Partial Reconfiguration IP Core User Guide

ug-partrecon
2016.10.31
1 Partial Reconfiguration IP Core

1.1 Instantiating the Partial Reconfiguration IP Core in the Qsys Interface

1.2 Instantiating the Partial Reconfiguration IP Core in the Quartus Prime IP Catalog

1.3 Enable Compression

1.3.1 Generating an Encrypted PR Bitstream

1.4 Enable Enhanced Decompression

1.5 Data Compression Comparison

1.6 Bitstream Compatibility Check

1.7 The Importance of Clock-to-Data Ratio (CD Ratio)

1.8 Partial Reconfiguration IP Core Parameters

1.9 Partial Reconfiguration IP Core Ports

1.10 Using the Avalon Memory Mapped Slave Interface

1.11 Avalon Memory Map Slave Interface Data/CSR Memory Map

1.11.1 Interrupt Interface

1.12 FPGA Control Block Interface

1.12.1 Partial Reconfiguration IP Core Timing Specification

1.12.2 Avalon Memory Map Slave Interface Read and Write Transfer Timing

1.13 Freeze Logic for PR Regions

1.14 Data Source Controller

1.15 Standard Partial Reconfiguration Data Interface

1.16 JTAG Debug Mode for Partial Reconfiguration

1.16.1 Configuring Partial Reconfiguration Bitstream in JTAG Debug Mode

1.16.2 Timing Constraints

1.17 Partial Reconfiguration IP Core User Guide Archives

1.18 Revision History
Partial reconfiguration (PR) offers you the ability to reconfigure part of the design's core logic such as LABs, MLABs, DSP, and RAM, while the remainder of the design continues running. The PR IP core provides a standard interface to this ability and eliminates the need to manually instantiate an interface with the PR control block. You can instantiate the PR IP core through Qsys or Qsys Pro, or via the Quartus® Prime IP Catalog. The Quartus Prime software supports the PR feature for the Stratix® V and Arria® 10 device families, as well as Cyclone® V devices whose part number ends in "SC", for example, 5CGXFC9E6F3518NSC.

You can perform partial reconfiguration through either an internal host residing in the core logic, or as an external host via dedicated device pins. The advantage of the internal host is that you store all the logic needed to perform PR on the device, without the need for external devices.

**Figure 1. PR IP Core Components**

When you instantiate the PR IP core, the Main Controller module is instantiated. This module includes the Control Block (CB) Interface Controller, Freeze/Unfreeze Controller, Bitstream Decoder, and the Data Source Controller. A Data Source Interface module provides you with an optional JTAG Debug Interface and PR Data Interface. If you choose to use the PR IP core in internal host mode, the IP core automatically instantiates the corresponding crcblock and prblock WYSIWYG atom primitives.
If used as external host (placed in another FPGA or CPLD), the PR IP core provides the required interface ports. Connect to the dedicated PR pins and CRC_ERROR pin on the target FPGA undergoing partial reconfiguration.

**Figure 2. Managing Partial Reconfiguration with an Internal or External Host**

The figure shows how to connect these blocks to the PR control block (CB). In your system, you include either the external host or the internal host, but not both. During PR, the PR Control Block (CB) is in Passive Parallel x16 programming mode for 28nm devices, and x32 programming mode for the Arria 10 device family. In external host mode, the PR control block is not instantiated in the core of the device undergoing PR, because there is a direct connection from the external PR pins to the internal control block.

You can instantiate the PR IP core as the internal host for all supported devices. When you specify it as the internal host, both `prblock` and `crcblock` WYSIWYG atom primitives are auto-instantiated as part of the design. You can instantiate the PR IP core as the external host on any supported devices, as specified in the device family list.

**Related Links**

- [FPGA Control Block Interface](#) on page 19
  - When you instantiate the PR IP core, you can choose to use it as either an internal host or external host.
- [Data Source Controller](#) on page 23
  - This controller handles the source of PR data, either from JTAG or standard data interface.
- [Standard Partial Reconfiguration Data Interface](#) on page 23
- [JTAG Debug Mode for Partial Reconfiguration](#) on page 23

### 1.1 Instantiating the Partial Reconfiguration IP Core in the Qsys Interface

**Partial Reconfiguration** (PR) is available as a Qsys or Qsys Pro component through the Qsys interface. Instantiate the core as an internal host or an external host.

You can configure the PR IP core to use [Avalon-Streaming](#) and [Conduit](#) interfaces, or an [Avalon-MM](#) interface. Enable the [Avalon-MM](#) interface using the Enable Avalon-MM Slave Interface option. If you use Qsys and want PR included as a component, for example in a design with both Qsys and non-Qsys partitions, you must instantiate the PR IP core in the Qsys interface.
To instantiate the PR IP core with Qsys:
1. Click **Tools ➤ Qsys**.
2. In the **Qsys** interface **IP Catalog**, click **Basic Functions ➤ Configuration and Programming** and select **Partial Reconfiguration**.
3. Configure your IP core variation using the settings appropriate to your design.

**Figure 3. Partial Reconfiguration IP Core in the Qsys Interface**

4. Optionally, turn on **Enable Avalon-MM slave interface** to use the Avalon Memory Map Slave interface rather than the Conduit interface.
5. Turn on **Enable enhanced decompression** to use this optional feature.
6. Select an appropriate clock-to-data ratio for your other options.
7. Click **Finish**.

**Related Links**
- **Enable Enhanced Decompression** on page 9
  For more information on the enhanced decompression feature.
- **The Importance of Clock-to-Data Ratio (CD Ratio)** on page 11
  A proper CD Ratio ensures that the data being processed synchronizes during the PR operation whether you are using compression, encryption, or both, or neither.
- **Partial Reconfiguration IP Core Parameters** on page 12
1.2 Instantiating the Partial Reconfiguration IP Core in the Quartus Prime IP Catalog

Partial Reconfiguration (PR) is available from the IP Catalog. You can choose to instantiate the core as an internal host or an external host.

