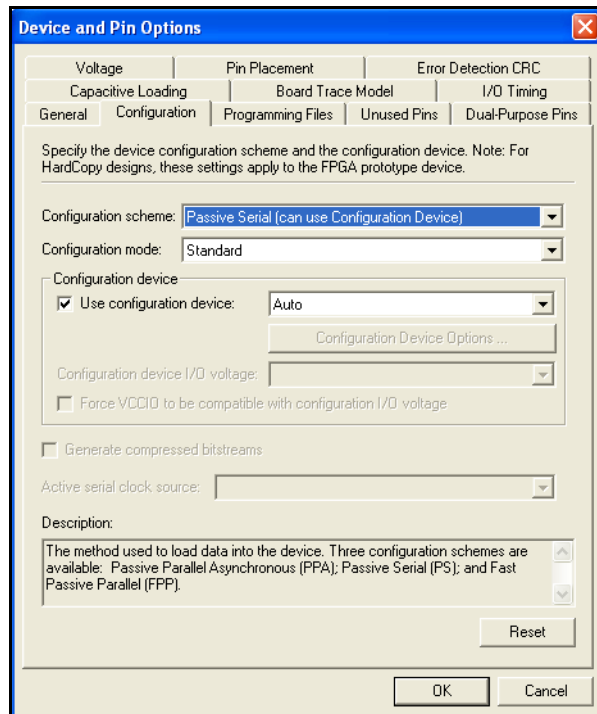


Device configuration options can be set in the **Device & Pin Options** dialog box. To open this dialog box, choose **Device** (Assignments menu), then click on the **Device & Pin Options...** button. You can specify your configuration scheme, configuration mode, and your configuration device used (if applicable) in the **Configuration** tab of the **Device & Pin Options** dialog box (Figure 6–1).

**Figure 6–1. Configuration Dialog Box**



## Configuration Scheme

The **Configuration scheme** drop-down list will change with the chosen device family to only show the configuration schemes supported by that device family. The **Configuration mode** selection is only available for devices that support remote and local update, such as Stratix® and Stratix GX devices. If you are not using remote or local update, you should select **Standard** as your configuration mode. For devices that do not support remote or local update, the **Configuration mode** selection will be greyed out.

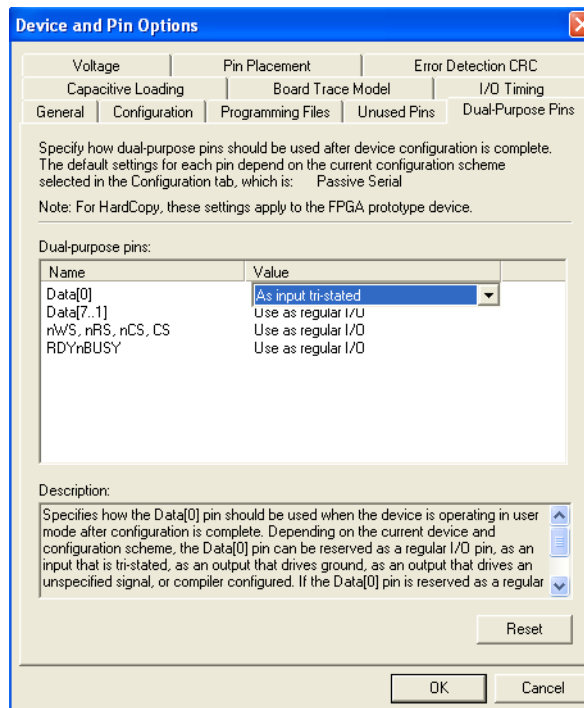
© 2011 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX are Reg. U.S. Pat. & Tm. Off. and/or trademarks of Altera Corporation in the U.S. and other countries. All other trademarks and service marks are the property of their respective holders as described at [www.altera.com/common/legal.html](http://www.altera.com/common/legal.html). Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes 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. 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.

If you are using a configuration device, turn on **Use configuration device** and specify which configuration device you are using. Choosing the configuration device will direct the Quartus® II compiler to generate the appropriate programming object file (.pof). The **Use configuration device** drop-down list will change with the chosen device family to only show the configuration devices that can be used to configure the target device family. If you choose **Auto** as the configuration device, the compiler will automatically choose the smallest density configuration device that fits your design and the **Configuration Device Options...** button will be greyed out.

