# altlvds Megafunction

# User Guide

101 Innovation Drive
San Jose, CA 95134
www.altera.com

Software Version 7.0
Document Version: 3.2
Document Date: March 2007

UG-MF9504-3.2

# Contents

## Chapter 3. Specifications

# About this User Guide

## Revision History

The following below shows the revision history for the chapters in this User Guide.

| Date | Document Version | Changes Made |
|---|---|---|
| March 2007 | 3.2 | Updated for Quartus II version 7.0, including Cyclone® III information. |
| December 2006 | 3.1 | Updated Table 1–1 to include Stratix® III information. |
| November 2006 | 3.0 | Updated for Quartus II version 6.1. |
| June 2006 | 2.0 | Updated for Quartus II version 6.0. |
| August 2005 | 1.1 | Minor content changes. |
| December 2004 | 1.0 | Initial release. |

## How to Contact Altera

For the most up-to-date information about Altera® products, refer to the following table.

| Information Type | Contact *(1)* |
|---|---|
| Technical support | www.altera.com/mysupport/ |
| Technical training | www.altera.com/training/custrain@altera.com |
| Product literature | www.altera.com/literature |
| Altera literature services | literature@altera.com |
| FTP site | ftp.altera.com |

***Note to table:***
(1) You can also contact your local Altera sales office or sales representative.

# Typographic Conventions

This document uses the typographic conventions shown below.

| Visual Cue | Meaning |
|---|---|
| **Bold Type with Initial Capital Letters** | Command names, dialog box titles, checkbox options, and dialog box options are shown in bold, initial capital letters. Example: **Save As** dialog box. |
| **bold type** | External timing parameters, directory names, project names, disk drive names, filenames, filename extensions, and software utility names are shown in bold type. Examples: $\mathbf{f_{MAX}}$, **\qdesigns** directory, **d:** drive, **chiptrip.gdf** file. |
| *Italic Type with Initial Capital Letters* | Document titles are shown in italic type with initial capital letters. Example: *AN 75: High-Speed Board Design.* |
| *Italic type* | Internal timing parameters and variables are shown in italic type. Examples: $t_{PIA}$, $n + 1$.<br><br>Variable names are enclosed in angle brackets (< >) and shown in italic type. Example: *<file name>*, *<project name>***.pof** file. |
| Initial Capital Letters | Keyboard keys and menu names are shown with initial capital letters. Examples: Delete key, the Options menu. |
| "Subheading Title" | References to sections within a document and titles of on-line help topics are shown in quotation marks. Example: "Typographic Conventions." |
| `Courier type` | Signal and port names are shown in lowercase Courier type. Examples: `data1`, `tdi`, `input`. Active-low signals are denoted by suffix `n`, e.g., `resetn`.<br><br>Anything that must be typed exactly as it appears is shown in Courier type. For example: `c:\qdesigns\tutorial\chiptrip.gdf`. Also, sections of an actual file, such as a Report File, references to parts of files (e.g., the AHDL keyword `SUBDESIGN`), as well as logic function names (e.g., `TRI`) are shown in Courier. |
| 1., 2., 3., and<br>a., b., c., etc. | Numbered steps are used in a list of items when the sequence of the items is important, such as the steps listed in a procedure. |
| ■ ● • | Bullets are used in a list of items when the sequence of the items is not important. |
| ✓ | The checkmark indicates a procedure that consists of one step only. |
|  | The hand points to information that requires special attention. |
|  | A caution calls attention to a condition or possible situation that can damage or destroy the product or the user's work. |
|  | A warning calls attention to a condition or possible situation that can cause injury to the user. |
| ↵ | The angled arrow indicates you should press the Enter key. |
|  | The feet direct you to more information on a particular topic. |

# Chapter 1. About this Megafunction

## Device Family Support

The altlvds megafunctions support the following target Altera® device families:

- Stratix® III
- Stratix II
- Stratix II GX
- Stratix
- Stratix GX
- Cyclone® III
- Cyclone II
- Cyclone
- HardCopy® II
- HardCopy Stratix
- MAX® II
- MAX 7000AE
- MAX 7000B
- MAX 7000S
- MAX 3000A
- ACEX 1K®
- APEX™ II
- APEX 20KC
- APEX 20KE
- FLEX 10K®
- FLEX® 10KA
- FLEX 10KE
- FLEX 6000

## Introduction

As design complexities increase, use of vendor-specific IP blocks has become a common design methodology. Altera provides parameterizable megafunctions that are optimized for Altera device architectures. Using megafunctions instead of coding your own logic saves valuable design time. Additionally, the Altera-provided functions may offer more efficient logic synthesis and device implementation. You can scale the megafunction's size by setting parameters.

## Features

The altlvds megafunctions implement either an LVDS deserializer receiver or an LVDS serializer transmitter and offer many additional features:

- Parameterizable data channel widths
- Parameterizable serializer/deserializer (SERDES) factors
- Registered inputs and outputs
- Dynamic Phase Alignment (DPA) mode support in Stratix III, Stratix II GX, Stratix II and Stratix GX receivers
- Ability to share fast phase-locked loops (PLL) between transmitter and receivers
- PLL control signals
- Have flexibility to implement in dedicated circuitry or in logic cells. This varies according to the device.
- Support for separate PLL in Stratix III, Stratix II, Stratix II GX, Cyclone III, Cyclone II, Cyclone, and HardCopy II devices

## General Description

The altlvds megafunctions are provided in the Quartus® II software MegaWizard® Plug-In Manager. The altlvds megafunctions instantiate either an LVDS transmitter (altlvds_tx) or an LVDS receiver (altlvds_rx). The altlvds_tx megafunction implements a serialization transmitter, and the altlvds_rx megafunction implements a deserialization receiver. These megafunctions can be used to take advantage of the dedicated SERDES circuitry for high-speed differential data transfer applications in Stratix III, Stratix II, Stratix II GX, Stratix GX, Stratix, HardCopy II, and HardCopy devices. The altlvds megafunctions can also be used to implement SERDES circuitry using logic elements (LEs) and PLLs in Cyclone III, Cyclone II and Cyclone devices. By using the altlvds megafunctions, you can customize and control LVDS data received or transmitted to many source synchronous channels. The altlvds megafunctions have features that are unique to each supported device family.

To achieve high data transfer rates, the Stratix III, Stratix II, Stratix II GX, Stratix GX, Stratix, HardCopy II, and HardCopy devices support True-LVDS™ differential I/O interfaces that have dedicated SERDES circuitry for each differential I/O pair. Refer to the specific device data sheets for transmitting/receiving data rates for respective devices.

On the receiver side, the high-frequency clock generated by the PLL shifts the serial data through a shift register (also called the deserializer). The parallel data is clocked out to the logic array that is synchronized with the low frequency clock.

On the transmitter side, the parallel data from the logic array is first clocked into a parallel-in, serial-out shift register that is synchronized with the low-frequency clock and then transmitted out by the output buffers.

The DPA circuitry supports multiple SERDES factors. Each channel has its own DPA circuit that provides independent data alignment for each channel; therefore, DPA can reduce channel-to-channel skew as well as clock-to-channel skew.

Cyclone series (Cyclone III, Cyclone II, and Cyclone) devices do not support the dedicated DPA circuitry.

Cyclone series (Cyclone III, Cyclone II, and Cyclone) devices also allow the SERDES implementation at certain data rates for a specific device. Refer to the specific device data sheets for supported data rates. For the LVDS transmitter and receiver, the altlvds megafunctions implement serialization and deserialization using LEs and PLLs.

### Common Applications

The SERDES interface is used to transmit and receive high-speed differential data. This enables moving data from board-to-board or box-to-box with great efficiency. This high performance consumes a minimum of power, is relatively immune to noise, and is cost effective. Typical applications of LVDS technology are common in PC/computing (such as flat panel displays and monitor links), telecom/datacom systems (such as routers, hubs, and switches) and other common consumer and commercial applications (such as set-top boxes and in-flight entertainment).

For additional information about common applications supported, and for full details on the I/O standards and high-speed protocols supported, refer to the applicable data sheet or device Handbook for each device family.

## Resource Utilization and Performance

Every instantiation of altlvds uses one fast PLL. The Quartus II software properly configures the PLL according to the settings you apply in the altlvds wizard. Stratix III, Stratix II, Stratix II GX, Cyclone III, Cyclone II, Cyclone, HardCopy II, and HardCopy Stratix devices provide the option to use an external PLL, which requires you to enter the appropriate PLL parameters.

Refer to the *altpll Megafunction User Guide* on the Literature page at **www.altera.com** for more information on the PLL parameters. All Stratix families support the **Use Shared PLL(s) for Receiver and Transmitter** option to place both the LVDS transmitter and the LVDS receiver in the same device I/O bank. The Quartus II software lets the transmitter and receiver share the same fast PLL when both use the same input clock frequency. Although you must separate the transmitter and receiver modules in your design, the Quartus II software merges the fast PLLs, when appropriate, and gives you the following message:

```
Info: Receiver fast PLL <lvds_rx PLL name> and transmitter
fast PLL <lvds_tx PLL name> are merged together
```

The Quartus II software displays the following message when it cannot merge the fast PLLs for the LVDS transmitter and receiver pair in the design:

```
Warning: Can't merge transmitter-only fast PLL
<lvds_tx PLL name> and receiver-only fast PLL
<lvds_rx PLL name>
```

For Stratix device families, the side I/O banks contain dedicated SERDES circuitry, which includes the fast PLLs, serial shift registers, and parallel registers. The transmit and receive functions use varying numbers of LEs depending on the number of channels and serialization and deserialization factors. For best performance, these LEs must be placed in the columns as close to the SERDES circuitry and LVDS pins as possible. This is done by the Quartus II software automatically during place-and-route.

Cyclone III, Cyclone II, and Cyclone devices use DDIO registers as part of the SERDES interface. Since data is clocked on both the rising and falling edge, the clock frequency must be half the data rate; therefore, the PLL runs at half the frequency of the data rate. The core clock frequency for the transmitter is data rate divided by the serialization factor (J). For the Odd serialization factors, depending on the output clock-divide factor (B) and device family, an optional core clock frequency of data rate divided by two times the serialization factor is also available (J).

Use Table 1–1 and Table 1–2 to determine the clock and data rate relationships.

*Table 1–1. Cyclone III, Cyclone II, and Cyclone altlvds Receiver Clock Relationships*

| Clock Type | J = Even | J = Odd |
|---|---|---|
| Fast Clock | Data Rate / 2 | Data Rate / 2 |
| Slow Clock (outclock) | Data Rate / J | Data Rate / J |

*Table 1–2. Cyclone III, Cyclone II, and Cyclone altlvds Transmitter Clock Relationships*

| Clock Type | J = Even | J = Odd |
|---|---|---|
| Fast Clock | Data Rate / 2 | Data Rate / 2 |
| Slow Clock (outclock) | Data Rate / 2 * B | Data Rate / 2 * B |
| Core Clock | Data Rate / J | Data Rate / J |

The Quartus II software reports the number of LEs used per altlvds function in the **Fitter Resource Utilization by Entity** section within the **Resource** section of the Compilation Report file.

# 2. Getting Started

## Software and System Requirements

The instructions in this section require Quartus® II software version 7.0 or higher.

For operating system support information, refer to:

http://www.altera.com/support/software/os_support/oss-index.html

## MegaWizard Plug-In Manager Customization

The MegaWizard® Plug-In Manager helps you create or modify design files that contain custom altlvds variations, which you can then instantiate in a design file. These custom altlvds variations are based on Altera-provided megafunctions that are optimized to use device resources in the most efficient manner. The MegaWizard Plug-In Manager wizard asks questions about the values you want to set for the custom altlvds parameters and about the optional ports you want to use.

The MegaWizard Plug-In Manager automatically generates a Component Declaration file (**.cmp**) used in VHDL Design Files (**.vhd**), and AHDL Include files (**.inc**) used in Text Design Files (**.tdf**) and Verilog Design Files (**.v**). The MegaWizard Plug-In Manager also creates a sample instantiation template with the extension **_inst.tdf** for AHDL designs, **_inst.vhd** for VHDL designs, and **_inst.v** for Verilog HDL designs; the program also creates a sample instantiation declaration file **_bb.v** for Verilog HDL designs. The sample instantiation files contain module and port declarations for the custom megafunction variations.

The MegaWizard Plug-In Manager is the most efficient way to customize your altlvds functions. Stratix® III, Stratix II GX, Stratix II, Stratix GX, Stratix, HardCopy II, and HardCopy® devices each have different features available when using the dedicated serializer/deserializer (SERDES) circuitry. Cyclone® III, Cyclone II, and Cyclone devices do not have dedicated SERDES circuitry; therefore, the functions are implemented in registers using logic elements. The wizard displays only the features that are appropriate to the selected device family. Within each selected device, certain features are available, depending on the modes that you specify.

The MegaWizard Plug-In Manager creates or modifies design files that contain custom megafunction variations that can then be instantiated in a design file. The MegaWizard Plug-In Manager provides a wizard that lets you specify options for the altlvds megafunction features in the design.

Start the MegaWizard Plug-In Manager using one of the following methods:

- On the Tools menu, click **MegaWizard Plug-In Manager**.
- When working in the Block Editor, click **MegaWizard Plug-In Manager** in the **Symbol** dialog box.
- Start the stand-alone version of the MegaWizard Plug-In Manager by typing the following command at the command prompt:
  `qmegawiz↵`

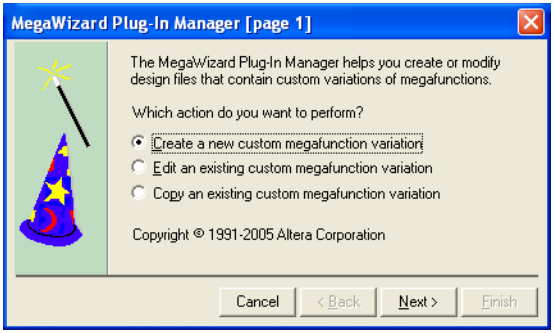
## Using the MegaWizard Plug-In Manager

This section describes the options available on the individual pages of the altlvds wizard. For information about how to use the MegaWizard Plug-In Manager for Stratix III or Cyclone III devices, refer to the section "Options for Stratix III and Cyclone III Devices" on page 2–12.

The first two pages of the megafunction are the same for Stratix series, HardCopy II, HardCopy, Cyclone series, APEX™ II, APEX20KC, and APEX 20KE devices.

Page 1 of the MegaWizard Plug-In Manager is shown in Figure 2–1.

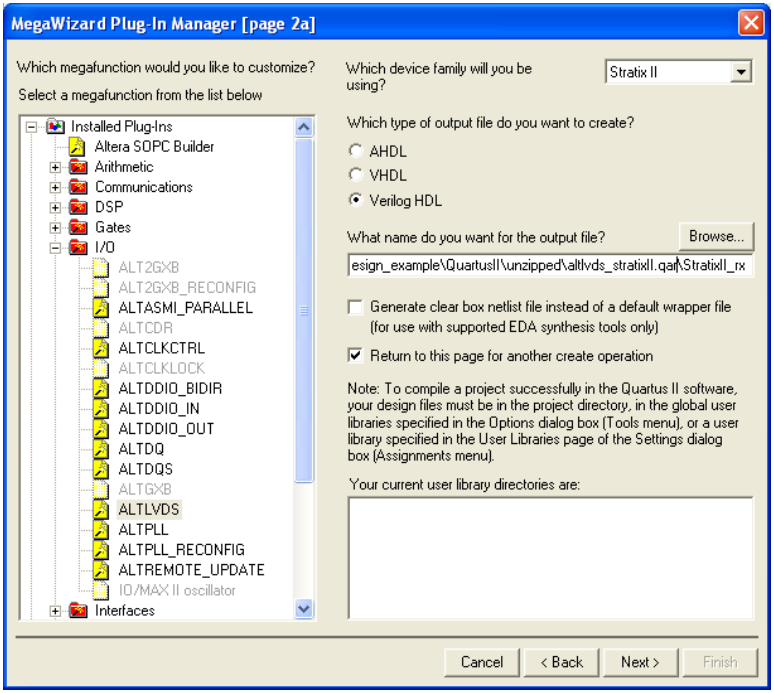
***Figure 2–1. MegaWizard Plug-In Manager [page 1]***




You can choose to create, edit, or copy a custom megafunction variation.

Page 2a lets you create a new custom megafunction variation, as shown in Figure 2–2.

*Figure 2–2. MegaWizard Plug-In Manager [page 2a]*




On page 2a, select the desired megafunction in the megafunction list. The altlvds megafunction is in the **I/O** folder. Specify a file name for your custom megafunction and specify a directory. You can place the megafunction in your project directory or use another directory that you add to your project as a User Library. Select the output file format and choose a device family (alternately, you can specify the device family on page 3 of the wizard).

Pages 2b and 2d are for editing or copying an existing custom megafunction variation. Each page has a directory selection box so you can browse to the custom megafunction variation that you want to edit or copy. Once your selection is made, the wizard will continue to page 3.

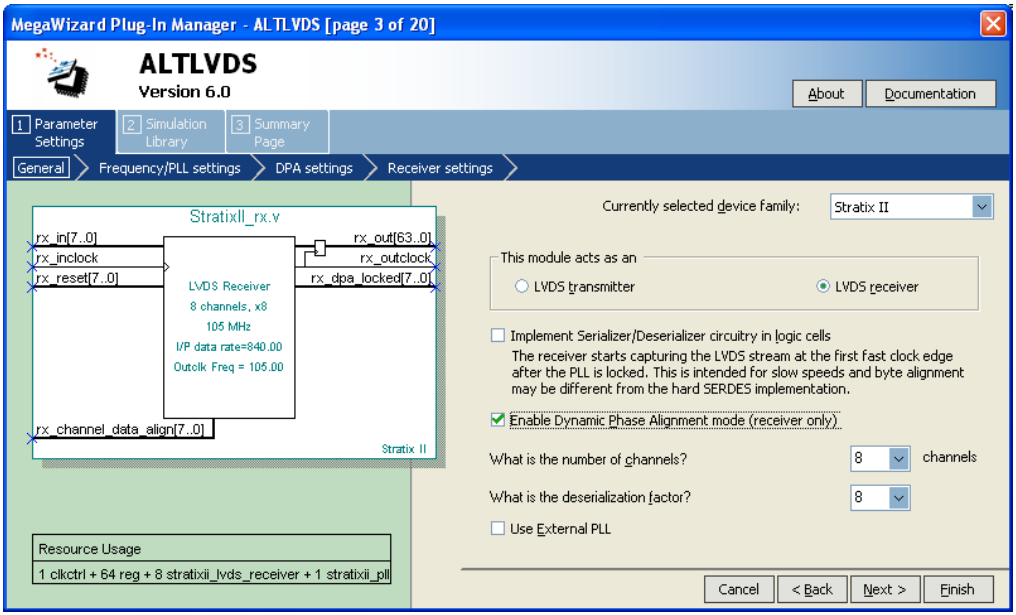
The selections you make on page 3 dictate the format of the remaining pages. For example:

- The LVDS transmit pages have the same appearance for Stratix II, Stratix, Stratix II GX, Stratix GX, Cyclone II, and Cyclone devices. The details of the transmit pages are described

in "Generate an altlvds Receiver and altlvds Transmitter" on page 2–46.

- The Stratix II device LVDS receiver pages are also described in "Generate an altlvds Receiver and altlvds Transmitter" on page 2–46.
- The Stratix GX device LVDS receiver pages with Dynamic Phase Alignment (DPA) mode turned on are described in "Stratix GX DPA-Mode Receiver MegaWizard Page Descriptions" on page 2–8. The LVDS receiver pages for Stratix, Stratix GX (non-DPA mode), Cyclone II, and Cyclone devices have the same appearance and are described in Figure 2–3.

*Figure 2–3. MegaWizard Plug-In Manager—altlvds [page 3 of 20]*




The device family that you select determines the ports and parameters that are available on this and the following pages. For example, the **Enable Dynamic Phase Alignment mode (receiver only)** and **Use External PLL** options are not available for all Stratix series devices.

More information about External PLL Options can be found in the application note *AN 409: Design Example Using the altlvds Megafunction & the External PLL Option in Stratix II Devices*, available at **www.altera.com** on the Altera Literature page under **Application Notes**.

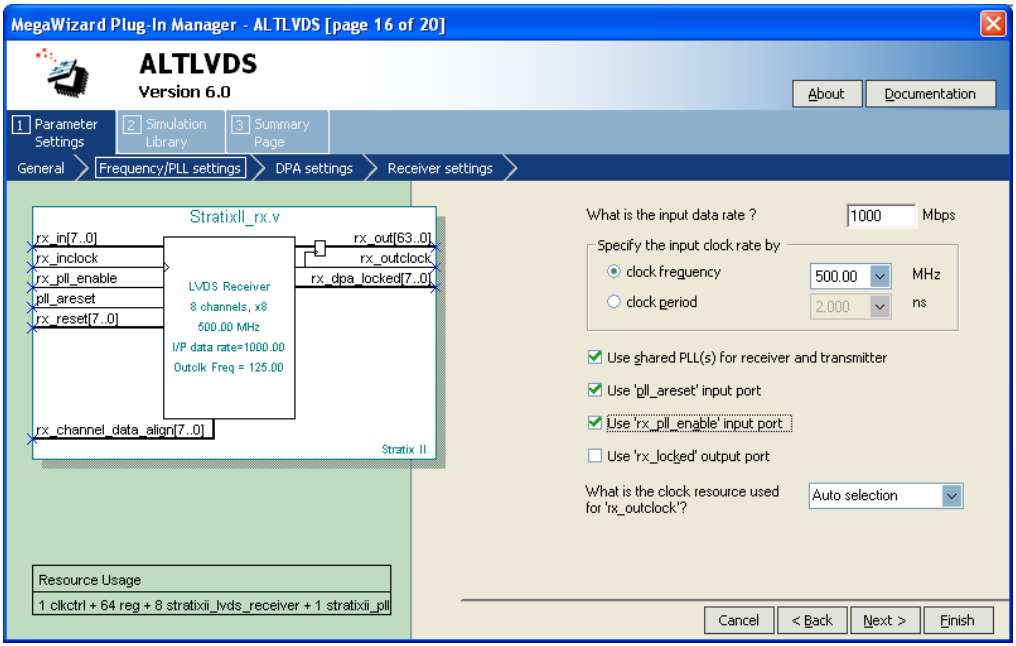
You must also specify whether this megafunction is an LVDS transmitter or receiver, the number of channels, and the deserialization factor. The number of channels you select changes the width of the `rx_in` port, and the deserialization factor changes the width of the `rx_out` port. If the required number of channels is not available in the list, type the desired number in the **What is the number of channels?** box. The list for the deserialization factor is device-dependent. The left side of the MegaWizard Plug-In Manager shows a schematic representation of the custom function variation that you are creating. The schematic is updated as the ports and parameters are changed.

For online help assistance, click the **Documentation** button to view the Quartus II Help file for the LVDS transmit or receive megafunction.

On page 16 of the MegaWizard, enter the input data rate (Figure 2–4). The input clock rates available depend on the data rate entered. Choose the appropriate clock for your design from the **clock frequency** or **clock period** lists.

The other options on this page are described in Table 3–6 on page 3–14.

*Figure 2–4. MegaWizard Plug-In Manager—altlvds (Dedicated SERDES) [page 16 of 20]*

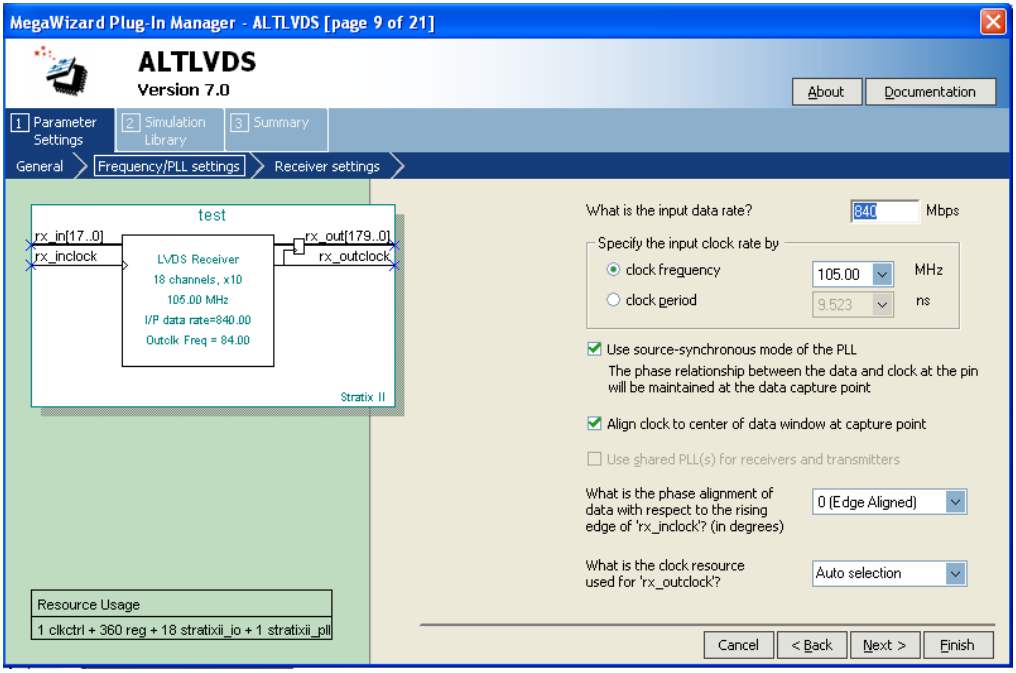


For Stratix II (LE Implementation) and Cyclone II, with the Quartus II 7.0 release there is the option to customize these configurations. The option is **Use source-synchronous mode of the PLL**. This option automatically does the required phase adjustments to guarantee a consistent relationship between the clock and data at the capture register and at the pin.

This should always be used, unless you have already performed all of the necessary phase adjustments manually. Altera recommends that you enable this option when using flexible LVDS schemes (Figure 2–5).

When the **Align clock to center of data window at capture point** is enabled, a phase shift of 90 degrees is added to the clock to center the clock in the data. Use this parameter when the `PLL_OPERATION_MODE` parameter value is set to `SOURCE_SYNCHRONOUS` for Cyclone II and Stratix II (LE Implementation) (Figure 2–5).

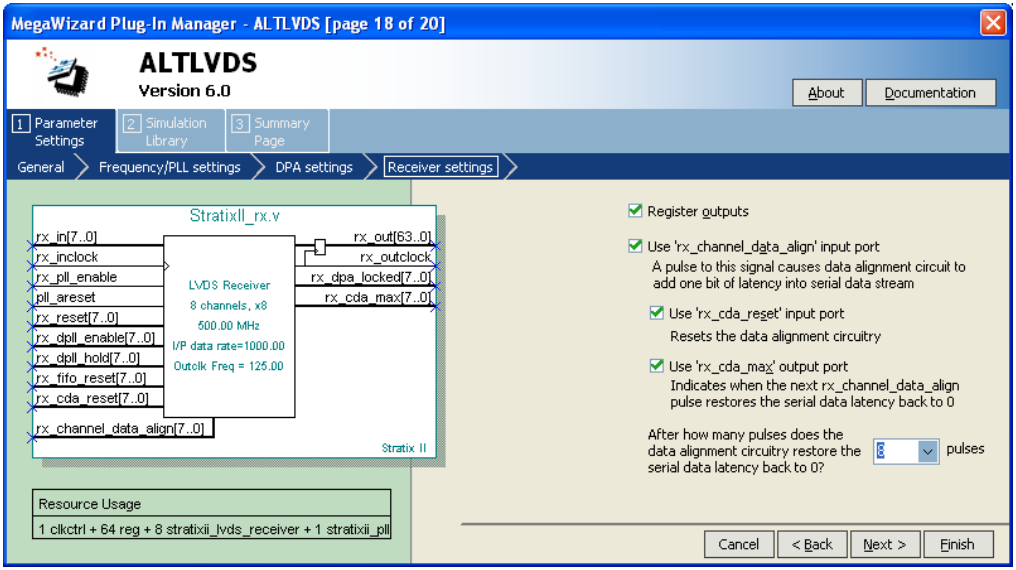
*Figure 2–5. MegaWizard Plug-In Manager—altlvds (Flexible LVDS) [page 9 of 21]*




On page 18 (shown in Figure 2–6), turn ports in your megafunction on or off. Turning on **Register outputs** adds registers to the `rx_out` port. If **Register outputs** is turned off, a warning appears that states the `rx_out`

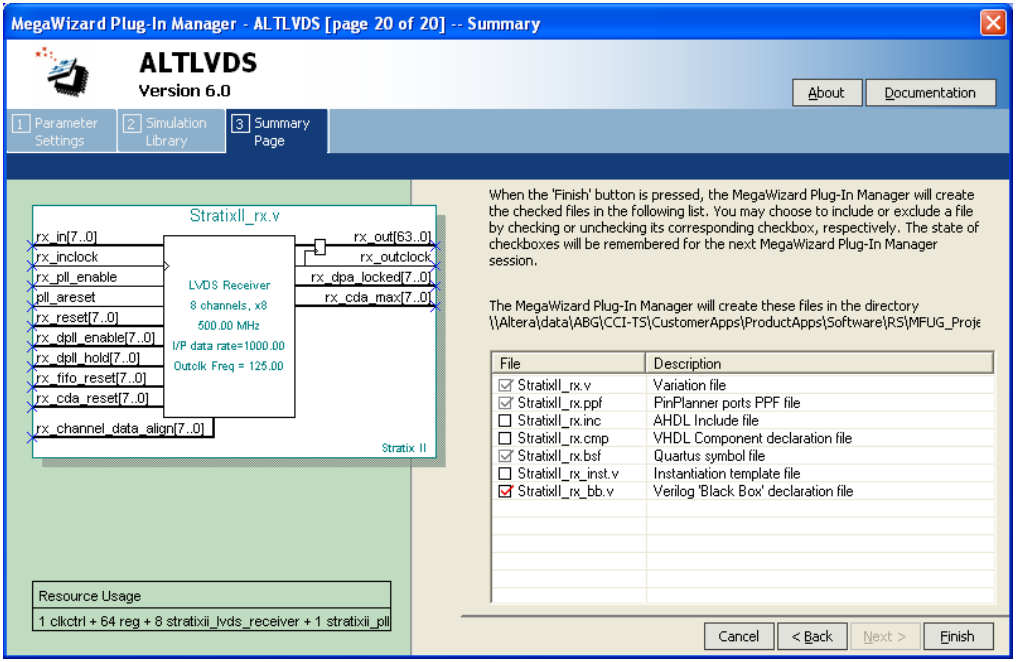
port must be registered in the logic fed by the receiver and a 'Source Multiply" assignment from the receiver to the output registers must be specified with a value equal to the deserialization factor. All other ports are described in Table 3–4 on page 3–10 and Table 3–5 on page 3–13.

