3. Stratix II GX Dynamic Reconfiguration

SIIGX52007-1.1

Introduction

The Stratix® II GX gigabit transceiver block gives you a simplified means to dynamically reconfigure:

- Transmit and receive analog settings
- Transmit data rate in the multiples of 1, 2, and 4
- One channel at a time
- Channel and clock multiplier unit (CMU) PLL
- CMU PLL only

Typically, to achieve the intended bit error rate (BER) for a system, you will take advantage of the multiple analog settings provided in the Stratix II GX device. Being able to change the analog settings is a powerful tool that you can use during link and system debug.

The following analog settings can be dynamically changed:

- Pre-emphasis settings
- Equalization settings
- DC gain settings
- Voltage output differential (V_{OD}) settings

In addition to allowing you to change the equalization settings during runtime, the dynamic reconfiguration controller provides an option to dynamically control the adaptive equalization (AEQ) hardware present in each of the transceiver channels. The AEQ hardware continuously tunes the receiver equalization settings based on the frequency content of the incoming signal.

The dynamic data rate switch feature on the transmitter is enabled through a PLD signal. Depending on the setting of this signal, the transmitter data rate can be divided in steps of 1, 2, or 4 per channel.

Another important feature is the ability to dynamically reconfigure from one mode to another mode. This mode reconfiguration may involve reconfiguring the transceiver data path or data rate or both. You can reconfigure the transceiver data rate either by switching to the other CMU PLL or by dynamically reconfiguring the CMU PLL. The former is enabled in the Quartus® II software version 6.1 and later, while the latter is enabled in the Quartus II software version 7.1 and later.

The dynamic reconfiguration feature facilitates mode transitions involving:

- Protocol functional mode (×1 only) to and from Basic functional mode
- Protocol functional mode (×1 only) to Protocol functional mode (×1 only)
- One Basic functional mode to other Basic functional modes

This is a very useful and powerful feature for transceiver system applications because it enables channels in a system to adapt to multiple serial data rates and system protocols.

Table 3–1 shows dynamic reconfiguration features supported in various Quartus II software versions.

Table 3–1. Software Support for Dynamic Reconfiguration

Version	Transmitter and Receiver Analog Settings (PMA Controls)	Transmitter Data Rate Switch (×1, ×2, ×4)	Channel Reconfiguration	Channel and CMU PLL Reconfiguration	CMU PLL-Only Reconfiguration
Quartus II 6.0	✓	—	—	—	—
Quartus II 6.1	✓	✓	✓	—	—
Quartus II 7.1	✓	✓	✓	✓	✓

Dynamic Reconfiguration Controller Architecture

The Stratix II GX device offers a simplified dynamic reconfiguration controller in the Quartus II ALT2GXB_RECONFIG module to control the configurable settings of the transceiver. The dynamic reconfiguration controller is a soft IP which utilizes Stratix II GX device PLD resources. It is optimized for minimal PLD resource usage. Only one controller is allowed per transceiver block. The dynamic reconfiguration controller does not have the capability to control multiple Stratix II GX devices or any off-chip interface.

The dynamic reconfiguration capability is only intended for Stratix II GX devices, having no backward compatibility to Stratix GX devices.

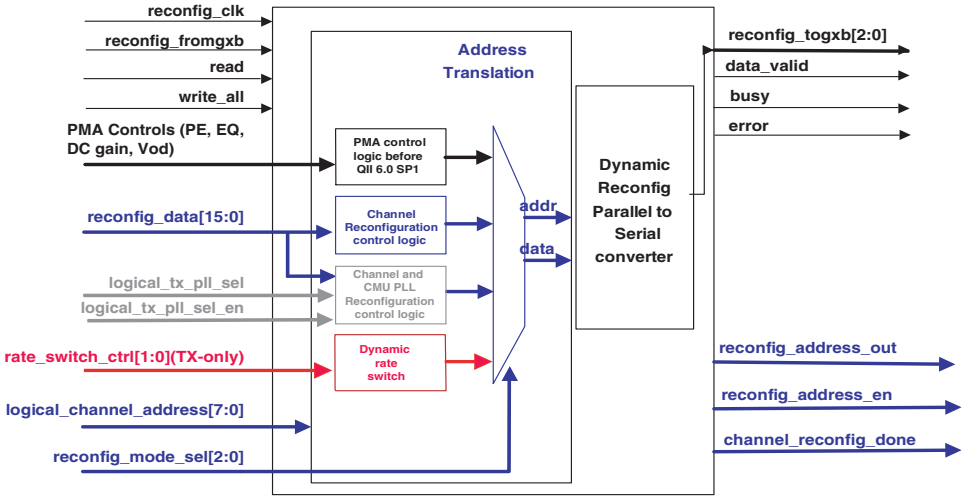
Stratix II GX dynamic reconfiguration is very flexible because of the following features:

- Two transmit PLLs enabled—This allows you to achieve multiple data rates and protocols in a single transceiver block.

- Basic double-width modes—The minimum data rate is lowered to 1 Gbps. This helps if you want to only switch data rates without changing the data path width.
- More optional features in Basic mode.
- PLD interface clocking of the transceiver is enhanced by introducing "Core Clocking Options". These core clocking options help you optimize clock resource usage and allows you to set up the proper PLD interface clocking on transmit and receive paths.

Figure 3–1 shows a conceptual view of these features.

Figure 3–1. Block Diagram of the Dynamic Reconfiguration Controller (ALT2GXB_RECONFIG)



The following items are not supported as part of the dynamic reconfiguration feature:

- Mode switch to and from any ×4 and ×8 configurations
- Not backward compatible with Stratix GX devices
- To and from PCI Express (PIPE) mode with NFRI IP
- Testability features (pseudo-random binary sequence [PRBS] and built-in self test [BIST])

Dynamic Reconfiguration Setup in the MegaWizard Plug-In Manager

The optional dynamic reconfiguration interface must be enabled through the MegaWizard® Plug-In Manager (dynamic reconfiguration is turned OFF by default).

The dynamic reconfiguration interface has the following signals:

- `reconfig_togxb[2:0]` as an input signal bus
- `reconfig_fromgxb` as an output signal from ALT2GXB instance. `reconfig_fromgxb` is a transceiver block-based signal; for example, if the number of the channels selected in ALT2GXB are:
  - 0 < Channels < 4, then signal `reconfig_fromgxb` = 1 bit
  - 4 < Channels < 8, then signal `reconfig_fromgxb` = 2 bits
  - 8 < Channels < 12, then signal `reconfig_fromgxb` = 3 bits
  - 12 < Channels < 16, then signal `reconfig_fromgxb` = 4 bits
  - 16 < Channels < 20, then signal `reconfig_fromgxb` = 5 bits

After the dynamic reconfiguration option is enabled in the ALT2GXB MegaWizard, you must set one more setting—the **What is the dynamic reconfig starting channel number?** option. The dynamic reconfiguration starting channel number setting range is from 0 - 156 in multiples of 4 (because the dynamic reconfiguration interface is per transceiver block). This range of 0 - 156 is the logical channel address based purely on the number of possible ALT2GXB instances.

To better understand how logical addressing works, consider the scenario of 20 separate transmit and receive instances of the ALT2GXB megafunction in a design and how to set the address of the starting channel of each instance.

The first instance of a transmit and receive channel has the starting channel number setting of **0**. The second instance of a transmit and receive channel has the starting channel number setting of **4**. And so on. The twentieth instance of the same configuration has the starting channel number of **76**.

Extending the same logic to the maximum possible instances case of 20 transmit-only and 20 receive-only configurations, targeted for a five transceiver block Stratix II GX device, the maximum starting channel number of the dynamic reconfiguration option is **156** (40 instances * 4).

Configure the ALT2GXB_RECONFIG and the ALT2GXB modules, depending on the number of transceiver channels that are controlled by the dynamic reconfig controller (ALT2GXB_RECONFIG). Use the logical channel views with the above mentioned logical addressing in the

ALT2GXB instance. The Quartus II fitter errors out if the dynamic reconfiguration option is enabled in the ALT2GXB megafunction, but the `reconfig_fromgxb` and `reconfig_togxb` ports are NOT connected to the ALT2GXB_RECONFIG instance.

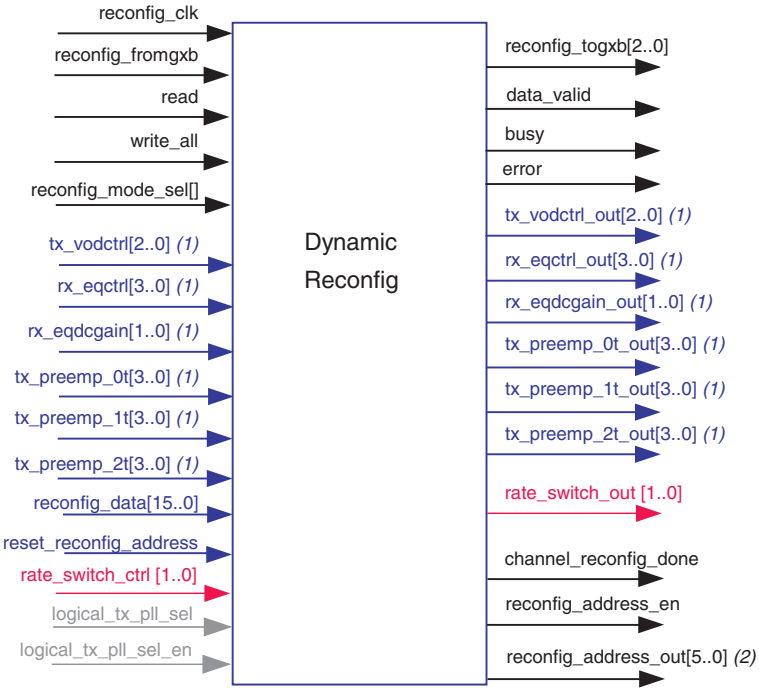
The megafunction and pre-fitter automatically map the logical channel into the physical placements. This physical placement includes merging (automatically done by the Quartus II software). The software performs merging (packing channels into the same transceiver block) only when multiple channels of the same data rate and data path configuration are controlled by one dynamic reconfiguration (ALT2GXB_RECONFIG) controller instance. Channels connected to multiple ALT2GXB_RECONFIG controllers will not be merged.

Dynamic Reconfiguration Controller Interface

The dynamic reconfiguration controller supports write and read transactions. Figure 3–2 shows the dynamic reconfiguration interface list. The following transactions are allowed, based on the dynamic reconfiguration features:

- Analog Settings Reconfiguration—Write and Read (read is optional)
- Channel Reconfiguration—Write Transaction Only
- Dynamic Transmit Rate Switch—Write and Read (read is optional)
- Channel and CMU PLL Reconfiguration
- CMU PLL-Only Reconfiguration

Figure 3–2. Dynamic Reconfiguration Interface



Notes to Figure 3–2:

(1) Optional control and status signals. At least one control signal must be enabled if only analog settings reconfiguration is enabled.

(2) If the channel reconfiguration feature is selected in the ALT2GXB_RECONFIG MegaWizard, the `reconfig_address_out` is 5-bits wide [4..0]. If the Channel and TXPLL select/reconfig feature is selected, the `reconfig_address_out` is 6-bits wide [5..0].

The `reconfig_mode_sel` signal determines the reconfiguration mode. This control signal is 3-bits wide if the **Adaptive Equalization control** option is **not** selected. If this option is selected, the `reconfig_mode_sel` signal is 4-bits wide. Encoding of the `reconfig_mode_sel` signal (when the **Adaptive Equalization control** option is **not** selected) is as follows:

- `reconfig_mode_sel [2:0]`:
  - 000 – Reconfiguration of Analog controls. The Analog controls feature has been enabled in the Quartus II software version 6.0 and later
  - 001 – Channel Reconfiguration
  - 011 – Dynamic Transmit rate switch
  - 100, 101, 110 – Channel and CMU PLL Reconfiguration

Refer to "Channel and PMA Controls Reconfiguration" on page 3–20 and "Channel and Clock Multiplier Unit (CMU) PLL Reconfiguration" on page 3–87 for detailed `reconfig_mode_sel[2:0]` signal encoding.

As described in "Stratix II GX ALT2GXB Megafunction User Guide" on page 4–1, the signals `reconfig_togxb[2:0]` and `reconfig_fromgxb` are the interface signals between the ALT2GXB instance and the ALT2GXB_RECONFIG instance. The dynamic reconfiguration controller runs at a frequency determined by the clock `reconfig_clk` signal. The supported frequency range of the `reconfig_clk` is 2.5 MHz – 50 MHz.

Altera recommends the `reconfig_clk` signal be driven on a global clock resource.

You must set the following two settings in the ALT2GXB_RECONFIG MegaWizard:

1. What is the number of channels controlled by the controller?

You must provide the number of channels for the megafunction, depending on the design setup supported. There are two ways of using dynamic reconfiguration controllers. They are:

- Single Dynamic Reconfiguration Controller—one controller controlling all the instances of the ALT2GXB in a device. When multiple instances of the ALT2GXB megafunction are controlled by a single ALT2GXB_RECONFIG controller, the following rules should be followed for setting the "**What is the number of channels controlled by the controller**?" option:
  - Each instance of the megafunction must have a set of the consecutive channel numbers beginning with a unique number that is a multiple of four.
  - The number of channels controlled is the last channel number.

- Multiple Dynamic Reconfiguration Controllers—for multiple instances of the ALT2GXB, it is not possible to have two dynamic reconfiguration controllers controlling the same ALT2GXB instance. One controller is allowed to control multiple ALT2GXB instances or every channel will have its own dynamic reconfiguration controller. If every channel has its own dynamic reconfiguration controller, there may be problems with fitting.
  - The Quartus II software cannot merge multiple transceiver channel instances into a transceiver block if multiple dynamic reconfiguration controllers are used, even if the channels are configured to the same protocol functional mode and data rate.

For example, ALT2GXB instance1 has five channels of the same data rate and functional mode; ALT2GXB instance2 has three channels of same data rate and functional mode. Both ALT2GXB instances have separate dynamic reconfiguration controllers controlling them. These two ALT2GXB instances (a total of eight channels) cannot be merged into two transceiver blocks. These two instances can be merged only if they are controlled by one dynamic reconfiguration controller. This merging will not change the behavior of the silicon compared to functional simulations.

2. **Use the same control signals for all channels**.

Check this option when you know that the same analog control signals are used for all the channels in the design. By checking this option, the Quartus II software uses one set of analog signals to control all channels used in all transceiver blocks that are controlled by this reconfiguration controller.

Table 3–2 describes the ports for the dynamic reconfiguration controller.

Table 3–2. Port List of the Dynamic Reconfiguration Controller (ALT2GXB_RECONFIG) (Part 1 of 6)

Port Name	Input/Output	Description
reconfig_clk	Input	Input reference clock for the dynamic reconfiguration controller. The frequency range of this clock is 2.5 MHz to 50 MHz. The assigned clock uses global resources by default. This same clock should be connected to ALT2GXB.
ALT2GXB - ALT2GXB_RECONFIG Interface Signals		
reconfig_fromgxb	Input	Interface bus signal from ALT2GXB to ALT2GXB_RECONFIG instance. The width of the signal in ALT2GXB_RECONFIG is determined by the number of channels controlled by the controller.
reconfig_togxb[2..0]	Output	Fixed bus interface between ALT2GXB_RECONFIG and ALT2GXB. This signal is independent of the number of channels.
PLD Interface Signals		
write_all	Input	Control signal to initiate a write transaction. This signal is active high. When the analog settings (V_{OD}, equalization, etc.) are reconfigured, the reconfiguration controller writes to all the transceiver channels connected to the controller.
busy	Output	Status signal to indicate that the reconfiguration controller has not completed the read or write transaction.

Table 3–2. Port List of the Dynamic Reconfiguration Controller (ALT2GXB_RECONFIG) (Part 2 of 6)

Port Name	Input/Output	Description
read	Input	Control signal to initiate a read transaction. This signal is active high. When the analog settings (V_{OD}, equalization, etc.) are read, the reconfiguration controller reads the analog setting values from all the transceiver channels connected to the controller. When you select this signal, at least one of the output control ports (for example, tx_vodctrl_out) should be selected. Otherwise, when you initiate a read transaction, the reconfiguration controller may get into a deadlock state (since it cannot send data to any output).
data_valid	Output	Status signal for the read transaction. If data_valid is high, the read back data is valid. That is, the current data on the output control signals after data_valid is asserted high is the valid data read out. This signal is only enabled when at least one read control port is enabled. When a read control port is enabled and a write transaction is finished, the data_valid signal goes high and the busy signal goes low.
error	output	Optional status signal to indicate that an unsupported operation is attempted. The error port can be enabled by selecting the options in the **Error checks/data rate switch** tab. The dynamic reconfiguration controller de-asserts the busy signal and asserts the error signal for two reconfig_clk cycles when you attempt an unsupported operation.

Table 3–2. Port List of the Dynamic Reconfiguration Controller (ALT2GXB_RECONFIG) (Part 3 of 6)

Port Name	Input/Output	Description
Analog Settings Control/Status Signals		
tx_vodctrl	Input	Optional transmit buffer voltage output differential (V_{OD}) control signal. It is 3-bits per channel. The number of settings varies based on the transmit buffer supply setting and the termination resistor setting in ALT2GXB instance. The following shows the V_{OD} values corresponding to the tx_vodctrl settings for 100-Ω termination. For V_{OD} values corresponding to other termination settings, refer to Table 2–8. **tx_vodctrl** — **V_{OD} (mV) for 1.5V V_{CCH}** — **V_{OD} (mV) for 1.2V V_{CCH}** 000 — N/A — N/A 001 — 400 — 320 010 — 600 — 480 011 — 800 — 640 100 — 1000 — 800 101 — 1200 — 960 110 — 1400 — N/A 111 — N/A — N/A
tx_preemp_0t *(1)*	Input	Optional pre-emphasis control for pre-tap for the transmit buffer. It is 4-bits per channel. This signal controls both pre-emphasis positive and its inversion. 0 represents 0 1–7 represents -7 to -1 9–15 represents 1 to 7 8 maps to 0
tx_preemp_1t *(1)*	Input	Optional pre-emphasis control for first post tap for the transmit buffer. It is 4-bits per channel.
tx_preemp_2t *(1)*	Input	Optional pre-emphasis control for second post-tap for the transmit buffer. It is 4-bits per channel. This signal controls both pre-emphasis positive and its inversion. 0 represents 0 1–7 represents -7 to -1 9–15 represents 1 to 7 8 maps to 0
rx_eqctrl	Input	Optional equalization control signal on the receive side of the PMA. It is a 4-bit bus per each channel.

Table 3–2. Port List of the Dynamic Reconfiguration Controller (ALT2GXB_RECONFIG) (Part 4 of 6)

Port Name	Input/Output	Description
rx_eqdcgain *(2)*	Input	Optional equalizer DC gain control. It supports three legal settings and is 2-bits wide per channel. 00 corresponds to 0 dB 01 and 10 correspond to 3 dB 11 corresponds to 6 dB
tx_vodctrl_out	Output	Optional transmit V_{OD} output signal. This signal reads out the value written into the V_{OD} control register. The signal width of this output signal is the same as its corresponding input signal.
tx_preemp_0t_out	Output	Optional pre-tap, pre-emphasis output signal. This signal reads out the value written by its input control signal. The signal width of this output signal is the same as its corresponding input control signal.
tx_preemp_1t_out	Output	Optional first post-tap, pre-emphasis output signal. This signal reads out the value written by its input control signal. The signal width of this output signal is the same as its corresponding input control signal.
tx_preemp_2t_out	Output	Optional second post-tap pre-emphasis output signal. This signal reads out the value written by its input control signal. The signal width of this output signal is the same as its corresponding input control signal.
rx_eqctrl_out	Output	Output signal to read the setting of equalization setting of the ALT2GXB instance. The signal width of this output signal is the same as its corresponding input signal.
rx_eqdcgain_out	Output	Equalizer DC gain output signal. This signal reads out the settings of the ALT2GXB instance DC gain. The signal width of this output signal is the same as its corresponding input signal.
Channel Reconfiguration Signals		
reset_reconfig_address	Input	Synchronous reset signal to the ALT2GXB_RECONFIG to reset the reconfig_address_out port to **0**. Use this signal when you want to restart the reconfiguration of a channel by initiating writing the memory initialization file (MIF) word 0.
reconfig_data[15:0]	Input	Sixteen bits input data word. You input it from the location that has the MIF to reconfigure the registers. This input port is only used in the Channel Reconfiguration or Channel and CMU PLL Reconfiguration feature (discussed in "Channel and Clock Multiplier Unit (CMU) PLL Reconfiguration" on page 3–87).

Table 3–2. Port List of the Dynamic Reconfiguration Controller (ALT2GXB_RECONFIG) (Part 5 of 6)

Port Name	Input/Output	Description
reconfig_mode_sel[2:0]	Input	Select the reconfiguration mode for the ALT2GXB_RECONFIG megafunction. The signal encoding is as follows: 000 - Reconfiguration for analog controls. This feature has been enabled in the Quartus II software version 6.0 and later versions. 001 - Channel Reconfiguration 010 - Not supported (do not attempt to read or write with this value) 011 - Dynamic Transmit data rate switch *100 - TXPLL *101 - Channel and TXPLL reconfiguration *110 - Channel reconfiguration with TXPLL select 111 - Not supported (do not attempt to read or write with this value) *The features corresponding to these values are discussed in "Channel and Clock Multiplier Unit (CMU) PLL Reconfiguration" on page 3–87.
logical_channel_address [CHANNEL_ADDRESS_WIDTH-1:0]	Input	Specify the logical channel address for the channel that needs to be reconfigured. The CHANNEL_ADDRESS_WIDTH parameter is determined through the NUMBER_OF_CHANNELS parameter.
reconfig_address_out[5:0]	Output	This signal indicates the address out and that the address read out is the current address to be reconfigured by the ALT2GXB_RECONFIG megafunction during channel reconfiguration. This signal is 5-bit wide in channel reconfiguration mode and 6-bit wide in channel and CMU PLL reconfiguration mode.
reconfig_address_en	Output	This port indicates the current address to be reconfigured for the ALT2GXB_RECONFIG megafunction had already changed during channel reconfiguration.
channel_reconfig_done	Output	This port indicates that the ALT2GXB_RECONFIG megafunction has finished writing all the words of a MIF. This is only applicable for channel reconfiguration mode.
Dynamic Transmit Rate Switch Signals		
rate_switch_ctrl[1:0]	Input	This input is the control signal to write the desired division factors on a per-channel transmitter basis. This port is only applicable when reconf_mode_sel is set to **011**. The output value is listed below: 00 - Divide by 1 01 - Divide by 2 10 - Divide by 4 11 - Not supported (do not attempt to read or write with this value)

Table 3–2. Port List of the Dynamic Reconfiguration Controller (ALT2GXB_RECONFIG) (Part 6 of 6)

Port Name	Input/Output	Description
rate_switch_out[1:0]	Output	This signal reads out the value that has written in for the rate switch of specified transmitter outputs. This output port is only applicable when reconf_mode_sel is set to **011**. The output value is listed below: 00 - Divide by 1 01 - Divide by 2 10 - Divide by 4
Channel and CMU PLL Reconfiguration		
logical_tx_pll_sel	Input	This control signal allows you to select the CMU PLL that you wish to reconfigure. It also allows you to select the CMU PLL to which the channel is listening in Channel Reconfiguration with TX PLL Select mode. Refer to "Logical TX PLL Select" on page 3–105 for more information.
logical_tx_pll_sel_en	Input	This signal validates the logical_tx_pll_sel signal. Refer to "Logical TX PLL Select" on page 3–105 for more information.

Notes to Table 3–2:
(1) Not all combinations of bits are legal values.
(2) In PIPE mode, this input should be tied to **01** to be PCI E-compliant.

Dynamic Configuration Controller (ALT2GXB_RECONFIG), ALT2GXB Design Examples

The following design examples illustrate the various possible topologies of the dynamic reconfiguration controller with ALT2GXB instances. The first two design examples specifically discuss a single controller controlling multiple instances of an ALT2GXB and a single controller controlling one instance of an ALT2GXB. Design example three discusses the HDL construct needs if you are stamping the ALT2GXB instances. Each instance of an ALT2GXB in turn can have more than one transceiver channel. Also, in all the design examples, it is assumed that only the Analog (PMA) settings reconfiguration is enabled, to simplify the illustration. In the real system, you can enable other supported features along with the analog setting reconfiguration.

Example 1

Consider a design with two instances of an ALT2XGB configuration, Instance1 with five transceiver channels and Instance2 with three transceiver channels.

Assume the following for this example:

- Instance1 and Instance2 cannot be merged due to their configurations.
- One dynamic reconfiguration controller controls all eight channels.
- Only the transmit V_{OD} and receiver equalization controls are enabled.

The following are the typical steps that help setup the configuration:

Five Channel Transceiver Instance:

- In the ALT2GXB MegaWizard, set the **What is the number of channels?** option to **5** along with other options in the ALT2GXB MegaWizard.
- Enable the **Analog controls** option under the dynamic reconfiguration settings (to dynamically change equalization values, also enable the **Enable equalizer settings** option).
- The output signal `reconfig_fromgxb` is transceiver-block based, so the number of bits for this instance is two since the number of channels is five. The input signal `reconfig_togxb` is a fixed bus width of three bits.
- Set the **What is the starting channel number?** option to **0**.

Three Channel Transceiver Instance:

- In the ALT2GXB MegaWizard, set the **What is the number of channels?** option to **3**.
- Enable the **Analog controls** option under the dynamic reconfiguration settings (to dynamically change equalization values, also enable the **Enable equalizer settings** option).
- The output signal `reconfig_fromgxb` is transceiver block based, so the number of bits for this instance is one since the number of channels is three. The input signal `reconfig_togxb` is a fixed bus width of three bits.
- Set the **What is the starting channel number?** option to **8**. This address of eight is warranted because the previous ALT2GXB instance has five channels which logically fits into two transceiver blocks (transceiver blocks with starting channel numbers 0 and 4), and since this case has multiple instances of the ALT2GXB controlled by one dynamic controller, the numbering is consecutive channel numbers in multiples of four.

ALT2GXB_RECONFIG Setup for PMA Controls Reconfiguration:

- Launch the ALT2GXB_RECONFIG MegaWizard.
- Set the **What is the number of channels controlled by the controller?** option to **12**. The setting for this option has a number that is more than the total number of channels needed to be controlled

(eight channels) by dynamic reconfiguration. This is needed because based on this setting, the Quartus II software chooses the bus width of the signal `reconfig_fromgxb` in addition to the width of the analog control signals. In this case, the design needs 3-bits wide signals so the controller can control a total of three transceiver blocks (five channels in two transceiver blocks and three channels into one transceiver block).

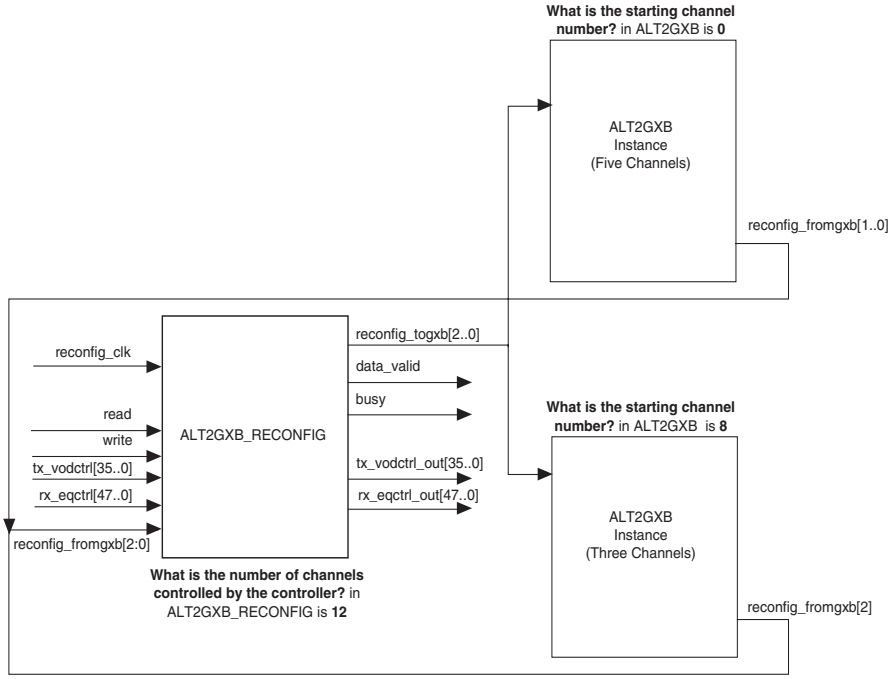
To make it simple, choose the channel number based on a rounded-up channel number to the nearest transceiver block multiple. In this case, it is eight channels required and since no merging is allowed, eight channels require three transceiver blocks. The three transceiver blocks round up to a transceiver block multiple channel number of 12 (3 × 4 = 12). Refer to the "**1. What is the number of channels controlled by the controller?**" option in "Dynamic Reconfiguration Controller Interface" on page 3–5 for more information about this setting.

- Select the necessary analog control signals to write in and read out for V_{OD} and equalization from all the options available in the MegaWizard. Also note the analog control signal widths are for 12 channels since the above channel setting is 12. Control signals for unused channels 5 to 7 and channel 11 can be tied to logic low (zero/ground). In this design scenario, the V_{OD} signal (`tx_vodctrl`) width is 36 bits (12 channel × `tx_vodctrl[2:0]` = `tx_vodctrl[35:0]`). Tie `tx_vodctrl[35:33]` and `tx_vodctrl[23:15]` to ground. Use similar methods for the equalization setting.

ALT2GXB Instances and ALT2GXB_RECONFIG Instance Connections:

- Connect the `reconfig_fromgxb` signal from the ALT2GXB instance to the same signal in the ALT2GXB_RECONFIG instance. The lowest starting channel number transceiver block is connected to the lowest significant bit and so on. In this case, the configuration instance with five channels of the ALT2GXB instance has a starting channel of zero, which has the signal `reconfig_fromgxb[1:0]` which should be connected to `reconfig_fromgxb[1:0]` of the ALT2GXB_RECONFIG instance. The other three channel instances of ALT2GXB, with a starting channel of eight, has the signal `reconfig_fromgxb` which should be connected to ALT2GXB_RECONFIG `reconfig_fromgxb[2]`. Refer to Figure 3–3 for more information.
- Connect the `reconfig_togxb` signal from the ALT2GXB_RECONFIG instance to the same signal on the ALT2GXB instance.

Figure 3–3. ALT2GXB Modules with One ALT2GXB_RECONFIG Module



Example 2

This design example has two instances of distinct configurations: Instance1 with five transceiver channels and Instance2 with three channels. This configuration requires separate dynamic reconfiguration controllers for the two instances. This scenario covers the case of multiple dynamic reconfiguration controllers controlling multiple instances of the ALT2GXB. Assume that the analog settings (transmit V_{OD} and receive equalization controls) for both instances are enabled. The following are the typical steps to setup the configuration:

Five Channel Transceiver Instance1:

- In the ALT2GXB MegaWizard, set the **What is the number of channels?** option to **5** along with other options in the ALT2GXB MegaWizard.
- Enable the **Analog controls** option under the dynamic reconfiguration settings (to dynamically change equalization values, also enable the **Enable equalizer settings** option).

- The output signal `reconfig_fromgxb` is transceiver-block based so the number of bits for this instance is two since the number of channels is five. The input signal `reconfig_togxb` is a fixed width of three bits.
- Set the **What is the starting channel number?** option to **0**.

Dynamic Reconfiguration Controller Instance1:

- Launch the ALT2GXB_RECONFIG MegaWizard.
- Set the **What is the number of channels controlled by the controller?** option to **5**. This option helps the Quartus II software choose the bus width of the signal `reconfig_fromgxb` in addition to the width of the analog control signals. In this case, the design needs 2-bits wide signals so the controller can control a total of two transceiver blocks (five channels in two transceiver blocks). Refer to "Introduction" on page 3–1 for more information about this setting.
- Select the necessary analog control signals to write in and read out from the V_{OD}, pre-emphasis, equalization, and DC gain options for this setup.

Three Channel Transceiver Instance2:

- Set the **What is the number of channels?** option to **3**.
- Enable the **Analog controls** option under the dynamic reconfiguration settings (to dynamically change equalization values, also enable the **Enable equalizer settings** option).
- The output signal `reconfig_fromgxb` is transceiver block based so the number of bits for this instance is one since the number of channels is three. The input signal `reconfig_togxb` is a fixed width of three bits.
- Set the **What is the starting channel number?** option to **0**. This address number of 0 is the same as the previous five channel ALT2GXB instance setting. You do not need to have a consecutive channel starting number (multiples of four) since these two ALT2GXB instances are controlled by different dynamic reconfiguration controllers.

Dynamic Reconfiguration Controller Instance2:

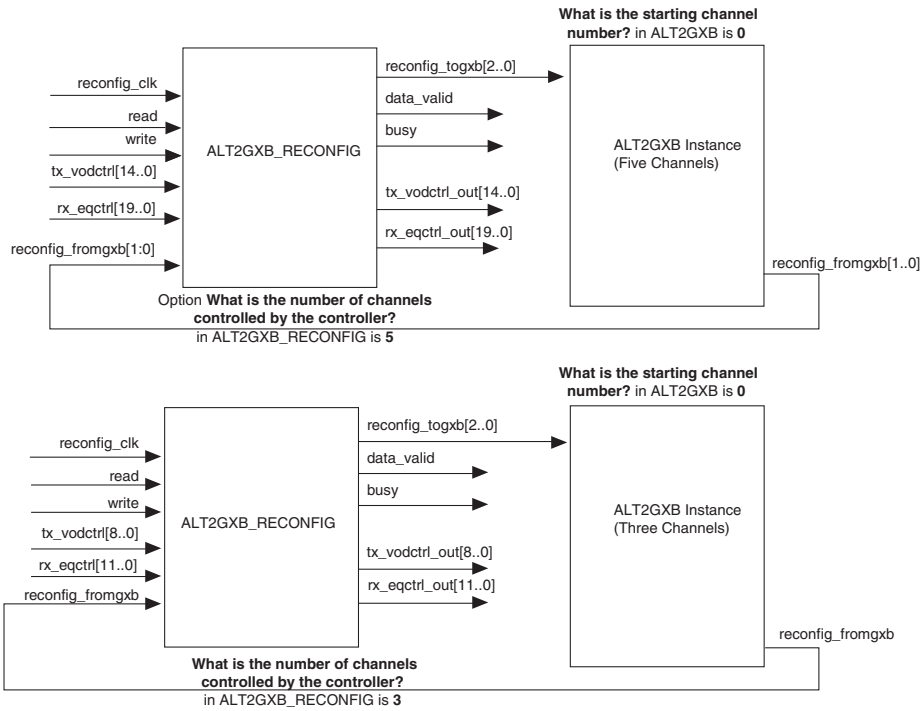
- Launch the ALT2GXB_RECONFIG MegaWizard.
- Set the **What is the number of channels controlled by the controller?** option to **3**. This option helps the Quartus II software choose the bus width of the signal `reconfig_fromgxb`, in addition to the width of the analog control signals. In this case, the design needs a 1-bit wide signal so the controller can control a total of one transceiver block (three channels into one transceiver block) and have the option set to at least three so that the Quartus II software enables three channels of the analog control signals in the options sections.

- Select the necessary analog control signals to write in and read out from the V_{OD}, pre-emphasis, equalization, and DC gain options.

ALT2GXB Instances and ALT2GXB_RECONFIG Instance Connections:

- Connect the `reconfig_fromgxb` signal from the ALT2GXB instance to the same signal of the corresponding ALT2GXB_RECONFIG instance. Refer to Figure 3–4 for more information.
- Connect the `reconfig_togxb` signal from the ALT2GXB_RECONFIG instance to the same signal of the corresponding ALT2GXB instance.

Figure 3–4. ALT2GXB_RECONFIG Modules with Two ALT2GXB Modules



Example 3

This design example consists of five channels of transceivers with the same data rate and functional mode. This configuration has one dynamic reconfiguration controller to control five channels. This scenario covers the case stamping five instantiations of one channel ALT2GXB instance configuration.

One Channel ALT2GXB Configuration:

- Set the **What is the number of channels?** option to **1** along with other options in the ALT2GXB MegaWizard.
- Enable the **Analog controls** option under the dynamic reconfiguration settings (to dynamically change equalization values, also enable the **Enable equalizer settings** option).
- The output signal `reconfig_fromgxb` is transceiver-block based so the number of bits for this instance is one since the number of channels is one. The input signal `reconfig_togxb` is a fixed width of three bits.
- Set the option **What is the starting channel number**? to **0**.

