Microsemi®

Flash Programming Frequently Asked Questions

November 2015

Table of Contents

General

1. How many ways can flash devices be programmed?

Flash devices can be programmed in the system board or using a programmer prior to installation in the board. Depending on the device family, multiple methods are available for system board level programming.

2. How do I program ProASIC devices?

ProASIC® (A500K) devices can be programmed using FlashPro (discontinued) and the Silicon Sculptor III programmer, using an ISP adapter module after mounting them in the system board. They can also be programmed using the Silicon Sculptor III programmer and a package specific adapter module before mounting in the board. Refer to *AC188: In-System Programming ProASIC Devices* application note for more details.

3. How do I program ProASIC$^{\underline{PLUS}}$ devices?

ProASIC$^{\underline{PLUS}}$® (APA) devices can be programmed using FlashPro (discontinued), FlashPro Lite, and Silicon Sculptor III after mounting them in the system board. To program using the Silicon Sculptor III programmer, a package-specific adapter module is required. Refer to *AC309: In-System Programming ProASIC$^{\underline{PLUS}}$ Devices* application note for more details.

ProASIC$^{\underline{PLUS}}$ devices can be also programmed through a microprocessor or microcontroller using a STAPL player or Microsemi's DirectC method. Refer to *AC208: Performing Microprocessor Programming for Microsemi ProASIC$^{\underline{PLUS}}$ Devices* application note for more details.To program the ProASIC^{PLUS} device through USB, QuickFlash Programmer can be used as alternative. Microsemi has tested the functionality of *QuickFlash Programmer*. However, Microsemi does not support programming or functional failures resulted by using third part programmers

4. How do I program new generation flash devices?

New generation flash devices (such as ProASIC3/E, ProASIC3L, IGLOO®, Fusion, SmartFusion®, IGLOO2, and SmartFusion2) can be programmed in the system board using the FlashPro3 , FlashPro4or FlashPro5 programmers. They can also be programmed using a Silicon Sculptor III programmer and a package-specific adapter module before mounting in the board. They can also be programmed through a microprocessor or microcontroller using Microsemi's DirectC method. For more details, refer to the *DirectC* page.

SmartFusion devices can be programmed by several other methods, which are documented in *AC362: SmartFusion cSoC: Programming FPGA Fabric and eNVM Using In-Application Programming Interface* application note.

SmartFusion2 / IGLOO2 devices can be programmed by several other methods, which are documented in *UG0451: IGLOO2 and SmartFusion2 Programming User Guide*.:

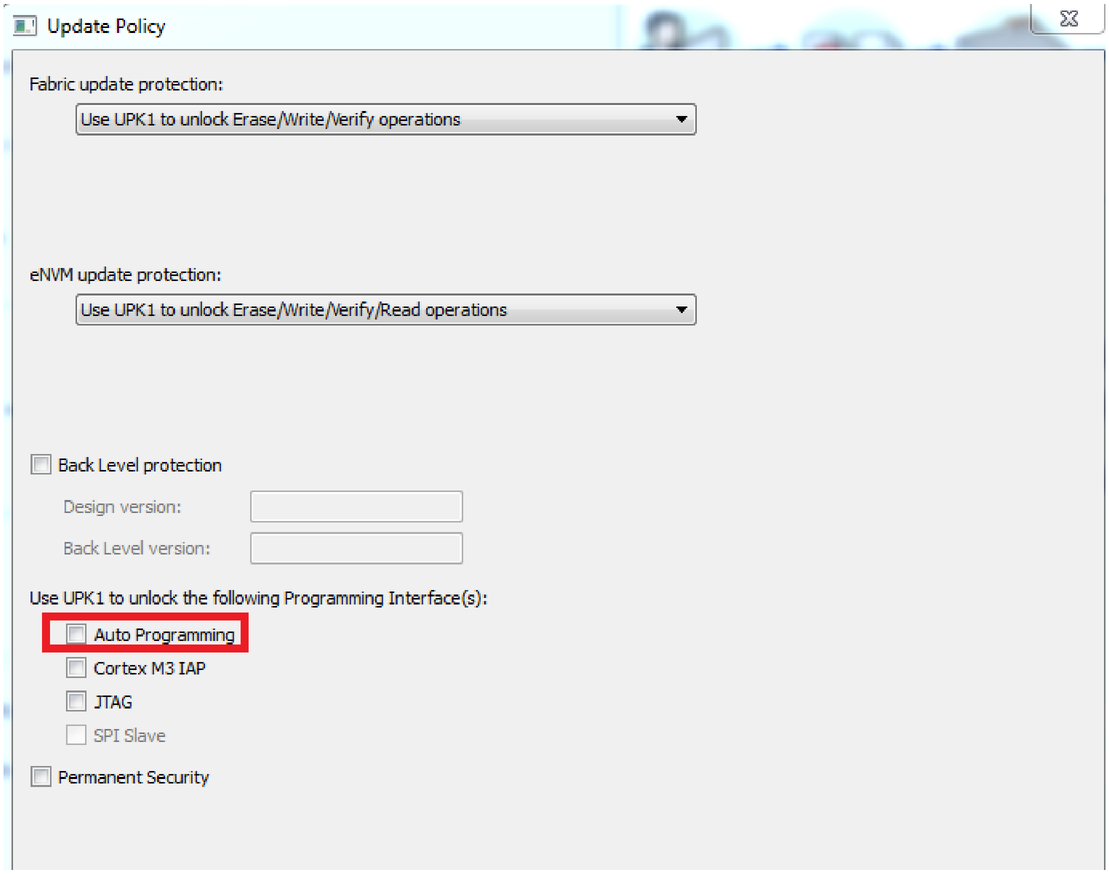
RTG4 devices can programmed by several other methods, which are documented in *UG0602: RTG4 FPGA Programming User Guide*.

