Libero SoC v12.2

Release Notes

12/2019

Microsemi 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

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided "as is, where is" and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.

About Microsemi

Microsemi, a wholly owned subsidiary of Microchip Technology Inc. (Nasdaq: MCHP), 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. Learn more at www.microsemi.com.

51300226-1.3/12.19

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 1.3

Revision 1.3 includes the following changes:

- Updated template boilerplate info
- Added resolved issue in section 3.1 (case # 493642-2655141910)
- Revised section 4.14
- Added known issue in section 4.15.1

Revision 1.2

Revision 1.2 includes the following change:

- Added section 4.11
- Updated section 4.5.4 to include instructions for Linux

Revision 1.1

Revision 1.1 includes the following change:

- Updated RTG4FCCC core version number in section 2.1.3

Revision 1.0

Revision 1.0 was the first publication of this document.

Contents

1 Libero SoC v12.2 Software Release Notes

The Libero® system on chip (SoC) v12.2 unified design suite is Microchip's flagship FPGA software, for designing with Microsemi's latest power efficient flash FPGAs, SoC FPGAs, and rad-tolerant FPGAs. The suite integrates industry standard Synopsys Synplify Pro® synthesis and Mentor Graphics ModelSim® simulation with best-in-class constraints management, debug capabilities, and secure production programming support.

Use Libero SoC v12.2 for designing with Microsemi's RTG4 Rad-Tolerant FPGAs, SmartFusion®2 and IGLOO® 2® SoC FPGAs, and PolarFire FPGAs.

To design with Microsemi's older Flash FPGA families, use Libero SoC v11.9 and subsequent service packs.

To access datasheets, silicon user guides, tutorials, and application notes, visit www.microsemi.com, navigate to the relevant product family page, and click the **Documentation** tab. Development Kits & Boards are listed in the **Design Resources** tab.

Note: Libero SoC v12.2 does not support Classic Constraint Flow. IGLOO2, SmartFusion2 and RTG4 projects using the 'Classic' flow cannot be opened in this release. See Migrating an Existing Project Created with Classic Constraint Flow to Enhanced Constraint Flow for details about how to migrate Classic Constraint Flow projects to the Enhanced Constraint Flow.

1.1 Customer Notification (CN) Support

Libero SoC v12.2 includes changes that address certain important issues.

1.1.1 CN 19009, 18009.7: RTG4 PLL Lock Stability

In some cases, the RTG4 PLL can experience a loss of lock event from which it does not automatically self-recover. In these cases, the assertion of the PLL_ARST_N input is required to regain lock. In Libero SoC v12.2 (and Libero SoC v11.9 SP4), the RTG4 CCC configurator **v2.0.200** introduces an option to **Enable PLL Loss of Lock Auto Reset Logic**. This option is enabled by default whenever the PLL is not bypassed, and will insert an additional fabric logic circuit to monitor the PLL lock signal and issue a reset command to the PLL, if loss of lock is detected. This circuit requires a 50 MHz free-running clock that is readily available from the on-chip RC oscillator. You must instantiate the RCOSC_50MHZ macro and distribute its output through a GLx of any one CCC and eventually connect to the exposed CLK50_MHZ input pin of the CCC using the PLL. The CCC connected to RCOSC_50MHZ could even be the same CCC containing the PLL whose reset logic requires the CLK50_MHZ input. If you do not want to use the on-chip RC oscillator, the CLK50_MHZ input pin of the CCC can be driven by an external free-running oscillator running at 50MHz or slower.

Review Customer Notifications 19009 and 18009.7 for more information.

When the PLL is not bypassed, PLL_ARST_N, PLL_POWERDOWN_N, and READY_VDDPLL signals are always exposed. The auto-reset circuit for the PLL will still function even if you connect the exposed PLL_ARST_N input to a logic-high because the circuit combines this user input with the status of the voted lock. If the PLL_ARST_N, PLL_POWERDOWN_N, and READY_VDDPLL inputs are unused in your design, these additional exposed ports can be tied high.

Note: Refer to UG0586: RTG4 FPGA Clocking Resources User Guide for guidelines on selecting the appropriate PLL Lock Window. This is especially important for applications requiring phase alignment between the CCC input reference clock and the CCC outputs.

1.1.2 CN 19011.2: RTG4 I/O Driven High Momentarily at the Start of Programming

Libero SoC v 12.2 (and Libero SoC v11.9 SP4) eliminates an I/O glitch at the end of programming a blank RTG4 device. See CN19011.2 for details.

When a design that has completed Place and Route in a prior release is first opened in Libero SoC v12.2, there is no design state invalidation. To take advantage of the new behavior, you must use Libero SoC v12.2 to regenerate the bitstream for programming.

1.1.3 SmartFusion2 MSS Watchdog Timer: WD_TIMEOUT port can no longer be exposed to Fabric

Libero SoC v12.2 adds a DRC to the SmartFusion2 MSS Watchdog Timer. Starting with this release, the WD_TIMEOUT port can no longer be exposed to the FPGA Fabric. The option to expose this port is now grayed out in the MSS Watchdog Timer configurator and in the System Builder. If your design uses this port, you must edit your MSS Watchdog Timer configuration or your System Builder configuration as appropriate and disable the port.

1.1.4 SmartFusion2, IGLOO2 RGB Area Coverage Limitation

A certain RGB (row global) configuration in SmartFusion2 and IGLOO2 designs has been disabled as of Libero SoC v12.2 onward. If your design uses such a configuration, you will not be able to generate a programming file using this release. In that case, rerun Place and Route using the Incremental option, and then regenerate the programming file.

1.1.5 PolarFire IOD HS_IO_CLK_PAUSE Signal

The gearboxes within the PolarFire IOD logic include a HS_IO_CLK_PAUSE signal. This signal is found to be sensitive to constraints in Libero SoC place and route iterations. If you are using generic IOD gearboxes in your design, you must download the updated PF_QDR, PF_IOD_GENERIC_RX, and PF_IOD_GENERIC_TX cores, and regenerate your design with the updated cores. These cores are correctly constrained, removing the sensitivity in certain place and route runs. Refer to section 2.1.1 for details on migrating existing designs containing the above cores to this release.
Change Notifications for this item will be issued shortly.

1.1.6 PolarFire Production Timing Changes

In Libero SoC v12.2, minor changes have been made to the timing parameters of sequential elements, and select routing resources, including a tightening of the clock skew. These changes may result in a minor impact on timing margins for existing pre-12.2 designs. Clock performance may vary by an average of -0.5% for -1 speed grade devices, and -4% for STD speed grade devices. Hold violations may vary by +/-7ps, with no significant change anticipated in the number of violating paths.

For any Libero SoC v12.1 or earlier project targeting the MPF100/200/300T (1.0V) device using the STD speed grade, you must regenerate the timing reports at all corners using Libero SoC v12.2. If violations are seen, they must be addressed to ensure proper silicon functionality, and a new bitstream generated.

For Libero SoC v12.1 or earlier projects targeting the -1 speed grade, the impact is within the built in safety margin. Therefore, a thoroughly tested working silicon (over PVT) created using Libero SoC v12.1 will still be working on silicon.

Change Notifications for this item will be issued shortly.

1.2 New Device Support

Libero SoC v12.2 includes the following enhancements for PolarFire devices:

- Production timing and power support for all PolarFire devices. New for this release is Production timing and power support for the following devices:
  - MPF100T/S/TS/TLS (1.05V)
  - MPF200T/S/TS/TLS (1.05V)
  - MPF300T/S/TS/TLS (1.05V)
  - MPF500T/S/TS/TLS (1.05V and 1.0V)

  Note: Refer to section 4.13 for a known issue regarding fabric jitter timing for the SERDES global clocks

- Production data for SSN analysis for all die/package combinations
- Production status for the following cores:
  - Transceiver Enhanced Receiver Management
  - IOD CDR
  - System Services
  - Clock Conditioning Circuitry (PLL post-divider feedback and PLL external feedback features; the remaining features were already in Production status)
- The MPF300T-1FCG484E device is now supported with a Gold license; in previous releases, this device required a Platinum license.