*Figure 2–6. MegaWizard Plug-In Manager—altlvds [page 18 of 20]*




Page 20 shows a summary of the files that the MegaWizard Plug-In Manager creates for this instance of your megafunction (Figure 2–7). The variation file type changes depending on the output file you select in page 2 of the MegaWizard Plug-In Manager. Turn on files you need for your design approach. Click **Finish** to generate the selected files. The files are now ready to be used in your design.

*Figure 2–7. MegaWizard Plug-In Manager—altlvds [page 20 of 20] — Summary*

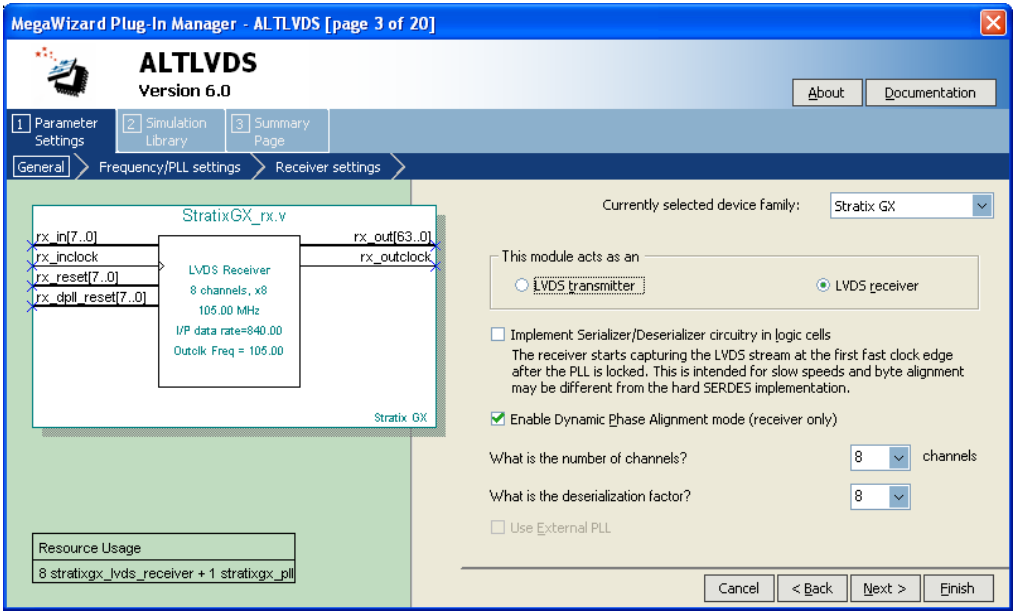



## Stratix GX DPA-Mode Receiver MegaWizard Page Descriptions

Stratix II, Stratix GX, and HardCopy II devices support dynamic phase alignment (DPA) for the LVDS receiver function. The appearance of the MegaWizard Plug-In Manager is different for these devices when DPA mode is turned on. Details for the Stratix II GX device MegaWizard Plug-In Manager pages for the altlvds receiver with DPA are in "Generate an altlvds Receiver and altlvds Transmitter" on page 2–46. This section describes the Stratix GX device MegaWizard Plug-In Manager pages for the altlvds receiver using DPA.

Page 3 of the wizard has the same appearance as all of the other altlvds supported devices and is shown in Figure 2–8.

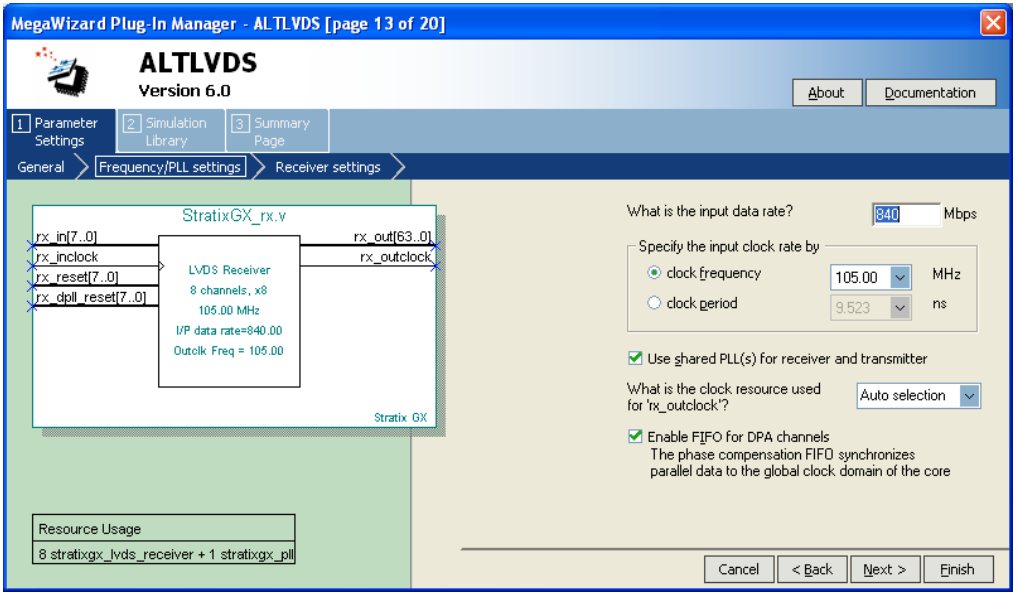
*Figure 2–8. MegaWizard Plug-In Manager—altlvds [page 3 of 20]*




The **Enable Dynamic Phase Alignment mode (receiver only)** option is available only when Stratix II and Stratix GX is chosen from the **Use which device family?** list. When **Enable Dynamic Phase Alignment mode (receiver only)** is turned on, pages 13 (Figure 2–9) and 14 of the MegaWizard Plug-In Manager show the available ports and parameters for DPA mode. The number of channels you select changes the width of the `rx_in[]` port, and the deserialization factor changes the width of the `rx_out[]` port. If the required number of channels is not available in the list, type the desired number in the **What is the number of channels?** box.

The list for the deserialization factor is device-dependent. When DPA mode is turned on, the deserialization factors available in the list are 8 and 10. The left side of the MegaWizard Plug-In Manager shows the schematic representation of the custom function variation that you are creating. The schematic is updated as the ports and parameters are changed.

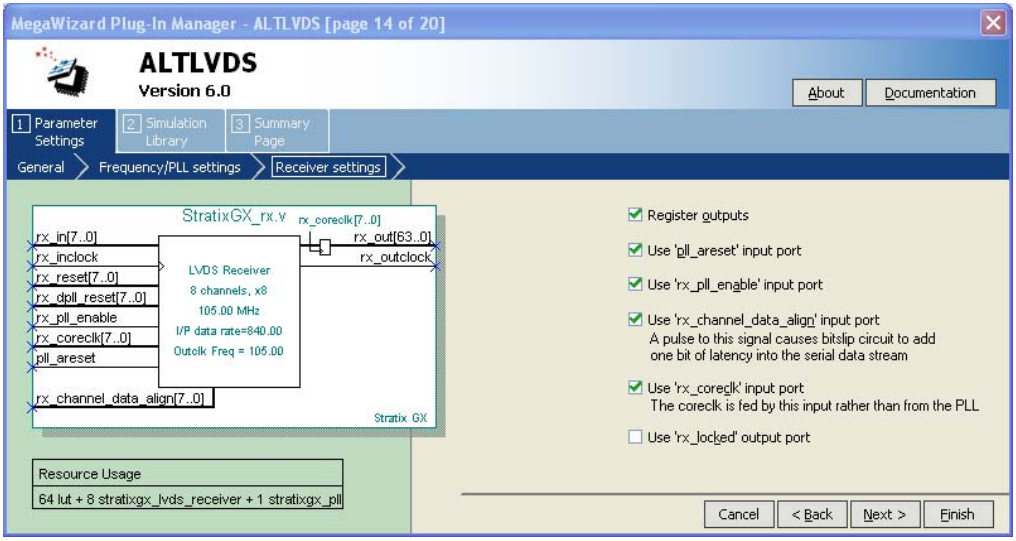
*Figure 2–9. MegaWizard Plug-In Manager—altlvds [page 13 of 20]*




On page 13 of the wizard, enter the input data rate. Data rates up to 1 Gbps (shown in the wizard as 1000 Mbps) are supported when DPA mode is turned on. The input clock rates available depend on the data rate you specify. Choose the appropriate clock for your design from the **clock frequency** or **clock period** lists. The other options on this page are described in Table 3–6 on page 3–14.

When **Register outputs** is turned on (Figure 2–10), a register is shown on the `rx_out[]` port. The registers on the `rx_out[]` ports are clocked by `rx_outclock` by default.

*Figure 2–10. MegaWizard Plug-In Manager—altlvds [page 13 of 20]*




Stratix GX devices use a clock generated by the fast phase-locked loops (PLL) to operate the parallel side of the deserializer, the data realignment circuitry, and the final register stage. This clock is driven by the fast PLL to the global clock tree, then back into the receiver. Turn on **Use 'rx_coreclk' input port** to feed a different clock back into the receiver in place of the clock that is generated by the fast PLL. This will also switch the clock on the output registers from `rx_outclock` to `rx_coreclk[]`.

For more information about the DPA receiver circuit, refer to Figure 9 in *AN 236: Using Source-Synchronous Signaling with DPA in Stratix GX Devices*.

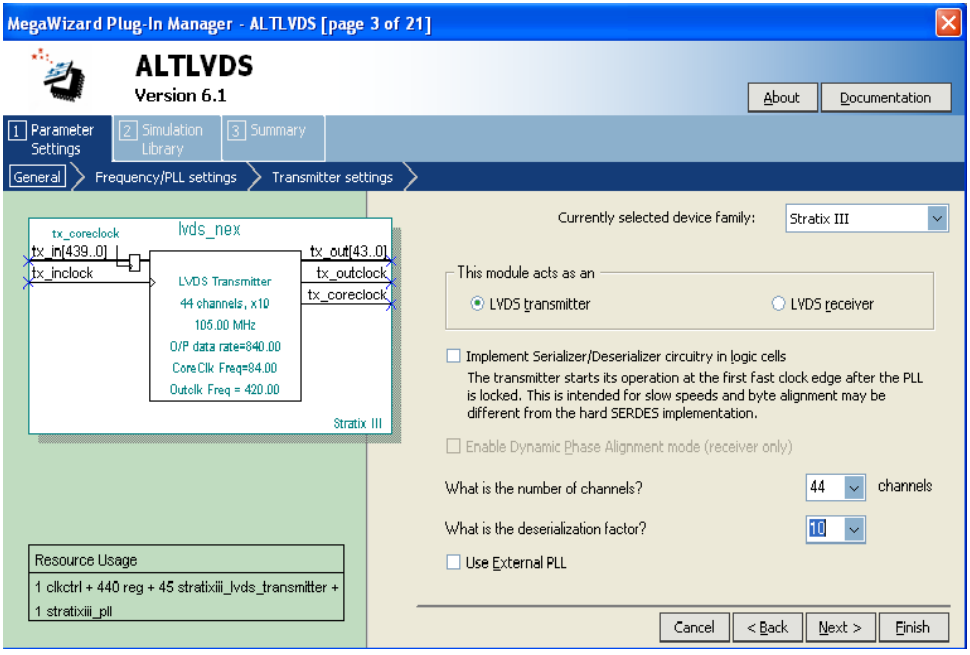
All of the other ports on this page are described in Table 3–4 on page 3–10 and Table 3–5 on page 3–13.

Page 6 of the wizard is the summary of the files that are generated for your altlvds megafunctions. It is the same for all device families and is described in "Using the MegaWizard Plug-In Manager" on page 2–2.

## Options for Stratix III and Cyclone III Devices

The selections you make on page 3 (shown in Figure 2–11) dictate the format of the remaining pages for Stratix III and Cyclone III. Here you have the option to customize the megafunction as a LVDS Transmitter or as a LVDS Receiver. Note that some pages are only applicable to Cyclone III or Stratix III.

*Figure 2–11. MegaWizard Plug-In Manager—altlvds [page 3 of 21] (Stratix III Device)*



## LVDS Transmitter Pages for Stratix III and Cyclone III Devices

On page 3 (Figure 2–11) of the wizard, select the LVDS Transmitter. Several options can be customized here for the transmitter settings. The options are described in Table 2–1.

*Table 2–1. Transmitter Settings Options*

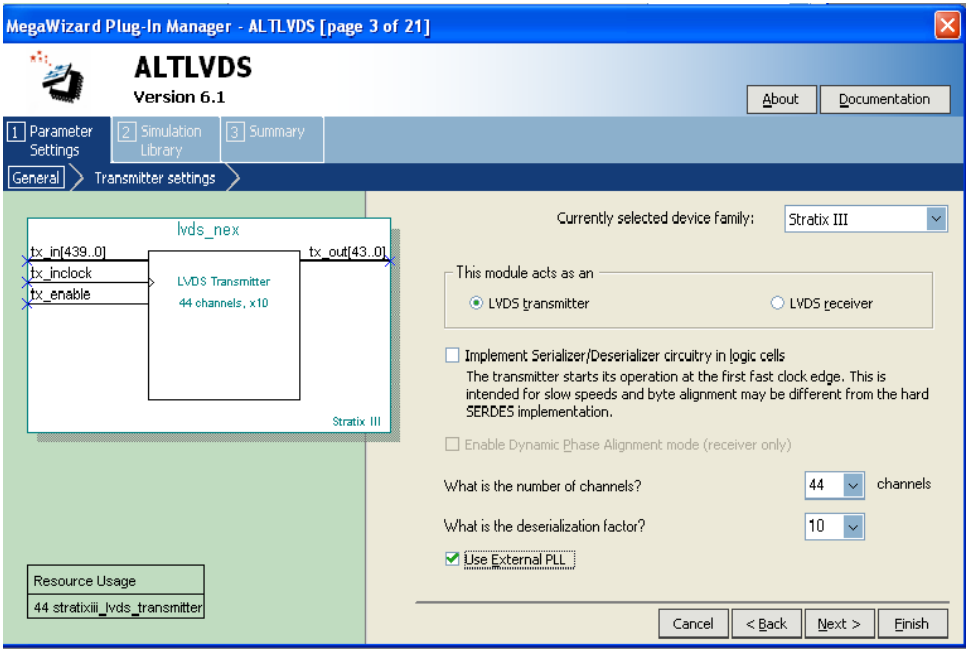
| Parameter | Description | Comments |
|---|---|---|
| Implement Serializer/Deserializer in logic cells? | If not enabled, the megafunction takes advantage of dedicated SERDES circuitry in the Stratix III device. If enabled, SERDES circuitry is implemented in logic cells. This kind of option is only available for Stratix III devices. For Cyclone III devices, this option is always enabled. | If enabled, the transmitter starts its operation the first fast clock edge after the PLL is locked. This option is intended for slow speeds and byte alignment may be different from the hard SERDES implementation. |
| What is the number of channels? | Number of output channels available for the LVDS transmitter. You can select from 1 to 44 channels for Stratix III devices or 1 to 18 channels for Cyclone III devices. | If the number of channels is specified as 44, there will be a `tx_out [43..0]` port created. You can either select a value from the drop-down list, or enter the desired value. |

*Table 2–1. Transmitter Settings Options*

| Parameter | Description | Comments |
|---|---|---|
| What is the deserialization factor? | This determines the number of bits in the transmitter. For Stratix III devices, if using dedicated SERDES, the supported deserialization factors are 1, 2, 4, 6, 7, 8, and 10. If using Logic Elements implementation, the deserialization factors supported are 1, 2, 4, 5, 6, 7, 8, 9, and 10.<br><br>For Cyclone III devices (which always uses LE Implementation), the deserialization factors supported are 1, 2, 4, 5, 6, 7, 8, 9, and 10. | If the number of channels is specified as 44 and deserialization factor as 10, a `tx_in [439..0]` port is created indicating 440 input signals. |
| Use External PLL | This determines whether to use an external PLL to clock the SERDES transmitter. If not enabled, the megafunction will automatically implement a PLL which can be customized on the next page. If enabled, a separate PLL will need to be used to take care of the clocking source. It is the user's responsibility to take care of the necessary connections. | If not enabled, the next page displays the PLL settings. The PLL settings page is skipped if this option is enabled. |

If the Use External PLL option is enabled, the LVDS transmitter block looks different, as shown in Figure 2–12.

*Figure 2–12. MegaWizard Plug-In Manager—altlvds [page 3 of 21] (Stratix III Device Using External PLL)*




The Frequency/PLL settings are only available if the **Use External PLL** option is disabled on page 3.

On page 11 of the wizard (shown in Figure 2–13), you can customize the PLL. Several options can be customized for the frequency and PLL settings. These options are described in Table 2–2.

*Figure 2–13. MegaWizard Plug-In Manager—altlvds [page 11 of 21] (Stratix III Device)*

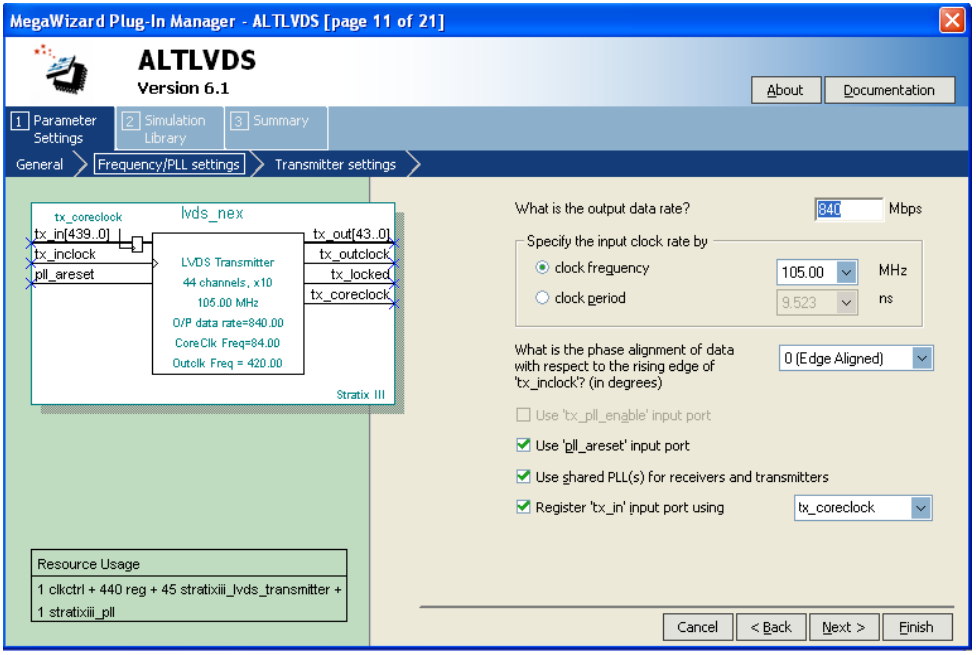



*Table 2–2. Frequency/PLL Settings Options when Using an LVDS Transmitter for a Stratix III or Cyclone III Device (Part 1 of 2)*

| Parameter | Description | Comments |
|---|---|---|
| What is the output data rate? | This specifies the data rate it can support for Stratix III devices or Cyclone III devices. For data rate ranges, refer to the specific *DC & Switching Characteristics* chapter in the appropriate device handbook. | This will affect the input clock rate. |
| Specify the input clock rate by | This specifies the `tx_inclk` frequency. It is dependent on the output data rate selected. | If output data rate is 1250 Mbps, the available input clock frequency can range from 41.67 MHz to 625.00 MHz. |

*Table 2–2. Frequency/PLL Settings Options when Using an LVDS Transmitter for a Stratix III or Cyclone III Device (Part 2 of 2)*

| Parameter | Description | Comments |
|---|---|---|
| What is the phase alignment of data with respect to the rising edge of 'tx_inclk'? (in degrees) | This determines the phase alignment of the data transmitted by the transmitter core with respect to the `tx_inclock`. Available values are 0 (edge-aligned), 45, 90, 135, 180 (center-aligned), 225, 270, 315. | — |
| Use "PLL_areset" input port | This option gives you control of the asynchronous reset port of the PLL that is used with this function. | When the transmitter shares the PLL with the receiver and the `pll_areset` port is used, the port must be used in both megafunctions and tied together in the design file or the port can be omitted from both megafunctions. If the port is used in one of the megafunctions and not the other, the PLLS are not shared and a warning is shown during compilation. |
| Use shared PLL(s) for receivers and transmitters | When this option is enabled, receivers and/or transmitters can share the same PLL. This is based on certain requirements and connections between the PLL and altlvds modules. This is done automatically by the compiler. | This setting can be used if the receiver and the transmitter use the same clock. |
| Register "tx_in" input port using? | If enabled, specify input registers to be clocked by either `tx_inclock` or `tx_coreclock`.<br><br>For better timing performance, choose `tx_coreclock`. This is the default selection in the Quartus II software. | If you turn this parameter off, a warning message appears directing you to pre-register the inputs in the logic that feeds the transmitter and also directing you to specify values for **Multicycle** and **Multicycle Hold** assignment from the input registers to the transmitter. The values are equal to the deserialization factor minus one, and the deserialization factor, respectively. |

Page 12 (shown in Figure 2–14) is used to specify the transmitter settings options. The options for this page are described in Table 2–3.

*Figure 2–14. MegaWizard Plug-In Manager—altlvds [page 12 of 21] (Stratix III Device)*

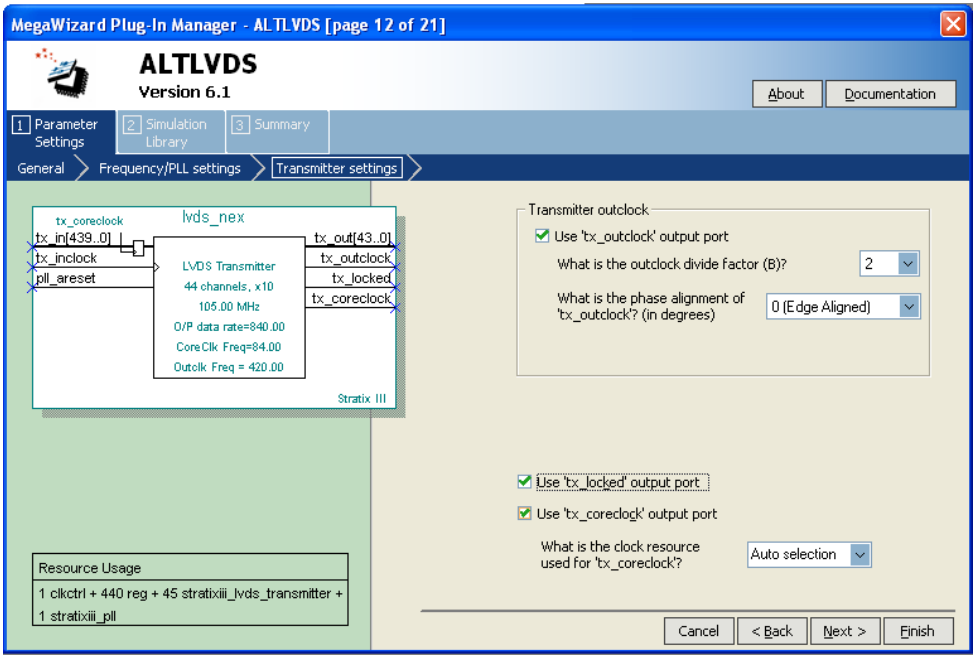



*Table 2–3. Transmitter Settings Options when Using an LVDS Transmitter for a Stratix III or Cyclone III Device (Part 1 of 2)*

| Parameter | Description | Comments |
|---|---|---|
| Use "tx_outclock" output port | `tx_outclock` is associated with the serial transmit data stream. | — |
| What is the outclock divide factor (B)? | For Stratix III devices, available choices are 1, 2, or 10. For Cyclone III devices, available choices are 2, 4, 10, or 20. The outclock_divide_by factor depends on the deserialization factor. For J=4, it is 1, 2 and 4; for J=7, it is 1 and 7. | The `tx_outclock` frequency is the data rate divided by the outclock divide factor. |

*Table 2–3. Transmitter Settings Options when Using an LVDS Transmitter for a Stratix III or Cyclone III Device (Part 2 of 2)*

| Parameter | Description | Comments |
|---|---|---|
| What is the phase alignment of "tx_outclock"? (in degrees) | This specifies the phase alignment of `tx_outclock` with respect to the `tx_inclock`. Available values are 0 (edge-aligned), 45, 90, 135, 180 (center-aligned), 225, 270, 315. | — |
| Use "tx_locked " output port | This port gives you the option of monitoring the lock status of the PLL. | The status of the lock port is identical between the transmitter and receiver when shared PLLs are used. |
| Use “tx_coreclock” output port | This port is used for registering the inputs (which does not require enabling this port). | This setting is used to show the core clock frequency during simulation. The `tx_coreclk` port should be used to register all logic that feeds the transmit function. |
| What is the clock resource use for “tx_coreclock”? | Can specify Auto Selection (determined by the compiler), Global Clock, or Regional clock to assist with clock management in a a Stratix III or Cyclone III device. | — |

Page 20 highlights the simulation libraries necessary for functional simulation in third-party tools. Figure 2–15 shows the libraries necessary for the LVDS transmitter when implemented in the dedicated SERDES hardware (Stratix III only). Figure 2–16 shows the libraries necessary for the LVDS transmitter when implemented in logic elements (applies either to Stratix III or Cyclone III device).

