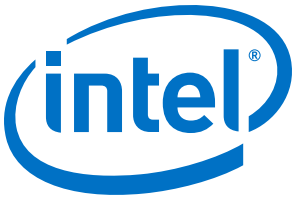


Intel® Quartus® Prime Pro Edition User Guide

Programmer

Updated for Intel® Quartus® Prime Design Suite: **18.1**

Subscribe
Send Feedback

UG-20134 | 2018.10.09
Latest document on the web: PDF | HTML

Contents

1. Generating Programming Files

In the Intel® Quartus® Prime Compiler, the Assembler module generates files for device programming according to specifications in the **Device & Pin Options** dialog box. Most compilation flows direct the Assembler module to generate SRAM Object Files (`.sof`). The Intel Quartus Prime software allows converting these `.sof` into other files to further specify how to save the logic in the target FPGA or configuration device.

Related Information

- Generating Programming Files
  In *Intel Quartus Prime Pro Edition User Guide: Compiler*
- SRAM Object File (.sof) Definition
  In *Intel Quartus Prime Help*
- Configuration Devices Overview

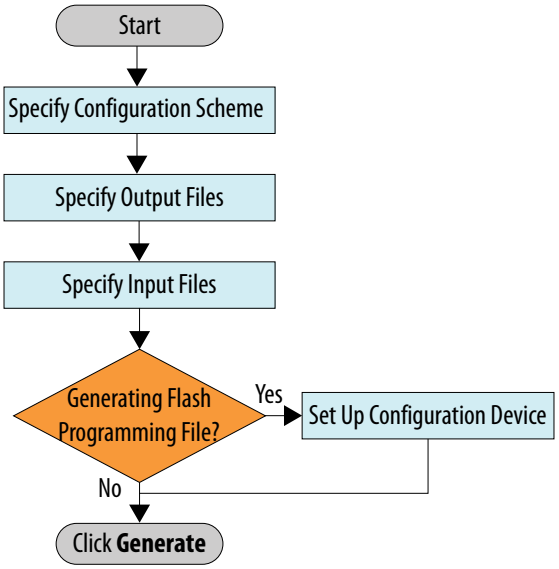
1.1. Generating Programming Files Walkthrough

The Intel Quartus Prime Programming File Generator tool (**File ➤ Programming File Generator**) allows you specify and generate programming files for a target device.

The Programming File Generator's graphical interface adapts to your previous choices by only displaying options that generate valid results, and enabling file generation only after the setup is complete.

The following figure illustrates the stages of the file generation process:

Figure 1. Programming File Generation Flow



Related Information

AN 827: Unified Tool for Generating Programming Files

1.1.1. Specifying Device and Configuration Scheme

1. Select the device family that the files must target.

   If you leave **Device Family** in **Auto**, the Programming File Generator displays all output file types available for the configuration mode that you select.
2. Select a configuration scheme.

Related Information

Supported Devices on page 4

1.1.1.1. Supported Devices

Table 1. Device Families that the Programming File Generator Tool Supports

Software	Version	Supported Device Families
Intel Quartus Prime Pro Edition	18.1	• Intel Stratix® 10 • Intel MAX® 10 • Intel Cyclone® 10 LP • MAX V

1.1.2. Specifying Output Files

During this stage you specify the output files that contain the design logic.

1. Type or browse to the location where you want the output files.

The default value is the project directory.

2. Type the name of the output files.

   The default value is `output_file`.

3. Select one or more output files to generate.
4. If you want to view or edit the properties of an output file, select the file and then click **Edit...**.

   For example, RPD files have the **Bit swap** property, which you can turn **On** or **Off**.

Related Information

Output File Types on page 10

1.1.3. Specifying Input Files

You specify input files in the **Input Files** tab. The input files that you can add depend on the output file types that you selected in the **Output** tab.

1. To add bitstream (SOF/PMSF/POF) files, click **Add Bitstream...**.

   Depending on the target device, the Intel Quartus Prime software may allow you to add multiple SOF files.

2. To add raw data (HEX/RBF/BIN) files, click **Add Raw Data...**.
3. Optionally, set properties of the input files by selecting the file and then clicking **Properties**.

