FlashPoint User's Guide

Actel Corporation, Mountain View, CA 94043

© 2004 Actel Corporation. All rights reserved.

Printed in the United States of America

Part Number: 50200035-0

Release: October 2004

No part of this document may be copied or reproduced in any form or by any means without prior written consent of Actel.

Actel makes no warranties with respect to this documentation and disclaims any implied warranties of merchantability or fitness for a particular purpose. Information in this document is subject to change without notice. Actel assumes no responsibility for any errors that may appear in this document.

This document contains confidential proprietary information that is not to be disclosed to any unauthorized person without prior written consent of Actel Corporation.

Trademarks

Actel and the Actel logotype are registered trademarks of Actel Corporation.

Adobe and Acrobat Reader are registered trademarks of Adobe Systems, Inc.

Mentor Graphics, Precision RTL, Exemplar Spectrum, and Leonoardo Spectrum are registered trademarks of Mentor Graphics, Inc.

WaveFormerLite is a registered trademark of SynaptiCAD, Inc.

Synplify is a registered trademark of Synplicity, Inc.

Sun and Sun Workstation, SunOS, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc

Synopsys is a registered trademark of Synopsys, Inc.

Verilog is a registered trademark of Open Verilog International.

Viewlogic, ViewSim, ViewDraw and SpeedWave are trademarks or registered trademarks of Viewlogic Systems, Inc.

Windows is a registered trademark and Windows NT is a trademark of

Microsoft Corporation in the U.S. and other countries.

UNIX is a registered trademark of X/Open Company Limited.

All other products or brand names mentioned are trademarks or registered trademarks of their respective holders.



Table of Contents

Generate a programming file
Silicon signature
Programming security settings
Custom security levels
Programming the FlashROM
Custom serialization data for FlashROM region11
Custom serialization data file format
Syntax14
Semantics
Hex serialization data file example13
Binary serialization data file example14
Decimal serialization data file example14
Text serialization data file example14
Programming the FPGA Array
Reprogramming a secured device

Table of Contents



Generate a programming file

Generate a programming file

FlashPoint enables you to program security settings, FPGA Array, and FlashROM features for ProASIC3/E devices. You can program these features separately using different programming files or you can combine them into one programming file. Each feature is listed as a silicon feature in the GUI.

You can generate a programming file with one, two, or all of the silicon features from the Generate Programming File page.

To generate a programming file:

1. Enter the **Output file name**.

Click the **Browse** button if you need to find your file or select your directory, and then enter the file name to save your output file.

2. Select the **Silicon feature**(s) you want to program.

Security settings FPGA Array FlashROM

./fip4.stp			Browse	
Silicon feature(s) to be	programmed			
Security settin	gs			
🔽 FPGA Array				
🔽 FlashROM				
FlashROM co	nfiguration file:			
D:\testarea\	designs\g3_test\from.ufc	2	 Browse	
	usly secured device(s)			
Silicon signature (max l				
123				
1275				

3. Click the **Programming previously secured device(s)** check box if you are reprogramming a device that has been secured.

Because the ProASIC3/E family enables you to program the Security Settings separately from the FPGA Array and/or FlashROM, you must indicate if the Security Settings were previously programmed into the target device. This requirement also applies when you generate programming files for reprogramming.

- 4. Enter the silicon signature (0-8 HEX characters). See <u>Silicon Signature</u> for more information.
- 5. Click Next.

Silicon signature

With Designer tools, you can use the silicon signature to identify and track Actel designs and devices. When you generate a programming file, you can specify a unique silicon signature to program into the device. This signature is stored in the design database and in the programming file, and programmed into the device during programming.

The silicon signature is accessible through the USERCODE JTAG instruction.

NOTE: If you set the security level to high, medium, or custom, you must program the silicon signature along with the Security Setting. If you have already programmed the Security Setting into the target device, you cannot reprogram the silicon signature without reprogramming the Security Setting.

The previously programmed silicon signature will be erased if:

- You have already programmed the silicon signature and
- You are programming the security settings, but you do not have an entry in the silicon signature field

Programming security settings

FlashPoint allows you to set a security level of high, medium or none.

To program Security Settings on the device:

1. If you choose to program Security Settings on the device from the **Generate Programming File** page, the wizard takes you to the **Security Settings** page (see figure below).



Programming security settings

Medium	Pass Key - Use AES key to write or verify f - Use AES key to write the Flash		
None ass Key (max length is 32 H C8E6A1BA6AE8A34CCE		Default Level Generate random key	
ES Key (max length is 32 HI	EX chars): 18742BBB96B89	Generate random key	
and a second second second		Generate random key	

2. Move the sliding bar to select the security level for FPGA and FlashROM (see table for a description of the security levels).