If you are not using PR as a component of the Qsys interface, then you can instantiate PR with the Quartus Prime IP Catalog.

1. Click Tools ▶ IP Catalog.
2. Click Installed IP ▶ Library ▶ Basic Functions ▶ Configuration and Programming and select Partial Reconfiguration.
3. In the Save IP Variation dialog box, type the name for your partial reconfiguration IP variation. Choose whether to use Verilog or VHDL. Click OK to save your variation.
4. Configure your IP core variation using the settings appropriate to your design.

Figure 4. Partial Reconfiguration IP Core in the IP Catalog

5. Optionally, turn on Enable Avalon-MM slave interface to use the Avalon Memory Map Slave interface rather than the Conduit interface.
6. Turn on **Enable enhanced decompression** to use this optional feature.
7. Select an appropriate clock-to-data ratio for your other options.
8. Click **Finish**.
   The IP Catalog instantiates your IP core variation and displays a completion dialog box.

9. Click **Exit**.

**Related Links**
- **Enable Enhanced Decompression** on page 9
  For more information on the enhanced decompression feature.
- **The Importance of Clock-to-Data Ratio (CD Ratio)** on page 11
  A proper CD Ratio ensures that the data being processed synchronizes during the PR operation whether you are using compression, encryption, or both, or neither.
- **Partial Reconfiguration IP Core Parameters** on page 12

**1.3 Enable Compression**

You can enable bitstream compression (and decompression) to reduce the size of your bitstream by specifying options during programming.

*Note:* The terms "standard" and "enhanced" compression only apply when using PR. Enhanced compression is only available when using the PR IP Core with supported device families.

**Related Links**
- **Enable Enhanced Decompression** on page 9
Enhanced compression (and decompression) can reduce the size of your bitstream at the expense of greater resources use.

- **Generate PR Bitstreams**
  For more information about enabling compression duiring programming refer to the Quartus Prime Handbook.

- **Data Compression Comparison** on page 10
  For more information on comparing compression rates.

- **The Importance of Clock-to-Data Ratio (CD Ratio)** on page 11
  A proper CD Ratio ensures that the data being processed synchronizes during the PR operation whether you are using compression, encryption, or both, or neither.

### 1.3.1 Generating an Encrypted PR Bitstream

To partially reconfigure your device with encrypted bitstream:

1. Create a 256-bit key file (.key).

2. To generate the key programming file (.ekp) from the Quartus Prime shell, type the following command:

   ```bash
   quartus_cpf --key <keyfile>:<keyid> <base_sof_file> <output_ekp_file>
   ```

   For example:

   ```bash
   quartus_cpf --key my_key.key:key1 base.sof key.ekp
   ```

3. To generate the encrypted PR bitstream (.rbf), run the following command:

   ```bash
   quartus_cpf -c <pr_pmsf_file> <pr_rbf_file>
   qcrypt -e --keyfile=<keyfile> --keyname=<keyid> --lockto=<qlk file> --keystore=<battery|OTP> <pr_rbf_file> <pr_encrypted_rbf_file>
   ```

   - **lockto**—specifies the encryption lock.
   - **keystore**—specifies the volatile key (battery) or the non-volatile key (OTP).

   For example:

   ```bash
   quartus_cpf -c top_v1.pr_region.pmsf top_v1.pr_region.rbf
   qcrypt -e --keyfile=my_key.key --keyname=key1 --lockto=qlk --keystore=battery
   top_v1.pr_region.rbf top_v1_encrypted.rbf
   ```

4. To program the key file as volatile key (default) into the device, type the following command:

   ```bash
   quartus_pgm -m jtag -o P;<output_ekp_file>
   ```

   For example:

   ```bash
   quartus_pgm -m jtag -o P;key.ekp
   ```

5. To program the base image into the device, type the following command:

   ```bash
   quartus_pgm -m jtag -o P;<base_sof_file>
   ```
For example:

```
quartus_pgm -m jtag --pr <output_encrypted_rbf_file>
```

For more information on the design security features in Arria 10 devices, refer to Using the Design Security Features in Altera FPGAs.

**Related Links**

Using the Design Security Features in Altera FPGAs

### 1.4 Enable Enhanced Decompression

Enhanced compression (and decompression) can reduce the size of your bitstream at the expense of greater resources use.

**Important:**

Do not use enhanced compression with bitstream encryption.

Enhanced compression is available for both Stratix V and Arria 10 devices. Additionally, you can also use enhanced compression with double-compression.

You can generate a bitstream with enhanced compression in one of two ways:

- In the **Convert Programming Files** dialog box under the properties of a Partial-Masked SRAM Object File (.pmsf)
- From the command line with this command:

```
quartus_cpf -c -o enhanced_bitstream_compression=on <input_filename>.pmsf <output_filename>.rbf
```

**Note:**

Enhanced compression uses the same CD Ratio setting as a plain, uncompressed bitstream.

**Related Links**

- **Instantiating the Partial Reconfiguration IP Core in the Quartus Prime IP Catalog** on page 6
  
  Partial Reconfiguration (PR) is available from the IP Catalog. You can choose to instantiate the core as an internal host or an external host.

- **Instantiating the Partial Reconfiguration IP Core in the Qsys Interface** on page 4
  
  Partial Reconfiguration (PR) is available as a Qsys or Qsys Pro component through the Qsys interface. Instantiate the core as an internal host or an external host.

- **Enable Compression** on page 7
  
  You can enable bitstream compression (and decompression) to reduce the size of your bitstream by specifying options during programming.
• **Generate PR Bitstreams**
  For more information about enabling compression during programming refer to the Quartus Prime Handbook.