5. What is the typical programming time for SmartFusion2/IGLOO2 devices?

Programming times may vary with the Host PC Processor, RAM, OS, Speed, and so on. Refer the *Knowledge Base Article*.

6. Is there any way to permanently disable Auto programming mode in SmartFusion2 / IGLOO2 devices?

By default Auto programming is enabled in the SmartFusion2 / IGLOO2 devices. To disable Auto programming permanently, the user must check **Auto Programming** and **Permanent Security** in Security Policy Manager (SPM) of Libero SoC as shown in Figure 1.



Update Policy

Fabric update protection:

Use UPK1 to unlock Erase/Write/Verify operations

eNVM update protection:

Use UPK1 to unlock Erase/Write/Verify/Read operations

□ Back Level protection

Design version:

Back Level version:

Use UPK1 to unlock the following Programming Interface(s):

□ Auto Programming

□ Cortex M3 IAP

□ JTAG

□ SPI Slave

□ Permanent Security

Figure 1: **Disabling Auto Programming**

Programming Operation

7. How do I know the number of times the device has been programmed?

There is no way to find out how many times ProASIC or ProASIC$^{\underline{PLUS}}$ devices have been programmed. The device will fail programming when it reaches the maximum number of programming cycle (refer to the applicable datasheet). Note that if the Erase operation is interrupted during programming, the write count will be lost and you will not be able to program the device again.

For the newer flash devices (ProASIC3/E, IGLOO, Fusion, SmartFusion, IGLOO2, SmartFusion2 and so on), the DEVICE_INFO command will print the number of programming cycles in the log file. The programming cycle is supported only for the FPGA fabric. The programming cycle information for FlashROM and nonvolatile memory (eNVM) is not available through any programming actions.

However, this is a page write count for each page. This can be read out by using SmartDebug. If the eNVM is corrupted, the write count number will be corrupted as well.

8. Do I need to use the RCK pin while programming the device?

FlashPro and FlashPro Lite programmers do not require the RCK pin for programming. You can tie this pin to GND. Do not leave this pin floating. Due to noise on the RCK pin, the programming may fail to complete. If RCK is provided on-board and you are trying to program with FlashPro/Lite, a STAPL file modification is required to change **USE_RCK = 0** to **USE_RCK = 1**.

RCK is needed for an embedded programming solution such as DirectC programming.

9. What is the state of the flash cells of a blank/virgin device?

The core or fabric of all Flash FPGAs coming out from the fab have their flash cells erased. For ProASIC and ProASIC$^{\underline{PLUS}}$ the erased state is '0' (Off state) and for ProASIC3/E, IGLOO, Fusion, SmartFusion, SmartFusion2, and IGLOO2, the erase state is '1' (On state).

However, during manufacturing flow a special design is programmed into these Flash FPGAs for safe power up (that is, FPGA core is turned off) and then shipped to customers.

10. How is the FPGA programming verified?

Following sections provide information for both older and newer Flash FPGA programming verification:

ProASIC and ProASIC$^{\underline{PLUS}}$ Devices

FPGA Array Verification: During verification, the programming software reads the array data and compares against expected data. The verification is performed one word at a time.