Instantiating Five Times Using the Above 1-Channel ALT2GXB:

- Instantiate the *ALT2GXB.v* file or the symbol file five times.
- Note that after instantiating five times, add the **starting channel number** parameter to the symbol file. Change the parameter option to **4**, **8**, **12**, and **16** for the instances 2, 3, 4, and 5 just created.
- If the instantiations are done in a verilog file, use the following command to force the parameter option to **4**, **8**, **12**, and **16** for the instances 2, 3, 4, and 5:

```
defparam inst2. starting_channel_number= 4;

defparam inst3. starting_channel_number= 8;
```

Dynamic Reconfiguration Controller Instance:

- Launch the ALT2GXB_RECONFIG MegaWizard.
- Set the **What is the number of channels controlled by the controller?** option to **20** so that five interface signals are enabled (`reconfig_fromgxb[4:0]`).
- Select the necessary analog control signals to write in and read out from the V_{OD}, pre-emphasis, equalization, and DC gain options.

ALT2GXB Instances and ALT2GXB_RECONFIG Instance Connections:

- Connect the `reconfig_fromgxb` signal from the ALT2GXB instance to the same signal in the `ALT2GXB_RECONFIG` instance.
- Connect the `reconfig_togxb` signal from the ALT2GXB_RECONFIG instance to the same signal in the ALT2GXB instance.

Channel and PMA Controls Reconfiguration

The write transaction of the controller is initiated on the assertion of the `write_all` signal. In PMA reconfiguration mode, the `write_all` signal writes the current state of all the selected input signals into the ALT2GXB instance channels. The write transaction involves the following sequence:

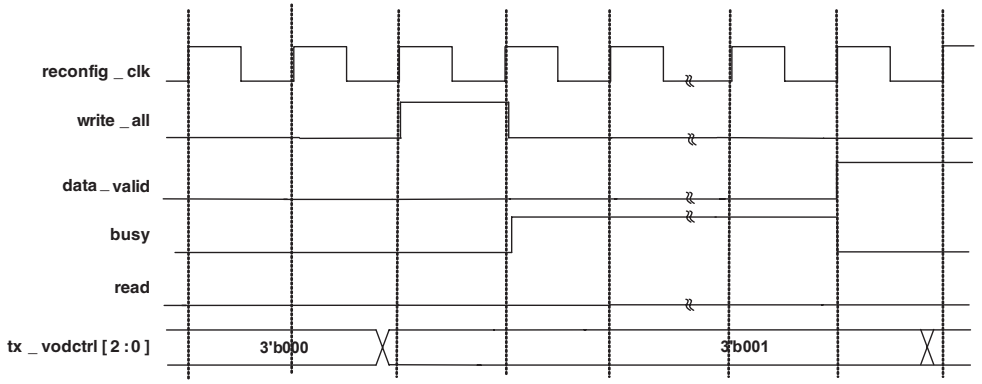
1. Read the control analog registers (read before write).
2. Write the current state of input signals of all channels into control registers.
3. Update the output control signals (optional read control ports if any of the read control ports are enabled).

If you select the read control port, the `data_valid` signal is enabled. Reading and updating all the output control signals is part of the write transaction. Therefore, the `data_valid` signal is asserted only when the write transaction is finished (`busy` signal is low) and all the output control ports are updated with the new data. When a write transaction is initiated and a set of values for the selected analog settings is being written, you cannot change the input values of the control ports until the transaction is completed. Otherwise, the results are unpredictable. The dynamic reconfiguration controller asserts the `busy` signal when you initiate a read or write transaction and is deasserted after the operation is complete.

Simultaneous write and read transactions are not allowed.

Figure 3–5 illustrates a write transaction for a transmit analog setting V_{OD} (`tx_vod`). The waveform shows a typical write transaction initiated by the pulsing of the `write_all` signal and also shows the behavior of the status signals `busy` and `data_valid`. Set the `reconfig_mode_sel` signal to **000** to reconfigure the analog settings of a transceiver channel.

Figure 3–5. Write Transaction Waveform - V_{OD}, Analog Settings Reconfiguration



In channel reconfiguration, only a write transaction can occur—no read transactions are allowed. Set the `reconfig_mode_sel` control signal to **001** to use the channel reconfiguration feature. When you use this feature, the dynamic reconfiguration controller requires that you provide a 16-bit word (`reconfig_data[15:0]`) on every write transaction, using the `write_all` signal. This 16-bit word is part of a Memory Initialization File (**.mif**, also known as MIF) that is generated by the Quartus II software when an ALT2GXB instance is compiled. Refer to "Channel Reconfiguration" on page 3–30 for more information about the MIF.

The dynamic reconfiguration controller ignores a new 16-bit word if the previously initiated write transaction is not complete. As explained above, an on-going or active write transaction is signified by the `busy` signal. You can only input a new word of 16-bits when the `busy` signal is de-asserted.

To properly initiate and complete a write transaction during channel reconfiguration, the dynamic reconfiguration controller provides additional signals. These signals are listed below and are classified into control and status signals.

The following are control signals (other than the write_all and reconfig_mode_sel signals):

- logical_channel_address[7:0]: Use this control signal to set the logical channel number of the channel that is being reconfigured by the dynamic reconfiguration controller. This signal gets enabled when the number of channels controlled by the dynamic reconfiguration controller is more than one. Since the channel reconfiguration is done on a per-channel basis, you have to use this signal and provide the necessary logical channel address to write the MIF words so that a successful channel reconfiguration is achieved for that channel.
- reset_reconfig_address: Use this optional control signal to reset the reconfig_address_out value to **0**. This reset control signal is only applicable in channel reconfiguration.

The following are status signals (other than the busy signal):

- reconfig_address_en: This is an optional output signal. The ALT2GXB_RECONFIG asserts this signal to indicate the change in value on the reconfig_address_out port. This signal only gets asserted after the dynamic reconfiguration controller completes writing the 16-bit data.
- reconfig_address_out[4:0]: This is an optional output signal. It provides the address value that you can use to read the appropriate word from the MIF. Use the value in this port in combination with the reconfig_address_en signal to decide when to initiate a new write transaction.
- channel_reconfig_done: This signal is available when you select the **Channel Reconfiguration** option in the dynamic reconfiguration controller. This port indicates that the ALT2GXB_RECONFIG megafunction has finished writing all the words of a MIF in a sequence. This signal is very useful for user logic to implement reset recommendations during and after dynamic reconfiguration. Refer to "Reset Recommendations" on page 3–66 for more information about using this signal.

- Error: The ALT2GXB_RECONFIG provides this status signal when you select the **Enable illegal mode checking** option or the **Enable self recovery** option in the **Error checks/data rate switch** tab. The conditions under which the error signal is asserted, when the above two options are enabled, are:
  - **Enable illegal mode checking** option—when you select this option, the dynamic reconfiguration controller checks whether an attempted operation falls under one of the seven conditions listed below. The dynamic reconfiguration controller detects these conditions within two reconfig_clk cycles, de-asserts the busy signal, and asserts the error signal for two reconfig_clk cycles.

    1. PMA controls - read operation:

    - None of the analog PMA read output ports (rx_eqctrl_out, rx_eqdcgain_out, tx_vodctrl_out, tx_preemp_0t_out, tx_preemp_1t_out, and tx_preemp_2t_out) are selected in the ALT2GXB_RECONFIG MegaWizard
    - reconfig_mode_sel is set to **0**
    - read signal is asserted

    2. PMA controls - write operation:

    - None of the analog PMA control write input ports (rx_eqctrl, rx_eqdcgain, tx_vodctrl, tx_preemp_0t, tx_preemp_1t, and tx_preemp_2t) are selected
    - reconfig_mode_sel is set to **0**
    - write_all signal is asserted

    3. Channel and/or TX PLL reconfiguration - read operation:

    - reconfig_mode_sel input port is set to **1**, **4**, **5**, or **6**
    - read signal is asserted

    4. Data rate switch - write operation with unsupported value:

    - The rate_switch_ctrl[1:0] input port is set to **11**
    - reconfig_mode_sel input port is set to **4** (if other reconfiguration mode options are selected in the **Reconfiguration settings** tab)
    - write_all is asserted

5. Data rate switch - write operation without input port:

- The `rate_switch_ctrl` input port is not used
- `reconfig_mode_sel` port is set to **4** (if other reconfiguration mode options are selected in the **Reconfiguration settings** tab)
- `write_all` is asserted

6. Data rate switch - read operation without output port:

- The `rate_switch_out` output port is not used
- `reconfig_mode_sel` port is set to **4** (if other reconfiguration mode options are selected in the **Reconfiguration settings** tab)
- `read` is asserted

7. Adaptive Equalization - read operation:

- `reconfig_mode_sel` input port is set to **7**, **8**, **9**, or **10**
- `read` signal is asserted

- **Enable self recovery** option—When this option is selected, the dynamic reconfiguration controller waits for a pre-defined number of `reconfig_clk` cycles based on the operation selected. If the `busy` signal does not go low within the pre-defined number of clock cycles, it asserts the `error` signal for two `reconfig_clk` cycles.

Example for Using Logical Channel Address to Perform Channel Reconfiguration

The dynamic reconfiguration controller provides an output port called `logical_channel_address`. This port is required for the channel reconfiguration and Channel and CMU PLL reconfiguration features to specify the logical transceiver channel that is to be reconfigured. The `logical_channel_address` value depends on how the ALT2GXB is instantiated in the design. In this section, the different ways of setting up the ALT2GXB instantiation and the corresponding `logical_channel_address` values for these transceiver channels are shown.

Example 1:

Consider a design example in which the ALT2GXB instantiation has six transceiver channels:

- In the ALT2GXB MegaWizard (in the **RECONFIG** tab) set the **starting channel number** option to **0**.
- In the ALT2GXB_RECONFIG MegaWizard, set the **Number of channels controlled by the reconfig controller** option to **6**.
- The `logical_channel_address` value for channel 0 is **0** (Channel 0 is the one that is assigned to `tx_dataout[0]`). Similarly, the `logical_channel_address` values for channels 1 through 5 are **1** through **5**, respectively.

Example 2:

Consider a design example with ALT2GXB instance an that has one transceiver channel (assume the instantiation name is **instantiation0**). The **starting channel number** option value for this channel is **0**. If you use this instantiation to create five additional transceiver channels, you will need the following **defparam** parameter settings (for Verilog designs) to change the starting channel number for the stamped instantiations:

defparam <instantiation1>. `starting_channel_number` = **4**

defparam <instantiation2>. `starting_channel_number` = **8**

defparam <instantiation3>. `starting_channel_number` = **12**

defparam <instantiation4>. `starting_channel_number` = **16**

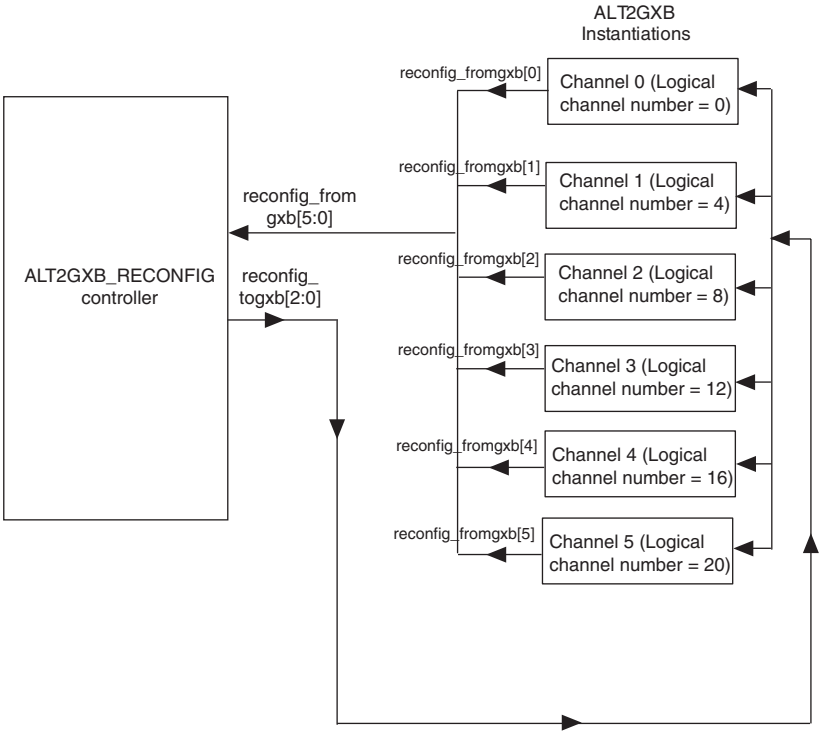
defparam <instantiation5>. `starting_channel_number` = **20**

Therefore, the **starting channel number** option values for channels 0 through 5 are **0**, **4**, **8**, **12**, **16**, and **20**, respectively.

- In the ALT2GXB_RECONFIG MegaWizard, set the **Number of channels controlled by the reconfig controller** option to **24**. By setting this option, you get the `reconfig_fromgxb` port with a bus width of 6.
- Connect the `reconfig_fromgxb[0 to 5]` port of the ALT2GXB_RECONFIG instantiation to the `reconfig_fromgxb` ports of transceiver channels 0 to 5, respectively (as shown in Figure 3–6).
- The `logical_channel_address` values for transceiver channels 0 through 5 (`tx_dataout[0]` to `tx_dataout[5]`) are **0**, **4**, **8**, **12**, **16**, and **20**, respectively.

The `logical_channel_address` value depends on the **starting channel number** option value that you set in the ALT2GXB MegaWizard for the transceiver channel. However, it does not depend on the physical placements of the transceiver channel. For example, you can physically assign `tx_dataout[1]` (`tx_dataout` of instantiation1) in the same transceiver block or in the other transceiver block. For both these assignments, the `logical_channel_address` value is **4** for instantiation1.

Figure 3–6. Multiple Stampings of a Single Channel ALT2GXB Instantiation



Example 3:

Consider a design example with ALT2GXB instance an that has two transceiver channels (assume the example name is **instantiation0**). The **starting channel number** option for this instance is set to **0**. If you want to create six transceiver channels, stamp this instance three times. Modify the **starting channel number** option for the other two instances to **4** and **8** using the **defparam** setting (for verilog design):

defparam <instantiation1>. `starting_channel_number` = **4**

defparam <instantiation2>. `starting_channel_number` = **8**

- In the ALT2GXB_RECONFIG MegaWizard, set the **Number of channels controlled by the reconfig controller** option to 12.
- Connect the `reconfig_fromgxb[0 to 2]` port of the ALT2GXB_RECONFIG instantiation to the `reconfig_fromgxb` ports of instantiation0 to instantiation2, respectively (as shown in Figure 3–7).
- In this case, the `logical_channel_address` values for transceiver channels 0 and 1 (`tx_dataout[0]` and `tx_dataout[1]`) are **0** and **1**. Similarly, the `logical_channel_address` values for channels 2 to 5 are **4**, **5**, **8**, and **9**, respectively. (The **starting channel number** option value for instantiation1 is **4**. Therefore, the `logical_channel_address` value for channel 2 is **4**).

Figure 3–7. Multiple Stampings of a Two Channel ALT2GXB Instantiation

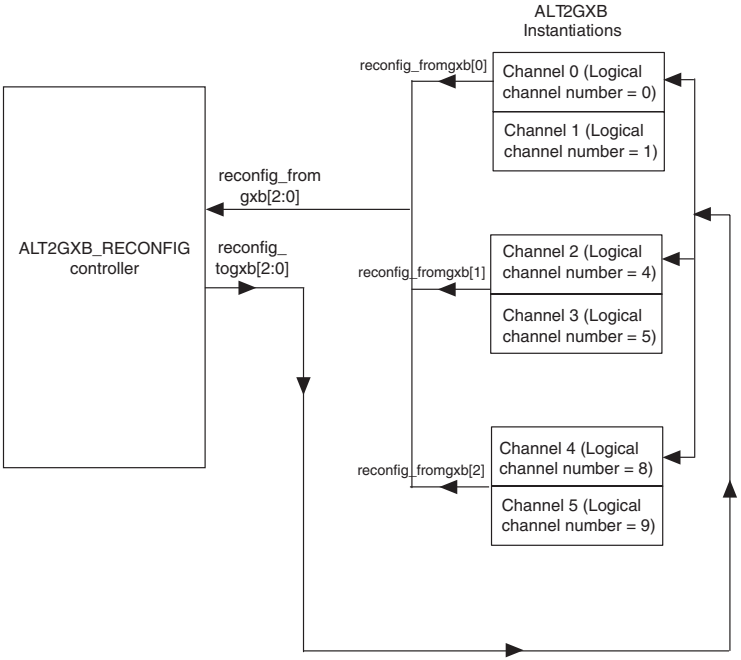
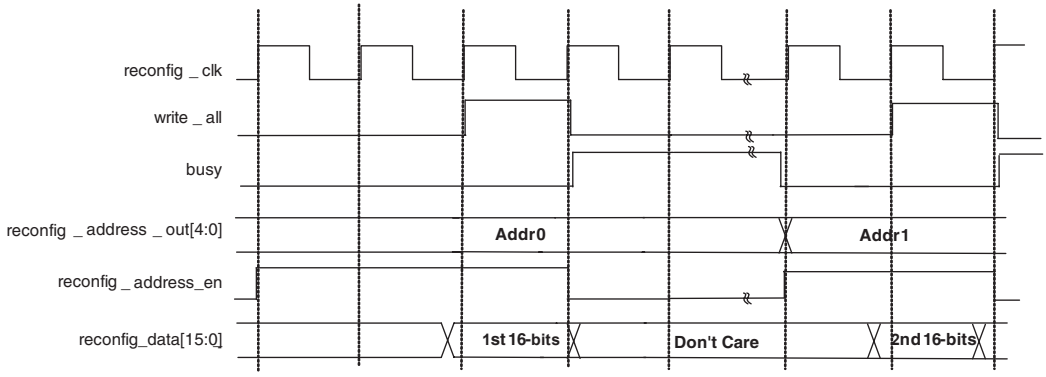


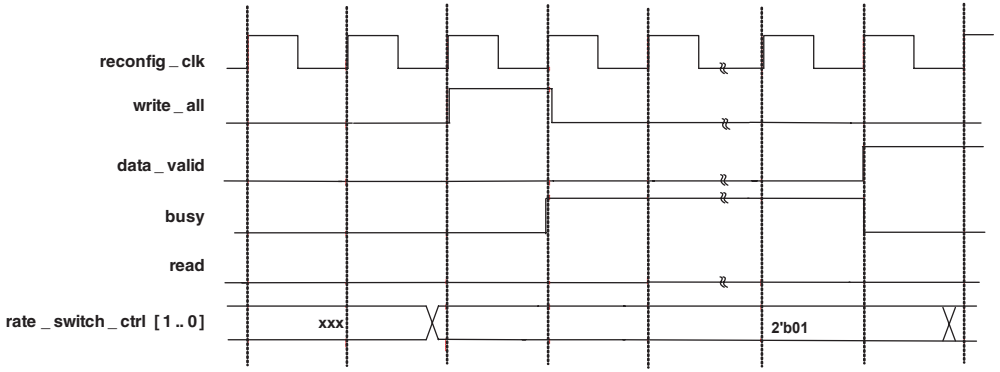
Figure 3–8 illustrates the write transaction for channel reconfiguration.

Figure 3–8. Write Transaction General Waveform – Channel Reconfiguration



A write is allowed in dynamic transmit rate switch mode. The control signal `rate_switch_ctrl[1:0]` determines which division factor is written into the ALT2GXB transmitter (Figure 3–9).

Figure 3–9. Write Transaction General Waveform – Dynamic Transmit Rate Switch Reconfiguration (Division 2)



To initiate a read transaction, assert the `read` signal. The data on the output control ports is not valid until the `data_valid` signal is high. The `data_valid` signal goes high when the entire selected output signals have valid read values. Both read and write transactions are based on the `reconfig_clk` and are edge triggered. Assert the `write_all` and `read` signal for one `reconfig_clk` cycle.

Figure 3–10. Read Transaction Waveform – V_{OD}, Analog Settings Reconfiguration

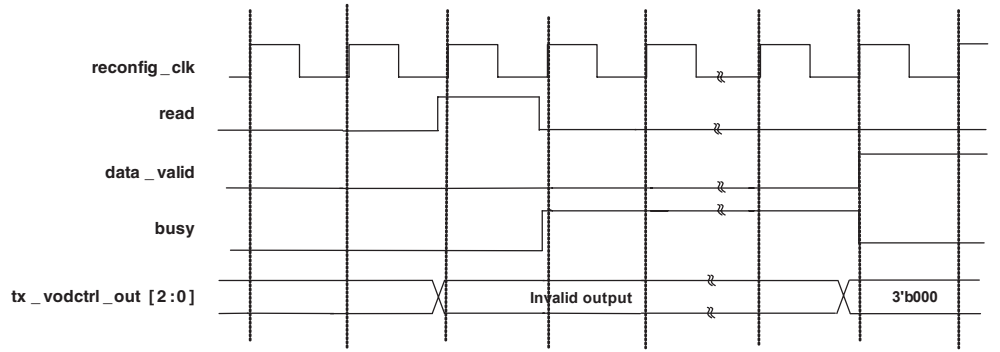
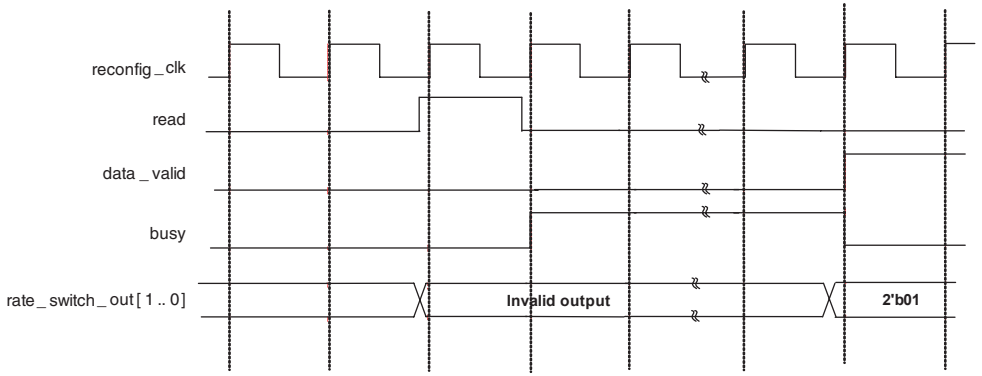


Figure 3–11. Read Transaction Waveform – Dynamic Transmit Rate Switch Reconfiguration (Division 2)



In addition to the PMA reconfiguration, the Quartus II software (version 6.1 and later) dynamic reconfiguration controller enables these two features:

- Channel Reconfiguration
- Dynamic Transmit Rate Switch

The following two sections explain these features.

Channel Reconfiguration

Introduction

Channel reconfiguration provides you the flexibility to reconfigure a channel by writing a new set of legal register bits into the ALT2GXB by the dynamic reconfiguration controller. With this feature you can either reconfigure the data rate of a channel or functional mode (including Basic mode with the custom mode enumeration [CME] features), or a mix of data rates and functional modes. The CME features are additional transceiver features introduced in Basic functional mode. Some of the CME features are controlled by PLD signals that allow you to dynamically control certain features in real time. However, some of the CME features are static and set through the Quartus II ALT2GXB configuration.

Channel reconfiguration only affects the channel involved in the reconfiguration; other channels are not affected.

Channel reconfiguration can be classified into two major areas—data rate reconfiguration and functional mode reconfiguration:

- Data Rate Reconfiguration—Data rate reconfiguration involves switching the data rate of a channel by switching between two TX PLLs and reconfiguring the RX PLLs. The two TX PLLs can be set to different base rates. With data rate reconfiguration, you can also switch the data rate using local clock dividers present in the transmit and receive sides of every transceiver channel. You can reconfigure these clock dividers to 1, 2, and 4. When you reconfigure the clock dividers, ensure that the functional mode supports the minimum and maximum data rate.
- Functional Mode Reconfiguration—this can be:
  - switched between one protocol functional mode to another protocol functional mode
  - switched between a protocol functional mode to a Basic functional mode
  - switched between a Basic mode to another Basic mode

There is no limit to the number of mode switches in channel reconfiguration, assuming transceiver and core clocking supports the transition.

Channel reconfiguration supports the following configurations of the physical transceiver channel:

- Duplex Channels (TX and RX)
- TX Only
- RX Only
- Independent TX/Independent RX in one physical channel

For the following discussion, the reference of a channel is a duplex channel, unless mentioned as TX-only or RX-only.

Design Flow

The Quartus II software provides a design flow called user memory initialization file (**.mif**, also known as MIF) flow to use the channel reconfiguration feature. This design flow involves writing the entire contents of the MIF for a channel. The Quartus II software generates the MIFs when you provide appropriate project settings (discussed below) and then compile an ALT2GXB instance. Each MIF has the settings for a full-duplex transceiver channel. The settings are all legal register settings of the transceiver channel. The ALT2GXB_RECONFIG instance reads the value in the MIF using the `reconfig_data[15..0]` port for every write transaction.

Each MIF contains twenty-eight 16-bit words if you enable the settings shown below.

For the Channel and CMU PLL Reconfiguration feature, the Quartus II software provides new settings that generates a MIF file with 38 words. This is discussed in "Quartus II Settings and Requirements" on page 3–111.

The Quartus II software creates the MIF under the **<Project_DIR>/reconfig_mif** folder. The file name is based on the design name and the rx_ and tx_ pin names. For example: **reconfig_datarate_1Gto2G_pin_af1_pin_af4.mif** (the Quartus II software automatically generates file name).You can change the MIF name. One design can have multiple MIFs (no limit) and one MIF can be used to reconfigure multiple channels. These MIFs can be stored in on-chip or off-chip memory.

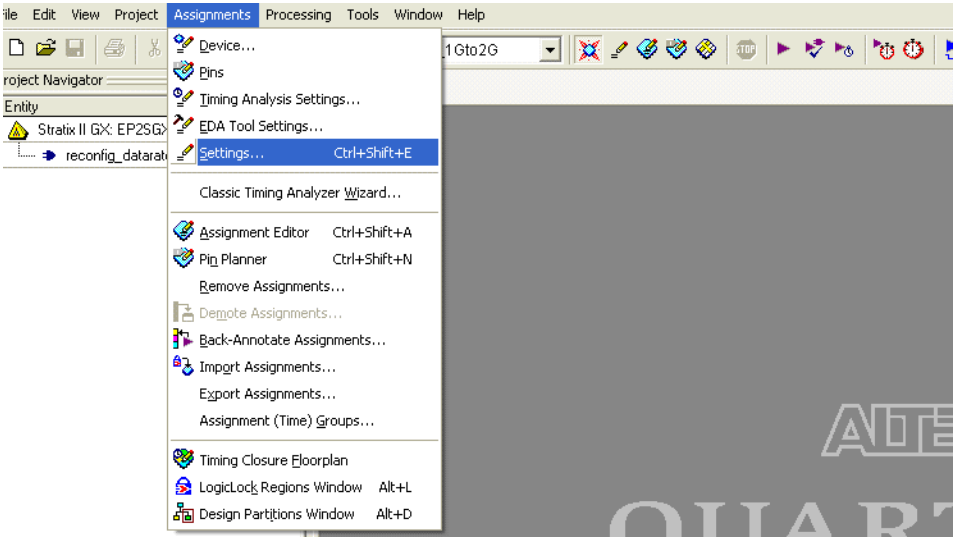
If you do not specify pins for the `tx_dataout` and `rx_datain` for the transceiver channel, the Quartus II software selects a channel and generates a MIF for that channel. However, the MIF can still be used for any transceiver channel.

MIF Generation in Quartus II Software

The MIF is not generated by default in a Quartus II compilation. There are three steps to enable MIF generation. Once the Quartus II software settings are enabled, a MIF is generated after you compile an ALT2GXB instance. The three steps to enable MIF generation are shown below.

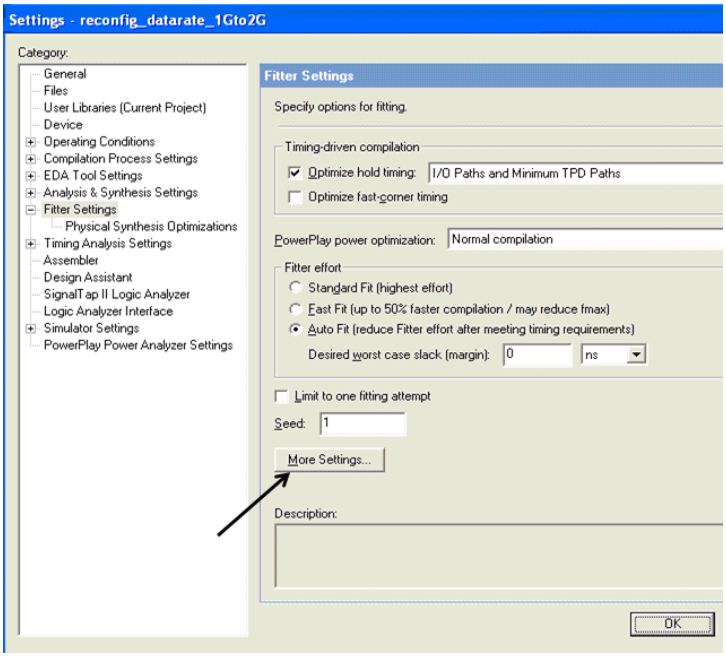
1. On the Assignments menu, select **Settings** (Figure 3–12).

Figure 3–12. MIF Generation, Step 1 (Settings Option)



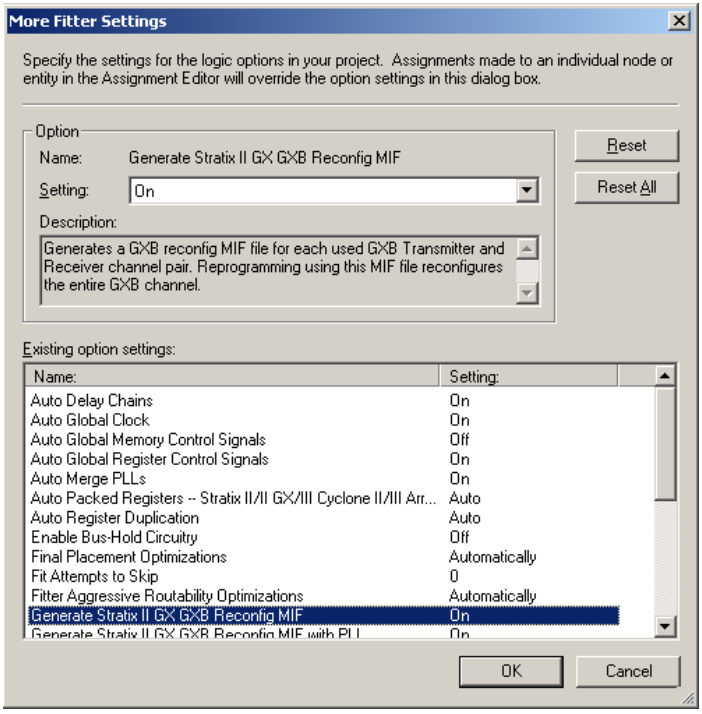
2. Select **Fitter settings**, then choose **More Settings** (Figure 3–13).

Figure 3–13. MIF Generation, Step 2 (Fitter Settings)



3. In the Option box of the **More Fitter Settings** page, set the **Generate Stratix II GX GXB reconfig MIF** option to **On** c (Figure 3–14).

Figure 3–14. MIF Generation, Step 3 (Enable Settings)



The MIF is generated in the Assembler stage of the compilation process. However, for any change in the design or the above settings, the Quartus II software runs through the fitter stage before starting the assembler stage.

As previously discussed, the channel reconfiguration can be a data rate reconfiguration using two TX PLLs and local clock dividers, or a functional mode reconfiguration, or both. To reconfigure a channel successfully, select the appropriate options in the ALT2GXB MegaWizard (discussed in the sections below).

ALT2GXB Configuration Related to Channel Reconfiguration

You must setup the following two system design aspects in a ALT2GXB MegaWizard instance:

- Transceiver and Core Clocking
- PLD Data Path Interface

Transceiver and Core Clocking

You must set up the core clocking and transceiver clocking options as part of channel reconfiguration for functional mode switchover or data rate transition. Transceiver clocking covers all the clock options you need to set up:

- Two TX PLLs for data rates and functional modes
- Input reference clocks for transmit and receive
- Internal clock MUX reference index setups

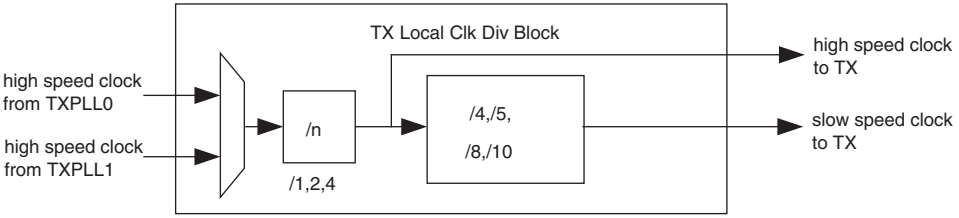
Core clocking covers the PLD interface clocking. PLD interface clocking is related to the parallel transmit and receive clocks (`tx_clkout` and `rx_clkout`). These clocks are used to parallel transmit data into and parallel receive data out of the transceiver. Core clocking is needed in any channel reconfiguration. Core clock assignments (clock grouping assignment and 0 PPM assignments) will override the core clocking set in the ALT2GXB instance. The details related to transceiver and core clocking are discussed in the following section. Transceiver and core clocking are classified as:

- Data rate switch using local clock block dividers
- Data rate switch based on clock frequencies of two PLLs in the transceiver block

Data Rate Switch Using Local Clock Block Dividers

If you intend to switch the data rate in multiples of 1, 2, and 4 of the base data rate, use the local clock dividers. Local clock dividers further divide the TX PLL base rate and are present in transmit and receive block of every transceiver channel (refer toFigure 3–15).

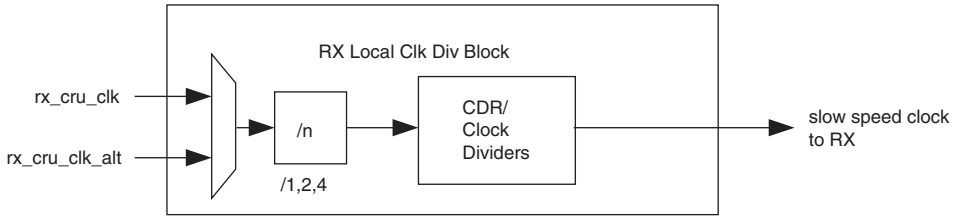
Figure 3–15. Transmit Local Clock Divider Block



Transmit local clock dividers are placed after the CMU PLLs and thus the TX PLLs are not affected during a data rate switch using local clock dividers.

Receive local clock dividers are placed before the RX PLL (CDR). Thus the RX PLL is affected every time the data rate switch using local clock dividers occurs. The Quartus II software data rate division factor chooses a combination of local clock dividers and feedback dividers present in the CDR that yields the best performance (refer to Figure 3–16).

Figure 3–16. Receive Local Clock Divider Block



To configure the local divider using the same TX PLL base setting, use the following steps:

1. Set the base setting on the CMU PLL (use the fastest data rate that is intended to be reconfigured to).
2. Set the local clock divider setting (use the effective data rate for that configuration).
3. Enable either the **Channel Internals** or **Channel Interface** option (refer to "Channel Internals" on page 3–53 and "Channel Interface" on page 3–53 for more information).

4. Setup core clocking (refer to "Transmitter Core Clocking" on page 3–45 and "Receiver Core Clocking" on page 3–48 for more information).

5. Finish the ALT2GXB configuration.

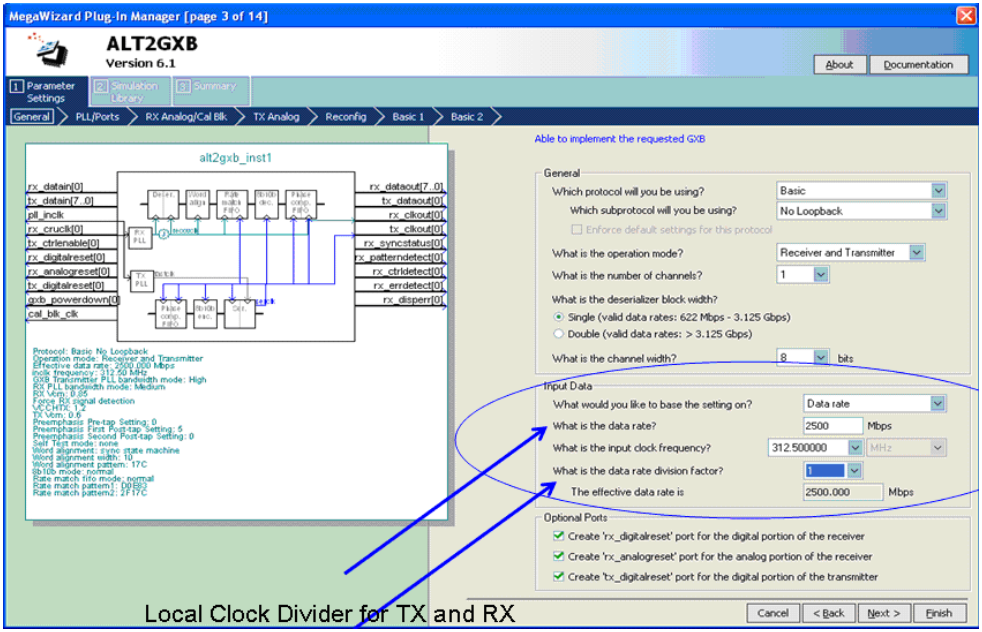
6. Repeat the previous 5 steps with the same TX PLL base setting and different local clock divider settings.

