA families.

The Libero SoC v11.7 SP2 release includes the following enhancements:

- Production data for timing and power analysis, enabling customers to go to production with RTG4 devices
- Two new die/package combinations for SmartFusion®2 devices and two new die/package combinations for IGLOO®2 devices
- Addition of IGLOO®2 die/package for Simultaneous Switching Noise(SSN) support

To access datasheets, silicon user guides, Tutorials, and Application Notes, visit www.microsemi.com, select the relevant product family and click the **Documentation** tab. Development Kits & Boards are listed in the **Design Resources** tab.

1.1 System Requirements

See the web page System Requirements for more information regarding operating system's support and minimum system requirements. A 64-bit OS is required for designing with SmartFusion2, IGLOO2, and RTG4 devices.

Setup Instructions for Linux OS can be found on the Libero SoC Documents web page.

1.1.1 Operating System Support

Supported

- Windows 7, Windows 8.1.
- RHEL 5* and RHEL 6, CentOS 5* and CentOS 6.
- SuSE 11 SP4 (Libero only; FlashPro Express, SmartDebug, and Job Manager are not supported)

Note: *RHEL 5 and CentOS 5 do not support programming using FlashPro5.

Not Supported

- 32-bit operating system is no longer supported.
- Windows XP is no longer supported.
- Support for the following operating systems will cease in the first half of 2017:
  - Solaris Flexlm license daemon support; Libero SoC is not supported on Solaris.
  - Libero SoC software support for RedHat Enterprise Linux 5 and CentOS 5.

1.2 Download Libero SoC v11.7 SP2

The Libero SoC v11.7 SP2 service pack is an incremental update and must be installed on top of the Libero SoC v11.7 SP1 or Libero SoC v11.7 SP1.1.

Note: If your Libero SoC installation is the v11.7 production release, first update it to Libero SoC v11.7 SP1 or v11.7 SP1.1, and then apply the Libero SoC v11.7 SP2 service pack update.

Note: Installation requires Administrator privileges to the system.

The following links have the steps for downloading the Libero SoC v11.7 SP2 on Windows and Linux operating systems:

- Windows Download
- Linux Download

1.3 Download SoftConsole 3.4/4.0

The Libero SoC v11.7 SP2 is compatible with SoftConsole v3.4 SP1 and SoftConsole v4.0. The following links have the download packages and explains the steps for downloading different versions of SoftConsole on different operating systems:

- Download SoftConsole v4.0 for Windows.
- Download SoftConsole v4.0 for Linux.
- Download SoftConsole v3.4 SP1 for Windows.

1.4 Silicon Feature Support

1.4.1 SmartFusion2/IGLOO2 - Four New Parts

The Libero SoC v11.7 SP2 release introduces two new SmartFusion2 parts and two new IGLOO2 parts.

Family	Die	Pin/Package	Speed Grade	Core Voltage	Temperature
SmartFusion2	M2S010S	144 TQ	-1	1.2 V	TGrade2
	M2S005S	144 TQ	-1	1.2 V	TGrade2
IGLOO2	M2GL010S	144 TQ	-1	1.2 V	TGrade2
	M2GL005S	144 TQ	-1	1.2 V	TGrade2

1.4.2 RTG4 - Dual-Port Large SRAM Feed-Through Write Option Removed

RTG4 no longer supports the Feed-Through Write feature in the large SRAM (LSRAM). See the ER0193: RTG4 FPGA Errata for details. In the Dual-Port Large SRAM Configurator, the **DIN** option for **DOUT on Write** has been removed.

For existing designs migrated to the Libero SoC v11.7 SP2, on opening the project, Libero SoC will check every LSRAM for any Feed-Through Write feature. If the Feed-Through Write feature is detected during the Compile step (for Classic Constraint Flow) or the Synthesis step (for Enhanced Constraint Flow), an error message appears: “Instance XYZ used feed-through mode which is not supported”.