ProASIC/E, IGLOO/e/Plus, Fusion, SmartFusion, SmartFusion2, IGLOO2 Devices

FPGA Array Verification: During verification, the programming software sends a row of comparison data into the device. The device performs the verification by comparing the data received with the data that was programmed into the flash cell. The verification is performed one row at a time. The device returns a single-bit PASS/FAIL status.

11. What is the difference between programming verify and standalone verify?

There are two voltage levels of the flash cell: beginning of life (BOL) and end of life (EOL). When a device is programmed, Microsemi guarantees that each flash cell will have the minimum voltage defined by BOL. If a flash cell fails to program to the BOL minimum voltage, this device is FAILED at programming time (verify failure during programming operation). Over time and temperature, the flash cell voltage will decay to the EOL voltage level. The EOL voltage is used to determine the performance reported in Microsemi delay extraction tools. After the flash cell voltage has deteriorated below the EOL threshold, the performance of the device is no longer guaranteed. The standalone verify is set to the EOL limit. If the device failed during standalone verify, the flash cell fails the EOL limit. If the array is within the programming cycle limit, reprogramming the flash cell would restore the original performance parameter.

12. What will happen to the device if the programming is aborted in the middle of the operation?

If ProASIC or ProASIC$^{\underline{\text{PLUS}}}$ device programming is interrupted, the device is no longer programmable. Tests show that ProASIC3 and newer flash devices can be reprogrammed even if the programming is interrupted.

SmartFusion2 and IGLOO2 devices support Programming Recovery operation. Programming recovery can reprogram the device upon power up if there was any prior interruption. Refer to *UG0451: IGLOO2 and SmartFusion2 Programming User Guide* for more details.

13. How do I modify the TCK frequency?

If the programming is unsuccessful due to suspected signal integrity issues in the board, you can try reducing the TCK frequency during the programming operation. Note that scan chain has a default frequency of 1 MHz, which cannot be modified. For other STPL operations, you can modify the TCK frequency by following the method described in the *Knowledge Base* article.

14. Is the programming data TCK level sensitive?

Yes. Data on TDI is sampled by the device on the rising edge of TCK. Data on TDO is set by the device and guaranteed to be stable on the falling edge of TCK.

15. What are the voltages needed for performing various programming actions?

During scan chain, VCC (core) and VJTAG must be set to the device pins as well as programming header. For all other operations, VPUMP must be turned on.

Fusion and SmartFusion programming requires the supply to additional pins: VCCNVM/VCCENVM and VCCOSC/VCCROSC.

16. Is there any checksum command for FlashROM and nonvolatile memory (NVM)?

No, the checksum is only calculated for FPGA fabric data. Checksum is not supported for FlashROM and NVM.

17. What is the state of the I/Os during programming?

The default I/O state is tristated during programming. For the SmartFusion2 / IGLOO2 devices it is tristate with weak pull up during programming. FlashPro software allows controlling different I/O states during programming. Refer to the *FlashPro User Guide* for details. The I/O state of blank devices is tristate with weak pull-up. The following are the I/O states during various types of programming;

I/O state during JTAG programming

You can define this in Libero SoC (check Online Help for details).

IO state during SPI-Slave programming

If the part is already programmed (i.e. operating), it will use the configured FF state. If the device is blank, it should stay tri-stated with weak pulled up.

I/O state during Auto programming, Auto Update/Recovery

Any boot-initiated programming (i.e. auto-programming, auto-update/recovery) will not first power up the fabric (which would be required to get the user IO configuration) so the IOs will remain in their default state (tristate with pull up). It is applicable for both blank and programmed devices.

2 Step IAP and ISP (ISP is only for SmartFusion2)

During ISP/IAP (the device has to be programmed first with IO state during F*F attribute) the IO will follow F*F state (I/O state during F*F as defined in Libero).

18. Is parallel programming supported?

Yes, the newer flash family devices can be programmed in parallel mode. FlashPro software supports a single GUI to drive multiple FlashPro3/4/5 programmers for parallel programming. In a Microsemi test, parallel programming is approximately 5 times faster than programming 16 devices sequentially.