Related Information

File Types Definition
In *Intel Quartus Prime Help*

1.1.4. Specifying Configuration Device

During this stage you specify the flash device that you want to use for configuration.

Note: The Programming File Generator supports specifying only one configuration device.

1. Click the **Configuration Devices** tab.
2. Click **Add Device**, and select the configuration device from the drop-down menu. The device now appears in the list.
3. If you want to specify flash partitions, click **Add Partitions** and follow the instructions in *Specifying Flash Partitions*
4. Click **Generate**.

Related Information

- Intel MAX 10 Flash Memory User Guide
- Intel Stratix 10 Configuration User Guide

1.1.4.1. Specifying Flash Partitions

Flash partitions allow you to store bitstreams or raw data.

Note: The Programming File Generator supports defining flash partitions only for JIC or POF programming files.

To create flash partitions in the **Configuration Devices** tab:

1. Select the device and click **Add Partition**.
2. In the **Add Partition** dialog box, define the following parameters, and then click **OK**:

Parameter	Description
Name	Name that you give to the partition
Input File	Input file to program into the flash partition
Page	Configuration devices can store multiple configuration bitstreams in flash memory, called pages. CFI configuration device can store up to eight configuration bitstreams. Intel Stratix 10 devices can store up to four configuration bitstreams, including factory image. In Intel Stratix 10 devices, with the remote system update feature enabled, **Page** represents the parity.
Address Mode	The options are: **Option** / **Description** **Auto** — The tool automatically allocates a block in the flash device to store the data. **Block** — You specify the start and end address of the flash partition. **Start** — You only specify the start address of the partition. The tool assigns the end address of the partition based on the input data size.
Start Address	Specifies the start address of the partition. Only enabled when **Address Mode** is **Block** or **Start**.
End Address	Specifies the end address of the partition. Only enabled when **Address Mode** is **Block**.

   The partition associated to the device appears in the device list.
3. If you want to change the parameters of a partition, click the partition and then click **Edit...**.
4. If you want to remove a partition, click the partition and then click **Remove**.

After specifying the settings for all flash partitions, click **Generate.**

1.2. Design Security Keys

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

For details on using design security features in Intel FPGAS, refer to *AN 556: Using the Design Security Features in Intel FPGAs*.

Related Information

AN 556: Using the Design Security Features in Intel FPGAs

1.3. Convert Programming Files Tool

The Intel Quartus Prime software provides the **Convert Programming Files** tool (**File ➤ Convert Programming Files**), which allows you to convert programming files from one file format to another. The **Convert Programming Files** dialog box is a legacy tool that supports file conversion for older device families.

The **Convert Programming Files** tool supports the following design families:

Table 2. Device Families that the Convert Programming Files Tool Supports

Software	Version	Supported Device Families
Intel Quartus Prime Pro Edition	18.1	• Intel Stratix 10(1) • Intel Arria® 10 • Intel MAX 10 • Intel Cyclone 10 GX • Intel Cyclone 10 LP • MAX V

The **Convert Programming Files** tool also allows you to configure multiple devices with an external host, such as a microprocessor or CPLD. For example, you can combine multiple `.sof` files into one `.pof` file.

To save time in subsequent conversions, click **Save Conversion Setup** to write the conversion specifications in a Conversion Setup File (`.cof`).

To load a `.cof` setup in the **Convert Programming Files** dialog box, click **Open Conversion Setup Data** .

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.