Security Level	Security Option	Description
High	Protect with a 128-bit Advanced Encryption Standard (AES) key and a Pass Key	Access to the device is protected by an AES Key and the Pass Key.
	1 455 1009	The Write and Verify operations of the FPGA Array use a 128-bit AES encrypted bitstream.
		From the JTAG interface, the Write and Verify operations of the FlashROM use a 128- bit AES encrypted bitstream. Read back of the FlashROM content via the JTAG interface is protected by the Pass Key.
		Read back of the FlashROM content is allowed from the FPGA Array.
Medium	Protect with Pass Key	The Write and Verify operations of the FPGA Array require a Pass Key.
		From the JTAG interface, the Read and Write operations on the FlashROM content require a Pass Key.

Custom security levels

Security Level	Security Option	Description
		You can Verify the FlashROM content via the JTAG interface without a Pass Key. Read back of the FlashROM content is allowed from the FPGA Array.
None	No security	The Write and Verify operations of the FPGA Array do not require keys. The Read, Write, and Verify operations of the FlashROM content also do not require keys.

3. Enter the Pass Key and/ or the AES Key as appropriate. You can generate a random key by clicking the Generate random key button.

The Pass Key protects all the Security Settings for the FPGA Array and/or FlashROM.

The **AES Key** decrypts FPGA Array and/or FlashROM programming file content. Use the AES Key if you intend to program the device at an unsecured site or if you plan to update the design at a remote site in the future.

You can also customize the security levels by clicking the **Custom Level** button. For more information, see the <u>Custom Security</u> <u>Levels</u> section.

Custom security levels

For advanced use, you can customize your security levels.

To set custom security levels:

1. Click the **Custom Level** button in the **Setup Security** page. The **Custom Security** dialog box appears (see figure below).

Custom Security Level		X
Security of the FPGA Array Lock for both writing and verifying Lock for writing Use AES key for both writing and verifying Allow write and verify	Security of the FlashROM C Lock for both reading and writing Lock for writing C Use AES key for both writing and verifying Allow reading, writing, and verifying	
Permanently lock the security settings.	Cancel Help	



Custom security levels

2. Select the FPGA Array Security and the FlashROM Security levels.

The FPGA Array and the FlashROM can have different Security Settings. See the tables below for a description of the custom security option levels for FPGA Array and FlashROM.

FPGA Array

Security Option	Description
Lock for both writing and verifying	Allows writing/erasing and verification of the FPGA Array via the JTAG interface only with a valid Pass Key.
Lock for writing	Allows the writing/erasing of the FPGA Array only with a valid Pass Key. Verification is allowed without a valid Pass Key.
Use the AES Key for both writing and verifying	Allows the writing/erasing and verification of the FPGA Array only with a valid AES Key via the JTAG interface.
	This configures the device to accept an encrypted bitstream for reprogramming and verification of the FPGA Array.
	Use this option if you intend to complete final programming at an unsecured site or if you plan to update the design at a remote site in the future. Accessing the device security settings requires a valid Pass Key.
Allow write and verify	Allows writing/erasing and verification of the FPGA Array with plain text bitstream and without requiring a Pass Key or an AES Key. Use this option when you develop your product in-house.

Note: The ProASIC3/E family FPGA Array is always read protected regardless of the Pass Key or the AES Key protection. FlashROM

Security Option	Description
Lock for both reading and writing	Allows the writing/erasing and reading of the FlashROM via the JTAG interface only with a valid Pass Key. Verification is allowed without a valid Pass Key.
Lock for writing	Allows the writing/erasing of the FlashROM via the JTAG interface only with a valid Pass Key. Reading and verification is allowed without a valid Pass Key.
Use the AES Key for both writing and verifying	Allows the writing/erasing and verification of the FlashROM via the JTAG interface only with a valid AES Key. This configures the device to accept an encrypted bitstream for reprogramming and verification of the FlashROM. Use this option if you complete final programming at an unsecured site or if you plan to update the design at a remote site in the future. Note: The bitstream that is read back from the FlashROM is always unencrypted (plain text).
Allow writing and verifying	Allows writing/erasing, reading and verification of the FlashROM content with a plain text bitstream and without requiring a valid Pass Key or an AES Key.

Note: The FPGA Array can always read the FlashROM content regardless of these Security Settings.

3. To make the Security Settings permanent, select the **Permanently lock the security settings** check box. This option prevents any future modifications of the Security Setting of the device. A Pass Key is not required if you use this option.

Note: When you make the Security Settings permanent, you can never reprogram the <u>Silicon Signature</u>. If you Lock the write

operation for the FPGA Array or the FlashROM, you can never reprogram the FPGA Array or the FlashROM, respectively. If you use an AES key, this key cannot be changed once you permanently lock the device.

