Quartus II Programmer 18

2013.11.04

QII53022 Subscribe Send Feedback

The Quartus II Programmer allows you to program and configure Altera® CPLD, FPGA, and configuration devices. After compiling your design, use the Quartus II Programmer to program or configure your device, to test its functionality on a circuit board.

Related Information

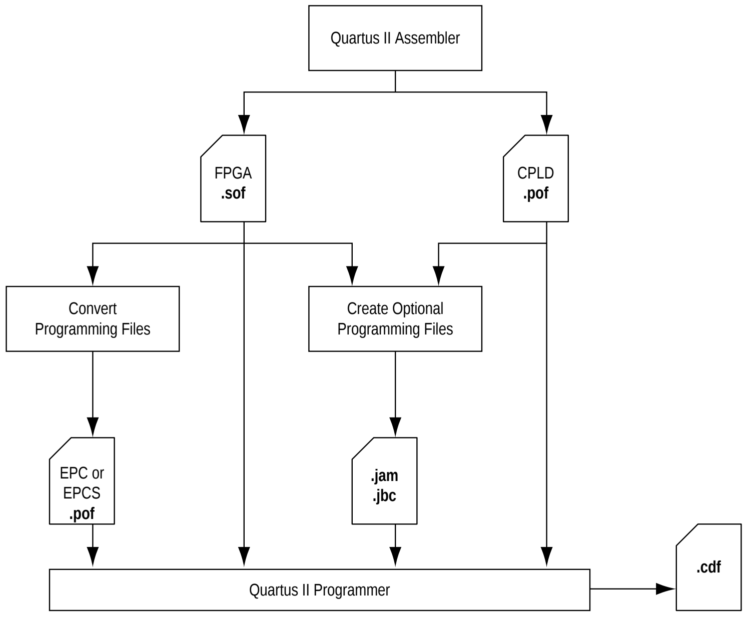
- Programming Devices

Programming Flow

The following steps describe the general overview of the programming flow:

1. Compile your design, such that the Quartus II Assembler generates the programming or configuration file.

Figure 18-1: Programming File Generation Flow



Note: For more information about Chain Description Files (**.cdf**), refer to **About Programming** in Quartus II Help.

2. Convert the programming or configuration file to target your configuration device and, optionally, create secondary programming files.

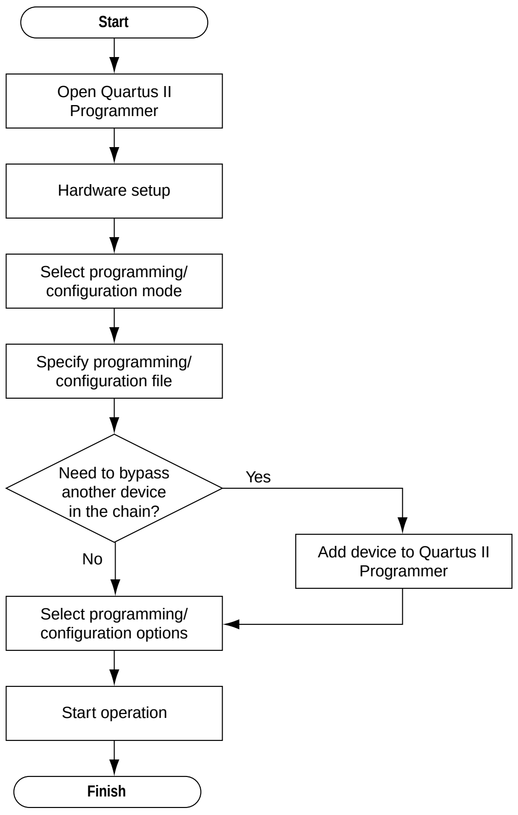
   The following table lists the programming and configuration file formats supported by Altera FPGAs, CPLDs, and configuration devices.

Table 18-1: Programming and Configuration File Format

File Format	FPGA	CPLD	Configuration Device	Serial Configuration Device
SRAM Object File (**.sof**)	Yes	—	—	—
Programmer Object File (**.pof**)	—	Yes	Yes	Yes
JEDEC JESD71 STAPL Format File (**.jam**)	Yes	Yes	Yes	—
Jam Byte Code File (**.jbc**)	Yes	Yes	Yes	—

3. Program and configure the FPGA, CPLD, or configuration device using the programming or configuration file with the Quartus II Programmer.

Figure 18-2: Programming Flow


Start
Open Quartus II Programmer
Hardware setup
Select programming/ configuration mode
Specify programming/ configuration file
Need to bypass another device in the chain?
Yes
Add device to Quartus II Programmer
No
Select programming/ configuration options
Start operation
Finish

Optional Programming or Configuration Files

The Quartus II software can generate optional programming or configuration files in various formats that you can use with programming tools other than the Quartus II Programmer. When you compile a design in the Quartus II software, the Assembler automatically generates either a **.sof** or **.pof**. The Assembler also allows you to convert FPGA configuration files to programming files for configuration devices.