• **Data Compression Comparison** on page 10
  For more information on comparing compression rates.

• **The Importance of Clock-to-Data Ratio (CD Ratio)** on page 11
  A proper CD Ratio ensures that the data being processed synchronizes during the PR operation whether you are using compression, encryption, or both, or neither.

### 1.5 Data Compression Comparison

Standard compression results in a 30-45% decrease in RBF size. Use of the enhanced data compression algorithm results in 55-75% decrease in RBF size. The algorithm increases the compression at the expense of additional core area required to implement the compression algorithm.

The following figure shows the compression ratio comparison across PR designs with varying degrees of Logic Element (LE):

**Figure 5. Compression Ratio Comparison between Standard Compression and Enhanced Compression**

![Compression Ratio Comparison Chart](chart.png)

**Related Links**

• [Bitstream Compatibility Check](#) on page 11

• **Enable Enhanced Decompression** on page 9
  Enhanced compression (and decompression) can reduce the size of your bitstream at the expense of greater resources use.

• **Enable Compression** on page 7
  You can enable bitstream compression (and decompression) to reduce the size of your bitstream by specifying options during programming.

• **The Importance of Clock-to-Data Ratio (CD Ratio)** on page 11
  A proper CD Ratio ensures that the data being processed synchronizes during the PR operation whether you are using compression, encryption, or both, or neither.
1.6 Bitstream Compatibility Check

Turn on the **Enable bitstream compatibility check** when instantiating the PR IP core from either Qsys or the IP Catalog for Stratix V or Cyclone V devices. The software then verifies the partial reconfiguration PR Bitstream file (.rbf). If an incompatible bitstream is detected, the PR operation aborts and the status output reports an error.

**Note:** Bitstream compatibility is not currently supported for the Arria 10 device family.

![Bitstream Compatibility Check Diagram](image)

Figure 6. **Bitstream Compatibility Check**

This option prevents you from accidentally corrupting the static region of your design with a bitstream from an incompatible .rbf and risking damage to the chip being programmed.

When you turn on **Enable bitstream compatibility check**, the PR IP core creates a **PR bitstream ID** and displays the bitstream ID in the configuration dialog box.

**Related Links**
- [Data Compression Comparison](#) on page 10
  
  For more information on specifying PR IP Core Parameters
- [Partial Reconfiguration IP Core Parameters](#) on page 12
- [Partial Reconfiguration IP Core Ports](#) on page 14

1.7 The Importance of Clock-to-Data Ratio (CD Ratio)

The proper clock-to-data ratio (CD Ratio) is critical for partial reconfiguration. A proper CD Ratio ensures that the data being processed synchronizes during the PR operation whether you are using compression, encryption, or both, or neither.

The Control Block (CB) interface receives data and sends it during a PR event with the CD Ratio you specify when you instantiate the Partial Reconfiguration IP core.
Table 1. Valid combinations and CD Ratio for Bitstream Encryption and Compression for Stratix V Devices

<table>
<thead>
<tr>
<th>Compressed</th>
<th>Encrypted</th>
<th>CD Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Off</td>
<td>Off</td>
<td>1</td>
</tr>
<tr>
<td>Off</td>
<td>On</td>
<td>2</td>
</tr>
<tr>
<td>On</td>
<td>Off</td>
<td>4</td>
</tr>
<tr>
<td>On</td>
<td>On</td>
<td>4</td>
</tr>
</tbody>
</table>

CD Ratio for the PR IP core in Stratix V designs must be exact for the bitstream type. CD Ratio for plain Programmer Object File (POF) must be 1. CD Ratio for compressed POF must be 2, 4 or 8, depending on the width. Do not specify the CD Ratio as the necessary minimum to support different bitstream types.

Table 2. Valid combinations and CD Ratio for Bitstream Encryption and Compression for Arria 10

<table>
<thead>
<tr>
<th>Configuration Data Width</th>
<th>Enhanced Encryption</th>
<th>Basic Compression</th>
<th>CD Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>x8</td>
<td>Off</td>
<td>Off</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Off</td>
<td>On</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>On</td>
<td>Off</td>
<td>1</td>
</tr>
<tr>
<td>x16</td>
<td>Off</td>
<td>Off</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Off</td>
<td>On</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>On</td>
<td>Off</td>
<td>2</td>
</tr>
<tr>
<td>x32</td>
<td>Off</td>
<td>Off</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Off</td>
<td>On</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>On</td>
<td>Off</td>
<td>4</td>
</tr>
</tbody>
</table>

For designs that target the Arria 10 device family, the CD Ratio value is the minimum required ratio.

Note: Standard encryption uses the same CD Ratio setting as a plain, uncompressed bitstream.

Related Links
- Instantiating the Partial Reconfiguration IP Core in the Quartus Prime IP Catalog on page 6
  For more information on PR IP core instantiation in Qsys.
- Instantiating the Partial Reconfiguration IP Core in the Qsys Interface on page 4
  For more information on PR IP core instantiation in the IP Catalog.