*Figure 2–15. MegaWizard Plug-In Manager—altlvds [page 20 of 21] (Stratix III Device Using Dedicated SERDES)*

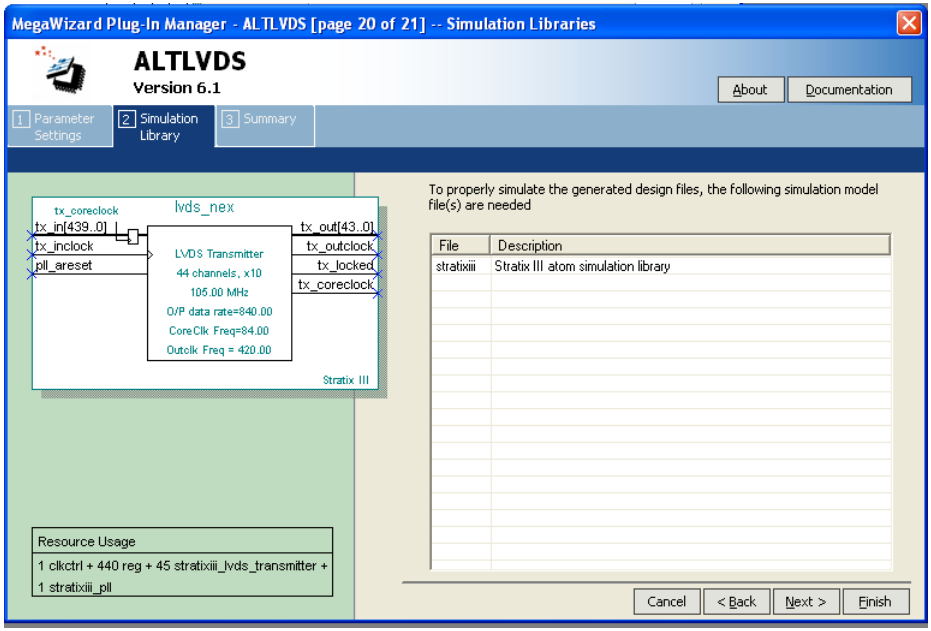
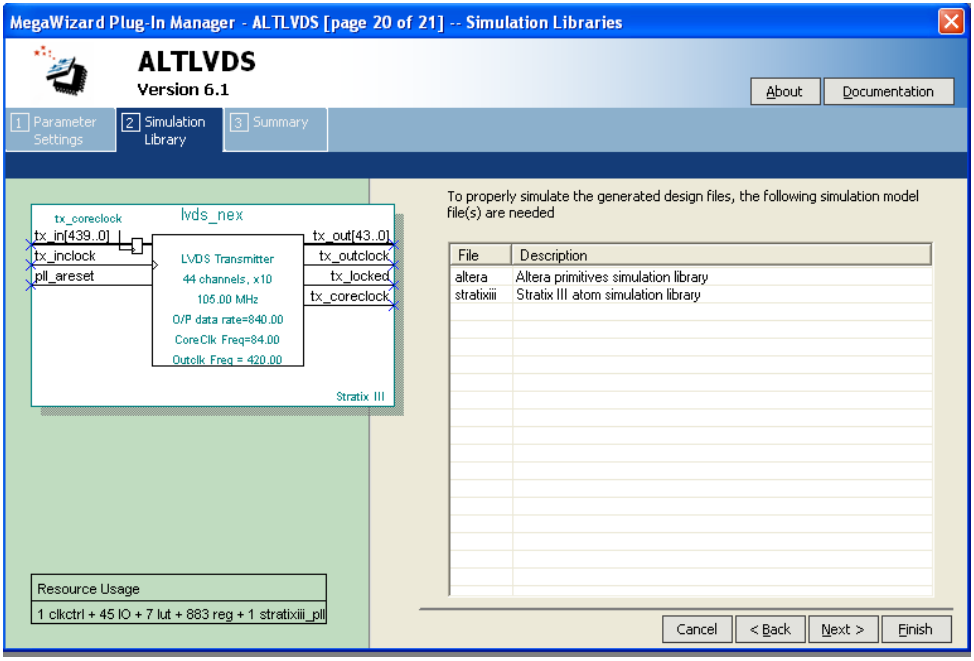


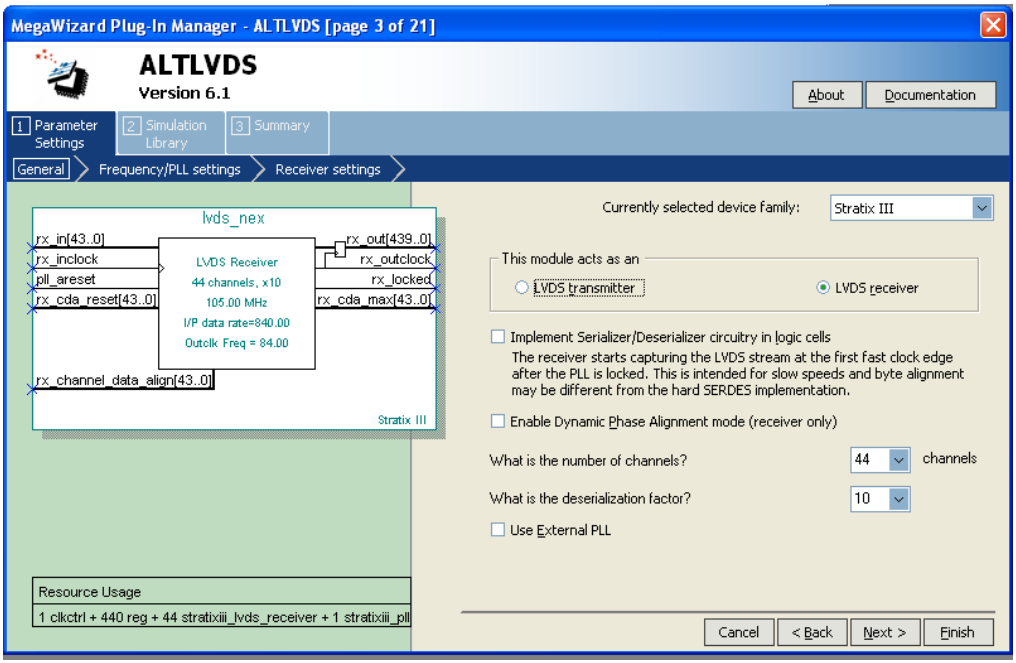
*Figure 2–16. MegaWizard Plug-In Manager—altlvds [page 20 of 21] (Stratix III or Cyclone III Device Using LE (Logic Eement))*



## LVDS Receiver Pages for Stratix III and Cyclone III Devices

On page 3 of the wizard, select the LVDS receiver as shown in Figure 2–17.

*Figure 2–17. MegaWizard Plug-In Manager—altlvds [page 3 of 21] (Stratix III Device without DPA)*



Several options can be customized on this page for the receiver settings. These options are described in Table 2–4.

*Table 2–4. General Receive Options when Using an LVDS Receiver for a Stratix III or Cyclone III Device (Part 1 of 2)*

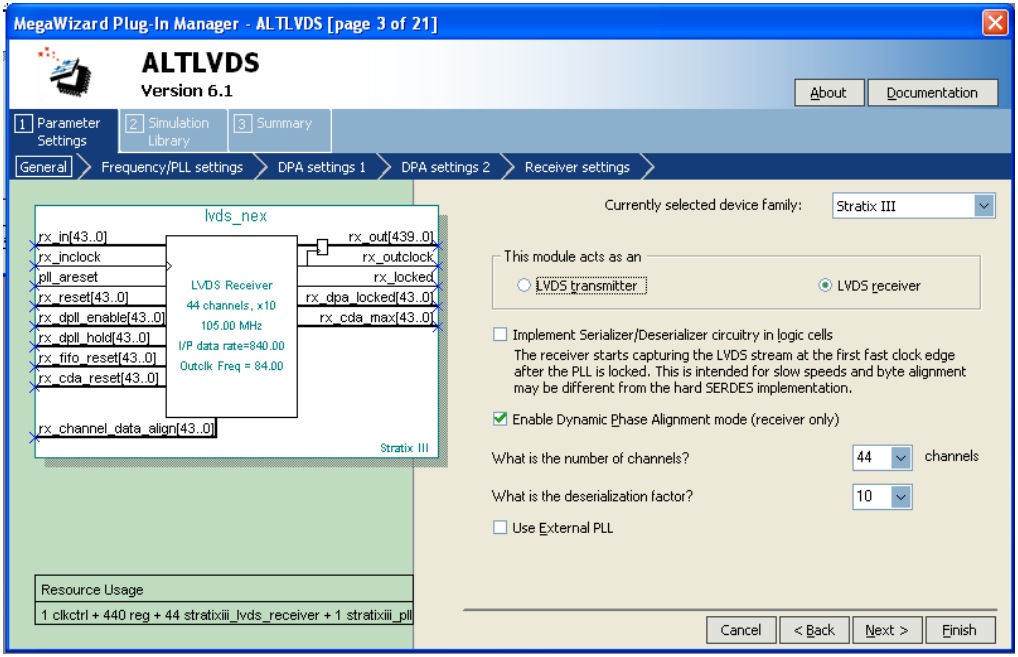
| Parameter | Description | Comments |
|---|---|---|
| Implement Serializer/Deserializer in logic cells? | If not enabled, the megafunction takes advantage of dedicated SERDES circuitry in the Stratix III device. If enabled, SERDES circuitry is implemented in logic cells. This kind of option is only available for Stratix III devices. For Cyclone III devices, this option is always enabled. | If enabled, the receiver starts capturing the LVDS stream at the first fast clock edge after the PLL is locked. This option is intended for slow speeds and byte alignment may be different from the hard SERDES implementation. |
| What is the number of channels? | Number of input channels available for the LVDS receiver. Can select from 1 to 44 channels for Stratix III devices or 1 to 18 channels for Cyclone III devices. | If the number of channels is specified as 44, there will be a `rx_in [43..0]` port created. You can select a value from the drop-down menu, or enter the desired value. |
| What is the deserialization factor? | This determines the number of bits in the receiver. For Stratix III devices, if using dedicated SERDES, the supported deserialization factors are 1, 2, 4, 6, 7, 8, and 10. If using Logic Elements implementation, the deserialization factors supported are 1, 2, 4, 5, 6, 7, 8, 9, and 10.<br><br>For Cyclone III devices (which always uses LE Implementation), the deserialization factors supported are 1, 2, 4, 5, 6, 7, 8, 9, and 10. | If the number of channels is specified as 44 and the deserialization factor is specified as 10, a `rx_out [439..0]` port will be created indicating 440 output signals. |

*Table 2–4. General Receive Options when Using an LVDS Receiver for a Stratix III or Cyclone III Device (Part 2 of 2)*

| Parameter | Description | Comments |
|---|---|---|
| Use External PLL | This determines whether to use an external PLL to clock the SERDES receiver. If not enabled, the megafunction automatically implements a PLL that can be customized on the next page. If enabled, a separate PLL is needed to address the clocking source. (It is the user's responsibility to resolve all necessary connections.) | If not enabled, the next page displays the Frequency/PLL settings. The Frequency/PLL settings page is skipped if this option is enabled. |
| Enable Dynamic Phase Alignment mode (Receiver Only) | This changes the appearance of the graphic representation of the megafunction in the left-hand pane. DPA mode adds several ports and parameters that are not available in non-DPA mode. When DPA mode is turned on, two pages are added to the wizard to include the additional DPA mode ports and parameters. Use DPA mode to correct skew created by different trace lengths on the data channels routed to the Stratix III device. This option is not available for the Cyclone III device. | — |

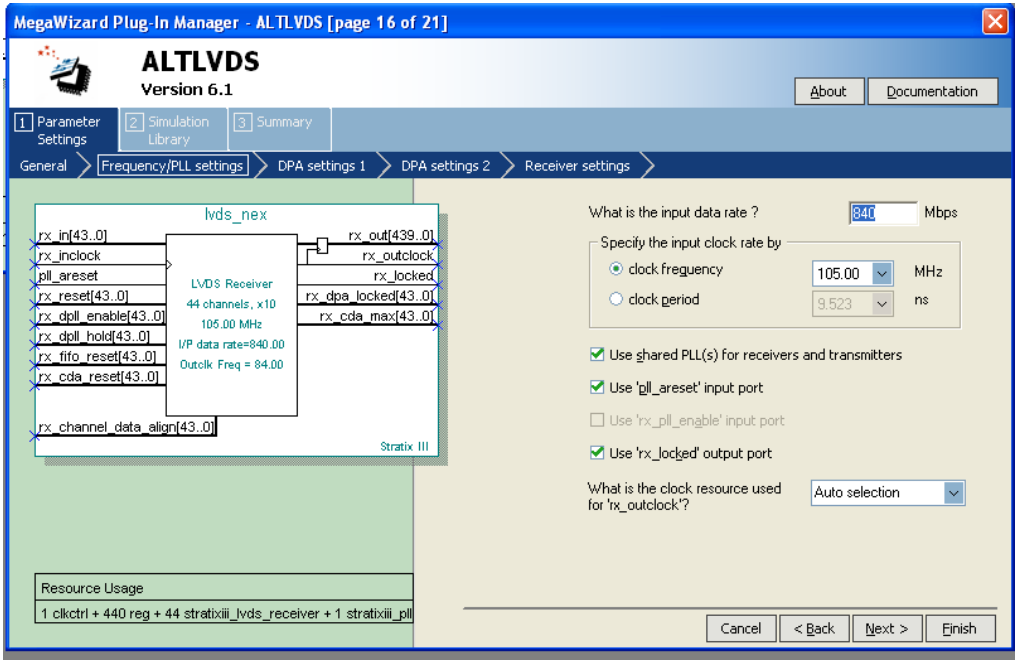
If the DPA mode option is enabled, the LVDS receiver block will look different, as shown in Figure 2–18.

*Figure 2–18. MegaWizard Plug-In Manager—altlvds [page 3 of 21] (Stratix III Device with DPA)*




Page 16 (shown in Figure 2–19) displays the Frequency/PLL settings options available only if the **Use External PLL** option is disabled on page 3.

*Figure 2–19. MegaWizard Plug-In Manager—altlvds [page 16 of 21] (Stratix III Device with DPA)*




These options are described in Table 2–5.

*Table 2–5. Frequency/PLL Settings Options When Using an LVDS Receiver for a Stratix III or Cyclone III Device (Part 1 of 3)*

| Parameter | Description | Comments |
|---|---|---|
| What is the input data rate? | This specifies the data rate it can support for Stratix III or Cyclone III devices. For data rate ranges, refer to the specific *DC & Switching Characteristics* chapter in the appropriate device handbook. | This will affect the input clock rate. |
| Specify the input clock rate by | This specifies the `rx_inclk` frequency. It is dependent on the input data rate selected. | If input data rate is 1250 Mbps, available input clock frequency can range from 41.67 MHz to 625.00 MHz. |
| Use "rx_locked " output port | This port gives you the option of monitoring the lock status of the PLL. | The status of the lock port is identical between the transmitter and receiver when shared PLLs are used. |

*Table 2–5. Frequency/PLL Settings Options When Using an LVDS Receiver for a Stratix III or Cyclone III Device (Part 2 of 3)*

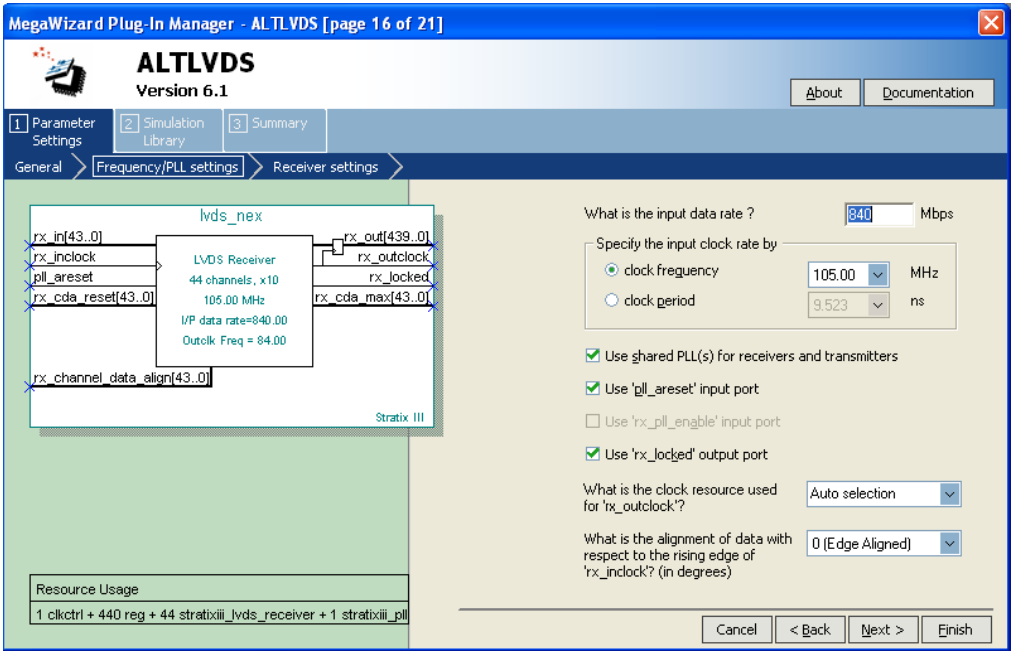
| Parameter | Description | Comments |
|---|---|---|
| Use shared PLL(s) for receivers and transmitters | When this option is enabled, the receivers and/or transmitters can share the same PLL. This is based on certain requirements and connections between the PLL and altlvds modules. This is done automatically by the compiler. | When the receiver shares the PLL with other the transmitters and/or receivers and the `pll_areset` port is used, the port must be used in all megafunctions and tied together in the design file or the port can be omitted from all megafunctions. If the port is used in one of the megafunctions and not the others, the PLLs are not shared and a warning is shown during compilation. |
| Use "PLL_areset" input port | This option gives you control of the asynchronous reset port of the PLL that is used with this function | This setting can be used if the receiver and the transmitter use the same clock. |
| What is the clock resource use for "rx_outclock"? | You can specify Auto Selection (determined by the compiler), Global Clock, or Regional clock to assist with clock management in a Stratix III device. This option is not available in Cyclone III devices. | — |
| What is the phase alignment of data with respect to the rising edge of 'rx_inclock'? (in degrees) | This determines the phase alignment of the data received by the receiver core with respect to the `rx_inclock`. Available values are 0 (edge aligned), 45, 90, 135, 180 (center aligned), 225, 270, and 315. | — |

*Table 2–5. Frequency/PLL Settings Options When Using an LVDS Receiver for a Stratix III or Cyclone III Device (Part 3 of 3)*

| Parameter | Description | Comments |
|---|---|---|
| Use source synchronous mode of the PLL | The phase relationship between data and clock at the pin will be maintained at the data capture point. This specifies the source synchronous mode for Cyclone III devices and Stratix III devices (LE Implementation) of the PLLs. For Cyclone III devices and Stratix III (LE Implementation) devices, this option is always enabled. | — |
| Align clock to center of data window at capture point | When enabled, a phase shift of 90 degrees is added to the clock to center the clock in the data. Use this parameter when the `PLL_OPERATION_MODE` parameter value is set to "SOURCE_SYNCHRONOUS" for Cyclone III devices and Stratix III (LE Implementation) devices. For Cyclone III devices and Stratix III (LE Implementation) devices, this option is always enabled. | — |

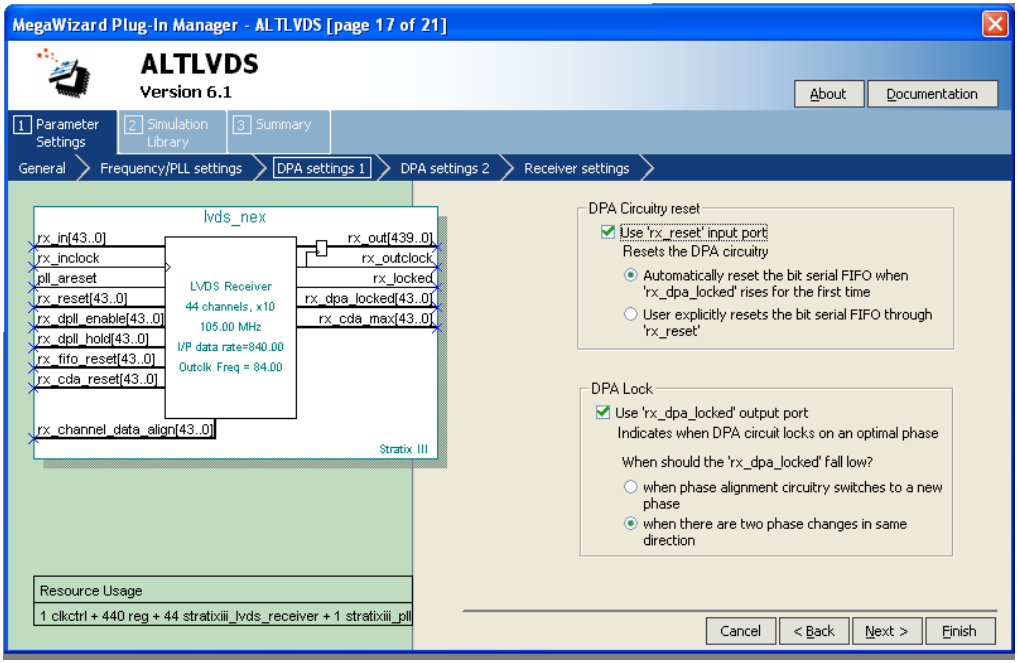
Page 16, which shows the Frequency/PLL settings, will appear slightly different if DPA mode is not used (shown Figure 2–20). Notice that the extra two pages for DPA have been removed, but the option, **What is the alignment of data with respect to the rising edge of 'rx_inclock'? (in degrees)** has been added. This option determines the phase alignment of the data received by the receiver core with respect to the `rx_inclock`. Available values are 0 (edge-aligned), 45, 90, 135, 180 (center-aligned), 225, 270, 315.

*Figure 2–20. MegaWizard Plug-In Manager—altlvds [page16 of 21] (Stratix III Device without DPA)*




On page 17 of the wizard, there are options to customize the DPA settings such as the DPA circuitry reset and the DPA lock. This is available only when DPA mode is enabled, as shown in Figure 2–21.

*Figure 2–21. MegaWizard Plug-In Manager—altlvds [page 17 of 21] (Stratix III Device with DPA)*



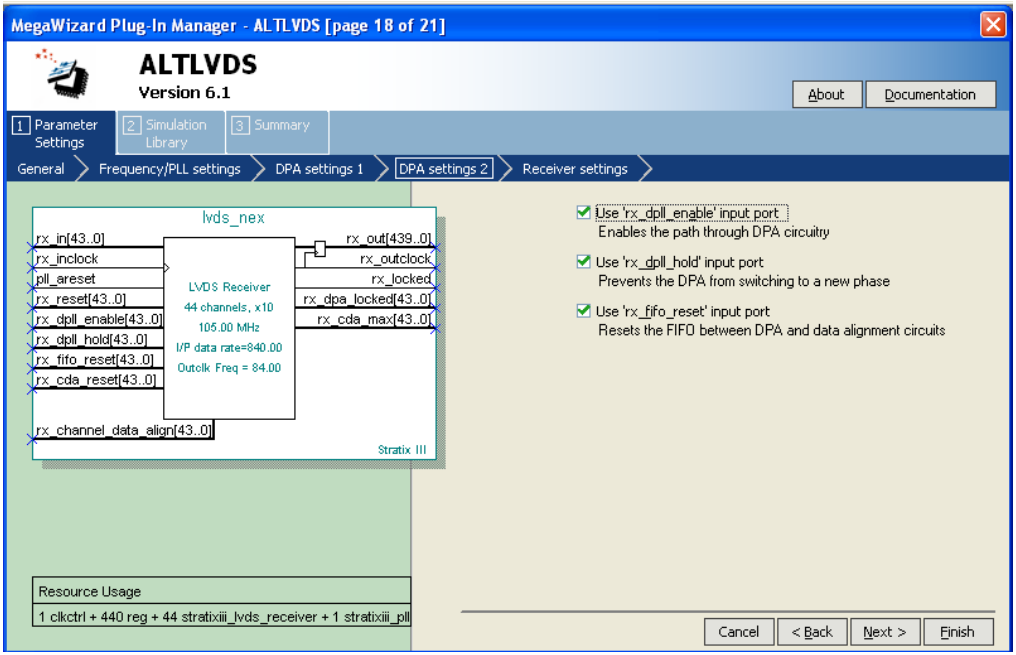
These options are described in Table 2–6.

*Table 2–6. DPS Settings Options When Using an LVDS Receiver for a Stratix III Device*

| Parameter | Description | Comments |
|---|---|---|
| Use "rx_reset" input port | This port resets all of the components of the DPA circuit. You can also specify conditions of DPA circuit reset which can be either automatic reset when "rx_dpa_locked " rises for the first time or user explicitly resets the serial first-in first-out (FIFO) through "rx_reset". | The DPA circuit must be re-trained after it has been reset using this port. |
| Use "rx_dpa_locked" output port | The DPA block samples the data on one of eight phase clocks with a 45-degree resolution between phases. This port lets you monitor the status of the DPA circuit and determine when it has locked onto the phase closest to the incoming data phase. The `rx_dpa_locked` ports de-assert (drive low), depending on the selections made in the **When should 'rx_dpa_locked' fall low?** box. This can occur when a new phase has been selected or when the DPA has made two 45-degree phase shifts in the same direction. The data may still be valid even when the `rx_dpa_locked` port is de-asserted. | Altera recommends that you use data checkers to validate the data in this condition. |

Page 18 highlights the remaining settings for DPA mode, as shown in Figure 2–22.

*Figure 2–22. MegaWizard Plug-In Manager—altlvds [page 18 of 21] (Stratix III Device with DPA)*



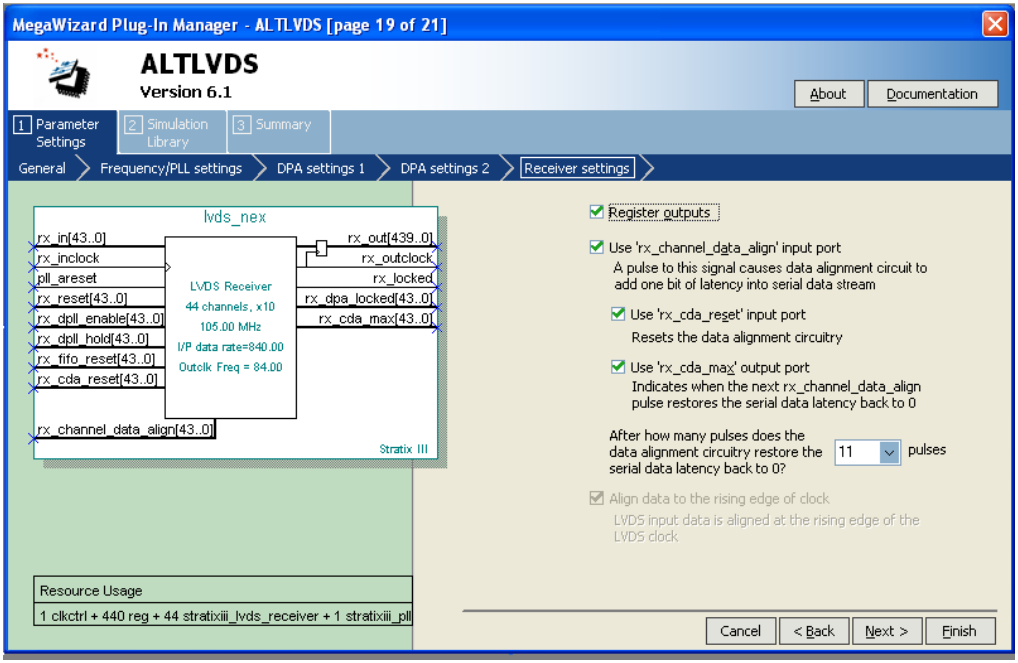
These options are described in Table 2–7.

*Table 2–7. Page 18 Options When Using an LVDS Receiver for a Stratix III Device*

| Parameter | Description | Comments |
|---|---|---|
| Use "rx_dpll_enable" input port | Enables the path through DPA circuitry. This port allows you to have dynamic channel-by-channel control of the DPA circuitry. | If the DPA circuitry is needed for any channel, set the port corresponding to the target channel to **high**. When this port is not used, all channels are enabled by the Quartus II software. |
| Use "rx_dpll_hold" input port | Prevents the DPA from switching to a new phase. Each DPA block continuously monitors the phase of the incoming data stream and selects a new clock phase, if needed. This port is used to prevent the DPA from switching to a new phase on a channel-by-channel basis. When this port is high, the selected channels hold their current phase setting. | — |
| Use "rx_fifo_reset" input port | Resets the FIFO between DPA and data alignment circuits. The FIFO buffer is placed between the DPA circuit and the data alignment circuit. This port is used to pass the data between the DPA and LVDS clock domains. When this port is high, the FIFO in the selected channels are reset. | — |

Page 19, highlights the receiver settings for DPA mode, as shown in Figure 2–23.

*Figure 2–23. MegaWizard Plug-In Manager—altlvds [page 19 of 21] (Stratix III Device with DPA)*



These options are described in Table 2–8.

*Table 2–8. Receiver Settings Options When Using an LVDS Receiver for a Stratix III or Cyclone III Device (Part 1 of 2)*

| Parameter | Description | Comments |
|---|---|---|
| Turn on Register outputs. | The outputs of the receiver are registered by `rx_outclock`. | If you do not register them here, you must register them in the design logic that is fed by the receiver and specify a 'Source Multiply' assignment from the receiver to the output registers with a value equal to the deserialization factor. |
| Turn on Use 'rx_channel_data_align' input port. | This lets you control bit insertion on a channel-by-channel basis to align the word boundaries of the incoming data. The data slips one bit for every pulse on the `rx_channel_data_align` port. This option is not available for Stratix III (LE implementation) devices or for Cyclone III devices. | The following requirements must be met for this port:<br>• The minimum pulse width is one period of the parallel clock in the logic array (`rx_outclock`).<br>• The minimum low time between pulses is one period of the parallel clock.<br>• There is no maximum high or low time.<br>• Valid data is available two parallel clock cycles after the rising edge of `rx_channel_data_align`. |
| Turn on Use 'rx_cda_reset' input port. | This port is available only if `Use 'rx_channel_data_align' input port` is turned on. The port resets the data alignment circuitry, restoring the latency bit counter to zero. This option is not available for Stratix III (LE implementation) devices or for Cyclone III devices. | — |
| Turn on Use 'rx_cda_max' output port. | This port is available only if `Use 'rx_channel_data_align' input port` is turned on. This port indicates when the rollover point has been reached in the data alignment circuit. This option is not available for Stratix III (LE implementation) devices or for Cyclone III devices. | — |

*Table 2–8. Receiver Settings Options When Using an LVDS Receiver for a Stratix III or Cyclone III Device (Part 2 of 2)*

| Parameter | Description | Comments |
|---|---|---|
| After how many pulses does the data alignment circuitry restore the serial latency back to 0? | Available option range from 1 to 11 and does not have to be the same as the deserialization factor. This option is not available for Stratix III (LE implementation) or for Cyclone III. | — |
| Align data to the rising edge of clock | LVDS input data is aligned at the rising edge of the LVDS clock. This option is not available for Stratix III (LE Implementation) devices or Cyclone III devices. | — |

Page 20 highlights the simulation libraries necessary for functional simulation in third party tools. Figure 2–24 shows the libraries necessary for the LVDS receiver when implemented in the dedicated SERDES hardware (applicable only to Stratix III) and Figure 2–25 shows the libraries necessary for the LVDS receiver when implemented in a logic element (applicable to Stratix III devices and Cyclone III devices).