To continue with the design process in the new release, you need to modify the design in one of the following ways:

- Preserve the functionality in the RTL source through the workaround methods described below.
- Change the functionality that requires Feed-Through Write in the RTL source or in the instances of the Dual-Port LSRAM Configurator. See the section on Large SRAM in RTG4 FPGA Fabric User Guide.
    - **Note:** in any CoreABC component, disable the following three options:
        - **Internal Data/Stack Memory**
        - **CALL, RETURN, RETISR**
        - **PUSH, POP**

Modify the RTL source by adding attributes and synthesize. Below are the two use case examples that describe the RTL scenario and the attributes that can be added such that SynplifyPro will not configure the Feed-Through Write mode while inferring RAM1Kx18_RT.

Use case 1:

RTL has common read and write address for memory array. Read address is pipelined (and read data is registered).

Workaround: Add *syn_keep* attribute to the read address pipeline register and read data register in the RTL code.

```
module sp_ram1(clk, we, addr, din, dout);
parameter address_width = 10;
parameter data_width = 9;
parameter ram_depth = 1 << address_width;

input clk, we;
input   [address_width-1:0] addr;
input      [data_width-1:0] din;
output reg [data_width-1:0] dout;

reg        [data_width-1:0] ram[ram_depth-1:0];
reg     [address_width-1:0] addr_reg;
wire    [address_width-1:0] addr_w  /* synthesis syn_keep=1 */;
wire       [data_width-1:0] dout_w  /* synthesis syn_keep=1 */;

assign addr_w = addr_reg;
assign dout_w = din;

always @(posedge clk)
begin
   addr_reg <= addr;
   if(we)
   begin
        ram[addr] <= din;
        dout      <= dout_w;
   end
   else
        dout      <= ram[addr_w];
end
endmodule
```

Resource Usage Report:

```
Total SLE: 10
Block Rams (RAM1K18_RT) : 1
Total LUTs:    9
```

Use case 2:

RTL has common read and write address for memory array. Read data is registered.

Workaround: Add *syn_keep* to read data register and *syn_ramstyle=”rw_check”* to the memory array in the RTL code.

```
module sp_ram2(clk, we, addr, din, dout);
parameter address_width = 10;
parameter data_width = 9;
parameter ram_depth = 1 << address_width;

input clk, we;
```

```
input   [address_width-1:0] addr;
input      [data_width-1:0] din;
output reg [data_width-1:0] dout;

reg        [data_width-1:0] ram[ram_depth-1:0] /* synthesis syn_ramstyle="rw_check" */;
wire       [data_width-1:0] dout_w             /* synthesis syn_keep=1 */;

assign dout_w = din;

always @(posedge clk)
begin
   if(we)
   begin
        ram[addr] <= din;
        dout      <= dout_w;
   end
   else
        dout      <= ram[addr];
end
endmodule
```

Resource Usage Report:

```
Total SLE: 10
Block Rams (RAM1K18_RT): 1
Total LUTs:    9
```

1.4.3 RTG4 - Updated Drive Strength Settings for SSTL18I, SSTL18II, SSTL25I, and SSTL25II

With the updated drive strength settings, the maximum frequency specifications of MSIO and MSIOD I/O banks for SSTL I and SSTL II I/O standards are met at 1.8 V and 2.5 V. Existing designs containing any of the above I/O standards, on migration to the Libero SoC v11.7 SP2 release, will get the new drive strengths when you execute the Compile step (for Classic Constraint Flow) or the Synthesis step (for Enhanced Constraint Flow).

1.5 Software Enhancements

Unless otherwise noted, the Software Enhancements apply to SmartFusion2, IGLOO2, and RTG4 devices.

1.5.1 RTG4 - Updated Timing and Power Data

Timing and Power data for RTG4 are updated and generally improved from preliminary to production quality in the Libero SoC v11.7 SP2 release. You must open any existing design in this release, and regenerate the timing and power reports.

1.5.2 RTG4 - Updated Timing Data for SpaceWire channels

Timing Data for RTG4 SpaceWire channels have been updated in the Libero SoC v11.7 SP2 release. Refer to KB FQ1416 for I/O Delay Adjustment.

