

Using the Serial FlashLoader With the Quartus II Software

July 2006, ver. 3.0

Application Note 370

Introduction

Using the Joint Test Action Group (JTAG) interface, the Altera® Serial FlashLoader (SFL) is the first in-system programming solution for Altera serial configuration devices. The SFL is available with the Quartus® II software, version 4.1 SP1 and higher. SFL Megafunction is available with Quartus II version 6.0 and higher.

Because serial configuration devices do not support the JTAG interface, the conventional method to program them is via the active serial (AS) programming interface. With the AS programming interface, the configuration data used to program serial configuration devices is downloaded via programming hardware.

However, with the SFL you can program serial configuration devices in-system via the JTAG interface. To do so, use an FPGA as a bridge between the JTAG interface and the serial configuration device. Figure 1 shows both the conventional method of programming serial configuration devices as well as the in-system programming method using the SFL solution. Table 1 lists the advantages and disadvantages of both methods.

Table 1. Advan	tages & Disadvantages	
Method	Advantage	Disadvantage
Conventional: (Active Serial Programming)	Simple and fast	Requires separate programming interface to configure FPGAs and program serial configuration devices.
SFL solution: (JTAG Programming)	Able to configure the FPGA and program serial configuration devices using the same JTAG interface	Slow because the SFL solution needs to configure the FPGA before programming serial configuration devices.

SFL supports FPGA families that configure using active serial configuration scheme. With SFL Megafunction, you can instantiate SFL image into user design. This feature allows you to do SFL programming without resetting the user design in the FPGA. The SFL solution provides more hardware programming options. For example, you can use the ByteBlaster[™] II or USB-Blaster[™] download cable, production tester, and other tools that have a JTAG interface.

Whenever the term "serial configuration device(s)" is used in this document, it refers to Altera EPCS1, EPCS4, EPCS16, and EPCS64 devices.

This application note describes the following:

- Programming single and multiple serial configuration devices with the SFL solution
- Generating programming files in Quartus II software
- Using SFL Megafunction in Quartus II software
- Programming serial configuration devices with JIC or Jam[™] files

Figure 1. Conventional Versus the In-System Programming Method





Steps for Programming Single & Multiple Serial Configuration Devices With the SFL Solution

This section describes the three steps to program both single and multiple serial configuration devices with the SFL solution.

To program serial configuration devices using the SFL solution, follow these steps (see Figures 2, 3, and 4):

- 1. To bridge the JTAG interface with the active serial memory interface (ASMI) block in the FPGA device, configure the SFL image into the FPGA. The previous design will be replaced with the SFL image.
- 2. Program the serial configuration device(s) via the SFL image's JTAG-ASMI bridge.
- You can bypass this step if the SFL image existed in the FPGA.
- 3. Reconfigure the FPGA with the new configuration data. The SFL image will be replaced with the new design. To reconfigure the FPGA with the new configuration data, pull the nConfig pin low and release it to start configuration.



Figure 2. Serial Flash Loader Programming Flow



Figure 3. Programming a Single Serial Configuration Device With the SFL Solution

Note to Figure 3:

(1) You can bypass this step if the SFL image existed in the FPGA.

Figure 4 shows the process for programming multiple serial configuration devices with the SFL solution.



Figure 4. Programming Multiple Serial Configuration Devices with the SFL Solution

Notes to Figure 4:

- (1) "1st device" and "2nd device" represent serial configuration devices.
- (2) You can bypass this step if the SFL image existed in the FPGA.

Using the Serial Flash Loader Megafunction in Quartus II Software

SFL Megafunction allows you to instantiate the SFL image into your design. This feature allows SFL programming without resetting your design in the FPGA with the SFL image.

Instantiating SFL Megafunction in the Quartus II Software

Perform the following steps to generate a SFL Megafunction instantiation. You should then instantiate the SFL Megafunction in your FPGA top-level design.

- 1. Choose MegaWizard Plug-In Manager (Tools menu).
- 2. Select Create a new custom megafunction variation and click Next.
- 3. Select the FPGA device family.
- 4. Select Serial Flash Loader from the Megafunction list.
- 5. Select the Hardware Description Language (HDL) output file type and name the file. Click **Next** (Verilog HDL was chosen for this example). After making these settings, the dialog box appears, as shown in Figure 5.