*Figure 2–24. MegaWizard Plug-In Manager—altlvds [page 20 of 21] (Stratix III Device Using Dedicated SERDES)*

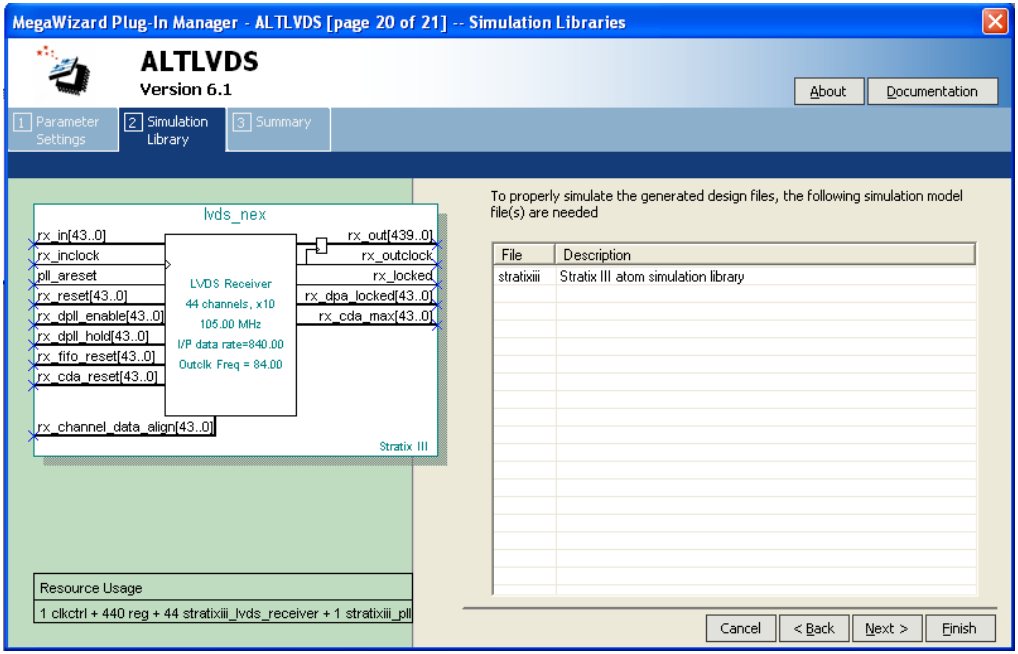
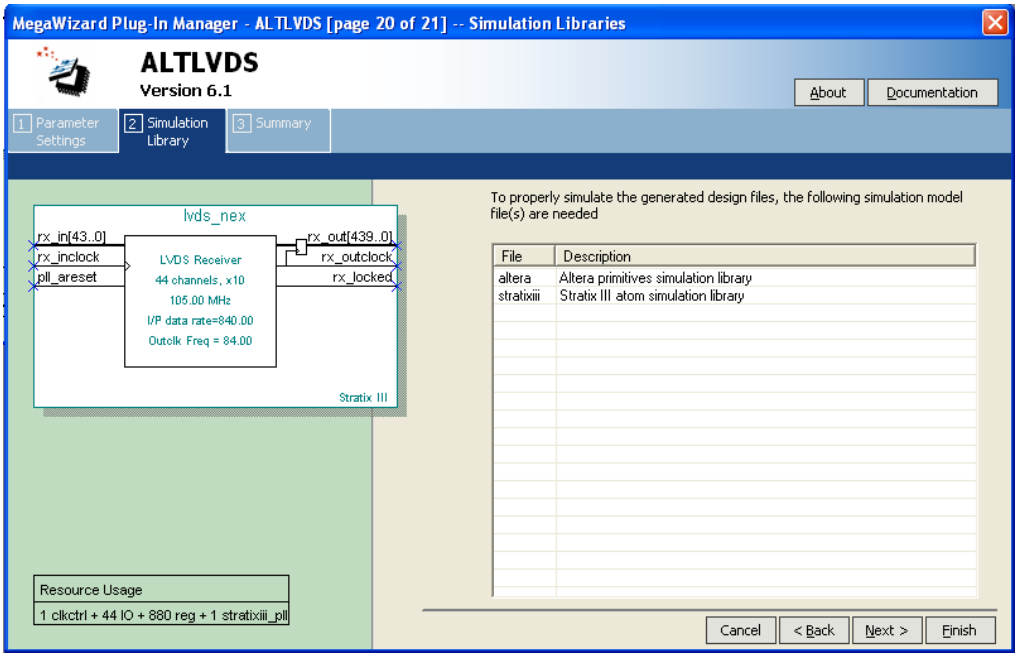


*Figure 2–25. MegaWizard Plug-In Manager—altlvds [page 3 of 21] (Stratix III or Cyclone III Device Using SERDES Implemented in LE)*



## Inferring Megafunctions from HDL Code

Synthesis tools, including Quartus II integrated synthesis, recognize certain types of HDL code and automatically infer the appropriate megafunction when a megafunction will provide optimal results. However, altlvds cannot be inferred and must be instantiated in your design. Refer to "Instantiating Megafunctions in HDL Code" and "Instantiating Megafunctions Using the Port and Parameter Definition" for details about instantiating megafunctions.

## Instantiating Megafunctions in HDL Code

When you use the MegaWizard Plug-In Manager to set up and parameterize a megafunction, it creates either a VHDL or Verilog HDL wrapper file that instantiates the megafunction (a black box methodology). For some megafunctions, you can generate a fully synthesizable netlist for improved results with EDA synthesis tools, such as Synplify and Precision RTL Synthesis (a clear box methodology). Altera recommends that you use the MegaWizard Plug-In Manager to create all your altlvds transmit and receive functions. However, you can create functional blocks directly in HDL code using the ports and parameters that are specific to the altlvds megafunctions.

For information about both clear box and black box methodologies, refer to volume 1 of the *Quartus II Handbook*.

## Instantiating Megafunctions Using the Port and Parameter Definition

You can create altlvds receiver and transmitter megafunctions directly in HDL code without using the MegaWizard Plug-In Manager. You must have a good understanding of the ports and parameters and their availability in Stratix III, Stratix II, Stratix, Stratix GX, Cyclone III, Cyclone II, and Cyclone devices. Also, you must understand that some parameters are available only when other features have been enabled. For example, in Stratix GX devices, the `rx_channel_data_align[]` port is available only when the DPA mode option is turned on. When DPA mode is turned off, the `rx_data_align` port is available for use.

Refer to the Chapter 3, Specifications in this User Guide for descriptions of the ports and parameters.

Ports and parameters that are defined as required must be in your HDL code to properly instantiate the megafunction. Any ports and parameters listed as optional are features you may or may not use depending on your design requirements.

Example 2–1 contains the VHDL code from the Stratix LVDS receiver created in "Using the MegaWizard Plug-In Manager" on page 2–2. This template can be used for any VHDL altlvds functions. Substitute appropriate ports and parameters as needed in your design.

***Example 2–1. VHDL Code from the Stratix LVDS Receiver***

```
INCLUDE "altlvds_rx.inc";

SUBDESIGN stratix_rx
(
    rx_in[0..0] : INPUT;
    rx_inclock : INPUT = GND;
    rx_pll_enable : INPUT = VCC;
    rx_data_align : INPUT;
    pll_areset : INPUT = GND;
    rx_out[3..0] : OUTPUT;
    rx_locked : OUTPUT;
    rx_outclock : OUTPUT;
)

VARIABLE

    altlvds_rx_component : altlvds_rx WITH (
           INTENDED_DEVICE_FAMILY = "Stratix",
           NUMBER_OF_CHANNELS = 1,
           DESERIALIZATION_FACTOR = 4,
           LPM_TYPE = "altlvds_rx",
           COMMON_RX_TX_PLL = "ON",
           OUTCLOCK_RESOURCE = "AUTO",
           INCLOCK_PERIOD = 9523,
           INPUT_DATA_RATE = 840,
           INCLOCK_DATA_ALIGNMENT = "EDGE_ALIGNED",
           REGISTERED_OUTPUT = "ON",
           REGISTERED_DATA_ALIGN_INPUT = "ON"
           );

BEGIN

    rx_locked = altlvds_rx_component.rx_locked;
    rx_out[3..0] = altlvds_rx_component.rx_out[3..0];
    rx_outclock = altlvds_rx_component.rx_outclock;
    altlvds_rx_component.pll_areset = pll_areset;
    altlvds_rx_component.rx_inclock = rx_inclock;
    altlvds_rx_component.rx_in[0..0] = rx_in[0..0];
    altlvds_rx_component.rx_data_align = rx_data_align;
    altlvds_rx_component.rx_pll_enable = rx_pll_enable;
END;
```

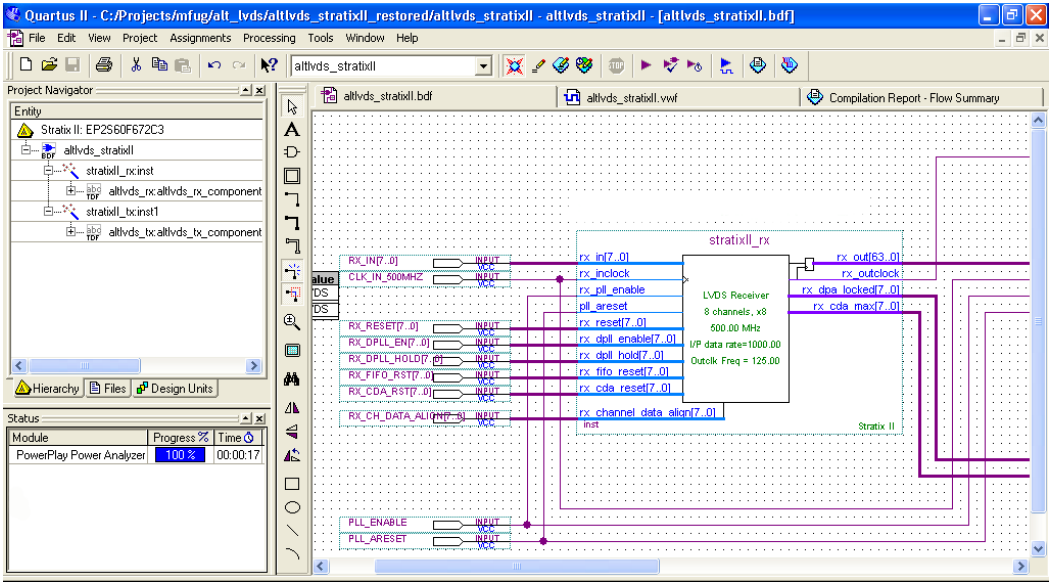
## Verifying the Megafunction

To verify that your megafunction is instantiated in your project correctly, view its placement in the floorplan, view the resources used, simulate its functionality, and view the timing analysis results.

## Identifying a Megafunction After Compilation

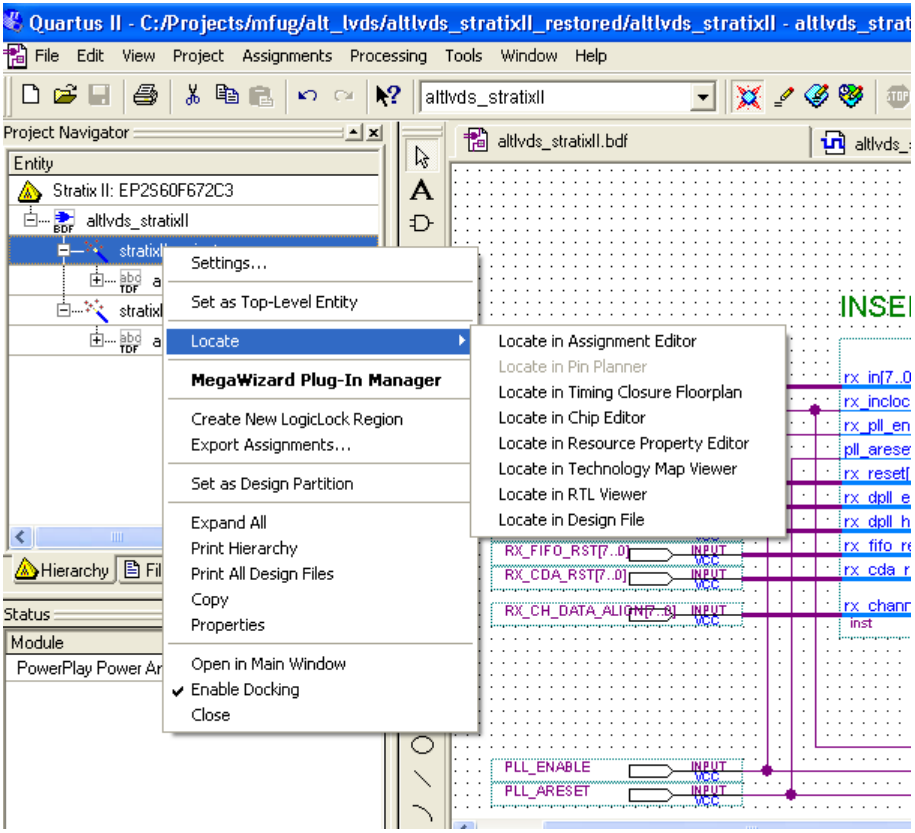
Every megafunction has an instantiation number in the Quartus II project hierarchy. You can find each instantiation by megafunction name and instantiation number in the compilation hierarchy. If you expand the megafunction name, you will see the functions that were written by the Quartus II software for your custom megafunction variation. For transmitter functions, the instantiations are called `altlvds_tx:altlvds_tx_component`. For receiver functions, the instantiations are called `altlvds_rx:altlvds_rx_component` (Figure 2–26).

*Figure 2–26. Compilation Hierarchy Window*



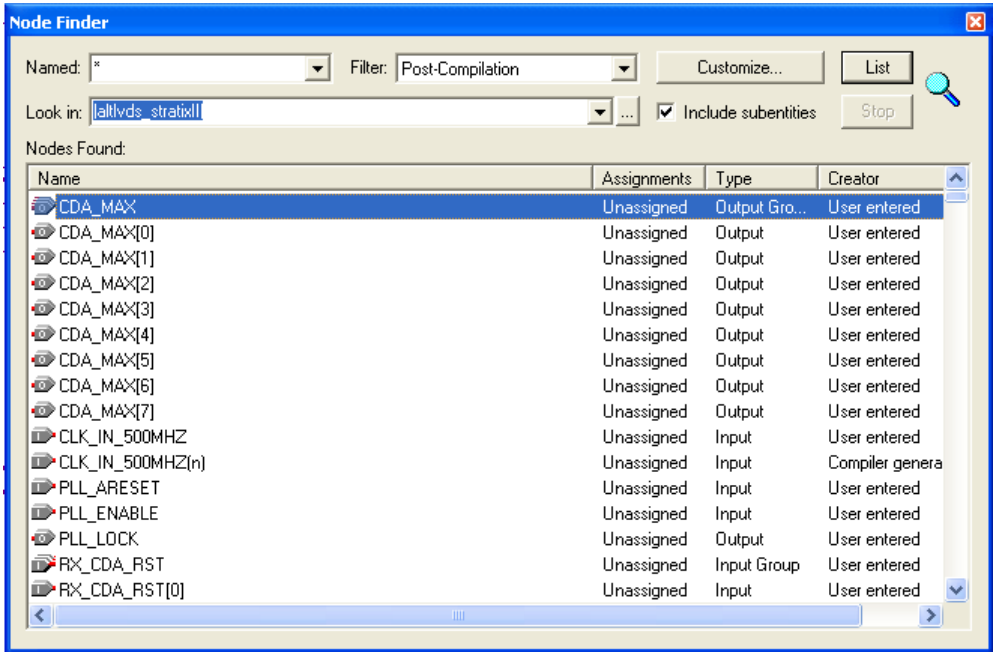

You can use the project hierarchy to find megafunction instantiations in the Floorplan Editor. Right-click on the megafunction you want to view, and choose the Quartus II editor in which you want to view it.

*Figure 2–27. Compilation Hierarchy Navigation Option*




You can also locate your megafunctions in the Node Finder when making assignments in the assignment editor. Figure 2–28 shows an example of how the megafunction logic is shown in the Node Finder.

*Figure 2–28. Node Finder View of altlvds Transmitter Megafunction*




# Simulation

The Quartus II Simulation tool provides an easy-to-use, integrated solution for performing simulations. The following sections describe the simulation options.

## Quartus II Simulation

The Quartus II Simulator is a powerful tool for testing and debugging the logical operation and internal timing of Altera megafunctions instantiated in your design.

With the Quartus II Simulator, you can perform two types of simulations: functional and timing. A functional simulation in the Quartus II program enables you to verify the logical operation of your design without taking into consideration the timing delay in the FPGA. This simulation is performed using only your register transfer level (RTL) code. When performing a functional simulation, you add only signals that exist before synthesis.You can find these signals with the pre-synthesis, design entry, or pin filters in the Node Finder. These three filters find the top-level ports of megafunctions.

In contrast, timing simulation in the Quartus II software verifies the operation of your design with annotated timing information. This simulation is performed using the post place-and-route netlist. When performing a timing simulation, add only signals that exist *after* place-and-route. These signals are found with the post-compilation filter of the Node Finder. During synthesis and place-and-route, the names of RTL signals change. Therefore, it might be difficult to find signals from your megafunction instantiation in the post-compilation filter. To preserve the names of your signals during the synthesis and place and route stages, use the synthesis attributes `keep` or `preserve`. These are Verilog and VHDL synthesis attributes that direct Analysis and Synthesis to keep a particular wire, register, or node intact. You can use these synthesis attributes to keep a combinational logic node so you can observe the node during simulation.

For more information about these attributes, refer to volume 1 of the *Quartus II Handbook*.

## EDA Simulation

The *Quartus II Handbook* shows you how to perform functional and gate-level timing simulations that include the megafunctions, with details about the files that are needed and the directories where those files are located.

The models required for your instantiations are listed in the VHDL or Verilog HDL files generated by the wizard. For VHDL instantiations, the **altera_mf.vhd** and the **altera_mf_components.vhd** files are required. For Verilog instantiations, the **altera_mf.v** file is required. These files are located in the <*Quartus II install*>**\eda\sim_lib** directory.

For information about gate-level timing simulations, refer to the appropriate chapter for the simulation tool you are using in volume 3 of the *Quartus II Handbook*.

For information about timing simulation in the third party EDA simulators, refer to the Altera support page on the Altera web site at **www.altera.com**.

## SignalTap II Embedded Logic Analyzer

The SignalTap® II embedded logic analyzer provides a non-intrusive method of debugging all of the Altera megafunctions within your design. With the SignalTap II embedded logic analyzer, you can capture and analyze data samples for the top-level ports of the Altera megafunctions in your design while your system is running at full speed.

To monitor signals from your Altera megafunctions, first configure the SignalTap II embedded logic analyzer in the Quartus II software, and then include the analyzer as part of your Quartus II project. The Quartus II software then embeds the analyzer along with your design in the selected device seamlessly.

For more information about using the SignalTap II embedded logic analyzer, refer to volume 3 of the *Quartus II Handbook*.

## Design Example: LVDS-to-LVDS Bridge Using Different Clock Frequencies

With the inclusion of dynamic phase alignment (DPA) circuitry, Stratix II devices offer enhanced support for source-synchronous protocols. The enhanced source-synchronous channels on Stratix II devices support 1-Gbps data transfer, while the dedicated DPA circuitry simplifies printed circuit board design by eliminating signal-alignment issues introduced by clock-to-channel and channel-to-channel skew. Stratix II devices support a wide array of high-speed protocols and can be used to bridge high-speed interfaces. This design example uses a Stratix II device to bridge a 1-Gbps LVDS interface using a 500-MHz reference clock to a 1-Gbps LVDS interface using a 250-MHz reference clock. The altlvds receiver function uses DPA circuitry. The transmitter and receiver share one fast PLL.

The focus of this design is to illustrate the features available in the MegaWizard Plug-In Manager. No user logic is shown between the receiver and transmitter blocks, and all ports are connected to input or output pins. Most designs use custom logic for many of the control ports within the FPGA, however, for this design example, all such parameters are controlled outside of the Stratix II device.

### Design Files

The example design files are available in the *Quartus II Projects* section of the *Design Examples* page of the Altera web site.

In this example, you will create two files using the MegaWizard Plug-In Manager: one for an altlvds receiver and one for an altlvds transmitter. Both files are created in Verilog HDL and will be included in your project directory when completed.

*Example*

This example uses the MegaWizard Plug-In Manager in the Quartus II software. As you go through the wizard, each page is described in detail. When you are finished with this example, you can incorporate it into an overall design.

In this example, you will perform the following activities:

- Generate a high-speed differential receiver in DPA mode using the altlvds megafunction and the MegaWizard Plug-In Manager
- Generate a high-speed differential transmitter using the altlvds megafunction and the MegaWizard Plug-In Manager
- Implement the receiver and transmitter functions in the device by adding your custom megafunctions to the project and compiling the project
- Simulate the high-speed differential interface design

## Generate an altlvds Receiver and altlvds Transmitter

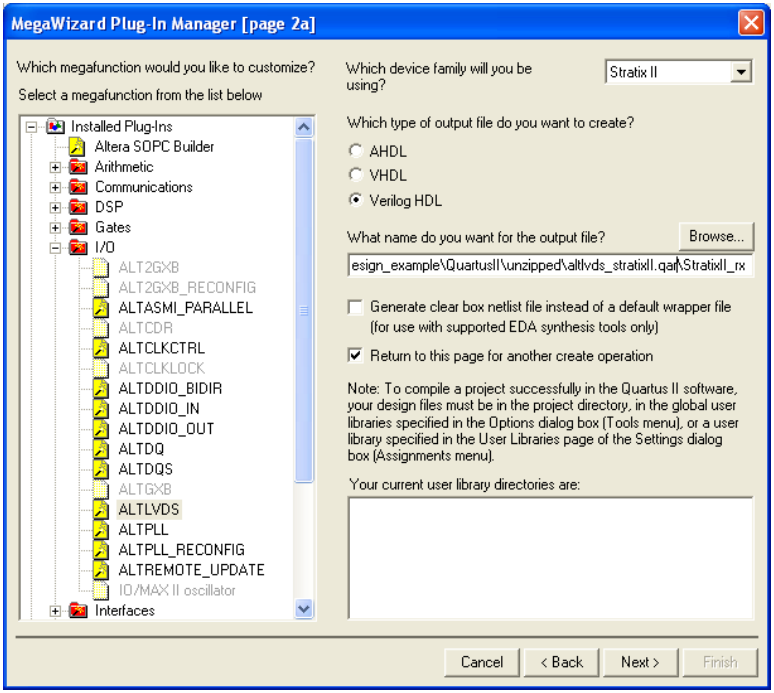
Unzip the **altlvds_DesignExample.zip file** to your preferred location. Then restore the Quartus II archive project **altlvds_stratixII.qar** and open the top-level block editor file **altlvds_stratixII.bdf**. This is an incomplete file that you will complete in the course of this example. The altlvds megafunctions created in this example are added to the top-level file.

1. Double-click anywhere in the white space in the block editor file. The Symbol window appears.

2. Click on **MegaWizard Plug-In Manager**. Page 1 appears. Choose **Create a new custom megafunction variation**.

3. Click **Next**. Page 2a of the wizard appears (Figure 2–29).

*Figure 2–29. MegaWizard Plug-In Manager [page 2a]*




4. On page 2a, in the **Which device family will you be using?** list, choose **Stratix II**.

5. Under **Which megafunction would you like to customize?,** in the **Installed Plug-Ins** list, expand the **I/O** folder and select **ALTLVDS**.

   Your project directory displays by default in the **What name do you want for the output file?** box. You can browse to a different directory if you want to save your megafunction there.

   If you save the megafunction to a different directory, you must specify that location in the **User Libraries (Current Project)** page.

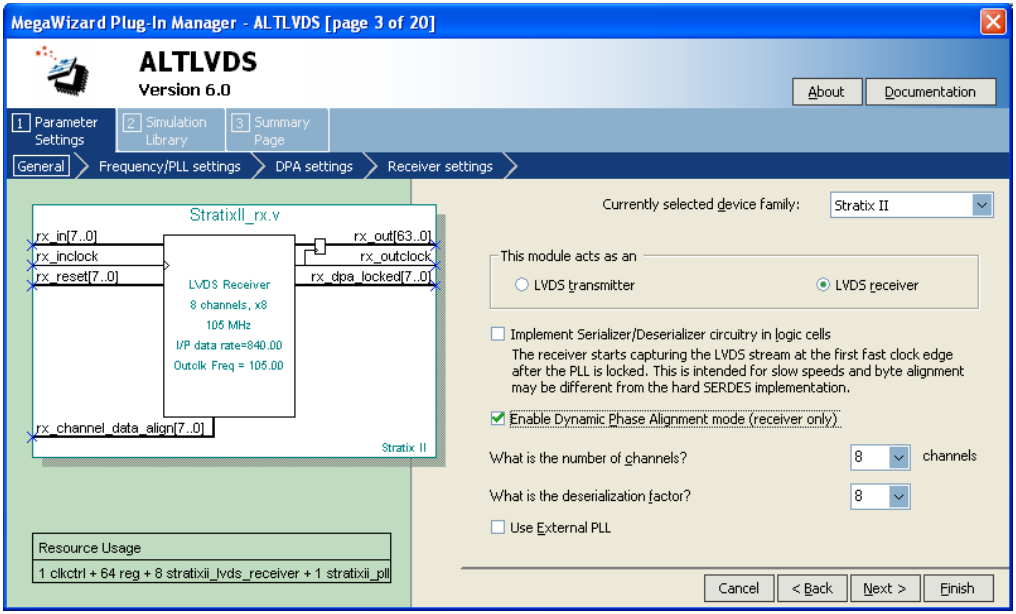
   To access this page, on the Assignments menu, click **Settings**. In the **Settings** dialog box, in the **Category** list, select **User Libraries (Current Project)**.

6. Enter a filename for your megafunction. The first one created in this example is a receiver. Name it `stratixii_rx`.

7. Turn on the **Return to this page for another create operation** option so you can create the transmitter once the receiver is complete.

8. Click **Next**. Page 3 of the wizard appears as shown in Figure 2–30.

*Figure 2–30. MegaWizard Plug-In Manager—altlvds [page 3 of 20]*




9. On page 3, in the **Currently selected device family** list, ensure that **Stratix II** is selected.

10. Under **This module acts as an**, turn on **LVDS receiver**.

11. Turn on **Enable Dynamic Phase Alignment mode (receiver only)**. This changes the appearance of the graphic representation of the megafunction in the left-hand pane. DPA mode adds several ports and parameters that are not available in non-DPA mode. When DPA mode is turned on, one page is added to the wizard to include the additional DPA mode ports and parameters. Use DPA mode to correct skew created by different trace lengths on the data channels routed to the Stratix II device. DPA mode is also available for Stratix GX devices.

12. Turn off **Use External PLL**. In this example, the software sets the fast PLL parameters automatically. To use the Stratix II fast PLL features that cannot be accessed through the altlvds wizard, turn on this option.

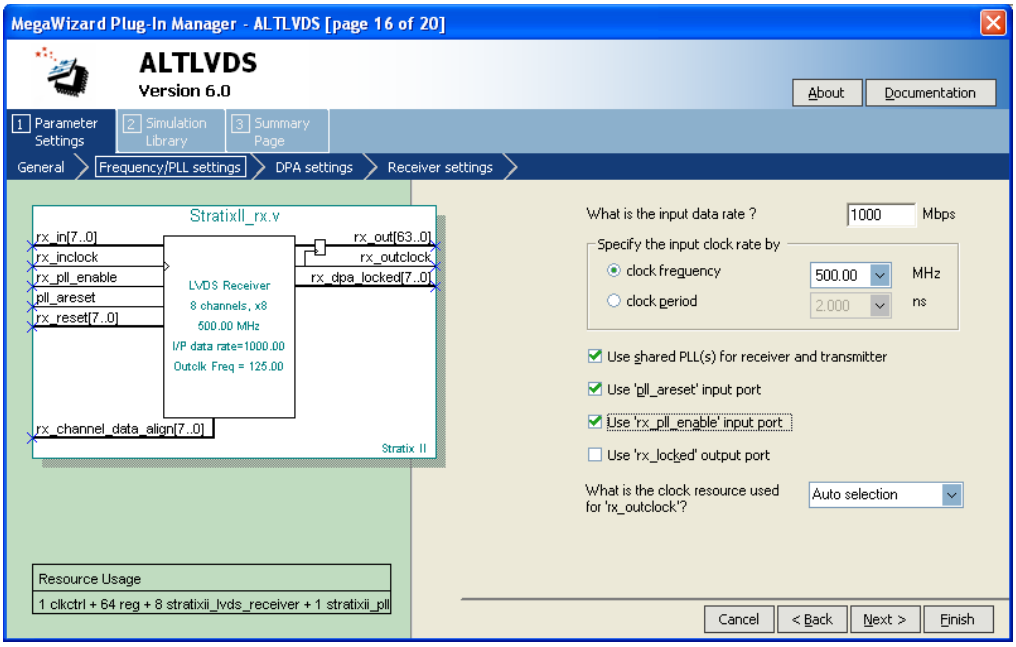
13. In the **What is the number of channels?** list, select **8**. If you want more data channels than are available in the list, you can type the number of channels you want.