Related Information

- **About Optional Programming Files**
- **AN 425: Using Command-Line Jam STAPL Solution for Device Programming**

  Describes how to use the **.jam** and **.jbc** programming files with the Jam STAPL Player, Jam STAPL Byte-Code Player, and the `quartus_jli` command-line executable.

Secondary Programming Files

The Quartus II software generates programming files in various formats for use with different programming tools.

The following table lists the file types generated by the Quartus II software and supported by the Quartus II Programmer.

Table 18-2: File Types Generated by the Quartus II Software and Supported by the Quartus II Programmer

File Type	Generated by the Quartus II Software	Supported by the Quartus II Programmer
.sof	Yes	Yes
.pof	Yes	Yes
.jam	Yes	Yes
.jbc	Yes	Yes
JTAG Indirect Configuration File (**.jic**)	Yes	Yes
Serial Vector Format File (**.svf**)	Yes	—
In System Configuration File (**.isc**)	Yes	—
Hexadecimal (Intel-Format) Output File (**.hexout**)	Yes	—
Raw Binary File (**.rbf**)	Yes	—
Raw Binary File for Partial Reconfiguration (**.rbf**)	Yes	—
Tabular Text File (**.ttf**)	Yes	—
Raw Programming Data File (**.rpd**)	Yes	—

Related Information

- **Generating Secondary Programming Files**

Quartus II Programmer GUI

The Quartus II Programmer GUI is a window in which you can add your programming and configuration files, specify programming options and hardware, and start the programming or configuration of the device.

To open the Programmer window, on the Tools menu, click **Programmer**. As you proceed through the programming flow, the Quartus II Message window reports the status of each operation.

Related Information

- **Programmer Window**
  Describes the Programmer window.
- **Programmer Page (Options Dialog Box)**
  Describes the options in the **Tools** menu.

Editing the Device Details of an Unknown Device

If the Quartus II Programmer automatically detects devices with shared JTAG IDs, the Programmer prompts you to specify the correct device in the JTAG chain.

If the Programmer does not prompt you to specify the correct device in the JTAG chain, then you must add a user defined device in the Quartus II software for each unknown device in the JTAG chain and specify the instruction register length for each device.

To edit the device details of an unknown device, follow these steps:

1. Double-click on the unknown device listed under the device column.
2. Click **Edit**.
3. Change the device **Name**.
4. Enter the **Instruction register Length**.
5. Click **OK**.
6. Save the **.cdf**.

Setting Up Your Hardware

The Quartus II Programmer provides the flexibility to choose a download cable or programming hardware. Before you can program or configure your device, you must have the correct hardware setup.

Related Information

- **Setting Up Programming Hardware**
  Describes the steps to set up your hardware.
- **Setting up Programming Hardware in Quartus II Software**
  Describes the programming hardware driver installation.

Setting the JTAG Hardware

The JTAG server allows the Quartus II Programmer to access the JTAG hardware. You can also access the JTAG download cable or programming hardware connected to a remote computer through the JTAG server of that computer. With the JTAG server, you can control the programming or configuration of devices from a single computer through other computers at remote locations. The JTAG server uses the TCP/IP communications protocol.

Related Information

- **Using the JTAG Server**
  Lists how to use the JTAG Server

Using the JTAG Chain Debugger Tool

The JTAG Chain Debugger tool allows you to test the JTAG chain integrity and detect intermittent failures of the JTAG chain. In addition, the tool allows you to shift in JTAG instructions and data through the JTAG interface and step through the test access port (TAP) controller state machine for debugging purposes. You access the tool from the Tools menu on the main menu of the Quartus II software.

Related Information

- **Using the JTAG Chain Debugger**

Stand-Alone Quartus II Programmer

Altera offers the free stand-alone Quartus II Programmer, which has the same full functionality as the Quartus II Programmer in the Quartus II software. The stand-alone Quartus II Programmer is useful when programming your devices with another workstation, so you do not need two full licenses. You can download the stand-alone Quartus II Programmer from the Download Center on the Altera website.

Related Information

- **Download Center**
  You can download the stand-alone Quartus II Programmer from this page.

Programming and Configuration Modes

The following table lists the programming and configuration modes supported by Altera devices.

Table 18-3: Programming and Configuration Modes

Configuration Mode Supported by the Quartus II Programmer	FPGA	CPLD	Configuration Device	Serial Configuration Device
JTAG	Yes	Yes	Yes	—
Passive serial (PS)	Yes	—	—	—
Active serial (AS)	—	—	—	Yes
Configuration via Protocol (CvP)	Yes	—	—	—
In-socket modes (ISM)	—	Yes (except for MAX II CPLDs)	Yes	Yes

Related Information

- About Programming
- Configuration via Protocol (CvP) Implementation in Altera FPGAs User Guide
  Describes the CvP configuration mode.
- Programming Adapters
  Contains a list of programming adapters available for Altera devices.