Libero SoC v12.2 includes the following enhancements for RTG4 devices:

- Production timing support for the following devices:
  - RT4G150 352-CQFP (STD and –1)
  - RT4G150L 352-CQFP (STD)

1.3 Design Performance and Runtime Improvements

Libero SoC v12.2 includes the following runtime improvements over the Libero SoC v12.1 release:

- A 20% speedup in simulation runtime with ModelSim ME Pro 2019.02, the OEM Simulation Tool packaged with Libero SoC v12.2

  Note: ModelSim ME Pro 2019.02 is based on ModelSim PE 2019.02, which provides additional runtime optimizations for VHDL designs

- A 10% runtime reduction in Timing Analysis for PolarFire 300T and 500T devices
- A 20% runtime reduction in Timing Analysis for PolarFire 100T and 200T devices
- A 20% runtime reduction in high effort Place and Route for RTG4, SmartFusion2 and IGLOO2 designs larger than 90k logic elements

Libero SoC v12.2 also includes the following design performance improvements over the Libero SoC v12.1 release:

- 5% average Fmax improvement in regular effort Place and Route for PolarFire designs
- 4% average Fmax improvement in high effort Place and Route for PolarFire designs
- 8% average Fmax improvement in regular effort Place and Route for RTG4, SmartFusion2 and IGLOO2 designs larger than 90K logic elements
- 5% average Fmax improvement in high effort Place and Route for RTG4, SmartFusion2 and IGLOO2 designs larger than 90K logic elements

1.4 Software Features and Enhancements

1.4.1 Programming

FlashPro6 Support

Libero SoC v12.2 adds support for the next-generation Microsemi Flash Programming System, FlashPro6. FlashPro6 offers faster Programming time and faster device debugging actions. See the appendix for details.

FlashPro6 currently supports the following platforms:

- **Device families**: SmartFusion2, IGLOO2, RTG4, PolarFire
- **Operating systems**: Windows 7, Windows 10, RedHat and CentOS versions 6.6-6.11; 7.2-7.6
- **Software Tools**: Libero, FlashPro Express, SmartDebug

Note

- RTG4 devices are supported by FlashPro 6; however, they do not fully benefit from the Programming time improvements. This will be addressed in an upcoming Libero SoC release.

Other Enhancements

Libero's Export Bitstream Configuration dialogs have been redesigned for an improved user experience. A similar redesign has been done for the FlashPro Express Job Configuration dialog.

1.4.2 Design Hierarchy and Language Support Enhancements

Libero SoC v12.2 adds support for the VHDL-2008 'context' feature, to enable simultaneous import of multiple required packages. In addition, for new projects created with Libero SoC v12.2 and subsequent releases, System Verilog is the default import language for Verilog files. Projects created with earlier Libero releases will remain unaffected by this change.

The Design Hierarchy view has been enhanced to allow most user actions to be performed when the Design Hierarchy is out of date.

Starting with Libero SoC v12.2, you must explicitly set the top-level module (root) of your project before proceeding with the Design Flow (e.g. Constraints Entry, Simulation, Synthesis, etc.). This is a requirement in both interactive and Tcl flows.

Important Note: Existing Tcl scripts that do not explicitly set a project's root must be edited to add the set_root Tcl command, as appropriate.

1.4.3 Licensing

Libero SoC v12.2 implements the following Licensing enhancements:

- License Selector:
  - When Libero is invoked in Interactive mode, if multiple licenses are available, a dialog with valid license options is displayed, to allow selection of the preferred license for that session of Libero.
  - When Libero is invoked in batch mode, use the –LICENSE_TYPE command line argument to specify a license for that session. This is only necessary if there are multiple licenses available.

    E.g.: ./libero LICENSE_TYPE:Platinum script:<script file name>

    License type options are Platinum, Gold, Silver, and Evaluation

- The License Details dialog (accessible by clicking Help -> License Options) has been enhanced to display the list of IP cores that can be used with the selected license
- Upon invocation of Libero, a warning message notifies users if the selected license expires within 15 days

Refer to the Libero online help for more information.

1.4.4 Tcl Support Enhancements

Libero SoC v12.2 includes the following new Tcl commands:

- configure_core: to modify the configuration of an existing Component
- delete_component: to delete a Component from the project
- SmartDesign Tcl commands: sd_create_bif_net, sd_create_bus_net, sd_create_scalar_net, sd_rename_net, and sd_connect_net_to_pins

For details, refer to the Libero SoC v12.2 Tcl Command Reference Guide or the PolarFire FPGA Tcl Command Reference Guide.

1.4.5 I/O Editor and Chip Planner

Libero SoC v12.2 adds a complete view of a fully placed and routed design in the Chip Planner and I/O Attribute Editor. Note that this mode is for viewing only and changes made in this view will not invalidate the design state. You cannot commit changes in this view, but you can export the constraints to a PDC file on disk. Click the “View” button in the Constraints Manager Tool’s I/O Attributes or Floor Planner tabs to access the viewer.

For details, refer to UG0750: I/O Editor User Guide or UG0821: Chip Planner User Guide.

1.4.6 Text Editor

The Libero SoC v12.2 Text Editor adds numerous features found in modern text editors, such as: change indicators in the margin, cursor line highlighting, autocomplete, bracket matching, find all on select, and auto indentation.

1.4.7 Synthesis

SynplifyPro O201809MSP1-1 is the OEM Synthesis tool integrated with Libero SoC v12.2. This version of SynplifyPro provides key enhancements and bug fixes.

- Enhanced CLKINT (global) inference:
  - SynplifyPro will infer CLKINT instances on generated clock nets.
  - SynplifyPro will infer CLKINT instances per class of pins on a mixed net. The mixed net may drive clock, asynchronous reset/preset or data pin. CLKINT will be inferred on each pin depending on each of the fanout threshold parameters. If the driver of the mixed net is a fabric register, it will be duplicated.
- Ternary arithmetic optimization: SynplifyPro will optimize carry-chains for additions or subtractions in series.
- PolarFire Automatic Compile Point optimization: With this release, when Automatic Compile Point Synthesis is used, the average area usage is within 2%, and the average Fmax performance is within 0.2%, for designs larger than 90k logic elements. SynplifyPro will execute multi-threaded when Compile points are enabled. Design iterations will resynthesize only those Compile points that were modified reducing the synthesis time.

1.4.8 Place and Route

In Libero SoC v12.2, the Place and Route tool has been enhanced to take into account the *set_clock_uncertainty* timing constraint for both max-delay optimization and repair of min-delay violations.

1.4.9 SmartDebug

In this release, SmartDebug adds the following enhancements:

- FlashPro6 hardware support.
- DDC versioning – Starting with this release, when creating a new project using the standalone version of SmartDebug, the DDC file must have been created with the same Libero version. In cases of incompatibility, the DDC file must be regenerated:
  - To regenerate a DDC file using Libero, after running “Generate SmartDebug FPGA Array Data”, run “Export SmartDebug Data”.
- SmartDebug adds half-duplex mode support for PolarFire Transceiver Interfaces.

For details, refer to the SmartDebug User Guide (SmartFusion2, IGLOO2, RTG4) or the SmartDebug User Guide (PolarFire).

1.5 New Silicon Features and Enhancements

1.5.1 PolarFire Transceiver Solution

Libero SoC v12.2 enhances the PolarFire Transceiver Solution to support independent receive and transmit data rates in PMA and PCS-8b10b modes on the same physical lane. This release also adds support for only using a transmit or receive lane, turning off the unused direction to save power.

Each PolarFire Transceiver component has the following options: full duplex, transmit only, receive only, and independent transmit and receive. These options allow for efficiently assigning functionality to each lane and will help customers using small package offerings where transceiver lanes are limited.

Note: Independent receive mode does not support the PolarFire Transceiver Enhanced Receiver Management (ERM) option in the Libero SoC v12.2 release.

1.5.2 PolarFire I/O’s

This release adds the PolarFire I/O Configurator, enabling users to configure and instantiate individual I/Os in their design. The following types of I/O options are available with this configurator:

- **Direction**: Input, Output, Bidirectional, Tribuf
- **Reference Voltage:** Single-ended, Differential
- **Register Mode:** Non-registered, SDR, DDR
- **Dynamic Delay Line**

For details, refer to UG0686: PolarFire FPGA User I/O User Guide.

Libero SoC v12.2 also includes the following enhancements targeting PolarFire I/O support:

- **Manual I/O Register Combining:** Libero SoC v12.2 adds support for manually specifying I/Os to be combined with registers, using an NDC command.

  For details, refer to UG0715: PolarFire PDC Commands User Guide.