19. Is JTAG chain programming supported?

Yes. Microsemi SoC flash devices can be chained together with the same or different part number and devices from other vendors. Note that while programming Microsemi flash device along with some JTAG non-compliant devices in a JTAG chain, you need to enable **discrete TCK toggling** (**Tools** > **Programmer setting** > **FlashPro4**). You must use the FlashPro4 programmer or FlashPro5 programmer. The FlashPro3 programmer does not support discrete toggling.

20. Is it possible to read the silicon signature from the FPGA fabric?

No. The FPGA fabric has no access to this functionality. You need to read back the silicon signature from the JTAG ports.

21. Can I read the checksum if the device is permanently locked?

Yes. You can run DEVICE_INFO and find the checksum information in the log file.

22. Can I perform boundary scan testing if the device is permanently locked?

Yes. You can still run boundary scan testing if the device is permanently locked.

23. What is the correct voltage of JTAG_SEL for SmartFusion device?

The correct voltage value is 3.3 V. For more details refer to the *Knowledge Base* article.

24. What are the states of the FPGA and MSS I/O during programming of the FPGA fabric?

The FPGA and MSS I/O states vary pending on programming method used. Refer to the *Knowledge Base* article for more details.

25. What is the I/O state of an unprogrammed device during boundary scan?

While doing boundary scan testing of a programmed device, the I/Os are defined per design (use design-specific BSDL). I/Os of an unprogrammed device have default LVTTL 3.3V, lowest drive strength, slow slew, and weak pull-up enabled (where applicable). This default state can be overridden by IOCONFIG instruction.

26. How many times can Standalone Verify or Digest be performed in the IGLOO2 or SmartFusion2 device?

It can be performed for 2000 cycles at 100C for Digest and for standalone Verify

27. For the IGLOO2 or SmartFusion2 device, does a Verify cycle need to be considered same as Program?

Verify does not introduce the same level of stress as programming, therefore we allow more verify cycles than program. Verify cycles may be performed in addition to the programming cycles.

28. For the IGLOO2 or SmartFusion2 device, is there a built-in counter that records and checks if Verify and Digest cycles exceed?

There is no counter to track the number of verify cycles and digest cycles on the device. It is user's responsibility to keep record of the digest and verify cycles.

29. How long does computing each digest take in the IGLOO2 or SmartFusion2 device?

Digest time (Fabric + eNVM) is 3.2 seconds, in typical conditions. This data point was collected on very few samples, and this is not a specification.

Programming Software

30. Is the FlashPro software supported in both the Windows® and Linux® platforms?

The FlashPro software is supported in all Windows platforms. Refer to the *Libero SoC* page for more details.

FlashPro Express software along with the FlashPro5 programmer can be used on supported Linux Platforms.

31. Is there any character limitation for project path or file name in the FlashPro software?

Yes. The total number of characters in the project path name should be less than 256.

32. Can I change the programming configuration inside the FlashPro software?

Yes, but only for IGLOO series, ProASIC3 series, Fusion, and SmartFusion devices. If you load the PDB file in the FlashPro software, you can modify the security setting and I/O state during programming, eNVM, and FlashROM configuration. Refer to the *FlashPro User Guide* for details.

33. What are the types of programming files the Microsemi SoC flash programmer can support?

The FlashPro3/4/5 programmers support STP and PDB files, whereas the FlashPro Lite and FlashPro programmers (discontinued) support only STP files. The Silicon Sculptor 3 programmer supports bit and STP files for ProASIC and ProASIC$^{\underline{\text{PLUS}}}$ devices and STP files only for ProASIC3/E and newer flash devices.

34. Does the Microsemi SoC Flash programmer support SVF files?

Although SVF is an industry standard file format, Microsemi programmer does not support this file type. Most third party JTAG test tools support SVF file programming. The SVF files generated by FlashPro software are tested with the ASSET InterTech JTAG tester.

35. Can FlashPro software generate any programming files?

FlashPro software can export IEEE 1532 files (BSD, ISC), DirectC DAT files (DAT), STP, and SVF files. The file type supported depends on the device type. Refer to the *FlashPro User Guide* for details.

36. What is the way to clear ECC2 errors in eNVM pages of Fusion and SmartFusion devices?

Execute the TCL command "recover_flash_memory" to remove the ECC2 errors that are caused due to memory corruption by reprogramming specified flash memory (NVM) pages to zeros.

37. What are the types of programming files supported for SmartFusion2 devices?

Refer to the *Knowledge Base* article to access the list of programming files supported for SmartFusion2 devices.