/hich megafunction would you like to customize?	Which device family will you be using?	Cyclone	•
 Installed Plug-Ins Altera SOPC Builder Arithmetic Gates I/O Memory Compiler 	Which type of output file do you want AHDL YHDL Verilog HDL What name do you want for the output	to create? ut file?	Browse
ELEO	D:\work\Quart2\sfl mf\sfl mf 1c20	vsfl	
FIFU partitioner Flash Memory RAM: 1-PORT RAM: 2-PORT RAM: 3-PORT ROM: 1-PORT ROM: 2-PORT ROM: 2-PORT Parallel Flash Loader SignalTap II Logic Analyzer	Return to this page for another or Note: To compile a project successfu your design files must be in the project libraries specified in the Uptions dialog library specified in the User Libraries p box (Assignments menu). Your current user library directories and the set of the set o	eate operation Illy in the Quartus t directory, in the g box (Tools men page of the Settin re:	II software, global user u), or a user gs dialog
 	gafunction		

6. Specify the directory and output filename. Click **Next**, as shown in Figure 6.

Figure 6. SFL Megafunction Parameter Settings

MegaWizard	Plug-In Manager [page 3 of 5]	× X
1 Parameter	Serial Flash Loader Version 6.0	About Documentation
X-noe_N	s1	Currently selected <u>d</u> evice family: Cyclone
		Cancel < Back Next > Einish

7. Check the **Share ASMI interface with your design** check box if you need to share the ASMI interface with your design. This option provides additional control pins for controlling the ASMI interface, as shown in Figure 7.

Figure 7. SFL Megafunction with "Share ASMI interface with your design" Option

MegaWizard	Plug-In Manager [page 3 of 5]		
さ	Serial Flash Loader Version 6.0	About Document	tation
1 Parameter Settings	2 Simulation 3 Summary Library Page		
	sfl	Currently selected <u>d</u> evice family: Cyclone	Y
CLK_IN	L DATA0_OUT → N ASMI_ACCESS_REQUEST → CCESS_GRANTED	☑ Share ASMI interface with your design	
	-	Cancel < <u>B</u> ack <u>N</u> ext >	jinish

- 8. Click **Next** until you reach the summary page.
- 9. Click **Finish** to generate the SFL Megafunction. Quartus II software generates the megafunction in the form of the HDL file you specified.

Table 2. Input and Output Signa	als for the SFL	Megafunction (Part 1 of 2)
DCLK_IN (1)	Input	Clock signal from user design to DCLK.
nCSO_IN (1)	Input	Control signal from user design to nCSO pin. A low signal enables the EPCS.
ASDO_IN (1)	Input	Control signal from user design to ASDO pin for sending data into EPCS.
nOE_IN	Input	Control signal to enable the SFL Megafunction. A low signal enables the Megafunction. SFL tri-states ASMI interface when it is disabled.

Table 2. Input and Output Signa	als for the SF	L Megafunction (Part 2 of 2)
ASMI_ACCESS_GRANTED (1)	Input	Control signal to allow SFL to access the DCLK, nCSO, ADSO and DATA0 pins using the ASMI interface. A high signal allows SFL to access the ASMI interface. A low signal allows user design to access to ASMI interface.
DATA0_OUT (1)	Output	Signal from DATA0 pin to user design.
ASMI_ACCESS_REQUEST (1)	Output	A high signal indicates SFL is requesting ASMI interface access. SFL starts accessing ASMI interface when ASMI_ACCESS_GRANTED is high.

Note for Table 2:

(1) These ports are available when "Share ASMI interface with your design" option is selected in the megafunction.

Generating Programming Files in the Quartus II Software

You can program serial configuration devices with either JTAG indirect configuration (.jic) or Jam (an ASCII text file in the STAPL format) programming files. To generate JIC or Jam programming files with the Quartus II software, you first need to generate a user-specified SRAM object file (.sof), which is the input file. Next, you need to convert the SOF to a JIC file. Alternatively, if you prefer to use Jam programming files, you will need to convert the JIC file to a Jam file. This section provides the instructions and describes:

- Converting SOF to JIC files
- Converting JIC files to Jam files
- Reviewing JIC and Jam file contents

Converting SOF to JIC Files in Quartus II Software

To convert a SOF to a JIC file, follow these steps:

- 1. Choose Convert Programming Files (File menu).
- 2. In the **Convert Programming Files** dialog box, scroll to the **JTAG Indirect Configuration File** (.jic) from the **Programming file type** field.
- 3. In the **Configuration device** field, specify the targeted serial configuration device.