SET Mitigation	I/O Bank	Number of Available Channels	Max Frequency (MHz)	Max Data Rate (Mbps)
OFF	MSIO/MSIOD	14	200	400
	DDRIO	2	100	200
ON	MSIO/MSIOD	14	125	250
	DDRIO	2	75	150

1.5.3 RTG4 – Multiple Asynchronous-Reset Placement

In the RTG4 architecture, there are limits on the number of globals and asynchronous-resets that can drive pins which are placed within the same area of the chip. An RT4G150 device contains only

one chip-wide asynchronous-reset network with the GRESET at its root. The same network can be fractured at every half-row and driven from alternative sources like a CLKINT global or an RGRESET local buffer. Each half-row has the following capacity constraints:

- up to one asynchronous-reset signal
- up to 7 or 8 globals other than asynchronous-reset

Asynchronous-reset nets can be more difficult to resolve than other global nets because of tighter routing constraints. To produce a routable placement in this architecture, the placer must ensure that no more than one asynchronous-reset net drives the sinks in each half-row area of the chip. In most cases, the Libero SoC v11.7 SP2 release is able to satisfy these restrictions.

However, in designs with a large number of asynchronous-resets or globals, or tightly-interconnected asynchronous-reset logic, or large fanout, high device utilization which limits the ability to move cells, or pre-existing floorplan constraints, it may not be able to find a solution. The Libero SoC v11.7 SP2 will then print the following detailed message for each violated area:

```
--------------------------------------------------------------------------------
Row from (x1,y1) => (x2,y2), containing the following nets:
    Net Type: R/Greset, Name: net1, Driver: instance1:Y
    Net Type: R/Greset, Name: net2, Driver: instance2:Y
and the following fixed instances which violate constraints but cannot be moved:
    Instance: instance3 (type: type3)
        Input Pin: Aln, Driven by Net: net1
    Instance: instance4 (type: type4)
        Input Pin: Aln, Driven by Net: net2
--------------------------------------------------------------------------------
```

There are several ways to mitigate this problem and to help the Layout tools discover a routable placement:

- Create a region constraint for each net identified (by the Placer) in the Layout log window. The goal of the region constraints is to “manually separate” the logic associated with these nets into distinct areas of the chip. The region constraints should cover unique rows which overlap with as few other asynchronous-reset and global nets as possible.
- Your design may contain some low-to-medium fanout nets which have been placed onto the global network. These nets can increase the global routing congestion. It may be possible to demote these nets to fabric routing resources, and in so doing, alleviate some of the routing congestion for the remaining global nets. Individual global nets can be demoted via a synthesis directive in the RTL, and more general control can be exerted by adjusting the globals demotion options in the **Synthesize Options** dialog box.
- Your design may contain floorplan constraints which fix cells into particular rows or areas of the chip; in this scenario, the fixed cells may be preventing globals from successful routing. Examine any pre-existing floorplan constraints for potential problems.

1.5.4 SmartFusion2 and IGLOO2 - Simulation Library Updated to include Global Buffers

The back-annotation netlist of SmartFusion2 and IGLOO2 designs generated by the Libero SoC v11.7 SP2 release now includes new macros representing global buffers: GB_NG, GBM_NG and RGB_NG. The updated simulation libraries support back-annotated simulation of globals in both gating and non-gating mode. The Macro Library Guide has been updated with descriptions for the following macros: GCLKINT, RGCLKINT, GCLKBUF, GCLKBIBUF, GB_NG, GBM_NG and RGB_NG.

1.5.5 SmartFusion2, IGLOO2, and RTG4 – High Speed Serial Interface Configurator Updated to include CTLE Disabled Option

The **CTLE Disabled** option (available only for XAUI and EPCS protocols) has been added to the High Speed Serial Interface Configurator. This is the default option for CTL Equalization setting. This setting is required when RX_AMP and RX_CUT are both 0x0.

- For existing designs created in the Libero SoC v11.7 release (or subsequent Service Pack releases) and migrated to the Libero SoC v11.7 SP2 release, the previous user-selected settings (Pre-Defined Settings 1 through 14) are preserved and remain valid.
- For existing designs created in Libero SoC software releases prior to 11.7 and migrated to this release:
  - If the register default value is left at 0x0, the Configurator maps it to “CTLE Disabled” when the configurator opens.
  - If the register value is anything other than 0x0, the Configurator maps it to the closest Pre-Defined 1 through 14 setting and displays it in the Configurator.

1.5.6 IGLOO2 - Expansion of Simultaneous Switching Noise Support

Simultaneous Switching Noise (SSN) support is expanded to include IGLOO2 devices, in addition to SmartFusion2 devices supported since the Libero SoC v11.6 release. With this release, SSN supports the following family/die/package combinations.