Example 1. Conversion Setup File Contents

```
<?xml version="1.0" encoding="US-ASCII" standalone="yes"?>
<cof>
    <output_filenameoutput_file.pof</output_filename>
    <n_pages>1</n_pages>
    <width>1</width>
    <mode>14</mode>
    <sof_data>
        <user_name>Page_0</user_name>
        <page_flags>1</page_flags>
        <bit0>
            <sof_filename>/users/user1/template/output_files/
template_test.sof</sof_filename>
        </bit0>
    </sof_data>
    <version>7</version>
    <create_cvp_file>0</create_cvp_file>
    <create_hps_iocsr>0</create_hps_iocsr>
    <auto_create_rpd>0</auto_create_rpd>
    <options>
        <map_file>1</map_file>
    </options>
    <MAX10_device_options>
        <por>0</por>
```

(1) For single image file conversion

```
        <io_pullup>1</io_pullup>
        <auto_reconfigure>1</auto_reconfigure>
        <isp_source>0</isp_source>
        <verify_protect>0</verify_protect>
        <epof>0</epof>
        <ufm_source>0</ufm_source>
    </MAX10_device_options>
    <advanced_options>
        <ignore_epcs_id_check>0</ignore_epcs_id_check>
        <ignore_condone_check>2</ignore_condone_check>
        <plc_adjustment>0</plc_adjustment>
        <post_chain_bitstream_pad_bytes>-1</post_chain_bitstream_pad_bytes>
        <post_device_bitstream_pad_bytes>-1</post_device_bitstream_pad_bytes>
        <bitslice_pre_padding>1</bitslice_pre_padding>
    </advanced_options>
</cof>
```

Related Information

- quartus_cpf Command Line Tool on page 12
- Convert Programming Files Dialog Box
  In *Intel Quartus Prime Help*

1.3.1. Debugging the Configuration

The **Advanced** option in the **Convert Programming Files** dialog box allows you to debug the configuration. Only choose advanced settings that apply to the design's target Intel FPGA device.

Note: **Advanced Options** are not available for Intel Stratix 10 devices.

You enable or disable advanced options in the **Advanced Options** dialog box. Changes in the **Advanced Options** dialog box affect `.pof`, `.jic`, `.rpd`, and `.rbf` files.

The following table describes the **Advanced Options** settings:

Table 3. Advanced Options Settings

Option Setting	Description	Values
Disable EPCS/EPCQ ID check	Directs the FPGA to skips the EPCS/EPCQ silicon ID verification. Applies to single and multi device AS configuration modes on all devices.	Default setting is OFF (EPCS/EPCQ ID check is enabled).
Disable AS mode CONF_DONE error check	Directs the FPGA to skip the CONF_DONE error check. Applies to single- and multi-device (AS) configuration modes on all devices.	Default setting is OFF (AS mode CONF_DONE error check is enabled).
Program Length Count adjustment	Specifies the offset you can apply to the computed PLC of the entire bitstream. Applies to single- and multi-device (AS) configuration modes on all FPGA devices.	Integer (Default = 0)
Post-chain bitstream pad bytes	Specifies the number of pad bytes appended to the end of an entire bitstream.	If the bitstream of the last device is uncompressed, default value is 0. Otherwise, default is 2
Post-device bitstream pad bytes	Specifies the number of pad bytes appended to the end of the bitstream of a device.	Zero or positive integer. Default is 0

continued...

Option Setting	Description	Values
	Applies to all single-device configuration modes on all FPGA devices.	
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. 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.	0 or 1 Default value is 1

The following table lists possible symptoms of a failing configuration, and describes the advanced options necessary for configuration debugging.

Failure Symptoms	Disable EPCS/ EPCQ 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 (2)	Yes (3)	—
Decompression feature is enabled.	—	Yes	Yes	Yes (2)	Yes (3)	—
Encryption feature is enabled.	—	Yes	Yes	Yes (2)	Yes (3)	—
CONF_DONE stays low after a configuration cycle.	—	Yes	Yes (4)	Yes (2)	Yes (3)	—
CONF_DONE goes high momentarily after a configuration cycle.	—	Yes	Yes (5)	—	—	—
FPGA does not enter user mode even though CONF_DONE goes high.	—	—	—	Yes (2)	Yes (3)	—
Configuration failure occurs at the beginning of a configuration cycle.	Yes	—	—	—	—	—
EPCS128	Yes	—	—	—	—	—
Failure in .pof generation for EPC device using Intel	—	—	—	—	—	Yes

continued...