To use the Permanent FlashLockTM feature, select Disable Write and Verify for **FPGA Array** and Disable Read, Write and Verify for **FlashROM** and select the **Permanently lock the security settings** checkbox as shown in the figure below. This will make your device one-time-programmable.

ecurity of the FPGA Array	Security of the FlashROM
Lock for both writing and verifying	Cock for both reading and writing
Lock for writing	C Lock for writing
Use AES key for both writing and verifying	Use AES key for both writing and verifying
Allow write and verify	C Allow reading, writing, and verifying
Permanently lock the security settings. The following silicon features will not be - Security settings. AES key, and silicon - FPGA Array - FlashROM	

4. Click the **OK** button.

The Security Settings page appears with the Custom security setting information as shown in the figure below.



Programming the FlashROM

Programming the FlashROM

You can program selected memory pages and specify the region values of the FlashROM.

To program FlashROM:

- 1. Select **FlashROM** from the **Generate Programming File** page.
- 2. Enter the location of the FlashROM configuration file. The **FlashROM Settings** page appears (see figure below).

Programming the FlashROM

	FlashRC	IM re	gions:	1					_								Region_7_8	
ogram	words	15	14 1	3 12	11	10	9	8	7	6	5	4	3	2	1	0	Properties:	
age	pages							and the second second	-		-				_		Name	Region_7_8
	7																Start page	7
-	6								Argument of		-						Start word	8
	-	1000	00000.00	-			0000	11100			10000	0000	2000				Length Content	8 Auto Inc
	5																Start value	Auto inc
Ē	4				1		-			-							(HEX)	0
	-	_	-	-	-		-	-		-	-	-	_	_	-	_	Step value (HEX)	1
-	3																Max value (HEX)	1000
-	2																	
	1		Contraction of the															
Ξ.	0																	
	OM progr Single pr		- Et S	533	ralld	evice	s				1	0	One	prog	ramm	ingl	, file per device	
lumber	of device:	s to p	rogran	c			1								Т	arge	et Programmer	

- 3. Select the FlashROM memory page that you want to program.
- 4. Enter the data value for the configured regions.
- 5. If you selected the region with a **Read From File**, specify the file location. See <u>Custom Serialization Data for FlashROM Region</u> for more information.
- 6. If you selected the Auto Increment region, specify the Start and Max values.
- 7. Complete steps 8 and 9 if you have a **Read from file** and/or **Auto Increment** region in the FlashROM.
- 8. Select the type of FlashROM programming files you want to generate from the two options below:

- Single programming file for all devices option: generates one programming file with all the generated increment values or with values in the custom serialization file.

- One programming file per device: generates one programming file for each generated increment value or for each value in the custom serialization file.

- 9. Enter the number of devices you want to program.
- 10. Click the Target Programmer button.

The Select Programmer Type dialog box appears (see figure below).



Custom serialization data for FlashROM region

Select Programmer Type	
Programmer types	
C Generic STAPL player	
Silicon Sculptor II, BP Auto Programmer or FlashPro3	
OK Cancel Help	

- 11. Select your target Programmer type.
- 12. Click OK.

FlashPoint generates your programming file.

Note: You cannot change the FlashROM region configuration from FlashPoint. You can only change the configuration from the ACTgen FlashROM core generator.

For more information, see ACTgen online help.

Custom serialization data for FlashROM region

FlashPoint enables you to specify a custom serialization file as a source to provide content for programming into a **Read from file FlashROM** region. You can use this feature for serializing the target device with a custom serialization scheme.

To specify a FlashROM region:

1. From the **Properties** section in the **FlashROM Settings** page, select the file name of the custom serialization file (see figure below). For more information on custom serialization files, see <u>Custom Serialization Data File Format</u>.

Custom serialization data file format



- 2. Select the FlashROM programming file type you want to generate from the two options below:
 - Single programming file for all devices option: generates one programming file with all the values in the custom serialization file.
 - One programming file per device: generates one programming file for each value in the custom serialization file.
- 3. Enter the number of devices you want to program.
- 4. Click the Target Programmer button.
- 5. Select your target **Programmer type**.
- 6. Click OK.

Custom serialization data file format

FlashPoint supports custom serialization data files that specify the data in binary, HEX, decimal, or ASCII text. The custom serialization data files may contain multiple data with the Line Feed (LF) character as the delimiter.

You can create a file by entering serialization data into any type of text editor. Depending on the serialization data format (hex, ASCII, binary, decimal), input the serialization data according to the size of the region you specified in the FlashROM settings page.



Semantics

Each custom serialization file has only one type of data format (binary, decimal, Hex or ASCII text). For example, if a file contains two different data formats (i.e. binary and decimal) it is considered an invalid file.