1.8 Partial Reconfiguration IP Core Parameters
### IP Core Option

<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use as PR internal host</strong></td>
<td>On or Off</td>
<td>On</td>
<td>Turn on this option to use the PR IP core as an internal host. Both prblock and CRCblock WYSIWYG atom primitives are auto-instantiated as part of your design. Disable this option to use the PR IP core as an external host. You must connect additional interface signals to the dedicated PR pins if you use the PR IP core as an external host.</td>
</tr>
<tr>
<td><strong>Enable JTAG debug mode</strong></td>
<td>On or Off</td>
<td>On</td>
<td>To perform partial reconfiguration turn on this option to access the PR IP core with the Programmer.</td>
</tr>
<tr>
<td><strong>Enable Avalon-MM slave interface</strong></td>
<td>On or Off</td>
<td>Off</td>
<td>Turn on this option to use the Avalon Memory-Mapped (Avalon-MM) slave interface.</td>
</tr>
<tr>
<td><strong>Enable interrupt interface</strong></td>
<td>On or Off</td>
<td>Off</td>
<td>Enable this option to use the interrupt interface. You can only enable this interface if you turned on the Enable Avalon-MM slave interface parameter.</td>
</tr>
<tr>
<td><strong>Enable bitstream compatibility check</strong></td>
<td>On or Off</td>
<td>Off</td>
<td>Turn on this option to check the bitstream compatibility during PR operations for external host. The bitstream compatibility check feature is always enabled for PR internal host. Specify the PR bitstream ID value if you enable this option for PR external host.</td>
</tr>
<tr>
<td><strong>PR bitstream ID</strong></td>
<td>-2147483648 to 2147483647</td>
<td>0</td>
<td>Specifies a signed 32-bit integer value of the PR bitstream ID for external host. This value must match the PR bitstream ID generated during compilation for the target PR design. You can find the PR bitstream ID value of the target PR design in the Assembler compilation report (.asm.rpt).</td>
</tr>
<tr>
<td><strong>Input data width</strong></td>
<td>1, 8, 16, or 32</td>
<td>16</td>
<td>Specifies the data width in bits. This option affects the data bus width.</td>
</tr>
<tr>
<td><strong>Target device family for partial reconfiguration</strong></td>
<td>Arria 10, Cyclone V, Stratix V</td>
<td>&quot;Stratix V&quot;</td>
<td>Select the target device family for partial reconfiguration when you use the PR IP core as external host. Note: This option is ignored for PR internal host.</td>
</tr>
<tr>
<td><strong>Clock-to-Data ratio</strong></td>
<td>Cyclone V or Stratix V: 1, 2, or 4, Arria 10: 1, 2, 4, or 8</td>
<td>1</td>
<td>Specifies the ratio between PR clock and PR data.</td>
</tr>
<tr>
<td><strong>Divide error detection frequency by</strong></td>
<td>1, 2, 4, 8, 16, 32, 64, 128, or 256</td>
<td>1</td>
<td>Only available when you use the IP core as an internal host. The CRCblock WYSIWYG atom primitive is auto-instantiated as part of the design. Specifies the divide value of the internal clock, which determines the frequency of the error detection cyclic redundancy check (CRC). The divide value must be a power of two. Refer to the device handbook to find the frequency of the internal clock for the selected device.</td>
</tr>
</tbody>
</table>

---

1. Not currently supported for the Arria 10 device family.
<table>
<thead>
<tr>
<th>IP Core Option</th>
<th>Value</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto-instantiate CRC block</td>
<td>On or Off</td>
<td>On</td>
<td>This option is only applicable for an internal host. Disable this option to manually instantiate a CRC block.</td>
</tr>
<tr>
<td>Auto-instantiate PR block</td>
<td>On or Off</td>
<td>On</td>
<td>This option is only applicable for an internal host. Disable this option to manually instantiate a PR block.</td>
</tr>
<tr>
<td>Enable enhanced decompression</td>
<td>On or Off</td>
<td>Off</td>
<td>Enable this option is to use the enhanced decompressor. Decompress bitstreams that are compressed with the enhanced compression algorithm.</td>
</tr>
</tbody>
</table>

**Related Links**
- [Instantiating the Partial Reconfiguration IP Core in the Quartus Prime IP Catalog](#) on page 6
  For more information on PR IP core instantiation in Qsys.
- [Instantiating the Partial Reconfiguration IP Core in the Qsys Interface](#) on page 4
  For more information on PR IP core instantiation in the IP Catalog.
- [Using the Avalon Memory Mapped Slave Interface](#) on page 17
- [Avalon Memory Map Slave Interface Read and Write Transfer Timing](#) on page 21
  For more information on the timing specification for the Avalon Memory Mapped Slave interface.
- [Bitstream Compatibility Check](#) on page 11
- [The Importance of Clock-to-Data Ratio (CD Ratio)](#) on page 11
  A proper CD Ratio ensures that the data being processed synchronizes during the PR operation whether you are using compression, encryption, or both, or neither.

### 1.9 Partial Reconfiguration IP Core Ports

#### I/O Port List for PR IP Core

**Table 3. Clock/Reset Ports**

<table>
<thead>
<tr>
<th>Port Name</th>
<th>Width</th>
<th>Direction</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>nreset</td>
<td>1</td>
<td>Input</td>
<td>Asynchronous reset for the PR IP core. Resetting the PR IP core during a partial reconfiguration operation can cause the device to lock up.</td>
</tr>
<tr>
<td>clk</td>
<td>1</td>
<td>Input</td>
<td>User input clock to the PR IP core. This signal is ignored during JTAG debug operations. The input clock must be free-running.</td>
</tr>
</tbody>
</table>

**Note:** These options are always available.

**Table 4. Conduit Interface**

<table>
<thead>
<tr>
<th>Port Name</th>
<th>Width</th>
<th>Direction</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>freeze</td>
<td>1</td>
<td>Output</td>
<td>Active high signal used to freeze the PR interface signals of the region undergoing partial reconfiguration. De-assertion of this signal indicates the end of PR operation.</td>
</tr>
</tbody>
</table>
### Table 5. Conduit Interface

