Programming and Debug Tools v12.4

Release Notes

04/2020

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.

51300237-1.0/04.20

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 was the first publication of this document.

Contents

1 Programming and Debug Tools v12.4 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 v12.4 supports IGLOO2, SmartFusion2, RTG4 and PolarFire families.

2 What's New in Programming and Debug Tools v12.4

Programming and Debugging Tools v12.4 includes the following new features and enhancements.

2.1 2TB and Greater Partition Size Support

FlashPro Express and SmartDebug v12.4 now fully supports Linux environments using partitions greater than 2TB, including installing Libero and locating Libero projects.

2.2 PolarFire Transceiver Debug Enhancement

SmartDebug v12.4 for PolarFire simplifies transceiver receiver signal integrity analysis by including an eye mask that specifies the minimum eye opening.

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
493642-2544111080	FP6: Add EXTEST2 instruction to CM3 code to solve I/O glitch issue when programming with PPD

4 Known Issues and Limitations

4.1 Programming

Programming Limitations

- Updating PolarFire security or sNVM with a security-only bitstream or sNVM-only bitstream on a device that has the Fabric programmed disables the Fabric. If the Fabric is disabled, you must reprogram the Fabric to enable it.

Workaround:

1. sNVM-only bitstreams: Field-update bitstream files should always program the Fabric with sNVM.
2. Security-only bitstreams: Security-only bitstream should be used on a blank device only.

- If a PolarFire, SmartFusion2, or IGLOO2 device is programmed with a blank Silicon Signature field, it does not get erased.

Workaround:

1. Specify a Silicon Signature that is not blank and program the device to change the value.
2. Perform the Erase program action to erase it.

- A PolarFire, SmartFusion2 or IGLOO2 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:

Leave the optional Silicon Signature Field in “Configure Programming Options” blank before generating a SPI bitstream. If Recovery bitstream must include a silicon signature, use Libero SoC v12.0 software.

- For SmartFusion2 or IGLOO2, if the Silicon Signature that is part of the security segment is unspecified and the security is not programmed, the previous value of Silicon Signature is retained.

- Serialization feature does not work for SmartFusion2 and IGLOO2 in starting from Libero SoC v12.0.

- The action device_info shows programming file type as SVF instead of PPD when job-ppd is used.

sNVM Write Fails Due to ROM Client Created by Previous Design

- If a PolarFire device is programmed with a design with an sNVM client, and then reprogrammed with a different design without an sNVM client, the sNVM client is not erased when programming for the second design is completed. If there are sNVM pages that are locked, writes to those pages fail.
- There is no programming action to erase the sNVM completely.

Workaround: Create a mock (dummy) sNVM client (filled with 0's) in the second design. The issue will be fixed in the upcoming release.

Verify During PROGRAM Action Fails for Design with Custom User Security

For Libero SoC v12.0 and above that use PolarFire designs with Custom user security options, enabling the DO_VERIFY optional procedure in PROGRAM action and executing PROGRAM action in Libero (via Run PROGRAM action) fails with "Invalid/Corrupted encryption key".

Workaround:

Run the standalone VERIFY action after PROGRAM separately if needed.

SPI-Flash Programming Limitations

- FlashPro Express does not support SPI-Flash only programming in a JTAG chain with SmartFusion2/IGLOO2 and PolarFire devices.
- This release supports only the following Micron SPI Flash memory devices:
    - Using FlashPro5: MT25QL01G only
    - Using FlashPro6: all members of N25Q and MT25Q device families

  Note: Contact Microchip Technical Support about support for Flash memory devices from other vendors and device families using FlashPro6.

- This tool erases the SPI Flash prior to programming. It is recommended to program the SPI Flash with Programming and Debug Tools v12.4 prior to programming other data on the SPI Flash using non-Libero programming solutions.
- Partial update of the SPI Flash is currently not supported.
- Avoid large gaps between clients in the SPI Flash, since gaps are currently programmed with 1's and increase programming times.

See the appendix for programming tables.

STAPL Programming is Not Supported Using FP6

FlashPro6 does not support STAPL programming.

Workaround:

Generate FlashPro Express jobs with a PPD file to enable programming for FlashPro6. Note that the PPD file is also supported by FlashPro 4 and 5.

Verify Digest May Not Work with FlashPro6

"Verify Digest" action may cause intermittent failure and generate the error messages "FP6 connection failed" and "Failed to disable FP6 programmer." To continue, remove FlashPro6 from the USB port, and then plug it back into the USB port

Workaround:

Use VERIFY action to verify that the device is programmed with the expected design.

4.2 SmartDebug

General SmartDebug Limitations

- Initializing RAM blocks with random values in the Design Initialization Data and Memory tool results 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, memories can be read/write using physical view.
    - LSRAM/uSRAM configurations inferred using CoreAHBLtoAXI (Verilog flow) and CoreFIFO (Verilog and VHDL flow).
    - HDL modules inferring RAM blocks that are instantiated in SmartDesign.
    - TPSRAM with ECC enabled.

Workaround: None

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

  Workaround: Before starting traffic, use the Power On Eye Monitor option to enable Eye Monitor. This powers on the DFE and EM receivers in CDR mode, and no spurious errors are seen during eye plot.