(2) Use only for multi-device chain

(3) Use only for single-device chain

(4) Start with positive offset to the PLC settings

(5) Start with negative offset to the PLC settings

Failure Symptoms	Disable EPCS/ EPCQ ID Check	Disable AS Mode CONF_DONE Error Check	PLC Settings	Post-Chain Bitstream Pad Bytes	Post-Device Bitstream Pad Bytes	Bitslice Padding Value
Quartus Prime Convert Programming File Utility when the decompression feature is enabled.						

1.4. Output File Types

The following table describes the output file types that you can generate with the Programming File Generator or the Converting Programming Files tools:

Table 4. Output File Types

Programming File Type	Extension	Description
Hexadecimal (Intel-Format) Output File for SRAM	`.hexout`	Contains configuration data for use outside the Intel software.
Jam Byte Code File	`.jbc`	Stores programming data for programming, configuring, verifying, and blank-checking one or more devices in a JTAG chain, in binary format. *Note:* `.jbc` conversion is available in the command line.
Jam File	`.jam`	Similar to `.jbc` files, but in ASCII format. *Note:* `.jam` conversion is available in the command line.
JTAG Indirect Configuration File	`.jic`	• Proprietary Intel FPGA file type. • Enables serial flash programming via Intel FPGA JTAG pins. • Works only for Active Serial configuration. • Before programming the flash, the Programmer first configures the FPGA with the Serial Flash Helper Design.
Programmer Object File	`.pof`	• Proprietary Intel FPGA file type. • Allows to program into an external flash, such as programming CFI flash with the PFL IP core via JTAG header, or programming a serial flash via Active Serial header.
Raw Binary File	`.rbf`	You use `.rbf` files for passive configuration mode, such as Passive Serial (PS), Fast Passive Parallel (FPP), or Avalon®-Streaming (AvST) mode. If the design does not use the Intel PFL Intel FPGA IP, then external hosts such as a CPU or microcontroller use this file to configure Intel FPGAs.
Raw Programming Data File	`.rpd`	• For Active Serial configuration. • You can program this file into the serial flash with any third-party programmer or Intel FPGA IP, such as ASMI Parallel or Serial Flash Controller. • The `.rpd` file content has a bit swapped if compared to the output file.
Serial Vector Format File	`.svf`	Stores programming data for programming, verifying, and blank-checking one or more fixed-algorithm devices in a JTAG chain in Automated Test Equipment (ATE)-type programming environments. `.svf` files can do FPGA configuration and flash programming. *Note:* `.svf` conversion is only available in the command line.
Tabular Text File	`.ttf`	Contains configuration data for use outside the Intel Quartus Prime software.

Related Information

- Intel FPGA Configuration Devices Support Page
- File Types Definition
  In *Intel Quartus Prime Help*

1.4.1. Optional Programming or Configuration Files

The Intel Quartus Prime software can generate optional programming or configuration files in formats that you can use with programming tools other than the Intel Quartus Prime Programmer.

When you compile a design in the Intel Quartus Prime software, the Assembler automatically generates either a `.sof` or `.pof` file. The Assembler also allows you to convert FPGA configuration files to programming files for configuration devices.

1.4.2. Secondary Programming Files

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

Table 5. Intel Quartus Prime Software Support for Secondary File Types

File Type	Intel Quartus Prime Software Generate	Intel Quartus Prime Programmer Support
`.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	—
Hexadecimal (Intel-Format) Output File (`.hexout`)	Yes	—
Raw Binary File (`.rbf`)	Yes	Yes [6]
Tabular Text File (`.ttf`)	Yes	—
Raw Programming Data File (`.rpd`)	Yes	—

1.5. Scripting Support

The Intel Quartus Prime software allows generating programming files from the command line. You can incorporate these commands to scripted flows.

[6] The Intel Quartus Prime Programmer supports Raw Binary (`.rbf`) File format in Passive Serial (PS) configuration mode.

1.5.1. quartus_pfg Command Line Tool

The Programing File Generator is also available as the `quartus_pfg` executable. You can specify conversion settings in the command line or through a PFG setting file (`.pfg`). This ability is useful for advanced designs that require multiple images or multiple user data files (HEX/RBF), because you define the settings once in the GUI and then export for subsequent use in the command line.