7. Group core clocking.

8. Lock down the pin assignments for the clocks and generate the MIFs for above instances.

Steps 1 and 2 are the only steps related to the local clock divider settings. Step 4 is a mandatory step and is an important part of clocking in every channel reconfiguration (refer to Figure 3–17).

Figure 3–17. ALT2GXB Instance—TX/RX Local Clock Divider



Data Rate Switch Based on Clock Frequencies of Two PLLs

If your application requires the transceiver to switch between multiple data rates, you can use channel reconfiguration to switch between the two TX PLLs in the transceiver block. The following sections explain how to setup two PLLs and achieve multiple data rates using channel reconfiguration:

1. Set the primary PLL (mode1) data rate setting.
2. Set the local clock divider (if needed).
3. Enable the **Channel Internals** option in the dynamic reconfiguration section of the ALT2GXB (refer to "Channel Internals" on page 3–53 for more information).
4. In **Channel Internals** option, enable the **use alternate reference clock (Mode 2)** option.
   - Set all the parameters related to alternate PLL protocols, data rates, bandwidth, and clock frequency.
5. Set the **what is the logical reference index?** option (refer to the Logical Reference Index).
6. Set the core clocking options—transmit and receive
   - This is a mandatory step for every channel reconfiguration that uses `tx_clkout` and `rx_clkout` (refer to "Transmitter Core Clocking" on page 3–45).
7. If there are no other settings to configure in the ALT2GXB, select **finish the ALT2GXB instantiation**.
8. Lock down the input reference clocks pin placements (refer to Pin Assignments).
9. Compile and generate a MIF for Mode1 as primary and Mode2 as alternate.
10. Similarly, generate a MIF for Mode2 as primary and Mode1 as alternate by going through steps 1 through 9 again (refer to "Example 1" on page 3–13).

Figure 3–18 illustrates steps 1 and 2.

Figure 3–18. Local Clock Divider Settings in the ALT2GXB MegaWizard

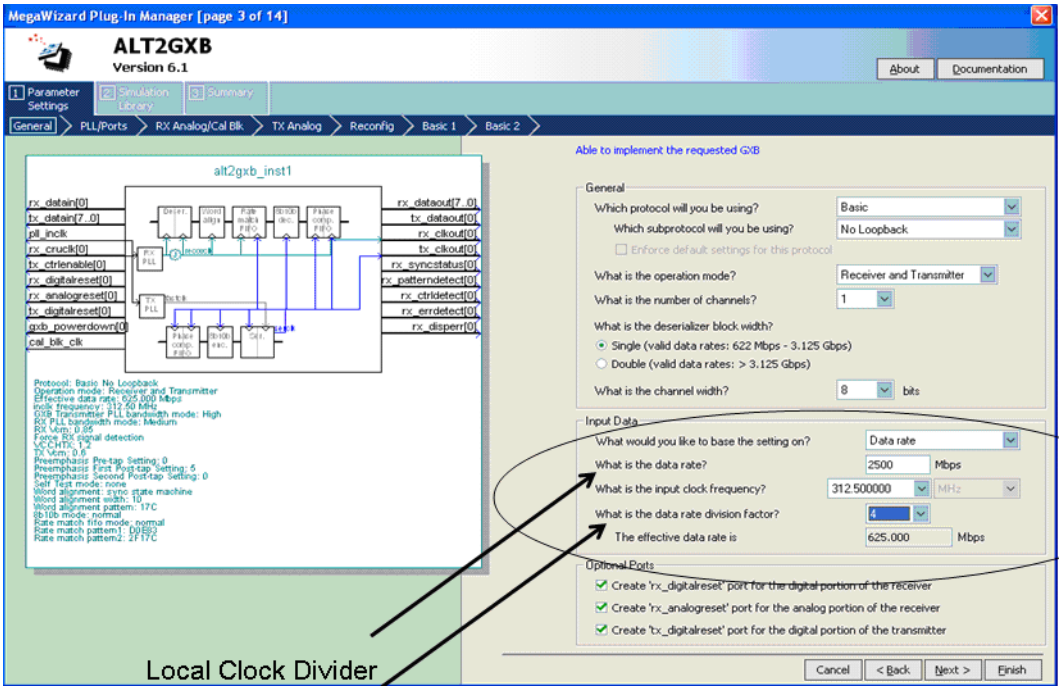
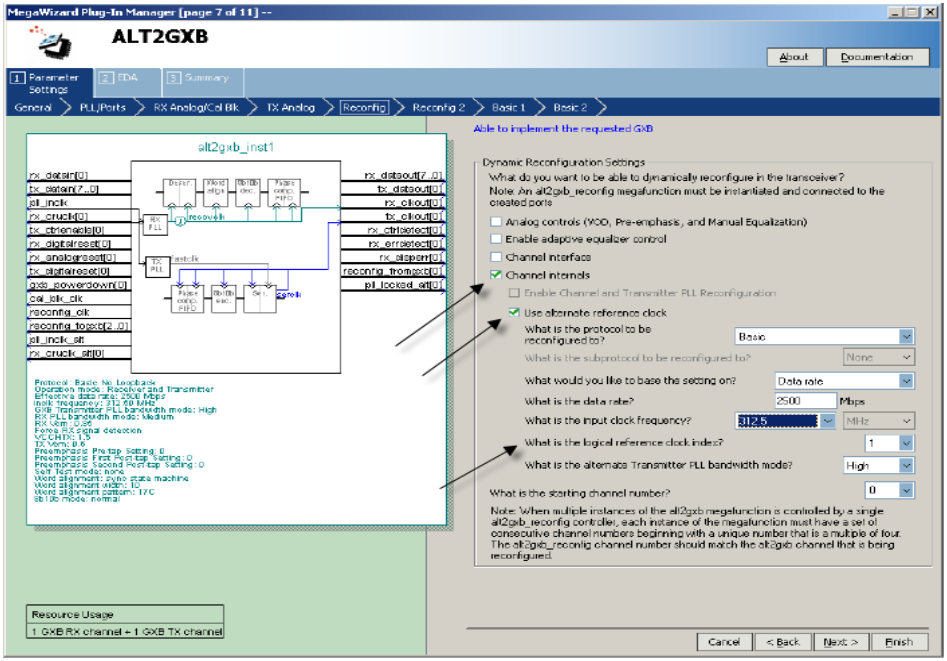


Figure 3–19 illustrates steps 3 and 4 using Basic mode at 2.5 Gbps. In Basic mode, the alternate PLL setup is the most flexible, you can choose and set from the supported bandwidth options and input reference clock frequencies. For example, if the alternate PLL happens to be a protocol functional mode like PCI-E or GIGE, the alternate PLL related options will be automatically populated by the Quartus II software. For more information about the **channel internal** option, refer to "Channel Interface" on page 3–53.

Figure 3–19. Channel Reconfiguration Settings for the ALT2GXB Megafunction



To enable dynamic reconfiguration of a transceiver channel, select either the **channel internals** or **channel interface** options. Selecting these fields creates the `reconfig_fromgxb` and `reconfig_togxb` ports in the ALT2GXB instance. The ALT2GXB_RECONFIG uses these ports to configure the transceiver channel.

In step 5, selecting the **What is the local reference clock index?** option controls the:

- MUX that selects the high-speed clocks from the two TX PLLs
- MUX that selects one of the two input reference clocks (`rx_cruclk` or `rx_cruclk_alt`) on the receive side

For example, consider a system switching from GIGE to SONET/SDH and vice versa. Since both protocols (GIGE with 125-MHz input reference clock and SONET/SDH OC48 with a 77.76-MHz input reference clock) cannot be achieved by one TX PLL, you need a two TX PLL setup. As part of the two TX PLL setup, you will set the logical reference index. To generate a MIF for the GIGE protocol, set the GIGE as the main configuration in the ALT2GXB instance and SONET/SDH mode as the

alternate protocol. This means that GIGE is achieved with the main PLL and the alternate PLL/input reference clock configuration is SONET/SDH OC48. Assume that you set the **Logical Reference Index** option value to **0** (in the **Reconfig** tab).

By setting the logical reference index to **0**, you provide the Quartus II software with the following information.

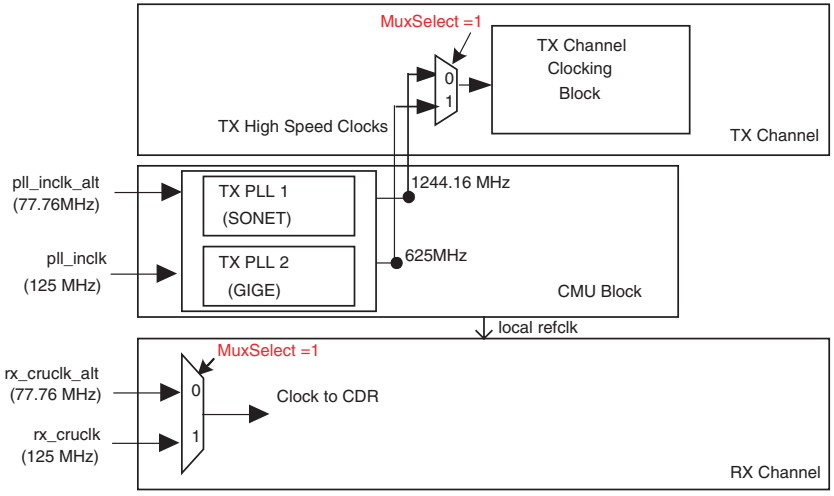
Selection values for the two MUXs mentioned above. The signal name `MuxSelect*` is an assumed name.

- Logical reference index = alternate input reference clock input leg
- MuxSelect* = ~(logical reference index)

In this case, since the logical reference index is set to **0** (represents the SONET/SDH), the TX PLL based on GIGE is routed to input1 of the clock MUX, and the alternate PLL configured for SONET/SDH is connected to input0 of the clock MUX. In the GIGE MIF, the clock MUX select value is set to **1** to choose the clock from the GIGE TX PLL.

Figures 3–20 and 3–21 show the clock MUX connections for GIGE and SONET/SDH, respectively.

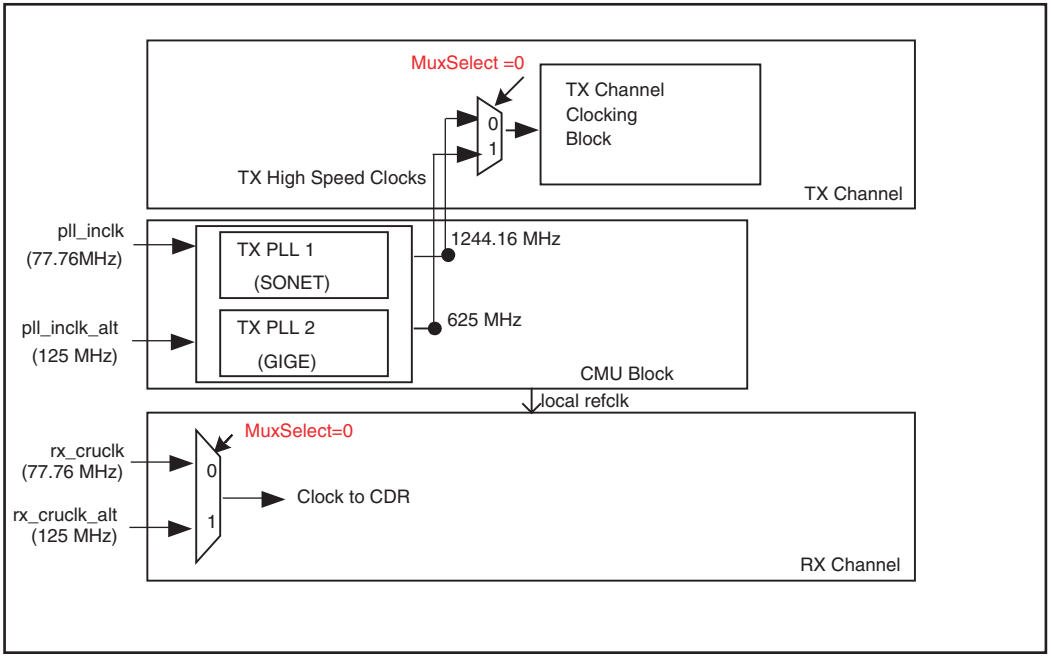
Figure 3–20. TX PLL for GIGE and SONET/SDH OC48 Mode Reconfiguration



To generate a MIF for the SONET/SDH protocol, set the SONET/SDH as the main configuration in the ALT2GXB instance and GIGE as the alternate protocol. This means SONET/SDH OC48 is achieved by the main PLL and the alternate PLL/input reference clock configuration is GIGE. Set the **Logical Reference index** option to **1** (since you have set the logical reference index to **0** for the GIGE instance).

In the SONET/SDH MIF, the clock MUX select value is set to **0** to choose the clock from the SONET/SDH TX PLL

Figure 3–21. MUX Setting - GIGE and SONET/SDH Mode, Logical Reference Clock Index = 1



When two modes are configured to switch from one to another using two TX PLLs, you have to carefully select the logical reference index. In this case, make sure the logical reference index that is set in one MIF is a complement in the second MIF.

Steps 6 is discussed in "Core Clocking" on page 3–45.

Channel Reconfiguration Supported Modes

Channel reconfiguration is supported in the following modes:

- Duplex channels (TX and RX)
- TX only
- RX only
- Independent TX/Independent RX in one physical channel

In the TX-only configuration, there is only one transmitter in a physical transceiver channel. The MIF for the TX-only file has the bits of the unused receiver, but these bits are disabled. The RX-only configuration is the same as the TX-only configuration except it pertains to the receiver.

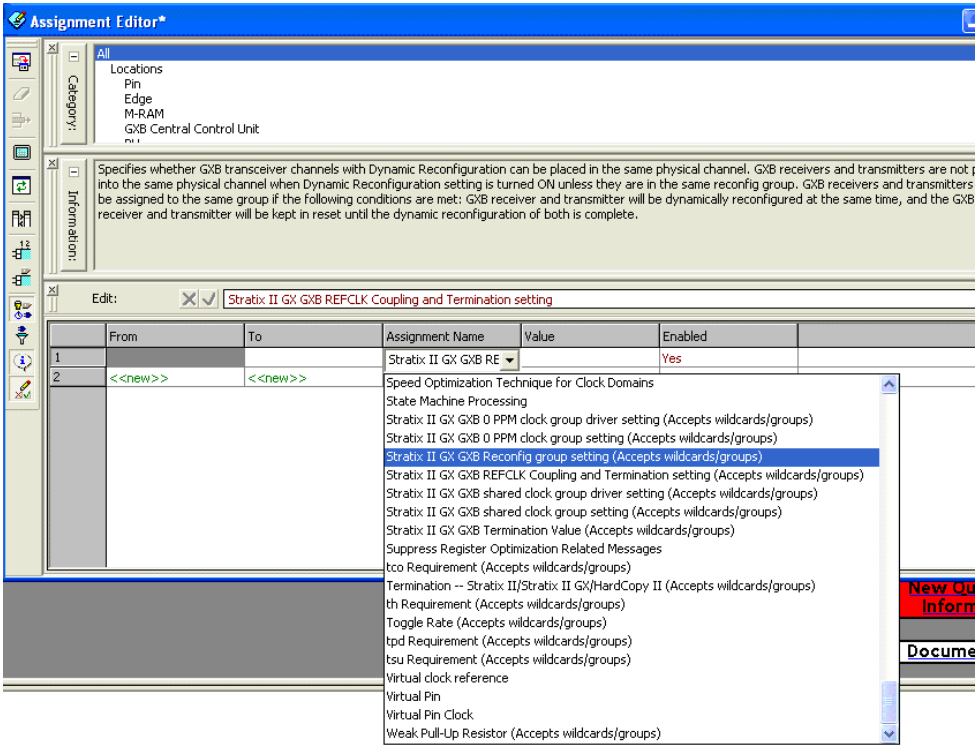
Channel reconfiguration from a TX-only mode to an RX-only mode and vice versa is not allowed.

The Quartus II software allows Independent TX-only configuration with another Independent RX-only configuration in one physical channel. To place an Independent TX configuration and an Independent RX configuration in one physical channel, follow the steps below:

- Perform the pin assignments accordingly
- Instruct the Quartus II software to merge or group the TX and RX register settings into one MIF

There are constraints with the Independent TX-only and Independent RX-only configurations. Both transmitter and receiver have to go through a reset sequence, even if the TX or RX is reconfigured. To merge or group the Independent TX-only and Independent RX-only configurations, place the RX and TX pins into one physical channel. You can accomplish this with the appropriate pin assignment and generation of a MIF through the Quartus II Assignment Editor by setting the **Stratix II GXB reconfig group setting** option to **ON** in the **Quartus Assignment Editor** (Figure 3–22).

Figure 3–22. Quartus II Assignment Editor – TX-Only/RX-Only Merge Option



Core Clocking

Core clocking configuration setup is a mandatory step in every channel reconfiguration. Core clocking is the write and read clock options for the Transmit Phase Comp FIFO and the Receive Phase Comp FIFO, respectively. Core clocking can be further classified to:

- Transmitter core clocking
- Receiver core clocking

Transmitter Core Clocking

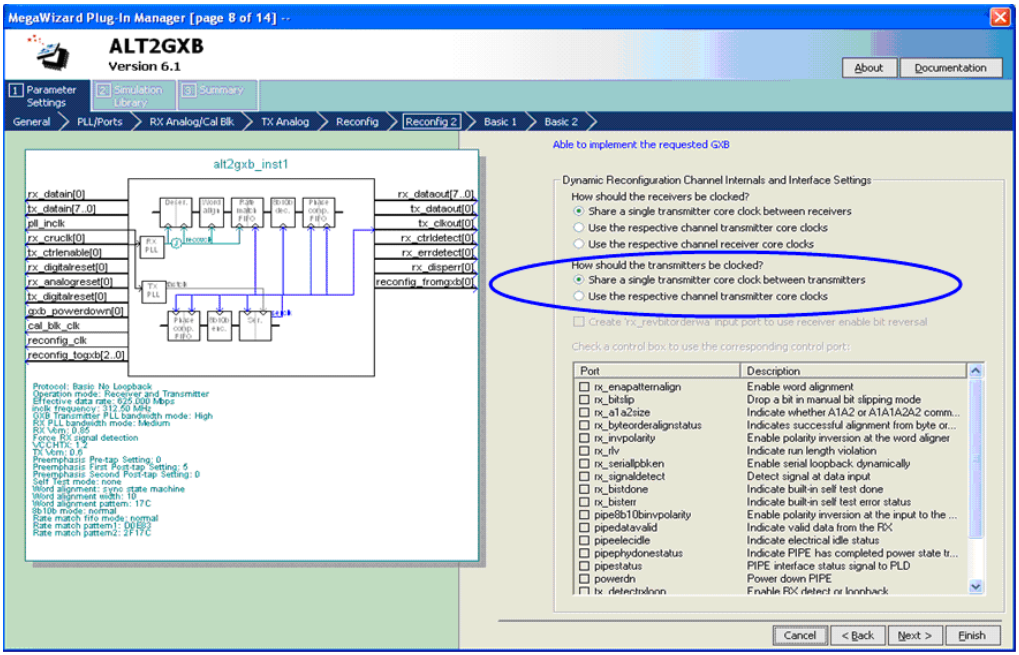
Transmitter core clocking is the write clocking options for the Transmit Phase Comp FIFO. The transmitter core clocking is used to write the parallel data into the Transmit Phase Comp FIFO from the PLD interface.

The possible transmit core clock options are:

- `tx_clkout` (the Quartus II software automatically routes to PLD and back into Transmitter Phase Comp FIFO)
- `tx_coreclk` (user-supplied input clock)

Dynamic reconfiguration allows both transmit clock options. The ALT2GXB MegaWizard provides two options only for the `tx_clkout` settings. When you select the `tx_clkout` options, ensure that the selected `tx_clkout` option is compatible for all the intended reconfiguration modes for the transceiver channel. The `tx_coreclk` selection and clock grouping assignments (Assignment editor) overrides the `tx_clkout` settings set in the ALT2GXB MegaWizard. Figure 3–23 shows the two options in transmit core clocking for `tx_clkout` routing.

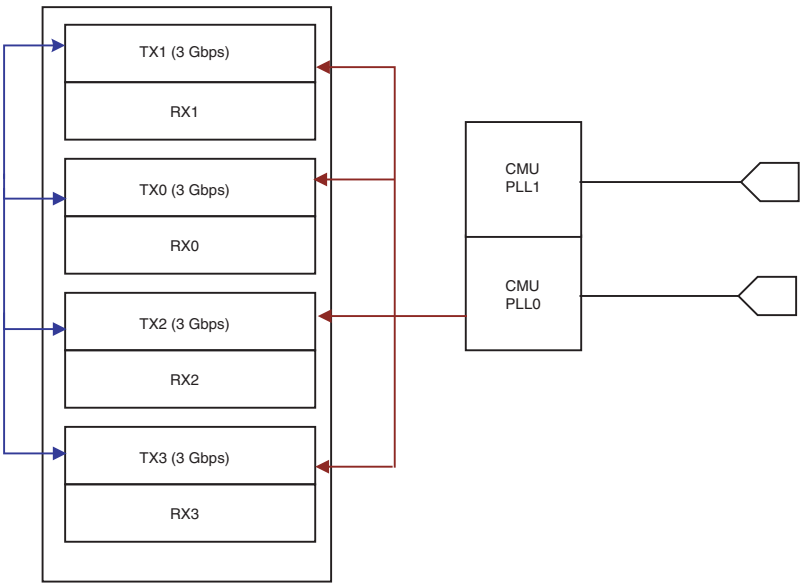
Figure 3–23. ALT2GXB MegaWizard Reconfiguration – Transmit Core Clocking Options



Option 1: Share a Single Transmitter Core Clock Between Transmitters
This option enables the Quartus II software to select channel 0 `tx_clkout` of a transceiver block and routes it to itself and three other channels. This is typically used when all four transmit channels are of the same mode (and also the same data rate) and switch to another mode.

For example, Figure 3–24 shows a setup which has all the transmits configured at 3 Gbps and in the same functional mode. With the dynamic reconfiguration controller and using the channel reconfiguration feature, all four channels switch to 1.5 Gbps and vice versa. Option 1 is applicable in this case and saves clock resources.

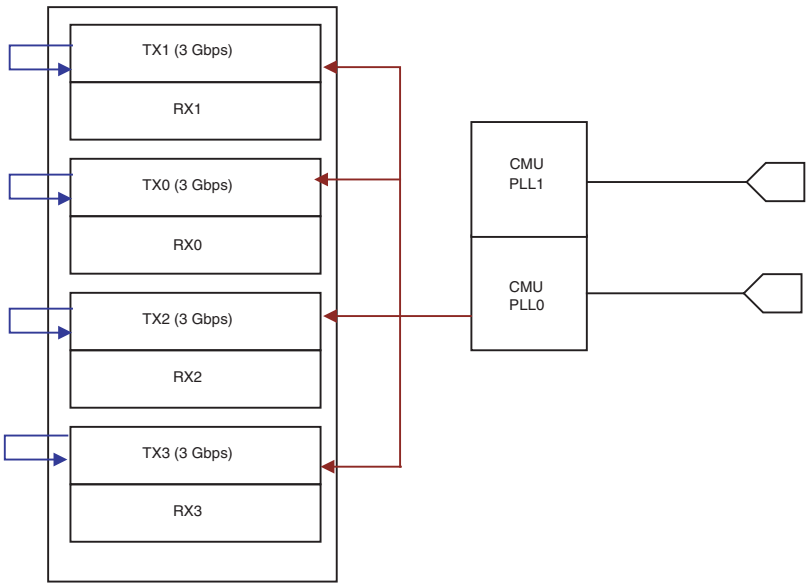
Figure 3–24. Option 1: Channel Reconfiguration—Transmit Core Clocking



Option 2: Use Respective Channel Transmitter Core Clocks
This option enables the Quartus II software to select the individual channel `tx_clkout` signals and route them back through PLD write clock resources to the TX Phase Comp FIFO. This type of core clocking configuration is needed when individual transmit channels can switch modes (basically, each channel switches to a different mode using channel reconfiguration).

Figure 3–24 shows a setup with all the transmitters configured at 3 Gbps and each one at a unique functional mode. Each channel can be switched to a different functional mode using the channel reconfiguration feature of the dynamic reconfiguration controller. In this case, option 2 is applicable.

Figure 3–25. Option 2: Channel Reconfiguration—Transmit Core Clocking



Receiver Core Clocking

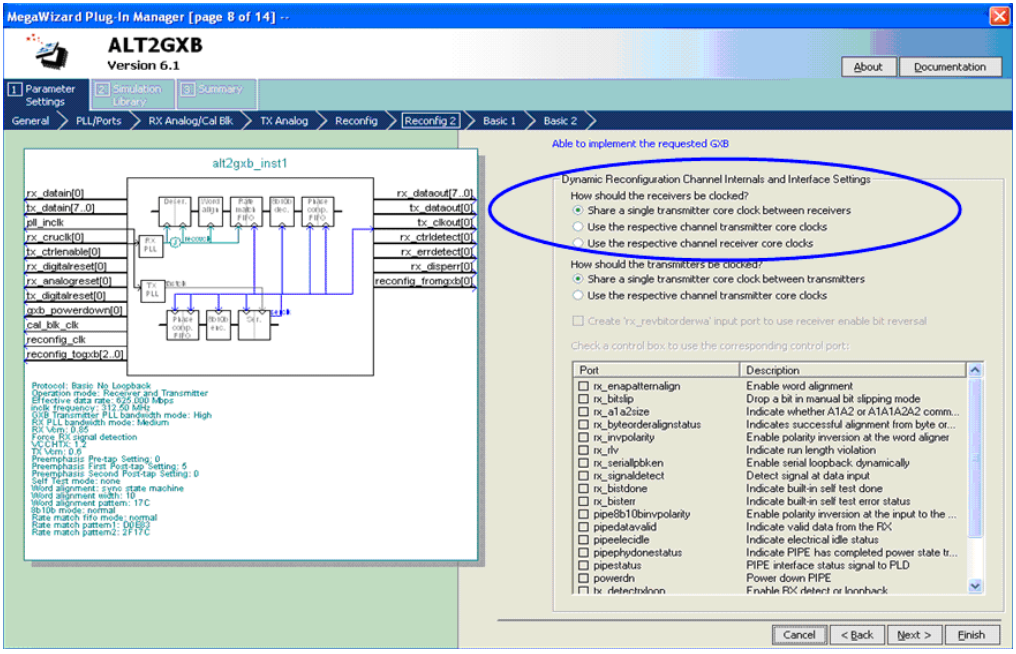
Receiver core clocking is the read clocking options for the Receive Phase Comp FIFO. The receiver core clocking is used to read the parallel data into the Receive Phase Comp FIFO from the PLD interface. The possible transmit core clock options are:

- `rx_clkout` (the Quartus II software automatically routes to PLD and back into Phase Comp)
- `rx_coreclk` (user-supplied input clock)

Dynamic reconfiguration supports both receive clock options. The ALT2GXB MegaWizard only asks for the `rx_clkout` settings. The Quartus II software automatically routes the clock paths based on a given mode setup. You must verify that clock routing is compatible with each

mode. The rx_coreclk selection and its grouping will override the rx_clkout settings set in the ALT2GXB MegaWizard. There are three options in the receiver core clocking for rx_clkout routing.

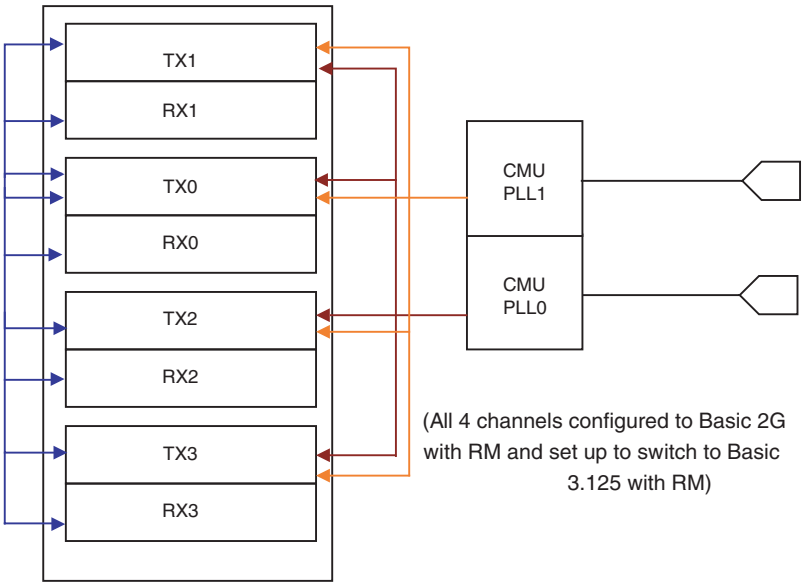
Figure 3–26. ALT2GXB MegaWizard Reconfiguration – Receive Core Clocking Options



Option 1: Share a Single Transmitter Core Clock Between Receivers
This option enables the Quartus II software to select channel 0 tx_clkout of a transceiver block and route it to all four receiver channels. This option is typically set when a transceiver block (all four channels) is in Basic or Protocol mode, with rate matching, switches to another Basic or Protocol mode with rate matching.

Figure 3–27 shows a setup with all four channels configured to a Basic 2 Gbps mode with rate matching, and then switches to a Basic 3.125 Gbps mode with rate matching. In this case, option 1 is applicable.

Figure 3–27. Option 1: Channel Reconfiguration—Receive Core Clocking



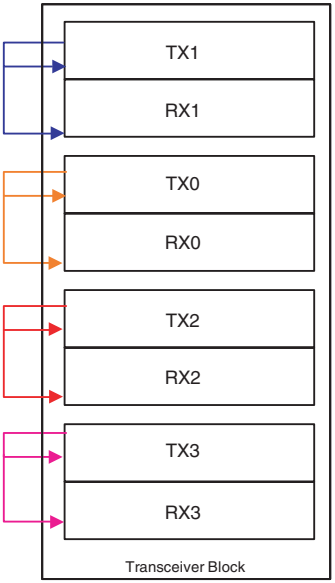
Option 2: Use Respective Channel Transmitter Core Clocks

This option enables the Quartus II software to select the individual channel `tx_clkout` signal and route it to the same channel's receiver PLD interface clock signal. Typically, this option is used when the individual channels in a transceiver block have rate matching with different data rates switched to another Basic or Protocol mode with rate matching.

Figure 3–28 illustrates a setup which has to switch between the following modes:

- TX1/RX1: Basic 1 Gbps with rate matching to Basic 2 Gbps with rate matching
- TX3/RX3: Basic 4 Gbps with rate matching to Basic 1 Gbps with rate matching
- TX0/RX0: Basic 3.125 Gbps with rate matching to 1 Gbps with rate matching and vice versa

Figure 3–28. Option 2: Separate Transmitter Core Clocks—Receive Core Clocking



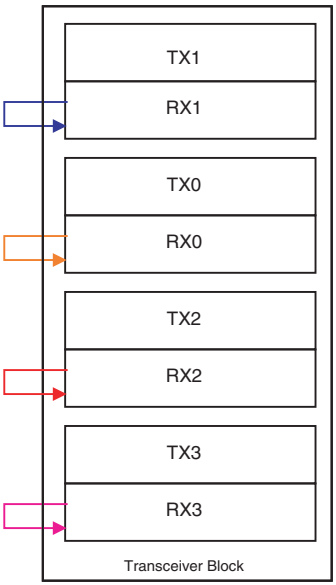
Option 3: Use Respective Channel Receiver Core Clocks
This option enables the Quartus II software to select the individual channel `rx_clkout` signal and route it to the same channel's receiver. Typically, this option is used when a channel is set up to switch from a Basic or Protocol mode with or without rate matching to another Basic or Protocol mode with or without rate matching.

Figure 3–29 illustrates a setup which intends to switch between the following modes:

- TX1/RX1: GIGE to SONET/SDH OC48
- TX2/RX2: Basic 2.5 Gbps no rate matching to Basic 1.244 G bps no rate matching

In this case, option 3 is applicable.

Figure 3–29. Option 3: rxclk_out—Receive Core Clocking



PLD Data Path Interface

For successful channel reconfiguration, you need to set up the following two system design aspects in the ALT2GXB MegaWizard:

- Transceiver and core clocking
- PLD data path interface

Transceiver and core clocking has been explained in detail in the preceding sections. This section discusses the PLD data path interface.

The PLD data path interface needs to be set up when dynamic reconfiguration involves the following:

- Mode switches involving PLD data width changes
- Mode switches involving enabling and disabling of PCS blocks or features (for example, CME features) in a transceiver channel

In the ALT2GXB instance's reconfiguration section, the PLD data path interface can be set up through two subsections:

- Channel Internals
- Channel Interface

Channel Internals

You should enable the **Channel internals** option if the modes that are switched to and from involve the following:

- Static PCS features (including CME features) are enabled or disabled.
- A data rate that needs another TX PLL to be set up (options related to alternate TXPLL need to be configured; for example, the **Use alternate reference clock** option and other sub-options).
- As long as the PLD data path width is not changed and no additional control and status signals are needed. In this case, you only need to enable the **Channel internals** option.

To reconfigure between two modes that differ only in the static features, generate the following MIFs:

- Generate a MIF with the **Channel internals** option enabled and set the appropriate PCS and analog features in the ALT2GXB megafunction.
- Generate a MIF with the **Channel internals** option enabled but with a different set of PCS features (same analog features) configured in the ALT2GXB megafunction.

In this case, the **Use alternate reference clock** option is not enabled, since the reconfiguration did not involve any changes to the data rate that would require another TX PLL.

You can use the **Channel internals** option in conjunction with the **Channel interface** option.

Channel Interface

The **Channel interface** option is enabled if the mode switches involve:

- PLD data path width changes
- PLD control and status flag changes

The **Channel interface** option involves the following:

- A new port called `tx_datainfull[43:0]` is enabled to the PLD interface port list on the transmit side (44-bits wide)
- A new port called `rx_dataoutfull[63:0]` is enabled to the PLD interface port list on the receive side (64-bits wide)
- Enabling the channel interface provides an option pane in the ALT2GXB megafunction where you can select the necessary ports for control and status signals that are needed for each of their channel reconfiguration.

The signals tx_datainfull[43:0] and rx_dataoutfull[63:0] replace the existing tx_datain and rx_dataout ports of a channel. The Quartus II fitter and mapper imposes fewer legal checks related to the connectivity of the signals in tx_datainfull, rx_dataoutfull, and other optional signals. For example, the PIPE mode signals pipestatus and powerdn can be potentially enabled through the ALT2GXB MegaWizard (enabled through the **Reconfig2** tab); the Quartus II software will not restrict this selection. In this case, the software assumes you are planning to switch to and from a PCI-E mode. Figures 3–30 and 3–31 show the MegaWizard pages you use to select the **channel internals** and **channel interface** options.