For more information about configuration device options, refer to the appropriate configuration device data sheet.

You can specify how dual-purpose pins should be used after FPGA configuration is complete through the **Dual-Purpose Pins** tab of the **Device & Pin Options** dialog box. Figure 6-2 shows the **Dual-Purpose Pins** tab for a design that targets a Stratix device and is being configured through PS mode.

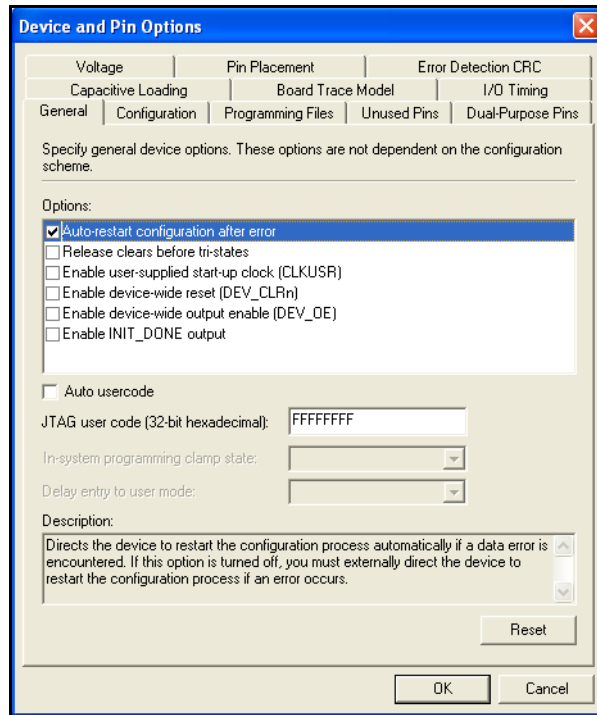
**Figure 6-2. Dual-Purpose Pins Dialog Box**



The drop-down lists will be greyed-out if the pins are not available in the target device family. For the pins that are available, they can be used in one of four states after configuration: as a regular I/O pin, as inputs that are tri-stated, as outputs that drive ground, or as outputs that drive an unspecified signal. If the pin is not used in the mode, the drop-down list will default to the **Use as regular IO** choice. For example, in PS mode, DATA[7..1] are not used, therefore the default usage after configuration for these pins is as a regular I/O pin. If the pin is reserved as a regular I/O, the pin can be used as a regular user I/O after configuration.

You can set device options in the Quartus II software from the **General** tab of the **Device & Pin Options** dialog box (refer to [Figure 6-3](#)).

**Figure 6-3. Configuration Options Dialog Box**



You can set device options in the MAX+PLUS® II development software by choosing **Global Project Device Options** (Assign menu). [Table 6-1](#) summarizes each of these options.