Refer to the *Stratix II Device Handbook* for channel limitations for the device size you are targeting.

14. In the **What is the deserialization factor?** list, select **8**. Choose the deserialization factor from the list. The deserialization factors (J factors) available in the list are device-family dependent.

15. Click **Next**. Page 16 of the wizard appears (Figure 2–31).

*Figure 2–31. MegaWizard Plug-In Manager—altlvds [page 16 of 20]*




16. Enter `1000` for **What is the input data rate?** This example receives serial data at 1 Gbps.

17. Under **Specify the input clock rate by** click **clock frequency** and select **500 MHz**. You can specify the input clock rate either by specifying a clock frequency or a clock period. The clock frequencies and clock periods in the lists are updated with all of the possible clock frequencies and periods that correspond to the input data rate that you entered in the previous step. This example receives a 500-MHz reference clock with the serial data.

18. Turn on **Use shared PLL(s) for receiver and transmitter**. This can be used if the receiver and transmitter use the same clock.

19. In the **What is the clock resource used for rx_outclock?** list, select **Auto selection**. You can specify **Global** or **Regional clock** to assist with clock management in a Stratix II device.

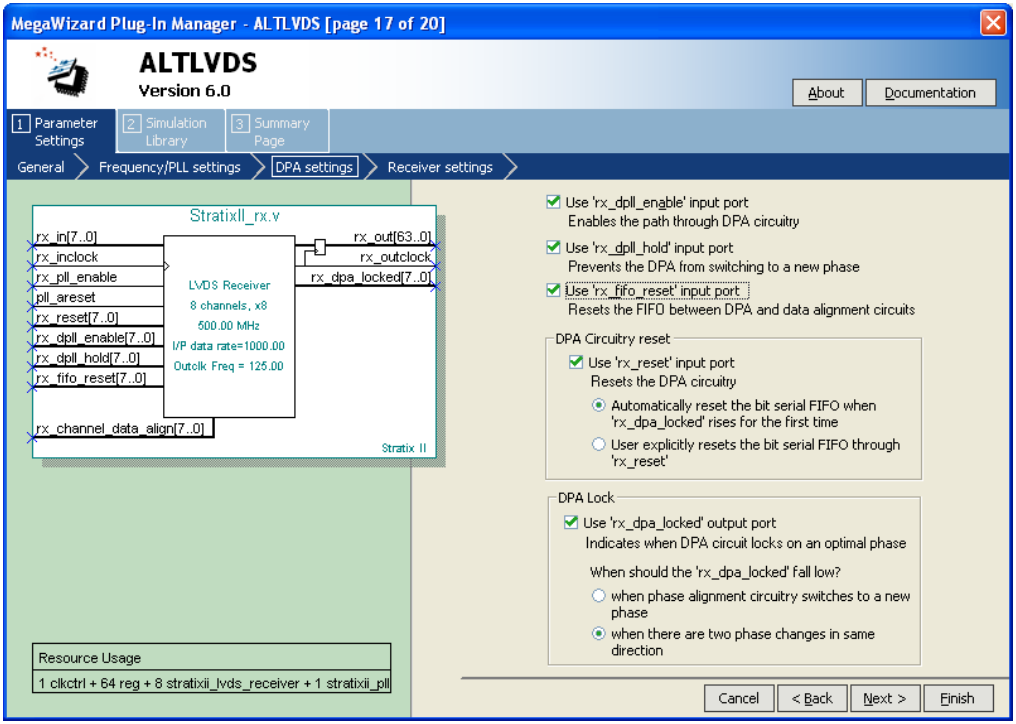
20. Turn on **Use 'pll_areset' input port**. This port gives you control of the asynchronous reset port of the fast PLL that is used with this function. When the receiver shares the PLL with the transmitter and the `pll_areset` port is used, the port must be used in both megafunctions and tied together in the design file, or the port can be omitted from both megafunctions. If the port is used in one megafunction and not the other, the PLLs are not shared and a warning is shown during compilation.

21. Turn on **Use 'rx_pll_enable' input port**. This port gives you control of the enable port of the fast PLL that is used with this function. When the receiver shares the PLL with the transmitter and the `pll_enable` port is used, the port must be used in both megafunctions, and tied together in the design file, or the port can be omitted from both megafunctions. If the port is used in one megafunction, and not the other, the PLLs are not shared and a warning is shown during compilation.

22. Turn off **Use 'rx_locked' output port** for this example. This port gives you the option of monitoring the lock status of the PLL. Because the PLLs are shared, we use the lock output from the transmitter megafunction. The status of the lock port is identical for the transmitter and receiver when shared PLLs are used.

23. Click **Next**. Page 17 of the wizard appears (Figure 2–32).

*Figure 2–32. MegaWizard Plug-In Manager—altlvds [page 17 of 20]*




24. Turn on **Use 'rx_dpll_enable' input port**. This port allows you to have dynamic channel-by-channel control of the DPA circuitry. If the DPA circuitry is needed for any channel, you can enable the DPA circuitry by setting the port corresponding to the target channel to high.

25. Turn on **Use 'rx_dpll_hold' input port**. Each DPA block continuously monitors the phase of the incoming data stream and selects a new clock phase, if needed. This port is used to prevent the DPA from switching to a new phase on a channel-by-channel basis. When this port is high, the selected channels hold their current phase setting.

26. Turn on **Use 'rx_fifo_reset' input port**. The FIFO buffer is placed between the DPA circuit and the data alignment circuit. This port is used to pass the data between the DPA and LVDS clock domains. When this port is high, the FIFO in the selected channels are reset.

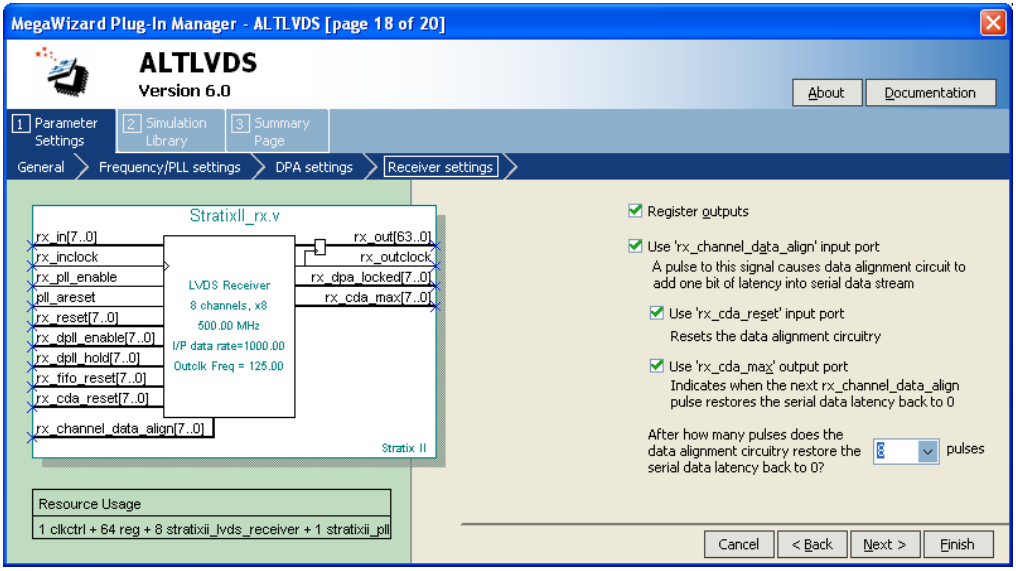
27. Turn on **Use 'rx_reset' input port**. This port resets all of the components of the DPA circuit. Choose **Automatically reset the bit serial FIFO when 'rx_dpa_locked' rises for the first time**. The DPA circuit must be re-trained after it has been reset using this port.

Refer to the *Stratix II Device Handbook* for information about training the DPA circuit.

28. Turn on **Use 'rx_dpa_locked' output port**. The DPA block samples the data on one of eight phase clocks with a 45-degree resolution between phases. This port lets you monitor the status of the DPA circuit and determine when it has locked onto the phase closest to the incoming data phase. The `rx_dpa_locked` ports de-assert (drive low), depending on the selections made in the **When should 'rx_dpa_locked' fall low?** box. This can occur when a new phase has been selected or when the DPA has made two 45-degree phase shifts in the same direction. The data may still be valid even when the `rx_dpa_locked` port is de-asserted. It is recommended that you use data checkers to validate the data in this condition.

29. Click **Next**. Page 18 of the wizard appears (Figure 2–33).

*Figure 2–33. MegaWizard Plug-In Manager—altlvds [page 18 of 20]*



30. Turn on **Register outputs**. The outputs of the receiver are registered by `rx_outclock`. If you do not register them here, you must register them in the design logic that is fed by the receiver and specify a 'Source Multiply' assignment from the receiver to the output registers with a value equal to the deserialization factor.

31. Turn on **Use 'rx_channel_data_align' input port**. This lets you control bit insertion on a channel-by-channel basis in order to align the word boundaries of the incoming data. The data slips one bit for every pulse on the `rx_channel_data_align` port. The following requirements must be met for this port:

    - The minimum pulse width is one period of the parallel clock in the logic array (`rx_outclock`).
    - The minimum low time between pulses is one period of the parallel clock.
    - There is no maximum high or low time.
    - Valid data is available two parallel clock cycles after the rising edge of `rx_channel_data_align`.

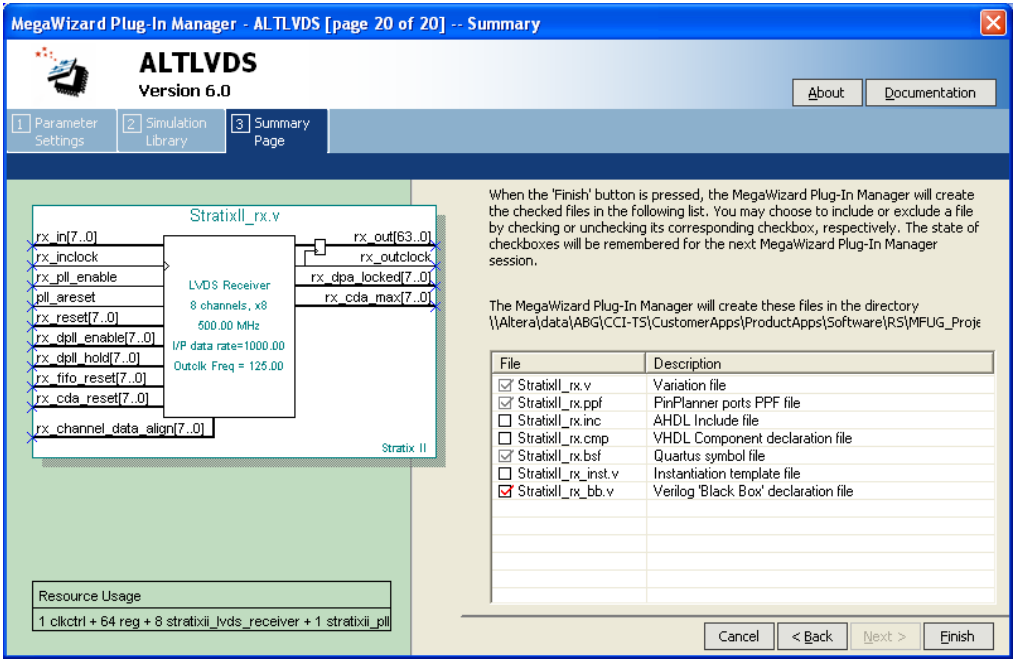
32. Turn on **Use 'rx_cda_reset' input port**. This port is available only if **Use 'rx_channel_data_align' input port** is turned on. The port resets the data alignment circuitry restoring the latency bit counter to zero.

33. Turn on **Use 'rx_cda_max' output port**. This port is available only if **Use 'rx_channel_data_align' input port** is turned on. This port indicates when the rollover point has been reached in the data alignment circuit.

34. In the **After how many pulses does the data alignment circuitry restore the serial latency back to 0?** list, select **8** pulses. Available values are 1 through 11 and do not have to be the same as the deserialization factor.

35. Click **Finish**. Page 20 of the wizard appears (Figure 2–34).

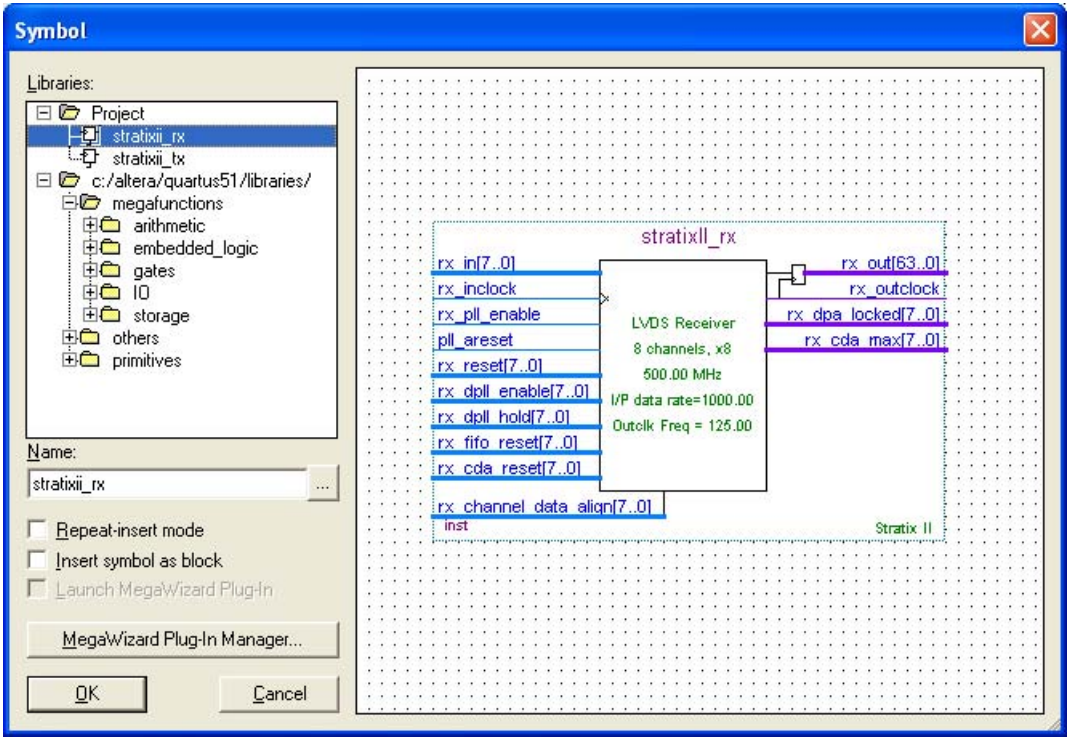
*Figure 2–34. MegaWizard Plug-In Manager—altlvds [page 20 of 20]—Summary*




The final page of the wizard shows the files that are generated for your custom megafunction variation. The gray check marks indicate files that are always generated; the other files are optional and are generated only if selected (indicated by a red check mark). Turn on the boxes to select the files that you want generated.

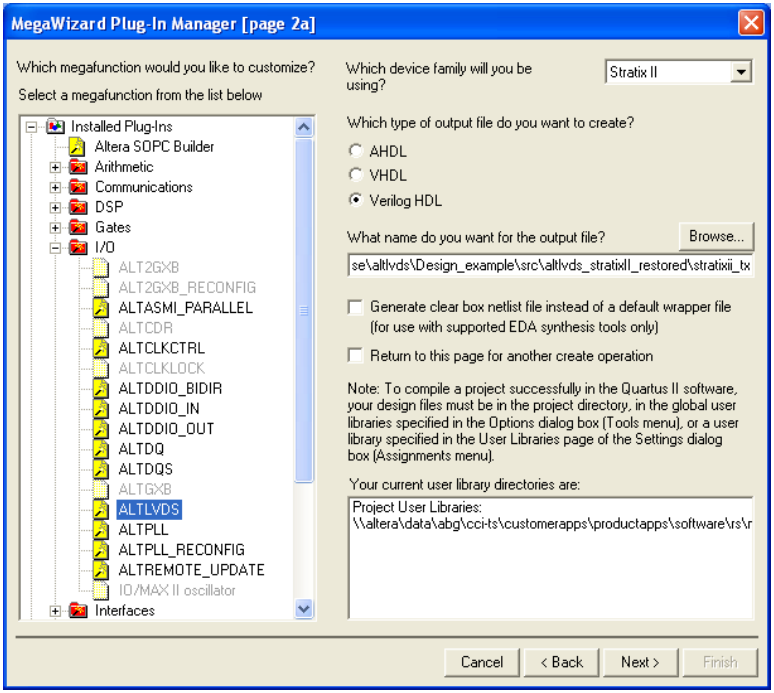
36. Click **Finish**. The symbol for the altlvds receive function you just created appears in the window (Figure 2–35).

*Figure 2–35. altlvds Receiver Symbol*




37. Click **OK**. The MegaWizard Plug-In Manager resets to page 2a so you can create a new custom function variation (Figure 2–36).

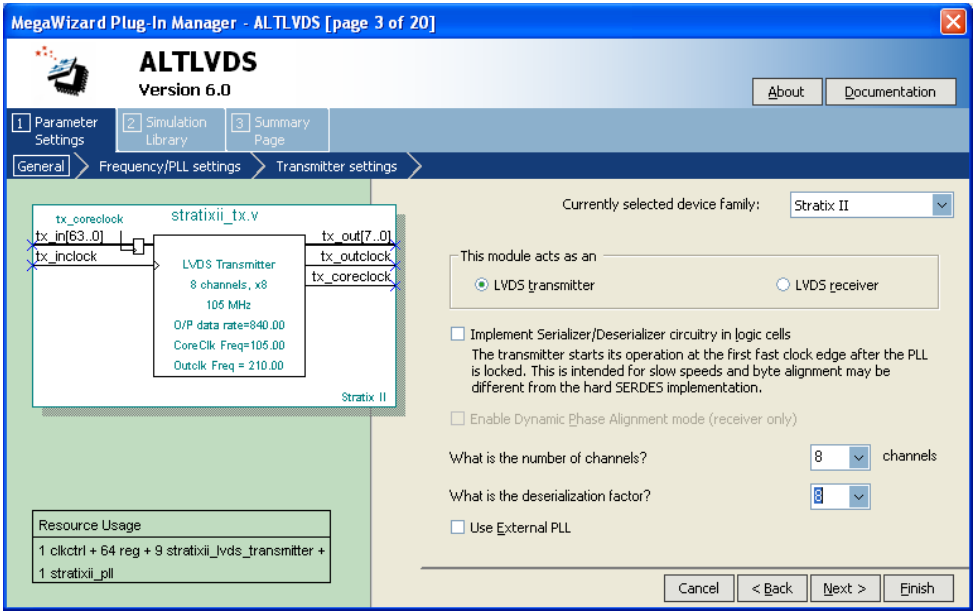
*Figure 2–36. MegaWizard Plug-In Manager [page 2a]*




38. In the **Which device family will you be using?** list, select **Stratix II**.
39. In the **Which type of output file do you want to create?** list, select **Verilog HDL**.
40. Under **Which megafunction would you like to customize?,** in the **Installed Plug-Ins** list, select **ALTLVDS**. You may have to expand the **I/O** folder.
41. Your project directory displays by default in the **What name do you want for the output file?** box. You can browse to a different directory if you want to save your megafunction there. If so, you must specify that location in the **User Libraries (Current Project)** page.
42. Enter a filename for your megafunction. This function is a transmitter. Name it `stratixii_tx`.
43. Turn off **Return to this page for another create operation**.

44. Click **Next**. Page 3 of the wizard appears (Figure 2–37).

*Figure 2–37. MegaWizard Plug-In Manager—altlvds [page 3 of 20]*




45. In the **Currently selected device family:** list, ensure that **Stratix II** is selected.

46. In the **This module acts as an** section, turn on **LVDS transmitter**.

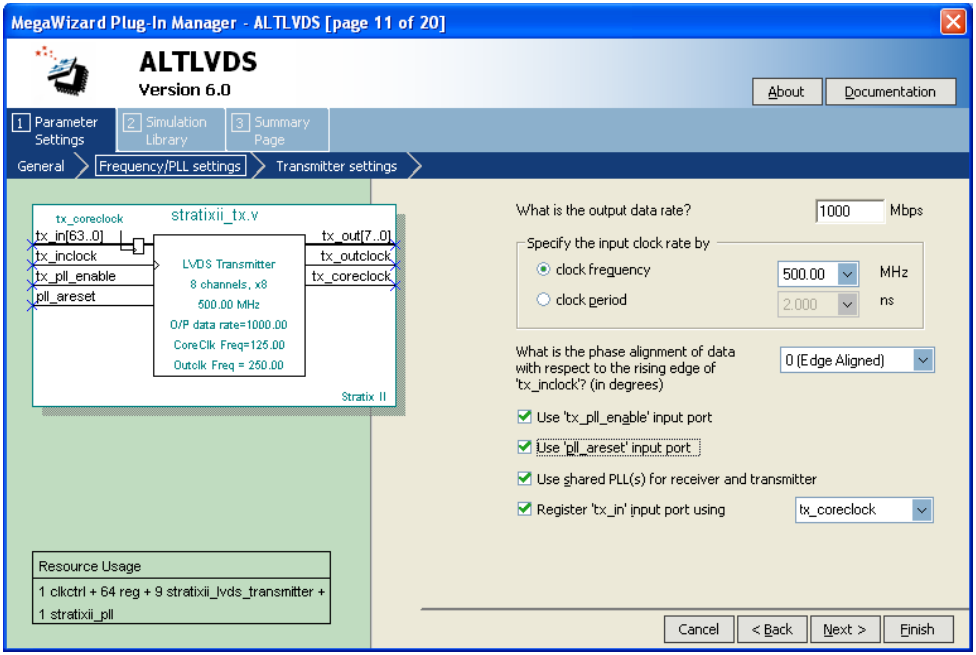
47. Turn off **Use External PLL**. In this example, the software sets the fast PLL parameters automatically. If you want to take advantage of the Stratix II fast PLL features that cannot be accessed through the altlvds wizard, turn on this option.

48. In the **What is the number of channels?** list, select **8**. If you want more channels than are available in the list, you can type the number of channels you want. Refer to the *Stratix II Device Handbook* for channel limitations for the device size you are targeting.

49. In the **What is the deserialization factor?** list, select **8**. The deserialization factors available in the list are device-family dependent. Although the wizard shows **deserialization**, the transmit function is serializing the data.

50. Click **Next**. Page 11 of the wizard appears (Figure 2–38).

*Figure 2–38. MegaWizard Plug-In Manager—altlvds [page 11 of 20]*




51. Type `1000` for **What is the output data rate?** This example transmits serial data at 1 Gbps.

52. Under **Specify the input clock rate by** click **clock frequency**, and select **500 MHz**. You can specify the input clock rate either by specifying a clock frequency or a clock period. The clock frequencies and clock periods in the lists are updated with all of the possible clock frequencies and periods that correspond to the input data rate that you entered in the previous step. This example receives a 500-MHz reference clock, the same one used by the receiver block already created. The transmitter and receiver must use the same `inclock` for the PLL to be shared.

53. In the **What is the phase alignment of data with respect to the rising edge of the 'tx_inclock'? (in degrees)** list, select **0 (Edge Aligned)**. This specifies the phase alignment of the data transmitted by the transmitter core with respect to the `tx_inclock`. This example has the data edge-aligned, but there are eight possible selections with 45-degree resolution.

54. **Register 'tx_in' input port using** is turned on by default. If you turn this parameter off, a warning message appears directing you to pre-register the inputs in the logic that feeds the transmitter and also directing you to specify values for **Multicycle** and **Multicycle Hold** assignment from the input registers to the transmitter. The values are equal to the deserialization factor minus one, and the deserialization factor, respectively.

    Leave this parameter turned on for this example. The list to the right has two options for selecting which clock to use to register the inputs: `tx_inclock` or `tx_coreclock`. Select `tx_coreclock`.

    When PLLs are shared, the `tx_inclock` should be connected to the same reference clock as the receiver function. In this example, `tx_inclock` is connected to the 500-MHz reference clock which does not equal the parallel data rate. The parallel data should be registered using `tx_coreclock` because `tx_coreclock` is equal to the output serial data rate divided by the deserialization factor. This matches the parallel data rate from the receiver.

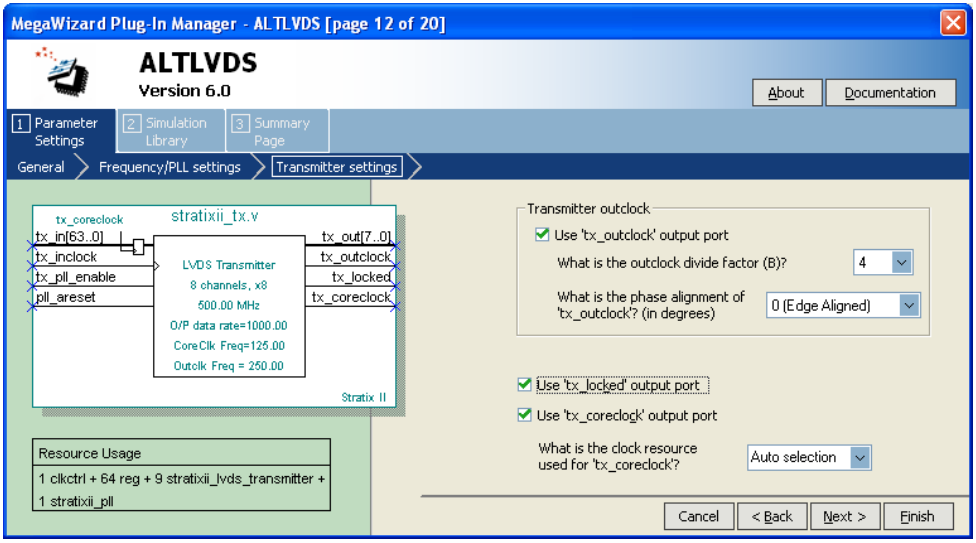
55. Turn on **Use 'tx_pll_enable' input port**. This port gives you control of the enable port of the fast PLL that is used with this function. When the transmitter shares the PLL with the receiver and the `pll_enable` port is used, the port must be used in both megafunctions and tied together in the design file, or the port can be omitted from both megafunctions. If the port is used in one megafunction and not the other, the PLLs are not shared and a warning is shown during compilation.

56. Turn on **Use 'pll_areset' input port**. This port gives you control of the asynchronous reset port of the fast PLL that is used with this function. When the transmitter shares the PLL with the receiver and the `pll_areset` port is used, the port must be used in both megafunctions and tied together in the design file, or the port can be omitted from both megafunctions. If the port is used in one megafunction and not the other, the PLLs are not shared and a warning is shown during compilation.

57. Turn on **Use shared PLL(s) for receiver and transmitter**. This setting can be used if the receiver and transmitter use the same clock.

58. Click **Next**. Page 12 of the wizard appears (Figure 2–39).

*Figure 2–39. MegaWizard Plug-In Manager—altlvds [page 12 of 20]*




59. Turn on **Use tx_locked output port**. This port gives you the option of monitoring the lock status of the PLL. The PLLs are shared so the lock output will be used from the transmitter megafunction in this example. The status of the lock port is identical between the transmitter and receiver when shared PLLs are used.

60. In the **What is the outclock divide factor (B)?** list, select **4**. Available choices are 1, 2, 4, or 8. This example uses 4 to create a `tx_outclock` of 250 MHz to be associated with the serial transmit data stream. The `tx_outclock` frequency is the data rate divided by the outclock divide factor (B).

61. In the **What is the phase alignment of tx_outclock? (in degrees)** list, select **0 (Edge Aligned)**. This specifies the phase alignment of `tx_outclock` with respect to the `tx_inclock`. This example has the output clock edge-aligned, but there are eight possible selections with 45-degree resolution available for your application.

62. Turn on **Use 'tx_locked' output port**.

63. Turn on **Use 'tx_coreclock' output port**. Although this port is not needed in this example design for anything other than registering the inputs (which does not require enabling this port), it is used to show the core clock frequency during simulation. The `tx_coreclk` port should be used to register all logic that feeds the transmit function. If another clock is used in your design, clock domain transfer is necessary.

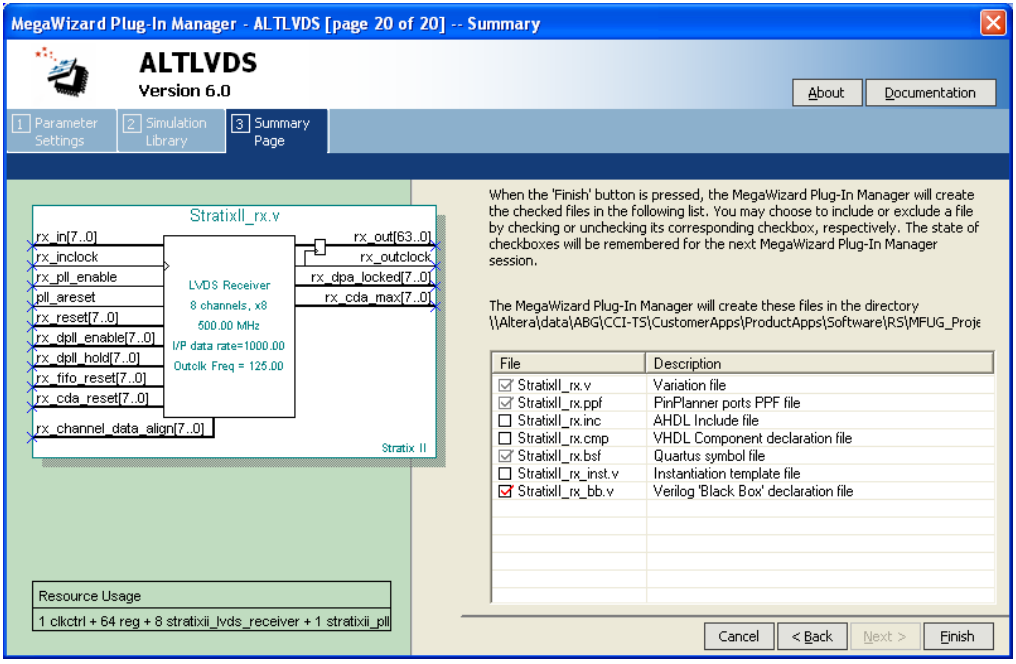
64. In the **What is the clock resource used for 'tx_coreclock'?** list, select **Auto selection**. You can specify **Global** or **Regional clock** to assist with clock management in a Stratix II device.