The length of each data must be shorter or equal to the selected region length. If the data is shorter then the selected region length, the most significant bits shall be padded with 0's. If the specified region length is longer than the selected region length, it is considered an invalid file.

The digit / character length is as follows:

-Binary digit: 1 bit.

-Decimal digit: 4 bits.

-Hex digit: 4 bits.

-ASCII Character: 8 bits.

Standard Example

If you wanted to use, for example, device serialization for three devices with serialization data 123, 321, and 456, you would create file name from_read.txt. (See figure below). Each line in from_read.txt corresponds to the serialization data that will be programmed on each device. For example, the first line corresponds to the first device to be programmed, the second line corresponds to the second device to be programmed, and so on.

123	
321	
456	

Hex serialization data file example

The following example is a Hex serialization data file for a 40-bit region:

```
123AEd210
```

AeB1

0001242E

Note: If you enter an invalid Hex digit such as 235SedF1, an error occurs. An error also occurs if you enter data that is out of range, i.e. 4300124EFE.

The following is an example of programming "AeB1" into Region_7_1 located on page 7, from Word 5 to Word 1 in the **FlashROM** settings page. See <u>Custom Serialization Data for FlashROM Region</u> for more information.

	Word 15	 Word 6	Word 5	Word 4	Word 3	Word 2	Word 1	Word 0
Page 7		 	00	00	00	AE	B1	

Custom serialization data file format

Binary serialization data file example

The following example is a binary serialization data file for a 16-bit region:

1100110011010001 100110011010011 11001100110101111 (This is an error: data out of range) 1001100110110111 1001100110110112 (This is an error: invalid binary digit)

Decimal serialization data file example

The following example is a decimal serialization data file for a 16-bit region:

65534

65535 65536 (This is an error: data out of range) 6553A (This is an error: invalid decimal digit)

Text serialization data file example

The following example is a text serialization data file for a 32-bit region: AESB A)e ASE3 23 (This is an error: data out of range) 65A~ 1234 AEbF

Syntax

Custom serialization data file = <hex region data list> | <decimal region data list> |
 <br

Hex region data list = < hex data> <new line> { < hex data> <new line> }
Decimal region data list = <decimal data> <new line> {<decimal data><new line> }
Binary region data list = <binary data> <new line> { <binary data> <new line> }

ASCII text region data list = < ascii text data> <new line>

{ < ascii text data> <new line> }

hex data = <hex digit> {<hex digit>}



Programming the FPGA Array

```
decimal data = < decimal digit> {< decimal digit>}
binary data = < binary digit> {< binary digit>}
ASCII text data = <ascii character> {< ascii character >}
new line = LF
binary digit = `0' |`1'
decimal digit = `0' |`1'|`2' |`3' |`4' |`5' |`6' |`7' |`8' | `9'
hex digit = `0' |`1' |`2' |`3' |`4' |`5' |`6' |`7' |`8' |`9' |`A' |`B' |`C' |`D' | `E' | `F' |
`a' | `b' | `c' | `d' | `e' | `f'
```

ascii character = characters from SP(0x20) to'~'(0x7E).

Programming the FPGA Array

You can program the FPGA Array by selecting the silicon feature, **FPGA Array** in the **Generate Programming File** page and clicking **OK**. See <u>Generate a Programming File</u> for more information.

Reprogramming a secured device

You must know the previous Security Settings of the device before you can reprogram a device with Security Settings.

To program a secured device:

1. In the Generate Programming File page, click the Programming previously secured devices(s) check box (see figure below).

Reprogramming a secured device

licon f	feature(s) to be programmed:		
Г	Security settings		
1	FPGA Array		
Г	FlashROM		
	FlashPGM configuration file:		
	D:\testarea\designs\g3_test\from ufc	Browse	
Prog	gramming previously secured device(s)		
	signature (max length is 8 HEX chans)		
Г			
6	i) Salart Saruth satings above to generate silings signal ve		
-			
icon s		_	

- 2. Specify the previously programmed security setting for the FlashROM and/or the FPGA Array.
- 3. If you programmed the device with a custom security level, click the **Custom Level** button to open the **Custom security** dialog box, and select the **Security Settings** for the FPGA Array or the FlashROM that you programmed (see figure below).



Reprogramming a secured device

Custom security settings	Security settings for FPGA Array: - Lock the FPGA Array for both writing - Use the Pass Key to write or verify. Security settings for FlashROM: - Lock the FlashROM for writing via the - Use the Pass Key to write		
Pass Key (max length is 32 HEX chars	Custom Level.	Default Level	
C8E6A1BA6AE8A34CCE14E48537	he one this device.	Generate landom key	
r		Generale rendom key	

- 4. Enter the previously programmed Pass Key and/or the AES Key.
- 5. Click Finish.

Note: Enter the AES Key only if you want to perform encrypted programming.