**Table 6-1. Configuration Options (Part 1 of 4)**

Device Option	Option Usage	Option On	Option OFF
Auto-restart configuration after error	<p>When a configuration error occurs, the FPGA drives <math>nSTATUS</math> low, which resets itself internally. The FPGA will release its <math>nSTATUS</math> pin after a reset time-out period. The <math>nSTATUS</math> pin is then pulled to <math>V_{CC}</math> by a pull-up resistor, indicating that reconfiguration can begin.</p> <p>You can choose whether reconfiguration is started automatically or manually (by toggling the <math>nCONFIG</math> pin).</p>	<p>Reconfiguration starts automatically and the system does not need to pulse <math>nCONFIG</math>. After the FPGA releases <math>nSTATUS</math> and it is pulled to <math>V_{CC}</math> by a pull-up resistor, reconfiguration can begin.</p> <p>In passive configuration schemes that use a configuration device, the FPGA's <math>nSTATUS</math> pin is tied to the configuration device's OE pin. Hence, the <math>nSTATUS</math> reset pulse also resets the configuration device automatically. When the FPGA releases <math>nSTATUS</math> and the configuration device releases its OE pin (that is pulled high), reconfiguration begins.</p> <p>For more information about how long the reset time-out period is, refer to the appropriate device family chapters.</p>	<p>The configuration process stops and to start reconfiguration the system must pulse <math>nCONFIG</math> from high-to-low and back high.</p>
Release clears before tri-states	<p>During configuration, the device I/O pins are tri-stated. During initialization, you can choose the order for releasing the tri-states and clearing the registers.</p>	<p>The device releases the tri-states on its I/O pins before releasing the clear signal on its registers.</p>	<p>The device releases the clear signals on its registers before releasing the tri-states. This option allows the design to operate before the device drives out, so all outputs do not start up low.</p>
Enable user-supplied start-up clock (CLKUSR) (APEX™ 20K, APEX II, Cyclone® series, Mercury™, and Stratix series devices only).	<p>This option allows you to select which clock source is used for initialization, either the internal oscillator or external clocks provided on the CLKUSR pin.</p>	<p>The device's internal oscillator (typically 10 MHz) supplies the initialization clock and the FPGA will take care to provide itself with enough clock cycles for proper initialization.</p> <p>The CLKUSR pin is available as a user I/O pin.</p>	<p>The initialization clock must be provided on the CLKUSR pin. This clock can synchronize the initialization of multiple devices. The clock should be supplied when the last data byte is transferred. Supplying a clock on CLKUSR will not affect the configuration process.</p> <p>For more information about how many clock cycles are needed to properly initialize a device, refer to the appropriate device family chapters.</p>

**Table 6–1. Configuration Options (Part 2 of 4)**

Device Option	Option Usage	Option On	Option OFF
<p>Enable user-supplied start-up clock (CLKUSR) (ACEX® 1K, FLEX® 10K, and FLEX 6000 devices only.)</p>	<p>This option allows you to select which clock source is used for initialization, either external clocks provided on the CLKUSR pin or on the DCLK pin.</p>	<p>In PS and PPS schemes, the internal oscillator is disabled. Thus, external circuitry, such as a configuration device or microprocessor, must provide the initialization clock on the DCLK pin. Programming files generated by the Quartus II or MAX+PLUS II software already have these initialization clock cycles included in the file.</p> <p>In the PPA and PSA configuration schemes, the device's internal oscillator (typically 10 MHz) supplies the initialization clock and the FPGA will take care to provide itself with enough clock cycles for proper initialization.</p> <p>The CLKUSR pin is available as a user I/O pin.</p>	<p>The initialization clock must be provided on the CLKUSR pin. This clock can synchronize the initialization of multiple devices. The clock should be supplied when the last data byte is transferred. Supplying a clock on CLKUSR will not affect the configuration process.</p> <p>For more information about how many clock cycles are needed to properly initialize a device, refer to the appropriate device family chapters.</p>
<p>Enable device-wide reset (DEV_CLRn)</p>	<p>Enables a single pin, DEV_CLRn, to reset all device registers.</p>	<p>Chip-wide reset is not enabled. The DEV_CLRn pin is available as a user I/O pin.</p>	<p>Chip-wide reset is enabled for all registers in the device. All registers are cleared when the DEV_CLRn pin is driven low.</p> <p>The DEV_CLRn pin cannot be used to clear only some of the registers; every device register is affected by this global signal.</p>
<p>Enable device-wide output enable (DEV_OE)</p>	<p>Enables a single pin, DEV_OE, to control all device tri-states.</p>	<p>Chip-wide output enable is not enabled. The DEV_OE pin is available as a user I/O pin.</p>	<p>Chip-wide output enable is enabled for all device tri-states. After configuration, all user I/O pins are tri-stated when DEV_OE is low.</p> <p>The DEV_OE pin cannot be used to tri-state only some of the output pins; every output pin is affected by this global signal.</p>


Table 6–1. Configuration Options (Part 3 of 4)