65. Turn on **Use 'tx_locked' output port**. This port gives you the option of monitoring the lock status of the PLL. The PLLs are shared so the lock output will be used from the transmitter megafunction in this example. The status of the lock port is identical between the transmitter and receiver when shared PLLs are used.

66. Click **Finish**. Page 20 of the wizard appears (Figure 2–40).

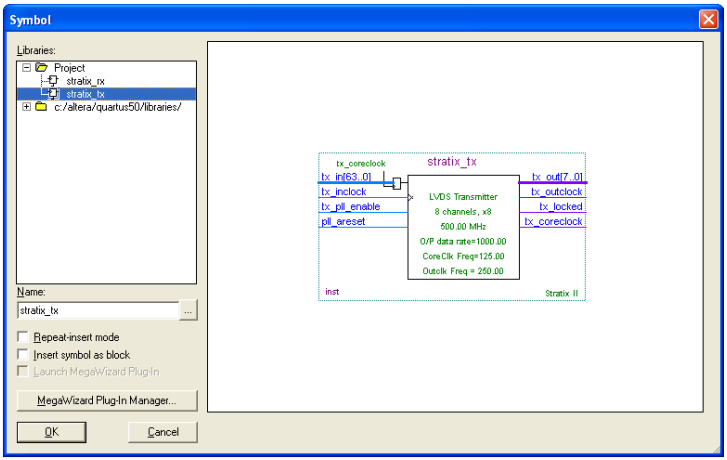
    The final page of the wizard shows the files that will be generated for your custom function variation. The gray check marks indicate files that are always generated; the other files are optional and are generated only if selected (indicated by a red check mark). Turn on the boxes to select the files that you want generated.

*Figure 2–40. MegaWizard Plug-In Manager—altlvds [page 20 of 20]—Summary*




67. Click **Finish**. The symbol for the altlvds transmit function that you just created appears (Figure 2–41).

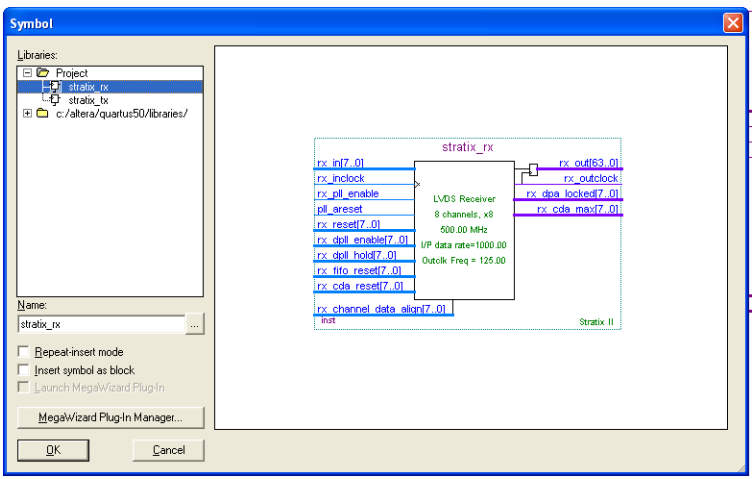
*Figure 2–41. altlvds Transmitter Symbol*




68. Click **OK** and place the **stratixii_tx** symbol in the **altlvds_stratixii** block editor design file under the text **INSERT STRATIXII_TX HERE** and align the input and output ports with the signals already present in the design file.

69. Double-click anywhere in the white space of the design file. The **Symbol** dialog box appears (Figure 2–42).

*Figure 2–42. Symbol Dialog Box*

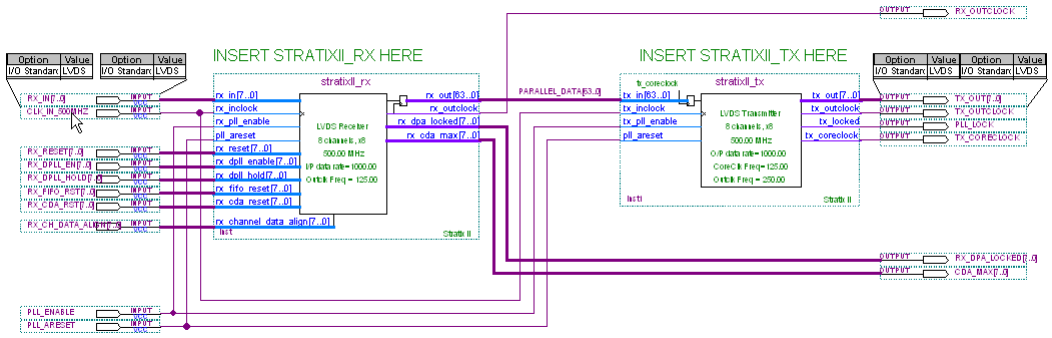


70. Choose **stratixii_rx** from the **Project** library list and click **OK**. You may have to expand the **Project** folder to see the megafunctions it contains.

71. Place the **stratixii_rx** symbol in the **altlvds_stratixII** block editor design file under the text **INSERT STRATIXII_RX HERE** and align the input and output ports with the signals already present in the design file.

72. The block diagram file should look similar to Figure 2–43.

*Figure 2–43. atlvds_stratixII.bdf*




73. On the File menu, click **Save**.

74. On the Processing menu, click **Start Compilation**.

75. When the **Simulator was successful** message box appears, click **OK**.

## Functional Results—Simulate the altlvds Receiver/Transmitter Design in Quartus II

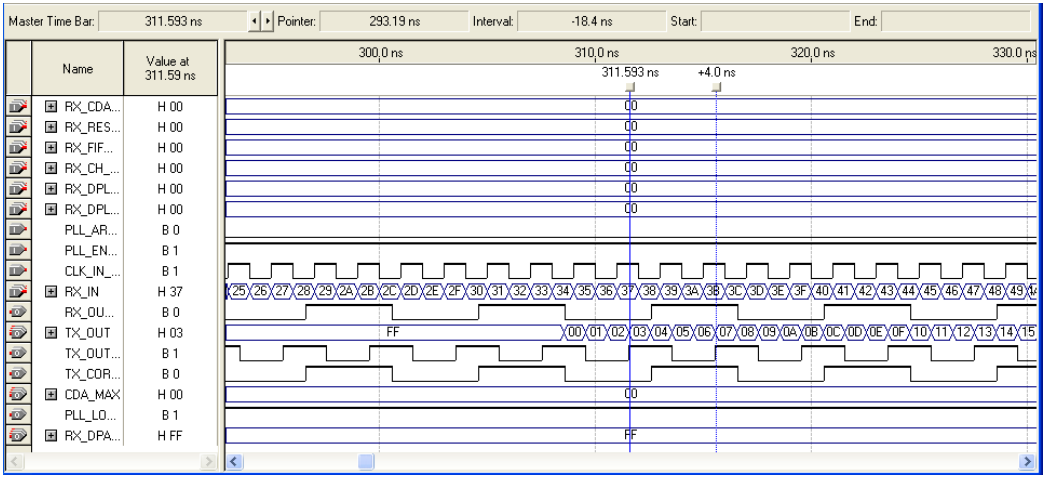
This section describes how to verify the `altlvds_stratixii` design example you just created. It illustrates the data and clock relationship on the serial channels received and transmitted by the Stratix II device. The design example files include a simple vector waveform created for this example. The following steps assume that you have successfully compiled the `altlvds_stratixii` project.

1. On the Processing menu, click **Generate Functional Simulation Netlist**.

2. When the **Functional Simulation Netlist Generation was successful** message box appears, click **OK**.
3. On the Assignments menu, click **Settings**.
4. In the Category list, select **Simulator Settings**.
5. In the Simulation mode list, select **Functional**.
6. In the Simulation input box, type `altlvds_stratixII.vwf`, or click **Browse (...)** to select the file in the project folder.
7. Turn on the **End simulation at:** option and type `1` and select **us** from the list.
8. Turn on the **Automatically add pins to simulation output waveforms** and **Simulation coverage reporting** options.
9. Turn off the **Check outputs** option.
10. Turn off **Overwrite simulation input file with simulation results**.
11. Turn off the **Generate Signal Activity File** option. Click **OK**.
12. On the Processing menu, click **Start Simulation**.
13. When the **Simulator was successful** message box appears, click **OK**.
14. Verify the simulation results in the Simulation Report window.

    Open the file **altlvds_stratixII.vwf**. A portion of the vector waveform output is shown in Figure 2–44.

*Figure 2–44. Waveform Editor*




This waveform file contains only the input and output pins that are shown. You can add internal post-compilation nodes with the Node Finder to view all of the data paths in the design. You can verify that `tx_outclock` has a 4-ns period, which corresponds to a 250-MHz clock rate, as specified in the design.

The input serial data `rx_in` is edge-aligned to the reference clock clk_in_`500MHz`, and the output serial data `tx_out` is edge-aligned to the `tx_outclock` as specified in the transmit megafunction.

This simulation output verifies that 1-Gbps data can be successfully received by and transmitted from the Stratix II device.

You can change the input vectors in the waveform editor to experiment with the other features, such as data alignment, which illustrates the functionality and helps you understand the altlvds megafunction.

## Functional Results—Simulate the altlvds Receiver/Transmitter Design in ModelSim-Altera

Simulate the design in ModelSim to compare the results of both simulators.

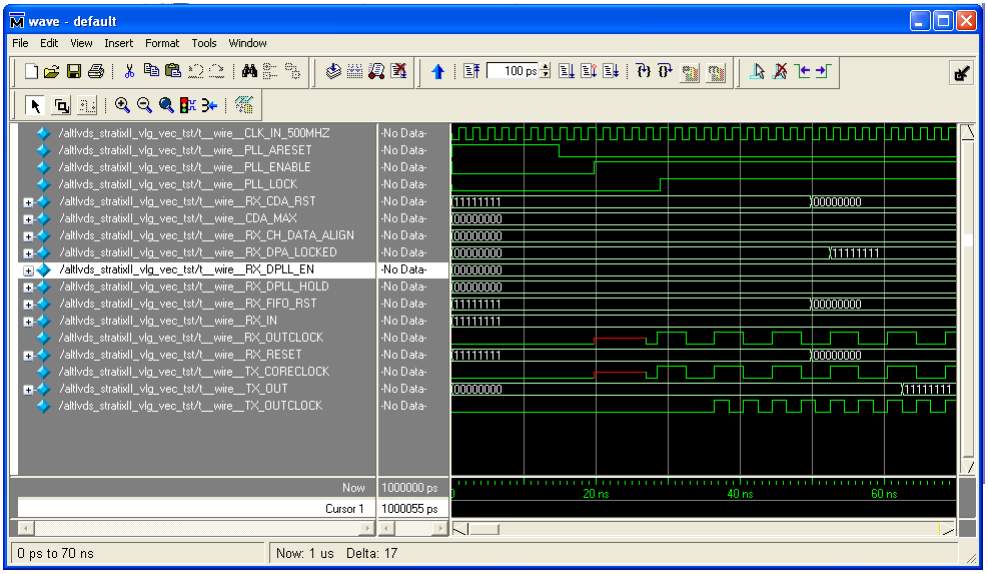
This User Guide assumes that you are familiar with using ModelSim-Altera before trying out the design example. If you are unfamiliar with using ModelSim-Altera, refer to the ModelSim-Altera support page at http://www.altera.com/support/software/products/modelsim/mod-modelsim.html. This page provides links to various topics such as installation, usage, and troubleshooting.

Set up the ModelSim-Altera simulator by performing the following steps.

1. Unzip the **altlvds_msim.zip** file to any working directory on your PC.
2. Browse to the folder in which you unzipped the files and open the **altlvds_stratixII.do** file in a text editor.
3. In line 1 of the **altlvds_stratixII.do** file, replace <*insert_directory_path_here*> with the directory path of the appropriate library files. For example, `C:/Modeltech_ae/altera/verilog/stratixii`
4. On the File menu, click **Save**.
5. Start **ModelSim-Altera**.
6. On the File menu, click **Change Directory**.
7. Select the folder in which you unzipped the files. Click **OK**.
8. On the Tools menu, click **Execute Macro**.
9. Select the **altlvds_stratixII.do** file and click **Open**. This is a script file for ModelSim that automates all the necessary settings for the simulation.
10. Verify the results by looking at the Waveform Viewer window.

    You may need to rearrange signals, remove redundant signals, and change the radix to suit the results in the Quartus II Simulator. Figure 2–45 shows the expected simulation results in ModelSim.

*Figure 2–45. ModelSim Simulation Results*



# Design Example: Cyclone II altlvds Using External PLL Option

The following design example illustrates the connection scheme used in Cyclone II devices for an altlvds receiver and transmitter using a common PLL. The altlvds_rx and altlvds_tx functions are set up using the external PLL option. This example shows step by step how to set up the altlvds_rx and altlvds_tx functions. Note that the altpll function has already been set-up in the design file.

For PLL-specific information and for examples of the PLL clock relationships, refer to the *altpll Megafunction Users Guide*.

The external PLL option is only available when using even SERDES factors.

You can use the external PLL option in altlvds to give you direct access to all of the PLL clocks which are not accessible when you allow altlvds to infer the PLL for you. This gives you the ability to use the PLL output clocks throughout your design for other functions besides the serialization and deserialization of data.

## Design Files

In this example, you will create two files using the MegaWizard Plug-In Manager: one for an altlvds receiver and one for an altlvds transmitter. All output files are created in VHDL and will be included in your project directory when completed. The PLL block is already customized in the design.

The example design files are available in the *Quartus II Projects* section of the *Design Examples* page of the Altera web site.

### *Example*

This example uses the MegaWizard Plug-In Manager in the Quartus II software. As you go through the wizard each page is described in detail for the altlvds_rx and altlvds_tx functions. A general description is provided for the altpll function. When you are finished with this example, you can incorporate it into an overall design. This example is for a 600 Mbps receiver with a deserialization factor of 8 and a 600 Mbps transmitter also with a deserialization factor of 8. The input reference clock is 75 MHz.

In this example, you will perform the following activities:

1. Generate a high-speed differential receiver using the altlvds megafunction and the MegaWizard Plug-In Manager.

2. Generate a high-speed differential transmitter using the altlvds megafunction and the MegaWizard Plug-In Manager.

3. Use guidelines to create a PLL megafunction using the MegaWizard Plug-In Manager.

4. Implement the receiver, transmitter, and PLL in the device by adding your custom megafunctions to the project and compiling the project.

5. Simulate the high-speed differential interface design.

## Generate an altlvds Receiver and altlvds Transmitter

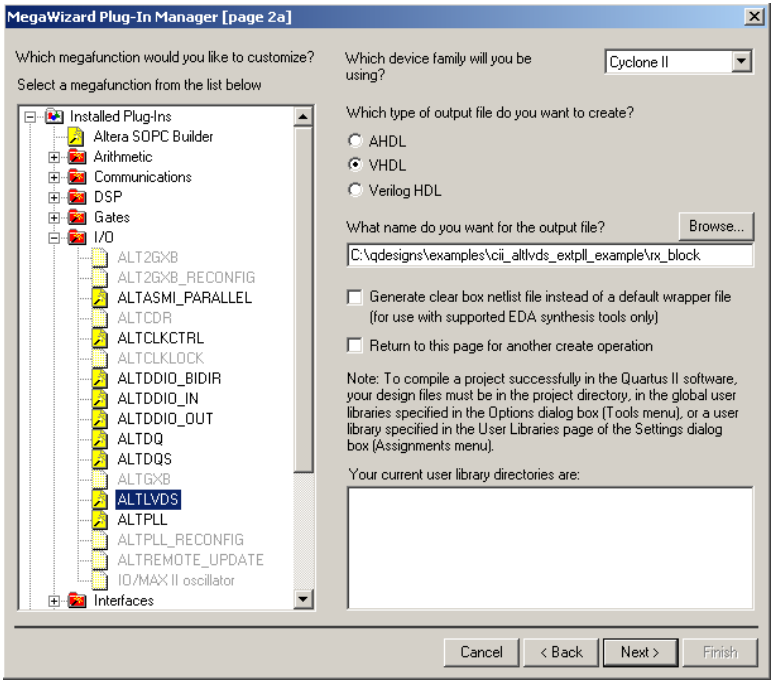
Unzip the **altlvds_DesignExample_ex2.zip** file to your preferred location. Then restore the Quartus II archive project **cii_altlvds_extpll.qar** and open the top-level block editor file **cii_altlvds_extpll.bdf**. This is an incomplete file you will complete during the course of this example. The altlvds megafunctions created in this example are added to the top-level file.

1. Double-click anywhere in the white space in the block editor file. The **Symbol** dialog box appears.

2. Click **MegaWizard Plug-In Manager**. Page 1 appears. Choose **Create a new custom megafunction variation**.

3. Click **Next**. Page 2a of the wizard appears, as shown in Figure 2–46.

*Figure 2–46. MegaWizard Plug-In Manager [page 2a]*




4. On page 2a, in the **Which device family will you be using?** list, choose **Cyclone II**.

5. Under **Which megafunction would you like to customize?**, in the **Installed Plug-Ins** list, expand the I/O folder and select **ALTLVDS**.

Your project directory displays by default in the **What name do you want for the output file?** box. You can browse to a different directory if you want to save your megafunction there.

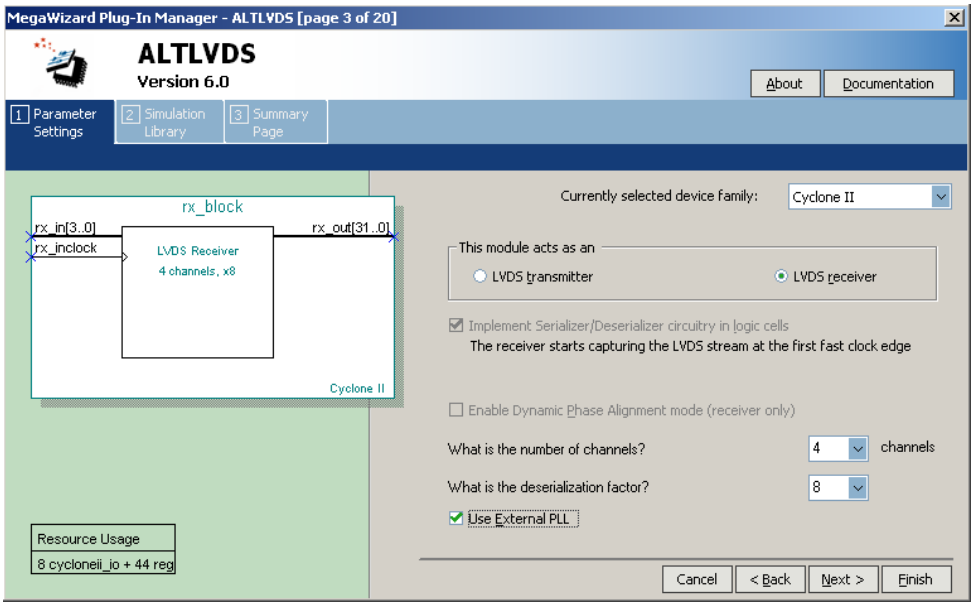
If you save the megafunction to a different directory, you must specify that location in the **User Libraries (Current Project)** page.

To access this page, on the Assignments menu, click **Settings**. In the **Settings** dialog box, in the **Category** list, select **User Libraries (Current Project)**.

6. Enter a filename for your megafunction. The first one created in this example is a receiver. Name it `rx_block`.

7. Turn off **Return to this page for another create operation**.
8. Click **Next**. Page 3 of the wizard appears, as shown in Figure 2–47.

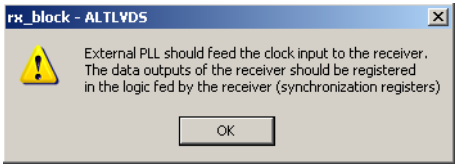
*Figure 2–47. MegaWizard Plug-In Manager—altlvds [page 3 of 20]*




9. On page 3, in the **Currently selected device family** list, ensure that **Cyclone II** is selected.
10. Under **This module acts as an**, turn on **LVDS receiver**.
11. In the **What is the number of channels?** list, select **4**. If you want more data channels than are available in the list, enter the number of channels you want. Since there is no dedicated SERDES circuitry in Cyclone II devices, the only limit to the number of channels that can be supported is the number of LVDS pin pairs you can receive in your device.
12. In the **What is the deserialization factor?** list, select **8**. Choose the deserialization factor from the list. The deserialization factors (J factors) available in the list are feature and device-family dependent. The external PLL option is not available for odd SERDES factors.

13. Turn on **Use External PLL**. A pop-up message appears, as shown in Figure 2–48.

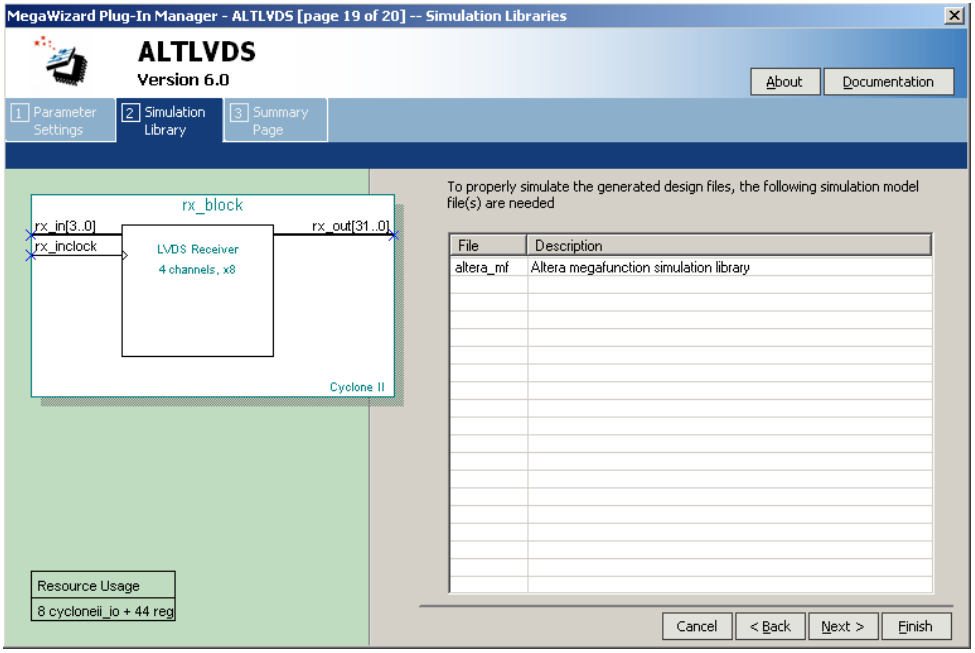
*Figure 2–48. MegaWizard Plug-In Manager [rx_block*




This message informs you that you will need to add synchronization registers to your design which must be clocked by the external PLL function.

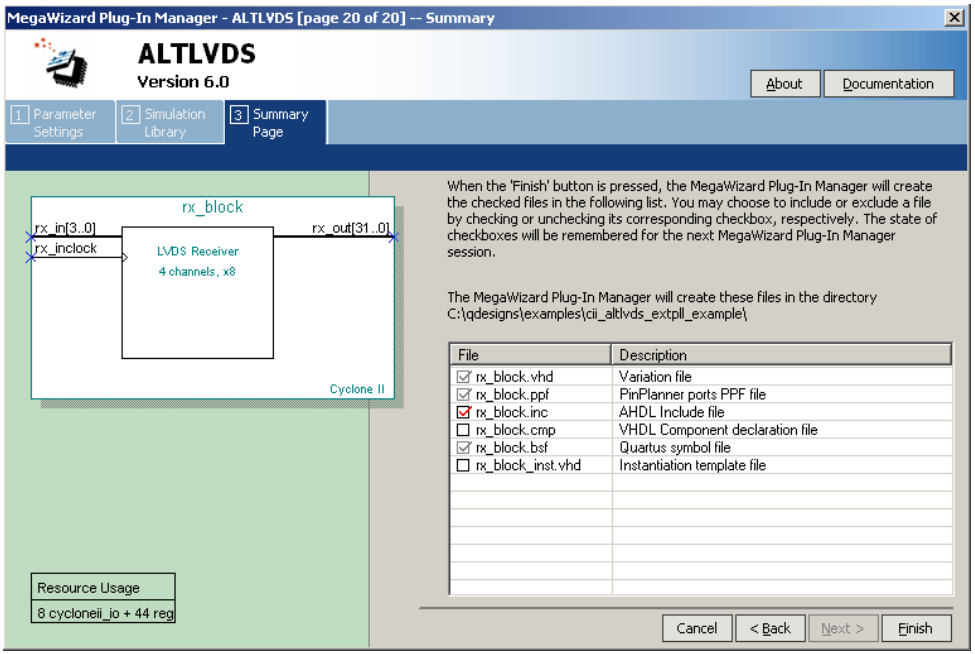
14. Click **OK**. Notice the **Resource Usage** reduces the number of registers used by `altlvds_rx`, and it no longer uses a PLL. The PLL will be added in a separate step. Click **Next**. Page 19 of the wizard appears, as shown in Figure 2–49.

*Figure 2–49. MegaWizard Plug-In Manager—altlvds [page 19 of 20]*



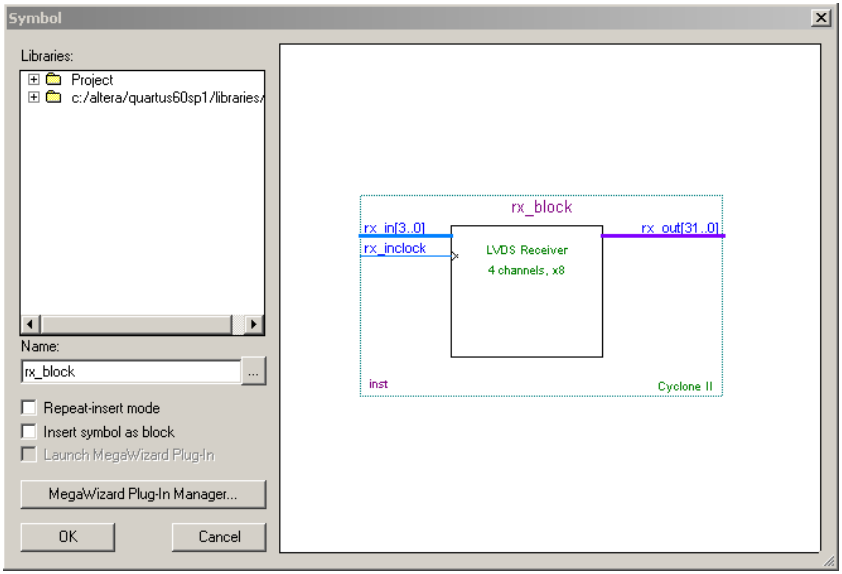
15. The wizard jumps ahead to the Simulation Library page. This shows which libraries will be needed if you choose to simulate this function in third-party simulation software. When using altlvds in external PLL mode for Cyclone II devices, no other altlvds wizard pages are necessary since all other functions and controls are accomplished through `altpll` and external register components. Click **Next**. Page 20 of the wizard appears, as shown in Figure 2–50.

*Figure 2–50. MegaWizard Plug-In Manager—altlvds [page 20 of 20]*




16. The final page of the wizard shows the files that are generated for your custom megafunction variation. The gray check marks indicate files that are always generated; the other files are optional and are generated only if selected (indicated by a red check mark). Turn on the boxes to select the files that you want generated. Click **Finish**. The symbol for the altlvds receive function you just created appears in the window (shown in Figure 2–51).