Design Security Keys

The Quartus II Programmer supports the generation of encryption key programming files and encrypted configuration files for Altera FPGAs that support the design security feature. You can also use the Quartus II Programmer to program the encryption key into the FPGA.

Related Information

- AN 341: Using the Design Security Feature in Stratix II and Stratix II GX Devices
  Describes how to use the feature in Stratix II and Stratix II GX devices.
- AN 512: Using the Design Security Feature in Stratix III Devices
  Describes how to use the feature in Stratix III devices.

Convert Programming Files Dialog Box

The **Convert Programming Files** dialog box in the Programmer allows you to convert programming files from one file format to another. For example, to store the FPGA data in configuration devices, you can convert the **.sof** data to another format, such as **.pof**, **.hexout**, **.rbf**, **.rpd**, or **.jic**, and then program the configuration device.

You can also configure multiple devices with an external host, such as a microprocessor or CPLD. For example, you can combine multiple **.sof** files into one **.pof**.

To access the **Convert Programming Files** dialog box, on the main menu of the Quartus II software, click **File**, and then click **Convert Programming Files**.

Related Information

- Convert Programming Files Dialog Box

Debugging Your Configuration

Use the **Advanced** option in the **Convert Programming Files** dialog box to debug your configuration. You must choose the advanced settings that apply to your Altera device. You can direct the Quartus II software to enable or disable an advanced option by turning the option on or off in the **Advanced Options** dialog box.

When you change settings in the **Advanced Options** dialog box, the change affects **.pof**, **.jic**, **.rpd**, and **.rbf** files.

The following table lists the **Advanced Options** settings in more detail.

Table 18-4: Advanced Options Settings

Option Setting	Description
Disable EPCS ID check	FPGA skips the EPCS silicon ID verification. Default setting is unavailable (EPCS ID check is enabled). Applies to the single- and multi-device AS configuration modes on all FPGA devices.
Disable AS mode CONF_DONE error check	FPGA skips the `CONF_DONE` error check. Default setting is unavailable (AS mode `CONF_DONE` error check is enabled). Applies to single- and multi-device (AS) configuration modes on all FPGA devices. The `CONF_DONE` error check is disabled by default for Stratix V, Arria V, and Cyclone V devices for AS-PS multi device configuration mode.
Program Length Count adjustment	Specifies the offset you can apply to the computed PLC of the entire bitstream. Default setting is 0. The value must be an integer. Applies to single- and multi-device (AS) configuration modes on all FPGA devices.
Post-chain bitstream pad bytes	Specifies the number of pad bytes appended to the end of an entire bitstream. Default value is set to 0 if the bitstream of the last device is uncompressed. Set to 2 if the bitstream of the last device is compressed.
Post-device bitstream pad bytes	Specifies the number of pad bytes appended to the end of the bitstream of a device. Default value is 0. No negative integer. Applies to all single-device configuration modes on all FPGA devices.

Option Setting	Description
Bitslice padding value	Specifies the padding value used to prepare bitslice configuration bitstreams, such that all bitslice configuration chains simultaneously receive their final configuration data bit. Default value is 1. Valid setting is 0 or 1. Use only in 2, 4, and 8-bit PS configuration mode, when you use an EPC device with the decompression feature enabled. Applies to all FPGA devices that support enhanced configuration devices.

The following table lists the symptoms you may encounter if a configuration fails, and describes the advanced options you must use to debug your configuration.

Failure Symptoms	Disable EPCS ID Check	Disable AS Mode CONF_ DONE Error Check	PLC Settings	Post-Chain Bitstream Pad Bytes	Post-Device Bitstream Pad Bytes	Bitslice Padding Value
Configuration failure occurs after a configuration cycle.	—	Yes	Yes	Yes (1)	Yes (2)	—
Decompression feature is enabled.	—	Yes	Yes	Yes (1)	Yes (2)	—
Encryption feature is enabled.	—	Yes	Yes	Yes (1)	Yes (2)	—
`CONF_DONE` stays low after a configuration cycle.	—	Yes	Yes (3)	Yes (1)	Yes (2)	—

(2) Use only for single-device chain
(1) Use only for multi-device chain
(3) Start with positive offset to the PLC settings

Failure Symptoms	Disable EPCS ID Check	Disable AS Mode CONF_ DONE Error Check	PLC Settings	Post-Chain Bitstream Pad Bytes	Post-Device Bitstream Pad Bytes	Bitslice Padding Value
`CONF_DONE` goes high momentarily after a configuration cycle.	—	Yes	Yes [4]	—	—	—
FPGA does not enter user mode even though `CONF_DONE` goes high.	—	—	—	Yes [1]	Yes [2]	—
Configuration failure occurs at the beginning of a configuration cycle.	Yes	—	—	—	—	—
Newly introduced EPCS, such as EPCS128.	Yes	—	—	—	—	—
Failure in **.pof** generation for EPC device using Quartus II Convert Programming File Utility when the decompres-sion feature is enabled.	—	—	—	—	—	Yes

[4] Start with negative offset to the PLC settings

Converting Programming Files for Partial Reconfiguration

The **Convert Programming File** dialog box supports the following programming file generation and option for Partial Reconfiguration:

- Partial-Masked SRAM Object File (**.pmsf**) output file generation, with **.msf** and **.sof** as input files.
- **.rbf** for Partial Reconfiguration output file generation, with a **.pmsf** as the input file.

