Programming and Debug Tools v12.3

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

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.

51300232-1.1/12.19

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 1.1

Revision 1.1 includes the following updates:

- Updated section 2.1
- Added known issue in section 4.1
- Updated table and footnotes in section 7.2

Revision 1.0

Revision 1.0 was the first publication of this document.

Contents

1 Programming and Debug Tools v12.3 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 v12.3 supports IGLOO2, SmartFusion2, RTG4 and PolarFire families.

2 What's New in Programming and Debug Tools v12.3

Programming and Debugging Tools v12.3 includes the following new features and enhancements.

2.1 FlashPro6 Support

Programming and Debugging Tools v12.3 continues to enhance support for the next-generation Microsemi Flash Programmer, FlashPro6. FlashPro6 offers faster Programming time and faster device debugging actions.

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
- **Programming interface**: JTAG and SPI-Slave

Users of FlashPro6 will see the following speedups in Libero SoC v12.3 (relative to FlashPro5):

- Up to 25% speedup in Programming actions for RTG4 devices
- Up to 50% speedup in Programming actions for PolarFire devices
- Significant speedup in combined Erase, Program and Verify actions for SPI Flash memory devices
- Significant speedup in SmartDebug actions

Refer to the Appendix for details on Programming and Debug time improvements possible with FlashPro6

2.2 FlashPro Express

2.2.1 DEVICE_INFO Enhancements

With this release, the DEVICE_INFO programming action now reports the following new data for SmartFusion2 and IGLOO2 devices:

- The family name and product name of the device being programmed
- The status of the Cortex-M3 (enabled or disabled)

2.2.2 PolarFire SPI-Slave Programming

With this release, PolarFire SPI-Slave Programming is supported using FlashPro6.

Note:

1. To enable SPI-Slave programming, in 12.3 using Flashpro6, you must update your Flashpro6 programmer using the utility posted in the Programmer Hardware page under the Flashpro6 section, which will be added when FlashPro6 is available to order.
2. PolarFire SPI-Slave Programming is only supported using FlashPro6 only moving forward.

2.2.3 SPI-Flash Memory Programming support

With this release, Libero and FlashPro Express support programming all members of N25Q and MT25Q Flash Memory family.

Note: For support Flash memory devices from other vendors and device families using FlashPro6, contact Microsemi Technical Support.

2.3 SmartDebug

The following new SmartDebug features have been added in this release:

- FPGA Hardware Breakpoint (FHB) Auto Instantiation support for Synthesized Verilog Netlist files (files that are imported using Import → “Synthesized Verilog Netlist(*.vm) option in Libero)
- Addition of Project Checksum field to let users know if the design programmed in the device matches the design used for debug

3 Resolved Issues

The following table lists the customer-reported SARs resolved in libero SoC v11.9 SP5 Libero SoC v12.3. Resolution of previously reported “Known Issues and Limitations” is also noted in this table.

3.1 List of Resolved Issues

Case Number	Description
493642-2574818193, 493642-2590831388, 493642-2596337143, 493642-2659811742	Unsigned driver actelsvc.sys in FlashPro v11.9 installation
493642-2549800616	Support PolarFire SPI-Slave programming with FlashPro6

4 Known Issues and Limitations

4.1 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 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.
- 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 Programming and Debug Tools v12.3. Serialization of the eNVM client is not working for Programming and Debug Tools v12.3.

  Workaround: Use Programming and Debug Tools v11.9.

- The action device_info shows programming file type as SVF instead of PPD when job-ppd is used.
- Programming is not supported for the RTG4150L device.

4.1.1 SPI Flash Programming

This release includes the following limitations:

- Supports only the following Micron SPI Flash memory devices:
  - Using FlashPro5: MT25QL01G only
  - Using FlashPro6: All members of N25Q and MT25Q device families.

  Note: For support Flash memory devices from other vendors and device families using FlashPro6, contact Microsemi Technical Support.

- This tool erases the SPI Flash prior to programming. It is recommended to program the SPI Flash with Programming and Debug Tools v12.3 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.

4.1.2 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.

4.1.3 FlashPro will error out, if an existing PDB 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.

4.1.4 Verify during PROGRAM action fails for design with custom user security

For Programming and Debug Tools v12.0 and above, PolarFire designs having Custom user security options, enabling DO_VERIFY optional procedure in PROGRAM action and executing PROGRAM action in Libero (via Run PROGRAM action) will fail with "Invalid/Corrupted encryption key".

Workaround:
Run standalone VERIFY action after PROGRAM separately if needed.

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 currently.

4.2.2 PolarFire Transceiver Support Limitations

- Plot Eye introduces a burst of errors in data traffic on Transceiver Interface lanes when started for lanes configured in CDR mode. 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. This will be fixed in an upcoming Libero SoC PolarFire 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 Programming and Debug Tools v12.3. 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 PolarFire release.

  Workaround: There are no workarounds for the above issues.
- Designs using PCIe1 controller:
  - RXPLL lock status is not shown in the Configuration Report UI of Debug XCVR. This will be fixed in an upcoming Libero SoC PolarFire release.
  - Signal Integrity parameters are shown in the Configuration Report UI of Debug XCVR. This will be fixed in an upcoming Libero SoC PolarFire release.
- Designs using Dual PCIe i.e. PCIe0 and PCIe1:
  - PCIe debug is not supported for designs using dual PCIe controllers. This will be fixed in an upcoming Libero SoC PolarFire release.

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 PolarFire 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 SmartFusion2/IGLOO2 FPGA Hardware Breakpoint (FHB) Limitations

- 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.

4.2.6 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.

4.2.7 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.

4.2.8 Standalone SmartDebug Limitations

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

  Workaround: Use SmartDebug through Libero to perform these operations.