- Libero SoC v12.2 enables PolarFire users to create an internal LVDS fail-safe solution. This configuration uses a combination of the following new features:

- Dynamic on-die-termination (ODT) access per I/O
- Weak pullup/pulldown Resistor for differential inputs

For details, refer to UG0686: PolarFire FPGA User I/O User Guide.

1.5.3 PolarFire Clock Configuration Circuitry (CCC)

The following enhancements have been made in this release to the CCC Configurator:

- The actual PLL bandwidth is now reported in the CCC UI, next to the bandwidth selection
- The "Power/Jitter" options have been simplified; you can choose to maximize or minimize VCO to trade-off between jitter and power consumption.

1.5.4 PolarFire Memory Initialization

Libero SoC v12.2’s Design Initialization Data and Memories configuration tool fixes an issue with initialization of LSRAMs configured with ECC. The ECC bits are now correctly set for LSRAMs initialized at powerup.

1.5.5 RTG4 PLL Lock Stability

In some cases, the RTG4 PLL can experience a loss of lock event from which it does not automatically self-recover. In these cases, the assertion of the PLL_ARST_N input is required to regain lock. In Libero SoC v12.2 (and Libero SoC v11.9 SP4), the RTG4 CCC configurator **v2.0.201** introduces an option to **Enable PLL Loss of Lock Auto Reset Logic**. This option is enabled by default whenever the PLL is not bypassed, and will insert an additional fabric logic circuit to monitor the PLL lock signal and issue a reset command to the PLL, if loss of lock is detected. This circuit requires a 50 MHz free-running clock that is readily available from the on-chip RC oscillator. You must instantiate the RCOSC_50MHZ macro and distribute its output through a GLx of any one CCC and eventually connect to the exposed CLK50_MHZ input pin of the CCC using the PLL. The CCC connected to RCOSC_50MHZ could even be the same CCC containing the PLL whose reset logic requires the CLK50_MHZ input. If you do not want to use the on-chip RC oscillator, the CLK50_MHZ input pin of the CCC can be driven by an external free-running oscillator running at 50MHz or slower.

Review Customer Notifications 19009 and 18009.7 for more information.

When the PLL is not bypassed, PLL_ARST_N, PLL_POWERDOWN_N, and READY_VDDPLL signals are always exposed. The auto-reset circuit for the PLL will still function even if you connect the exposed PLL_ARST_N input to a logic-high because the circuit combines this user input with the status of the voted lock. If the PLL_ARST_N, PLL_POWERDOWN_N, and READY_VDDPLL inputs are unused in your design, these additional exposed ports can be tied high.

Note: Refer to UG0586: RTG4 FPGA Clocking Resources User Guide for guidelines on selecting the appropriate PLL Lock Window. This is especially important for applications requiring phase alignment between the CCC input reference clock and the CCC outputs.

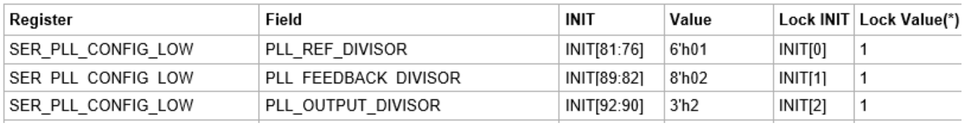
1.5.6 RTG4 I/O Driven High Momentarily at the Start of Programming

Libero SoC v 12.2 (and Libero SoC v11.9 SP4) eliminates an I/O glitch at the end of programming a blank RTG4 device. See CN19011.2 for details.

When a design that has completed Place and Route in a prior release is first opened in Libero SoC v12.2, there is no design state invalidation. To take advantage of the new behavior, you must use Libero SoC v12.2 to regenerate the bitstream for programming.

1.5.7 RTG4 Configuration Report

Libero SoC v12.2 will generate a Configuration report for SERDES, FDDR and CCC blocks present in RTG4 designs. Run the 'Generate FPGA Array Data' step in the tool flow and then invoke the 'View Report' option to see the configured values for various registers. This report shows the value of the register fields of each block and their mapping to its INIT parameter.



Register	Field	INIT	Value	Lock INIT	Lock Value(*)
SER_PLL_CONFIG_LOW	PLL_REF_DIVISOR	INIT[81:76]	6'h01	INIT[0]	1
SER PLL CONFIG LOW	PLL FEEDBACK DIVISOR	INIT[89:82]	8'h02	INIT[1]	1
SER_PLL_CONFIG_LOW	PLL_OUTPUT_DIVISOR	INIT[92:90]	3'h2	INIT[2]	1

1.5.8 RTG4 New Cores

New for this release is the RTG4 SRAM (AHBLite and AXI) core, available in the Libero Catalog. This core offers AHBLite and AXI interfaces to RTG4 LSRAM and uRAM blocks.

2 Migrating Designs to Libero SoC v12.2

2.1 Notes on Design Migration

2.1.1 Core Invalidation

Designs containing the following cores will be invalidated upon migrating a Libero SoC v12.1 project to Libero SoC v12.2:

- PolarFire IOD Generic Receive Interfaces
- PolarFire IOD Generic Transmit Interfaces
- PolarFire QDR

For the above cores, you must do the following:

1. Download the latest version of the core into your vault.
2. Upgrade each configured core in your design to the latest version by right-clicking on the core component in the design hierarchy and selecting ‘Replace Component Version…’.
3. Regenerate the design.
4. Rerun the Derive Constraints step.
5. Rerun the tool flow.

2.1.2 Core Upgrade

If a Libero SoC v12.1 project contains the following cores, and the cores have been generated, they do not need to be upgraded upon migrating the project to Libero SoC v12.2. However, if the core needs to be generated again for any reason (for example, a change in parameters), the latest version from the Catalog must be downloaded and used.

- PolarFire CORESMARTBERT
- PolarFire IOD CDR
- PolarFire IOD CDR CCC
- PolarFire Tamper
- PolarFire SRAM (AHBLite and AXI)
- PolarFire CCC[1]
- PolarFire DDR3
- PolarFire DDR4
- PolarFire LPDDR3
- PolarFire UPROM
- PolarFire Transceiver Interface
- PolarFire PCI Express Interfaces
- PolarFire RGMII to GMII
- PolarFire System services
- RTG4UPROM

[1]When the PolarFire CCC is upgraded to the latest version in GUI mode, the obsolete LOW_POWER option is automatically changed to use the MIN_VCO option. However, if the LOW_POWER option is used in Tcl core creation/configuration, the core generation will fail. Note that the migration may create an invalid configuration because the output frequency DRC is different for LOW_POWER and MIN_VCO. Review the configuration of the CCC for correctness after upgrading to the latest version.

For the above cores, if you want to change their configuration, you must do the following:

1. Download the latest version of the core into your vault.

2. Upgrade each configured core in your design to the latest version by right-clicking on the core component in the design hierarchy and selecting ‘Replace Component Version…’.
3. Regenerate the design.
4. Rerun the Derive Constraints step.
5. Rerun the tool flow.

2.1.3 RTG4 CCC Updates

There have been functional updates to the RTG4 CCC core to address CN19009 (RTG4 PLL Lock Stability).

Existing projects created in prior releases will be invalidated when the project is first opened in Libero SoC v12.2, if a CCC with PLL is instantiated in the design. Upon opening the project, a pop-up window alerts you to the invalidation. The design state reverts to the pre-synthesis state. Download the latest RTG4FCCC core version (**v2.0.200** or later) and replace the CCC component by right-clicking it in the Design Hierarchy pane and selecting “Replace Component Version”. Make new port connections as necessary for any additional ports added to the component and regenerate the design. To continue with the design flow, rerun Synthesis. Designs which continue to use an older RTG4 FCCC core will fail to synthesize until the FCCC instance is migrated to v2.0.200 or later.

Note the following changes:

- The PLL Options Tab, Miscellaneous settings includes a new setting called “Enable PLL Loss of Lock Auto-Reset Logic”. This setting is enabled by default. Before unchecking this option, designers are encouraged to review CN19009 and CN18009.7 for additional details.
- When the PLL is not bypassed, and the PLL Loss of Lock Auto-Reset Logic is enabled, the PLL_ARST_N, PLL_POWERDOWN_N, and READY_VDDPLL signals are always exposed to provide additional user access to these signals. They can be tied to logic high if they are not used in the design.
- A new INFO bubble tooltip has been added on the PLL options tab to direct the user to the RTG4 FPGA Clocking Resources User’s Guide for more information on selecting the minimum PLL Lock Window setting for a given PLL configuration.