To export PFG settings to a `.pfg` file, click **File ➤ Save**. The Programming File Generator only saves settings that are consistent.

For more information about the `quartus_pfg` executable, type the following in the command line:

```
quartus_pfg --help
```

Differences Between GUI and Command Line Tool

The command line tool supports single image conversion only.

1.5.2. quartus_cpf Command Line Tool

The Convert Programming Files tool is also available as the `quartus_cpf` command line executable. You can specify conversion settings in the command line or with a conversion setup file (`.cof`).

For help with the `quartus_cpf` executable, type the following at the command line:

```
quartus_cpf --help
```

Related Information

- Convert Programming Files Tool on page 7
- Convert Programming Files Dialog Box
  In *Intel Quartus Prime Help*

1.5.2.1. Generating a Partial-Mask SRAM Object File using a Mask Settings File and a SRAM Object File

- To generate a `.pmsf` file with the `quartus_cpf` executable, typing the following in the command line:

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

Note: The `-p` option is available for designs targeting Intel Arria 10 and Intel Cyclone 10 GX device families.

Related Information

Generating PR Bitstreams for Intel Arria 10 Designs
In *Intel Quartus Prime Pro Edition User Guide: Partial Reconfiguration*

1.6. Generating Programming Files Revision History

Document Version	Intel Quartus Prime Version	Changes
2018.10.09	18.1.0	• Added MAX V to the list of devices that the Programming File Generator tool supports. • Added table : *Device Families that the Convert Programming Files Tool Supports*.
2018.09.24	18.1.0	• Added topic: *quartus_cpf Command Line Tool*. • Stated that the Convert Programming Files dialog box is a legacy tool that supports file conversion for older device families. • In topic: *Output File Types*, specified that the list includes file types generated by the Converting Programming Files tool.
2018.08.07	18.0.0	Reverted document title to *Programmer User Guide: Intel Quartus Prime Pro Edition*.
2018.06.27	18.0.0	• Created the new chapter with information from the *Programming Devices* chapter. • Included information about the Programming File Generator tool.

Related Information

Documentation Archive

For previous versions of the *Intel Quartus Prime Handbook*, search the documentation archives.

2. Programming Intel FPGA Devices

The Intel Quartus Prime Programmer allows you to program and configure Intel CPLD, FPGA, and configuration devices. The files that the Programmer loads into devices are part of the output of a design compilation. After you program the design you can test functionality on a circuit board.

2.1. Stand-Alone Intel Quartus Prime Programmer

Intel FPGA offers the free stand-alone Programmer, which has the same full functionality as the Programmer in the Intel Quartus Prime software.

The stand-alone Programmer is useful when programming devices with another workstation because you do not need extra full licenses. You can download the stand-alone Programmer from the Download Center on the Intel website.

The stand-alone Programmer supports combining or converting Intel Quartus Prime programming files with the **Convert Programming Files** and the **Programming File Generator**. tools. You access both tools from the **File** menu in the Programmer window.

Related Information

Download Center

2.1.1. Stand-Alone Programmer Memory Limitations

The following operations increase memory usage in the stand-alone Programmer:

- Auto-detect
- Adding programming files to the flash memory
- Manually attaching the flash into the Programmer window

In 32-bit Windows, the stand-alone Programmer has the following memory limitations:

Table 6. Stand-Alone Programmer Memory Limitations

Application	Maximum Flash Device Size	Flash Device Operation Using PFL
32-bit Stand-Alone Programmer	Up to 512 Mb	Single Flash Device
64-bit Stand-Alone Programmer	Up to 2 Gb	Multiple Flash Device

2.2. Intel Quartus Prime Programmer Window

The Intel Quartus Prime **Programmer** window allows you to:

- Add your programming and configuration files.
- Specify programming options and hardware.
- Start the programming or configuration of the device.

To open the **Programmer** window, click **Tools ➤ Programmer**. As you proceed through the programming flow, the Intel Quartus Prime **Message** window reports the status of each operation.