  Note: The **.rbf** for Partial Reconfiguration file is only for Partial Reconfiguration.

- Providing the **Enable decompression during Partial Reconfiguration** option to enable the option bit for bitstream decompression during Partial Reconfiguration, when converting a full design **.sof** to any supported file type.

Related Information

- Design Planning for Partial Reconfiguration

Generating .pmsf using a .msf and a .sof

To generate the **.pmsf** in the **Convert Programming Files** dialog box, follow these steps:

1. In the **Convert Programming Files** dialog box, under the **Programming file type** field, select **Partial-Masked SRAM Object File (.pmsf)**.
2. In the **File name field**, specify the necessary output file name.
3. In the **Input files to convert** field, add necessary input files to convert. You can add only a **.msf** and **.sof**.
4. Click **Generate**.

Generating .rbf for Partial Reconfiguration Using a .pmsf

After you have successfully generated the **.pmsf**, you can convert the **.pmsf** to a **.rbf** for Partial Reconfiguration in the **Convert Programming Files** dialog box.

To generate the **.rbf** for Partial Reconfiguration, follow these steps:

1. In the **Convert Programming Files** dialog box, in the **Programming file type** field, select **Raw Binary File for Partial Reconfiguration (.rbf)**.
2. In the **File name** field, specify the output file name.
3. In the **Input files to conver**t field, add input files to convert. You can add only a **.pmsf**.
4. After adding the **.pmsf**, select the **.pmsf** and click **Properties**. The **PMSF File Properties** dialog box appears.
5. Make your selection either by turning on or turning off the following options:
   - **Compression option**—This option enables compression on Partial Reconfiguration bitstream. If you turn on this option, then you must turn on the **Enable decompression during Partial Reconfiguration** option.
   - **Enable SCRUB mode option**—The default of this option is based on AND/OR mode. This option is valid only when Partial Reconfiguration masks in your design are not overlapped vertically. Otherwise, you cannot generate the **.rbf** for Partial Reconfiguration.
   - **Write memory contents option**—This option is a workaround for initialized RAM/ROM in a Partial Reconfiguration region.

For more information about these option, refer to the Design Planning for Partial Reconfiguration.

6. Click **OK**.
7. Click **Generate**.

Enable Decompression during Partial Reconfiguration Option

You can turn on the **Enable decompression during Partial Reconfiguration** option in the **SOF File Properties: Bitstream Encryption** dialog box, which can be accessed from the **Convert Programming File** dialog box. This option is available when converting a **.sof** to any supported programming file types listed in Table 18-2.

This option is hidden for other targeted devices that do not support Partial Reconfiguration. To view this option in the **SOF File Properties: Bitstream Encryption** dialog box, the **.sof** must be targeted on an Altera device that supports Partial Reconfiguration.

If you turn on the **Compression** option when generating the **.rbf** for Partial Reconfiguration, then you must turn on the **Enable decompression during Partial Reconfiguration** option.

Flash Loaders

Parallel and serial configuration devices do not support the JTAG interface. However, you can use a flash loader to program configuration devices in-system via the JTAG interface. You can use an FPGA as a bridge between the JTAG interface and the configuration device. The Quartus II software supports parallel and serial flash loaders.

Related Information

- About Flash Loaders

JTAG Debug Mode for Partial Reconfiguration

The JTAG debug mode allows you to configure Partial Reconfiguration bitstream through the JTAG interface. Use this feature to debug Partial Reconfiguration bitstream and eventually helping you in your Partial Reconfiguration design prototyping. This feature is available for internal and external host.

During JTAG debug operation, the JTAG command sent from the Quartus II Programmer ignores and overrides most of the Partial Reconfiguration megafunction interface signals (`clk`, `pr_start`, `double_pr`, `data[]`, `data_valid`, and `data_read`).

Note: The TCK is the main clock source for Partial Reconfiguration megafunction during this operation.

You can view the status of Partial Reconfiguration operation in the messages box and the Progress bar in the Quartus II Programmer. The `PR_DONE`, `PR_ERROR`, and `CRC_ERROR` signals will be monitored during Partial Reconfiguration operation and reported in the Messages box at the end of the operation.

The Quartus II Programmer can detect the number of `PR_DONE` instruction(s) in plain or compressed Partial Reconfiguration bitstream and, therefore, can handle single or double Partial Reconfiguration cycle accordingly. However, only single Partial Reconfiguration cycle is supported for encrypted Partial Reconfiguration bitstream in JTAG debug mode (provided that the specified device is configured with the encrypted base bitstream which contains the Partial Reconfiguration megafunction in the design).