- 4. In the **File name** field, browse to the target directory and specify an output file name.
- 5. Highlight the SOF data in the **Input files to convert** section. See Figure 8.

Figure 8. Convert Programming Files Dialog Box

Upen Cony	ersion Setup Data		<u>S</u> ave Co	nversion Setup
Output programming file				
Programming file type:	JTAG Indirect Configuratio	n File (.jic)		
Options	Configuration device: E	PCS4	Mode:	Active Serial Configuration
File name:	D:/temp/output_file.jic		,	
	Remote/Local update diffe	rence file: NONE		
	Memory Map File			
nput files to convert	1-			
File/Data area	Prope	erties		Add Dat
SOF Data	Page	Λ		Add File
Hex User Data		-		
				<u>R</u> emove
				<u>Up</u>
				Down

- 6. Click Add File.
- 7. Select the SOF that you want to convert to a JIC file.
- 8. Click OK.

9. Highlight FlashLoader and click Add Device. See Figure 9.

Figure 9. Highlight FlashLoader

~ ~		1			
Upen Conv	ersion Setup Data		<u>S</u> ave Co	nversion Setup	
Output programming file					
Programming file type:	JTAG Indirect Configuration	File (.iic)			-
Options	Configuration device: EP	CS4 💌	Mode:	Active Serial Configuration	-
File <u>n</u> ame:	D:/temp/output_file.jic				
	Remote/Local update differe	nce file: NONE			Ţ
	Memory Map File				_
nput files to convert	Presed	ine		440	- 1
Flash Loader	Flopen	les		<u>A</u> dd U al	a
SOF Data	Page 0			Add Devic	:e)
test_design.sof	EP1C2	JF400			
Hex User Data				<u>R</u> emov	8
				11-	
				<u></u>	
					1

- 10. Click **OK**. The **Select Devices** page displays.
- 11. Select the targeted FPGA that you are using to program the serial configuration device. See Figure 10.

ASC devices Cyclone ASC devices Cyclone Chanced Configuration Devices EPC1 EPC2 CxtALIBUR_ARM FLEX10K4 FLEX10K4 FLEX10K4 FLEX10K6 FLEX000 CEL2x000 MAX11 MAX3000A MAX7000B MAX700B MAX70B MAX70		■ EP1C12 ■ EP1C12F256 ■ EP1C2F324 ■ EP1C20 ■ EP1C20F300 ■ EP1C3 ■ EP1C31100 ■ EP1C4 ■ EP1C4 ■ EP1C66 ■ EP1C556 ■ EP1C67144	New Import Export Edit Eemove Check
--	--	--	--

Figure 10. Select Devices Page

12. Click **OK**. The **Convert Programming Files** page displays. See Figure 11.

Figure 11. Convert Programming Files Page

pecity the input files to c ou can also import input uture use.	onvert and the type of file information from ot	programming file her files and save	to generate the conve	e. rsion se	tup informatio	n created here for
Upen Cony	ersion Setup Data			<u>s</u> ave u	onversion Se	rup
Output programming file						
Programming file type:	JTAG Indirect Conl	iguration File (.jic	:]			•
Options	<u>Configuration</u> device	EEPCS4	• !	Mode:	Active Seria	Configuration 💌
File name:	D:/temp/output_file	e, jic				
	Remote/Local upda	ate difference file:	NONE			v
	Memory Map Fil	e .	J			_
		-				
Input files to convert						
input lies to convert						
File/Data area		Properties				<u>A</u> dd Data
File/Data area		Properties				Add Data
File/Data area File/Data area Flash Loader EP1C20F400 SOF Data		Properties Page 0				Add Data
File/Data area File/Data area File/Data Loader EP1C20F400 SOF Data Less_design.so	f	Properties Page 0 EP1C20F400				Add Data Add Device
File/Data area □ Flash Loader □ EP1C20F400 □ S0F Data □ Lest_design.so Hex User Data	f	Properties Page 0 EP1C20F400				Add Data Add Device
File/Data area □ Flash Loader □ Elash Loader □ EPIC20F400 □ SOF Data □ test_design.so Hex User Data	f	Properties Page 0 EP1C20F400				Add Data Add Device <u>R</u> emove
File/Data area □ Flash Loader □ Flash Loader □ Elash Loade	f	Properties Page 0 EP1C20F400				Add Data Add Device <u>R</u> emove <u>Up</u> <u>D</u> own
File/Data area □ Flash Loader □ Flash Loader □ Elash Loade	f	Properties Page 0 EP1C20F400				Add Data Add Device <u>H</u> emove <u>Up</u> <u>D</u> own