<table>
<thead>
<tr>
<th>Port Name</th>
<th>Width</th>
<th>Direction</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>pr_start</td>
<td>1</td>
<td>Input</td>
<td>A signal arriving at this port asserted high initiates a PR event. You must assert this signal high for a minimum of one clock cycle and de-assert it low prior to the end of the PR operation. This makes the PR IP core ready to accept the next pr_start trigger event when the freeze signal is low. The PR IP core ignores this signal during JTAG debug operations.</td>
</tr>
<tr>
<td>data[]</td>
<td>1, 8, 16, or 32</td>
<td>Input</td>
<td>Selectable input PR data bus width, either x1, x8, x16, or x32. Once a PR event is triggered, it is synchronous with the rising edge of the clk signal whenever the data_valid signal is high and the data_ready signal is high. The PR IP core ignores this signal during JTAG debug operations.</td>
</tr>
<tr>
<td>data_valid</td>
<td>1</td>
<td>Input</td>
<td>A signal arriving at this port asserted high indicates the data[] port contains valid data. The PR IP core ignores this signal during JTAG debug operations.</td>
</tr>
<tr>
<td>data_ready</td>
<td>1</td>
<td>Output</td>
<td>A signal arriving at this port asserted high indicates the PR IP core is ready to read the valid data on the data[] port whenever the data_valid signal is asserted high. The data sender must stop sending valid data if this port is low. This signal deasserts low during JTAG debug operations.</td>
</tr>
<tr>
<td>status[2..0]</td>
<td>1</td>
<td>Output</td>
<td>A 3-bit error output used to indicate the status of PR event. Once an error is detected (PR_ERROR, CRC_ERROR, or incompatible bitstream error), this signal latches high and only resets at the beginning of the next PR event, when pr_start is high and freeze is low. For example: 3'b000 – power-up or nreset asserted 3'b010 – CRC_ERROR was triggered 3'b011 – Incompatible bitstream error detected 3'b100 – PR operation in progress 3'b101 – PR operation passed 3'b110 – Reserved 3'b111 – Reserved</td>
</tr>
<tr>
<td>double_pr</td>
<td>1</td>
<td>Input</td>
<td>When the pr_start signal is triggered, until the de-assertion of a freeze signal, a signal asserted high on this port indicates the PR event requires double PR cycle. A low signal on this port indicates a single PR cycle event. If your PR design targets a Stratix V device and requires the use of double PR because you have initialized RAM in the PR region, you must assert the double_pr input signal high.</td>
</tr>
</tbody>
</table>

*Note:* This option is always available.
This assertion ensures that the controller handles double PR properly. If you are instantiating the PR IP in a design that is not using initialized on-chip RAMs, connect this port to 0. You must assert this signal high if the PR bitstream (.rbf) is generated with the Write memory contents option turned on. Failure to do so causes a PR_ERROR assertion during partial reconfiguration. The PR IP core ignores this signal during JTAG debug operations.

Note: These options are available when Enable Avalon-MM slave interface parameter is turned Off.

### Table 6. Avalon-MM Slave Interface

<table>
<thead>
<tr>
<th>Port Name</th>
<th>Width</th>
<th>Direction</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>avmm_slave_address</td>
<td>1</td>
<td>Input</td>
<td>Avalon-MM address bus. The address bus is in the unit of Word addressing. Refer to the Qsys Component section for more details on the address mapping. The PR IP core ignores this signal during JTAG debug operations.</td>
</tr>
<tr>
<td>avmm_slave_read</td>
<td>1</td>
<td>Input</td>
<td>Avalon-MM read control. The PR IP core ignores this signal during JTAG debug operations.</td>
</tr>
<tr>
<td>avmm_slave_readdata</td>
<td>16 or 32</td>
<td>Output</td>
<td>Avalon-MM read data bus. The PR IP core ignores this signal during JTAG debug operations.</td>
</tr>
<tr>
<td>avmm_slave_write</td>
<td>1</td>
<td>Input</td>
<td>Avalon-MM write control. The PR IP core ignores this signal during JTAG debug operations.</td>
</tr>
<tr>
<td>avmm_slave_writedata</td>
<td>16 or 32</td>
<td>Input</td>
<td>Avalon-MM write data bus. The PR IP core ignores this signal during JTAG debug operations.</td>
</tr>
<tr>
<td>avmm_slave_waitrequest</td>
<td>1</td>
<td>Output</td>
<td>Asserted to indicate that the IP is busy. Also indicates that the IP core is unable to respond to a read or write request. This signal is pulled high during JTAG debug operations.</td>
</tr>
</tbody>
</table>

Note: These options are available when Enable Avalon-MM Slave Interface parameter is turned On.

### Table 7. Interrupt Interface

<table>
<thead>
<tr>
<th>Port Name</th>
<th>Width</th>
<th>Direction</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>irq</td>
<td>1</td>
<td>Output</td>
<td>The interrupt signal.</td>
</tr>
</tbody>
</table>

Note: This option is available when Enable interrupt interface parameter is turned On.

---

2 Double PR is only supported for Cyclone V SC devices and Stratix V devices. This port is not used in Arria 10 devices.
Table 8. **CRCBLOCK Interface**

These options are available when *Use as PR Internal Host* parameter is turned *Off* or the CRCBLOCK is instantiated manually for an internal host.

<table>
<thead>
<tr>
<th>Port Name</th>
<th>Width</th>
<th>Direction</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>crc_error_pin</td>
<td>1</td>
<td>Input</td>
<td>Available when you use the PR IP core as an External Host. Connect this port to the dedicated CRC_ERROR pin of the FPGA undergoing partial reconfiguration.</td>
</tr>
</tbody>
</table>