If the **Channel interface** option is enabled, the following signals are disabled:

- Receiver PLD interface:
  - rx_dataout[39:0]
  - rx_syncstatus[3:0]
  - rx_patterndetect[3:0]
  - rx_a1a2sizeout[3:0]
  - rx_ctrldetect[3:0]
  - rx_errdetect[3:0]
  - rx_disperr[3:0]
- Transmitter PLD interface:
  - tx_datain[39..0]
  - tx_ctrlenable[3:0]
  - tx_forcedisp[3:0]
  - tx_dispval[3:0]

Figure 3–30. ALT2GXB Reconfiguration – Channel Interface Enabled

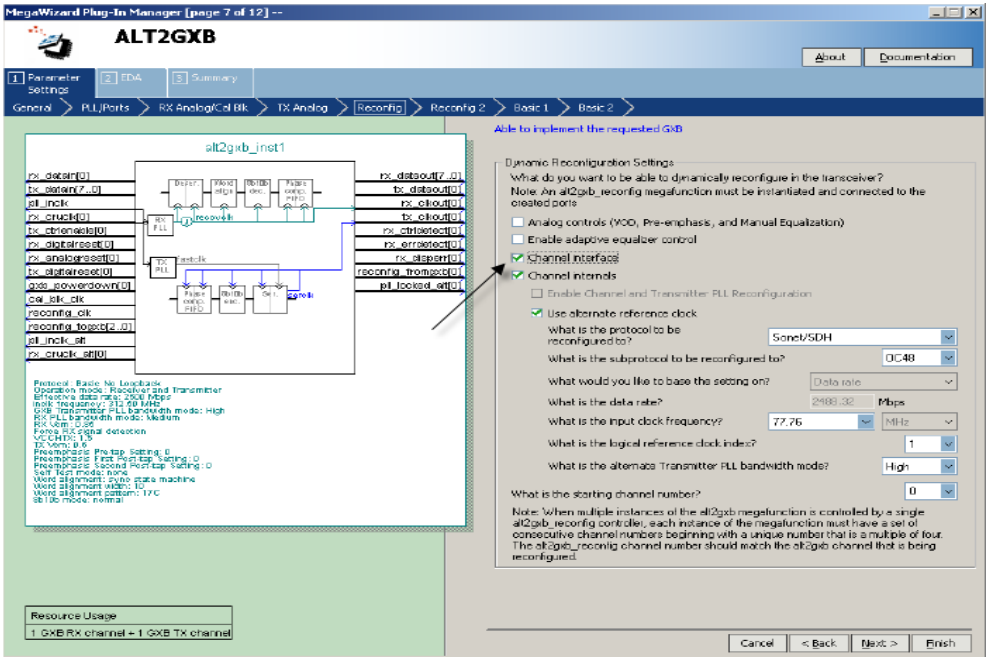
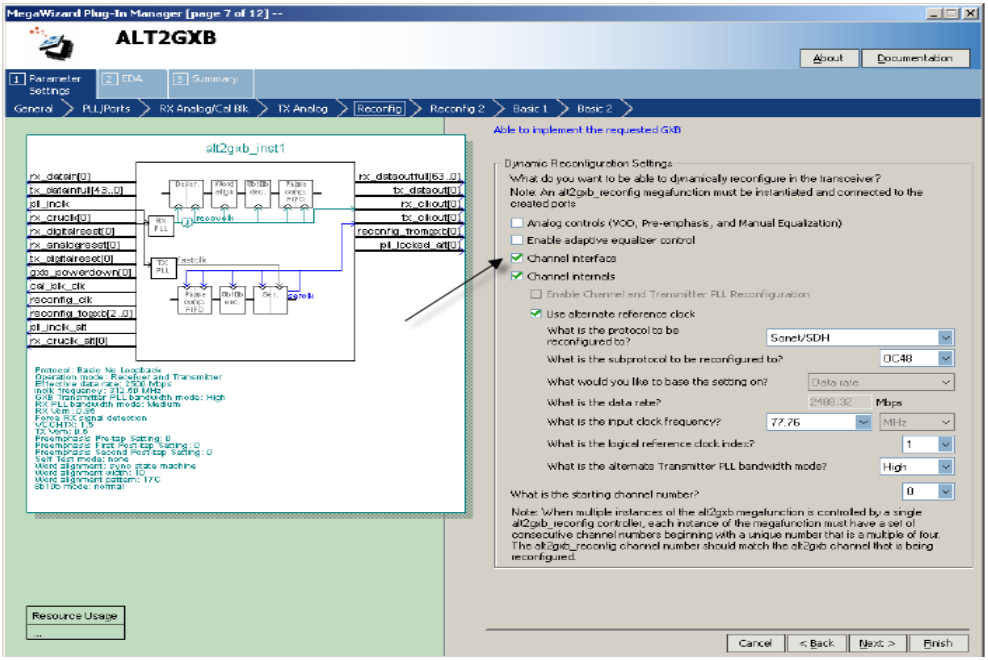


Figure 3–31. ALT2GXB Reconfiguration – Control/Status Signals, Channel Interface Enabled



Signal descriptions for `TX_datainfull[43:0]` and `RX_dataoutfull[63:0]` are shown in Tables 3–3 and 3–4.

Table 3–3. tx_datainfull[43:0] PLD Data Signal Descriptions (Part 1 of 3)

PLD Interface Description	Transmit Signal Description (Based on Stratix II GX Supported PLD Interface Widths)
8-bit PLD Interface	`tx_datainfull[7:0]`: 8-bit data (`tx_datain`)
	The following signals are used only in 8B/10B modes:
	`tx_datainfull[8]`: Control bit (`tx_ctrlenable`)
	`tx_datainfull[9]`: Force disparity enable for `tx_datainfull[7:0]` (non PIPE mode). Transmitter force disparity Compliance (PIPE) (`tx_forcedisp`) in all modes except PIPE. For PIPE mode, (`tx_forcedispcompliance`) is used.
	`tx_datainfull[10]`: Forced disparity value for `tx_datainfull[7:0]` (`tx_dispval`)
10-bit PLD Interface	`tx_datainfull[9:0]`: 10-bit data (`tx_datain`)

Table 3–3. tx_datainfull[43:0] PLD Data Signal Descriptions (Part 2 of 3)

PLD Interface Description	Transmit Signal Description (Based on Stratix II GX Supported PLD Interface Widths)
16-bit PLD interface with PCS-PMA set to 16/20 bits	Two 8-bit Data (tx_datain) tx_datainfull[7:0] - tx_datain (LSByte) and tx_datainfull[18:11] - tx_datain (MSByte)
	The following signals are used only in 8B/10B modes:
	Two Control Bits (tx_ctrlenable) tx_datainfull[8] - tx_ctrlenable (LSB) and tx_datainfull[19] - tx_ctrlenable (MSB)
	Force Disparity Enable tx_datainfull[9] - tx_forcedisp (LSB) and tx_datainfull[20] - tx_forcedisp (MSB)
	Force Disparity Value tx_datainfull[10] - tx_dispval (LSB) and tx_datainfull[21] - tx_dispval (MSB)
16-bit PLD interface with PCS-PMA set to 8/10 bits	Two 8-bit Data (tx_datain) tx_datainfull[7:0] - tx_datain (LSByte) and tx_datainfull[29:22] - tx_datain (MSByte)
	The following signals are used only in 8B/10B modes:
	Two Control Bits (tx_ctrlenable) tx_datainfull[8] - tx_ctrlenable (LSB) and tx_datainfull[30] - tx_ctrlenable (MSB)
	Force Disparity Enable For non-PIPE: tx_datainfull[9] - tx_forcedisp (LSB) and tx_datainfull[31] - tx_forcedisp (MSB) For PIPE: tx_datainfull[9] - tx_forcedispcompliance (LSB) and tx_datainfull[31] - tx_forcedispcompliance (MSB)
	Force Disparity Value tx_datainfull[10] - tx_dispval (LSB) and tx_datainfull[32] - tx_dispval (MSB)
20-bit PLD interface with PCS-PMA set to 20 bits	Two 10-bit Data (tx_datain) tx_datainfull[9:0] - tx_datain (LSByte) and tx_datainfull[20:11] - tx_datain (MSByte)
20-bit PLD interface with PCS-PMA set to 10 bits	Two 10-bit Data (tx_datain) tx_datainfull[9:0] - tx_datain (LSByte) and tx_datainfull[31:22] - tx_datain (MSByte)

Table 3–3. tx_datainfull[43:0] PLD Data Signal Descriptions (Part 3 of 3)

PLD Interface Description	Transmit Signal Description (Based on Stratix II GX Supported PLD Interface Widths)
32-bit PLD interface with PCS-PMA set to 16/20 bits	Four 8-bit Data (tx_datain) tx_datainfull[7:0] - tx_datain (LSByte) and tx_datainfull[18:11] tx_datainfull[29:22] tx_datainfull[40:33] - tx_datain (MSByte)
	The following signals are used only in 8B/10B modes:
	Four Control Bits (tx_ctrlenable) tx_datainfull[8] - tx_ctrlenable (LSB) and tx_datainfull[19] tx_datainfull[30] tx_datainfull[41] - tx_ctrlenable (MSB)
	Force Disparity Enable (tx_forcedisp) tx_datainfull[9] - tx_forcedisp (LSB) and tx_datainfull[20] tx_datainfull[31] tx_datainfull[42] - tx_forcedisp (MSB)
	Force Disparity Value (tx_dispval) tx_datainfull[10] - tx_dispval (LSB) and tx_datainfull[21] tx_datainfull[32] tx_datainfull[43] - tx_dispval (MSB)
40-bit PLD interface with PCS-PMA set to 20 bits	Four 10-bit Data (tx_datain) tx_datainfull[9:0] - tx_datain (LSByte) and tx_datainfull[20:11] tx_datainfull[31:22] tx_datainfull[42:33] - tx_datain (MSByte)

Table 3–4. rx_dataoutfull[63:0] PLD Data Signal Descriptions (Part 1 of 6)

PLD Interface Description	Receive Signal Description (Based on Stratix II GX Supported PLD Interface Widths)
8-bit PLD fabric interface	**The following signals are used in 8-bit 8B/10B modes:**
	rx_dataoutfull[7:0]: 8-bit decoded data (rx_dataout)
	rx_dataoutfull[8]: Control bit (rx_ctrldetect)
	rx_dataoutfull[9]: Code violation status signal. It indicates error detected in rx_dataoutfull[7:0], which is replaced by invalid code-group (invalid or running disp.error) in GIGE mode. In PCI Express, when code violation occurs, the EDB character is placed on the erroneous data byte (= K30.7) (rx_errdetect)
	rx_dataoutfull[10]: rx_syncstatus
	rx_dataoutfull[11]: Disparity error status signal. It indicates disparity error detected in rx_dataoutfull[7:0] (rx_disperr)
	rx_dataoutfull[12]: Pattern detect status signal (rx_patterndetect)
	rx_dataoutfull[13]: Reserved
	rx_dataoutfull[14]: Reserved
	rx_dataoutfull[14:13]: PIPE/PCI-E mode: 2'b00: data OK; 2'b01: 1 SKP deletion; 2'b10: elastic buffer underflow if data is 0xFE, else 1 SKP insertion; 2b11: elastic buffer overflow (rx_pipestatus)
	rx_dataoutfull[15]: Reserved
	The following signals are used in 8-bit SONET/SDH mode:
	rx_dataoutfull[7:0]: 8-bit un-encoded data (rx_dataout)
	rx_dataoutfull[8]: rx_a1a2sizeout
	rx_dataoutfull[10]: rx_syncstatus
	rx_dataoutfull[11]: Reserved
	rx_dataoutfull[12]: rx_patterndetect
10-bit PLD fabric interface	rx_dataoutfull[9:0]: 10-bit un-encoded data (rx_dataout)
	rx_dataoutfull[10]: rx_syncstatus
	rx_dataoutfull[11]: Reserved
	rx_dataoutfull[12]: rx_patterndetect
	rx_dataoutfull[13]: Reserved
	rx_dataoutfull[14]: Reserved
	rx_dataoutfull[15]: Reserved

Table 3–4. rx_dataoutfull[63:0] PLD Data Signal Descriptions (Part 2 of 6)

PLD Interface Description	Receive Signal Description (Based on Stratix II GX Supported PLD Interface Widths)
16-bit PLD interface with PCS-PMA set to 16/20 bits	Two 8-bit un-encoded Data (rx_dataout) rx_dataoutfull[7:0] - rx_dataout (LSByte) and rx_dataoutfull[23:16]- rx_dataout (MSByte)
	The following signals are used in 16-bit 8B/10B modes:
	Two Control Bits rx_dataoutfull[8] - rx_ctrldetect (LSB) and rx_dataoutfull[24]- rx_ctrldetect (MSB)
	Two Receiver Error Detect Bits rx_dataoutfull[9] - rx_errdetect (LSB) and rx_dataoutfull[25]- rx_errdetect (MSB)
	Two Receiver Sync Status Bits rx_dataoutfull [10] - rx_syncstatus (LSB) and rx_dataoutfull[42] - rx_syncstatus (MSB)
	Two Receiver Disparity Error Bits rx_dataoutfull [11] - rx_disperr (LSB) and rx_dataoutfull[43] - rx_disperr (MSB)
	Two Receiver Pattern Detect Bits rx_dataoutfull[12] - rx_patterndetect (LSB) and rx_dataoutfull[44]- rx_patterndetect (MSB)
	rx_dataoutfull[13] and rx_dataoutfull[45]: Reserved
	rx_dataoutfull[14] and rx_dataoutfull[46]: Reserved
	Two 2-bit PIPE Status Bits rx_dataoutfull[14:13] - rx_pipestatus (LSB) and rx_dataoutfull[46:45] - rx_pipestatus (MSB) PIPE/PCI-E mode: 2'b00: data OK 2'b01: 1 SKP deletion 2'b10: elastic buffer underflow if data is hexFE, else 1 SKP insertion 2'b11: elastic buffer overflow
	rx_dataoutfull[15] and rx_dataoutfull[47]: Reserved

Table 3–4. rx_dataoutfull[63:0] PLD Data Signal Descriptions (Part 3 of 6)

PLD Interface Description	Receive Signal Description (Based on Stratix II GX Supported PLD Interface Widths)
16-bit PLD interface with PCS-PMA set to 8/10 bits	Two 8-bit Data rx_dataoutfull[7:0] - rx_dataout (LSByte) and rx_dataoutfull[39:32] - rx_dataout (MSByte)
	The following signals are used in 16-bit 8B/10B mode:
	Two Control Bits rx_dataoutfull[8] - rx_ctrldetect (LSB) and rx_dataoutfull[40] - rx_ctrldetect (MSB)
	Two Receiver Error Detect Bits rx_dataoutfull[9] - rx_errdetect (LSB) and rx_dataoutfull[41]- rx_errdetect (MSB)
	Two Receiver Sync Status Bits rx_dataoutfull[10] - rx_syncstatus (LSB) and rx_dataoutfull[42]- rx_syncstatus (MSB)
	Two Receiver Disparity Error Bits rx_dataoutfull[11] - rx_disperr (LSB) and rx_dataoutfull[43] - rx_disperr (MSB)
	Two Receiver Pattern Detect Bits rx_dataoutfull[12] - rx_patterndetect (LSB) and rx_dataoutfull[44] - rx_patterndetect (MSB)
	rx_dataoutfull[13] and rx_dataoutfull[45]: Reserved
	rx_dataoutfull[14] and rx_dataoutfull[46]: Reserved
	Two 2-bit PIPE Status Bits rx_dataoutfull[14:13] - rx_pipestatus (LSB) and rx_dataoutfull[46:45]- rx_pipestatus (MSB) PIPE/PCI-E mode: 2'b00: data OK 2'b01: 1 SKP deletion 2'b10: elastic buffer underflow if data is hexFE, else 1 SKP insertion 2'b11: elastic buffer overflow (rx_pipestatus)
	rx_dataoutfull[15] and rx_dataoutfull[47]: Reserved
	The following signals are used in 16-bit SONET/SDH mode:
	Two 8-bit Data rx_dataoutfull[7:0] - rx_dataout (LSByte) and rx_dataoutfull[39:32] - rx_dataout (MSByte)
	Two Receiver Alignment Pattern Length Bits rx_dataoutfull[8] - rx_a1a2sizeout (LSB) and rx_dataoutfull[40]- rx_a1a2sizeout (MSB)
	Two Receiver Sync Status Bits rx_dataoutfull[10] - rx_syncstatus (LSB) and rx_dataoutfull[42] - rx_syncstatus (MSB)

Table 3–4. rx_dataoutfull[63:0] PLD Data Signal Descriptions (Part 4 of 6)

PLD Interface Description	Receive Signal Description (Based on Stratix II GX Supported PLD Interface Widths)
16-bit PLD interface with PCS-PMA set to 8/10 bits (continued)	Two Receiver Pattern Detect Bits rx_dataoutfull[12] - rx_patterndetect (LSB) and rx_dataoutfull[44] - rx_patterndetect (MSB)
20-bit PLD interface with PCS-PMA set to 20 bits	Two 10-bit Data (rx_dataout) rx_dataoutfull[9:0] - rx_dataout (LSByte) and rx_dataoutfull[25:16] - rx_dataout (MSByte)
	Two Receiver Sync Status Bits rx_dataoutfull[10] - rx_syncstatus (LSB) and rx_dataoutfull[26] - rx_syncstatus (MSB)
	rx_dataoutfull[11] and rx_dataoutfull[27]: Reserved
	Two Receiver Pattern Detect Bits rx_dataoutfull[12] - rx_patterndetect (LSB) and rx_dataoutfull[28] - rx_patterndetect (MSB)
	rx_dataoutfull[13] and rx_dataoutfull[29]: Reserved
	rx_dataoutfull[14] and rx_dataoutfull[30]: Reserved
	rx_dataoutfull[15] and rx_dataoutfull[31]: Reserved
20-bit PLD interface with PCS-PMA set to 10 bits	Two 10-bit Data rx_dataoutfull[9:0] - rx_dataout (LSByte) and rx_dataoutfull[41:32] - rx_dataout (MSByte)
	Two Receiver Sync Status Bits rx_dataoutfull[10] - rx_syncstatus (LSB) and rx_dataoutfull[42] - rx_syncstatus (MSB)
	rx_dataoutfull[11] and rx_dataoutfull[43]: Reserved
	Two Receiver Pattern Detect Bits rx_dataoutfull[12] - rx_patterndetect (LSB) and rx_dataoutfull[44] - rx_patterndetect (MSB)
	rx_dataoutfull[13] and rx_dataoutfull[45]: Reserved
	rx_dataoutfull[14] and rx_dataoutfull[46]: Reserved
	rx_dataoutfull[15] and rx_dataoutfull[47]: Reserved

Table 3–4. rx_dataoutfull[63:0] PLD Data Signal Descriptions (Part 5 of 6)

PLD Interface Description	Receive Signal Description (Based on Stratix II GX Supported PLD Interface Widths)
32-bit mode	Four 8-bit un-encoded Data (rx_dataout) rx_dataoutfull[7:0] - rx_dataout (LSByte) rx_dataoutfull[23:16] rx_dataoutfull[39:32] rx_dataoutfull[55:48] - rx_dataout (MSByte)
	The following signals are used in 32-bit 8B/10B mode:
	Four Control Data Bits (rx_dataout) rx_dataoutfull[8] - rx_ctrldetect (LSB) rx_dataoutfull[24] rx_dataoutfull[40] rx_dataoutfull[56] - rx_ctrldetect (MSB)
	Four Receiver Error Detect Bits rx_dataoutfull[9] - rx_errdetect (LSB) rx_dataoutfull[25] rx_dataoutfull[41] rx_dataoutfull[57] - rx_errdetect (MSB)
	Four Receiver Pattern Detect Bits rx_dataoutfull[10] - rx_syncstatus (LSB) and rx_dataoutfull[26] rx_dataoutfull[42] rx_dataoutfull[58] rx_syncstatus (MSB)
	Four Receiver Disparity Error Bits rx_dataoutfull[11] - rx_disperr (LSB) rx_dataoutfull[27] rx_dataoutfull[43] rx_dataoutfull[59] - rx_disperr (MSB)
	Four Receiver Pattern Detect Bits rx_dataoutfull[12] - rx_patterndetect (LSB) rx_dataoutfull[28] rx_dataoutfull[44] rx_dataoutfull[60] - rx_patterndetect (MSB)
	rx_dataoutfull[13], rx_dataoutfull[29], rx_dataoutfull[45] and rx_dataoutfull[61]: Reserved
	rx_dataoutfull[14], rx_dataoutfull[30], rx_dataoutfull[46], and rx_dataoutfull[62]: Reserved
	rx_dataoutfull[15], rx_dataoutfull[31], rx_dataoutfull[47], and rx_dataoutfull[63]: Reserved

Table 3–4. rx_dataoutfull[63:0] PLD Data Signal Descriptions (Part 6 of 6)

PLD Interface Description	Receive Signal Description (Based on Stratix II GX Supported PLD Interface Widths)
	The following signals are used in 32-bit SONET/SDH scrambled backplane mode:
	Four Control Data Bits (rx_dataout) rx_dataoutfull[7:0] - rx_dataout (LSByte) rx_dataoutfull[23:16] rx_dataoutfull[39:32] rx_dataoutfull[55:48] - rx_dataout (MSByte)
	rx_dataoutfull[8], rx_dataoutfull[24], rx_dataoutfull[40], and rx_dataoutfull[56]: four Reserved
	Four Receiver Sync Status Bits rx_dataoutfull[10] - rx_syncstatus (LSB) rx_dataoutfull[26] rx_dataoutfull[42] rx_dataoutfull[58] - rx_syncstatus (MSB)
	Four Receiver Pattern Detect Bits rx_dataoutfull[12] - rx_patterndetect (LSB) rx_dataoutfull[28] rx_dataoutfull[44] rx_dataoutfull[60] - rx_patterndetect (MSB)
40-bit mode	Four 10-bit Control Data Bits (rx_dataout) rx_dataoutfull[9:0] - rx_dataout (LSByte) rx_dataoutfull[25:16] rx_dataoutfull[41:32] rx_dataoutfull[57:48] - rx_dataout (MSByte)
	Four Receiver Sync Status Bits rx_dataoutfull[10] - rx_syncstatus (LSB) rx_dataoutfull[26] rx_dataoutfull[42] rx_dataoutfull[58] - rx_syncstatus (MSB)
	Four Receiver Pattern Detect Bits rx_dataoutfull[12] - rx_patterndetect (LSB) rx_dataoutfull[28] rx_dataoutfull[44] rx_dataoutfull[60] - rx_patterndetect (MSB)

ALT2GXB_RECONFIG Setup for Channel Reconfiguration

The ALT2GXB_RECONFIG (dynamic reconfiguration controller) instance must be set up for the channel reconfiguration feature. You can have one dynamic reconfiguration controller for one ALT2GXB instance (each ALT2GXB instance can have multiple transceiver channels) or one dynamic reconfiguration controller controlling more than one ALT2GXB instance. Select the feature **Channel Reconfiguration (Protocol Switch)**. Set the **What is the number of channels controlled by the reconfig controller?** option and select the optional signals in the channel reconfiguration section. Connect the ALT2GXB and ALT2GXB_RECONFIG instances.

Dynamic Transmit Rate Switch

Dynamic rate switch is ONLY available for the transmit side and not for the receive side. The control signal `rate_switch_ctrl[1:0]` sets up the division factor for the local divider inside the transmit side of the transceiver channel. The following is the encoding for the `rate_switch_ctrl` port:

- 00 - Divide by 1
- 01 - Divide by 2
- 10 - Divide by 4
- 11 - not supported, do not set this value

The above values are written and based in the ALT2GXB_RECONFIG instance initiating a write transaction by pulsing the `write_all` signal. This feature can be enabled through two other features—the analog settings and the Channel Reconfiguration. When two or more features are enabled, the `reconfig_mode_sel` signal needs to be set to the desired feature before a write transaction is initiated. A read transaction is allowed in this feature and the `rate_switch_out[1:0]` is required to read out the current data rate division factor through the ALT2GXB_RECONFIG instance. Do not perform a read transaction in this mode if the `rate_switch_out` is not selected in the ALT2GXB_RECONFIG MegaWizard.

The dynamic rate switch has no effect on the dividers on the receive side of the transceiver channel. It can be used only for the transmitter.

You must be aware of the device operating range before you enable and use this feature. There are no legal checks that are imposed by the Quartus II software, since it is an on-the-fly control feature. You also need to ensure that a specific functional mode supports the data rate range before dividing the clock when using this rate switch option.

Reset Recommendations

Altera recommends that you follow a proper reset sequence during and after for PMA controls reconfiguration, channel reconfiguration, and dynamic transmit rate switching.

PMA Controls Reconfiguration

During the first time the dynamic reconfiguration controller initiates a read or write, for example to change or read the PMA controls (VOD, pre-emphasis, equalization, or DC gain), the transceiver channel switches permanently from the registers that contain static transceiver settings to registers that are written by the dynamic reconfiguration controller. Due to this asynchronous switching, there may a few bit errors and transitions in the transceiver status signals.

Therefore, perform a **one time** PMA control read or write transaction from the dynamic reconfiguration controller during system bringup (initialization). This operation sets the transceiver to listen to the registers written by the dynamic reconfiguration controller during system bringup. By performing this read or write transaction during system bringup, you avoid errors during normal system operation.

Channel Reconfiguration

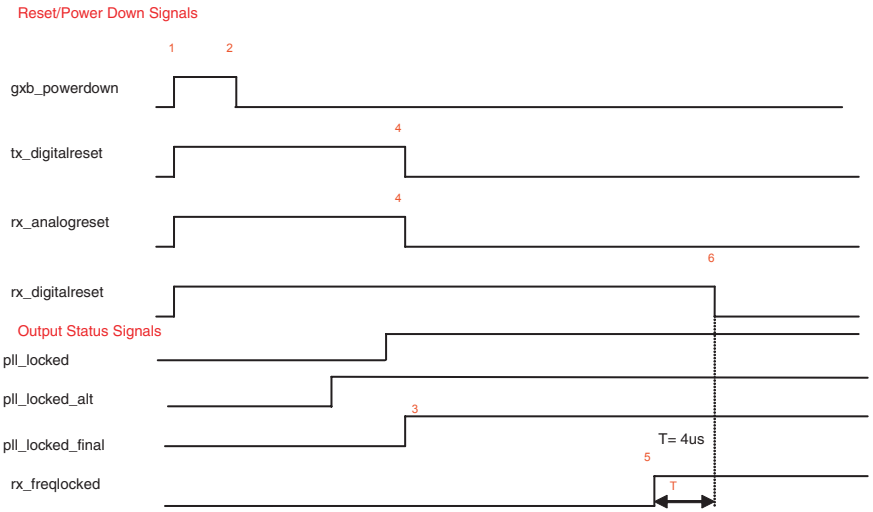
When you use the two TX PLLs in your design to reconfigure the channel, use the combination of the `pll_locked` and `pll_locked_alt` signals as part of your reset sequence.

In the two TX PLL designs, the updated TX PLL locked signal is:

`pll_locked_final* =` (`pll_locked` AND `pll_locked_alt`)

Figure 3–32 shows a waveform of the initialization and reset sequence of a design that uses main and alternate TX PLLs.

Figure 3–32. Reset Sequence (1 > 2 > 3 > 4 > 5 > 6)



The general reset sequence recommendations for bringing the device up are also valid.

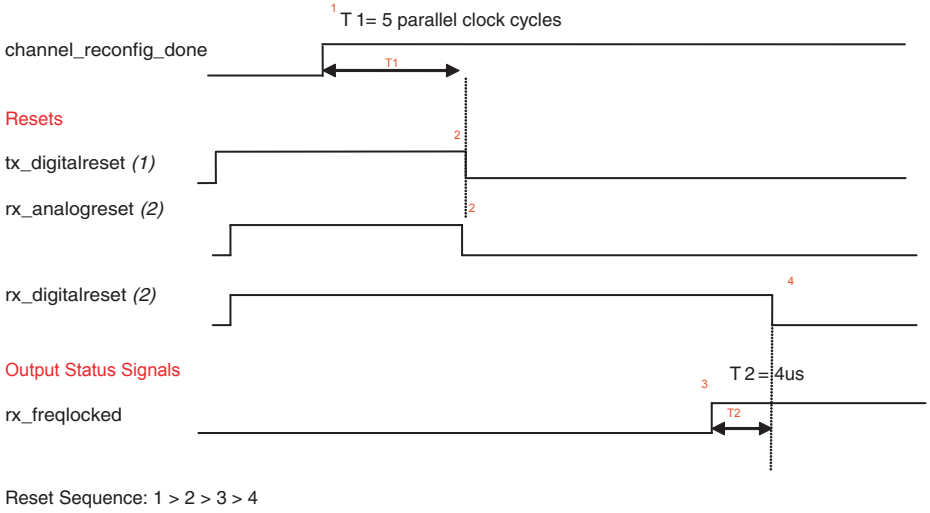
Refer to the Reset Control and Power Down section of the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Handbook* for more information.

In addition to the above recommendations, if you are using the Channel Reconfiguration feature, consider this additional recommendation:

If the channels are duplex and have individually switching modes using channel reconfiguration, consider the following:

- Reset controllers should be channel based
- The `channel_reconfig_done` signal can be used as a condition to reset the transmit and receive of the transceiver channel during and after channel reconfiguration
- Transmit digital resets (`tx_digitalreset`) are asserted during and after channel reconfiguration
- Assert the `rx_analogreset` signal and follow the reset sequence on the receiver side during and after the channel reconfiguration. Figure 3–33 shows the reset sequence for channel reconfiguration.

Figure 3–33. Reset Sequence (1 > 2 > 3 > 4) During and After Channel Reconfiguration



Notes to Figure 3–33:
(1) `tx_digitalreset` is valid in transmitter-only and duplex configurations.
(2) `rx_analogreset` and `rx_digitalreset` are valid in receiver-only and duplex configurations.

If the channel reconfiguration is done in a TX-only design, assert the transmit digital reset (`tx_digitalreset`) during and after channel reconfiguration, as shown in Figure 3–33. (`channel_reconfig_done` assertion signifies that the ALT2GXB_RECONFIG controller finished the channel reconfiguration by shifting an entire MIF into an ALT2GXB channel.)

If the channel reconfiguration is done in an RX-only design, assert the `rx_analogreset` signal and follow the reset sequence on the receiver side. Figure 3–33 shows the receiver sequence.

Dynamic Rate Switching

For a design using dynamic transmit rate switching, Altera recommends that you assert the `tx_digitalreset` when you initiate the rate switch operation until the `busy` signal goes low.

Overall Design Flow for Channel Reconfiguration

The following describes the design flow for Stratix II GX channel reconfiguration.

ALT2GXB Instantiation

1. Create an ALT2GXB MegaWizard instantiation. Select the protocol mode, single width, double width, data rate, and input reference clock frequency.
2. Select the required status signals.
3. In the **Reconfig** tab, select the channel internals.

If you intend to perform only rate division control, proceed to step 6.

4. If you would like to switch between two configurations that have different input clock frequencies, select the **Use alternate reference clock** option to configure the second TX PLL. Specify the **What is the logical reference clock index?** option value.
5. If the configuration requires different PLD interface widths or additional control signals provided in the **Reconfig2** tab, select the **Channel interface** option.
6. Select the appropriate clocking scheme in the **Reconfig2** tab.
7. Select the required additional control signals for the configuration in the **Reconfig2** tab (this is only enabled if the **Channel interface** option is selected).

MIF Generation:

8. Create a top-level design and connect the clock inputs in the RTL/schematic. Specifically, for the transceiver clock inputs, connect `pll_inclk` and `rx_cruclk` to the input pins that provide the clock for the protocol mode specified in the **General** tab of the ALT2GXB MegaWizard. Similarly, connect `pll_inclk_alt` and `rx_cruclk_alt` to the clock source that provides the clock for the protocol mode specified in the **Reconfig** tab of the ALT2GXB MegaWizard.

If you do not specify pins for `tx_dataout` and `rx_datain` for the transceiver channel, the Quartus II software selects a channel and generates a MIF for that channel. However, the MIF can still be used for any transceiver channel.

9. You can generate multiple MIF in the following two ways:

   Method 1:

- Compile the design created in step 8 and generate the first MIF.
- Update the ALT2GXB MegaWizard instance with the alternate configuration and connect the appropriate clock inputs, as mentioned in step 8.
- Compile the design to get the second MIF.

If you have to generate MIFs for many configurations, this method takes more time to complete.

Method 2:

- In the top-level design, instantiate all the different configurations of the ALT2GXB instantiation for which the MIF is required.
- Connect the appropriate clock inputs of all the ALT2GXB instantiations (see step 8).
- Generate the MIF. The MIFs are generated for all the ALT2GXB configurations.

This method requires attention when generating the MIF. Please check the following:

- The different ALT2GXB instantiations should have the appropriate **logical reference clock index** option values.
- The clock inputs for each instance should be connected to the appropriate clock source.
- When you generate the MIF, use proper naming for the files so you know the configuration supported by the MIF.

ALT2GXB_RECONFIG:

10. Create the ALT2GXB_RECONFIG instance. Select the **Channel reconfiguration** option to perform channel reconfiguration. Select `rate_switch_ctrl` for the transmit side data rate division.

11. Select the `reconfig_address_out` and `reset_reconfig_address` signals from the **Channel Reconfiguration** tab.

Control Logic for ALT2GXB_RECONFIG:

12. Implement logic to control the `ALT2GXB_RECONFIG` signals and to select the appropriate MIFs from memory and send the MIF data to ALT2GXB_RECONFIG.

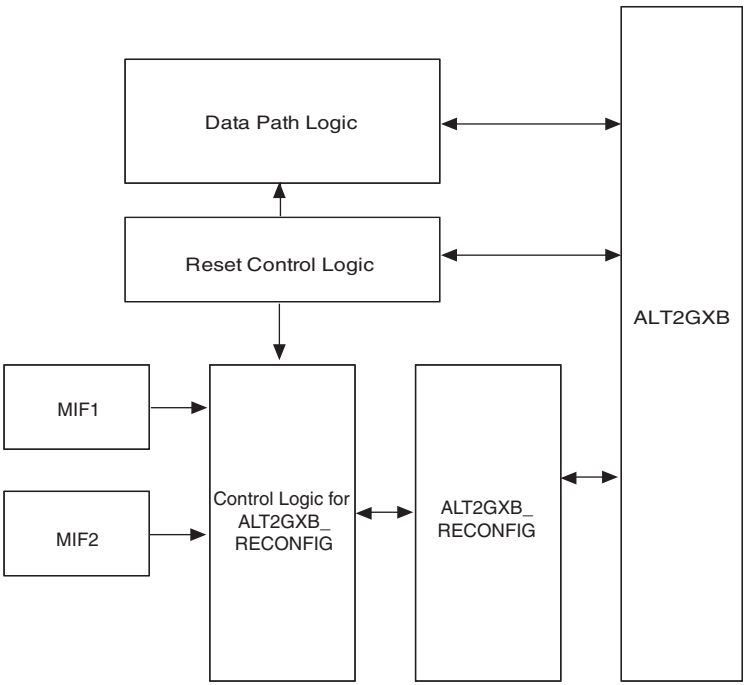
Data Path Logic:

13. Implement logic to handle the data from the transceiver. When you change the transceiver channel configuration, the data path, clocking, or PLD interface width may change. Therefore, implement logic in the PLD to transmit and receive data between the transceiver and the PLD logic, based on the transceiver configuration. Figure 3–34 shows the design functional blocks to perform channel reconfiguration.

Reset Control Logic:

14. Implement the reset control logic to handle the transceiver and the system resets. Refer to "Reset Recommendations" on page 3–66 for more information.

Figure 3–34. Functional Blocks for Channel Reconfiguration



Channel Reconfiguration Design Examples

This section provides three examples for performing dynamic reconfiguration on a transceiver channel.

Example 1—Configuring a Transceiver Channel Between GIGE Mode and SONET/SDH OC48 Mode

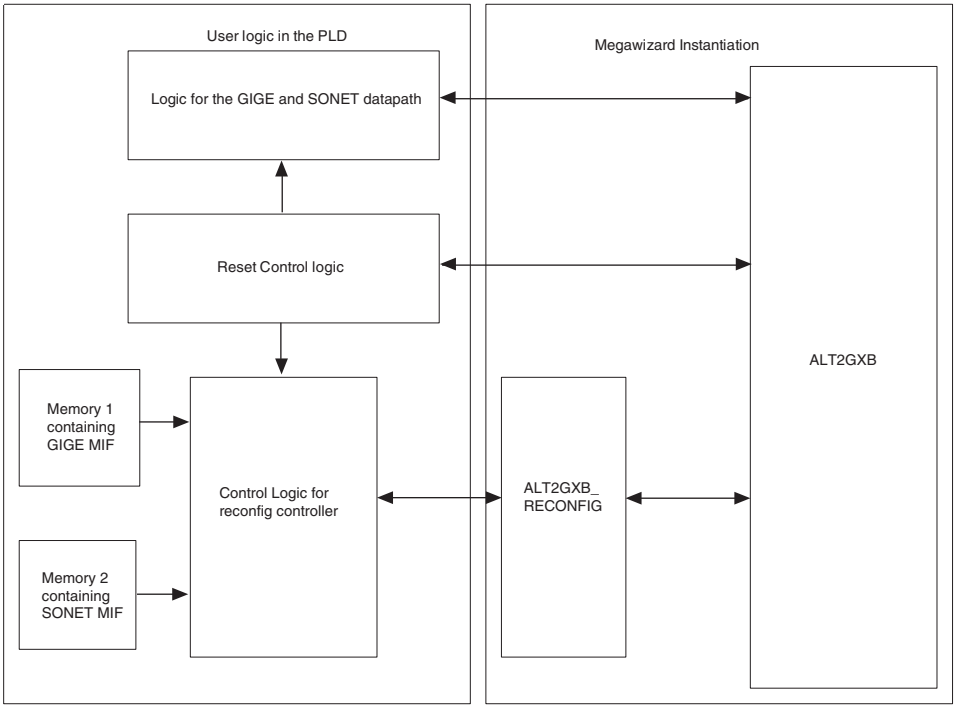
The GIGE mode in the ALT2GXB megafunction is different from the SONET/SDH OC48 mode in data path, clocking, and PLD interface width. The differences between the two modes are listed in Table 3–5.