- 13. Click OK.
- To program the serial configuration device(s) with the JIC file that you just created, add the file to the Quartus II Programmer window and follow the steps in "Programming Serial Configuration Devices Using the Quartus II Programmer & JIC Files" on page 17.

Converting JIC Files to Jam Files in the Quartus II Software

To convert a JIC to a Jam file in the Quartus II software, follow these steps:

- 1. Choose **Programmer** (Tools menu).
- 2. Click Add File. The Select Programming File window displays.
- 3. Browse to the JIC file that was created in "Converting SOF to JIC Files in Quartus II Software" on page 11. Add more JIC files if you are programming multiple serial configuration devices.
- 4. Click Open.
- 5. Choose **Create/Update** and scroll to **Create JAM, SVF, or ISC File** (File menu). See Figure 12.

Figure 12. Create Jam, SVF, or ISC File

🏶 Quartus II								
File Edit View Project Assignments	Processing Tools Window	Help						
D New Ctrl+N	c≥ №	-	- 1 💥 🗷 🔇	8 10 1	► N° №	2.4	8	
🗃 Qpen Ctrl+O								~
⊆lose Ctrl+F4								
🚵 New Project <u>W</u> izard	3yteBlasterII [LPT1]				Mo	ode: JTA(ì	
🗟 Open Project Ctrl+J		Device	Checksum	Usercode	Program/	Verifu	Blank-	Exami
Convert MAX+PLUS II Project	D:/temn/output_file.iic	EP1020	001CE522	FFFFFFFF	Configure	· ciny	Check	
Save Project	LEPCS4	21 1020	001CF522	FFFFFFF				
Clos <u>e</u> Project								
Save Ctrl+S								
Save As								
Save Current Report Section As								
Ele Properties	-							
Create / Update 🔹 🕨	Create HDL Design File fo	or Current File						
Export	Create Symbol Files for C	lurrent File						
Convert Programming Files	Create <u>A</u> HDL Include File:	s for Current File						
🛄 Page Setyp	Create <u>D</u> esign File from S	selected Block						
👌 Print Pre <u>v</u> iew	Update Design File from S	Selected Block						
Brint Ctrl+P	Create SignalTan II Eile fr	rom Design Instance(s)	-					
Recent Files	Create SignalTap II List F	ile						
	Create JAM, SVF, or ISC	File						
Recen <u>c</u> Projects	Create/Update IPS File							~
E <u>x</u> it Alt+F4		111						>
× System A Processing /								
Message:	Location:						-	Locate
Creates JAM, JBC, and SVF files					Idle		NU	1

6. The Create JAM, SVF, or ISC File window displays. See Figure 13.

🧘 Hardware Setup	ByteBlasterII [LPT1]					
Mode:	JTAG					-
Progress:			0%			
M Start	File	Device	Checksum	Usercode	Program/ Configure	Verify
Whi Store	E D:/temp/output_file.jic	EP1C20	001CF522	FFFFFFFF		
== 5(0p	LEPCS4		001CF522	FFFFFFFF		
Auto Detect						
∧ Delete	Croz	to IAM SVE or	ISC File	6		
🍰 Add File		ne shin, stri, or	ISC THE			
We change the	File	name: D:/temp/t	est1.jam		1	
unange File	File		ADI Format (inm)		1	
🗳 Save File	116	Ionnac Joebee 5	Ar E Formac (Jam)	<u>.</u>	1	
Add Device		peration	Programmi	ng options	7	
Add Device		peration Program	Programmi	n g options check	1	
Add Device		peration * Program * Verify	Programmi F Blank- Verify	n g options check		
Add Device		Peration Program Verify	Programmi F Blank- Venty	ng options check		
Add Device		peration Program Verify ock frequency	Programmi F Blank- Verity	ng options		
Add Device		peration Program Verify ock frequency CK frequency: 10	Programmi F Blank- Verity	ng options		
Add Device		peration ¹ Program ¹ Verify ock frequency CK frequency:	0.0 MHz	ng options]	
Add Device	C C T Sup	peration Program Verity ock frequency CK frequency:	0.0 MHz	ng options check		
Add Device	C C C C C C C C C C C C C C C C C C C	peration Program Verity ock frequency CK frequency: ply voltage: 1	Programmi Blank Verity	ng options check		
Add Device	CI CI TI Sup	peration Program Verify ock frequency CK frequency: ply voltage: 1	Programmi F Blank- Verity 0.0 MHz 8 volts	ng options check]	