Family	Die	Package
IGLOO2	M2GL090*	FG676
	M2GL060*	FG676
	M2GL050*	FG896
	M2GL025*	FG484
	M2GL010*	FG484
SmartFusion2	M2S090	FG676
	M2S060	FG676
	M2S050	FG896
	M2S025	FG484
	M2S010	FG484
Note: *New SSN support for Libero v11.7 SP2		

1.5.7 SmartFusion2/IGLOO2 – Warning Message for non-S Device

In the System Builder Security page or the MSS Security Configurator, if page lock is set for eNVM in non-S devices, Libero SoC v11.7 SP2 displays a warning message.

For the M2GL060 and M2S060 dies, the message reads:

“The device you have selected does not support the “Data Security” features. You must not set page lock for eNVM0 for the 060 devices, you can refer to the SmartFusion2 Device Errata and the IGLOO2 Device Errata for more information”.

For the M2GL090/150 and the M2S090/150 dies, the message reads:

“The device you have selected does not support the “Data Security” features, you must not set page lock for eNVM1 for the 090/150 device, you can refer to the SmartFusion2 Device Errata and the IGLOO2 Device Errata for more information”.

1.6 Resolved Issues

Customer Case Number	Description
493642-2144751821	SSN Analyzer needs to be supported for IGLOO2 devices.
493642-2159629608	SERDESIF Configurator using wrong default settings for RX_AMP/RX_CUT
493642-2039633013	Update SB Security settings master/slave tab for non-S device
493642-2115069902	check_sdc_constraints is only for enhanced constraints flow
493642-2104377814	Pin Separator should be changed from ":" to "/" for SDC constraint example
493642-1965192162	"Timer" is still mentioned in 11.6 UG (Designer Constraints)
493642-2134507641	SSN Analyzer Pass Fail criteria need to be mentioned in the Help Doc
493642-2161608255	P&R report missing input .sdc source file information
493642-2039633013	Update SB Security > master/slave tab for S device (005/010/025/050)
493642-2162176960	RTG4 Placer Failure
	Please provide some hints or tips when RTG4 layout fails
	Detect if fixed cells violate bandwidth constraints
493642-2082991226	VHDL encryption envelop example must be corrected
493642-2096643137	GB/RGB models with disabled gating
493642-2128727254	Macro Library UG needs to update all gated-clock macros
493642-2096643137	Some Back Annotated macros are not documented
493642-2163384484	Software IDE not visible in Tool Profiles
n/a	Min delay repair does not fix the hold violation on SpaceWire data pins
n/a	actlmgrd is needed to fix security issue

1.7 Known Limitations, Issues and Workarounds

Unless mentioned in the preceding Resolved Issues list, known issues in the Libero SoC v11.7 SP1.1 also apply to the Libero SoC v11.7 SP2. Review the Libero SoC v11.7 SP1.1 Release Notes for known issues in the Libero v11.7SP1.

1.7.1 RTG4 - Single Event Transient (SET) Mitigation ON may result in Hold Violations

Turning SET Mitigation ON may result in hold time violations in some register to MATH block paths, or between I/O register and register paths. Enable **Repair Minimum Delay Violations** in Place and Route Options to have the Place and Route tool mitigate these and other hold time violations.

1.7.2 RTG4 - Single Event Transient (SET) Mitigation Option Change Does Not Revert Design to Pre-Compile/Pre-Synthesis State

Changing the SET Mitigation option (from OFF to ON or vice versa) does not revert the design to the pre-Compile (for Classic Constraint Flow) or pre-Synthesis (for Enhanced Constraint Flow) state. Make sure that you re-run the Compile or Synthesis step and continue with the design flow.

1.7.3 RTG4 - SmartDebug: Device Resets during JTAG Operations with SmartDebug

After performing one or more JTAG operations, if a user closes and reopens the SmartDebug tool (either standalone or within the Libero SoC software), the device resets itself.