Table 3–5. Differences Between GIGE and SONET/SDH OC48

Number	Functional Block	GIGE	SONET/SDH OC48
1	PLD width	8	16
2	8B/10B enabled	Yes	No
3	Rate matcher	Yes	No
4	Byte order block	No	Yes
5	Clock used for synchronizing the receive output data (`rx_dataout`)	`tx_clkout` (since rate matcher is used)	`rx_clkout`
6	Data rate	1.25 Gbps	2.488 Gbps
7	Allowed input reference clock	62.5 MHz 125 MHz	77.76 MHz 155.52 MHz 311.04 MHz 622.08 MHz
8	PCS-PMA interface width	10 (since data is 8B/10B encoded)	8

These differences determine the selection of parameters in the ALT2GXB MegaWizard and the required PLD logic to configure a transceiver channel between these two modes. Figure 3–35 shows the required functional blocks to perform channel reconfiguration.

Figure 3–35. Reconfiguring Between GIGE and SONET /SDH OC48 Modes



The discussion of the functional blocks is divided into four sections. The topics discussed in each section are as follows:

- Section I—Lists the steps to configure the ALT2GXB instance to generate the MIF for GIGE and SONET/SDH OC48 modes. Then lists the steps to create the ALT2GXB_RECONFIG instance.
- Section II—Sets up the control logic for the ALT2GXB_RECONFIG controller.
- Section III—Logic to process the GIGE and SONET/SDH data. This logic is required due to the differences in the data interface widths and the clocking between the two modes (shown in Table 3–5).
- Section IV—Resets the control logic to control the transceiver and system resets.

Section I
Use the following steps to generate a MIF for GIGE and SONET/SDH OC48 modes:

1. Generate the ALT2GXB instantiation for GIGE mode.
2. Generate the AL2GXB_RECONFIG instantiation.
3. Create a top-level design and generate the MIF for the GIGE protocol mode.
4. Modify the ALT2GXB instantiation for SONET /SDH OC48 mode.
5. Generate the MIF for SONET/SDH OC48 mode.
6. Initialize two memory elements with the MIF contents and write logic to select the MIF and to control the ALT2GXB_RECONFIG instance.

Step 1—Generate the ALT2GXB Instantiation for GIGE Mode:
This example shows the ALT2GXB instantiation for one-channel GIGE and SONET/SDH mode.

1. Set the protocol to **GIGE** mode in the first screen.

   Select the following control and status signals:

   - `rx_digitalreset`
   - `tx_digitalreset`
   - `rx_analogreset`
   - `rx_pll_locked`
   - `rx_freqlocked`

2. Add the other required status signals.

For a list of ALT2GXB signals and their functionality, refer to "Stratix II GX ALT2GXB Ports List" on page 2–2 in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

3. In the **Reconfig** tab, set the following options:

   - If you need control of the transceiver PMA values, select the Analog PMA controls. For more information about PMA controls, refer to "Introduction" on page 3–1.

4. Select the **Channel internals** and **Use alternate reference clock** options. Selecting these options enable the second PLL for the SONET/SDH OC48 mode. A second PLL is needed because of the difference in the required input clock frequency and data rate between the GIGE and SONET/SDH OC48 modes (refer to rows 6 and 7 in Table 3–5).

5. Set the **What is the protocol to be reconfigured?** option to **SONET/SDH**.

6. Set the **sub protocol** option to **OC48** (see Figure 3–31).

7. Select the **input clock frequency** and **alternate transmitter PLL bandwidth mode** options based on the requirements. The allowed reference clock input frequencies for SONET/SDH OC48 are specified in row 7 of Table 3–5.

8. For the **What is the logical reference index?** option, select **1** or **0**. The Quartus II software uses the logical reference index to select the PLL clock outputs for the transmit and receive channels when configured to SONET/SDH OC48 protocol. The MUX values selected for the GIGE and SONET/SDH OC48 modes should be different.

For example, if you select **1** for the **What is the logical reference index?** option for the SONET/SDH OC48 mode, you should select **0** for GIGE mode. If you select the same values for the two modes, the transceiver behavior after reconfiguration becomes unpredictable.

9. Select the **Channel interface** option. Selecting **Channel interface** creates the data interface signals `tx_datainfull` and `rx_dataoutfull` that are comprised of control and data signals. This selection is required because of the differences in the PLD interface width between the GIGE and SONET/SDH modes (row 1 in Table 3–5). The description of individual bits of `tx_datainfull` and `rx_datainfull` are provided in "Channel Interface" on page 3–53.

10. In the **Reconfig2** tab, under the **How should the receivers be clocked?** option, check the **Use the respective channel core clocks** option. Selecting this option creates the `rx_clkout` port. Select this option because of the clocking differences between the two modes (row 5 of Table 3–5). Therefore, the PLD logic can clock the receive output of the ALT2GXB with `rx_clkout` for SONET/SDH mode and `tx_clkout` for the GIGE mode.

11. In the **How should the transmitters be clocked?** option, select any option. Since this example assumes a one-channel reconfiguration in the transceiver block, the above options will not make a difference. However, if the number of channels used in channel reconfiguration is more than one, Altera recommends you select the **share single transmitter core clock between transmitters** option to conserve clock routing resources.

12. Select signals in the **check a control box to use the corresponding control fields** option based on the requirements. The signals in this tab can be selected only if the **Channel interface** option is enabled in the **Reconfig** tab. For this example, select the signals `rx_byteorderalignstatus` and `rx_a1a2sizeout`, since these signals are required for SONET/SDH OC48 mode.

Some of the signals are meaningful only for the modes for which they are intended. For example, the `rx_byteorderalignstatus` signal is only meaningful in SONET/SDH OC48 mode. PLD logic should not use these signals for GIGE mode.

For more information about the protocol-specific ALT2GXB interface signals, refer to "Stratix II GX ALT2GXB Ports List" on page 2–2 in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

13. In the subsequent tabs, select the required signals and complete the MegaWizard instantiation.

Step 2—Generate the ALT2GXB_RECONFIG Instantiation:

1. Set the **What is number of channels controlled by the reconfig controller?** option to **1**.

2. Select **Analog controls** to modify the PMA values, if desired.

3. Select **Channel reconfiguration**. This selection is required to perform a channel reconfiguration.

4. Select the required signals under the **write control** and **read control** options, if the **Analog controls** option in screen 1 is selected.

Refer to the *ALT2GXB_RECONFIG Megafunction User Guide* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about write-control and read-control signals.

5. In the **channel reconfiguration** page, select the `reconfig_address_out`. This signal increments by 1, from 0 to 27, then starts at zero again. The other available control signals in the MegaWizard are `reconfig_address_en` and `reset_reconfig_address`. Selecting these signals is optional. The timing and description of these signals are provided in "Section II—Control Logic for the RECONFIG Controller:" on page 3–84.

6. Complete the ALT2GXB_RECONFIG MegaWizard instantiation.

Step 3—Create a Top-Level Design and Generate the MIF for GIGE Protocol Mode:

Clock input connections for the ALT2GXB megafunction are listed below. The clock source should feed the following clock inputs:

- GIGE mode—`pll_inclk` and `rx_cruclk` inputs.
- SONET/SDH OC48 mode—`pll_inclk_alt` and `rx_cruclk_alt` inputs.

1. Since GIGE is the protocol mode you selected in the first page of the ALT2GXB MegaWizard, the Quartus II software requires the GIGE clock source to be connected to `pll_inclk` and `rx_cruclk` inputs.

2. Connect the `cal_blk_clock` input of the ALT2GXB instance to a clock source.

Refer to the "Stratix II GX ALT2GXB Ports List" on page 2–2 in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for the `cal_block_clk` signal requirements.

3. Connect the `tx_dataout` and `rx_datain` ports to the top-level module. This is required for the Quartus II software to compile successfully. To generate the MIF, connecting the other input and output ports of the ALT2GXB instance is not mandatory.

4. Assign pins for the clock ports (`pll_inclk`, `rx_cruclk`, `pll_inclk_alt`, and `rx_cruclk_alt`). If pin assignments are not made for the `tx_dataout` and `rx_datain` ports of the ALT2GXB instantiation, the Quartus II software automatically selects pins for these ports and names the MIF with the pin name extension. The MIF can still be used by any physical transceiver channel to perform reconfiguration.

After compilation of the design, the Quartus II software creates the MIF in the **reconfig_mif** folder under the project directory. Copy the MIF and save it in a separate folder. Otherwise, the new MIF that is generated for the SONET/SDH mode will overwrite the current MIF.

Step 4—Modify the ALT2GXB Instantiation for SONET/SDH OC48 Mode:

1. To create a MIF for the SONET/SDH OC48 mode, either modify the existing ALT2GXB instantiation created for GIGE mode or create a new instantiation for SONET mode. However, the first method is easier since it does not require major RTL or schematic changes.

2. Open the existing ALT2GXB instantiation. Select the **Which protocol you will be using?** option and set it to **SONET/SDH**. Set the **sub protocol** option to **OC48**. All the other signals selected for GIGE mode should not be changed.

3. In the **Reconfig** tab, select the **Channel internals** and **alternate reference clock** options. In the protocol section, select **GIGE**. Select the same input clock frequency selected in step 1.

4. For the **logical reference clock index** option, choose the complement of what you selected in step 10.

5. Select the channel interface and complete the instantiation.

Step 5—Generate the MIF for SONET/SDH OC48 Mode:

1. Before compiling the design, in the RTL or schematic, connect the `pll_inclk` and `rx_cruclk` to the clock source that provides the SONET/SDH OC48 clock. Similarly, connect `pll_inclk_alt` and `rx_cruclk_alt` to the clock source that provides the GIGE clock.

The Quartus II software generates the new MIF in the **/reconfig_mif** directory.

Step 6—Initialize Two Memory Elements with the MIF Contents and Write Logic to Select the MIF and to Control the ALT2GXB_RECONFIG:

1. Create two memory elements, each 16-wide and 28-bits deep. The memory elements can be a RAM/ROM inside or outside the Stratix II GX device. Assign the two MIFs to each of these memory elements.

Section II—Control Logic for the RECONFIG Controller:
The control logic block is required to perform the following functions:

- Select the memory to configure a channel to the GIGE or SONET/SDH mode.
- Control the reconfiguration mode (namely the PMA mode or the channel configuration mode).
- Control the read and write signals to the ALT2GXB_RECONFIG megafunction based on the `busy`, `data valid`, and `address_out` signals.

The following is an example flow of channel reconfiguration by writing the MIF contents of memory location 1 to the reconfig controller:

1. Set the `reconfig_mode_sel` to **-001**.
2. Select the data input from memory location 1.
3. Wait until the `busy` signal from the ALT2GXB_RECONFIG megafunction is low.
4. Check whether the `reconfig_address_out` is less than 28 decimals.
5. Wait for the data out from memory location 1 corresponding to the new `reconfig_address_out` becomes available at the `reconfig_data` port before asserting the `write_all` signal.

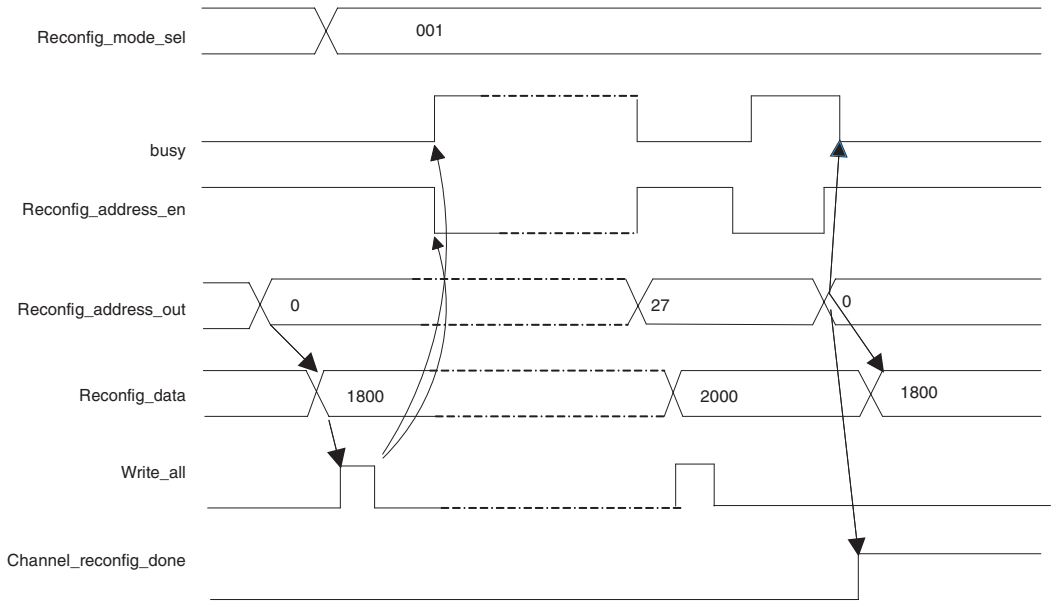
Figure 3–36 shows the various signal transitions during channel reconfiguration.

The ALT2GXB_RECONFIG megafunction provides these additional signals:

- `reconfig_address_en`
- `reset_reconfig_address`

The ALT2GXB_RECONFIG megafunction asserts the `reconfig_address_en` signal to indicate that the `reconfig_address_out` has changed. In test designs, the `reconfig_address_out` was monitored to determine the end of the write operation. You can also use the `channel_reconfig_done` signal to determine the end of the write operation. The `channel_reconfig_done` signal goes high one clock cycle after the `reconfig_address_out` signal changes from 27 back to 0.

Figure 3–36. Timing of the ALT2GXB_RECONFIG Signals



Section III—Logic and Clocking for the GIGE and SONET/SDH OC48 Datapath:

In the ALT2GXB MegaWizard, the channel interface that created `tx_datainfull` (44-bits wide) and `rx_dataoutfull` (64-bits wide) was selected. In addition, the `rx_byteorderalignstatus` and the `rx_a1a2size` signals were selected. The PLD logic should selectively use some of these signals based on whether the transceiver channel is configured in GIGE mode or SONET/SDH OC48 mode.

Table 3–6 provides descriptions for the tx_datainfull and rx_dataoutfull signals for GIGE and SONET/SDH OC48 modes.

Table 3–6. PLD Interface Signals—GIGE and SONET/SDH OC48 Modes

Signal Name	Description
GIGE MODE	
tx_datainfull[7:0]	8-bit unencoded data input to the transceiver channel
tx_datainfull[8]	tx_ctrlenable (control signal K/D)
rx_dataoutfull[7:0]	8-bit unencoded data output from the transceiver channel
rx_dataoutfull[8]	rx_ctrldetect (control signal K/D)
rx_dataoutfull[9]	rx_errdetect
rx_dataoutfull[10]	rx_syncstatus
rx_dataoutfull[11]	rx_disperr
rx_dataoutfull[12]	rx_patterndetect
SONET/SDH OC48 MODE	
tx_datainfull[7:0]	LSB data input to the transceiver channel
tx_datainfull[29:22]	MSB data input to the transceiver channel
rx_dataoutfull[7:0]	LSB data output from the transceiver channel
rx_dataoutfull[29:22	MSB data output from the transceiver channel
rx_dataoutfull[10], rx_dataoutfull[42]	rx_syncstatus[1:0]
rx_dataoutfull[12], rx_dataoutfull[44]	rx_patterndetect[1:0]

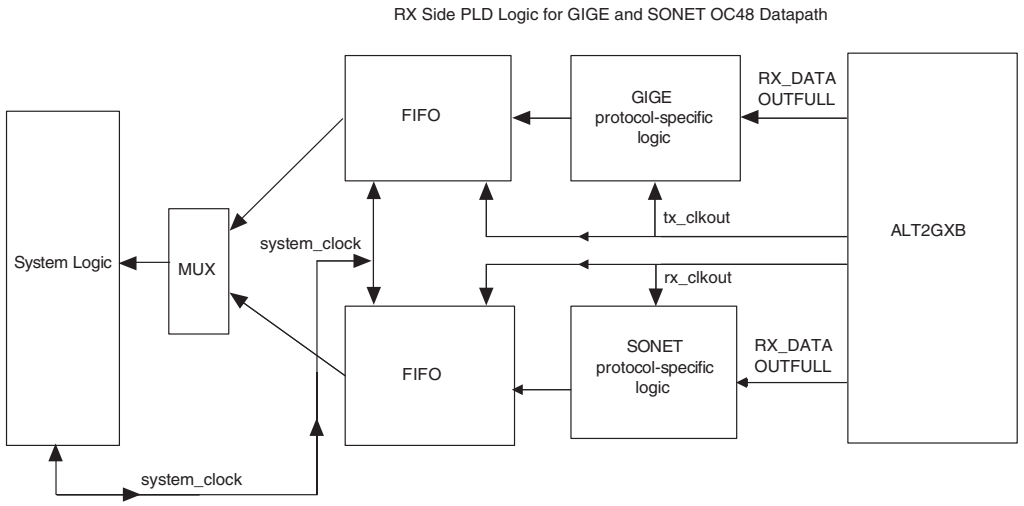
Clocking

For the transmit side, the PLD logic for the SONET/SDH OC48 and GIGE modes sends the data synchronized to the tx_clkout signal. Therefore, the clocking for the transmit side remains the same for the two modes.

For the receive side, the data and status signals from the ALT2GXB megafunction for the GIGE mode is synchronized to tx_clkout since rate matching is used. For the SONET/SDH OC48 mode, the signals are synchronized to rx_clkout. Therefore, the PLD logic has two functional protocol-specific logic blocks to handle data for the GIGE and SONET/SDH OC48 modes. Based on the configured protocol mode, the receive side logic selects the appropriate data path.

Figure 3–37 shows the block diagram for a receive side PLD logic to handle the GIGE and SONET/SDH OC48 datapath.

Figure 3–37. RX PLD Logic to Process GIGE and SONET/SDH OC 48 Data



Section IV—Reset Control Logic:
The reset control sequence for channel reconfiguration (explained in "Reset Recommendations" on page 3–66) must be followed during and after the channel configuration process. In addition, when resetting the transceiver channel, the reset control logic should reset the data path in the PLD logic to clear the error data received during the reconfiguration process.

For a PMA-only configuration (for example, changing the V_{OD}, equalization, DC gain, or pre-emphasis), the transceiver channel or the datapath in the PLD logic does not require a reset after reconfiguration. Reset is required only for channel reconfiguration or rate switch.

Simulation:
To simulate channel reconfiguration, some simulation tools only allow **.ram** or **.hex** files to initialize the memory. To convert the generated MIF to a **.hex** file, open the **.mif** in the Quartus II software and save it as a **.hex** file. Initialize the memory elements with the **.hex** file to simulate the design.

Example 2—Channel Configuration Between a Basic Mode Configured for 3.125 Gbps and a Basic Mode Configured for 2.000 Gbps

The PCS functional blocks and the PLD interface (16 bits) is the same for both modes. Given that the functional blocks are the same, to achieve the two data rates mentioned above, use two different input reference clock frequencies for the two modes. Table 3–7 shows the transceiver configuration for the two modes.

Table 3–7. Differences Between Two Basic Modes

Number	1 Channel Basic Mode 1	1 Channel Basic Mode 2
Data rate	3125	2000
Input reference clock frequency	156.25 MHz	125 MHz
PLD interface width	16	16
8B/10B enabled	Yes	Yes
Rate matcher	No	No
Clock used for synchronizing the receive output data (`rx_dataout`)	`rx_clkout`	`rx_clkout`

The description for this design example is divided into four sections:

- Section I—Steps to Create the MIF for the Two Transceiver Modes
- Section II—Steps to Create the ALT2GXB_RECONFIG Instantiation
- Section III—Sets Up Control Logic in the PLD for ALT2GXB_RECONFIG
- Section IV—Resets Control Logic

Section I—Steps to Create the MIF for the Two Transceiver Modes:

1. In the first page of the ALT2GXB MegaWizard, complete the following:
   - Set Mode to **Basic**
   - Select **single width**
   - Set the channel width to **16**
   - Set the number of channels to **1**
   - Set the data rate to **3.125 Gbps**.
2. Set the input reference clock to **156.25 MHz**.
3. Select all the resets, `pll_locked`, `rx_freqlocked`, and other required status signals.

4. In the **Reconfig** tab, select **Analog controls** if you wish to modify the PMA values dynamically (V_{OD}, pre-emphasis, DC gain, and equalization).

5. Select the **Channel internals** and **Use Alternate reference clock** options. These options must be selected since Basic mode 2 requires a different input clock frequency.

6. Set the **protocol** option to **Basic**, set the data rate to **2.000 Gbps**, and set the **input clock frequency** option to **125 MHz**. For this example, since both the configurations have the same PLD interface width and functional blocks (Table 3–7), you do not need the **Channel interface** option.

7. In the **Reconfig 2** tab, select the **Use respective receiver core clocks** and **Use respective transmitter core clocks** options (row 6, Table 3–7).

8. Complete the ALT2GXB MegaWizard instantiation.

Section II—Control Logic for the RECONFIG Controller:
Follow the same procedure as described in "Example 1—Configuring a Transceiver Channel Between GIGE Mode and SONET/SDH OC48 Mode" on page 3–72.

Section III—Logic for the Basic Mode 1 and Basic Mode 2 Datapath:
Clock the data to the transceiver with the `tx_clkout` signal on the transmitter side for both configurations. Similarly, for the receive side, clock the data from the transceiver with the `rx_clkout` signal for both configurations. A single logic block can handle data processing when the transceiver channel is configured between these two modes.

Section IV—Reset and Control Logic:
Follow the same procedure as described in "Example 1—Configuring a Transceiver Channel Between GIGE Mode and SONET/SDH OC48 Mode" on page 3–72.

Example 3: Dynamic Rate Switch for the Transmit Side Design Example

This design example explains the steps to dynamically divide the transmit data rate of a transceiver channel by 4, 2, or 1 without requiring MIF generation. The ALT2GXB_RECONFIG instance provides a `rate_switch_ctrl` signal for this purpose.

Use the rate_switch_ctrl signal only for dividing the data rate of the transmit side. To divide the data rate for both the transmit and receive sides, a MIF-based approach is required.

This example uses a Basic mode with 8B/10B enabled running at 4 Gbps data rate. You can configure the mode dynamically between 4.25 Gbps, 2.25 Gbps, and 1.125 Gbps.

The description for this design example is divided into the following sections:

- Section I—Create the ALT2GXB Instantiation for the Transceiver Channel
- Section II—Create the ALT2GXB_RECONFIG Instantiation
- Section III—Create the Top-Level Design

Section I—Create the ALT2GXB Instantiation for the Transceiver Channel:

1. Create a **Basic mode** by setting the operation mode to **Transmit and Receive**.

2. Select **double width** mode. This is required since the highest data rate in this example is 4.25 Gbps (**Single width** can be selected only up to 3.125 Gbps). Set the channel width to **32**. The lowest PLD frequency allowed in the Quartus II software is 25 MHz. Therefore, the transceiver runs at 1.125 Gbps with a 32-bit PLD interface. The PLD clock frequency in this case is 26.5 MHz (1125/40 = 26.5).

3. Set the input frequency to **106.25 MHz**.

4. In the **Reconfig** tab, check the **Channel internals** option. This is required to enable the ALT2GXB_RECONFIG instance to modify the channel local divider values dynamically. The alternate reference clock is not required since one clock source is used. Also, the data rates can be derived from the 106.25 MHz clock.

5. Complete the ALT2GXB MegaWizard instantiation.

Section II—Create the ALT2GXB_RECONFIG Instantiation:

1. Instantiate the ALT2GXB_RECONFIG megafunction as described in "Example 1—Configuring a Transceiver Channel Between GIGE Mode and SONET/SDH OC48 Mode" on page 3–72.

2. Select the **Modify the data rate using the local divider** option in the **Reconfiguration Settings** tab. This creates the rate_switch_ctrl and rate_switch_ctrl_out signals.

Table 3–8 shows the values for each of the rate_switch_ctrl settings.

Table 3–8. Rate Switch Control Signal Settings

rate_switch_ctrl[1:0] Settings	Local Divider Value in the Transmit Channel
00	1
01	2
10	4
11	Not applicable

3. Complete the ALT2GXB_RECONFIG MegaWizard instantiation.

Section III: Create the Top-Level Design
Create the ALT2GXB_RECONFIG instance control logic, reset control logic, and the PLD logic to handle the data path. Refer to "Reset Recommendations" on page 3–66 for information on transceiver resets.

4. Create the top-level design and connect the functional blocks.

Pseudo-Write Sequence for Simulating Channel Reconfiguration

If you are simulating channel reconfiguration, consider a case where you are using multiple transceiver channels in your design driven by a single dynamic reconfiguration controller. When you first perform channel reconfiguration on a transceiver channel, rx_freqlocked and rx_clkout of all channels that are connected to the reconfiguration controller go to 0 for a few clock cycles. This occurs because the receive PLLs in the simulation model require a relock when channel reconfiguration is enabled.

This issue happens only in simulation the first time you initiate channel reconfiguration.

To work around this issue, perform the following one-time write sequence as part of your system initialization when you assert the gxb_powerdown or rx_analogreset signals.

The signals that are referred to in the following write sequence correspond to the input and output ports of the ALT2GXB_RECONFIG instantiation in your design.

1. Set the reconfig_mode_sel signal to **001**. Write the default **.hex**/**.mif** file contents for two reconfig_address_out signal increments. That is, pulse the write_all signal for the reconfig_address_out 0 and 1 based on the busy and reconfig_address_en signals.

2. The **.hex**/**.mif** file selected for writing should correspond to the default configuration in the ALT2GXB MegaWizard. For example, if you have two **.hex**/**.mif** files that correspond to GIGE and SONET/SDH OC48 protocols, and if you have set GIGE as your default configuration (the protocol set in the **General** tab of the ALT2GXB MegaWizard), write the first two words of the **.hex**/**.mif** file generated for GIGE protocol.

3. After you complete writing the first two words, wait for the busy signal to go low and assert the reset_reconfig_address signal to initialize the reconfig_address_out to 0.

Channel and Clock Multiplier Unit (CMU) PLL Reconfiguration

Introduction

The Stratix II GX transceiver can be dynamically reconfigured to various protocols and data rates. This section discusses the dynamic reconfiguration features introduced in Quartus II software version 7.1. Altera assumes you have prior knowledge about the dynamic reconfiguration controller architecture (refer to "Dynamic Reconfiguration Controller Architecture" on page 3–2), the Stratix II GX transceiver architecture, and the memory initialization file (**.mif**, also known as MIF) flow (refer to "MIF Generation in Quartus II Software" on page 3–32) for dynamic reconfiguration.

Synopsis of Existing Dynamic Reconfiguration Features

In the Quartus II software version 7.0 and earlier, the following dynamic reconfiguration features were available:

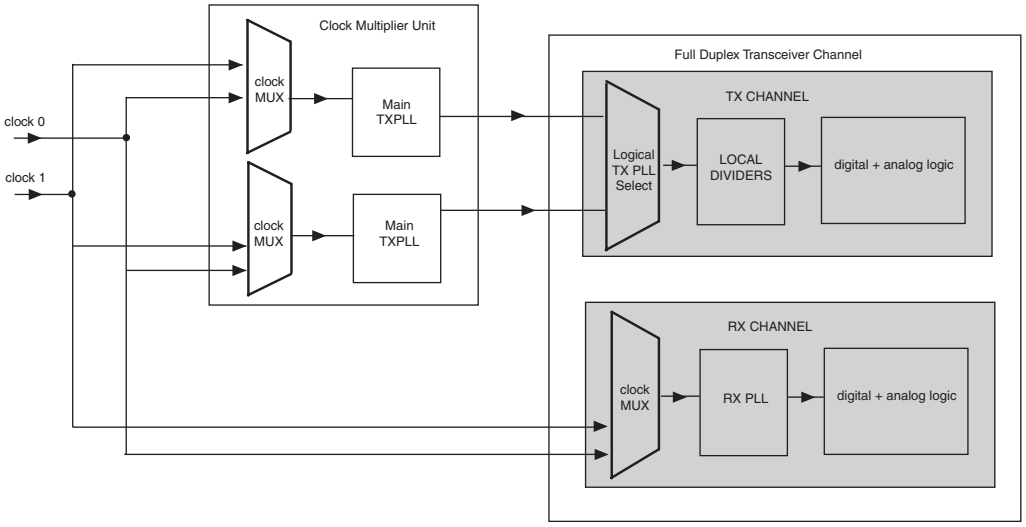
- PMA reconfiguration—to control voltage output differential (V_{OD}), pre-emphasis, equalization, and DC gain.
- Channel reconfiguration—to dynamically reconfigure the transceiver data rates, protocol modes, or a combination of these two options. This method requires a MIF to reconfigure a channel.
- Dynamic transmit rate switch—to dynamically reconfigure the transmit data rate by changing the local divider settings in the transmit side. The available local divider options are by /1, /2, or /4. This method does not require a MIF for reconfiguration.

This option can only change the transmitter data rate and not the receiver data rate.

Figure 3–38 shows the transceiver blocks (grayed out) that can be dynamically reconfigured using the channel reconfiguration feature.

Figure 3–38 shows that the TX PLLs and the clock multiplexer on the transmit side (inside the clock multiplier unit) could not be reconfigured using these features. It also shows that only two sources of input reference clocks were available for the TX PLLs and RX PLLs.

Figure 3–38. Reconfigured Functional Blocks with Channel Reconfiguration *Note (1)*



Note to Figure 3–38:
(1) Supported from the Quartus II software version 6.1.

Overview of Quartus II Software Version 7.1 Features for Dynamic Reconfiguration

The Quartus II software version 7.1 provides the following enhancements to support dynamic reconfiguration:

- Three additional features to dynamically reconfigure the transceiver channel and the TX PLLs:
  - TX PLL-only reconfiguration
  - Channel and TX PLL reconfiguration
  - Channel reconfiguration with TX PLL select
- The number of possible clock sources for the input reference clocks is increased from two to five.

- Settings to generate a MIF with 38 words (required for the features mentioned above). You will need this MIF just as you needed one in the channel reconfiguration feature (see "Channel Reconfiguration" on page 3–30).

To write the MIF, follow the same method used for channel reconfiguration (see "Channel Reconfiguration" on page 3–30 for more information). The functionality of all other signals, such as `write_all`, `channel_reconfig_done`, `reconfig_address_en`, `logical_channel_address`, `data_valid`, and `busy` is the same as that of the channel reconfiguration feature.

These new features provide flexibility to reconfigure a TX PLL to multiple data rates, dynamically switch the transmit channel to listen to any of the two TX PLLs, and to configure a transceiver channel. Using these enhancements, you can use the Stratix II GX transceiver to dynamically support multiple protocols and data rates. In the following sections, the new dynamic reconfiguration features and the different software settings required to implement these features are discussed in detail.

Conventions Used

Throughout this document, the following conventions are used:

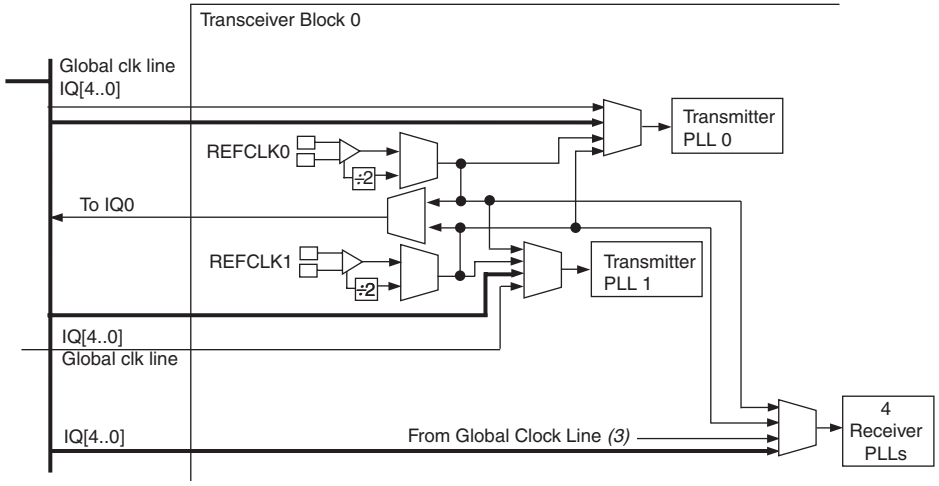
- Channel and CMU PLL reconfiguration—refers to the three dynamic reconfiguration features introduced in the Quartus II software version 7.1.
- Channel and TX PLL reconfiguration—refers to one of the features in Channel and CMU PLL reconfiguration.
- Channel—refers to the transceiver channel with digital and analog functional blocks.
- Main TXPLL— refers to the TX PLL that is configured in the **General** tab of the ALT2GXB MegaWizard.
- Alternate TXPLL—refers to the TX PLL that is configured in the **Reconfig Alt PLL** tab of the ALT2GXB MegaWizard.
- logical tx pll—refers to the logical identification value 0 or 1, assigned to the main and alternate TXPLLs. You set this value in the **Reconfig Clks 1** and **Reconfig Alt PLL** tabs of the ALT2GXB MegaWizard.
- Reconfig controller—refers to the dynamic reconfiguration controller that you instantiate using the ALT2GXB_RECONFIG MegaWizard. In this document, reconfig controller and ALT2GXB_RECONFIG are used interchangeably.

Clocking Enhancements and Requirements

To configure the TX PLLs and RX PLLs for multiple data rates, it is important to understand the input reference clock requirements. This helps you to efficiently create the clocking scheme for reconfiguration and to reuse the MIFs across all channels in the device. The new clocking enhancements and the implications of using input clocks from various clock sources are reviewed in this section.

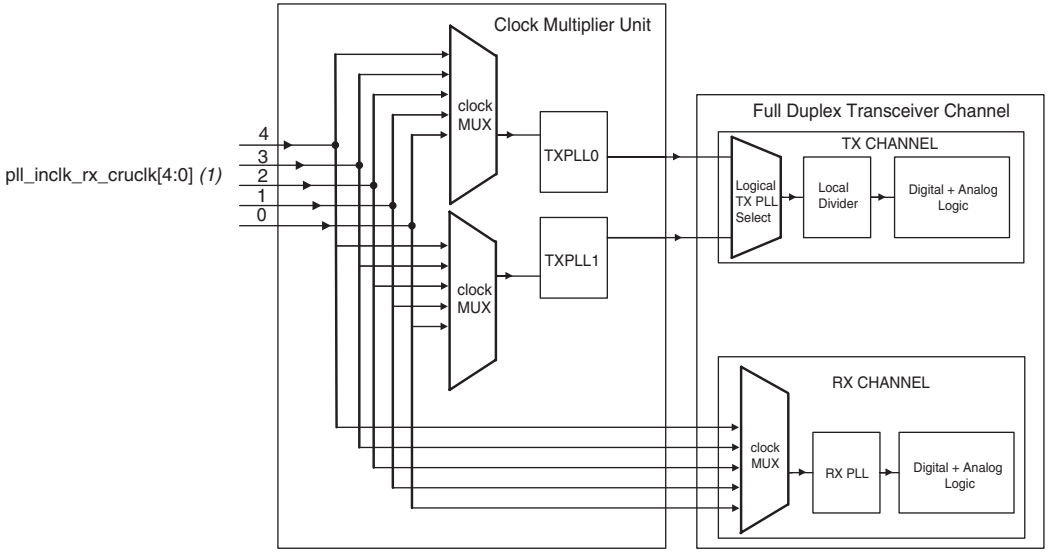
When you enable the **Channel and CMU PLL Reconfiguration** option in the ALT2GXB MegaWizard (by selecting the **Enable Channel and Transmitter PLL Reconfiguration** option in the **Reconfig** tab), the Quartus II software version 7.1 allows a maximum of five possible sources available for input reference clocks. Figure 3–39 shows the different clock sources that connect to the transceiver block.

Figure 3–39. Transceiver Block with Global Clock Line Connections



These five clock inputs appear as a `pll_inclk_rx_cruclk[]` port and can be provided from the inter transceiver block lines, also referred as Inter Quad (IQ) lines, or from the global clock networks that are driven by an input pin. Figure 3–40 shows the reference clock connections to TX PLLs and RX PLLs in a transceiver channel.

Figure 3–40. Input Reference Clock Connections to the Transceiver Channel



Note to Figure 3–40:
(1) These clocks can be provided from IQ lines, global clock networks, or dedicated local `refclks`.

Figure 3–40 shows the same input reference clocks connected to both the TX PLLs and RX PLLs. If you enable the **Channel and CMU PLL Reconfiguration** option for a full-duplex configuration, you cannot provide separate reference clocks to the TX PLLs and RX PLLs.

When you use the global clock line to provide input reference clocks, be aware of the following restrictions and implications:

- The hardware allows only one global clock input for the two TX PLLs in a transceiver block (refer to Figure 3–39).
- In a receiver-only channel configuration, the RX PLL of each channel in a transceiver block can be clocked by an independent global clock line. But, if you connect different clock input pins to the RX PLL in each channel, you cannot reuse the MIFs between these two channels. This constraint is explained further in "Input Reference Clock Requirements for Reusing MIFs" on page 3–94.
- Each global clock line consumes a local route input output (LRIO) resource. Since each transceiver block has a fixed LRIO resource, using the global clock line may restrict the number of clocks you can provide to the transceiver channels in your design.