Note: Configuring an incompatible PR bitstream to the specified device may corrupt your design, including the routing path and the Partial Reconfiguration megafunction placed in the static region. When this issue occurs, the Partial Reconfiguration megafunction stays in an undefined state, and the Quartus II Programmer is unable to reset the megafunction. As a result, the Quartus II Programmer generates the following error when you try to configure a new Partial Reconfiguration bitstream:

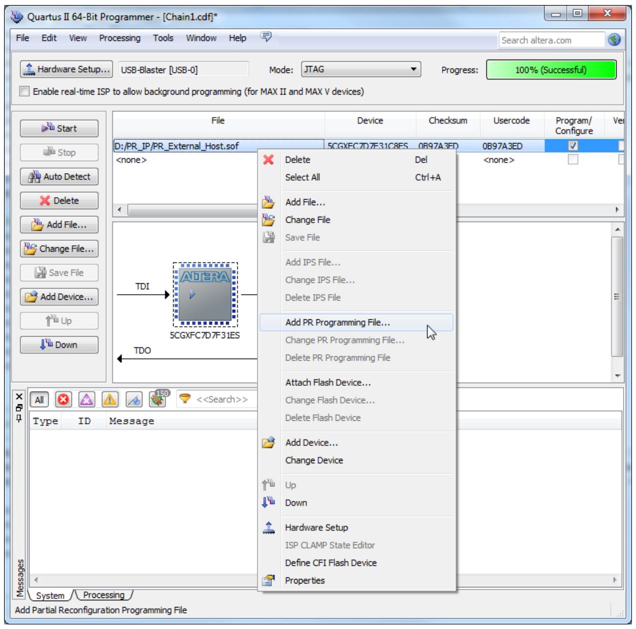
Error (12897): Partial Reconfiguration status: Can't reset the PR megafunction. This issue occurred because the design was corrupted by an incompatible PR bitstream in the previous PR operation. You must reconfigure the device with a good design.

Configuring Partial Reconfiguration Bitstream in JTAG Debug Mode

To configure the Partial Reconfiguration bitstream in JTAG debug mode, follow these steps:

1. In the Quartus II Programmer GUI, right click on a highlighted base bitstream (in **.sof**) and then click **Add PR Programming File** to add the Partial Reconfiguration bitstream (in **.rbf**).

Figure 18-3: Adding PR Programming File



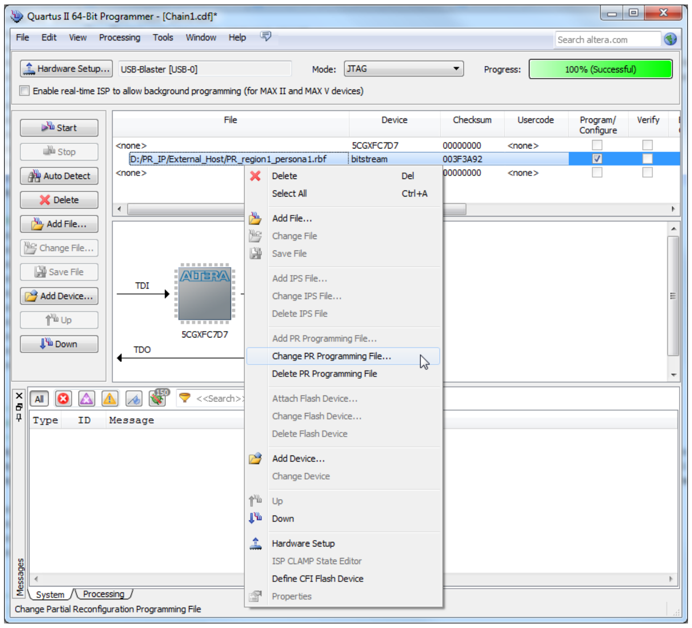
USB-Blaster [USB-0]   Mode: JTAG   Progress: 100% (Successful)

File	Device	Checksum	Usercode	Program/ Configure
D:/PR_IP/PR_External_Host.sof	5CGXFC7D7F31C8ES	0B97A3ED	0B97A3ED	✓
<none>			<none>	

Add Partial Reconfiguration Programming File

2. After adding the Partial Reconfiguration bitstream, you can change or delete the Partial Reconfiguration programming file by clicking **Change PR Programming File** or **Delete PR Programming File**.