Device Option	Option Usage	Option On	Option OFF
Enable INIT_DONE output	Enables the INIT_DONE pin, which signals the end of initialization and the start of user-mode with a low-to-high transition.	The INIT_DONE signal is not available. The INIT_DONE pin is available as a user I/O pin.	<p>The INIT_DONE signal is available on the open-drain INIT_DONE pin. When <math>nCONFIG</math> is low and during the beginning of configuration, the INIT_DONE pin will be high due to an external pull-up. When the option bit to enable INIT_DONE is programmed into the device (during the first frame of configuration data), the INIT_DONE pin will go low. When initialization is complete, the INIT_DONE pin will be released and pulled high. This low-to-high transition signals the FPGA has entered user mode.</p> <p>The INIT_DONE pin cannot be accessed internally. Therefore, you must loop the INIT_DONE signal back into the device to use the INIT_DONE signal.</p> <p>For more information about the value of the external pull-up resistor, refer to the appropriate device family chapters.</p>
Enable JTAG BST support (FLEX 6000 devices only.)	Enables post-configuration JTAG boundary-scan testing (BST) support in FLEX <sup>®</sup> 6000 devices.	JTAG BST can be performed before configuration; however, it cannot be performed during or after configuration. During JTAG BST, $nCONFIG$ must be held low.	JTAG BST can be performed before or after device configuration via the four JTAG pins (TDI, TDO, TMS, and TCK); however, it cannot be performed during configuration. When JTAG boundary-scan testing is performed before device configuration, $nCONFIG$ must be held low.

**Table 6–1. Configuration Options (Part 4 of 4)**

Device Option	Option Usage	Option On	Option OFF
Generate compressed bitstreams (Stratix II and Cyclone series devices only)	Enables Stratix II and Cyclone series FPGAs to receive compressed configuration bitstreams in AS and PS configuration schemes.	The Quartus II software generates uncompressed programming files and Stratix II and Cyclone series FPGAs do not decompress the configuration data.	The Quartus II software generates compressed programming files and Stratix II and Cyclone series FPGAs decompress the bitstream during configuration.
Auto usercode (Not available in FLEX 6000 devices.)	Allows you to program a 32-bit user electronic signature into the device during programming (typically for design version control). When the USERCODE instruction is loaded into the device, you can shift the signature out of the device.	If this option is off, the JTAG user code option is available and you can specify a 32-bit hexadecimal number for the target device. The JTAG user code is an extension of the option register. This data can be read with the JTAG USERCODE instruction.	Uses the checksum value from the SRAM Object File (.sof) as the JTAG user code. If this option is on, the JTAG user code option is dimmed to indicate that it is not available.

After enhanced configuration and EPC2 device programming you can choose to automatically configure the targeted FPGAs on board. This can be done by selecting the **Initiate Configuration After Programming** option under the **Programmer** section of the **Options** window (**Tools** menu). This option is similar to issuing the INIT\_CONF JTAG instruction, which means nINIT\_CONF of the enhanced configuration or EPC2 devices must be connected to the nCONFIG of the FPGA.

 For more information about the INIT\_CONF JTAG instruction, refer to the *Enhanced Configuration Devices (EPC4, EPC8, and EPC16) Data Sheet* or the *Configuration Devices for SRAM-Based LUT Devices Data Sheet*.

## Document Revision History

Table 6–2 lists the revision history for this chapter.

**Table 6–2. Document Revision History**

Date	Version	Changes
June 2011	2.5	Updated Table 6–1.
December 2009	2.4	<ul style="list-style-type: none"> <li>■ Updated Figure 6–1, Figure 6–2, and Figure 6–3.</li> <li>■ Removed “Referenced Documents” section.</li> </ul>
October 2008	2.3	<ul style="list-style-type: none"> <li>■ Added “Referenced Documents” section.</li> <li>■ Updated new document format.</li> </ul>
April 2007	2.2	Added “Chapter Revision History”.
October 2005	2.1	Removed active cross references referring to document outside Chapter 6.
July 2004	2.0	<ul style="list-style-type: none"> <li>■ Added Stratix II and Cyclone II device information throughout chapter.</li> <li>■ Updated Default Configuration and Modified Configuration of “auto-restart configuration after error” option in Table 6–1.</li> <li>■ Added paragraph regarding Initiate Configuration After Programming option on page 6–9.</li> </ul>
September 2003	1.0	Initial Release.