For more information on LRIO resource limitation, refer to the PLD Clock Resource section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Using Dedicated refclks

When you use dedicated `refclks` as input reference clocks, the `refclk` pre-divider is required if **one** of the following conditions is satisfied:

1. If the input clock frequency is greater than 325 MHz.
2. For functional modes with a data rate less than 3.125 Gbps (the data rate is specified in the **what is the data rate?** option in the **General** tab of the ALT2GXB MegaWizard). The TXPLL and RXPLL is configured with the data rate that you set in this option.
   - If the input clock frequency is greater than or equal to 100 MHz AND
   - If the ratio of data rate to input clock frequency is 4, 5, or 25
3. For functional modes with an data rate greater than 3.125 Gbps:
   - If the input clock frequency is greater than or equal to 100 MHz AND
   - If the ratio of data rate to input clock frequency is 8, 10, or 25

When you use the channel and CMU PLL reconfiguration feature, you can dynamically reconfigure the TX PLLs and the RX PLL in a transceiver block from any of the five available input reference clocks. The Quartus II software automatically instantiates the `refclk` pre-divider (if one of the above mentioned conditions is satisfied) for the clock sources that drive the main and alternate TX PLL.

You specify the information for main and alternate TX PLL in the **General** and **Reconfig Alt PLL** tabs, respectively.

For the other clock inputs, the ALT2GXB MegaWizard provides optional options in the **Reconfig Clks 1** and **Reconfig Clks 2** tabs to specify whether a `refclk` pre-divider should be instantiated by the Quartus II software. The available options are:

a. **what is the reconfig protocol driven by clock x?** ("x" can be 0, 1, 2, 3, 4)

b. **what is clock x input frequency?**

c. **use clock x reference clock divider?**

For specific option values in **a** and **b** (mentioned above), the Quartus II software automatically instantiates the `refclk` pre-divider (field **c** is automatically selected in the ALT2GXB MegaWizard). For example, when you select the **PCI Express (PIPE)** option in the **what is the reconfig protocol driven by clock0** field, the Quartus II software automatically instantiates the `refclk` pre-divider for clock source 0.

When you select values in **a** and **b** (mentioned above) for a clock source and if field **c** is enabled, determine whether the data rate or the input clock frequency for the clock input meets one of the three conditions mentioned above. If one of the conditions is met, select field **c**.

For example, if you intend to use clock `source1` to reconfigure the channel to Basic mode with a 100 MHz input reference clock and a data rate of 2500 MHz, clock `source1` satisfies condition 2 mentioned above. In the **Reconfig Clk 1** tab, set the following values for clock `source1`:

- set the **what is the reconfig protocol driven by clock1?** option to **Basic**
- set the **what is clock 1 input frequency?** option to **100 MHz**
- select the **use clock1 reference clock divider?** option (this enables the Quartus II software to instantiate the `refclk` pre-divider for clock `source1`).

You can also use a dedicated `refclk` input pin from an unused transceiver block. If the Quartus II software creates a pre-divider for this clock input, it automatically feeds the output of the pre-divider to the TX PLLs and RX PLLs of other transceiver blocks. The `refclks` do not use the LRIO resource.

Refer to Figure 2–4 in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for more information.

The `refclk` pre-divider is not part of the information stored in the MIF. It is a static setting created during the ALT2GXB MegaWizard configuration.

The above mentioned clocking scheme also determines whether you can reuse the MIF across all transceiver channels in your device. In the following section, the clocking requirements to reuse MIFs are discussed.

Input Reference Clock Requirements for Reusing MIFs

The MIF contains information about the input clock multiplexer and the functional blocks that you selected during the ALT2GXB MegaWizard instantiation. The Quartus II software generates a MIF for each channel. This MIF can be used in any of the other channels in the device if you satisfy the following two requirements for the input reference clocks:

- The order of the clock inputs must be consistent. For instance, assume that a MIF is generated for a transceiver channel in bank 13 and the clock source is connected to the `pll_inclk_rx_cruclk[0]` port. When the generated MIF is used in a channel in other transceiver blocks (for example, bank 14), the same clock source needs to be connected to the `pll_inclk_rx_cruclk[0]` port. Figures 3–41 and 3–42 show the incorrect and correct order of input reference clocks, respectively.
  - In Figure 3–41, the clocking is incorrect to reuse the MIF because the input reference clock is not connected to the corresponding `pll_inclk_rx_cruclk[]` ports in the two instances.

Figure 3–41. Incorrect Input Reference Clock Connection to Reuse the MIF

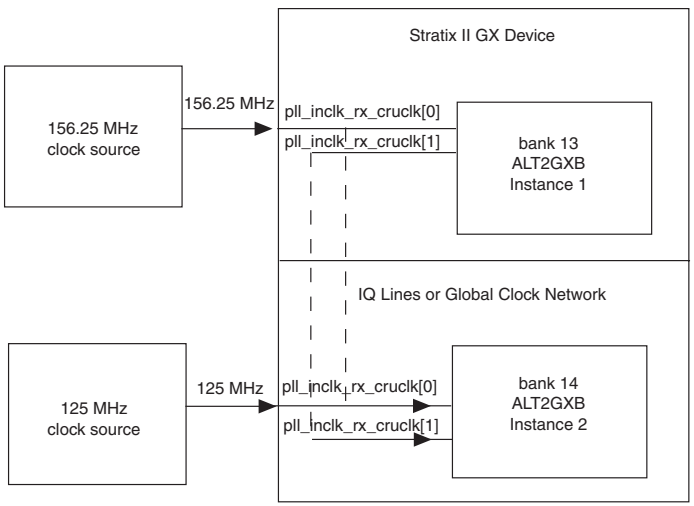
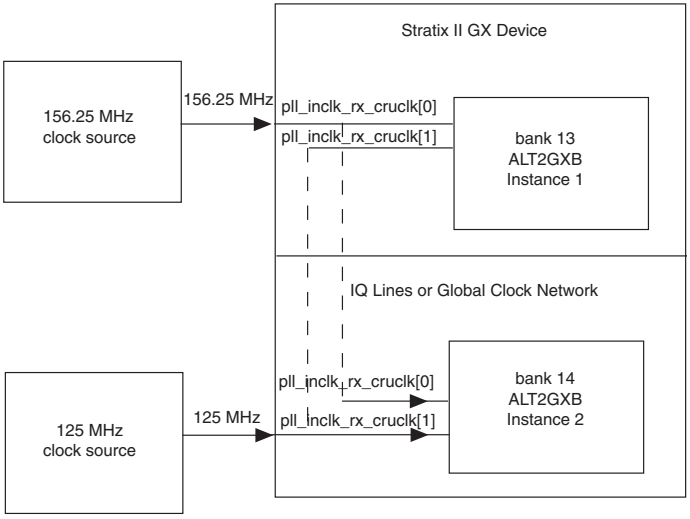


Figure 3–42. Correct Input Reference Clock Connection to Reuse the MIF



- If you connect the input reference clock ports of the ALT2GXB instances through different input pins, you cannot reuse the MIF generated between these two instances, even if you provide the same clock frequency on these two pins. For example, in Figure 3–43 the clock source provides 156.25 MHz clock to instance1 and instance2 through two different pins. In this case, if you generate a MIF for instance1, you cannot reuse it in instance2.
  - When you try to reconfigure using the MIF for instance1 in a transceiver block (for example, bank 13) on instance2 in another transceiver block (for example, bank 14), the reconfig controller remaps the clock input multiplexer information in the MIF (generated for instance1) to correspond to instance2. During this translation process, it assumes that the same clock input is connected to the `pll_inclk_rx_cruclk[]` port. Therefore, the reconfig controller selects the clock multiplexer value for the IQ line or global clock network that connects to the clock input of instance1.

If you want to reuse the MIF, connect the clock source to only one clock pin in the device. In your design, connect the clock input port of your transceiver instances to that clock pin. The Quartus II software automatically routes the clock input to all the transceiver blocks through IQ lines or global clock routing resources, depending on whether you

selected a dedicated refclk pin or a clock I/O pin. Figure 3–43 shows the incorrect clocking scheme. Figure 3–44 shows the correct clocking scheme.

Figure 3–43. Incorrect Clocking Scheme to Reuse MIF

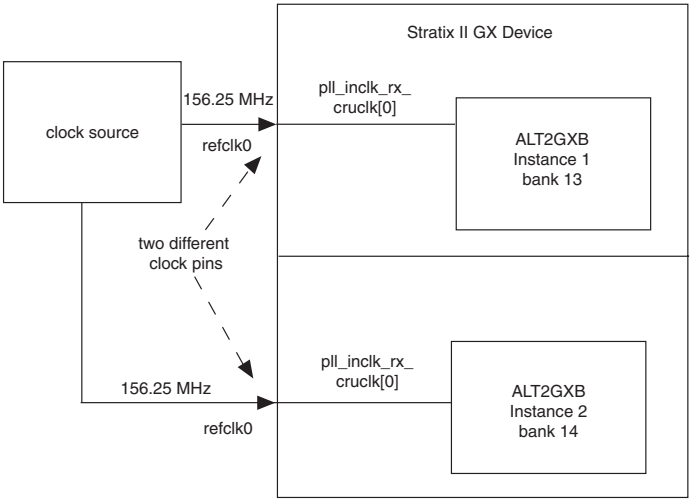
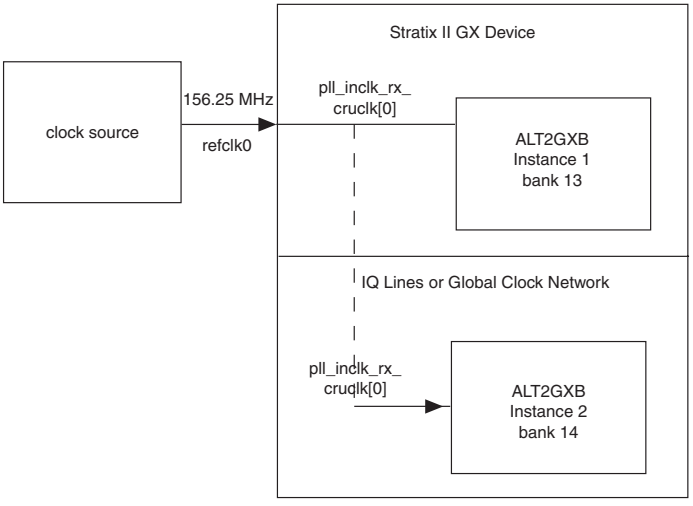


Figure 3–44. Correct Clocking Scheme for Reusing MIF



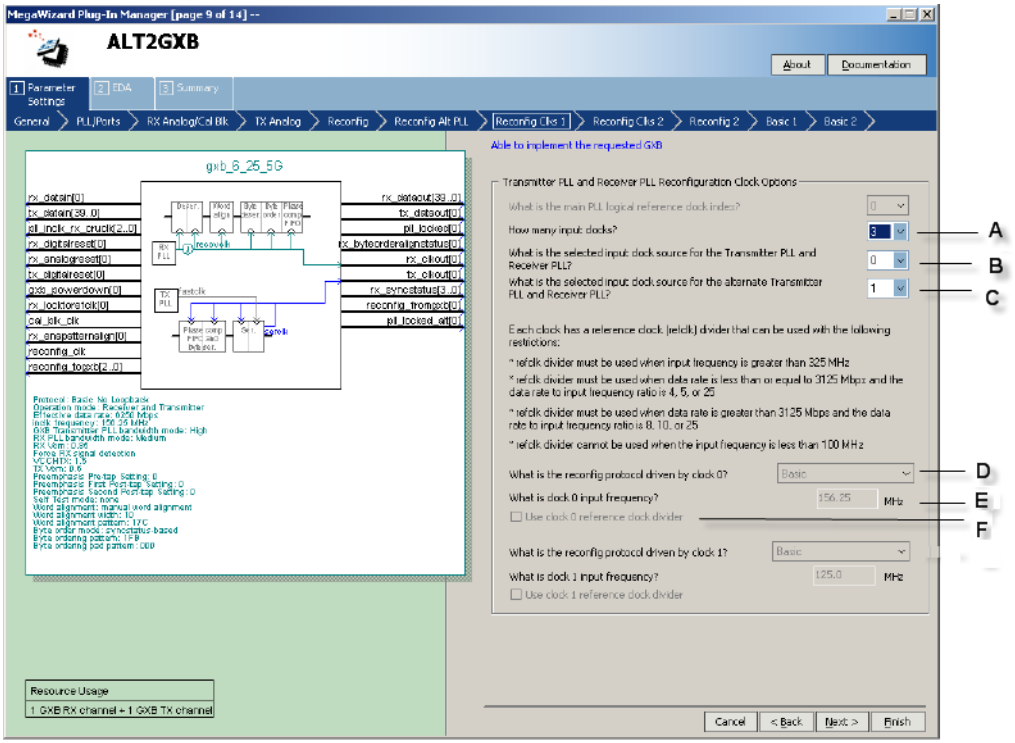
General Guidelines for Specifying the Input Reference Clocks

The following are general guidelines for your input reference clocks:

- Assign the identification numbers to all input reference clocks that are used in the design (0, 1, 2, 3, and 4). The identification numbers are indicated by "A" in Figure 3–45.
- Keep the identification numbering consistent for all the subsequent MIF configurations. Provide the identification numbers indicated by "B" and "C" in Figure 3–45.
- Maintain a consistent protocol, input reference clock frequencies, and reference clock pre-divider settings for all the MIF. Set these options indicated by "D", "E", and "F" in Figure 3–45.

These fields are explained in detail in "ALT2GXB MegaWizard Settings" on page 3–113.

Figure 3–45. Input Clock Settings of the Reconfig Clks 1 Tab



Logical TX PLL

In the channel reconfiguration feature ("Channel Reconfiguration" on page 3–30), you cannot reconfigure the TX PLLs dynamically. It was not necessary to know which of the two TX PLLs was configured for the specified data rate. You were required to set only the **logical reference index** option in the ALT2GXB MegaWizard to select the output clocks of the main and alternate TXPLLs and the **clock mux** in the RX PLL side (refer to Figure 3–40 for the location of these multiplexers).

When you use the **Channel and CMU PLL Reconfiguration** option, you can dynamically reconfigure the channel and the TX PLL. Therefore, to reconfigure the TX PLL during run time, you need the flexibility to select the TX PLL. When you enable this option, the ALT2GXB MegaWizard provides a logical identification for the main and alternate TXPLLs. This identification is referred to as the "logical tx pll" value. This value provides a logical identification to the TX PLL that is associated with a transceiver channel, without requiring the knowledge of its physical location.

In the ALT2GXB MegaWizard, when you provide the main TXPLL with a logical tx pll value, for example **0**, the alternate TXPLL automatically takes the complement value **1**. The logical tx pll value for the main TXPLL is stored along with the other transceiver channel information in the generated MIF. You can reuse the MIF generated for one TX PLL to reconfigure the other TX PLL in the same or in other transceiver blocks. The dynamic reconfig controller provides you an optional `logical_tx_pll_sel` port for this purpose. The method to use this port and reconfigure a TX PLL is explained in "Logical TX PLL" on page 3–98.

Using the Channel and CMU PLL Reconfiguration Feature

The channel and CMU PLL reconfiguration feature is divided into three categories based on the functionality:

- Channel and TX PLL reconfiguration
- TX PLL reconfiguration
- Channel reconfiguration with TX PLL select

You can select these features by setting the appropriate values in the reconfig_mode_sel port of the **ALT2GXB RECONFIG** tab. Table 3–9 shows the reconfig_mode_sel values for all the dynamic reconfiguration features.

Table 3–9. reconfig_mode_sel Values for All Dynamic Reconfiguration Modes

reconfig_mode_sel[2:0]	Description
000	PMA controls
001	Channel reconfiguration
010	Not supported (do not attempt to read or write with this value)
011	Dynamic Transmit rate switch
100	TX PLL
101	Channel and TX PLL reconfiguration
110	Channel Reconfiguration with TX PLL select
111	Not supported (do not attempt to read or write with this value)

The **read** operation is valid only for the reconfig_mode_sel value **000**. Do not use **read** for any other modes. Ensure that the reconfig_mode_sel port is set to the above mentioned supported values only. Setting the reconfig_mode_sel port to non-supported values may yield unpredictable transceiver behavior. When the **Enable Adaptive Equalization control** option is enabled, the reconfig_mode_sel port is 4-bits wide. In this case, set the most significant bit (MSB) of the reconfig_mode_sel to **0** when you use any of the above mentioned values.

As with the channel reconfiguration feature, the reconfig controller automatically increments the reconfig_address_out values to read the appropriate words from the MIF memory. Table 3–10 shows the address values incremented by the reconfig controller for the different features.

Table 3–10. Address Incremented by the Reconfig Controller (Part 1 of 2)

reconfig_mode_sel[2:0]	Incremented Address
001 (channel reconfiguration)	0-27
100 (TX PLL only)	0, 28-37

Table 3–10. Address Incremented by the Reconfig Controller (Part 2 of 2)

reconfig_mode_sel[2:0]	Incremented Address
101 (Channel and TX PLL reconfiguration)	0-37
110 (Channel reconfiguration with TX PLL select)	0-27

The procedure to write the MIF contents to the transceiver is the same as in with the channel reconfiguration feature (refer to Figure 3–8 on page 3–28).

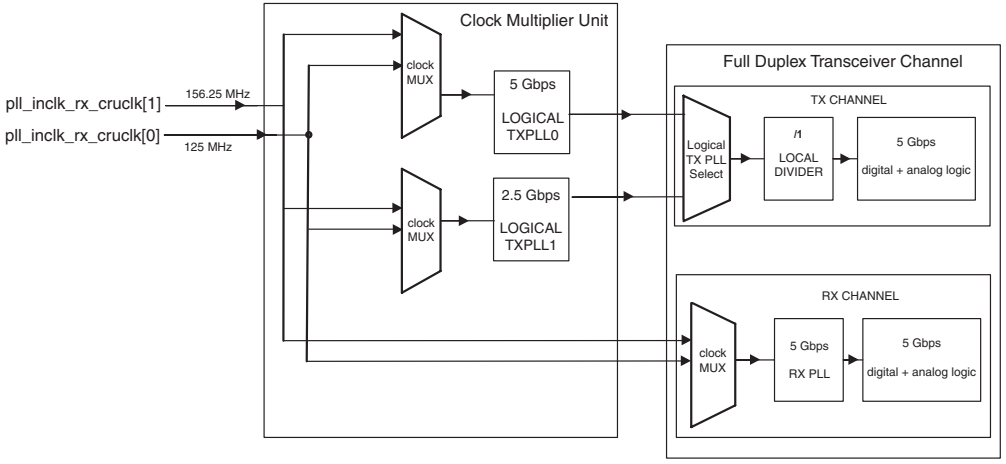
Channel and TX PLL Reconfiguration

The channel and TX PLL reconfiguration mode reconfigures the transceiver channel and the TX PLL that provide high-speed clocks to the transceiver channel. This mode helps to switch across multiple protocols that require different data rates and functional blocks. Since the TX PLL is also reconfigured when you use this feature, all the channels that are listening to the TX PLL are affected. To perform channel and TX PLL configuration, set the `reconfig_mode_sel` to **101** and write the MIF contents. During reconfiguration, the reconfig controller powers down the selected logical TX PLL until the new values are updated. The power down feature is explained in "TX PLL Powerdown" on page 3–109. To illustrate the functional blocks that are reconfigured, the following example is used. This same example is used for the three reconfiguration features.

Consider that you have an ALT2GXB instantiation with the following default configuration:

- The full-duplex channel with the main TXPLL configured to 5 Gbps data using a 156.25 MHz reference clock. The alternate TXPLL is configured to 2.5 Gbps using a 125 MHz reference clock (Figure 3–46 shows the default configuration).
- Assume that the logical tx pll value is set to **0** for the main TXPLL. (The settings to select the logical tx pll value for the TX PLLs are discussed in "ALT2GXB MegaWizard Settings" on page 3–113).

Figure 3–46. Transceiver Channel Default Configuration



Consider that you intend to switch to the following two modes:

mode1:

- Full-duplex channel with the main TXPLL configured to 6.25 Gbps data using a 156.25 MHz reference clock. Assume that the logical tx pll is set to **0** for the main TXPLL.
- Rate matcher is not enabled in the ALT2GXB megafunction.
- The alternate TXPLL is configured to 2.5 Gbps using a 125 MHz reference clock.

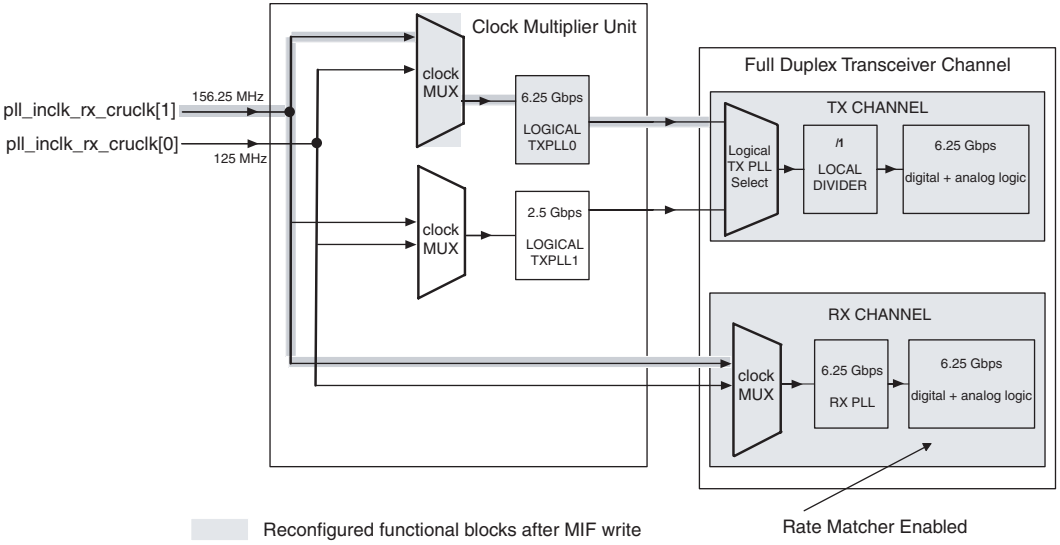
mode2:

- Full-duplex channel with the main TXPLL configured to 5 Gbps data using a 156.25 MHz reference clock. Assume that the logical tx pll is set to **0** for the main TXPLL.
- Rate matcher is enabled in the ALT2GXB megafunction.
- The alternate TXPLL is configured to 2.5 Gbps using a 125 MHz reference clock.

Consider that the MIF is generated for mode1 and mode2. (Details on the steps to generate the MIF are covered later in this section. The intent of this example is to show how the functional blocks are reconfigured based on the feature used).

Figure 3–47 shows the functional blocks that are reconfigured after the dynamic reconfig controller writes the mode2 MIF. Note that on the receive side, the rate matcher gets enabled after reconfiguration since the mode2 MIF contains settings to enable the rate matcher block.

Figure 3–47. Reconfigured Functional Blocks After the Channel and TX PLL Reconfiguration



TX PLL Reconfiguration

Using the TX PLL reconfiguration mode, you can reconfigure the TX PLLs. This mode is very useful in saving reconfiguration time in certain applications. Consider that you have four transmit-only instances in the same transceiver block that switch to different data rates together (assuming that the functional blocks are the same across the data rates). In this case, all these channels can listen to the same TX PLL. Instead of using the **Channel and TX PLL Reconfiguration** option (write 38 words) for individual channels, you can configure the TX PLL (write 10 words) once, to a different data rate. This, in turn, changes the transmit data rate of all the channels listening to this TX PLL.

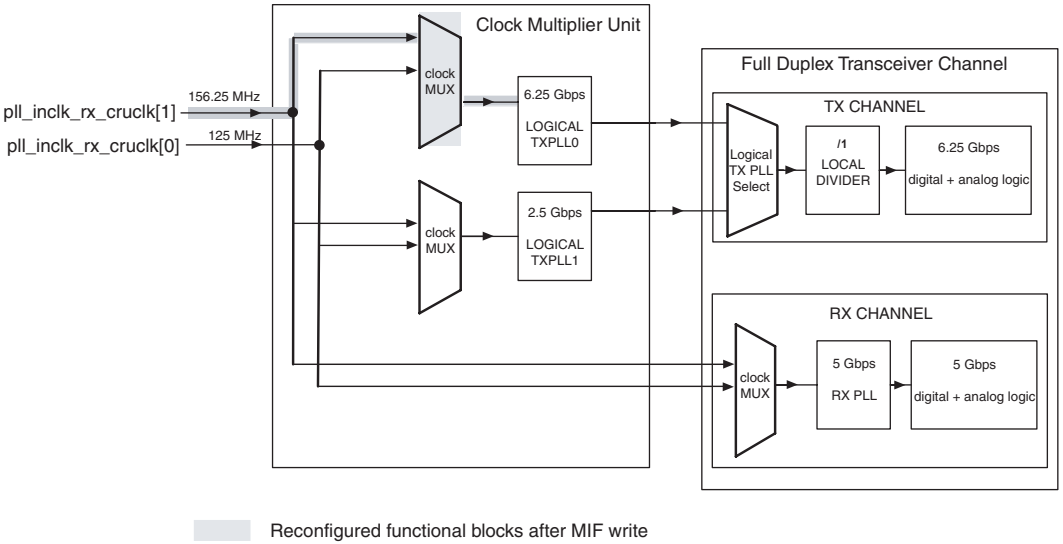
This mode is also useful when combined with the channel reconfiguration with TX PLL select mode. When the channel is listening to one TX PLL, you can reconfigure the other TX PLL, and later switch the transmit channel to listen to the configured TX PLL (explained in "Channel Reconfiguration with TX PLL Select" on page 3–103).

To perform TX PLL reconfiguration, set the `reconfig_mode_sel` value to **100** and write the MIF contents. The dynamic reconfig controller automatically increments the values on the `reconfig_address_out` port to word **0** and **28** through **37** from the specified MIF. The words 28-37 contain information to reconfigure the TX PLL. During reconfiguration, the reconfig controller powers down the TX PLL until the new values are written.

To understand the functional blocks that get reconfigured by this mode, consider the same example mentioned in "Channel and TX PLL Reconfiguration" on page 3–100. Figures 3–46 and 3–48 show the conditions before and after reconfiguration (using the mode1 MIF), respectively.

All the channels listening to the configured TX PLL are affected due to this reconfiguration.

Figure 3–48. Reconfigured Functional Blocks After TX PLL-Only Reconfiguration



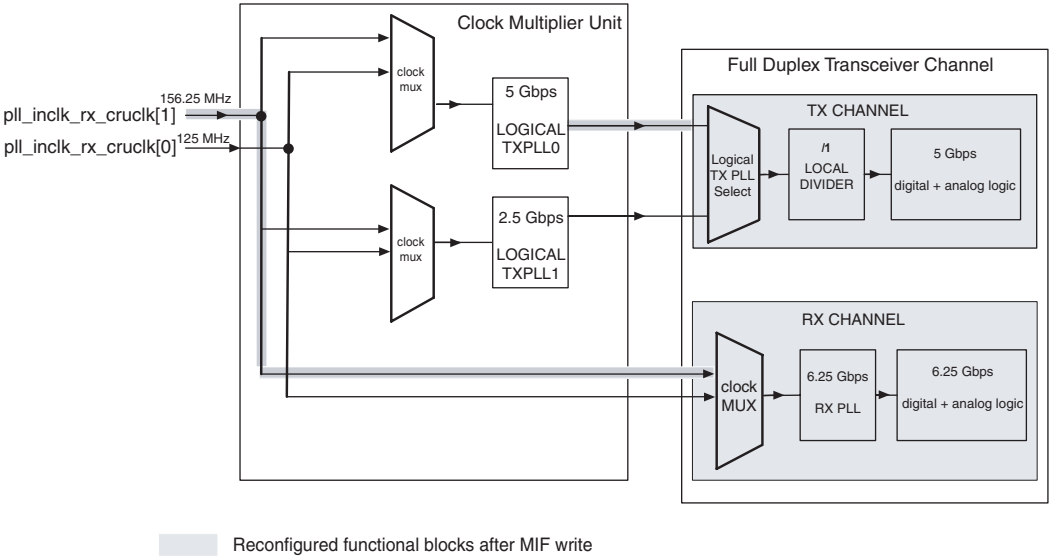
Channel Reconfiguration with TX PLL Select

This option reconfigures the channel and the logical TX PLL select multiplexer (shown in Figure 3–49) that selects the clock output from one of the TX PLLs. To use this feature, set the `reconfig_mode_sel` value to **110** and write the MIF contents. The **Channel Reconfiguration with TX PLL Select** option, in combination with the **TX PLL Reconfiguration**

option, is useful to switch the TX channel to multiple data rates. When the transmit side is using one TX PLL, reconfigure the second TX PLL using the TX PLL Reconfiguration feature. Then, use the channel reconfiguration with TX PLL select feature to switch the logical TX PLL select multiplexer to listen to the reconfigured TX PLL. This feature may require you to use the optional `logical_tx_pll_sel` port available in the **ALT2GXB RECONFIG** tab. The function of this port is explained in "Logical TX PLL" on page 3–98.

Consider the same example mentioned in "Channel and TX PLL Reconfiguration" on page 3–100 (refer to Figure 3–46 for conditions before reconfiguration). Figure 3–49 shows the reconfigured functional blocks after the mode1 MIF write is completed. Note that the TX PLLs are not reconfigured. Since the new MIF has the same functional blocks as the original configuration, there is no change in the functional blocks or the data rate in the transmit side after reconfiguration. Note that the MIF configures the RX PLL to 6.25 Gbps.

Figure 3–49. Reconfigured Functional Blocks After Channel and TX PLL Select Reconfiguration



Logical TX PLL Select

You can reuse the MIF created for one TX PLL on the other TX PLL using the optional `logical_tx_pll_sel` port in the ALT2GXB_RECONFIG MegaWizard. If the `logical_tx_pll_sel` port is enabled, the reconfig controller uses the value on this port irrespective of the logical tx pll value contained in the MIF. By using this port, you specify the identity of the TX PLL that you intend to reconfigure.

If you want to use the `logical_tx_pll_sel` only under some conditions and use the logical tx pll value stored in the MIF otherwise, enable an additional optional `logical_tx_pll_sel_en` port. If this port is enabled, the dynamic reconfig controller uses the value on the `logical_tx_pll_sel` port ONLY if the `logical_tx_pll_sel_en` port is set to **1** (refer to Figure 3–50). The values on these two ports should be held at a constant logic level until reconfiguration is completed. Table 3–11 shows the selected `logical_tx_pll` value under all the combinations of these two signals.

Figure 3–50. Effect of Using logical_tx_pll_sel and logical_tx_pll_sel_en Ports

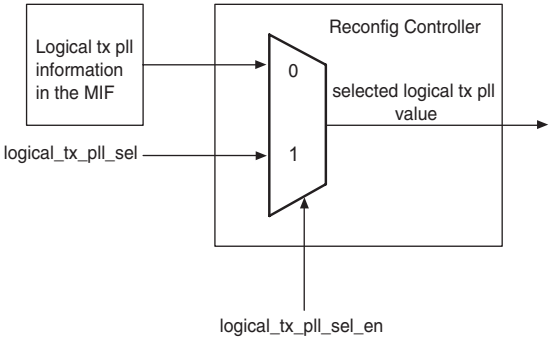


Table 3–11. logical_tx_pll_sel and logical_tx_pll_en Combinations (Part 1 of 2)

logical_tx_pll_sel Port	logical_tx_pll_se_en Port	Selected logical tx pll Value by the Reconfig Controller
enabled	enabled - value high	Value on the `logical_tx_pll_sel` port.
enabled	enabled - value zero	`logical tx pll` value stored in the MIF.

Table 3–11. logical_tx_pll_sel and logical_tx_pll_en Combinations (Part 2 of 2)

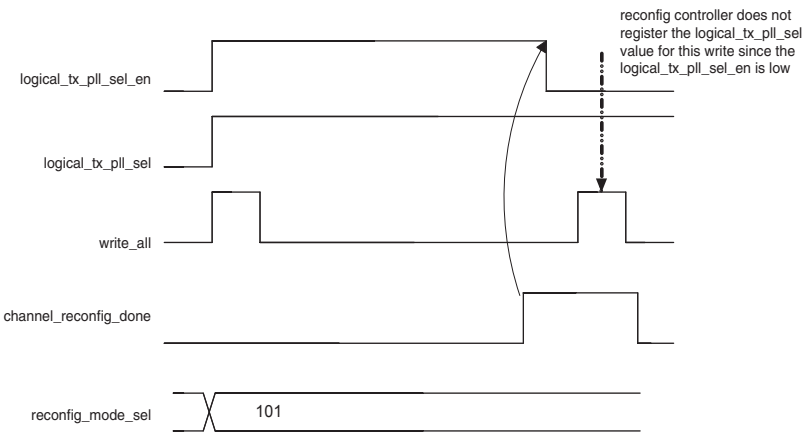
logical_tx_pll_sel Port	logical_tx_pll_se_en Port	Selected logical tx pll Value by the Reconfig Controller
enabled	not enabled	value on the `logical_tx_pll_sel` port.
not enabled	not enabled	logical `tx pll value` stored in the MIF.

When you configure a transceiver channel in the ALT2GXB MegaWizard, Altera recommends that you keep track of the TX PLL that drives the channel. You may require this information when you want to reconfigure the TX PLLs dynamically. This is illustrated in "Design Examples" on page 3–121.

The `logical_tx_pll_sel` port does not modify any transceiver setting on the RX side.

Figure 3–51 shows the required signal transitions to reconfigure the TX PLL with a `logical_tx_pll` value of **1**. Keep the `logical_tx_pll_sel` and `logical_tx_pll_sel_en` signals at a constant logic level until the reconfig controller asserts the `channel_reconfig_done` signal.

Figure 3–51. Signal Transitions of the logical_tx_pll_sel and logical_tx_pll_sel_en Ports

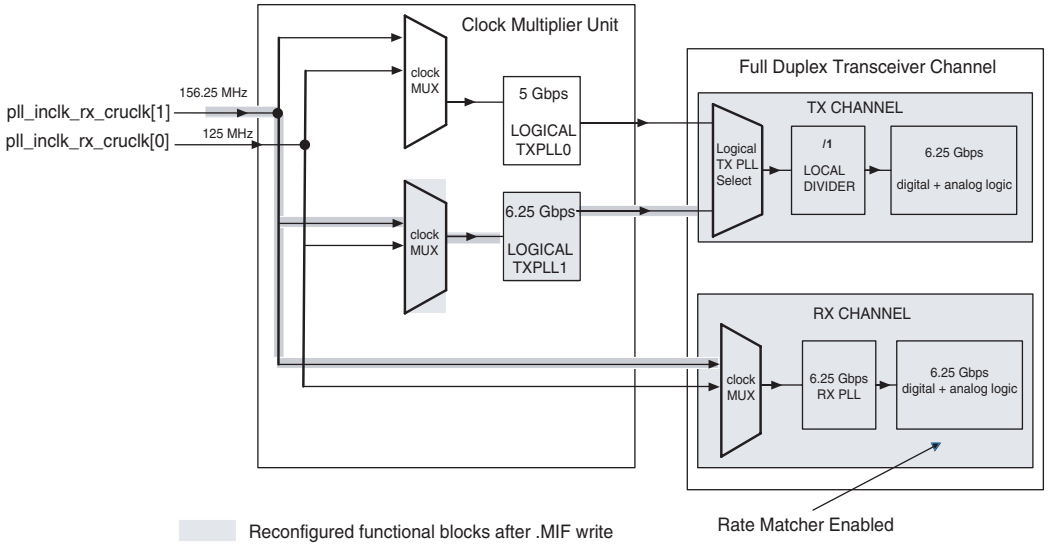


For illustration, the same example and MIF specified in the "Channel and TX PLL Reconfiguration" on page 3–100 is used here. The results of the reconfiguration under all the channel and CMU PLL reconfiguration modes are shown below. These results were achieved during run time, with the `logical_tx_pll_sel` input of the dynamic reconfig controller set to **1** (assuming the `logical_tx_pll_sel_en` is tied to **1**) and using the mode1 or mode2 MIF.

Channel and TX PLL Reconfiguration

Refer to Figure 3–46 for the channel configuration before the MIF write. Figure 3–52 shows the conditions after the channel is reconfigured using the mode2 MIF and setting the `logical_tx_pll_sel` to **1** during reconfiguration.