Table 9. **PR Block Interface**

<table>
<thead>
<tr>
<th>Port Name</th>
<th>Width</th>
<th>Direction</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>pr_ready_pin</td>
<td>1</td>
<td>Input</td>
<td>Connect this port to the dedicated PR_READY pin of the FPGA undergoing partial reconfiguration.</td>
</tr>
<tr>
<td>pr_error_pin</td>
<td>1</td>
<td>Input</td>
<td>Connect this port to the dedicated PR_ERROR pin of the FPGA undergoing partial reconfiguration.</td>
</tr>
<tr>
<td>pr_done_pin</td>
<td>1</td>
<td>Input</td>
<td>Connect this port to the dedicated PR_DONE pin of the FPGA undergoing partial reconfiguration.</td>
</tr>
<tr>
<td>pr_request_pin</td>
<td>1</td>
<td>Output</td>
<td>Connect this port to the dedicated PR_REQUEST pin of the FPGA undergoing partial reconfiguration.</td>
</tr>
<tr>
<td>pr_clk_pin</td>
<td>1</td>
<td>Output</td>
<td>Connect this port to the dedicated DCLK of the FPGA undergoing partial reconfiguration.</td>
</tr>
<tr>
<td>pr_data_pin[15 ..0]</td>
<td>16</td>
<td>Output</td>
<td>Connect this port to the dedicated DATA[15 ..0] pins of the FPGA undergoing partial reconfiguration.</td>
</tr>
</tbody>
</table>

*Note:* These options are available when *Use as PR Internal Host* parameter is turned *Off* or when the PRBLOCK is instantiated manually for an internal host.

**Related Links**

- [Bitstream Compatibility Check](#) on page 11
- [Avalon Memory Map Slave Interface Data/CSR Memory Map](#) on page 18
- [Freeze Logic for PR Regions](#) on page 22
  - When partially reconfiguring a design, freeze all the outputs of each PR region to a known constant value. This freezing prevents the signal receivers in the static region from receiving undefined signals during the partial reconfiguration process.
- [FPGA Control Block Interface](#) on page 19
  - When you instantiate the PR IP core, you can choose to use it as either an internal host or external host.

### 1.10 Using the Avalon Memory Mapped Slave Interface

Perform partial reconfiguration through the Avalon®-MM Slave interface by following these steps:
1. Avalon Memory Mapped Master component writes 0x01 (or 0x03 if the design requires double PR) to IP address offset 0x1 to trigger PR operation.

2. Avalon Memory Mapped Master component writes PR bitstream to IP address offset 0x0 until all the PR bitstream is written.

3. Avalon Memory Mapped Master component reads the data from IP address offset 0x1 to check the status[2:0] value. Optionally, the Avalon-MM Master component can read the status[2:0] of this IP during a PR operation to detect any early failure, for example, PR_ERROR.

Related Links
- **Partial Reconfiguration IP Core Parameters** on page 12
  For more information on PR IP core parameters.
- **Avalon Memory Map Slave Interface Read and Write Transfer Timing** on page 21
  For more information on the timing specification for the Avalon Memory Mapped Slave interface.

### 1.11 Avalon Memory Map Slave Interface Data/CSR Memory Map

**Table 10. Data/CSR Memory Map Format**

<table>
<thead>
<tr>
<th>Name</th>
<th>Address Offset</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PR_DATA</td>
<td>0x0</td>
<td>Write</td>
<td>Every data write to this address indicates this bitstream was sent to the IP core. Performing a read on this address returns all 0's.</td>
</tr>
<tr>
<td>PR_CSR</td>
<td>0x1</td>
<td>Read or Write</td>
<td>Control and status registers.</td>
</tr>
</tbody>
</table>

**Table 11. PR_CSR Control and Status Registers**

<table>
<thead>
<tr>
<th>Bit Offset</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Read and write control register for pr_start signal. Refer to Input/Output Port List section for more details on pr_start signal. pr_start = PR_CSR[0] PR_CSR[0] is de-asserted to value &quot;0&quot; by the IP core automatically one clock cycle after it is asserted. This streamlines the flow so you do not need to manually assert and de-assert this register to control pr_start signal.</td>
</tr>
<tr>
<td>1</td>
<td>Read and write control register for double_pr signal. double_pr = PR_CSR[1]</td>
</tr>
<tr>
<td>2-4</td>
<td>Read only status register for status[2:0] signal. PR_CSR[4:2] = status[2:0]</td>
</tr>
<tr>
<td>5</td>
<td>Read and write bit for interrupt. If the interrupt interface is enabled, reading this bit returns the value of the irq signal. Writing a &quot;1&quot; clears the interrupt. If the interrupt interface is disabled, reading this bit always returns a value of &quot;0&quot;.</td>
</tr>
<tr>
<td>6-15 or 6-31</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

3. Double PR is only supported for Cyclone V and devices.

4. Depending on the Avalon Memory Mapped data bus width.
Related Links

- **Avalon Memory Map Slave Interface Read and Write Transfer Timing** on page 21
  The Avalon-MM interface supports read and write transfers with a slave-controlled `waitrequest`. You can cause the slave to stall the interconnect for as many cycles as required by asserting the `waitrequest` signal. If a slave uses `waitrequest` for either read or write transfers, it must use `waitrequest` for both.

- **Partial Reconfiguration IP Core Ports** on page 14

### 1.11.1 Interrupt Interface

If you enable the Avalon Memory Mapped Slave interface, you can use the optional interrupt interface.

The IP core asserts `irq` during the following events.

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>3'b001</td>
<td>PR_ERROR occurred.</td>
</tr>
<tr>
<td>3'b010</td>
<td>CRC_ERROR occurred.</td>
</tr>
<tr>
<td>3'b011</td>
<td>The IP core detects an incompatible bitstream.</td>
</tr>
<tr>
<td>3'b101</td>
<td>The result of a successful PR operation.</td>
</tr>
</tbody>
</table>

After `irq` asserts the master performs one or more of the following:
- Query for the status of the PR IP core; `PR_CSR[4:2]`
- Carry out some action, such as error reporting.
- Once the interrupt is serviced, clear the interrupt by writing a "1" to `PR_CSR[5]`.