2.2 PolarFire Production Cores

Display Name	Libero SoC v12.2 version	Changes from Libero SoC v12.1
PolarFire Clock Conditioning Circuitry (CCC)	2.1.104	• Post-divider and external feedback modes ○ Moved to production status ○ Added missing generated constraints for post-divider and external feedback modes ○ DRC - Phase Shift is not allowed in Post-Divider and External feedback modes ○ DRC - post-divider and external feedback modes are not supported for ES and XT devices • Enhancements ○ Simplified the Power/Jitter options to have just 2 options: ▪ Maximize VCO for Lowest Jitter ▪ Minimize VCO for Lowest Power ○ Display the values of VCO and actual bandwidth computed

		• Misc. ○ PLL_POWERDOWN_N is exposed by default. It is optional for post-VCO mode, and is required for post-divider and external feedback modes.
PolarFire Clock Divider	1.0.103	None
PolarFire CoreSmartBERT	2.5.101	• No functional bug fix • Repackaged with latest PF_XCVR
PolarFire Crypto	1.0.106	None
PolarFire DDR3	2.4.107	• No functional bug fix • Repackaged with latest PF_CCC
PolarFire DDR4	2.4.107	• No functional bug fix • Repackaged with latest PF_CCC
PolarFire LPDDR3	2.3.107	• No functional bug fix • Repackaged with latest PF_CCC
PolarFire QDR	1.6.103	• No functional change • Fixed BFM for one use case (simulation flow) (107034) • HS_IO_CLK_PAUSE management • Add synchronization logic • Repackaged with latest PF_CCC
PolarFire Glitchless clock mux	1.0.101	None
PolarFire PCI Express	2.0.103	No functional changes X1 mode: removed extra unused TX/RX pads, turned off unused lane for saving power
PolarFire Dual Port Large SRAM	1.1.110	None
PolarFire Micro SRAM	1.1.107	None
PolarFire Two Port SRAM	1.1.108	None
PolarFire uPROM	1.0.109	• No functional bug fix • Added proper validator for Tcl configuration flow (98769)
PolarFire Dynamic Reconfiguration Interface	1.0.101	None
PolarFire I/O	1.0.103	New core • Single I/O basic configurator with Dynamic delay line support for normal I/O and DDR_IN/OUT, SDR_IN/OUT
PolarFire IOD Generic Receive Interfaces	1.4.105	• HS_IO_CLK_PAUSE fix in support of CN • Added 'Enable ODT_EN for LVDS failsafe' option

PolarFire IOD Generic Transmit Interfaces	1.3.103	HS_IO_CLK_PAUSE fix in support of CN
PolarFire IOD CDR Clocking	2.1.103	• No functional change • Repackaged with latest PF_CCC
PolarFire Initialization Minitors	2.0.103	None
PolarFire RC Oscillator	1.0.102	None
PolarFire RGMII to GMII	1.2.105	• No functional bug fix • Repackaged with latest PF_IOD_GENERIC_RX/TX
PolarFire SRAM (AHBLite and AXI)	1.2.100	• No functional bug fix • Added pipeline option
PolarFire Tamper	1.0.200	• No functional change • Added info in log file on generation to warn users about using RESET_DEVICE port correctly
PF Temperature and Voltage Sensor Interface	1.0.106	None
PF Transceiver Reference Clock	1.0.103	None
PolarFire Transmit PLL	2.0.006	No functional bug fix Updated for compatibility with Libero SoC v12.2
PolarFire Transceiver Interface(ERM)	3.0.100	• No functional bug fix • Added support for independent receive and transmit data rates • Moved to production status

3 Resolved Issues

The following table lists the customer-reported SARs resolved in Libero SoC v11.9 SP4 and Libero SoC v12.2. Resolution of previously reported “Known Issues and Limitations” is also noted in this table.

3.1 List of Resolved Issues

Case Number	Description
	In Libero SoC v12.1, when you execute generate_component command after creating and configuring System Builder cores (using create_and_configure_core tcl command) such as PF_DDR3, PF_DDR4 and PF_QDR with no parameters specified, Libero tool crashes. Libero SoC v12.2 provides implementation support for generate_component tcl command when a System Builder core is created and configured without specifying any parameters.
493642-2164708663, 493642-2220598517, 493642-2357498226	RTG4 uPROM addressing user prog space as warnings/errors/ - using ModelSim simulation
493642-2582937640	RTG4 LSRAM: Libero Log should report in ECC, non-pipelined mode
493642-2544111080	RTG4: IO glitch during programming
493642-2575742710	RTG4 CQFP 352: Pop up of unrecognized package while opening SynplifyPro interactively
493642-2551173245	RTG4 PLL LOCK Delay: not supported in simulation
	RTG4 CCC: Extra ports exposed when PLL is bypassed
	RTG4 FDDR w/INIT - disable verbose sim
	Mismatch between back annotated netlist and SDF bit order for carry chains
	RTG4 PCIe RTL simulation fails to access config space after link up
	SmartFusion2 SmartDebug: DPK not working when permanent settings are enabled
493642-2626789047	HDL_LANGUAGE: DH: VHDL errors with array size and constants
493642-2616150689	Remove references related to Cycle Accurate Power Analysis
493642-2626283862	[System Builder] Libero SoC v12.0/v12.1 crashes when MDDR parameters are exported
493642-2618088733	SmartTime within Libero SoC crashes with user sets
493642-2617347842	Issue in “select_libero_design_device” TCL command
493642-2183802277, 493642-2569124726	Linking to EDN source file not working when .edn file is selected.
493642-2469698407	Synthesis and Compile issue with PolarFire v2.1
493642-2605797205	Synthesis warnings automatically appear in the Find box after pressing "Ctrl + F"
493642-2616732783	Synthesis fails for VHDL loop statement
493642-2626477753	Placer fails immediately after clustering – IGLOO2 device in Libero SoC v12.1
493642-2582927771, 493642-2582900623	Place and Route needs to interpret the set_clock_uncertainty timing constraint for both max-delay optimization and repair of min-delay violations
493642-2340221200	PF_IOD: Add IO reg combining option for PolarFire devices

493642-2605634534	100T/200T with split HS_IO_CLK
493642-2600504382	Enhancement Request: FlashPro Express support for SPI-Flash Programming
493642-2599876010	SERDES EPCS mode is not working in Libero SoC v12.1; it works in Libero SoC v11.9
493642-2613651696	[Libero SoC v12.1] SmartFusion2 MSS System Builder, Peripheral Configurator fails to open
493642-2498929715	PF_IOD_GENERIC: PolarFire IOD Generic RX: SDR is not supported
493642-2604159892	PF_IOD: Issue with IOD clocking
493642-2544434392	RTG4_CCC: Creation of unnecessary CDC in Libero SoC v11.9 due to derived constraints for CCC
493642-2549712113	PF_RAMs: PF_SRAM_AHBL_AXI: PolarFire SRAM w/ ECC in Bus integrated LSRAM
493642-2599170506	RTG4DPSRAM: Initialize for Simulation option blank for DPLSRAM cores
493642-2606768400	Incorrect and inconsistent SLACK values in SmartTime Libero SoC v12.1
493642-2419814392	System Builder error when reconfiguring the peripherals of System Builder for the first time
493642-2609321998, 493642-2626283862	[System Builder] Libero SoC v12.0/v12.1 crashes while exporting configuration
493642-2613651696	[Libero SoC v12.1] MSS System Builder, Peripheral Configurator fails to open
493642-2569124718	Cannot kill g4layout with Ctrl + C
493642-2586875474	HDL_LANGUAGE: DH: VHDL_CONFIGURATION: Libero SoC v12.0 Synthesis issue
493642-2582927771, 493642-2582900623	RTG4 Timing violation in Libero SoC v12.0
493642-2588964932	FlashPro 5: programming support on CentOS v7.6 (and potentially RHEL 7.6)
493642-2577057196	RTG4: DDR configuration report
493642-2608841172	PF_CCC: PLL_POWERDOWN_N port option in GUI is broken
493642-2606768400	USER SET appendage in Libero SoC v12.0
493642-2602517383	Libero SoC v12.0 does not show Parallel Paths but Libero SoC v11.9 shows for the same design
493642-2616303798	[Libero SoC v12.1] Tao memory usage is very high
493642-2616732783	[Libero SoC v12.1] Synthesis fails for VHDL loop statement
493642-2573306385	PolarFire: During bitstream export EOB digest component shows all zeros
493642-2610533035	PF_TAMPER: Add information about potential issues using RESET_DEVICE
493642-2626789047	HDL_LANGUAGE: DH: VHDL errors with array size and constants
493642-2617203433	PolarFire NGMUX design: Passing in PolarFire v2.3 but fails in Libero SoC v12.0/v12.1
493642-2584895653	Active-HDL run.do file does not generate 2008 compile option
493642-2603443072	Cannot enter negative value for OUTPUT DELAY in Constraints Editor
493642-2586875474	HDL_LANGUAGE: VHDL_CONFIG: SYNOPSYS: Libero SoC v12.0 Synthesis Issue
493642-2560555254	HDL_LANGUAGE: DH: DUPLICATE: Libero does not use RTL from the updated core but instead uses an older version of the core
493642-2589714643	FlashPro5 Tck mode text should be replaced from free running to Discrete
493642-2592752842	Enhancement Request: error message should be clear for EDN flow
493642-1533321021	When double-clicking on an existing constraint in the Constraints Editor GUI, it will not open the constraint for editing as expected
493642-2586875474	HDL_LANGUAGE: NLV: VHDL_CONFIGURATION: Libero SoC v12.0 NLV issue for VHDL configuration
493642-2513649526	CCCB: SDC constraints ignored in SDC file that has quotation mark (“) in the commented line