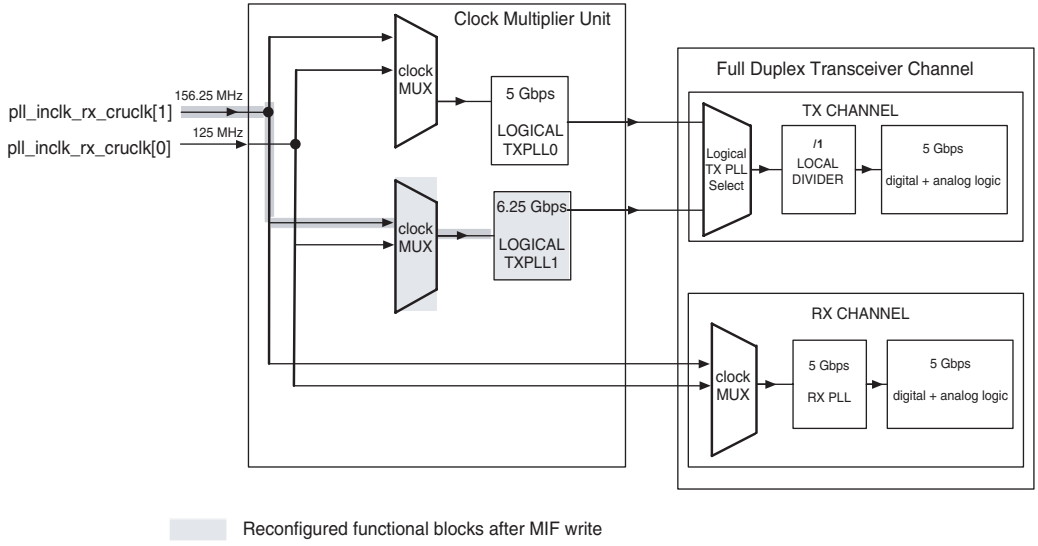
Figure 3–52. Reconfigured Functional Blocks Using logical_tx_pll_sel in Channel and TX PLL Mode



TX PLL Reconfiguration

Refer to Figure 3–46 for the channel configuration before the mode1 MIF is written. Figure 3–53 shows that the logical TXPLL1 is configured to 6.25 Gbps. The transmit channel still listens to the logical TXPLL0 and therefore runs at 5 Gbps (since in this mode, the logical tx pll select MUX is not reconfigured). The receive side is not configured with this feature.

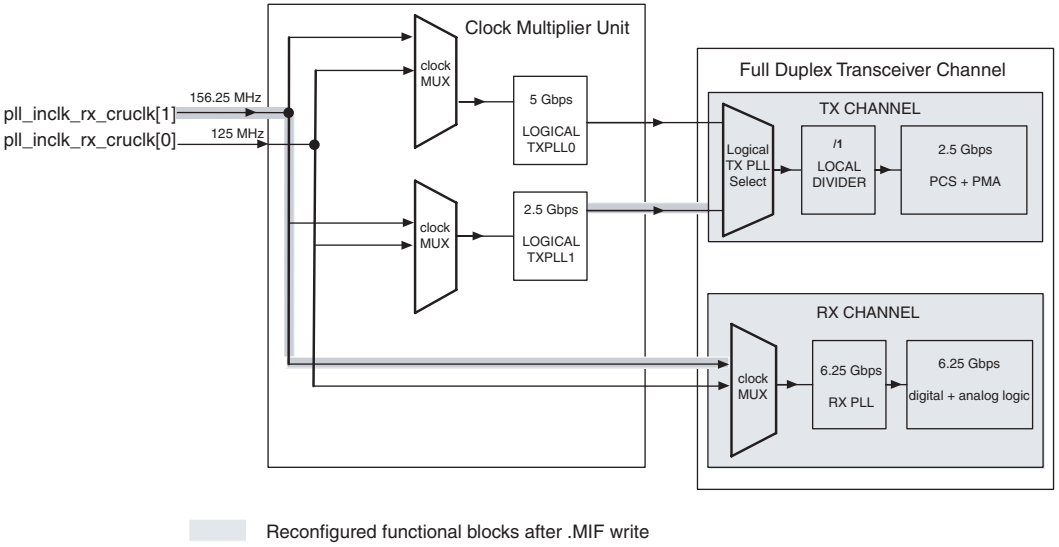
Figure 3–53. Reconfiguration Functional Blocks Using logical_tx_pll_sel in TX PLL Mode



Channel Reconfiguration with TX PLL Select

Refer to Figure 3–46 for the channel configuration before the mode1 MIF is written. Figure 3–54 shows the blocks that are reconfigured by the mode1 MIF and the `logical_tx_pll_sel` set to **1**. Note that in this case, the TX PLL is not configured. After the MIF is written, the logical TX PLL multiplexer gets configured to select the logical TXPLL1.

Figure 3–54. Reconfigured Functional Blocks using logical_tx_pll_sel in Channel Reconfiguration with TX PLL Select



You should keep the transceiver channel under reset during reconfiguration. Therefore, the channel may not be able to receive or transmit user data during reconfiguration.

You can use a MIF generated for one channel to all the other channels in the device if you meet the clocking requirements mentioned in "Clocking Enhancements and Requirements" on page 3–90.

TX PLL Powerdown

During channel and TX PLL reconfiguration or TX PLL Reconfiguration, the dynamic reconfig controller automatically powers down the selected TX PLL until it completes reconfiguring the selected TX PLL. The ALT2GXB_RECONFIG megafunction does not provide any external ports to control the TX PLL power down. If you reconfigure the main TXPLL, the `pll_locked` signal goes low. If you reconfigure the alternate TXPLL, the `pll_locked_alt` signal gets deasserted. Therefore, after reconfiguring the transceiver, wait for the `pll_locked` or `pll_locked_alt` signal from the ALT2GXB megafunction (depending on the TX PLL that is reconfigured) before continuing normal operation.

The dynamic reconfig controller powers down ONLY the selected TX PLL. The other TX PLL is not affected.

The main TXPLL corresponds to the TX PLL configuration set in the **General** tab of the ALT2GXB MegaWizard and the alternate TXPLL corresponds to the **Reconfig Alt PLL** tab.

Channel and CMU PLL Reconfiguration Duration

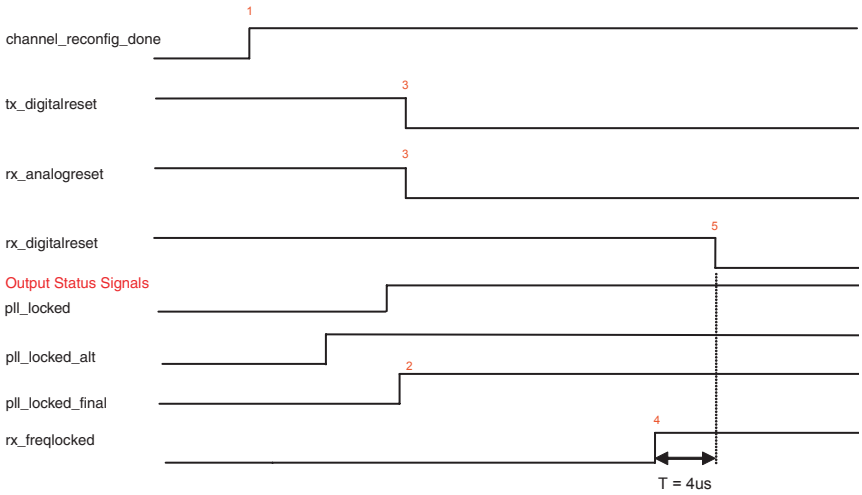
The dynamic reconfig controller takes the following number of reconfig clock cycles to write the contents in the MIF:

- Words 0 to 27 - approximately 260 clock cycles per word
- Words 28 to 37 - approximately 521 clock cycles per word

Reset Recommendations

Altera recommends that you follow a proper reset sequence during and after CMU PLL reconfiguration. Figure 3–55 shows the recommended reset sequence.

Figure 3–55. Reset Sequence During and After CMU PLL Reconfiguration (1 > 2 > 3 > 4 > 5)



As shown in Figure 3–55, assert the `tx_digitalreset`, `rx_digitalreset` and `rx_analogreset` when you initiate the CMU PLL reconfiguration MIF writes. After the dynamic reconfiguration control completes the CMU PLL reconfiguration, it asserts the `channel_reconfig_done` signal. After the `channel_reconfig_done` signal goes high, wait for `pll_locked` and `pll_locked_alt` (if you are using the alternate PLL) to go high (as

represented by pll_locked_final) and then de-assert the tx_digitalreset and rx_analogreset signals. Wait for a minimum of 4 µs after the rx_freqlocked signal goes high, then de-assert the rx_digitalreset signal.

Quartus II Settings and Requirements

The Quartus II software version 7.1 provides new assignments and settings to support the above mentioned channel and CMU PLL reconfiguration features.

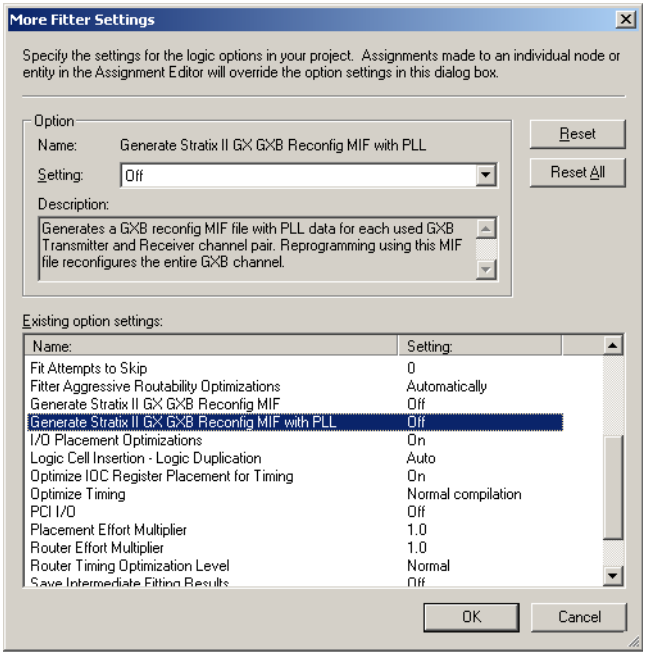
MIF Generation for Channel and CMU PLL Reconfiguration

To enable the Quartus II software version 7.1 to generate a MIF with 38 words, complete the following steps:

1. Go to the Assignments menu and select **Settings**, then **Fitter settings**.

2. Click the **more settings** button and set the **Generate Stratix II GX GXB Reconfig MIF with PLL** option to **ON** using the **Settings** option (as shown in Figure 3–56).

Figure 3–56. Quartus II Setting for MIF Generation



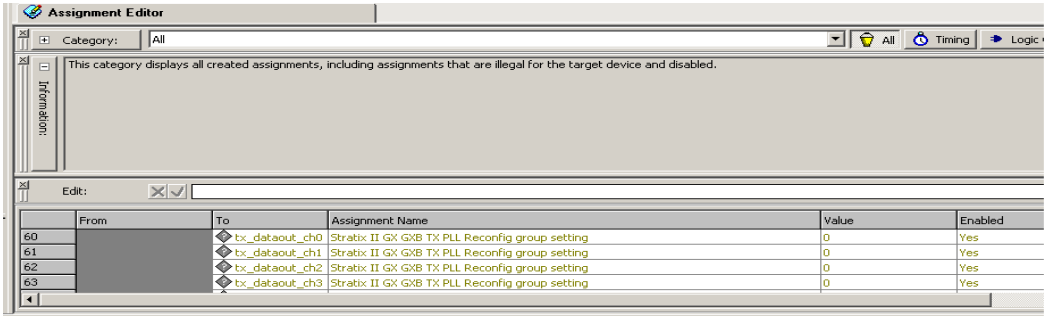
Grouping Transceiver Channels

The Quartus II software version 7.1 requires the following assignment editor setting for all channels assigned to the same transceiver bank, when you enable the **Channel and CMU PLL reconfiguration** option.

- Assignment setting: Assignment Name - **Stratix II GX GXB TX PLL Reconfig group setting** (as shown in Figure 3–57).

If you have more than one channel with the **Channel and CMU PLL Reconfiguration** feature enabled, and if you assign them to different reconfig groups without pin assignments for the `tx_dataout` pins, the Quartus II software automatically assigns these channels to different transceiver blocks. If you use a Stratix II GX device with one transceiver block, you cannot compile the design if you assign different TX PLL reconfig group values for the channels in your design.

Figure 3–57. Reconfig Group Setting Required for Channel and CMU PLL Reconfiguration



To understand the usage of this assignment setting, assume that you have two transmit channels in the same transceiver bank with the **Channel and CMU PLL Reconfiguration** option enabled. If the transmit output pins are `tx_dataout_ch0` and `tx_dataout_ch1`, set the following assignment setting to compile the design:

To : **tx_dataout_ch0**
Assignment Name: **Stratix II GX GXB TX PLL Reconfig group setting**
Value : **0**

To : **tx_dataout_ch1**
Assignment Name: **Stratix II GX GXB TX PLL Reconfig group setting**
Value : **0**

ALT2GXB MegaWizard Settings

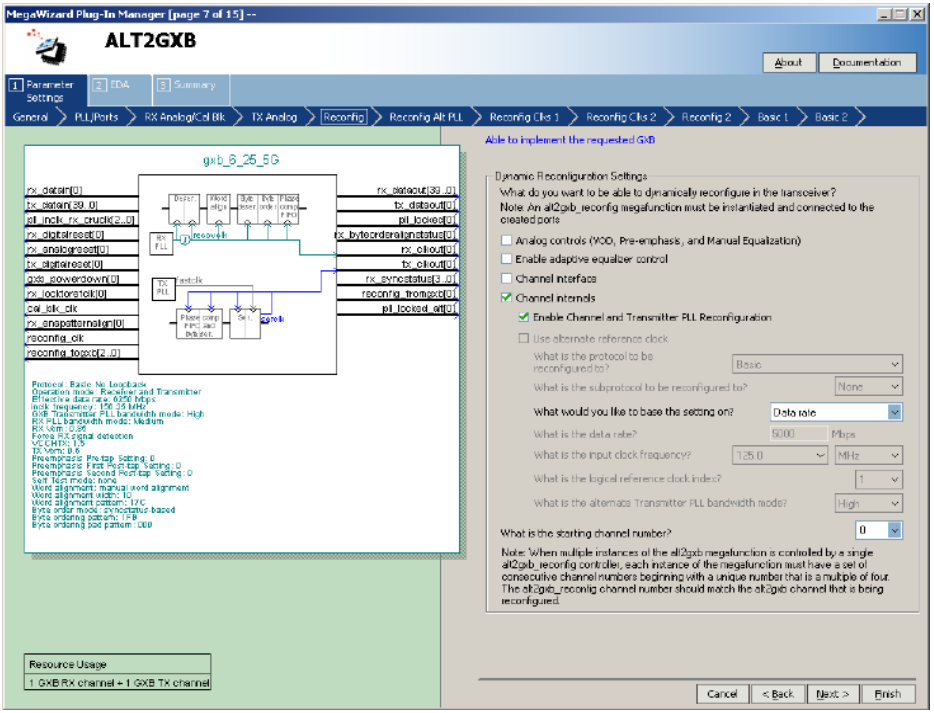
This section discusses the enhancements in the ALT2GXB MegaWizard to support the channel and CMU PLL reconfiguration feature.

Reconfig Tab Settings

When you enable the **channel internals** field, you can select the **Enable channel and Transmitter PLL** option (shown in Figure 3–58). This option allows you to use the channel and CMU PLL reconfiguration feature. When you select this option, the ALT2GXB MegaWizard enables new tabs—**Reconfig_Alt_PLL**, **Reconfig Clks 1,** and **Reconfig Clks 2**—to differentiate the settings for this feature from those of the channel reconfiguration feature (introduced in the Quartus II software version 6.1). This helps to maintain backward compatibility with the channel reconfiguration feature.

When you select the **Enable Channel and Transmitter PLL Reconfiguration** option, you cannot select the **Use alternate reference clock** option (used in channel reconfiguration feature). These two fields are mutually exclusive.

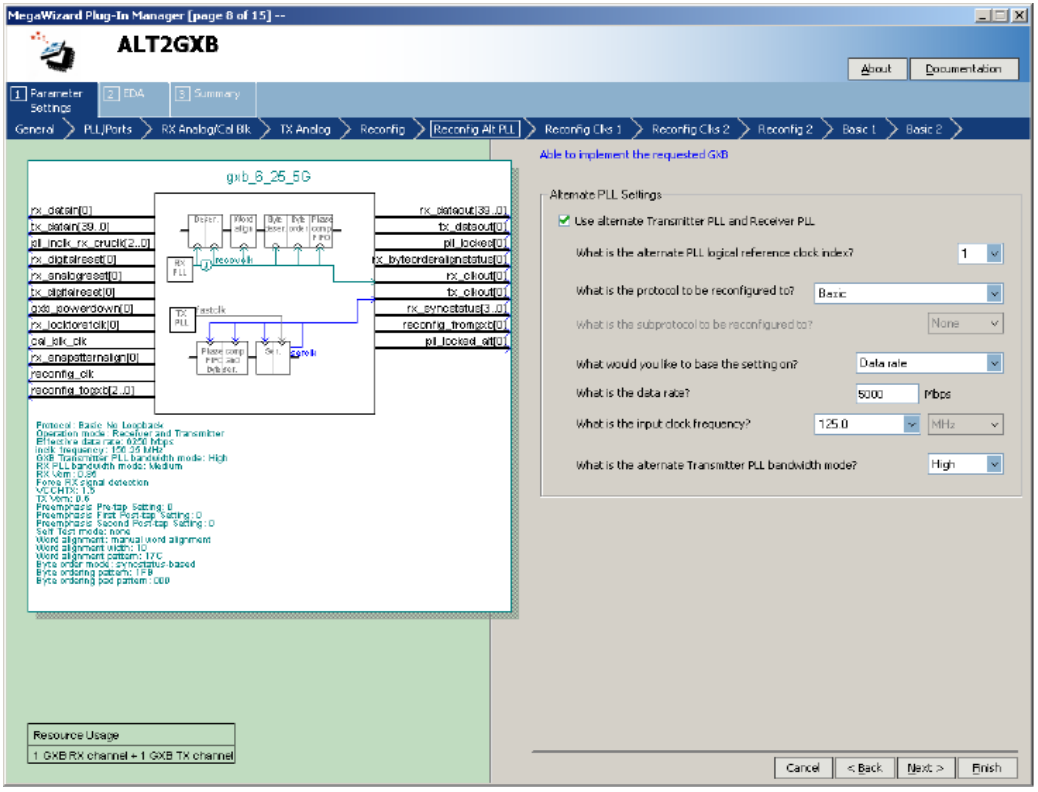
Figure 3–58. Reconfig Tab



Reconfig Alt PLL Tab Settings

When you select the **Use Alternate Transmitter and Receiver PLL** option, you can set the logical tx pll value to **0** or **1** for the alternate TXPLL from the **What is the alternate PLL logical reference index?** option (Figure 3–59).

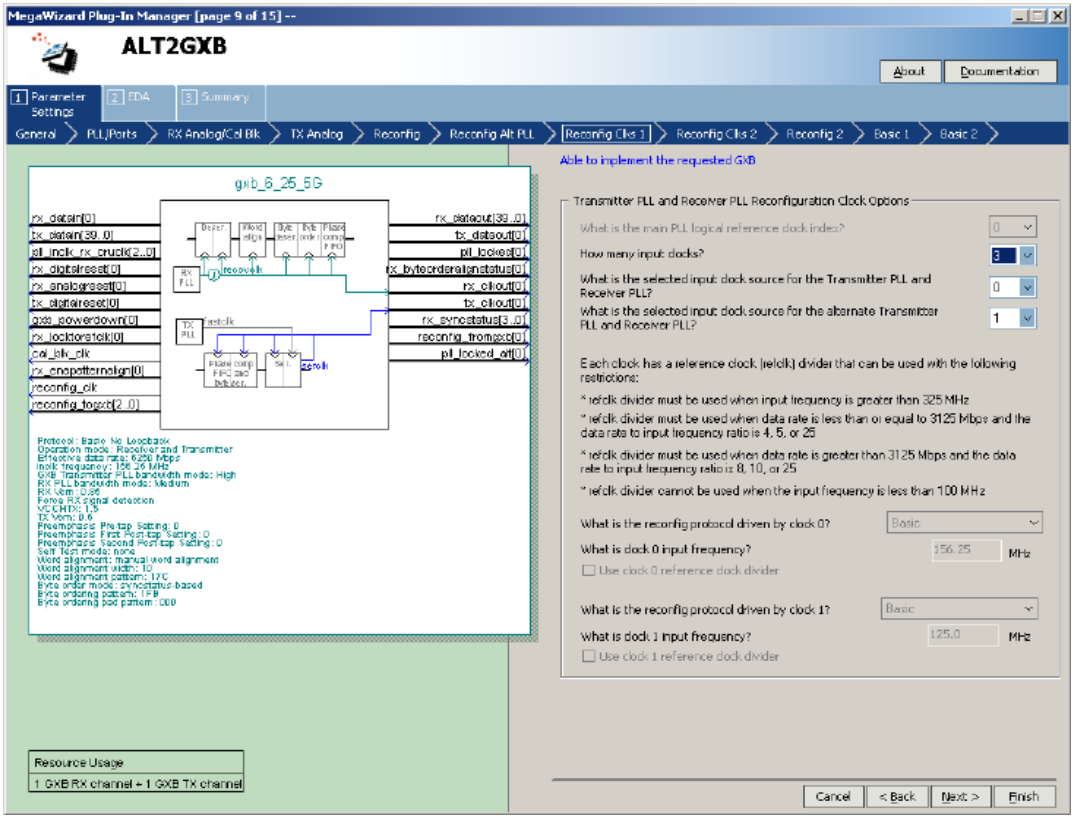
Figure 3–59. Reconfig Alt PLL Tab



Reconfig Clks 1 Tab

This tab provides options for the input reference clocks. The first option, **what is the main PLL logical reference clock index**, provides the logical tx pll value for the main TXPLL. If you have enabled the alternate TXPLL in the **Reconfig Alt PLL** tab, the ALT2GXB MegaWizard automatically selects the logical tx pll value of the main TXPLL as the complement of the alternate TXPLL. Otherwise, you can select the logical tx pll value for the main TXPLL in this tab (at the **What is the main PLL logical reference clock index?** option in Figure 3–60).

Figure 3–60. Reconfig Clks 1 Tab



The **How many input clocks?** option in Figure 3–60 shows the number of input reference clocks. When you set this field to **5**, the ALT2GXB Megawizard provides a **Reconfig Clks 2** tab to specify information about additional clock inputs. The **What is the selected input clock source for the Transmitter PLL and Receiver PLL?** and **What is the selected input clock source for the alternate Transmitter PLL and Receiver PLL?** options are used to select the input clocks for the main and alternate TXPLLs as well as the RX PLLs. The **What is the reconfig protocol driven by clock 0?** and **What is clock 0 input frequency?** options provide the protocol and clock frequency options for other clock sources that you anticipate you will use in your design.

For additional information on the input clock requirements, refer to "Clocking Enhancements and Requirements" on page 3–90.

Based on your settings in these fields, the ALT2GXB MegaWizard determines whether the `refclk` pre-divider should be enabled. For example, if you select the **SONET/SDH OC-12** protocol in the **what is the reconfig protocol driven by clock0** option, the ALT2GXB megafunction automatically enables the `refclk` pre-divider and connects the output of the pre-divider to the input reference clock port of the TX PLLs and RX PLLs. Similarly, if you select the input clock frequency greater than 325 MHz, the `refclk` pre-divider is enabled.

When you select the **Use clock 0 reference clock divide**r option, the Quartus II software instantiates the `refclk` pre-divider for the clock input.

If the information provided in the **General** and **Reconfig Alt Pll** tabs meet one of the conditions specified in "Using Dedicated refclks" on page 3–92, the Quartus II software automatically instantiates the `refclk` pre-divider for the corresponding clock input. For other clock inputs, you should determine whether the clock input frequency and the data rate meets one of the conditions specified in "Using Dedicated refclks" on page 3–92.

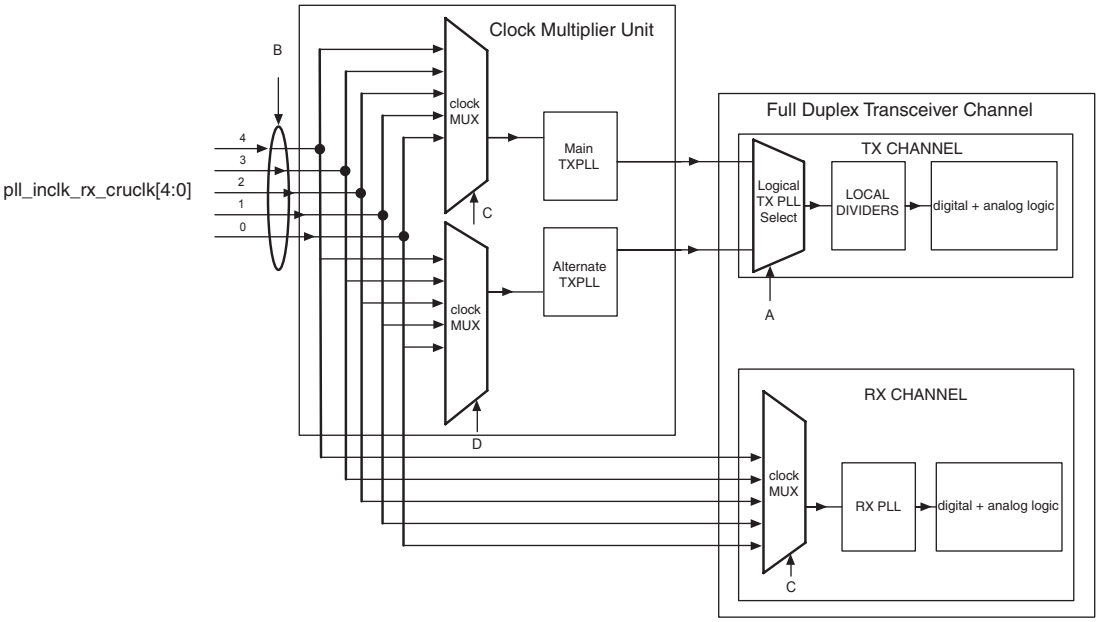
Example of a Condition to Select this Option:
Assume that you are using a clock input with a 125 MHz to configure the TX PLL to run the channel at the 3.125 Gbps data rate. In this case, the ratio of TX PLL data rate to input clock frequency is 25. This meets condition 2 specified in "Using Dedicated refclks" on page 3–92. Therefore, select this option so that the Quartus II software instantiates the `refclk` pre-divider for this clock source.

When the Quartus II software creates a pre-divider for a dedicated input reference clock (`refclk`), only the output of the pre-divider is available to clock the TX PLL and /or RX PLL.

If you would like to reuse the MIF across transceiver channels, you must have the same order of clock inputs across all the ALT2GXB instantiations that are using this MIF.

Figure 3–61 shows the mapping of the MegaWizard settings marked by "A", "B", "C", and "D" with the actual settings in the hardware (refer to Figure 3–60 for the **Reconfig Clks 1** tab).

Figure 3–61. Mapping Between the MegaWizard Settings and Hardware Settings



The other settings in the ALT2GXB MegaWizard are not specific to the channel and CMU PLL reconfiguration feature. Therefore, the other tabs are not discussed in this section.

Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about other ALT2GXB MegaWizard settings.

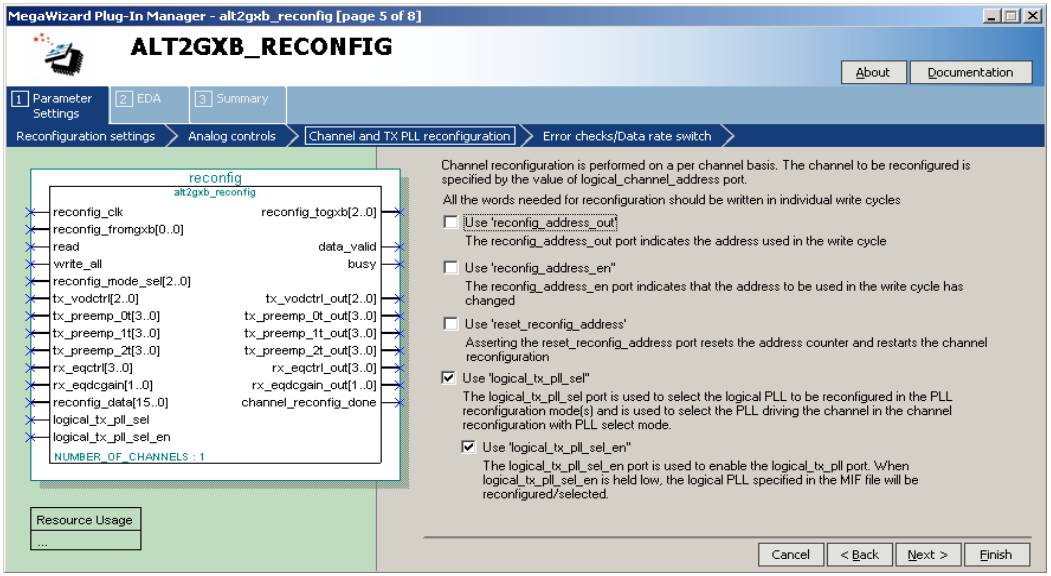
ALT2GXB RECONFIG Tab Settings

The Quartus II software version 7.1 has the following enhancements in the **ports/values** option in the ALT2GXB_RECONFIG MegaWizard.

- `reconfig_mode_sel`: The ALT2GXB_RECONFIG MegaWizard has new `reconfig_mode_sel` values to support the **channel and CMU PLL reconfiguration** option. For a complete list of `reconfig_mode_sel` values, refer to Table 3–9.

- `logical_tx_pll_sel`: The `logical_tx_pll_sel` port is optional. You can select this port in the **Channel and TX PLL Reconfiguration** tab. The value set in this port during reconfiguration overrides the logical tx pll value stored in the MIF. Refer to "Logical TX PLL Select" on page 3–105 for more information.
- `logical_tx_pll_sel_en`: The `logical_tx_pll_sel_en` port is optional. You can select this port in the **Channel and TX PLL Reconfiguration** tab. If this port is selected, the ALT2GXB_RECONFIG block registers the value on the `logical_tx_pll_sel` only if the `logical_tx_pll_sel_en` is asserted. Figure 3–62 shows the **Channel and TX PLL reconfiguration** tab in the ALT2GXB_RECONFIG MegaWizard in the Quartus II software version 7.1.

Figure 3–62. ALT2GXB RECONFIG Tab



The functionality of all other signals, such as `write_all`, `channel_reconfig_done`, `reconfig_address_en`, `logical_channel_address`, `data_valid`, and `busy`, have not changed since Quartus II software version 6.1. To write the MIF, follow the method used for the channel reconfiguration feature. Refer to Figure 3–8 on page 3–28 for more information.

Merging Transceiver Channels with Dynamic Reconfiguration Enabled

The following are the Quartus II software version 7.1 requirements for merging multiple transceiver channels that have the **Channel and CMU PLL Reconfiguration** option selected in the same transceiver bank:

- Assign all the channels to the same reconfig group. Refer to "Quartus II Settings and Requirements" on page 3–111 for more information on the reconfig group setting.
- All the channels should have the same reconfig options. That is, if you select **PMA controls**, **channel interface**, or **channel internals** in one channel, all the other channels should have the same selection. Some of the other scenarios for merging channels in the same transceiver bank are discussed below.

Case 1: Merging Transceiver Channels Listening to Two TX PLLs

Consider that you create an ALT2GXB instantiation for a full-duplex or TX-only configuration that has a main and alternate TXPLL. If you want to place other channels in the same transceiver bank, the other channels should also have a main and alternate TXPLL option to merge successfully. For example, consider that you create the following instantiation:

- **Instantiation1**—one full-duplex channel with the main TXPLL (assume a `logical_tx_pll` value of **0**), configured to 6.25 Gbps data rate and the alternate TXPLL configured to 2.500 Gbps.

Assume that you create another instantiation with the following configuration:

- **Instantiation2**—one full-duplex channel with only one TX PLL (assume a `logical_tx_pll` value of **0**), configured to 6.25 Gbps.

In this case you cannot merge instantiation1 and instantiation2 in the same transceiver bank since instantiation2 listens to only one TX PLL. To successfully merge the two instances, create instantiation2 with an alternate TXPLL configured to 2.500 Gbps.

Case II: Merging Transceiver Channels Listening to One TX PLL

Consider that you create an ALT2XB instantiation (full-duplex or TX-only configuration) that has only one TX PLL. If you would like to create another ALT2GXB instantiation configured at a different data rate in the

same transceiver bank, provide different logical tx pll values for the two instantiations. For example, to merge the following instantiations in the same transceiver bank:

- **Instantiation1**—full-duplex channel configured at 3.125 Gbps.
- **Instantiation 2**—full-duplex channel configured at 2.500 Gbps.

If you set the **what is the main PLL logical reference clock index** (in the **Reconfig Clks 1** tab) for instantiation1 to **0**, set this option to **1** for instantiation2. Since the Quartus II software requires separate TX PLLs for these two channels, the two instantiations should have different logical tx pll values.

Case III: Merging Separate Transmit-Only and Receive-Only Instantiation

In a full-duplex configuration with the **Channel and CMU PLL Reconfiguration** option enabled, the software automatically connects the same reference clock input to the TX PLL and RX PLL (explained in "Clocking Enhancements and Requirements" on page 3–90). If you merge a **transmit only** and a **receive only** configuration, the Quartus II software allows you to provide separate clock inputs for the TX PLL and RX PLL (you can connect the `pll_inclk_rx_cruclk[]` port of the two instances to two different clock source).

When you merge the **transmit only** and **receive only** configurations, you should add the **Stratix II GX Reconfig group setting** in the assignment editor for the `tx_dataout` and `rx_datain` pins and assign the same value to these two pins (**0** or **1**). This setting enables the Quartus II software to create a single (combined) MIF for the TX only and RX only instance.

Using this merging method, you can provide separate clock inputs to the TX PLL and RX PLL.

If you set the starting channel numbers in the ALT2GXB MegaWizard for the TX instance to **0** and RX instance to **4**, you can use `logical_channel_address` in the reconfig controller set to **0** or **4** to perform **Channel and CMU PLL Reconfiguration** on this transceiver channel.

Design Examples

This section covers the steps used in creating a design with the **Channel and CMU PLL reconfiguration** feature enabled.

Case I: Configuring Transceiver Channels to Switch Together Between GIGE, SONE-OC48, and Fibre Channel (FC)-4G Protocols

The GIGE, SONET/SDH OC48, and FC-4G have different input reference clocks, data path, and clocking requirements. For this example, assume the following Stratix II GX device configuration:

- Three transceiver banks
- Six full-duplex channels with two channels in each transceiver bank for a total of six channels (CH0, CH1, CH2, CH3, CH4, CH5).
- Each channel can independently switch between GIGE, SONET/SDH-OC48, and FC- 4G protocols.
- Assume that all channels are configured to FC-4G protocol at system power up.
- FC-4G uses Basic mode.

FC-4G and FC-2G refer to the fibre channel protocol at 4.25 Gbps and 2.125 Gbps data rate, respectively.

Table 3–12 shows the different parameters and ALT2GXB functional blocks for these three protocols.

Table 3–12. Differences in Functional Blocks Between GIGE, Fibre Channel, and SONET/SDH-OC48 *Note (1)*

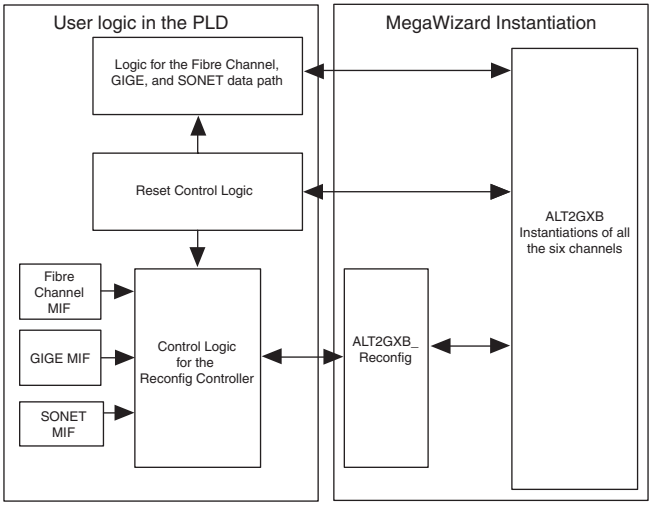
Parameters	Fibre Channel (Basic Mode) 4.25 Gbps	GIGE 1.25 Gbps	SONET/SDH OC48 2.488 Gbps
Selected input reference clock	106.25 MHz	125 MHz	77.76 MHz
PLD width	40	8	16
Byte Serializer/ Byte deserializer	yes	no	yes
8B/10B	no	yes	no
Rate Matcher	no	yes	no
Byte order block	yes	no	yes
Clock used for the receive side parallel interface	`rx_clkout`	`tx_clkout` (since rate matcher is used)	`rx_clkout`

Note to Table 3–12:

(1) The ALT2GXB MegaWizard allows more options for the input reference clock. For this example, we have selected the values shown in the table.