38. What are the types of programming files supported for IGLOO2 devices?

Refer to the *Knowledge Base* article to access the list of programming files supported for IGLOO2 devices:

39. What is the programming bitstream size for SmartFusion2 / IGLOO2 devices?

Refer to the *Knowledge Base* article which provides the programming bitstream size list supported for SmartaFusion2 / IGLOO2 devices.

40. Do I need the original programming file to read back the checksum of the design programmed into the device?

FlashPro software reads back the checksum from a programmed device and you do not need to load the original programming file to read the checksum. Any programming file that is targeted to same device would read back the checksum programmed in the part. This checksum is not calculated by the programmer rather it is read from UROW registers within the device.

Programming Hardware

41. What are the differences between the various FlashPro programmers?

Refer to the *Knowledge Base* article.

42. Does FlashPro Lite work in a USB port using a converter?

No. You can use a PCMCIA to parallel port converter or PCIe to parallel port converter if your computer or laptop does not have a parallel port. Microsemi has tested only a few such converters. Refer to *AC351: PCMCIA Adapter for Parallel Port Programming application brief* for more details.

43. What is the operating temperature limit for the FlashPro4 programmer?

Microsemi does not provide operating temperature information for the FlashPro programmers. Since it is assumed to be used as lab equipment or production equipment, Microsemi does not test at extended temperature. It is confirmed that all the devices in the unit are commercial temperature, so should operate within that range. The FlashPro4 went through a burn-in cycle operating at 100°C for 250 hours during quality testing. This involved only powering the programmers repeatedly; after burn-in they are used for programming. No programming was done during the burn-in cycles.

44. Is there any limitation in the length of the programmer cable?

Microsemi has done testing with a long cable. Cables up to 36 inches long were tested and the signal integrity issue began causing problems at 18 inches and greater.

45. Is the FlashPro programmer nonvolatile?

No. However, nonvolatile ICs are used in FlashPro programmers. Refer to the Customer Notification *CL: Flash Pro5 Statement of Volatility*. Refer to the *FlashPro* webpage for more information on flash programming.

46. Is the ASSET programmer supported for programming Microsemi SoC flash devices?

ASSET programming software (latest version) can interpret Microsemi STAPL files. Microsemi has tested ProASIC3/E, IGLOO/e/Plus, Fusion, and SmartFusion devices with SVF, IEEE 1532, STAPL and SmartFusion2, IGLOO2 devices with only STAPL on ASSET programmer. Although Microsemi does not support these programmers officially, there should not be any issue with using them to program Microsemi flash devices.

47. Can I program the SmartFusion device using the FlashPro3 or LCPS?

Yes, FlashPro3 can be used to program SmartFusion, SmartFusion2/IGLOO2 devices but it has been discontinued (only FlashPro4 and FlashPro5 are available now). LCPS (low programming stick) can also be used with some modification to the board. For further details, refer to the *Knowledge Base* article.

48. Can I use FlashPro3 and FlashPro both in the same PC?

No, you can connect either FlashPro3/4/5 or FlashPro but not both. refer to the *Knowledge Base* article.

49. Can I use the QuickFlash Programmer from Embeddetech™ to program my flash devices?

QuickFlash is compatible with the original 26-pin and 10-pin FlashPro connectors and supports in-system JTAG-based programming for all of the ProASIC^{PLUS} (APA), ProASIC-3 (A3P), IGLOO (AGL), Fusion (AFS), and SmartFusion (A2F) device families. Microsemi has tested the functionality of *QuickFlash Programmer*. However, Microsemi does not support programming or functional failures resulted by using third part programmers.

List of Changes

The following table shows the important changes made in this document for each revision.

Revision	Changes	Page
Revision 2 (October 2015)	Updated answer for Question 45 (SAR69204). Added Question 49, Question 26, Question 27, Question 28, Question 29, Question 40. (SAR 72461) Edited Question 3, Question 4, Question 5, Question 15, Question 17, Question 30. (SAR 72461)	10, 8, 9
Revision 1 (October 2013)	Added Question 5, Question 6, Question 23, Question 24, Question 25, Question 36, Question 37, Question 38, Question 39, Question 47, Question 48 (SAR49351).	4, 5, 8, 9, 10
	Updated answers for Question 4, Question 9, Question 10, Question 12, Question 15, Question 18, Question 45 (SAR49351).	4, 6, 7, 10

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA

Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996

E-mail: sales.support@microsemi.com

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for communications, defense & security, aerospace 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; 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, Calif., and has approximately 3,600 employees globally. Learn more at **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.