Figure 13. Converting a JIC File to a Jam File in the Quartus II Software

- 7. Click OK.
- To program the serial configuration device(s) with the Jam file that you just created, add the file to the Quartus II Programmer window and follow the steps in "Programming Serial Configuration Devices Using the Quartus II Programmer & Jam Files" on page 19.

JIC & Jam File Contents

The JIC and Jam files contain:

- A configuration image of the SFL:
 - Created by Quartus II software when you select the targeted FPGA during file conversion (that is, from SOF to JIC file).
 - Configured into the FPGA before the serial configuration device is programmed.
- Programming data for serial configuration devices:
 - Derived from the SOF.
 - Transferred into the serial configuration device during programming.

Figure 14 illustrates the generation of the SFL programming file.

Figure 14. Generation of the SFL Programming File



Programming Serial Configuration Devices With the Quartus II Programmer You can use the Quartus II Programmer to generate serial configuration device programming files. The Quartus II Programmer can generate both JIC and Jam files with the SFL.

This section discusses:

- Programming serial configuration devices using the Quartus II Programmer and JIC files
- Programming serial configuration devices using the Quartus II Programmer and Jam files

Programming Serial Configuration Devices Using the Quartus II Programmer & JIC Files

Use the following steps to program serial configuration devices with JIC files.

- 1. When the SOF-to-JIC file conversion is complete (refer to Figure 11 on page 14), add the JIC file to the Quartus II Programmer window:
 - a. Choose **Programmer** (Tools menu). The **Chain1.cdf** window displays.
 - b. Click **Add File**. From the **Select Programming File** page, browse to the JIC file.
 - c. Click Open.

- 2. Configure the FPGA with the SFL image by checking the FPGA **Program/Configure** box (see Figure 15). This process corresponds to Step 1 of Figure 3 on page 5.
- 3. Program the serial configuration device by checking the corresponding **Program/Configure** box (see Figure 15). This process corresponds to Step 2 of Figure 3 on page 5.
- 4. Click Start.

Figure 15 shows the Quartus II Programmer window with one JIC file.

Figure 15. Quartus II Programmer Window With One JIC File

Lardware Setup ByteBlasterII [LPT1]			Mode: JTAG	✓ Progress	Progress: 0 %			
Mart 🔪 📗	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examin
Stop	D:/temp/output_file.jic LEPCS4	EP1C20	001CF522 001CF522	FFFFFFF				
Auto Detect Delete Add File Change File Save File Add Daviso	Step 3: Click Start.	FPGA check Step 1: Chec configure th the FPGA.	to	Serial configuration device check box. Step 2: Check this box to program the serial configuration device.				
🕐 Up	<							

If the **Program/Configure** check boxes are not specified, the Quartus II Programmer bypasses the request. Also, if the FPGA does not have the SFL image when the serial configuration device data is programmed via the JTAG interface, the programming process fails.

You can program multiple serial configuration devices by including more than one JIC file in the Quartus II programmer.

FPGA has to be in active serial configuration mode to enable SFL to program.

Figure 16 shows the Quartus II Programmer window with multiple JIC files.

