Programming and Debug Tools v12.2

Release Notes

9/2019

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.

51300227-1.0/9.19

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 1.0

Revision 1.0 is the first publication of this document.

Contents

1 Programming and Debug Tools v12.2 Software Release Notes

Microsemi's Programming and Debug Tools installer is intended for laboratory and production environments where Libero is not installed, and allows you to install the following tools:

- FlashPro Express
- SmartDebug Standalone
- Job Manager

1.1 Device Support

Programming and Debug Tools v12.2 supports IGLOO2, SmartFusion2, RTG4 and PolarFire families.

2 What's New in Programming and Debug Tools v12.2

2.1 FlashPro6 Support

Libero SoC v12.2 adds support for the next-generation Microsemi Flash Programming System, FlashPro6. FlashPro6 offers faster Programming time than FlashPro3/4/5 when using PPD file. See the appendix for details.

FlashPro6 currently supports the following platforms:

- **Device families**: SmartFusion2, IGLOO2, RTG4, PolarFire
- **SPI-Flash:** N25Q00AA, MT25QL01G
- **Operating systems**: Windows 7, Windows 10, RedHat and CentOS versions 6.6-6.11; 7.2-7.6
- **Software Tools**: Libero, FlashPro Express, SmartDebug, Identify (PC only)

Note

- RTG4 devices are fully 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.
- For supporting other SPI-Flash devices on FlashPro6, contact technical support.

2.2 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 compatibility with the current software version is verified. In cases of incompatibility, the DDC file must be regenerated with the same Libero release version.
    - 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).

2.3 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.

3 Resolved Issues

The following table lists the customer-reported SARs resolved in libero SoC v11.9 SP4 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
493642-2544111080	RTG4: IO glitch during programming
	SmartFusion2 SmartDebug: DPK not working when permanent settings are enabled
493642-2600504382	Enhancement Request: FlashPro Express support for SPI-Flash Programming
493642-2588964932	FP5: programming support on CentOS v7.6 (and potentially RHEL 7.6)
493642-2573306385	PolarFire: During bitstream export EOB digest component shows all zeros
493642-2589714643	FlashPro5 Tck mode text should be replaced from free running to Discrete
	Change the warning message displayed while adding SmartBERT Core to Libero SoC v12.0 Catalog

4 Known Issues and Limitations

4.1 Programming

4.1.1 Libero Programming

- Updating the security-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: 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.

4.1.2 SPI Flash Programming

This release includes the following limitations:

- Only the Micron SPI Flash is currently supported with the PolarFire SPLASH 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, you must contact the tech 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.

Note: Depending on the SPI-Flash memory silicon version, you may observe a shorter erase time.

See the appendix for details.

4.1.3 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.

4.1.4 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.

4.1.5 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

4.1.6 Flashpro6 Turbo Mode Limitations

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

4.2 SmartDebug

4.2.1 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.

4.2.2 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.

4.2.3 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.

4.2.4 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

4.2.5 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.

4.2.6 RTG4 LSRAM Data corruption

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

4.2.7 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.

4.2.8 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**.
  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.

4.3 Secure Production Programming Solution

4.3.1 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.

4.3.2 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.

4.3.3 Job Manager does not support PolarFire DAT export

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

4.3.4 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).

4.3.5 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.

4.3.6 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.

4.4 Identify Debugger

4.4.1 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.

4.4.2 Identify Debugger supporting FP6

Identify Debugger cannot read back the sample values from the hardware using FP6 programmer for the PolarFire designs having UJTAG_SEC macro instantiated in it.

5 System Requirements

The Programming and Debug Tools 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 Programming and Debug Tools

Click the following links to download Programming and Debug Tools v12.2 on Windows and Linux operating systems:

- Windows Download
- Linux Download
- Mega Vault Download

Note: Installation requires administrator privileges to the system.

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 USB 2.0/3.0	TCK=20MHz[3] 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.