Eye Monitor Limitation on 8b10b PCS-PMA mode XCVR lanes

XCVR lanes configured as 8b10b PCS-PMA with receiver in DFE mode require a specific interface sequence to be followed for Eye Monitor to work.

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

1. Assert PCS RX RESET.
2. Optimize DFEReceiver.
3. Plot Eye.
4. De-assert PCS RX RESET.

SmartBERT Limitations

- The SmartBERT IP does not work when lanes are configured at 250Mbps data rate.
- SmartBERT IP PRBS tests using PRBS patterns from the SmartBERT IP take more time to start/stop/inject error on RHEL 7.x and Cent OS 7.x platforms than on RHEL 6.x and Windows OSes.

- 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).

  Workaround: If multiple lanes are selected to run the tests, test patterns on every lane should be either SmartBERT IP pattern or on-chip PMA pattern.

Static Pattern Transmit Limitation

- 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.

Eye Monitor TCL Limitation

- The Power ON eye monitor Tcl command (eye_monitor_power) does not work correctly in Programming and Debug Tools v12.3 and v12.4. The Receive PLL does not lock to the incoming data after this Tcl command is run. This issue will be fixed in an upcoming Libero SoC PolarFire release.

PCIe Debug Limitations

- Designs using PCIe1 controller:
  - RXPLL lock status is not shown in the Configuration Report UI of Debug XCVR.
  - Signal Integrity parameters are shown in the Configuration Report UI of Debug XCVR
- Designs using Dual PCIe i.e. PCIe0 and PCIe1:
  - PCIe debug is not supported for designs using dual PCIe controllers.

PolarFire FPGA Hardware Breakpoint (FHB) Limitations

- Soft reset behavior is not consistent when the DUT is clocked at frequencies less than 160MHz. Consequently, the device might not respond to a soft reset operation initiated from the FHB UI.

  Workaround: Perform the following steps:

  1. Halt the DUT via Live Probe.
  2. Use the FHB UI to initiate a soft reset operation.
  3. Halt the DUT again via Live Probe.

- TCL command for frequency_meter fails when run in SsmartDdebug. This issue will be fixed in an upcoming Libero SoC PolarFire release.
- Event Counter value is read as 2 for static signals (signals connected to GND/VCC) when assigned to Live Probe Channel A.

SmartFusion2/IGLOO2 FPGA Hardware Breakpoint (FHB) Limitations

- When the selected live probe trigger signal is zero/ground, disarming the trigger leads to the forced halt of Device Under Test (DUT).

  Workaround: use the reset option in FHB to bring FHB to a default state and un-halt the DUT.

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 a soft reset operation using the FHB UI to restart the DUT.
- Due to a silicon limitation, halting a clock domain driven by a CCC also halts all four clock domains of the CCCs in the design.
- 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.
- LiveProbe is not Retained on PROBE_READ_DATA Pin for 2nd SmartDebug Run Using FlashPro5 for RTG4 Devices
    - If the LiveProbe is set initially in the first SmartDebug session/run, it is not retained when a new session is invoked.

    Workaround: A def variable is provided "SMARTDEBUG_RTG4_FLASHPRO5_DISABLE_RESET" a value '0' is by default. A value '1' has to be overwritten to avoid reset.

RTG4 LSRAM Data Corruption

- LSRAM data corruption occurs when performing a read to LSRAM configured in 512x36 mode through SmartDebug on the active address location.

Standalone SmartDebug Limitations

- Microchip devices present in a chain 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.

Auto-Chain Construct Limitation in Standalone SmartDebug

- 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: Perform one of the following steps:

    1. 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**.
    2. Create a project by importing the DDC file without Auto-construct.

TCK Frequency Not Reflected in SmartDebug Operations

- 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 issue will be fixed in an upcoming Libero SoC PolarFire release.

FlashPro6 Programmer Not Detected When Creating a SmartDebug Project

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

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

Incorrect programmer information reported for FlashPro6

- In Design and Programmer Information UI, Programmer type is reported as FlashPRO 3 for FlashPRO 6 in SmartDebug main window. This issue will be fixed in an upcoming Libero SoC PolarFire release. The issue does not exist in Device Status.

4.3 Secure Production Programming Solution

Job Manager Limitations

- 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.
- Job Manager crashes when opening an existing Job Manager project from v11.9
- Job Manager v12.4 does not support Job Manager project files created with releases prior to v12.0
- Job Manager does not support PolarFire DAT export
  - PolarFire DAT file bitstream export from Job Manager is not supported in Programming and Debug Tools v12.4.
- 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.

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).

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 fails. To run the ERASE action successfully, make sure a ticket for the VERIFY action is included.

4.4 Installation

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."
```

  Workaround: Uninstall existing FlashPro6 drivers and restart the system before installing Libero SoC v12.4. If the software is already installed, ignore the above message if installation logs do not report any errors.

5 System Requirements

The Programming and Debug Tools v12.4 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 in UG0710 Libero SoC Linux Environment Setup User Guide.

6 Download Libero SoC v12.4 Programming and Debug Tools

Click the following links to download Programming and Debug Tools v12.4 on Windows and Linux operating systems:

- Windows Download
- Linux 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, take appropriate steps to ensure signal integrity of 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 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 program the device successfully at a high TCK frequency, take appropriate to ensure signal integrity of 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

NOTE:

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