493642-1850185640	SmartTime GUI: Timing Report Options - Drop the check box for 'Edit generated XML file name'
493642-2575264464	No bars in Slack Distribution window in SmartTime
493642-2576408074	Chip Planner Crossprobing issue (Libero SoC v12.0)
493642-2162176960	ECF: Post-Layout Design shows unplaced instances due to region constraints
493642-2605634534	100T/200T with split HS_IO_CLK
493642-2636540415	Libero crashing when run running TCL generated from RTG4 DDR Init
493642-2618497017	PF_QDR: PolarFire QDR simulation issue with 18bit and burst of 2 configuration
493642-2632922115	Timing fails for design in Libero SoC v12.1 but not in Libero SoC v11.9 SP4
493642-2463252686	Add I/O registers to RTG4 Macro Library Guide
493642-2552029001	Functional failure: state machine is not changing its state on the edge detect trigger event
493642-2634547986	Libero SoC v12.1 crash while generating back-annotated files
493642-2631662721	List of parameters for import_files TCL command
493642-2614132550	Enhancement Request: Option for Still Air in PolarFire Power Estimator with Heat Sink enabled
493642-2655141910	Sub level block cannot be assigned to regions
	PF_CCC: External Feedback: MPF: STA of PLL in External FB Mode
	PF_SRAM_AHBL_AXI: Instances with AXI interfaces are getting pruned by Synthesis tool
	SERDES_IF v1.2.210 crashes while exporting Register configurations in Libero SoC v12.1
	Libero SoC v12.0 and v12.1 crashes when trying to open configuration option in Generate Back Annotated Files in Design Flow
	Change the warning message displayed while adding SmartBERT Core to Libero SoC v12.0 Catalog
	ECF: Need an option to update the constraint with the IO input delay value used by layout

4 Known Issues and Limitations

4.1 Catalog Cores

4.1.1 Core Generation Language

In Libero SoC v12.2, the PolarFire cores listed in section 2.2 generate only Verilog files, regardless of the preferred HDL language selected in the Libero project.

VHDL users desiring to simulate designs containing affected cores must use mixed-language simulation (available with ModelSim ME Pro, which is bundled with this release, and requires a Gold, Platinum, or Eval license).

4.1.2 Linux: Core generation fails in batch mode when the DISPLAY variable is not set

The following Direct Cores cannot be generated in Libero in batch mode via Tcl when the DISPLAY variable is not set on a Linux machine:

- CoreAXI4SRAM
- CoreCIC
- CoreCordic
- CoreABC
- CoreEDAC
- CoreDDS
- CoreFIFO
- CoreFFT
- CoreFIR_PF
- CoreRSDEC
- CoreRSENC

4.1.3 Core version upgrade

While upgrading the core version for components created for PolarFire System Builder cores from an older version to the latest, it is required that both the core versions are downloaded to the vault. Core version upgrade fails otherwise.

4.2 Design Hierarchy: HDL language duplicate modules

- Opening an existing project does not show Design Hierarchy correctly when the design has duplicates between a core module and an HDL module.

  Workaround: Build the Design Hierarchy after opening the project.

- If a design has duplicates between the elaborated modules of the core and a normal HDL module, they are not shown as duplicate modules in the Design Hierarchy.
- If two different VHDL files have same signature (same inputs, outputs, and architecture), they are not detected as duplicate modules in the Design Hierarchy.

4.3 Tcl Support Limitations

Parameters for SgCore and System Builder components are not documented. To configure such cores using Tcl, do the following:

1. Use the GUI to configure the core as desired.
2. Export the core configuration Tcl description by selecting the "Export Component Description(Tcl)" action on the right-click menu of the component in the Design Hierarchy.
3. Use the exported Tcl command to create the configured core in a regular Tcl script.

Note: The following set of cores cannot be configured using Tcl; the Export Component Description (Tcl) option is thus not supported:

- SmartFusion2/IGLOO2 MSS/HPMS component
- SmartFusion2/IGLOO2 System Builder component
- RTG4 DDR memory controller with initialization (RTG4FDDRC_INIT)
- RTG4 High Speed Serial Interface 2 – EPCS and XAUI – with Initialization (NPSS_SERDES_IF_INIT)
- RTG4 High Speed Serial Interface 1 – EPCS and XAUI – with Initialization (PCIE_SERDES_IF_INIT)

4.4 SmartDesign

4.4.1 Modify Memory Map feature should not be used

The Modify Memory Map action used to connect peripherals to buses in the SmartDesign canvas should not be used in the Libero SoC v12.2 release. If used, Libero may crash or produce an incorrect/incomplete memory map. Connect peripherals to bus slave positions manually, as per the desired memory map.

4.4.2 Export Component Description (Tcl)

In some cases, when the Export Component Description (Tcl) command is executed on a SmartDesign, pin groups created by Libero are converted to Tcl and the exported Tcl script errors out when executed.

Workaround:

1. Delete the converted line(s) from the exported file.

2. Delete the created SmartDesign.

3. Re-execute the Tcl script.

4.5 Synthesis

4.5.1 MPF500T/TS/TL/TLS: encrypted blocks are limited to one top level module

To avoid Synthesis failures for Libero projects targeting the MPF500T device, ensure that each encrypted block in the design has exactly one top module. This issue also affects designs containing the Cortex-M1 IP core.

4.5.2 SynplifyPro mapping of sequential-shift to uSRAM does not support initial values

PolarFire devices do not support initial values on registers for Sequential-shift constructs mapped to uSRAMs. If an initial value is specified for a register in RTL, Synplify will ignore the value and issue a warning.

4.5.3 SynplifyPro version checking returns an error message on RHEL/CentOS 7.4

Checking the SynplifyPro version with the following command returns an error message: synplify_pro -version -batch

Error creating '"Internal Error: unsupported format used in message: ' Error creating '"Internal Error: unsupported format used in message: ' N-2017.09M-SP1-1

Note: In Libero SoC v12.2, the dialog box where a Synthesis profile is added will display the same error message.

This error message can be safely ignored – these operating systems are supported by Libero SoC v12.2.

4.5.4 Standalone Synthesis Flow

Libero SoC v12.2 users may synthesize their design outside the Libero SoC software by using Synopsys SynplifyPro directly. When using this flow, the following additional steps are necessary to successfully synthesize and implement a design:

- For Windows, ensure that the <install location>/Designer/data/aPA5M/polarfire_syn_comps.v is added as a source file to the SynplifyPro project. This file contains module declarations with timing information for PolarFire primitives not known to Synopsys.
- For Linux, ensure that the <install location>/Libero/data/aPA5M/polarfire_syn_comps.v is added as a source file to the SynplifyPro project. This file contains module declarations with timing information for PolarFire primitives not known to Synopsys.
- Many configured cores generate timing constraints. You must ensure that these constraint files are passed to synthesis for optimal results. These constraint files must also be imported into Libero along with the synthesis gate level netlist to get optimal place and route and timing analysis results. Core-generated constraint files must be modified so that constraints are expressed using the proper hierarchical name of the configured cores in the top-level design. Refer to the Custom Flow User Guide for more information.