Related Information

Programmer Page (Options Dialog Box)
In *Intel Quartus Prime Help*

2.3. Setting Up the Hardware

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

Related Information

AN 425: Using Command-Line Jam STAPL Solution for Device Programming

2.3.1. Editing the Details of an Unknown Device

When the Intel Quartus Prime Programmer automatically detects devices with shared JTAG IDs, the Programmer prompts you to specify the device in the JTAG chain. If the Programmer does not prompt you to specify the device, you must manually add each device in the JTAG chain to the Programmer, and define the instruction register length of each device.

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

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

2.3.2. Setting the JTAG Hardware

The JTAG server allows the Intel Quartus Prime 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.

2.3.2.1. Running JTAG Daemon with Linux

The JTAGD daemon is the Linux version of a JTAG server. The JTAGD daemon allows a remote machine to program or debug boards connected to a Linux host over the network. The JTAGD daemon also allows programs to share JTAG resources.

Running the JTAGD daemon prevents:

- The JTAGD server from exiting after two minutes of idleness.
- The JTAGD server from not accepting connections from remote machines, which might lead to an intermittent failure.

To run JTAGD as a daemon:

1. Create an `/etc/jtagd` directory.
2. Set the permissions of this directory and the files in the directory to allow read/write access.
3. Execute `jtagd` (with no arguments) from the `quartus/bin` directory.

The JTAGD daemon is now running and does not terminate when you log off.

2.3.2.2. 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 by clicking **Tools ➤ JTAG Chain Debugger** on the Intel Quartus Prime software.

Related Information

JTAG Chain Debugger
In *Intel Quartus Prime Help*

2.4. Verifying if Programming Files Correspond to a Compilation of the Same Source Files

Intel Quartus Prime programming files support the project hash property, which helps you determine if two or more programming files correspond to a compilation of the same set of source files.

During compilation, the Intel Quartus Prime software produces a unique project hash, and embeds this value in the programming files (`.sof`).

The project hash does not change for different builds of the Intel Quartus Prime software, or when you install a software update. However, if you upgrade any IP with a different build or patch, the project hash changes.

2.4.1. Obtaining Project Hash for Intel Arria 10 Devices

To obtain the project hash value of a `.sof` programming file for a design targeted to Intel Arria 10 devices, use the `quartus_asm` command-line executable (`quartus_asm.exe` in Windows) with the `--project_hash` option.

```
quartus_asm --project_hash <sof-file>
```

Example 2. Output of Project Hash Command:

In this example, the programming file is `worm.sof`.

```
Info: *******************************************************************
Info: Running Quartus Prime Assembler
Info: Version 17.0.0 Build 288 04/12/2017 SJ Pro Edition
Info: Copyright (C) 2017  Intel Corporation. All rights reserved.
Info: Your use of Intel Corporation's design tools, logic functions
Info: and other software and tools, and its AMPP partner logic
Info: functions, and any output files from any of the foregoing
Info: (including device programming or simulation files), and any
Info: associated documentation or information are expressly subject
Info: to the terms and conditions of the Intel Program License
Info: Subscription Agreement, the Intel Quartus Prime License Agreement,
Info: the Intel MegaCore Function License Agreement, or other
Info: applicable license agreement, including, without limitation,
Info: that your use is for the sole purpose of programming logic
Info: devices manufactured by Intel and sold by Intel or its
Info: authorized distributors.  Please refer to the applicable
Info: agreement for further details.
Info: Processing started: Fri Apr 14 18:01:47 2017
Info: Command: quartus_asm -t project_hash.tcl worm.sof
Info: Quartus(args): worm.sof
0x1ffdc3f47c57bbe0075f6d4cb2cb9deb
Info (23030): Evaluation of Tcl script project_hash.tcl was successful
Info: Quartus Prime Assembler was successful. 0 errors, 0 warnings
Info: Peak virtual memory: 1451 megabytes
Info: Processing ended: Fri Apr 14 18:01:56 2017
Info: Elapsed time: 00:00:09
Info: Total CPU time (on all processors): 00:00:04
```

2.5. 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 Intel Quartus Prime software supports parallel and serial flash loaders.