Workaround:

The device reset problem can be avoided using the FlashPro5 programmer and by setting a value of "1" on the def variable **"SMARTDEBUG_RTG4_FLASHPRO5_DISABLE_RESET".**

- For standalone SmartDebug
  - When invoking the tool from the command line, add the following argument:

    Console >

    ./sdebug.exe **SMARTDEBUG_RTG4_FLASHPRO5_DISABLE_RESET**:1
  - When invoking the tool from the GUI:

    Edit the sdebug.def file and change the value of def variable to '1' in the line below:

    data SMARTDEBUG_RTG4_FLASHPRO5_DISABLE_RESET 0 OVERRIDE
- For SmartDebug invoked from Libero:
  - Edit the sdbg.def file and change the value of def variable to "1" in the line below:

    data SMARTDEBUG_RTG4_FLASHPRO5_DISABLE_RESET 0 OVERRIDE
  - Add the following line in the libero.def file:

    data SMARTDEBUG_RTG4_FLASHPRO5_DISABLE_RESET 1 OVERRIDE
- For Tcl script-driven batch mode operation, add the following def variable and the value in a Tcl script:

  defvar_set -name **SMARTDEBUG_RTG4_FLASHPRO5_DISABLE_RESET** -value 1

Note: When this def variable is set to 1, the LiveProbe set in the previous SmartDebug session is not retained when a subsequent SmartDebug session is invoked.

1.7.4 RTG4 - Custom Flow with uPROM: uPROM content must be a single line file

If you use the custom flow and import the uPROM content using the import_component_data command, the uPROM memory file must not have any newlines.

1.7.5 SmartFusion2, IGLOO2 and RTG4 – Place and Route Tool Does Not Support get_nets or get_clocks SDC commands

The placer does not support the get_nets and get_clocks object access commands when used with SDC timing constraints. To ensure that the SDC timing constraints are honored by the placer tool, do not use get_nets or get_clocks in the SDC timing constraints.

Workaround: Use the get_pins command instead.

Example:

The following constraint uses get_nets command:

```
create_generated_clock -name {sys_clk} -divide_by 1 -source [ get_ports { CLK_40M } ] -
phase 0 [ get_nets { u_PLL/my_pll_0/GL0_net } ]
```

Rewrite the constraint to use get_pins command instead:

```
create_generated_clock -name {sys_clk} -divide_by 1 -source [ get_ports { CLK_40M } ] -
phase 0 [ get_pins { u_PLL/my_pll_0/CCC_INST/GL0 } ]
```

1.7.6 SmartFusion2/IGLOO2/RTG4 - Chip Planner Displays Some Unplaced Macros after Layout in Enhanced Constraint Flow

This is a Chip Planner display issue. It can be ignored if layout is successful.

- If the “Repair Minimum Delay Violations” option is enabled in layout options and the layout tool adds buffers to do the repair, opening Chip Planner after layout may display the added buffers as unplaced macros.
- If nets on Row Globals or local asynchronous resets for RTG4 are constrained to a user-created exclusive region in Chip Planner before layout is run, re-opening Chip Planner may display the macros connected to those constrained nets as unplaced macros even though the layout process has successfully completed.

1.7.7 SmartFusion2, IGLOO2, and RTG4 - Enhanced Constraint Flow Limitations

The following tools and flows are not supported in the Enhanced Constraint Flow in the Libero SoC v11.7 SP2 release:

- Precision Synthesis
- I/O Advisor
- Netlist Viewer
- Block Flow
- Design Separation Flow using MSVT

1.7.8 Extra Pop-Up Messages from SynplifyPro

When SynplifyPro synthesis is invoked interactively, SynplifyPro displays a pop-up message about the completion of Tcl script file execution, if any one of the following is true prior to the interactive invocation of SynplifyPro:

- Additional user-specified synthesis options are configured in a Tcl script and passed by Libero to SynplifyPro.
- The Synthesis Option is entered in the Configure Synthesis Option dialog box as a command line entry and passed to SynplifyPro.

These pop-up messages can be ignored. Click **OK** to continue with the SynplifyPro synthesis.

1.7.9 Programming – Programming Recovery Not Working After Programming Interruption

Exporting a SPI bitstream with Programming Recovery enabled with another programming file type (STAPL, DAT) will erase and reprogram the Programming Recovery setting. If a programming interruption occurs before the Programming Recovery setting is reprogrammed with the following programming method (Auto Update, Auto Programming, or IAP/ISP services), then Programming Recovery will not occur.