4.6 I/O Editor

4.6.1 DRC validation

The DRC check in the I/O Editor does not validate all the constraints set in the tool. You must run Place and Route to validate these constraints.

4.6.2 XCVR SI Parameters in Placement View

In the I/O Editor XCVR View, clicking on a placed XCVR lane in the Placement view does not show any Signal Integrity parameters corresponding to that lane in the Signal Integrity view.

Workaround: Click on the XCVR lane instance in the schematic view to select the lane and view the parameters corresponding to that lane in the Signal Integrity view.

4.7 Netlist Viewer

Opening two or more views (for example, Hierarchical RTL View and Flattened Post-Compile View) in Netlist Viewer may result in a crash due to memory usage. Avoid opening multiple views for large designs.

4.8 PolarFire Block Flow

Libero SoC v12.2 supports Block Flow. The limitation is that only Fabric components (LUT, SLE, RAM, MATH) may be instantiated in a Block. All other components (CCC, DDR, and so forth) must be part of the top-level design and cannot be instantiated in Blocks.

4.9 SmartTime

4.9.1 Incorrect slack for edge-shifted generated clocks

When an edge shift is specified on a generated clock, common clock period calculation for inter-clock domains with that clock may fail. In such cases, slack calculated for these inter-clock domains will be incorrect in the Max Timing Analysis Report.

4.10 IBIS Models

For all PolarFire MPF300T/TS/TL/TLS -FCG1152 devices, SSN Analyzer simulated data deviates from the Silicon measured data. This deviation can be between 20%-60%.

4.11 Initialize RAM at Power-up

- Unable to open memory file error in Fabric RAMs tab

  If the “Initialize RAM at Power-up” option is selected without specifying the memory file, the Fabric RAMs tab in the Configure Design Initialization Data and Memories and the Memory Initialization tool will error out as “Unable to open memory file.”

  Workaround:

  If you intend to initialize the memory file with zero’s (and therefore did not specify memory file option in the configurator), you must edit the corresponding client and select the initialization option “Content Filled with 0s” in the Fabric RAMs tab.

  If you intend to specify a memory file, you must select the “Content from file” option and specify a memory file.

4.12 Post-layout simulation is not supported for PolarFire

Post-layout simulation is not supported for PolarFire devices in the Libero SoC v12.2 release.

4.13 PolarFire Silicon Support Limitations

4.13.1 PLL

- Bypass option on output clocks is not available in this release.
- PLL External feedback mode limitations:
  - The PLL Lock does not assert in Post Synthesis Simulation.

4.13.1 PCIe

- During BFM simulation of the PCIe AXI interface (master or slave), the simulator may print warning messages about AHB signals, such as “HRESP”. The warning message can be ignored.

4.13.2 Transceiver Reference Clock

- Enabling on-die-termination and external VREF on the Transceiver Interface Reference Clock I/O is not supported in the I/O editor. However, these options can be set in the I/O PDC file.

  Refer to UG0715: PDC Commands User Guide (PolarFire) for more information.
- The connection from the Transceiver Interface Reference Clock I/O to the South-East PLL for all the reference clocks associated with Transceiver Interface Quad 0,2 and 4 lanes is not available in the software. Place and Route will fail if this connection is attempted.

4.13.3 Transceiver Interface

Post-synthesis simulations are not supported for the Transceiver Interface.

4.13.4 I/O's: SSTL15 On-Die Termination values are incorrectly programmed

For the MPF300XT/TES/TSES devices, when the ODT value for an SSTL15 I/O is selected as 20 Ohm or 30 Ohm, an incorrect setting is programmed

Workaround: Do not use the 20 or 30 Ohm on-die termination values for the affected devices.

4.13.5 IOD Receive Interface Data rate issue

In the PF_IOD_RX configurator, duplicate Data rate DRC check messages appear for the presets RX_DDRX_B_G_FA_HSIO, RX_DDRX_B_G_DYN_HSIO and RX_DDRX_B_R_DYN_HSIO. Ignore the second DRC message. The correct Data rates for the presets are 700, 1600 and 500, respectively.

4.13.6 ERM is not available for MPF300XT or ES devices

The XCVR_ERM core must not be used with the MPF300XT or MPF300T/TS_ES devices. This restriction is not currently enforced by the Libero software.

4.14 PolarFire XCVR sourced fabric clocks and jitter compensation

The PolarFire XCVR can source three different clocks into the fabric, TX_CLK, RX_CLK, and the REFCLK (FAB_REF_CLK). These clocks will contain high frequency jitter that is not reported by Libero in the timing report and SmartTime. It is recommended that users add clock uncertainty constraints to these clocks in their design.

The following is a list of recommended values for clock uncertainty per clock, resource, and speed-grade.

- FAB_REF_CLK on Global: 275ps for STD, 200ps for –1
- FAB_REF_CLK on Regional: Not supported
- TX_CLK_G on Global: 300ps for STD, 225ps for –1
- TX_CLK_R on Regional: 225ps for STD, 150ps for –1
- RX_CLK_G on Global: 325ps for STD, 250ps for –1
- RX_CLK_R on Regional: 250ps for STD, 175ps for –1

Below is an example clock uncertainty constraint. This constraint would be added to the user's timing SDC file.

```
# TX_CLK and RX_CLK on Regionals
set_clock_uncertainty -setup 0.150 [ get_pins { PF_XCVR_ERM_LANE2/I_XCVR/LANE0/TX_CLK_R } ]
set_clock_uncertainty -setup 0.175 [ get_pins { PF_XCVR_ERM_LANE2/I_XCVR/LANE0/RX_CLK_R } ]
```

```
# TX_CLK and RX_CLK on Globals
set_clock_uncertainty -setup 0.300 [ get_pins { PF_XCVR_ERM_LANE2/I_XCVR/LANE0/TX_CLK_G } ]
set_clock_uncertainty -setup 0.325 [ get_pins { PF_XCVR_ERM_LANE2/I_XCVR/LANE0/RX_CLK_G } ]

# FAB_REF_CLK on Global
set_clock_uncertainty -setup 0.275 [get_clocks PF_DDR4_C0_0/CCC_0/pll_inst_0/OUT1]
```

These constraints will be automatically added to the derived constraints from the Constraints Manager in a future release of Libero.

4.15 Programming

4.15.1 Libero Programming

- Updating the security or sNVM with a security-only bitstream or sNVM-only bitstream on a device that has the Fabric programmed will disable the Fabric.

  If the Fabric has been disabled, you must reprogram the Fabric to enable it.

  Workaround:

  a. sNVM only bitstreams: Field-update bitstream files should always program the Fabric with sNVM.
  b. Security only bitstreams: Security-only bitstream should be used on a blank device only.
- When a device is programmed with a blank Silicon Signature field, it will not get erased.

  Workaround:

  a. Specify a Silicon Signature that is not blank and program the device to change the value.
  b. Perform the Erase program action to erase it.
- A SPI file that contains a Silicon Signature setting (set in Configure Programming Options) cannot be imported as a SPI bitstream file for a Recovery/Golden client (in the SPI Flash configurator).

  Workaround: Use Libero SoC v12.0 software.
- If the USERCODE that is part of the security segment is unspecified and the security is not programmed, the previous value of USERCODE will be retained.
- Device Programming using Libero SoC v12.2 via SPI-Slave instead of JTAG is currently not supported. Support for this use model will be added in a future release.
- FlashPro Express does not support SPI-Flash only programming in a JTAG chain with SmartFusion2/IGLOO2 and PolarFire devices.
- Serialization is not working for SmartFusion2 and IGLOO2 in Libero SoC v12.2.

  Serialization of the eNVM client is not working for Libero SoC v12.2.

  Workaround: Use Libero SoC v11.9.
- SVF file format for SmartFusion2 and IGLOO2 devices is not supported in the Libero SoC v12.2 release.

  SVF will be supported in the upcoming Libero SoC v12.3 release.
- Generate Bitstream fails after running Export Job with SPI Flash only.

  Workaround: Go to Export Job, select "Program Design" and export the job. Generate Bitstream works as expected.