Related Information

- Generic Serial Flash Interface Intel FPGA IP Core User Guide
- Intel Parallel Flash Loader IP Core User Guide

2.6. Scripting Support

In addition to the Intel Quartus Prime Programmer GUI, you can access programmer functionality from the command line and from scripts with the Intel Quartus Prime command-line executable `quartus_pgm.exe` (or `quartus_pgm` in Linux).

The following command programs a device:

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

Where:

-c byteblasterII	specifies the Intel FPGA Parallel Port Cable download cable
-m jtag	specifies the JTAG programming mode
-o bpv	represents the blank-check, program, and verify operations
design.pof	represents the `.pof` containing the design logic

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

For Linux terminal, use:

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

Related Information

Intel Quartus Prime Scripting
In *Intel Quartus Prime Help*

2.6.1. The jtagconfig Debugging Tool

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

For more information about the `jtagconfig` utility, use the help available at the command prompt:

```
jtagconfig [–h | --help]
```

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

Related Information

- JTAG Chain Debugger
  In *Intel Quartus Prime Help*
- Command Line Scripting
  In *Intel Quartus Prime Pro Edition User Guide: Scripting*

2.7. Programming Intel FPGA Devices Revision History

Table 7. Document Revision History

Date	Intel Quartus Prime Version	Changes
2018.10.09	18.1.0	• Created topic: *Stand-Alone Programmer Memory Limitations* from content in topic: *Stand-Alone Programmer*. • Removed outdated support information.
2018.08.07	18.0.0	Reverted document title to *Programmer User Guide: Intel Quartus Prime Pro Edition*.
2018.06.27	18.0.0	• Moved information about programming file generator to new chapter: *Generating Programming Files*.
2018.05.07	18.0.0	• First release as part of the stand-alone *Programmer User Guide*
2017.05.08	17.0.0	• Added Project Hash feature.
2016.10.31	16.1.0	• Implemented Intel rebranding.
2015.11.02	15.1.0	Changed instances of *Quartus II* to *Intel Quartus Prime*.
2015.05.04	15.0.0	Added Conversion Setup File (.cof) description and example.
December 2014	14.1.0	Updated the Scripting Support section to include a Linux command to program a device.
June 2014	14.0.0	• Added Running JTAG Daemon. • Removed Cyclone III and Stratix III devices references. • Removed MegaWizard Plug-In Manager references. • Updated Secondary Programming Files section to add notes about the Quartus II Programmer support for **.rbf** files.
November 2013	13.1.0	• Converted to DITA format. • Added JTAG Debug Mode for Partial Reconfiguration and Configuring Partial Reconfiguration Bitstream in JTAG Debug Mode sections.
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.

continued...

Date	Intel Quartus Prime Version	Changes
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.

Related Information

Documentation Archive

For previous versions of the *Intel Quartus Prime Handbook*, search the documentation archives.

A. Intel Quartus Prime Pro Edition User Guides

Refer to the following user guides for comprehensive information on all phases of the Intel Quartus Prime Pro Edition FPGA design flow.

Related Information

- Intel Quartus Prime Pro Edition User Guide: Getting Started
  Introduces the basic features, files, and design flow of the Intel Quartus Prime Pro Edition software, including managing Intel Quartus Prime Pro Edition projects and IP, initial design planning considerations, and project migration from previous software versions.
- Intel Quartus Prime Pro Edition User Guide: Platform Designer
  Describes creating and optimizing systems using Platform Designer, a system integration tool that simplifies integrating customized IP cores in your project. Platform Designer automatically generates interconnect logic to connect intellectual property (IP) functions and subsystems.
- Intel Quartus Prime Pro Edition User Guide: Design Recommendations
  Describes best design practices for designing FPGAs with the Intel Quartus Prime Pro Edition software. HDL coding styles and synchronous design practices can significantly impact design performance. Following recommended HDL coding styles ensures that Intel Quartus Prime Pro Edition synthesis optimally implements your design in hardware.
- Intel Quartus Prime Pro Edition User Guide: Design Compilation
  Describes set up, running, and optimization for all stages of the Intel Quartus Prime Pro Edition Compiler. The Compiler synthesizes, places, and routes your design before generating a device programming file.
- Intel Quartus Prime Pro Edition User Guide: Design Optimization
  Describes Intel Quartus Prime Pro Edition settings, tools, and techniques that you can use to achieve the highest design performance in Intel FPGAs. Techniques include optimizing the design netlist, addressing critical chains that limit retiming and timing closure, and optimization of device resource usage.
- Intel Quartus Prime Pro Edition User Guide: Programmer
  Describes operation of the Intel Quartus Prime Pro Edition Programmer, which allows you to configure Intel FPGA devices, and program CPLD and configuration devices, via connection with an Intel FPGA download cable.
- Intel Quartus Prime Pro Edition User Guide: Block-Based Design
  Describes block-based design flows, also known as modular or hierarchical design flows. These advanced flows enable preservation of design blocks (or logic that comprises a hierarchical design instance) within a project, and reuse of design blocks in other projects.