*Figure 2–51. MegaWizard Plug-In Manager—altlvds [page 19 of 20]*



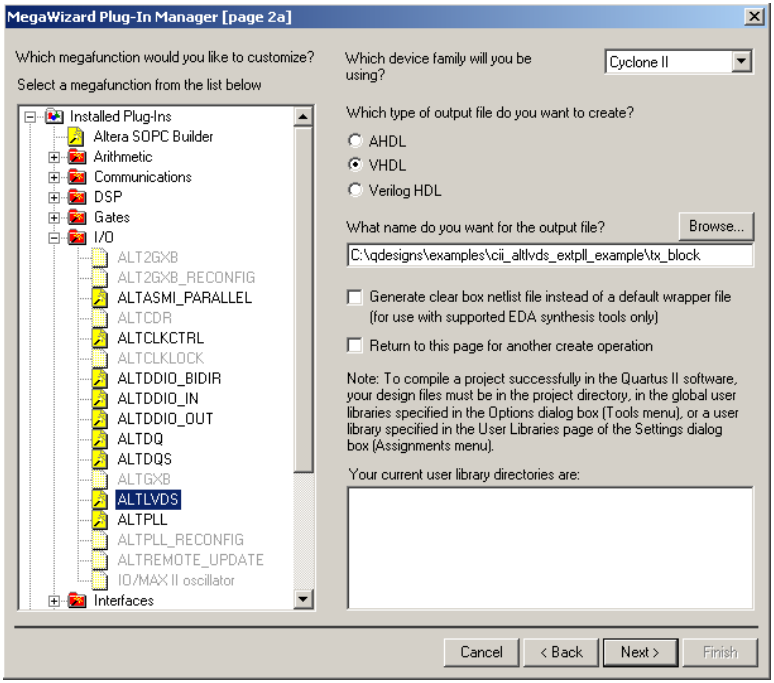

17. Click **OK** and you will be back in the block design editor with the `rx_block` attached to your cursor. Place it in the top-level file under the text **Place rx_block here** and align the input and output ports to the existing connections. Left-click to drop the `rx_block` in place.

18. Double-click in white space again to open the **Symbol** dialog box.

19. Click **MegaWizard Plug-In Manager**. Page 1 appears. Choose **Create a new custom megafunction variation**.

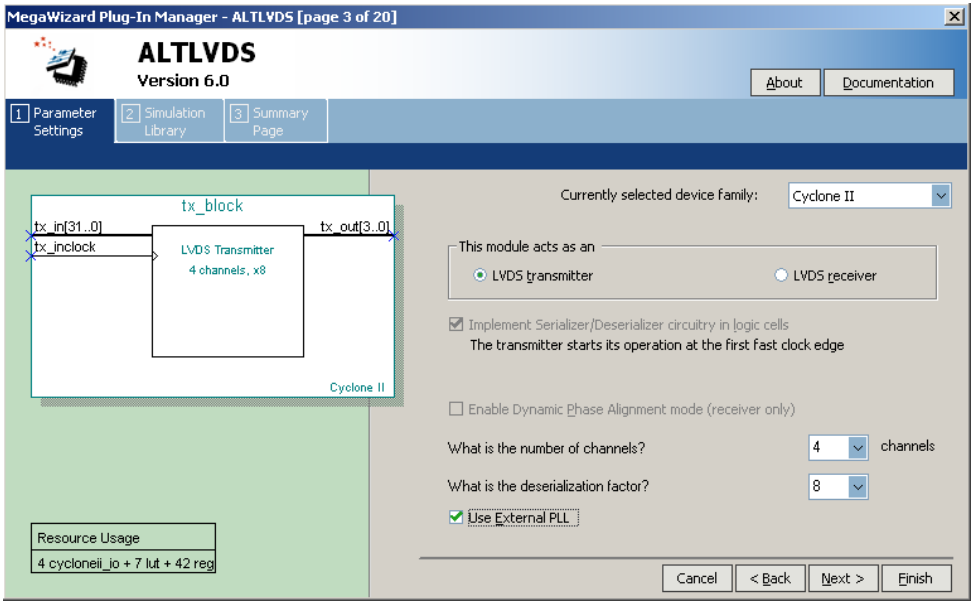
20. Click **Next**. Page 2a of the wizard appears, as shown in Figure 2–52.

*Figure 2–52. MegaWizard Plug-In Manager [page 2a]*




21. On page 2a, in the **Which device family will you be using?** list, choose **Cyclone II**.
22. Under **Which megafunction would you like to customize?**, in the **Installed Plug-Ins** list, expand the I/O folder and select **ALTLVDS**.
23. Enter a filename for your megafunction. The second one created in this example is a transmitter. Name it `tx_block`.
24. Turn off the **Return** to this page for another create operation.
25. Click **Next**. Page 3 of the wizard appears, as shown in Figure 2–53.

*Figure 2–53. MegaWizard Plug-In Manager—altlvds [page 3 of 20]*




26. On page 3, in the **Currently selected device family** list, ensure that **Cyclone II** is selected.

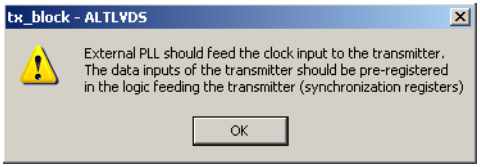
27. Under **This module acts as an**, turn on **LVDS transmitter**.

28. In the **What is the number of channels?** list, select **4**. If you want more data channels than are available in the list, enter the number of channels you wish to use. Since there is no dedicated SERDES circuitry in Cyclone II devices, the only limit to the number of channels that can be supported is the number of LVDS pin pairs you can transmit in your device.

29. In the **What is the deserialization factor?** list, select **8**. Choose the deserialization factor from the list. The deserialization factors (J factors) available in the list are feature and device-family dependent. The external PLL option is not available for odd SERDES factors.

30. Turn on **Use External PLL**. A pop-up message appears, as shown in Figure 2–54.

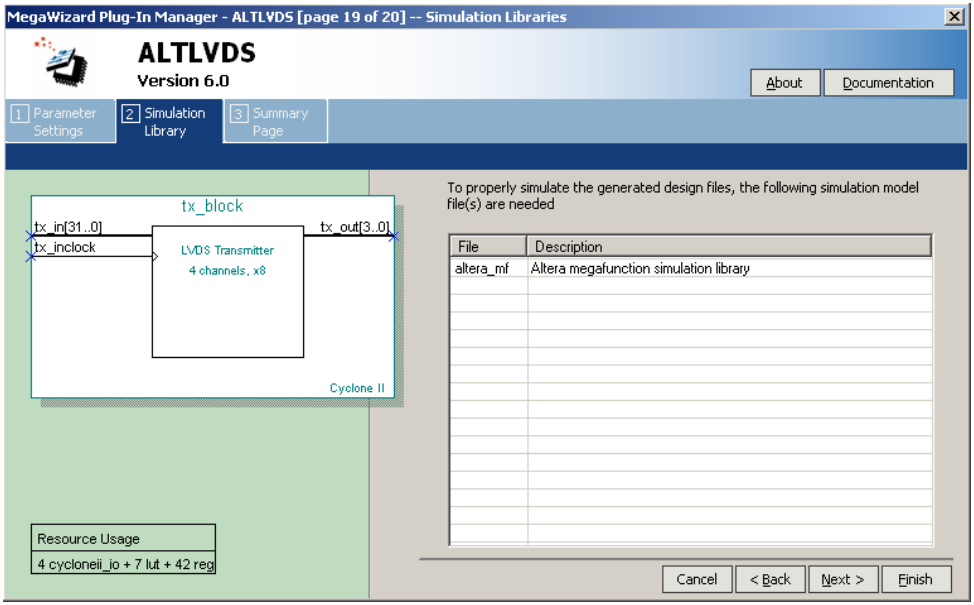
*Figure 2–54. External PLL Transmitter Pop-Up*




This message informs you that you will need to add synchronization registers to your design, which must be clocked by the external PLL function.

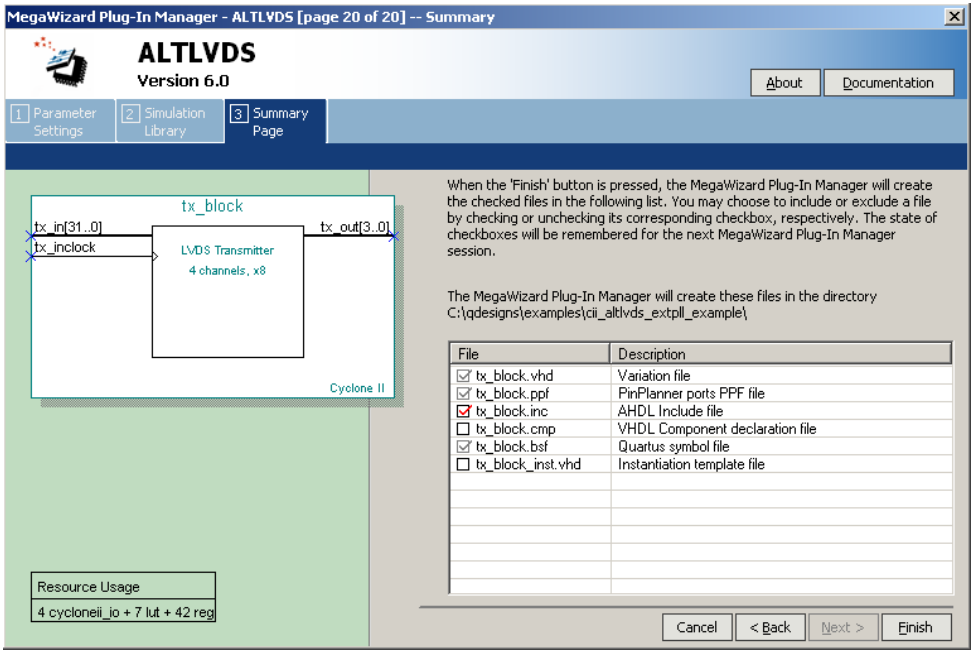
31. Click **OK**. Notice the Resource Usage reduces the number of registers used by `altlvds_tx`, and it no longer uses a PLL. The PLL will be added in a separate step. Click **Next**. Page 19 of the wizard appears, as shown in Figure 2–55.

*Figure 2–55. MegaWizard Plug-In Manager—altlvds [page 19 of 20]*



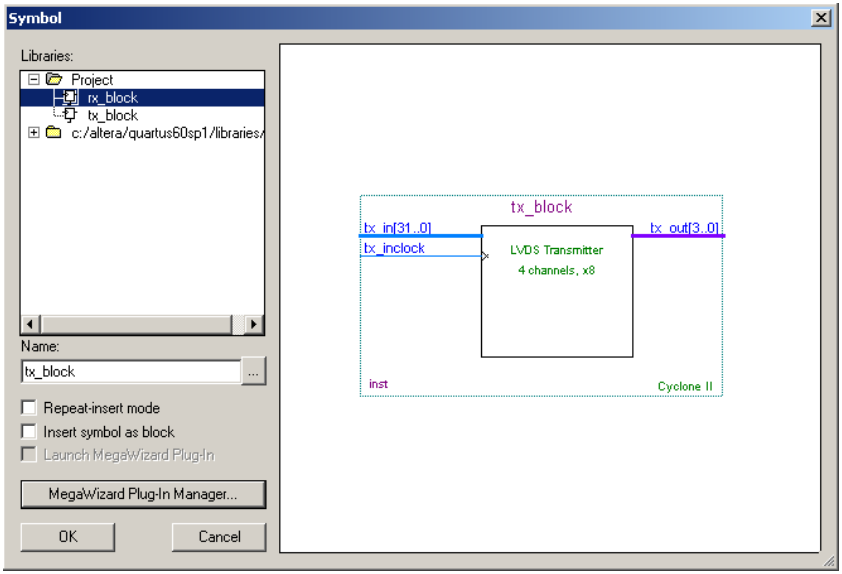
32. The wizard jumps ahead to the Simulation Library page. This shows which libraries will be needed if you choose to simulate this function in third-party simulation software. When using altlvds in external PLL mode for Cyclone II devices, no other altlvds wizard pages are necessary since all other functions and control are accomplished through `altpll` and external register components. Click **Next**. Page 20 of the wizard appears, as shown in Figure 2–56.

*Figure 2–56. MegaWizard Plug-In Manager—altlvds [page 20 of 20]*




33. The final page of the wizard shows the files that are generated for your custom megafunction variation. The gray check marks indicate files that are always generated; the other files are optional and are generated only if selected (indicated by a red check mark). Turn on the boxes to select the files that you want generated. Click **Finish**. The symbol for the altlvds transmit function you just created appears in the window (shown in Figure 2–57).

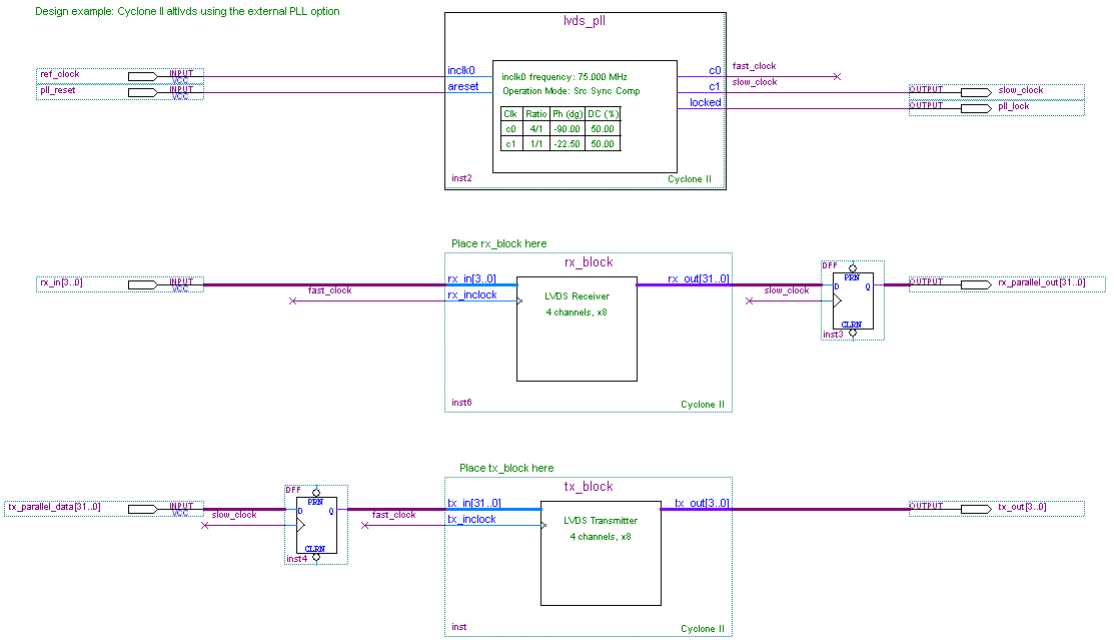
*Figure 2–57. MegaWizard Plug-In Manager—altlvds [page 19 of 20]*




34. Click **OK** to return to the block design editor with the `tx_block` attached to your cursor. Place it in the top-level file under the text **Place tx_block here** and align the input and output ports to the existing connections. Left-click to drop the `tx_block` in place.

The rest of the components for this design are already on the top-level file. They include two instantiations of DFF and one instantiation of `altpll` named `lvds_pll` in this example. The DFF instantiations are the synchronization registers mentioned in steps13 and 29. After you place the `rx_block` and `tx_block`, the top level should look like Figure 2–58.

*Figure 2–58. Top-Level Block Design File*




If you do not want to create the `rx_block` and `tx_block`, they are located in the backup folder within the project directory. You can move them to the top-level directory and insert them directly into the top-level design.

**Parameters Used by altpll**

When using altlvds in external PLL mode, the data rate and clocking is set up using `altpll`. When using Cyclone devices, you must use Normal Mode compensation. When using Cyclone II devices, you can choose either Normal Mode or Source-Synchronous Compensation Mode. Source-synchronous is the recommended compensation mode when using Cyclone II devices.

For details about how the PLL compensation modes work, refer to the PLL chapter of the Handbook for the device family you are using.

This design example is for a 600 Mbps data rate with deserialization and serialization factors of 8. The reference clock frequency is equal to the data rate divided by the deserialization/serialization factor, which is 75 MHz.

Cyclone and Cyclone II devices use DDIO registers as part of the SERDES interface. Since data is clocked on both the rising and falling edge, the clock frequency must be half the data rate; therefore, the fast clock from the PLL runs at half the frequency of the data rate. The core clock frequency (also referred to as the slow clock) is the data rate divided by the serialization factor (J). In this example, the slow clock will be the same frequency as the reference clock, but phase aligned to the fast clock from the PLL.

The PLL in this design example was created in Source-Synchronous Compensation mode.

- C0 is selected as **Which output clock will be compensated for?**
- 75 MHz is entered as **What is the frequency of the inclock0 input?**
- Ports are turned on for the asynchronous reset function and locked output.

C0 is used as the high speed clock. The data rate for this design example is 600 Mbps. C0 must be set up to be one half the data rate, so the output frequency must be set to 300 MHz. The phase you select will be dependent on your reference clock to data relationship at the pins of the device. Source-synchronous compensation mode maintains the clock to data relationship to the I/O element capture registers on the receiver.

In this design example, the reference clock and data are rising edge aligned (the bit boundary is synchronous to the rising edge of the reference clock, and 8 bits are received for every clock period). C0 must be phase-shifted to properly capture the data internally at the registers.

For edge-aligned interfaces, you would normally phase-shift your high speed clock by –180 degrees to center-align the clock and data relationship at the capture register. However, because Cyclone and Cyclone II devices use a half rate high-speed clock, due to the DDIO implementation; a –180 degree phase shift would simply switch the rising edge with the falling edge. To move the capture clock with respect to the DDIO data, a –90 degree phase shift would be required.

C1 is the low speed clock. For even SERDES factors, it must be the data rate divided by the SERDES factor (J). In this example, C1 must be 75 MHz (600 Mbps / 8). The phase shift for C1 must correspond to the phase shift on C0. The data is edge aligned with respect to the reference clock at the input pins, so the clocks have to phase shifted to center align the rising edge in the core. The high speed to low speed data transfer does not use DDIO circuitry, so you can use –180 degrees / J to determine your phase shift for this clock to data alignment. The result is –22.5 degrees. This is used as the C1 phase shift.

The receiver and transmitter use the same PLL in this example. The phase shifts were optimized to capture data at the receiver, but it has also given a fixed relationship of clock and data at the transmitter. If the slow clock is forwarded with the transmit data, it will be center aligned. If a different clock frequency and phase are desired, you can enable the `C2` output port of the PLL (in Cyclone devices, the third port of the PLL is `E0`). The frequency and phase will be restricted to the possibilities available according to the PLL parameters.

## Functional Results—Simulate the altlvds Receiver/Transmitter Design in Quartus II

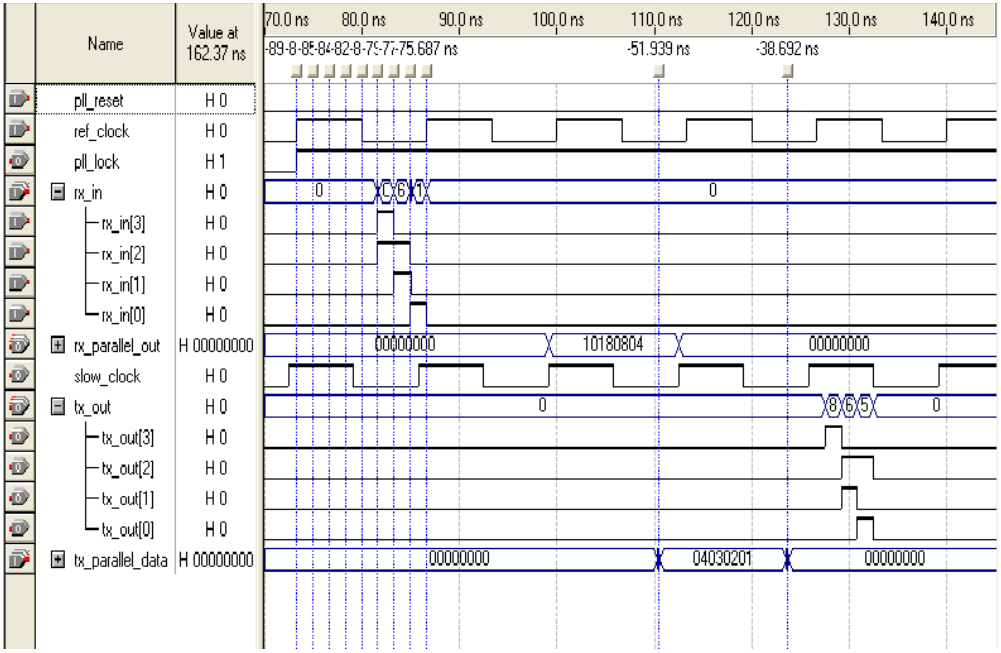
This section describes how to verify the `cii_altlvds_stratixii` design example you just created. It illustrates the data and clock relationship on the serial channels received and transmitted by the Cyclone II device. The design example files include a simple vector waveform created for this example. The following steps assume that you have successfully compiled the `cii_altlvds_extpll`.

1. On the Processing menu, click **Generate Functional Simulation Netlist**.
2. When the Functional Simulation Netlist Generation was successful message box appears, click **OK**.
3. On the Assignments menu, click **Settings**.
4. In the Category list, select **Simulator Settings**.
5. In the Simulation mode list, select **Functional**.
6. In the Simulation input box, type **cii_altlvds_extpll.vwf**, or click **Browse (...)** to select the file in the project folder.
7. Turn on **Run simulation until all vector stimuli are used**.
8. Turn on **Automatically add pins to simulation output waveforms and Simulation coverage reporting**.
9. Turn off **Check outputs**.
10. Turn off **Overwrite simulation input file with simulation results**.
11. Turn off **Generate Signal Activity File** option. Click **OK**.
12. On the Processing menu, click **Start Simulation.**

13. When the Simulator was successful message box appears, click **OK**.

14. Verify the simulation results in the Simulation Report window. A portion of the vector waveform output is shown in Figure 2–59.

*Figure 2–59. Vector Waveform Simulation Results*




## Timing Results—Simulate the altlvds Receiver/Transmitter Design in Quartus II

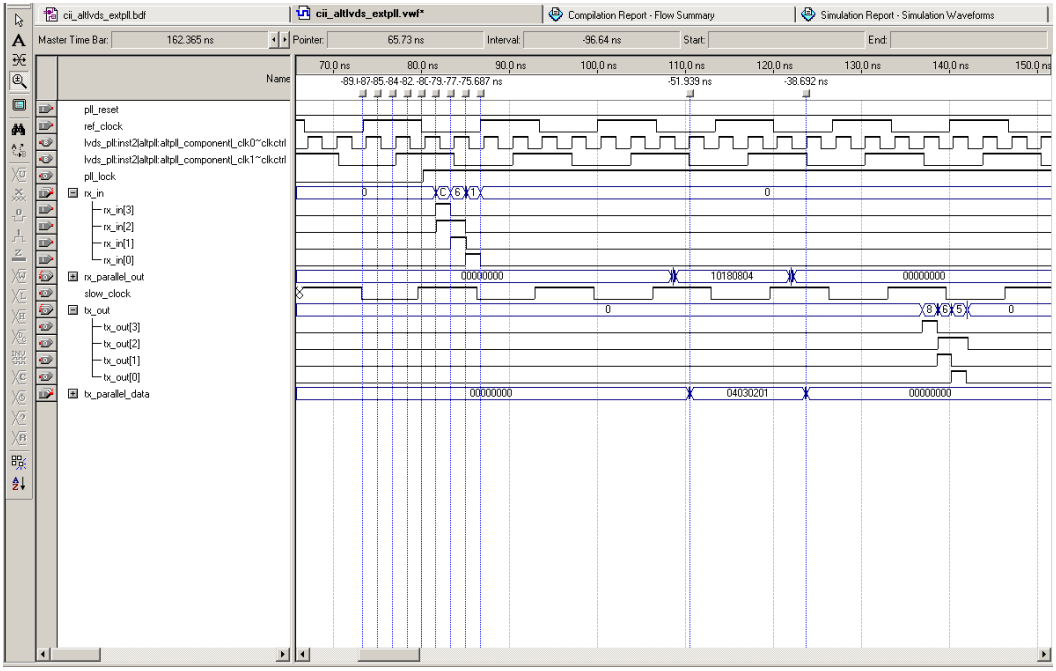
This section describes how to verify the `cii_altlvds_extpll` design example you just created via timing. It illustrates the requirement for word boundary detection logic to be added to the design. The design example files include a simple vector waveform created for this example. The following steps assume that you have successfully compiled the `cii_altlvds_extpll`.

1. On the Processing menu, click **Generate Functional Simulation Netlist**.

2. When the **Functional Simulation Netlist Generation was successful** message box appears, click **OK**.

3. On the Assignments menu, click **Settings**.
4. In the Category list, select **Simulator Settings**.
5. In the Simulation mode list, select **Timing**.
6. In the Simulation input box, type `cii_altlvds_extpll.vwf` or click **Browse (...)** to select the file in the project folder.
7. Turn on **Run simulation until all vector stimuli are used**.
8. Turn on **Automatically add pins to simulation output waveforms and Simulation coverage reporting**.
9. Turn off **Check outputs**.
10. Turn off **Overwrite simulation input file with simulation results**.
11. Turn off **Generate Signal Activity File**. Click **OK**.
12. On the Processing menu, click **Start Simulation.**
13. When the **Simulator was successful** message box appears, click **OK**.
14. Verify the simulation results in the Simulation Report window. A portion of the timing vector waveform output is shown in Figure 2–60.

*Figure 2–60. Timing Simulation Results*




The receiver channels bring serial data in to the device edge aligned with the reference clock (ref_clock). `Rx_in0` receives a data value of 1, `rx_in1` receives a data value of 2, `rx_in2` receives a data value of 3, and `rx_in3` receives a data value of 4. You can expand the parallel receiver channels (`rx_parallel_out`) in the waveform editor to see the word alignment position. The parallel data is shifted 2 bits toward the MSB in each channel.

The transmitter channel sends data from the device center aligned with the `slow_clock` output pin. Tx_out0 transmits a data value of 1, `tx_out1` transmits a data value of 2, `tx_out2` transmits a data value of 3, and `tx_out3` transmits a data value of 4. Depending on your system requirements, you can add the remaining output clock port of the PLL and forward a clock with a different position with respect to the data. The timing simulation shows the transmitted data is 3 bit positions shifted toward the MSB with respect to the `slow_clock`.

## Functional Results—Simulate the altlvds Receiver/Transmitter Design in Modelsim-Altera

Simulate the design in the ModelSim tool to compare the results of both. This User Guide assumes that you are familiar with using the ModelSim-Altera tool before trying out the design example.

If you are unfamiliar with ModelSim-Altera, refer to **www.altera.com/support/software/products/modelsim/mod-modelsim.html**. This page contains links to topics such as installation, usage, and troubleshooting.

Set up the ModelSim-Altera simulator by performing the following steps:

1. Unzip the **altlvds_ex2_msim.zip** file to any working directory on your PC.
2. Browse to the folder in which you unzipped the files and open the **altlvds_ex2.do** file in a text editor.
3. In line 1 of the **altlvds_ex2.do** file, replace <*insert_directory_path_here*> with the directory path of the appropriate library files. For example:

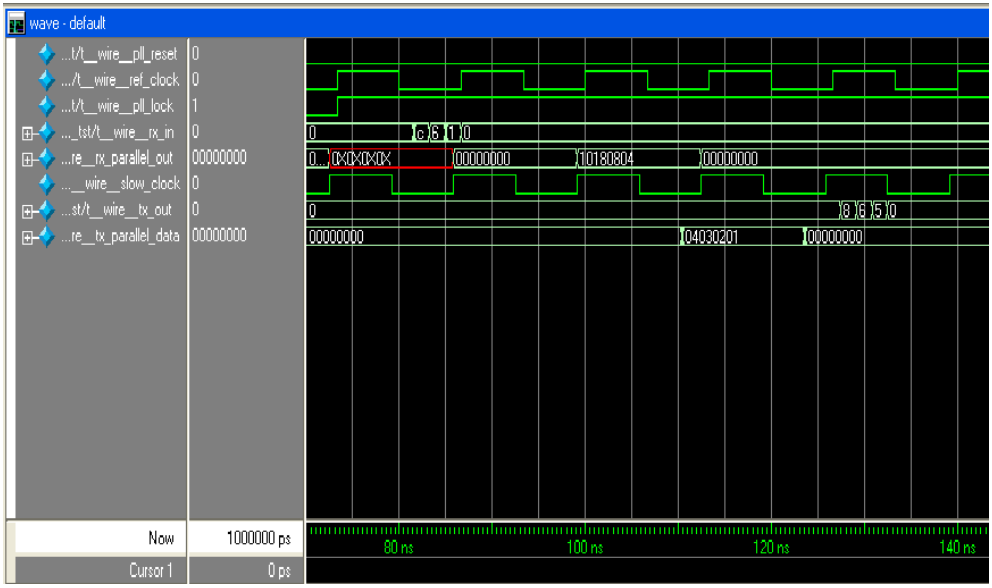
   ```
   C:/Modeltech_ae/altera/verilog/cycloneii
   ```

4. On the File menu, click **Save**.
5. Start ModelSim-Altera.
6. On the File menu, click **Change Directory**.
7. Select the folder in which you unzipped the files. Click **OK**.
8. On the Tools menu, click **Execute Macro**.
9. Select the **altlvds_ex2.do** file and click **Open**. This is a script file for ModelSim that automates all the necessary settings for the simulation.
10. Verify the results in the Waveform Viewer window.

You may need to rearrange signals, remove redundant signals, and change the radix to suit the results in the Quartus II Simulator.

Figure 2–61 shows the expected simulation results in ModelSim.

*Figure 2–61. Functional Simulation Results*




## Conclusion

The Quartus II software provides parameterizable megafunctions ranging from simple arithmetic units, such as adders and counters, to advanced phase-locked loop (PLL) blocks, multipliers, and memory structures. These megafunctions are performance-optimized for Altera devices and therefore, provide more efficient logic synthesis and device implementation, because they automate the coding process and save valuable design time. Altera recommends using these functions during design implementation so you can consistently meet your design goals.