4.15.2 SPI Flash Programming

This release includes the following limitations:

- Only the Micron SPI Flash is currently supported with the Evaluation Kit.
- N25Q00AA and MT25QL01G are the only SPI-Flash devices supported by Libero and FlashPro Express in this release. For other SPI-Flash device support, contact technical support.
- This tool erases the SPI Flash prior to programming. It is recommended to program the SPI Flash with Libero SoC v12.2 prior to programming other data on the SPI Flash using non-Libero programming solutions.
- Partial update of the SPI Flash is currently not supported.
- It is not recommended to have huge gaps between clients in the SPI Flash, since gaps are currently programmed with 1's and will increase programming times.

See the appendix for programming tables.

4.15.3 sNVM write fails due to ROM client created by previous design

In the scenario where a PolarFire device is first programmed with a design with an sNVM client, and then reprogrammed with a (different) design without an sNVM client, upon completion of programming with the second design, the sNVM client will not be erased. In such a case, if there are sNVM pages that are locked, writes to those pages will fail.

There is no programming action to erase the sNVM completely.

Workaround: Create a dummy sNVM client (filled with 0's) in the second design.

4.15.4 PolarFire VERIFY_DIGEST action may fail in certain cases

The VERIFY_DIGEST step in FlashPro Express for a PolarFire device in Libero SoC v12.2 release will fail in cases where the digest check is run for segments that are not programmed.

Workaround:

Using FlashPro v2.3 (part of Libero SoC PolarFire v2.3 or Programming and Debug Tools PolarFire v2.3), load the STAPL file into FlashPro, and deselect the digest checks for segments not programmed.

4.15.5 FlashPro will error out, if an existing PDV is modified to disable the fabric

If an existing PDB file is modified to disable the fabric and programmed only with eNVM, FlashPro will error out.

Workaround:

Create a new FlashPro project and create a new PDB. Enable eNVM and import the efc file required for programming eNVM. Save the PDB and use this PDB to program the device.

4.15.6 Flashpro6 Turbo Mode Limitations

FlashPro6 Turbo mode does not work at 1 MHz and 2 MHz frequencies.

4.16 SmartDebug

4.16.1 General SmartDebug Limitations

- Initializing RAM blocks with random values in the Design Initialization Data and Memory tool will result in SmartDebug displaying incorrect values for zeroed memory blocks.
- The logical view cannot be reconstructed for:
  - LSRAM/uSRAM for port widths of x1 inferred through RTL.
  - LSRAM/uSRAM configurations when a single net of an output bus is used and others are unused (i.e. A_DOUT[0]/B_DOUT[0] for DPSRAM/uSRAM and RD[0] for TPSRAM). In this scenario, the memories can be read/write using physical view.
  - LSRAM/uSRAM configurations inferred using CoreAHBLtoAXI (Verilog flow), CoreFIFO (Verilog and VHDL flow).
  - HDL modules inferring RAM blocks that are instantiated in SmartDesign.
  - TPSRAM with ECC enabled.

Workaround: There are no workarounds for the issues above at this time.

4.16.2 PolarFire Transceiver Support Limitations

- Plot Eye introduces a burst of errors in data traffic on Transceiver Interface lanes when started. This will be fixed in an upcoming Libero SoC PolarFire release.

  Workaround: Enable Eye Monitor using the Power On Eye Monitor option before starting the traffic. This will power on the DFE and EM receivers in CDR mode and no spurious errors will be seen during eye plot.

- The Custom DFE solution (using the Optimize DFE option in the Eye Monitor tab) does not work when the transceiver is configured in 8B10B PCS-PMA mode and the receiver is DFE.

  Workaround: Perform the following steps to obtain the expected eye output:

  a. Assert PCS RX RESET
  b. Optimize DFE
  c. Plot Eye
  d. De-Assert PCS RX RESET
- The SmartBERT IP does not work when lanes are configured at 250Mbps data rate.
- SmartBERT IP PRBS tests take more time to start/stop/inject error on RHEL 7.x and Cent OS 7.x platforms as compared to RHEL 6.x and Windows OS. This issue is seen only with PRBS patterns from the SmartBERT IP, and will be fixed in an upcoming Libero SoC release.
- During the Static Pattern Transmit operation, the Receiver PLL (RX PLL) does not lock to the max run length pattern when looped back from TX to RX.
- SmartBERT IP PRBS tests do not work when the first Transceiver lane uses an internal pattern (PRBS from XCVR PMA) and the following lane uses a SmartBERT pattern (PRBS from SmartBERT IP).
- The Power ON eye monitor Tcl command (eye_monitor_power) does not work correctly in Libero SoC v12.2. The Receive PLL does not lock to the incoming data after this Tcl command is run. This will be fixed in an upcoming Libero SoC release.

Workaround: There are no workarounds for the issues above at this time.

4.16.3 PolarFire Signal Integrity Support Limitations

- The RX Polarity Signal Integrity parameter (Polarity P/N reversal) has no effect when a PDC file is imported using the Import option in SmartDebug. This flow works without errors in GUI mode. This will be fixed in an upcoming Libero SoC release.

4.16.4 PolarFire FPGA Hardware Breakpoint (FHB) Limitations

- Soft Reset behavior is not consistent when the DUT is clocked at frequencies less than 160MHz. In such cases, the device may not respond to a Soft Reset operation initiated from the FHB UI. A potential workaround is to do the following:
    a. Halt the DUT via Live Probe
    b. Initiate a Soft Reset operation using the FHB UI
    c. Halt the DUT again via Live Probe

4.16.5 RTG4 FPGA Hardware Breakpoint (FHB) Limitations

- Synthesis fails when FHB auto instantiation is enabled on designs containing FCCC modules using instantiation flow (i.e. FCCC modules instantiated in RTL files).
- Live Probe channel assignment using a static signal (connected to GND) halts the DUT. If this occurs, initiate the Soft Reset operation using the FHB UI to restart the DUT.
- Halting a clock domain driven by a CCC will also halt all four clock domains belonging to the CCCs in the design. This is a silicon limitation.
- FHB is not supported for cascaded CCCs (CCCs that rely on other CCC outputs to source the reference/feedback clocks). FHB is auto-instantiated, but the PLAY/HALT/STEP operations do not work.

4.16.6 RTG4 LSRAM Data corruption

- LSRAM Data corruption is seen on doing a read to LSRAM configured in 512x36 mode through SmartDebug on the active address location.

4.16.7 Missing Programming information in View Device Status report

Information about the programmer, software version, programming file type and Programming Software is missing in the View Device Status report when the device is programmed with PPD file using FlashPro6.

4.16.8 Standalone SmartDebug Limitations

- Microsemi devices present in chain along with non-Microsemi devices cannot be debugged using standalone SmartDebug. In addition, the ID code of Microsemi devices cannot be read in this scenario.

  Workaround: Use SmartDebug through the Libero flow to perform these operations.

- Programming fails for RTG4 devices when a standalone SmartDebug project is created using the "Construct Chain Automatically" option, and a DDC file is imported in the Programming Connectivity and Interface dialog.

  Workaround:

    a. Close and reopen the Programming Connectivity and Interface UI and then click **Run Program Action**.

    (or)

    b. Create a project by importing the DDC file (without Auto-construct).

- If TCK frequency is set in the Programmer Settings in Standalone SmartDebug, it is not reflected in the SmartDebug operations for SmartFusion2, IGLOO2 and RTG4 devices.

  Workaround: Set TCK frequency; program the device and then use SmartDebug features to debug. This issue will be fixed in the upcoming Libero SoC v12.3 release.

- Standalone SmartDebug User Guide fails to open when no project is created

  Workaround: To open the user guide from standalone SmartDebug, a new project must be created. This will be fixed in upcoming releases.

- FlashPro6 programmer will not be detected during SmartDebug project creation when SmartDebug is invoked for the first time after the installation of Program and Debug software.

  Workaround: Exit the SmartDebug instance, invoke a new instance of SmartDebug, and the project creation will be successful.

  This issue will be fixed in upcoming releases.

4.17 Secure Production Programming Solution

4.17.1 Job Manager may crash during the Import Job Status operation

Running the Import Job Status operation may cause Job Manager to crash. This may happen for designs where the User ECC Key mode is enabled, targeting SmartFusion2 or IGLOO2 M2S/M2GL060, 090, or 150 devices.