- Programming fails for all device families 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 after importing the DDC file contents in Programming Connectivity and Interface, 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, RTG4 and PolarFire devices.

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

- Standalone SmartDebug User Guide fails when invoked from the UI.

  Workaround: There is no workaround for this; it will be fixed in an upcoming Libero SoC PolarFire release.

- 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 project creation will be successful.

- In Standalone SmartDebug, for designs containing security settings, program action (for a device that was already programmed with security settings) and erase action done in Programming Connectivity and Interface result in failure.

  Workaround: Use FlashPro Express or SmartDebug through Libero to Erase/Program the device for designs containing security settings. This will be fixed in an upcoming Libero SoC PolarFire release.

- Read eNVM in Standalone SmartDebug fails when security settings are enabled on the design.

  Workaround: Use SmartDebug through Libero to read eNVM when security settings are enabled on the device. This will be fixed in an upcoming Libero SoC PolarFire release.

- Project Checksum warning message gets retained after programming the device when debug project is created using import DDC option.

  Workaround: Close and reopen the debug project to clear the warning message.

  This will be fixed in an upcoming Libero SoC PolarFire release.

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 v12.3 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 Programming and Debug Tools v12.3.

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 Programming and Debug Tools v12.3 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 10 machines

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 in rare Windows 10 OS configurations.

Workaround: Use the Standalone Identify Instrumentor.

4.4.2 Identify Debugger is not supported with FlashPro6 Hardware

The Identify Debugger packaged with v12.3 does not support the new FlashPro6 programming hardware. Support will be added in a future release of Identify, to be released standalone on the Microsemi website.

4.5 Installation

4.5.1 FlashPro6 driver re-installation reports error message

Customers with FlashPro6 drivers previously installed on their system may see the following message at the end of the installation:

```
"The installation of Program Debug Tool v12.3 is finished, but some errors
occurred during the install. Please see the installation log for details."
```

Resolution: Uninstall existing FlashPro6 drivers and restart the system before installing Libero SoC v12.3. If the software is already installed, ignore the above message if installation logs do not report any errors.

4.5.2 Linux Package Note

- In Libero SoC v12.3, 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.
- If the installer does not come up in graphical mode, additional X window system libraries might be required. For RHEL/CentOS, the following system packages are recommended:

```
$ sudo yum install -y libXau libX11 libXi libxcb libXext libXtst libXrender
```

5 System Requirements

The Programming and Debug Tools v12.3 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 Programming and Debug Tools v12.3 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 v12.3 Programming and Debug Tools

Click the following links to download Programming and Debug Tools v12.3 on Windows and Linux operating systems:

- Windows Download
- Linux Download
- Windows Mega Vault Download
- Linux Mega Vault Download

Note: Installation requires administrator privileges to the system.

7 Appendix: Sample Programming and SmartDebug Times Using FlashPro5/FlashPro6

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

7.1 Microsemi FPGA Array Programming

The following table shows sample PPD programming times of the FPGA Array.

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
RTG4	2min 10sec	1min 56sec[4]	1min 33sec[4]
MPF100	39sec	28sec	23sec
MPF200	1min 3sec	43sec	28sec
MPF300	1min 33sec	1min 4sec	43sec
MPF500	1min 57sec	1min 34sec	1min

[1] FlashPro6 supports JTAG programming for all SmartFusion2, IGLOO2, RTG4 and PolarFire devices.
[2] To benefit from the improved programming time using FlashPro6, use the PPD file format for SmartFusion2, IGLOO2 and PolarFire devices.
[3] To ensure successful programming at 20MHz TCK, appropriate steps need to be taken to ensure signal integrity of the JTAG signals.
[4] New and improved programming time for RTG4 starting with Libero SoC/FlashPro Express v12.3 and later.

7.2 SPI Flash Programming

The following table shows sample SPI Flash Programming time using the PolarFire Splash Kit.

(N25Q00AA13GSF40G / MT25QL01GBBB8ESF-0SIT TR)[1] 10MByte data	SPI Flash 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
Erase and Program SPI Flash[4]	8min 15sec	4min 58sec	14min 53sec	5min 45sec	4min 54sec
Verify SPI Flash	1hr 57min 38sec	1hr 50min 45sec	16min 33sec	7min 53sec	7min 04sec
Read SPI Flash	2hrs 02min 43sec	1hr 55min 30sec	16min 12sec	7min 36sec	6min 47sec
Erase SPI Flash	18sec	18sec	1min 52sec	1min 50sec	1min 50sec

NOTES:

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

[2]FlashPro6 has longer erase and programming times for SPI Flash devices, when compared to FlashPro5. However, readback and verification times are significantly shorter. As a result, the total combined Erase, Program and Verify time is significantly lower compared to FlashPro5. Programming time for FlashPro6 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 of the JTAG signals.

[4]SPI Flash programming time may vary from device to device even though the part number is the same. This is due to die to die variation.

7.3 SmartDebug Runtime Samples

The following table shows sample runtimes of some SmartDebug key functions.

SmartDebug Function Runtimes		
SmartDebug Operations	**FlashPro5**	**FlashPro6[1]**
	TCK = 4MHZ	**TCK = 4MHZ**
	USB 2.0	**USB 2.0/3.0**
Active Probe Read (13,000 probe points)	28 sec	1 sec
Active Probe Write (13,000 probe points)	35 sec	6 sec
Logical View Read of LSRAM (340 LSRAM Blocks)	20 min	<5 min
Logical View Read to USRAM (32 USRAM Blocks)	1 sec	1 sec
FHB - Waveform dump to VCD file (160 probe points; 1,000 cycles)	7 min	25 sec

NOTES:

[1]FlashPro6 SmartDebug runtime is applicable for SmartDebug v12.3 and later only.