# Chapter 3. Specifications

## Ports and Parameters

The options listed in this section describe all of the ports and parameters that are available for each device to customize the altlvds megafunction according to your application.

Enabling or disabling some parameters changes which ports are available for use in the altlvds megafunction. For example, if you enable the Dynamic Phase Alignment (DPA) mode for Stratix® III, Stratix II, Stratix II GX, or Stratix GX devices, several different ports become available, some that replace non-DPA ports, and some to control features that are unique to the DPA function. The following tables show all of the available ports and parameters and which Stratix and Cyclone® families support those features.

Table 3–1 shows the transmitter input ports, Table 3–2 shows the transmitter output ports, and Table 3–3 shows the transmitter parameters. Table 3–4 shows the receiver input ports, Table 3–5 shows the receiver output ports, and Table 3–6 shows the receiver parameters.

These parameter details are only relevant for users who by-pass the MegaWizard® Plug-In Manager interface and use the megafunction as a directly parameterized instantiation in their design. The details of these parameters are hidden from the user of the MegaWizard Plug-In Manager interface.

Refer to the latest version of the Quartus® II Help for the most current information on the ports and parameters for this megafunction.

*Table 3–1. Transmitter Input Ports*

| Port Name | Required | Stratix | Stratix GX | Stratix II, Stratix II GX, and Stratix III | Cyclone | Cyclone II and Cyclone III | Description |
|---|---|---|---|---|---|---|---|
| sync_inclock | No | — | — | — | — | — | Optional clock for the input registers. This port is available for APEX™ 20KE, APEX 20KC, Mercury™, and APEX II devices only. |
| tx_in[] | Yes | ✓ | ✓ | ✓ | ✓ | ✓ | Input port [DESERIALIZATION_FACTOR × NUMBER_OF_CHANNELS – 1..0] wide. This is parallel data that is serially transmitted by the megafunction. |
| tx_inclock | Yes | ✓ | ✓ | ✓ | ✓ | ✓ | LVDS reference input clock. |
| tx_pll_enable | No | ✓ | ✓ | ✓ | ✓ | ✓ | Enable control for the LVDS PLL. Not available in Cyclone III device. |
| pll_areset | No | ✓ | ✓ | ✓ | ✓ | ✓ | Asynchronously resets all PLL counters to initial values. |
| tx_enable | No | — | — | ✓ | — | — | This port is used only when USE_EXTERNAL_PLL is selected and must connect to the enable0 or enable1 output port of an altpll megafunction configured in LVDS mode. |

*Table 3–2. Transmitter Output Ports*

| Port Name | Required | Stratix | Stratix GX | Stratix II, Stratix II GX, and Stratix III | Cyclone | Cyclone II and Cyclone III | Description |
|---|---|---|---|---|---|---|---|
| tx_out[] | Yes | ✓ | ✓ | ✓ | ✓ | ✓ | Output port [NUMBER_OF_CHANNELS – 1..0] wide. After serialization, tx_in[*n*-1] is the first bit transmitted and tx_in[0] is the last bit transmitted for channel one; for channel two, tx_in[2*n*-1] is the first bit transmitted and tx_in[*n*] is the last bit transmitted. |
| tx_outclock | No | ✓ | ✓ | ✓ | ✓ | ✓ | External reference clock. |
| tx_coreclock | No | ✓ | ✓ | ✓ | ✓ | ✓ | Output clock used to feed non-peripheral logic. |
| tx_locked | No | ✓ | ✓ | ✓ | ✓ | ✓ | Gives the status of the LVDS PLL. When the PLL is locked, this signal is VCC. When the PLL fails to lock, this signal is GND. |

*Table 3–3. Transmitter Parameters (Part 1 of 6)*

| Parameter Name | Required | Stratix | Stratix GX | Stratix II, Stratix II GX, and Stratix III | Cyclone | Cyclone II and Cyclone III | Description |
|---|---|---|---|---|---|---|---|
| INTENDED_DEVICE_FAMILY | Yes | ✓ | ✓ | ✓ | ✓ | ✓ | [String] Select from "Stratix III ", "Stratix II GX", "Stratix II", "Stratix", "Stratix GX", "Cyclone III","Cyclone II", or "Cyclone". The ports and parameters that are available in the wizard change depending on this selection, since some features are device-dependent. |
| NUMBER_OF_CHANNELS | Yes | ✓ | ✓ | ✓ | ✓ | ✓ | [Integer] Specifies the number of LVDS channels. |
| DESERIALIZATION_FACTOR | Yes | ✓ | ✓ | ✓ | ✓ | ✓ | [Integer] Specifies the number of bits per channel. Values are:<br>**Device** / **Values**<br>APEX II, Cyclone III, Cyclone II, Cyclone, HardCopy II, and Stratix II: 1, 2, 4, 5, 6, 7, 8, 9, 10<br>APEX 20KC, APEX 20KE: 1, 4, 7, 8<br>Mercury: 1, 2, 4, 7, 8, 9, 10, 11, 12, 14, 16, 18, 20<br>Stratix and Stratix GX: 1, 2, 4, 7, 8, 10<br>Stratix III, Stratix II, and Stratix II GX: 1, 2, 4, 5, 6, 7, 8, 9, 10 |
| LPM_TYPE | Yes | ✓ | ✓ | ✓ | ✓ | ✓ | [String] Select **LVDS transmitter** in the wizard to instantiate a serializer function. For HDL instantiation, the usage is "altlvds_tx". |

*Table 3–3. Transmitter Parameters (Part 2 of 6)*

| Parameter Name | Required | Stratix | Stratix GX | Stratix II, Stratix II GX, and Stratix III | Cyclone | Cyclone II and Cyclone III | Description |
|---|---|---|---|---|---|---|---|
| USE_EXTERNAL_PLL | No | — | — | ✓ | ✓ | ✓ | [String] Values are "ON" or "OFF". Specify "ON" to prevent the altlvds function from instantiating a fast PLL automatically.<br><br>This option is not available when using odd J factors in Cyclone III, Cyclone II, and Cyclone devices. |
| OUTCLOCK_DIVIDE_BY | No | ✓ | ✓ | ✓ | ✓ | ✓ | [Integer] Specifies the period of the tx_outclock port as INCLOCK_PERIOD × OUTCLOCK_DIVIDE_BY and the frequency of the tx_outclock port as INCLOCK PERIOD / OUTCLOCK_DIVIDE_BY. The default value for this parameter is DESERIALIZATION_FACTOR. |
| CORECLOCK_DIVIDE_BY | No | — | — | — | ✓ | ✓ | [Integer] Values are 1 or 2. Specifies the CORECLOCK output frequency to either be CORECLOCK or CORECLOCK divided by 2. This is only available when using odd SERDES factors. When using a divide by factor of 1 less device resources are used but you may not be able to achieve timing at higher data rates. Altera recommends using a divide by factor of 2 for higher data rates. |

*Table 3–3. Transmitter Parameters (Part 3 of 6)*

| Parameter Name | Required | Stratix | Stratix GX | Stratix II, Stratix II GX, and Stratix III | Cyclone | Cyclone II and Cyclone III | Description |
|---|---|---|---|---|---|---|---|
| OUTCLOCK_ALIGNMENT<br><br>Refer to Figure 3–1. | Yes | ✓ | ✓ | ✓ | ✓ | ✓ | [String] Specifies the alignment of tx_outclk with respect to the rising edge of tx_inclock. Figure 1 shows the relationship between tx_inclock and tx_ouclock for three different values of OUTCLOCK_ALIGNMENT. Values are "EDGE_ALIGNED", "45_DEGREES", "90_DEGREES", "135_DEGREES", "CENTER_ALIGNED", "235_DEGREES", "270_DEGREES", and "315_DEGREES". If omitted, the default is "EDGE_ALIGNED". |
| INCLOCK_PERIOD | Yes | — | — | — | — | — | [Integer] Specifies the input clock either by frequency (MHz) or period (ns in the wizard or ps in HDL code). This parameter is available for APEX 20KC, APEX 20KE, APEX II, and Mercury devices only. |
| OUTPUT_DATA_RATE | Yes | ✓ | ✓ | ✓ | ✓ | ✓ | [Integer] Specifies the data rate out of the PLL. The multiplication value for the PLL is OUTPUT_DATA_RATE / INCLOCK_PERIOD. |

*Table 3–3. Transmitter Parameters (Part 4 of 6)*

| Parameter Name | Required | Stratix | Stratix GX | Stratix II, Stratix II GX, and Stratix III | Cyclone | Cyclone II and Cyclone III | Description |
|---|---|---|---|---|---|---|---|
| INCLOCK_DATA_ALIGNMENT<br><br>Refer to Figure 3–2. | Yes | ✓ | ✓ | ✓ | ✓ | ✓ | [String] Specifies the alignment of input data with respect to the rising edge of the rx_inclock port. Figure 1 shows the relationship between input data and the clock for three different values of INCLOCK_DATA_ALIGNMENT. Supported values are "EDGE_ALIGNED", "45_DEGREES", "90_DEGREES", "135_DEGREES", "CENTER_ALIGNED", "235_DEGREES", "270_DEGREES", and "315_DEGREES". If omitted, the default is "EDGE_ALIGNED". |
| REGISTERED_INPUT | No | ✓ | ✓ | ✓ | ✓ | ✓ | [String] Indicates whether the tx_in[] ports should be registered. Values are "ON" and "OFF" and "TX_INCLK" and "TX_CORECLK" to select which clock will be used to register the parallel data. Tx_inclk must be used when the parallel data rate is equal to the tx_inclk frequency, but not equal to the tx_coreclk frequency. Tx_coreclk must be used when the parallel data rate is not equal to the tx_inclk frequency. When the tx_inclk frequency is equal to the tx_coreclk frequency, either can be used to register the inputs. The frequency of the tx_coreclk is equal to the OUTPUT_DATA_RATE / DESERIALIZATION_FACTOR. |

*Table 3–3. Transmitter Parameters (Part 5 of 6)*

| Parameter Name | Required | Stratix | Stratix GX | Stratix II, Stratix II GX, and Stratix III | Cyclone | Cyclone II and Cyclone III | Description |
|---|---|---|---|---|---|---|---|
| COMMON_RX_TX_PLL | No | ✓ | ✓ | ✓ | ✓ | ✓ | [String] Specifies whether the compiler may use the same PLL for both the LVDS receiver and the LVDS transmitter, provided they have the same input clock frequency. Values are "ON" and "OFF". If omitted, the default is "ON". |
| OUTCLOCK_RESOURCE | No | ✓ | ✓ | ✓ | ✓ | ✓ | [String] Specifies the resource used for the output clock. Values are "AUTO", "GLOBAL CLOCK", or "REGIONAL CLOCK". |
| IMPLEMENT_IN_LES | No | ✓ | ✓ | ✓ | ✓ | ✓ | [String] Specifies whether to implement serializer/deserializer circuitry in logic cells, which allows the circuitry to behave similar to Stratix LVDS circuitry. The IMPLEMENT_IN_LES parameter should be used for SERDES functions that require data rates that are lower than the dedicated circuitry. Values are "ON" and "OFF". This parameter is available for Stratix III, Stratix II, Stratix II GX, Stratix, and Stratix GX devices only. This parameter is always enabled for Cyclone III, Cyclone II, and Cyclone devices only. |
| MULTI_CLOCK | No | — | — | — | — | — | [String] Indicates whether the sync_inclock port is used for input registering. Values are "ON" and "OFF." If omitted, the default is "OFF". This parameter is available for APEX 20KC, APEX 20KE, Mercury, and APEX II devices only. |

*Table 3–3. Transmitter Parameters (Part 6 of 6)*

| Parameter Name | Required | Stratix | Stratix GX | Stratix II, Stratix II GX, and Stratix III | Cyclone | Cyclone II and Cyclone III | Description |
|---|---|---|---|---|---|---|---|
| INCLOCK_BOOST | No | — | — | — | — | — | [Integer] The effective clock period used to sample output data. Values are 1, 2, and 4-10. This parameter is available for APEX II and Mercury devices only. |
| CENTER_ALIGN_MSB | No | — | — | — | — | — | [String] Indicates whether the most significant bit center alignment is used for data bit alignment. Values are "ON" and "OFF". If omitted, the default is "OFF". This parameter is available for APEX II devices only. |

*Table 3–4. Receiver Input Ports (Part 1 of 3)*

| Port Name | Required | Stratix | Stratix GX | Stratix II, Stratix II GX, and Stratix III | Cyclone | Cyclone II and Cyclone III | Description |
|---|---|---|---|---|---|---|---|
| rx_in[] | Yes | ✓ | ✓ | ✓ | ✓ | ✓ | Input port [NUMBER_OF_CHANNELS - 1..0] wide. After deserialization, rx_out[*n*-1] is the first bit received and rx_out[0] is the last bit received for channel one; for channel two, rx_out[2*n*-1] is the first bit received and rx_out[*n*] is the last bit received. |
| rx_inclock | Yes | ✓ | ✓ | ✓ | ✓ | ✓ | LVDS reference input clock. |
| rx_pll_enable | No | ✓ | ✓ | ✓ | ✓ | ✓ | Enable control for the LVDS PLL. This port is not available in Cyclone III. |
| rx_data_align | No | ✓ | ✓ | — | — | — | Controls byte alignment circuitry by delaying the data on all channels by one bit when asserted. You can register this port using the rx_outclock port. For Stratix GX devices, this port is available in non-DPA mode. |
| rx_channel_data_align[] | No | — | ✓ | ✓ | — | — | Input port [NUMBER_OF_CHANNELS - 1..0] wide. Controls byte alignment circuitry. A pulse to this input causes the data alignment circuitry to remove one bit of latency from the serial data stream in Stratix GX devices, or add one bit of latency into the serial data stream for Stratix III, Stratix II, and Stratix II GX devices. You can register this port using the rx_outclock port. For Stratix GX devices, this port is available in DPA mode. |

*Table 3–4. Receiver Input Ports (Part 2 of 3)*

| Port Name | Required | Stratix | Stratix GX | Stratix II, Stratix II GX, and Stratix III | Cyclone | Cyclone II and Cyclone III | Description |
|---|---|---|---|---|---|---|---|
| rx_dpll_enable[] *(1)* | No | — | — | ✓ | — | — | Input port [NUMBER_OF_CHANNELS – 1..0] wide. Enables the path through DPA circuitry. |
| rx_dpll_hold[] *(1)* | No | — | — | ✓ | — | — | Input port [NUMBER_OF_CHANNELS – 1..0] wide. Prevents the DPA from switching to a new phase. |
| rx_reset[] *(1)* | No | — | ✓ | ✓ | — | — | Input port [NUMBER_OF_CHANNELS – 1..0] wide. Synchronizes all channels and resets the DPA circuitry. This port must be connected if the ENABLE_DPA_MODE parameter is turned on in Stratix GX. |
| rx_dpll_reset[] *(1)* | No | — | ✓ | — | — | — | Input port [NUMBER_OF_CHANNELS - 1..0] wide. Asynchronous reset for all channels. |
| rx_fifo_reset[] *(1)* | No | — | — | ✓ | — | — | Input port [NUMBER_OF_CHANNELS - 1..0] wide. Resets the FIFO between DPA and data alignment circuits. |
| rx_cda_reset[] | No | — | — | ✓ | — | — | Input port [NUMBER_OF_CHANNELS - 1..0] wide. Resets the data alignment circuitry. |
| rx_coreclk[] *(1)* | No | — | ✓ | — | — | — | Input port [NUMBER_OF_CHANNELS - 1..0] wide. LVDS reference input clock which is used to synchronize the output of the receiver to a clock used in the core. One clock is used for each channel. |

*Table 3–4. Receiver Input Ports (Part 3 of 3)*

| Port Name | Required | Stratix | Stratix GX | Stratix II, Stratix II GX, and Stratix III | Cyclone | Cyclone II and Cyclone III | Description |
|---|---|---|---|---|---|---|---|
| pll_areset | No | ✓ | ✓ | ✓ | ✓ | ✓ | Asynchronously resets all counters to initial values. |
| rx_enable | No | — | — | ✓ | — | — | This port is used only when USE_EXTERNAL_PLL is selected and must connect to the enable0 or enable1 output port of an altpll megafunction configured in LVDS mode. |
| rx_deskew | No | — | — | — | — | — | Specifies whether to activate calibration mode. This port is available for APEX 20KE and APEX 20KC devices only. |

*Note to Table 3–4:*

(1) These functions are only available when using DPA mode in Stratix III, Stratix II, Stratix II GX, and Stratix GX devices.

*Table 3–5. Receiver Output Ports*

| Port Name | Required | Stratix | Stratix GX | Stratix II, Stratix II GX, and Stratix III | Cyclone | Cyclone II and Cyclone III | Description |
|---|---|---|---|---|---|---|---|
| rx_out[] | Yes | ✓ | ✓ | ✓ | ✓ | ✓ | Output port [DESERIALIZATION_FACTOR × NUMBER_OF_CHANNELS - 1..0] wide. Deserialized data signal. |
| rx_outclock | No | ✓ | ✓ | ✓ | ✓ | ✓ | Internal reference clock. |
| rx_locked | No | ✓ | ✓ | ✓ | ✓ | ✓ | Provides the status of the LVDS PLL. When the PLL is locked, this signal is VCC. When the PLL fails to lock, this signal is GND. |
| rx_dpa_locked[] | No | — | — | ✓ | — | — | Output port [NUMBER_OF_CHANNELS - 1..0] wide. Indicates whether the channel is locked to the phase of rx_in[] when in DPA mode. |
| rx_cda_max[] | No | — | — | ✓ | — | — | Output port [NUMBER_OF_CHANNELS - 1..0] wide. Indicates when the next rx_channel_data_align[] pulse restores the serial data latency back to 0. |

*Table 3–6. Receiver Parameters (Part 1 of 7)*

| Parameter Name | Required | Stratix | Stratix GX | Stratix II, Stratix II GX, and Stratix III | Cyclone | Cyclone II and Cyclone III | Description |
|---|---|---|---|---|---|---|---|
| INTENDED_DEVICE_FAMILY | Yes | ✓ | ✓ | ✓ | ✓ | ✓ | [String] Select from "Stratix III", "Stratix II GX", "Stratix II", "Stratix", "Stratix GX", "Cyclone III", "Cyclone II", or "Cyclone". The ports and parameters that are available in the wizard change depending on this selection, since some features are device-dependent. |
| NUMBER_OF_CHANNELS | Yes | ✓ | ✓ | ✓ | ✓ | ✓ | [Integer] Specifies the number of LVDS channels. |
| DESERIALIZATION_FACTOR | Yes | ✓ | ✓ | ✓ | ✓ | ✓ | [Integer] Specifies the number of bits per channel. |
| ENABLE_DPA_MODE | No | — | ✓ | ✓ | — | — | [String] Turns on DPA mode. Values are "ON" and "OFF"; if omitted, the default value is "OFF". |
| LPM_TYPE | No | ✓ | ✓ | ✓ | ✓ | ✓ | [String] Select **LVDS receiver** in the wizard to instantiate a deserializer function. For HDL instantiation, the usage is "altlvds_rx". |
| USE_EXTERNAL_PLL | No | — | — | ✓ | ✓ | ✓ | [String] Values are "ON" or "OFF". Specify "ON" to prevent the altlvds function from instantiating a fast PLL automatically.<br><br>This option is not available when using odd J factors in Cyclone III, Cyclone II, and Cyclone devices. |

*Table 3–6. Receiver Parameters (Part 2 of 7)*

| Parameter Name | Required | Stratix | Stratix GX | Stratix II, Stratix II GX, and Stratix III | Cyclone | Cyclone II and Cyclone III | Description |
|---|---|---|---|---|---|---|---|
| COMMON_RX_TX_PLL | No | ✓ | ✓ | ✓ | ✓ | ✓ | [String] Specifies whether the compiler may use the same PLL for both the LVDS receiver and the LVDS transmitter, provided they have the same input clock frequency. Values are "ON" and "OFF". If omitted, the default is "ON". |
| OUTCLOCK_RESOURCE | No | ✓ | ✓ | ✓ | — | — | [String] Specifies the resource used for the output clock. Values are "AUTO", "GLOBAL CLOCK", or "REGIONAL CLOCK". |
| INCLOCK_PERIOD | Yes | ✓ | ✓ | ✓ | ✓ | ✓ | [Integer] Specify the input clock either by frequency (MHz) or period (ns in the wizard or ps in HDL code) |
| INPUT_DATA_RATE | Yes | ✓ | ✓ | ✓ | ✓ | ✓ | [Integer] Specifies the data rate into the PLL. The multiplication value for the PLL is INPUT_DATA_RATE / INCLOCK_PERIOD. |
| INCLOCK_DATA_ALIGNMENT *(1)* | Yes | ✓ | ✓ | ✓ | ✓ | ✓ | [String] Specifies the alignment of input data with respect to the rising edge of the rx_inclock port. Values are "EDGE_ALIGNED", "45_DEGREES", "90_DEGREES", "135_DEGREES", "CENTER_ALIGNED", "235_DEGREES", "270_DEGREES", and "315_DEGREES". If omitted, the default is "EDGE_ALIGNED". |

*Table 3–6. Receiver Parameters (Part 3 of 7)*

| Parameter Name | Required | Stratix | Stratix GX | Stratix II, Stratix II GX, and Stratix III | Cyclone | Cyclone II and Cyclone III | Description |
|---|---|---|---|---|---|---|---|
| ENABLE_DPA_FIFO *(2)* | No | — | ✓ | — | — | — | [String] Values are "ON" or "OFF". Indicates whether the DPA FIFO buffer is enabled for this channel. If omitted, the default is "ON". |
| USE_CORECLOCK_INPUT *(2)* | No | — | ✓ | — | — | — | [String] Values are "ON" or "OFF". Indicates whether the rx_coreclk port or the clock from PLL is used as the non-peripheral clock. The rx_coreclk port must be connected if you turn on this parameter. If omitted, the default is "OFF". |
| RESET_FIFO_AT_FIRST_LOCK | No | — | — | ✓ | — | — | [String] Values are "ON" to automatically reset the bit serial FIFO when rx_dpa_locked rises for the first time or "OFF" to explicitly reset the FIFO through rx_reset. Resets the FIFO between the DPA and the deserializer. |
| LOSE_LOCK_ON_ONE_CHANGE | No | — | — | ✓ | — | — | [String] Values are "ON" if phase alignment circuitry switches to a new phase or "OFF" if there are two phase changes in the same direction. Indicates when the DPA circuit has lost phase lock. |
| REGISTERED_OUTPUT | No | ✓ | ✓ | ✓ | ✓ | ✓ | [String] Values are "ON" or "OFF". Indicates whether the rx_out[] port should be registered. |
| REGISTERED_DATA_ALIGN_INPUT *(1)* | No | ✓ | ✓ | — | — | — | [String] Values are "ON" or "OFF". Specifies whether the rx_data_align port is registered, which controls the enabling of the synchronization register in the receiver. |

*Table 3–6. Receiver Parameters (Part 4 of 7)*

| Parameter Name | Required | Stratix | Stratix GX | Stratix II, Stratix II GX, and Stratix III | Cyclone | Cyclone II and Cyclone III | Description |
|---|---|---|---|---|---|---|---|
| DATA_ALIGN_ROLLOVER | No | — | — | ✓ | — | — | [Integer] Values are 1 through 11, which indicate the number of pulses it takes for the data alignment circuitry to restore the serial data latency back to 0. This parameter is only available if the rx_channel_data_align input port is used. |
| INCLOCK_BOOST | No | — | — | — | — | — | [Integer] The effective clock period used to sample output data. Values are 1, 2, and 4-10. This parameter is available for APEX II and Mercury devices only. |
| CDS_MODE | No | — | — | — | — | — | [String] Values are SINGLE_BIT, MULTIPLE_BIT, and UNUSED. If omitted, the default is UNUSED. When this parameter is set to SINGLE_BIT, the deskew circuitry will expect a bit pattern like 00001111, which can correct for clock skew up to 50% of one clock edge. When this parameter is set to MULTIPLE_BIT, the deskew circuitry will expect a pattern like 01010101, which can correct for any clock skew but requires user circuitry to align data. This parameter is available for APEX II devices only. |

*Table 3–6. Receiver Parameters (Part 5 of 7)*

| Parameter Name | Required | Stratix | Stratix GX | Stratix II, Stratix II GX, and Stratix III | Cyclone | Cyclone II and Cyclone III | Description |
|---|---|---|---|---|---|---|---|
| IMPLEMENT_IN_LES | No | ✓ | ✓ | ✓ | — | — | [String] Specifies whether to implement serializer/deserializer circuitry in logic cells, which allows the circuitry to behave similar to Stratix LVDS circuitry. The IMPLEMENT_IN_LES parameter should be used for SERDES functions that require data rates that are lower than the dedicated circuitry. Values are "ON" and "OFF". This parameter is available for Stratix III, Stratix II, Stratix, and Stratix GX devices only. |

*Table 3–6. Receiver Parameters (Part 6 of 7)*

| Parameter Name | Required | Stratix | Stratix GX | Stratix II, Stratix II GX, and Stratix III | Cyclone | Cyclone II and Cyclone III | Description |
|---|---|---|---|---|---|---|---|
| BUFFER_IMPLEMENTATION | No | ✓ | ✓ | ✓ | ✓ | ✓ | [String] Specifies where to implement the buffer.<br>BUFFER IMPLEMENTATION applies only to Cyclone III, Cyclone II, and Cyclone, and Stratix III, Stratix II, and Stratix (LE implementations only for Stratix series) for Odd deserialization factors.<br><br>To use the BUFFER_IMPLEMENTATION parameter, the IMPLEMENT_IN_LES parameter must be turned on. Also, the BUFFER_IMPLEMENTATION parameter can be used with deserialization factors of 5, 7, or 9 only.<br><br>Values are:<br><br>**Parameter Value:** MUX<br>**Description:** Uses mux implementation instead of buffer implementation.<br><br>**Parameter Value:** RAM<br>**Description:** Buffer is implemented in RAM blocks.<br><br>**Parameter Value:**<br>LES<br><br>**Description:**<br>Buffer is implemented in logic elements.<br><br>If omitted, the default is “RAM”. |

*Table 3–6. Receiver Parameters (Part 7 of 7)*

| Parameter Name | Required | Stratix | Stratix GX | Stratix II, Stratix II GX, and Stratix III | Cyclone | Cyclone II and Cyclone III | Description |
|---|---|---|---|---|---|---|---|
| PORT_RX_DATA_ALIGN | No | ✓ | ✓ | — | — | — | [String] Determines if the rx_data_align port is used or unused. Values are:<br><br>**Parameter Value:** PORT_USED<br>**Description :** Value assumes the rx_data_align port is used.<br><br>**Parameter Value:** PORT_UNUSED<br>**Description:** Value assumes the rx_data_align port is used.<br><br>**Parameter Value:** PORT_CONNECTIVITY<br>**Description:** Value checks the connectivity of the rx_data_align port to determine usage.<br><br>If omitted, the default is "PORT_CONNECTIVITY". |
| LPM_HINT | No | ✓ | ✓ | ✓ | ✓ | ✓ | [String] Allows you to specify Altera-specific parameters in VHDL Design Files (.vhd). The default is "UNUSED". |

*Notes to Table 3–6:*
(1) This parameter is not available on Stratix III, Stratix II, and Stratix GX devices when using DPA mode.
(2) This parameter is only available on Stratix III, Stratix II, and Stratix GX devices when using DPA mode.

*Figure 3–1. OUTCLOCK_ALIGNMENT*

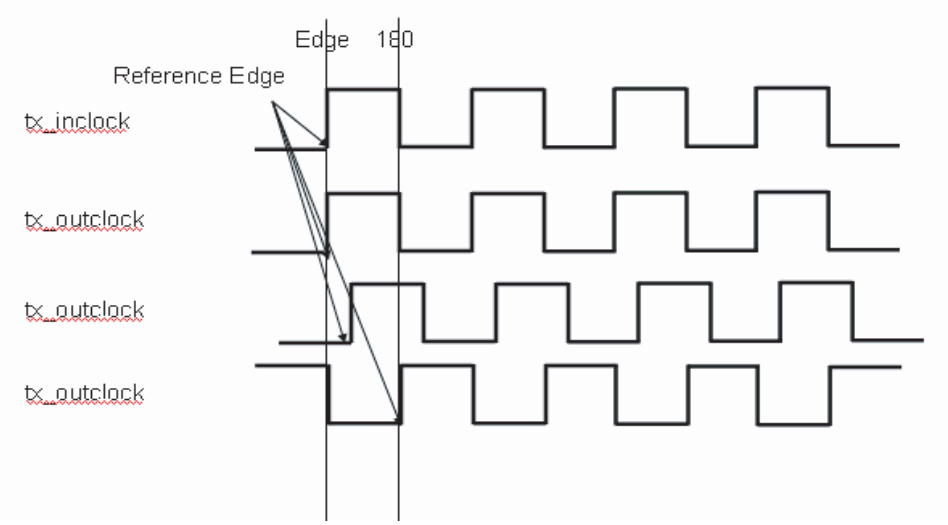



*Figure 3–2. INCLOCK_DATA_ALIGNMENT*