4.17.2 Job Manager crashes when opening an existing Job Manager project from v11.9

Job Manager v12.2 does not support Job Manager project files created with releases prior to v12.0.

4.17.3 Job Manager does not support PolarFire DAT export

PolarFire DAT file bitstream export from Job Manager is not supported in Libero SoC v12.2.

4.17.4 SmartFusion2/IGLOO2: eNVM update protection with FlashLock is no longer supported

Due to a silicon limitation, eNVM update protection with FlashLock has been defeatured. If a JDC file generated with a pre-v12.0 version of Libero SoC had the eNVM set to be protected by passkey, it must be regenerated with Libero SoC v12.2 without eNVM FlashLock Protection enabled. eNVM update protection continues to be provided by User Encryption Keys (UEK1, UEK2 or UEK3).

4.17.5 ERASE Action failure for FlashPro Express Job

If a HSM FlashPro Express job has tickets for PROGRAM and ERASE actions, without a ticket for the VERIFY action, the ERASE action will fail. To successfully run the ERASE action, ensure that a ticket for the VERIFY action is included.

4.17.6 Job Manager init_bitstream Tcl command limitation

On Windows, when you run non-HSM flow using Job Manager on PC, if the "enable_passkey_export" option in init_bitstream Tcl command is not specified, the exported bitstream files may include passkeys.

Workaround: You must explicitly set the "enable_passkey_export" option to either TRUE or FALSE in the init_bitstream Tcl command to export the correct bitstream files.

4.18 Identify Debugger

4.18.1 Identify Instrumentor may hang on some Windows 10machines

When the Identify Instrumentor is opened in Integrated mode on certain machines, the tool opens, but, upon interaction, it freezes. This is an isolated issue, and happens on rare Windows 10 OS configurations.

Workaround: Use the Standalone Identify Instrumentor.

4.18.2 Identify Debugger is not supported with FlashPro 6 Hardware

The Identify Debugger packaged with Libero SoC v12.2 does not support the new FlashPro 6 programming hardware. Support will be added in a future release of Identify, to be released standalone on the Microsemi web site.

4.19 Installation and System Limitations

4.19.1 Libero does not run on 8TB File Systems

Libero is currently only supported for partitions 2TB or smaller. If either the Libero install or the Libero project is located on a partition that is larger than 2TB, file access errors or tool crashes may occur. Support for larger partitions is expected to be added in an upcoming release.

4.19.2 4K and 8K screens are not supported

4K and 8K screens are not supported in the Libero SoC v12.2 release.

4.19.3 Installation on Local Drive Only

This release is intended for installation only on a local drive. The Installer might report permission rights problems if the release is installed across a networked drive.

4.19.4 Visual C++ Redistributable Installation Error

On some machines, the installer may display a message stating:

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

The above error message is benign. If it is seen, click Yes and Libero SoC v12.2 will be installed successfully.

4.19.5 Installation on Windows 7

During Libero SoC v12.2 installation on Windows 7 machines, you may see pop-up warning messages about shortcuts toward the end of installation process.

These messages can be safely ignored. Click OK to close the pop-up windows and the installation will proceed and complete as expected. All Windows shortcuts will appear correctly.

4.19.6 Installation fails when there is insufficient space

In Libero SoC v12.2, the web installer quits without any user notification or error message when there is insufficient space for the installation. In addition, the estimated space for the installation is incorrect – it reads as approximately 236MB required. Ensure that there is at least 20GB free space on the target hard drive before invoking the installer.

The DVD installer will also not proceed if there is insufficient space.

4.19.7 Windows Standalone Installer: Spaces in Extraction Path

During installation of the standalone (DVD) version, the folder to which the zip file is extracted must not contain spaces. If spaces are present, invocation of the installer will fail with the error *"Windows cannot find '<truncated path to extracted folder>'. Make sure you typed the name correctly, and then try again"*. Rename and/or move the extracted folder to one without spaces (in the entire path).

4.19.8 Linux Package Note

In Libero SoC v12.2, the script bin/check_linux_req/check_linux_req.sh incorrectly reports that the Linux package xz.i686 is required for RHEL/CentOS 7.x. Package xz.i686 is not required. The correct required packages are xz-libs.x86_64 and xz-libs.i686.

4.19.9 Antivirus Software Interaction

Many antivirus and HIPS (Host-based Intrusion Prevention System) tools will flag executables and prevent them from running. To eliminate this problem, users must modify their security setting 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 PolarFire successfully with no modification to their antivirus software. Microsemi is aware of issues for some antivirus tool settings that occur when using Symantec, McAfee, Avira, Sophos, and Avast tools. 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 v12.2, ModelSim ME and/or Synplify Pro ME may or may not be affected.

All public releases of Libero software are tested with several antivirus tools before they are released to ensure that they are not infected. In addition, Microsemi’s software development and testing environment is also protected by antivirus tools and other security measures.

4.20 Software Update

The manual software update check is not working. Even if there is an update available, clicking Help -> Check for Software Updates has no effect. This will be fixed in an upcoming release.

Automatic software update check is working and can be enabled in the Project -> Preferences -> Software Update menu.

5 System Requirements

The Libero SoC v12.2 release has the following system requirements:

- 64-bit OS
  - Windows 7, or Windows 10 OS
  - RHEL 6.6-6.11, RHEL 7.2-7.6, CentOS 6.6-6.11, and CentOS 7.2-7.6
- A minimum of 16 GB RAM

Note: Setup instructions for using Libero SoC v12.2 on Red Hat Enterprise Linux OS or CentOS are available here. As noted in that document, installation now includes running a shell script (bin/check_linux_req.sh) to confirm the presence of all required runtime packages.

6 Download Libero SoC v12.2 Software

The following are available for download:

- Libero SoC v12.2 for Linux
- Libero SoC v12.2 for Windows
- Libero SoC v12.2 MegaVault

Note: Installation requires administrative privileges.

After successful installation, clicking **Help-> About Libero** will show Version: 12.700.0.21

7 Appendix: Sample Programming Times Using FlashPro5/FlashPro6

The tables in this appendix show sample programming times using FlashPro5 and FlashPro6 programmers.

The following table shows sample PPD programming times.

Devices[1]	PPD Programming Time[2] (mm:ss)		
	FlashPro5	FlashPro6	
	TCK=4MHz USB 2.0	TCK=4MHz	TCK=20MHz[3]
		USB 2.0/3.0	USB 2.0/3.0
M2S/A2GL 050	2min 9sec	2min 10sec	2min 2sec
M2S/A2GL 150	4min 21sec	4min 19sec	3min 54sec
MPF100	39sec	28sec	23sec
MPF200	1min 3sec	43sec	28sec
MPF300	1min 33sec	1min 4sec	43sec
MPF500	1min 57sec	1min 34sec	1min

NOTE: To successfully program the device at a high TCK frequency, appropriate steps must be taken to ensure signal integrity on the JTAG signals.

The following table shows sample SPI Flash PPD programming times.

SPI Flash Programming

Splash Kit
ID: 00441021ba20

(N25Q00AA13GSF40G / MT25QL01GBBB8ESF-0SIT TR)[1] 10MByte data	PPD Programming Time				
	FlashPro5		FlashPro6[2]		
	TCK = 4MHZ	TCK = 15MHz[3]	TCK = 4MHZ	TCK = 15MHz[3]	TCK = 20MHz[3]
	USB 2.0	USB 2.0	USB 2.0/3.0	USB 2.0/3.0	USB 2.0/3.0
Program SPI Flash	8min 30sec	5min 29sec	22min 12sec	13min 9sec	12min 11sec
Verify SPI Flash	2hrs 22min 10sec	2hrs 10min 28sec	23min 05sec	14min 17sec	13min 21sec
Read SPI Flash	2hrs 34min 20sec	2hrs 23min 45sec	22min 30sec	13min 54sec	12min 55sec
Erase SPI Flash	19sec	18sec	1min 51sec	1min 49sec	1min 48sec

NOTES:

[1] SPI Flash programming has been tested on N25Q00AA and MT25QL01G/MT25QU01G devices only. Contact technical support for other SPI-Flash device support issues.

[2] FlashPro6 in the Programming and Debug Tools v12.2 release has longer programming time as compared to FlashPro5. However, readback and verification times are significantly shorter. Programming time will be improved in future releases.

[3] To successfully program the device at a high TCK frequency, appropriate steps must be taken to ensure signal integrity on the JTAG signals.