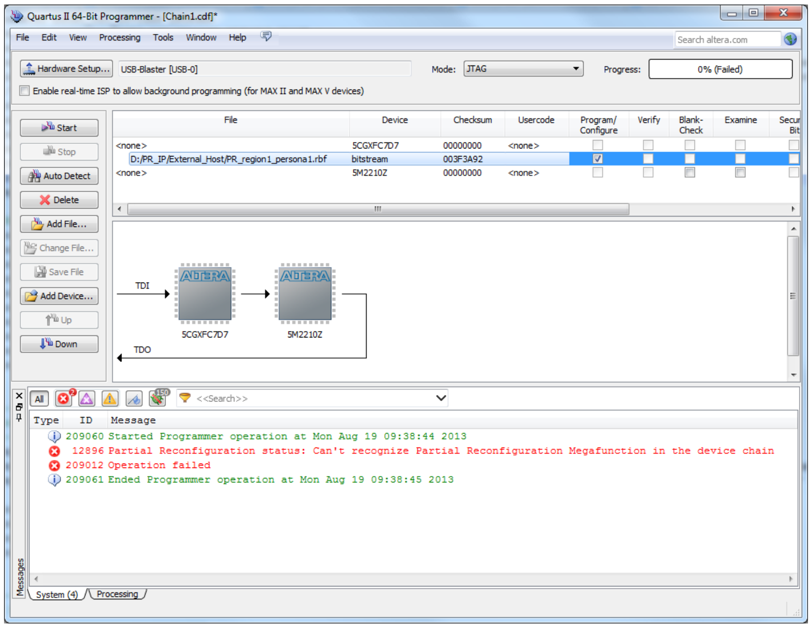
Figure 18-4: Change PR Programming File or Delete PR Programming File



File	Device	Checksum	Usercode	Program/ Configure	Verify
<none>	5CGXFC7D7	00000000	<none>	☐	☐
D:/PR_IP/External_Host/PR_region1_persona1.rbf	bitstream	003F3A92		✓	☐
<none>		00000000	<none>	☐	☐

3. Click **Start** to configure the Partial Reconfiguration bitstream. The Quartus II Programmer generates an error message if the specified device does not contain the Partial Reconfiguration megafunction in the design (you must instantiate Partial Reconfiguration megafunction in your design to use the JTAG debug mode).

Figure 18-5: Starting PR Bitstream Configuration

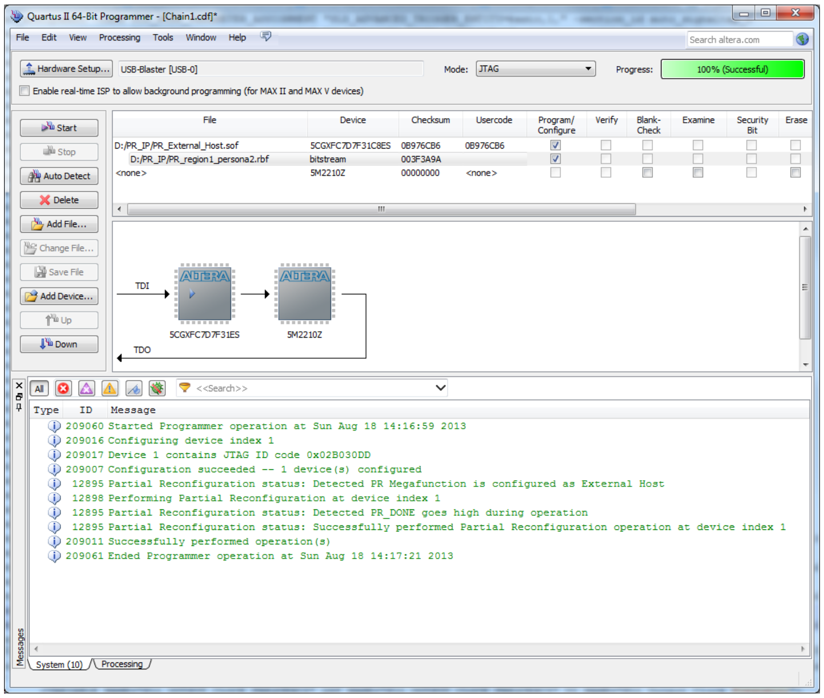


File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine	Secur Bit
<none>	5CGXFC7D7	00000000	<none>	☐	☐	☐	☐	☐
D:/PR_IP/External_Host/PR_region1_persona1.rbf	bitstream	003F3A92		✓	☐	☐	☐	☐
<none>	5M2210Z	00000000	<none>	☐	☐	☐	☐	☐

```
Type   ID   Message
     209060 Started Programmer operation at Mon Aug 19 09:38:44 2013
      12896 Partial Reconfiguration status: Can't recognize Partial Reconfiguration Megafunction in the device chain
     209012 Operation failed
     209061 Ended Programmer operation at Mon Aug 19 09:38:45 2013
```