To workaround this issue, export SPI bitstream only without any other programming file type. This will be resolved in Libero v11.8.

1.7.10 Programming - Libero crashes when Exporting FlashPro Express Job for UEK1 or UEK2 with eNVM

Libero crashes when the Export FlashPro Express Job tool is invoked to generate a programming job encrypted using UEK1/UEK2 and eNVM is the only selected component. To work around this problem, select both fabric and eNVM components for exporting the programming job encrypted using UEK1 or UEK2.

1.7.11 Programming - SPPS Flow: export_hsmtask fails when set_security_overwrite is followed by set_envm_update

If the user Tcl script has the security overwrite command followed by the eNVM update command, the export of HSM job will fail. In other words, if set_security_overwrite is followed by set_envm_update, export_hsmtask fails.

Workaround:

If both the security overwrite command and the eNVM update command are required, make sure that the eNVM update command is executed prior to the security overwrite command. Put the set_envm_update Tcl command before the set_security_overwrite Tcl command in the Tcl script.

1.7.12 Programming - No Programming Support for Virtual Machines

Programming is supported for physical machines only. Programming is not supported on any virtual machine (VM).

1.7.13 Programming - Inspect Device Feature Disabled in FlashPro

The Inspect Device feature is disabled in FlashPro for SmartFusion2/IGLOO2 devices beginning with the Libero 11.7 release. Use standalone SmartDebug instead.

1.7.14 Programming - SmartFusion Encrypted STP File Generation

Generating the encrypted STP files for SmartFusion takes 50 times longer than generating the non-encrypted plain STP.

1.7.15 SoftConsole - Restricts ARM® Cortex®-M3 Debug with Debug Pass Key

SoftConsole does not support this feature.

1.7.16 Documentation - Web-based documentation

Starting with the Libero SoC v11.7 release, most Users Guides for SmartFusion2, IGLOO2, and RTG4 are available on the Microsemi website. Libero and Programming/Debug tools will include links to the website.

If the machine on which you have installed Libero does not have access to the Internet, you (or a site administrator) can download all the Libero SoC v11.7 User Guides from the Libero SoC documentation site.

1.7.17 Documents on Linux: Firefox requirement for Online Help and User Guides

Libero SoC v11.7 SP2 requires the “Firefox” executable to be in your PATH variable on Linux. Alternatively, you can access the Reference Manuals on the Microsemi website, or by clicking **Help > Reference Manuals** in Libero. For the Libero SoC v11.7 SP2 release, the “Web Browser” selection in the Libero Preferences dialog is only used by Online Help and for some user guide links.

1.7.18 Installation

C++ installation error can be ignored. Required files will install successfully.

On some machines, the InstallShield wizard displays a pop-up message stating:

The installation of Microsoft Visual C++ Redistributable Package (x86) appears to have failed. Do you want to continue the installation?

Click **Yes** to complete the installation.

1.7.19 Antivirus Software Interaction

Many antivirus and Host-based Intrusion Prevention System (HIPS) tools flag executables and prevent them from running. To eliminate this problem, users must modify their security settings by adding exceptions for specific executables. This is configured in the antivirus tool. Contact the tool provider for assistance.

Many users are running Libero SoC successfully with no modification to their antivirus software. Symantec, McAfee, Avira, Sophos, and Avast tools have known issues. The combination of operating system, antivirus tool version, and security settings all contribute to the end result. Depending on the environment, the operation of Libero SoC, ModelSim ME, and/or Synplify Pro ME may or may not be affected.

1.7.20 Installation Issue on Linux

After installation of Libero on Linux, the attempt to run the udev_install script for FlashPro setup fails with the following message:

```
% ./udev_install
/bin/sh^M: bad interpreter: No such file or directory
```

Problem:

The script uses Windows CR/LF line termination instead of UNIX/Linux LF only line termination and, hence, is not a valid shell script.

Workaround:

Run the `dos2unix` command on the script to convert CR/LF line termination to LF only line termination:

```
% dos2unix udev_install
%. /udev_install
```

If the `dos2unix` command is not available, install the command first, and then run `dos2unix`, and `udev_install`:

```
% sudo yum install dos2unix
% dos2unix udev_install
%. /udev_install
```