🚉 Hardware Setuj	p ByteBlasterII [LPT1]		Mod	ie: JTAG	-	Progre:	\$\$:	0%	
Mastart 🕺	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine	Secur Bit
No Stop	D:/work/1configuration/S.	EP1C6	000A7F3B	FFFFFFF					
Auto Detect	EPUS4 D:/work/1configuration/S	EP1C12	000A7F3B 001294C3	FFFFFFF	2 2				
🗙 Delete	-EPCS4		001294C3	FFFFFFF					
Ъ Add File	1								
Change File									
Save File									
Add Device									
ա Սե	Present for								

Figure 16. Quartus II Programmer Window With Multiple JIC Files

Programming Serial Configuration Devices Using the Quartus II Programmer & Jam Files

When programming with Jam files, the Quartus II Programmer requires that you configure the FPGA and program the serial configuration device in one step, which is why Figure 17 shows just one **Program/Configure** check box.

Use the following steps to program serial configuration devices with Jam files.

- 1. When the JIC-to-Jam file conversion is complete (refer to Figure 13 on page 16), add the Jam file to the Quartus II Programmer window:
 - a. Choose **Programmer** (Tools menu). The **Chain1.cdf** window displays.
 - b. Click **Add File**. From the **Select Programming File** page, browse to the Jam file.
 - c. Click Open.

- 2. Configure the FPGA with the SFL image, and program the serial configuration device by checking the FPGA **Program/Configure** box (see Figure 17). This process corresponds to Step 1 and Step 2 of Figure 3 on page 5.
- 3. Click Start.
- The Jam file is generated from the JIC file via the chain description file (**.cdf**). Refer to Quartus II Help for more information.

Figure 17 shows the Quartus II Programmer window with one Jam file.

Figure 17. Quartus II Programmer Window With One Jam File

🖺 Chain3.cdf*										E		
Lardware Setup ByteBlasterII [LPT1]					Mode: JTAG			▼ Progress: 0%				
🏴 Start	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine	Security Bit	Erase	ISP CLAMP	
M Stop	D:/temp/jam1.jam	EP1C6	000A7F3B	FFFFFFFF								
Auto Detect												
🗙 Delete												
🍰 Add File												
Change File												
Save File												
Add Device												
🕐 Up												
🔑 Down	<				Í)>	

You can program multiple serial configuration devices with one Jam file in the Quartus II Programmer. Figure 18 shows the Quartus II Programmer window with one Jam file programming multiple serial configuration devices.

Figure 18. Quartus II Programmer Window With One Jam File Programming Multiple Serial Configuration Devices

🖺 Chain3.cdf*												
Aardware Setup ByteBlasterII [LPT1]					Mode: JTAG			Progress:			0%	
🏓 Start	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine	Security Bit	Erase	ISP CLAMP	
🖬 Stop	D:/work/1configuration/	MULTIPLE	MULTIPLE	MULTIPLE								
Auto Detect												
X Delete												
🍰 Add File												
Change File												
Save File												
Add Device												
1 Up												
🐠 Down	<										>	

Conclusion

The SFL offers an in-system programming solution for serial configuration devices via the JTAG interface. Because the JTAG is an industry standard interface, it is preferred over the AS interface. Thus, in addition to the conventional method of programming serial configuration devices via the AS interface, you now have the option to use the JTAG interface and the SFL solution.

Revision History Ve

Version 3.0

Information contained in *AN 370: Using the Serial FlashLoader With the Quartus II Software* version 3.0 supersedes information published in the previous version.

- Updated the first paragraph in the "Introduction" section
- Updated the first column of Table 1
- Updated the forth paragraph in the "Introduction" section
- Updated the bulleted list in the "Introduction" section
- Added Note to Step 2 of the "Steps for Programming Single & Multiple Serial Configuration Devices With the SFL Solution" section

- Added Figure 2 to the "Steps for Programming Single & Multiple Serial Configuration Devices With the SFL Solution" section
- Added notes to Figures 3 and 4
- Added the "Using the Serial Flash Loader Megafunction in Quartus II Software" section
- Added a note to Figure 11 and after Figure 11

Version 2.0

Information contained in *AN 370: Using the Serial FlashLoader With the Quartus II Software* version 2.0 supersedes information published in the previous version.

- Updated the first and forth paragraph and the bulleted list in the "Introduction" section
- Updated column one of Table 1
- Updated steps 2 and 3 in the "Steps for Programming Single & Multiple Serial Configuration Devices With the SFL Solution" section



101 Innovation Drive San Jose, CA 95134 (408) 544-7000 www.altera.com Applications Hotline: (800) 800-EPLD Literature Services: literature@altera.com Copyright © 2006 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make change.

es to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