4. Configure the valid **.rbf** in JTAG debug mode with the Quartus II Programmer.

Figure 18-6: Configuring Valid .rbf

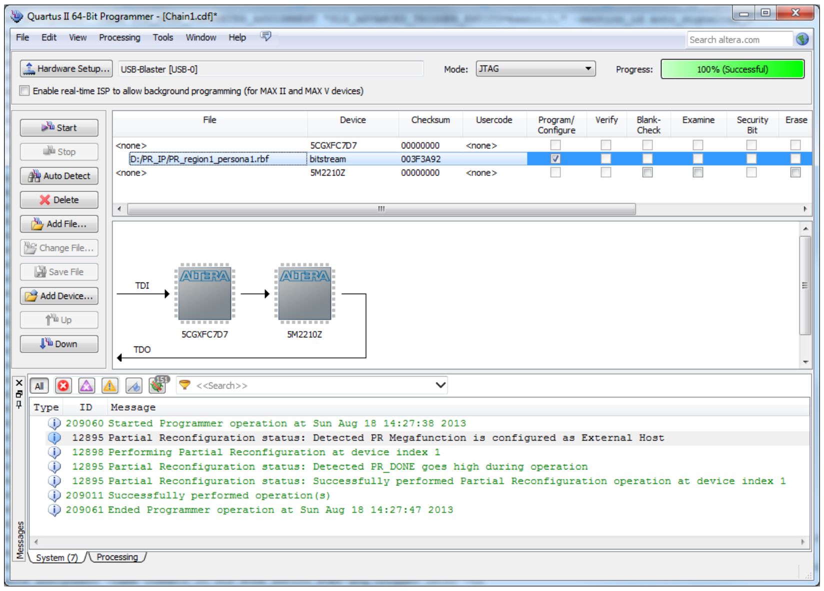


File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine	Security Bit	Erase
D:/PR_IP/PR_External_Host.sof	5CGXFC7D7F31C8ES	0B976CB6	0B976CB6	☑	☐	☐	☐	☐	☐
D:/PR_IP/PR_region1_persona2.rbf	bitstream	003F3A9A		☑	☐	☐	☐	☐	☐
<none>	5M2210Z	00000000	<none>	☐	☐	☐	☐	☐	☐

```
Type  ID     Message
      209060 Started Programmer operation at Sun Aug 18 14:16:59 2013
      209016 Configuring device index 1
      209017 Device 1 contains JTAG ID code 0x02B030DD
      209007 Configuration succeeded -- 1 device(s) configured
       12895 Partial Reconfiguration status: Detected PR Megafunction is configured as External Host
       12898 Performing Partial Reconfiguration at device index 1
       12895 Partial Reconfiguration status: Detected PR_DONE goes high during operation
       12895 Partial Reconfiguration status: Successfully performed Partial Reconfiguration operation at device index 1
      209011 Successfully performed operation(s)
      209061 Ended Programmer operation at Sun Aug 18 14:17:21 2013
```

5. The Partial Reconfiguration megafunction is pre-programmed on the specified device.

Figure 18-7: Partial Reconfiguration Megafunction Successfully Pre-programmed

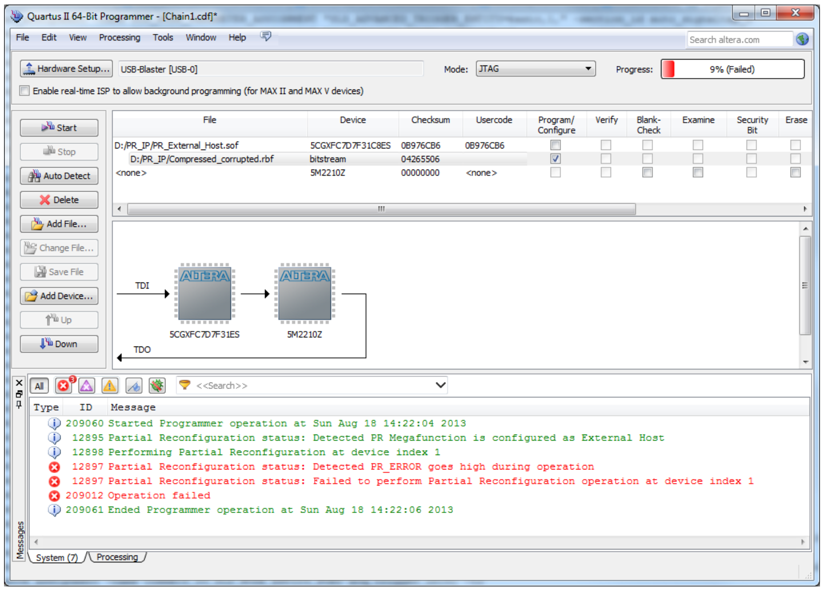


File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine	Security Bit	Erase
<none>	5CGXFC7D7	00000000	<none>	☐	☐	☐	☐	☐	☐
D:/PR_IP/PR_region1_persona1.rbf	bitstream	003F3A92		☑	☐	☐	☐	☐	☐
<none>	5M2210Z	00000000	<none>	☐	☐	☐	☐	☐	☐

```
Type  ID     Message
      209060 Started Programmer operation at Sun Aug 18 14:27:38 2013
       12895 Partial Reconfiguration status: Detected PR Megafunction is configured as External Host
       12898 Performing Partial Reconfiguration at device index 1
       12895 Partial Reconfiguration status: Detected PR_DONE goes high during operation
       12895 Partial Reconfiguration status: Successfully performed Partial Reconfiguration operation at device index 1
      209011 Successfully performed operation(s)
      209061 Ended Programmer operation at Sun Aug 18 14:27:47 2013
```