### 1.12 FPGA Control Block Interface

When you instantiate the PR IP core, you can choose to use it as either an internal host or external host.
- When you use the PR IP core as an internal host, it automatically instantiates the corresponding device `CRCBLOCK` and `PRBLOCK` WYSIWYG atom primitives.
- When you use the PR IP core as external host (placed in another FPGA or CPLD), the PR IP core provides the `CRCBLOCK` and `PRBLOCK` interface ports so you can connect the host to the dedicated PR pins and CRC_ERROR pin on the target FPGA being partially reconfigured.

**Related Links**

**Partial Reconfiguration IP Core Ports** on page 14

---

5 When using enhanced compression, Stratix V devices are limited to a width of 16. Arria 10 devices can use widths of 1, 8, 16, or 32.
1.12.1 Partial Reconfiguration IP Core Timing Specification

The following timing diagram illustrates a successful Partial Reconfiguration IP core operation. You determine whether the operations passes or fails with the status[2:0] output signal.

The PR operation is initiated when you assert the pr_start signal. Monitor the status[] or freeze signals to detect the end of the PR operation.

Figure 7. Partial Reconfiguration Timing

Note:

- You must assert pr_start signal high for a minimum of one clock cycle to initiate PR, Deassert pr_start before sending the last data.
- status[] signal is reset when pr_start is asserted. This signal changes during a PR operation if any error such as a CRC_ERROR, PR_ERROR, or bitstream incompatibility error is detected.
- status[] signal changes after a PR operation if CRC_ERROR is detected and no error happens during the previous PR operation.
- The data_valid signal is not required to be asserted at the same time as the pr_start. You can provide the data[] and assert data_valid when appropriate.
- You can either drive the data_valid signal low after sending the last data, or continue to assert data_valid high with dummy data on data[] until the end of PR, when freeze is driven low or status[] is updated.
- data[] is transferred only when data_valid and data_ready are asserted on the same cycle. Do not drive new data on the data bus, when both data_valid and data_ready are not high.
- The data_ready signal is driven low once the PR IP core receives the last data.

The data[], data_valid, and data_ready signals comply with the Avalon-ST specification for Data Transfer with Backpressure. The PR IP Core acts as a sink, with readLatency set to 0. For more information, refer to the Avalon Interface Specifications.
Important: The PR_CLK signal has a different nominal maximum frequency for each device. Most Stratix V devices have a nominal maximum frequency of at least 62.5 MHz. Arria 10 devices have a maximum rate of 100 MHz.

Related Links
Avalon Interface Specifications for Data Transfer with Backpressure

1.12.2 Avalon Memory Map Slave Interface Read and Write Transfer Timing

The Avalon-MM interface supports read and write transfers with a slave-controlled waitrequest. You can cause the slave to stall the interconnect for as many cycles as required by asserting the waitrequest signal. If a slave uses waitrequest for either read or write transfers, it must use waitrequest for both.

A slave typically receives address, read or write, and writedata after the rising edge of the clock. A slave asserts waitrequest before the rising clock edge to hold off transfers. When the slave asserts waitrequest, the transfer is delayed. The address and control signals are held constant. Transfers complete on the rising edge of the first clk after the slave port deasserts waitrequest.

Figure 8. Read and Write Transfers for Avalon-MM Slave Interface

The numbers in this timing diagram, mark the following transitions:
1. address and read are asserted after the rising edge of clk. waitrequest is asserted stalling the transfer.
2. waitrequest is sampled. Because waitrequest is asserted, the cycle becomes a wait-state. address, read, and write remain constant.
3. The slave presents valid readdata and deasserts waitrequest.
4. readdata and deasserted waitrequest are sampled, completing the transfer.
5. address, writedata, and write signals are asserted. The slave responds by asserting waitrequest stalling the transfer.
6. Deassert the waitrequest.
7. The slave captures writedata and ends the transfer.

Related Links
Avalon Memory Mapped Interfaces
For more information on read and write transfers with Avalon Memory Mapped Interfaces

**1.13 Freeze Logic for PR Regions**

When partially reconfiguring a design, freeze all the outputs of each PR region to a known constant value. This freezing prevents the signal receivers in the static region from receiving undefined signals during the partial reconfiguration process. Freezing is important for control signals that you drive from the PR region.

*Note:* For Arria 10 devices, there is no requirement to freeze the global and non-global inputs of a PR region.

*Important:* For Cyclone V or Stratix V devices require that you freeze global and non-global inputs of a PR region.

**Figure 9. Freeze Logic for Arria 10 Devices**

The Partial Reconfiguration IP core includes a freeze port for a single freeze signal that corresponds to the device you configure. When instantiating the IP core in your design, combine this freeze port with your system-level PR control logic to freeze the PR region output. If your design has multiple PR regions, create a decoding logic to route that freeze signal to the appropriate PR region’s freeze logic. Do not route the freeze signal to the PR regions unaffected by the current partial reconfiguration.

*Note:* If you are not using the Partial Reconfiguration IP core in your design, include logic to generate the freeze signal that you use for freezing the PR region outputs.

The static region logic must be independent of all the outputs from the PR regions for a continuous operation. Control the outputs of the PR regions by creating an RTL wrapper around the PR region.

**Related Links**

- Partial Reconfiguration IP Core Timing Specification on page 20
  
  The following timing diagram illustrates a successful Partial Reconfiguration IP core operation. You determine whether the operations passes or fails with the status[2:0] output signal.
1.14 Data Source Controller

This controller handles the source of PR data, either from JTAG or standard data interface.

The JTAG interface takes precedence over the standard PR data interface. For example, whenever JTAG is engaged through command from Quartus Prime Programmer tool, the PR data is sourced from the JTAG interface rather than the PR data interface.

1.15 Standard Partial Reconfiguration Data Interface

The PR data interface provides you with selectable input data width; x1, x8, x16, and x32. The data interface is connected to ASMI_PARALLEL as well as the Avalon interface to obtain PR data from on-chip RAM, external flash device, or PR over PCIe.