- Intel Quartus Prime Pro Edition User Guide: Partial Reconfiguration
  Describes Partial Reconfiguration, an advanced design flow that allows you to reconfigure a portion of the FPGA dynamically, while the remaining FPGA design continues to function. Define multiple personas for a particular design region, without impacting operation in other areas.
- Intel Quartus Prime Pro Edition User Guide: Third-party Simulation
  Describes RTL- and gate-level design simulation support for third-party simulation tools by Aldec*, Cadence*, Mentor Graphics*, and Synopsys* that allow you to verify design behavior before device programming. Includes simulator support, simulation flows, and simulating Intel FPGA IP.
- Intel Quartus Prime Pro Edition User Guide: Third-party Synthesis
  Describes support for optional synthesis of your design in third-party synthesis tools by Mentor Graphics*, and Synopsys*. Includes design flow steps, generated file descriptions, and synthesis guidelines.
- Intel Quartus Prime Pro Edition User Guide: Third-party Logic Equivalence Checking Tools
  Describes support for optional logic equivalence checking (LEC) of your design in third-party LEC tools by OneSpin*. Describes how to verify the logic equivalence between compilation netlists.
- Intel Quartus Prime Pro Edition User Guide: Debug Tools
  Describes a portfolio of Intel Quartus Prime Pro Edition in-system design debugging tools for real-time verification of your design. These tools provide visibility by routing (or "tapping") signals in your design to debugging logic. These tools include System Console, Signal Tap logic analyzer, Transceiver Toolkit, In-System Memory Content Editor, and In-System Sources and Probes Editor.
- Intel Quartus Prime Pro Edition User Guide: Timing Analyzer
  Explains basic static timing analysis principals and use of the Intel Quartus Prime Pro Edition Timing Analyzer, a powerful ASIC-style timing analysis tool that validates the timing performance of all logic in your design using an industry-standard constraint, analysis, and reporting methodology.
- Intel Quartus Prime Pro Edition User Guide: Power Analysis and Optimization
  Describes the Intel Quartus Prime Pro Edition Power Analysis tools that allow accurate estimation of device power consumption. Estimate the power consumption of a device to develop power budgets and design power supplies, voltage regulators, heat sink, and cooling systems.
- Intel Quartus Prime Pro Edition User Guide: Design Constraints
  Describes timing and logic constraints that influence how the Compiler implements your design, such as pin assignments, device options, logic options, and timing constraints. Use the Interface Planner to prototype interface implementations, plan clocks, and quickly define a legal device floorplan. Use the Pin Planner to visualize, modify, and validate all I/O assignments in a graphical representation of the target device.
- Intel Quartus Prime Pro Edition User Guide: PCB Design Tools
  Describes support for optional third-party PCB design tools by Mentor Graphics* and Cadence*. Also includes information about signal integrity analysis and simulations with HSPICE and IBIS Models.

- Intel Quartus Prime Pro Edition User Guide: Scripting
  Describes use of Tcl and command line scripts to control the Intel Quartus Prime Pro Edition software and to perform a wide range of functions, such as managing projects, specifying constraints, running compilation or timing analysis, or generating reports.