6. The Quartus II Programmer reports error when you try to configure the corrupted **.rbf** in JTAG debug mode.

Figure 18-8: Configuring Corrupted .rbf



USB-Blaster [USB-0] Mode: JTAG Progress: 9% (Failed)

Enable real-time ISP to allow background programming (for MAX II and MAX V devices)

File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank-Check	Examine	Security Bit	Erase
D:/PR_IP/PR_External_Host.sof	5CGXFC7D7F31C8ES	0B976CB6	0B976CB6	☐	☐	☐	☐	☐	☐
D:/PR_IP/Compressed_corrupted.rbf	bitstream	04265506		✓	☐	☐	☐	☐	☐
<none>	5M2210Z	00000000	<none>	☐	☐	☐	☐	☐	☐

```
Type  ID     Message
      209060 Started Programmer operation at Sun Aug 18 14:22:04 2013
       12895 Partial Reconfiguration status: Detected PR Megafunction is configured as External Host
       12898 Performing Partial Reconfiguration at device index 1
       12897 Partial Reconfiguration status: Detected PR_ERROR goes high during operation
       12897 Partial Reconfiguration status: Failed to perform Partial Reconfiguration operation at device index 1
      209012 Operation failed
      209061 Ended Programmer operation at Sun Aug 18 14:22:06 2013
```

Scripting Support

In addition to the Quartus II Programmer GUI, you can use the Quartus II command-line executable `quartus_pgm.exe` to access programmer functionality from the command line and from scripts. The programmer accepts **.pof**, **.sof**, and **.jic** programming or configuration files and Chain Description Files (**.cdf**).

The following example shows a command that programs a device:

```
quartus_pgm –c byteblasterII –m jtag –o bpv;design.pof
```

Where:

- `-c byteblasterII` specifies the ByteBlaster II download cable
- `-m jtag` specifies the JTAG programming mode
- `-o bpv` represents the blank-check, program, and verify operations
- `design.pof` represents the **.pof** used for the programming

The Programmer automatically executes the erase operation before programming the device.

Related Information

- About Quartus II Scripting

The jtagconfig Debugging Tool

You can use the `jtagconfig` command-line utility (which is similar to the auto detect operation in the Quartus II Programmer) to check the devices in a JTAG chain and the user-defined devices.

For more information about the `jtagconfig` utility, type one of the following commands at the command prompt:

```
jtagconfig -h

jtagconfig --help
```

Note: The help switch does not reference the `-n` switch. The `jtagconfig -n` command shows each node for each JTAG device.

Related Information

- Command-Line Scripting

Generating .pmsf using a .msf and a .sof

You can generate a **.pmsf** with the `quartus_cpf` command by typing the following command:

```
quartus_cpf -p <pr_revision.msf> <pr_revision.sof> <new_filename.pmsf>
```

Generating .rbf for Partial Reconfiguration using a .pmsf

You can generate a **.rbf** for Partial Reconfiguration with the `quartus_cpf` command by typing the following command:

```
quartus_cpf -o foo.txt -c <pr_revision.pmsf> <pr_revision.rbf>
```

Note: You must run this command in the same directory where the files are located.

Document Revision History

The following table lists the revision history.

Table 18-5: Document Revision History

Date	Version	Chages
November 2013	13.1.0	• Converted to DITA format. • Added JTAG Debug Mode for Partial Reconfiguration on page 18-11 and Configuring Partial Reconfiguration Bitstream in JTAG Debug Mode on page 18-12.

Date	Version	Chages
November 2012	12.1.0	• Updated Table 18–3 on page 18–6, and Table 18–4 on page 18–8. • Added "Converting Programming Files for Partial Reconfiguration" on page 18–10, "Generating .pmsf using a .msf and a .sof" on page 18–10, "Generating .rbf for Partial Reconfiguration Using a .pmsf" on page 18–12, "Enable Decompression during Partial Reconfiguration Option" on page 18–14 • Updated "Scripting Support" on page 18–15.
June 2012	12.0.0	• Updated Table 18–5 on page 18–8. • Updated "Quartus II Programmer GUI" on page 18–3.
November 2011	11.1.0	• Updated "Configuration Modes" on page 18–5. • Added "Optional Programming or Configuration Files" on page 18–6. • Updated Table 18–2 on page 18–5.
May 2011	11.0.0	• Added links to Quartus II Help. • Updated "Hardware Setup" on page 21–4 and "JTAG Chain Debugger Tool" on page 21–4.
December 2010	10.1.0	• Changed to new document template. • Updated "JTAG Chain Debugger Example" on page 20–4. • Added links to Quartus II Help. • Reorganized chapter.
July 2010	10.0.0	• Added links to Quartus II Help. • Deleted screen shots.
November 2009	9.1.0	No change to content.
March 2009	9.0.0	• Added a row to Table 21–4. • Changed references from "JTAG Chain Debug" to "JTAG Chain Debugger". • Updated figures.