For Cyclone V and Stratix V devices, if the input data width is other than x16, the PR IP core includes a data upsize or downsize module so that the data output to the Data Source Controller is always x16.

For Arria 10 devices, the output of the Data Source Controller is x16 or x32 depending on the input data width and JTAG setting.

1.16 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 PR bitstream and eventually helping you in your PR design prototyping. This feature is available for internal and external host. Using the JTAG debug mode forces the Data Source Controller to be in x16 mode.

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

Note: The TCK is the main clock source for PR IP core during this operation.

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

The Quartus Prime Programmer can detect the number of PR_DONE instruction(s) in plain or compressed PR bitstream and, therefore, can handle single or double PR cycle accordingly. However, only single PR 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 PR IP core in the design).
Note: Configuring an incompatible PR bitstream to the specified device may corrupt your design, including the routing path and the PR IP core placed in the static region. When this issue occurs, the PR IP core stays in an undefined state, and the Quartus Prime Programmer is unable to reset the IP core. As a result, the Quartus Prime Programmer generates the following error when you try to configure a new PR 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.

1.16.1 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 Prime Programmer GUI, right click on a highlighted base bitstream (in .sof) and then click Add PR Programming File to add the PR bitstream (.rbf).
Figure 10. Adding PR Programming File

2. After adding the PR bitstream, you can change or delete the Partial Reconfiguration programming file by clicking **Change PR Programming File** or **Delete PR Programming File**.
3. Click **Start** to configure the PR bitstream. The Quartus Prime Programmer generates an error message if the specified device does not contain the PR IP core in the design (you must instantiate the Partial Reconfiguration IP core in your design to use the JTAG debug mode).
4. Configure the valid `.rbf` in JTAG debug mode with the Quartus Prime Programmer.
5. The JTAG debug mode is also supported if the PR IP core is pre-programmed on the specified device.
Figure 14. Partial Reconfiguration IP Core Successfully Pre-programmed

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

Figure 15. Configuring Corrupted .rbf
1.16.2 Timing Constraints

The Partial Reconfiguration IP core contains an SDC file to properly constrain the IP core during place and route.

This SDC automatically detects the clock connected to the IP core and the JTAG clock when the JTAG debug mode is enabled. You can disable the PR IP Core SDC script with the Create Timing Constraints File option in the IP Catalog if the script is not required.

To detect the clocks, the SDC file for the PR IP must be read after any SDC that creates the clocks used by the IP core. You can facilitate this order by ensuring the .ip file for the PR IP core comes after any .ip files or SDC files used to create these clocks in the QSF file for your Quartus revision. In addition, when using the JTAG debug mode of the IP core, a modification is required to the standard JTAG SDC file template described in the JTAG Signals topic. Specifically, the constraint to set the JTAG TCK clock asynchronous to all other clocks must be disabled to ensure the PR IP core SDC can constrain the TCK clock correctly. Once constrained, the PR IP SDC sets the TCK clock asynchronous to all other required clocks.

Note: The following line must be disabled in the standard JTAG template.

```plaintext
set_clock_groups -asynchronous -group {altera_reserved_tck}
```

1.17 Partial Reconfiguration IP Core User Guide Archives

If an IP core version is not listed, the user guide for the previous IP core version applies.

<table>
<thead>
<tr>
<th>IP Core Version</th>
<th>User Guide</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.0</td>
<td>Partial Reconfiguration IP Core User Guide</td>
</tr>
<tr>
<td>15.1</td>
<td>Partial Reconfiguration IP Core User Guide</td>
</tr>
<tr>
<td>15.0</td>
<td>Partial Reconfiguration IP Core User Guide</td>
</tr>
<tr>
<td>14.1</td>
<td>Partial Reconfiguration IP Core User Guide</td>
</tr>
</tbody>
</table>

1.18 Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Changes</th>
</tr>
</thead>
</table>
| Oct 2016 | 2016.10.31 | Added support for Arria 10 devices.  
• Updated content for Clock-to-Data Ratio.  
• Updated the Parameters and Ports.  
• Updated Compression instructions for standard and enhanced compression.  
• Added supporting content on using PR with JTAG.  
• Minor edits and fixed typos |
| May 2016 | 2016.05.02 | Minor changes:  
• Fixed a link to the Designing for Partial Reconfiguration chapter in Vol 1 of the Quartus Prime Handbook.  
• Fixed typos. |
| Nov 2015 | 2015.11.20 | Revised the following topics: continued... |
# Partial Reconfiguration IP Core

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>• Partial Reconfiguration IP Core Parameters—updated supported data width</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Partial Reconfiguration IP Core Timing Specification—revised the timing diagram</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Deprecated the Sample Partial Reconfiguration IP Core as an External Host on the Same Device topic</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Added Enable Enhanced Compression topic</td>
</tr>
<tr>
<td>May 2015</td>
<td>2015.05.04</td>
<td>Revised the following topics:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Partial Reconfiguration IP Core Parameters—added new parameters for device family support</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Partial Reconfiguration IP Core Ports—added new port options</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Partial Reconfiguration IP Core Timing Specification—revised the timing diagram</td>
</tr>
<tr>
<td>January 2015</td>
<td>2015.01.29</td>
<td>Minor error corrections.</td>
</tr>
<tr>
<td>August 2014</td>
<td>2014.08.20</td>
<td>• Added Avalon Memory Map slave interface</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Updated Ports and Parameters to support Avalon Memory Map slave interface</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Added Bitstream compatibility checking</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Added sample pseudo-code for creating a freeze wrapper for multiple PR regions and creating an external host on the same device.</td>
</tr>
<tr>
<td>November 2013</td>
<td>2013.11.04</td>
<td>Initial release</td>
</tr>
</tbody>
</table>