The differences between the three protocols determine the ALT2GXB MegaWizard settings. Figure 3–63 shows the top-level block diagram of the example design.

Figure 3–63. Top-Level Block Diagram of the Example Design

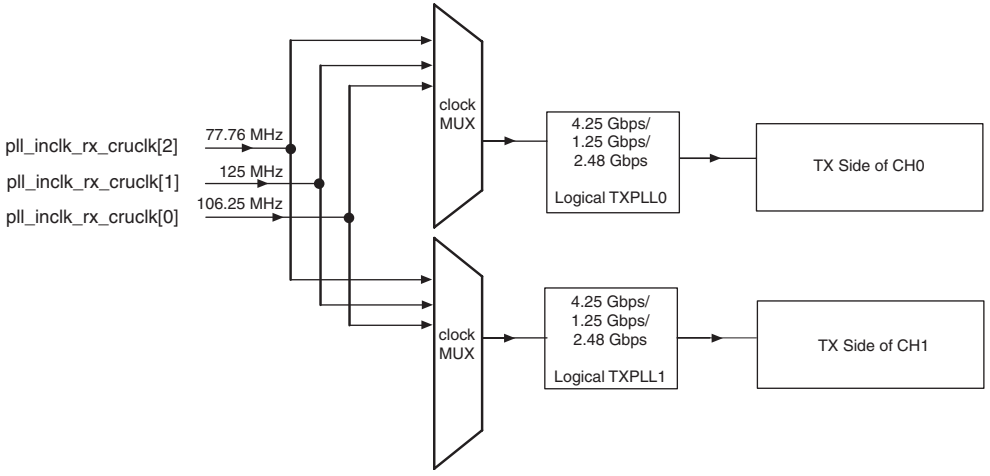


How Many MIFs do I Require?

For this example design, you can use one TX PLL per channel since you require only two full-duplex channels in the transceiver bank. To switch between three protocols, you need three MIFs. If you have consistent clocking across the three transceiver banks, you can reuse the same MIF across all the channels in the device. Figure 3–64 shows the clocking and TX PLL connections only for CH0 and CH1 (CH2-CH3, CH4-CH5 have the same configuration). To simplify the illustration, only the transmit channels of CH0 and CH1 and the TX PLL connections are shown.

If you create three MIFs (for FC-4G, GIGE, and SONET/SDH OC48) for one TX PLL, you can reuse the MIF in the other TX PLL using the `logical_tx_pll_sel` port in the dynamic reconfig controller. This means that you do not need a separate MIF for CH1 (since the other TX PLL is connected to CH1). Similarly, the same method can be applied for all the other channels in this example design. In total, you only need three MIFs for this example design.

Figure 3–64. TX PLL Connections to CH0 and CH1 for the Example Design



The 4.25/1.25/2.48 Gbps shown in Figure 3–64 indicates the possible switched data rates for the TX PLL to implement this example design.

The following discussion of the design is divided into five sections:

- Section I—ALT2GXB MegaWizard Settings for the Three Protocols
- Section II—ALT2GXB_RECONFIG MegaWizard Instantiation
- Section III—Steps to Create the MIF
- Section IV—Reset Control Logic and User Logic
- Section V—Top-Level Design and SRAM Object File (**.sof**) Generation

Section I— ALT2GXB MegaWizard Settings for the Three Protocols

Tables 3–13, 3–14, and 3–15 list the MegaWizard settings for each of the three protocols.

Table 3–13. FC-4G Protocol Settings (Part 1 of 3)

Tab Page and Option	Setting
General Tab Settings	
which protocol you will be using	basic
which sub protocol you will be using	serial loopback
operation mode	receiver and transmitter

Table 3–13. FC-4G Protocol Settings (Part 2 of 3)

Tab Page and Option	Setting
what is deserializer block width	double
what is channel width	40 (8b/10b encoder/decoder in the ALT2GXB is not used)
what is the data rate	4250 Mbps
what is the input clock frequency	106.25 MHz
what is the data rate division factor	1
select the `rxdigitalreset`, `txdigitalreset`, and `rxanalogreset` ports	
PLL/Ports Tab Settings	
select the `gxb_powerdown`, `rx_freqlocked`, `pll_locked` in the screen	
RX Analog/Cal BLK Tab Settings	
select the calibration block	
select the `cal_blk_powerdown` if required	
TX Analog Tab Setting	
select the appropriate settings based on your requirements	
Reconfig Tab Settings	
select channel interface	This is required since the three protocols require different PLD widths (refer to Table 3–12).
select channel internals and enable channel and transmitter PLL reconfiguration	
Reconfig Alt PLL Tab Setting	
In this example design, you are using only two channels in the transceiver block. Since there are two TX PLLs per transceiver block, use one TX PLL for each channel and reconfigure the same TX PLL to switch across protocols. Therefore, you do not need an alternate TXPLL for this instance.	
Reconfig Clks 1 Tab Settings	
what is the main PLL logical reference clock index	0

Table 3–13. FC-4G Protocol Settings (Part 3 of 3)

Tab Page and Option	Setting
how many input clocks	3 (77.76 MHz, 125 MHz, and 106.25 MHz). Assume: clock2 = 77.76 MHz clock1 = 125 MHz clock0 = 106.25 MHz
what is the select input clock source for transmitter and receiver PLL	0
what is the reconfig protocol driven by clock1	GIGE
what is clock1 input frequency	125 MHz
use clock 1 reference clock divider	do not check this option
what is the reconfig protocol driven by clock2	SONET/SDH
what is clock2 input frequency	77.76 MHz
use clock 2 reference clock divider	do not check this option
Reconfig2 Tab Settings	
how should the receivers be clocked	select **use respective core clocks** since you clock the receive parallel date with `tx_clkout` for the GIGE protocol and `rx_clkout` for the other two protocols. Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for more information about these options.
how should the transmitters be clocked	use the respective channel transmitter core clocks.
check the control box to use the corresponding control port	select the protocol-specific signals. For SONET/SDH, you need `rx_byteorderalignstatus`, `rx_ala2sizeout`, etc. Refer to the *Stratix II GX ALT2GXB Megafunction User Guide* chapter in volume 2 of the *Stratix II GX Device Handbook* for more information.
Basic1 and Basic2 Tab Setting	
select the word alignment and other ports based on your requirements and complete the MegaWizard	

.

Table 3–14. GIGE Protocol Settings (Part 1 of 2)

Tab Page and Option	Setting
General Tab Settings	
which protocol you will be using	GIGE
operation mode	receiver and transmitter
what is the input clock frequency	125 MHz
select the `rxdigitalreset`, `txdigitalreset`, and `rxanalogreset` ports	
PLL/Ports, RX Analog, Cal Blk, TX Analog, Reconfig Tab Settings	
Set the same settings as the FC-4G ALT2GXB instance mentioned in Tables 3–13.	
Reconfig Alt PLL Tab Setting	
no selection required.	
Reconfig Clks 1 Tab Settings	
what is the main PLL logical reference clock index	0 Note: Use this setting because you intend to generate the MIF with a logical tx pll value of **0**. Refer to "How Many MIFs do I Require?" on page 3–123
how many input clocks	3 (77.76 MHz, 125 MHz, and 106.25 MHz).
what is the select input clock source for transmitter and receiver PLL	1
what is the reconfig protocol driven by clock0	BASIC
what is clock0 input frequency	106.25 MHz
use clock 0 reference clock divider	do not check this option
what is the reconfig protocol driven by clock2	SONET/SDH
what is clock2 input frequency	77.76 MHz Note: The order of the clock inputs is the same as of the FC-4G instantiation shown in Table 3–13
use clock 2 reference clock divider	do not check this option
Reconfig2 Tab Settings	
same as of the FC-4G instantiation shown in Table 3–13	

Table 3–14. GIGE Protocol Settings (Part 2 of 2)	
Tab Page and Option	**Setting**
Basic1 and Basic2 Tab Setting	
select the word alignment and other ports based on your requirements and complete the MegaWizard	

.

Table 3–15. SONET/SDH OC48 Protocol Settings (Part 1 of 2)	
Tab Page and Option	**Setting**
General Tab Settings	
which protocol you will be using	SONET/SDH
which sub protocol	OC48
operation mode	receiver and transmitter
what is the input clock frequency	77.76 MHz
select the `rxdigitalreset`, `txdigitalreset`, and `rxanalogreset` ports	
PLL/Ports, RX Analog, Cal Blk, TX Analog, Reconfig Tab Settings	
set the same settings as the FC-4G ALT2GXB instance mentioned in Tables 3–13	
Reconfig Alt PLL Tab Setting	
no selection required.	
Reconfig Clks 1 Tab Settings	
what is the main PLL logical reference clock index	0
how many input clocks	3 (77.76 MHz, 125 MHz, and 106.25 MHz)
what is the select input clock source for transmitter and receiver PLL	2
what is the reconfig protocol driven by clock0	BASIC
what is clock0 input frequency	106.25 MHz
use clock 0 reference clock divider	do not check this option
what is the reconfig protocol driven by clock1	GIGE
what is clock1 input frequency	125 MHz

Table 3–15. SONET/SDH OC48 Protocol Settings (Part 2 of 2)	
Tab Page and Option	**Setting**
use clock 1 reference clock divider	do not check this option
Reconfig2 Tab Settings	
same as of the FC-4G instantiation shown in Table 3–13	
Basic1 and Basic2 Tab Setting	
select the word alignment and other ports based on your requirements and complete the MegaWizard	

The ALT2GXB MegaWizard instantiation for the three protocols is complete.

The following are the settings for channel 1:

- Assume that the default configuration of CH1 is FC-4G. You can copy the FC-4G instantiation created for CH0. The only change required for this new instantiation is in the **Reconfig Clks 1** tab.
- Set the **What is the main PLL logical reference clock index?** option to **1** (this is the logical tx pll value) and complete the MegaWizard. For CH0, set the value in this field to **0**. Since the design goal is to reconfigure these two channels independently, set different logical tx pll values for these two channels and complete the MegaWizard. For CH2 and CH4, reuse the CH0 instance. Similarly, for CH3 and CH5, reuse the CH1 instance.

Section II — ALT2GXB_RECONFIG MegaWizard Instantiation

The following are settings for the ALT2GXB_RECONFIG MegaWizard:

- Set the **What is the number of channels controlled by the reconfig controller?** option to **24**. In this design, you have six instantiations (for six channels). The starting channel numbers for each of these instantiations should be a multiple of four. Each of these ALT2GXB instantiation has a `reconfig_fromgxb` output port.
- The reconfig controller provides one `reconfig_fromgxb` input port for a multiple of 4 channels. Therefore, set the above field to **24** (rounded to the nearest transceiver block). For additional information on starting channel numbers and logical channel addressing, refer to "Introduction" on page 3–1.

- Select the **Channel and TX PLL select/reconfig** option.
- In the **Channel and TX PLL reconfiguration** tab, select `reconfig_address_out` and `reconfig_address_en`. Select `logical_tx_pll_sel` and `logical_tx_pll_sel_en` ports so that you can reuse the MIF. (Refer to "How Many MIFs do I Require?" on page 3–123.

Section III — Steps to Create MIFs

This section explains the steps to create all the MIFs at one time:

1. Go to the Assignments menu and select **Settings**, then **Fitter settings**.

2. Click the **more settings** button and set the **Generate Stratix II GX GXB Reconfig MIF with PLL** option to **ON** using the settings option (as shown in Figure 3–56).

3. Create a top-level design file and include the three instantiations created for CH0 (FC-4G, GIGE, and SONET/SDH OC48 protocol). Connect the `pll_inclk_rx_cruclk[]` ports to the following clock source:

   a. `pll_inclk_rx_cruclk[0]` - **106.25** MHz. Assume `refclk0` of transceiver bank 13.

   b. `pll_inclk_rx_cruclk[1]` - **125** MHz. Assume `refclk0` of transceiver bank 14.

   c. `pll_inclk_rx_cruclk[2]` - **77.76** MHz. Assume `refclk0` of transceiver bank 15.

4. For this example design, assume that the three clock inputs are provided from the dedicated `refclk` pins. If you provide input reference clocks through the global clock networks, refer to "Clocking Enhancements and Requirements" on page 3–90 for usage limitations.

5. Assign the `tx_dataout` and `rx_datain` pins of the FC-4G, GIGE, and SONET/SDH OC48 instantiations to transceiver banks 13, 14, and 15, respectively. Since you have assigned logical tx pll value to **0** to all these instantiations, place these channels in three different transceiver banks to compile successfully (refer to "Case II: Merging Transceiver Channels Listening to One TX PLL" on page 3–120). The intent of placing these instantiations in different banks is to generate all the MIFs at one time.

6. Rename the generated MIFs to indicate the protocol for which the MIF is configured.

Section IV — Reset Control Logic and User Logic

The reset control logic takes care of resetting the transceiver during system initialization and during reconfiguration (Altera recommends a specific reset sequence, refer to "Reset Recommendations" on page 3–66 for more information).

For the user logic, use different clocks (`tx_clkout` for GIGE protocol and `rx_clkout` for FC-4G and SONET/SDH OC48) for the parallel data in the receive interface of the ALT2GXB. The user logic is not discussed in this section. Refer to Figure 3–37 on page 3–82 for more information regarding user logic in a similar configuration.

Section V — Top-Level Design and SRAM Object File (.sof) Generation

Follow these steps to generate a SRAM object file:

1. Instantiate the six ALT2GXB channels in the top-level design. That is, stamp the FC-4G instance created for `logical_tx_pll` value **0** three times for CH0, CH2, and CH4.

2. Similarly, stamp the instance created for `logical_tx_pll` value **1** for CH1, CH3, and CH5.

3. Add the reset and user logic and connect the signals. In the assignment editor, use the **Stratix II GX GXB TX PLL Reconfig group setting** option and assign the `tx_dataout` of CH0 and CH1 to the same reconfig group (this is required to assign CH0 and CH1 to the same transceiver bank).

4. Similarly, assign the same reconfig groups for CH2-CH3 and CH4-CH5.

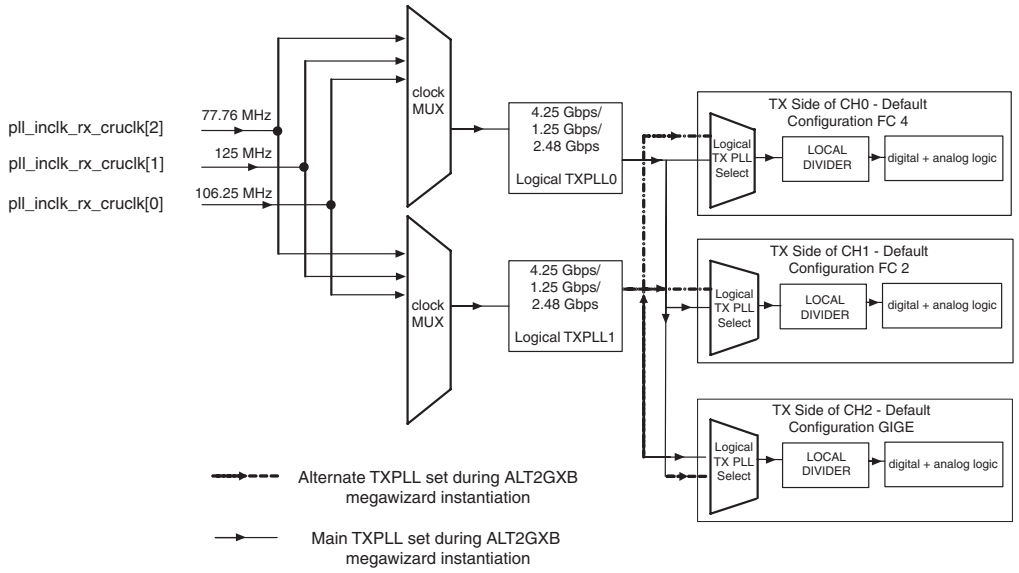
Case II: Configuring Transceiver Channels to Switch Independently Between Three Different Protocols

This example discusses the steps to reconfigure the three full-duplex channels (CH0, CH1, and CH2) in a transceiver bank between the FC-4G, FC-2G, GIGE, SONET/SDH OC48 protocols. In the previous example, the design used only two channels in a transceiver bank. Therefore, each channel could use a dedicated TX PLL.

In this example, the three channels are reconfigured in a transceiver bank. This means that two TX PLLs are shared between three channels. Therefore, if you use a main and alternate TXPLL for each channel, you can reconfigure the channel to any two of the four protocols by switching between the two TX PLLs.

Figure 3–65 shows the TX PLL connections of the TX side. (To simplify the illustration, only the TX side is shown). The figure shows that the default configurations of CH0, CH1, and CH2 are FC-4G, FC-2G, and GIGE, respectively.

Figure 3–65. Logical TX PLL Connections with the Transceiver Channel



How Many MIFs Do I Require?

You will need four MIFs for this design. You can generate the MIFs for one TX PLL and use the `logical_tx_pll_sel` in the reconfig controller to write the MIF contents into the second TX PLL. Assume that the TX PLL configured for FC- 4G data rate is assigned a logical tx pll value of **0**. This means that the other TX PLL configured for GIGE and SONET/SDH

OC48 protocol is assigned a logical tx pll value of **1**. Table 3–15 shows the number of MIFs required and the logical tx pll value assigned for the different configurations.

Table 3–16. MIF File for the Four Configurations

Number	MIF	logical tx pll Value
1	FC-4G	0
2	FC-2G	0
3	GIGE	1
4	SONET/SDH OC48	1

Use the same clocking connections and data path for these protocols that were provided in the previous example.

The following discussion of the design is divided into three sections:

- Section I—ALT2GXB MegaWizard Settings for the Three Protocols
- Section II—ALT2GXB_RECONFIG MegaWizard Instantiation
- Section III—Using the logical_tx_pll_sel During Reconfiguration

Section I — ALT2GXB MegaWizard Settings for the Three Protocols
In this section, only the ALT2GXB MegaWizard settings relevant to the channel and CMU PLL reconfiguration feature are discussed in Tables 3–17 through 3–20.

Refer to Tables 3–13, 3–14, and 3–15 for the tab settings that are not specified in this section.

Table 3–17. FC-4G Protocol Settings (Part 1 of 2)

Tab Page and Option	Setting
General Tab Settings	
which protocol you will be using	basic
which sub protocol you will be using	serial loopback
operation mode	receiver and transmitter
what is deserializer block width	double
what is channel width	40 (8b/10b encoder/decoder in the ALT2GXB is not used)
what is the data rate	4250 Mbps
what is the input clock frequency	106.25 MHz

Table 3–17. FC-4G Protocol Settings (Part 2 of 2)	
Tab Page and Option	**Setting**
what is the data rate division factor	1
select the `rxdigitalreset`, `txdigitalreset`, and `rxanalogreset` ports	
Reconfig Tab Settings	
select channel interface	this is required since the three protocols require different PLD widths (refer to Table 3–12)
select channel internals and enable channel and transmitter PLL reconfiguration	
Reconfig Alt PLL Tab Setting	
what is the alternate PLL reference clock index	1 you used a logical tx pll value of **0** for FC-4G. Therefore, this alternate index (for example, GIGE) should be set to **1**
protocol	GIGE
data rate	1.25 Gbps
clock frequency	125 MHz
Reconfig Clks 1 and Reconfig2 Tab Settings	
use the same clock order as the previous example. Refer to Table 3–13	

For the FC-2G configuration, use the TX PLL that provides FC-4G clock frequency and divide by two with the local divider in the TX channel.

Copy the instance created for the FC-4G instance. The only change required is in the **General** tab.

Table 3–18. FC-2G Protocol Settings (Part 1 of 2)	
Tab Page and Option	**Setting**
General Tab Settings	
which protocol you will be using	basic
which sub protocol you will be using	serial loopback
operation mode	receiver and Transmitter
what is deserializer block width	double
what is channel width	(8b/10b in the ALT2GXB)

Table 3–18. FC-2G Protocol Settings (Part 2 of 2)	
Tab Page and Option	**Setting**
what is the data rate	4250 Mbps
what is the input clock frequency	106.25 MHz
what is the data rate division factor	2

.

Table 3–19. GIGE Protocol Settings	
Tab Page and Option	**Setting**
General Tab Settings	
which protocol you will be using	GIGE
what is the input clock frequency	125 MHz
Reconfig Alt PLL Tab Setting	
what is the alternate PLL reference clock index	0 You used a logical tx pll value of **1** for GIGE. Therefore, the alternate index (for FC-4G) should be set to **0.**
protocol	BASIC
data rate	4.25 Gbps
clock frequency	106.25 MHz
Reconfig Clks 1 and Reconfig2 Tab Settings	
use the same clock order as the previous example. Refer to Table 3–13	

Copy the instance created for FC-4G instance. The only change required is in the **General** tab.

Table 3–20. FC-2G Protocol Settings (Part 1 of 2)	
Tab Page and Option	**Setting**
General Tab Settings	
which protocol you will be using	SONET/SDH
which sub protocol you will be using	OC48
what is the input clock frequency	77.76 MHz
Reconfig Alt PLL Tab Settings	

Table 3–20. FC-2G Protocol Settings (Part 2 of 2)	
Tab Page and Option	**Setting**
what is the alternate PLL reference clock index	0 You used a logical tx pll value of **1** for SONET/SDH OC48. Therefore, the alternate index (for FC-4G) should be set to **0**.
protocol	BASIC
data rate	4.25 Gbps
clock frequency	106.25 MHz

Section II — ALT2GXB_RECONFIG MegaWizard Instantiation
The following are the settings for the ALT2GXB_RECONFIG MegaWizard:

- Set the **What is the number of channels controlled by the reconfig controller** option to **12** (This will provide separate `reconfig_from_gxb` ports for each instance).
- Select the **Channel and TX PLL select/reconfig** option.
- In the **Channel and TX PLL Reconfiguration** tab, select the `reconfig_address_out`, `reconfig_address_en`, `logical_tx_pll_sel`, and `logical_tx_pll_sel_en` ports so that you can reuse the MIF. (Refer to "How Many MIFs Do I Require?" on page 3–132).

Section III — Using the logical_tx_pll_sel During Reconfiguration
Follow the same procedure as mentioned in "MIF Generation for Channel and CMU PLL Reconfiguration" on page 3–111 to create your MIFs. In the top-level design, assign the **Stratix II GX GXB TX PLL Reconfig group setting** and assign the same reconfig group to the three channels.

If you would like to reconfigure CH0 to SONET/SDH OC48 mode, use the following steps:

1. In the MegaWizard instantiation, set the logical tx pll value of the main TXPLL for CH0 to **0**. The SONET/SDH OC48 MIF contains the logical tx pll value of **1**. To use the SONET/SDH OC48 MIF for CH0, set both the `logical_tx_pll_sel` and `logical_tx_pll_sel_en` ports to **1** in the reconfig controller.

2. Set the `reconfig_mode_sel` value to **101** (channel and TX PLL reconfiguration) and write the SONET/SDH OC48 MIF.

Since CH0 and CH1 share the same TX PLL, configuring CH0 affects CH1. You can either configure CH1 to go to SONET/SDH OC48 mode or switch CH1 to listen to the GIGE mode by switching it to listen to the alternate TXPLL.

Adaptive Equalization (AEQ)

High-speed interface systems are used at different data rates with multiple backplane environments. These systems require different equalization settings to compensate for changing data rates and backplane characteristics. Manually selecting optimal equalization settings is cumbersome under these changing system characteristics. The adaptive equalization feature solves this problem by enabling the Stratix II GX device to continuously tune the receiver equalization settings based on the frequency content of the incoming signal. Five equalizer filters are tuned during this adaptive equalization process. The user logic can dynamically control the AEQ hardware through the dynamic reconfiguration controller.

This section explains the method to enable different options to control the AEQ hardware. Altera assumes that you have prior knowledge about the dynamic reconfiguration controller. For basic information, refer to "Dynamic Reconfiguration Controller Architecture" on page 3–2.

Conventions Used

The following conventions are used in this section:

- ALT2GXB_RECONFIG—Refers to the dynamic reconfiguration controller logic generated by the Quartus II AL2GXB_RECONFIG MegaWizard Plug-In Manager. ALT2GXB_RECONFIG and dynamic reconfiguration controller are used interchangeably in this section.
- Active channels—channels that have the **Enable adaptive equalizer control** option selected in the ALT2GXB MegaWizard. Selecting this option enables the adaptive equalization hardware.

AEQ Feature Requirements

The following are device requirements for the AEQ feature:

- Different device families require different silicon revisions, as shown in Table 3–21:

Table 3–21. Silicon Revision Requirements for the AEQ Feature

Device Family	Silicon Revision *(1)*
2SGX30	Revision A
2SGX60	Revision A
2SGX90	Revision C
2SGX130	Revision B

Note to Table 3–21:

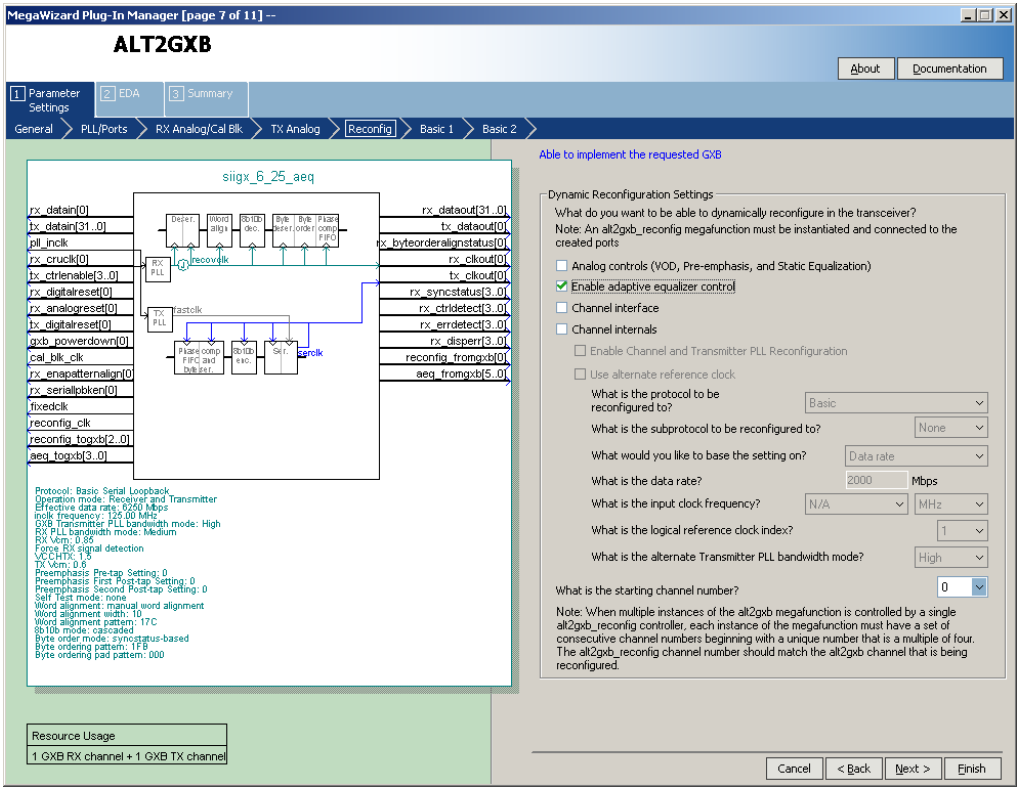
(1) You can identify the silicon revision by looking at the print below the device name in the device package. The third letter from the left indicates the silicon revision. For example, the print **AAC9X0607A** (third letter from left, **"C**") indicates a REV C silicon.

- The transceiver data rate needs to be > 2.5 Gbps
- The receive data needs to be 8B/10B encoded.
- Not available in PCI-Express (PIPE) functional mode (since the adaptive equalization hardware cannot perform the equalization process when the receive link is under the **electrical idle** condition)
- The receiver input signal should have a minimum envelope of 400 mv (differential peak-to-peak). The Quartus II software does not check for this requirement.
- AEQ is supported only in device speed grades C3, C4, or I4.

Enabling the AEQ Hardware

The AEQ hardware is available for each transceiver channel in the Stratix II GX device. To enable the AEQ hardware, select the **Enable adaptive equalizer control** option in the **Reconfig** page of the ALT2GXB MegaWizard plug-in Manager (Figure 3–66).

Figure 3–66. Enable the Adaptive Equalization Hardware



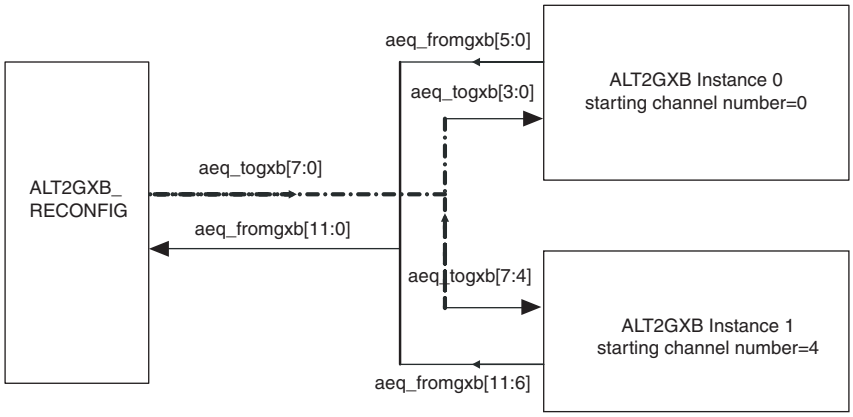
When you select this option, the ALT2GXB MegaWizard provides the following additional ports:

- `aeq_fromgxb[]`
- `aeq_togxb[]`
- `fixedclk`

The `aeq_fromgxb[]` and `aeq_togxb[]` ports provide the interface between the transceiver channel (ALT2GXB) and the dynamic reconfiguration controller (ALT2GXB_RECONFIG). For each channel, the width of the `aeq_fromgxb[]` and `aeq_togxb[]` ports are 6 bits and 4 bits, respectively. If you have multiple transceiver instances, connect the least significant byte of the `aeq_togxb[3:0]` and `aeq_fromgxb[5:0]` ports between the ALT2GXB_RECONFIG and the transceiver channel

with `logical_channel_address` value of **0**. Figure 3–67 shows the connections between multiple ALT2GXB instances and the dynamic reconfiguration controller.

Figure 3–67. Interface Connection Between the ALT2GXB and the ALT2GXB_RECONFIG Instance

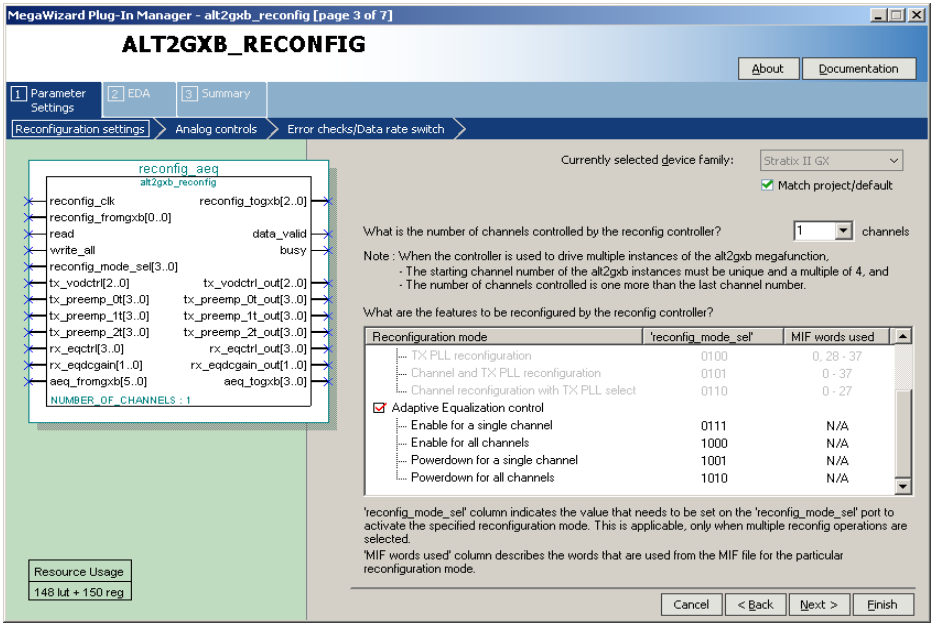


The `fixedclk` port provides the clock input to run the AEQ hardware. The range of the input clock frequency to the `fixedclk` port should be between 2.5 MHz and 125 MHz. To save clock routing resources, you can use the same clock pin to provide input clocks for the `fixedclk` and `reconfig_clk` ports.

When the transceiver channel is configured for PCI-Express (PIPE) protocol, the `fixedclk` is used to operate the receiver detect circuitry. In this protocol mode, `fixedclk` requires a fixed 125 MHz input clock frequency. The AEQ feature is not available in PCI-Express (PIPE) protocol mode.

The ALT2GXB_RECONFIG block provides a simple interface between the user logic and the transceiver channel to control the AEQ hardware. Figure 3–68 shows the ALT2GXB_RECONFIG MegaWizard page with different AEQ options.

Figure 3–68. ALT2GXB_RECONFIG with Different Adaptive Equalization Control Options



Controlling the AEQ Hardware

The ALT2GXB_RECONFIG provides different options to start and power down the adaptive equalization hardware. You can select these options by setting different values in the `reconfig_mode_sel[]` port. To use these options, set the `reconfig_mode_sel[]` port to the corresponding value shown in Table 3–22.

The dynamic reconfiguration controller provides the options shown in Table 3–22 to control the adaptive equalization operation.

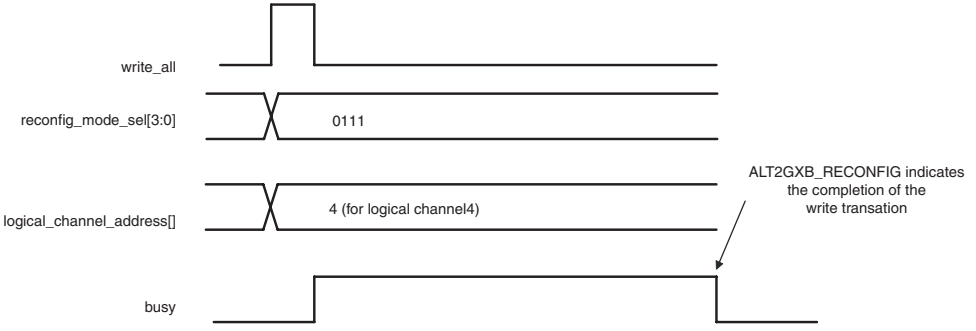
Table 3–22. reconfig_mode_sel Port Options for the Adaptive Equalization Feature

reconfig_mode_sel[3:0]	Adaptive Equalization Options
0111	Enable for a single channel
1000	Enable for all channels
1001	Power down for a single channel
1010	Power down for all channels

Enable for a Single Channel

This option, shown in Figure 3–68, provides the flexibility to start the adaptive equalization operation in a specific transceiver channel. To initiate the AEQ operation for a single channel, set the logical address of the channel in the `logical_channel_address[]` port. (For more information, refer to "Example for Using Logical Channel Address to Perform Channel Reconfiguration" on page 3–24). Set the `reconfig_mode_sel[3:0]` port to **0111** and assert the `write_all` signal for one `reconfig_clk` cycle, as shown in Figure 3–69.

Figure 3–69. AEQ Write Timing Diagram on a Single Channel



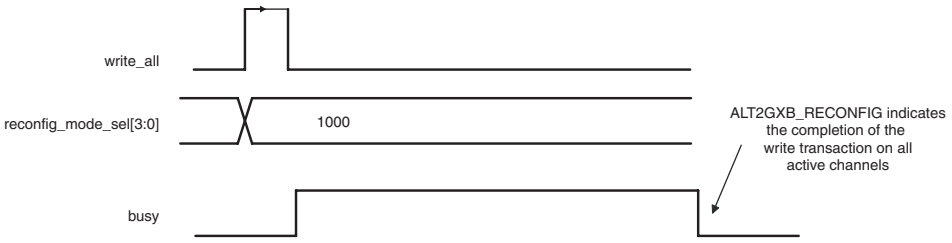
When the `write_all` signal is asserted, the dynamic reconfiguration controller writes the initialization and control values into the transceiver registers and initiates the adaptive equalization process. During this initialization process, the dynamic reconfiguration controller powers down the receiver buffer. This results in transient bit errors on the parallel interface on the receive side. The ALT2GXB_RECONFIG de-asserts the `busy` signal after the write transaction is completed.

Figure 3–69 shows the value **4** in the `logical_channel_address[]` signal as an example. The ALT2GXB_RECONFIG takes a maximum of approximately 7,000 `reconfig_clk` cycles to complete the AEQ write operation.

Enable for All Channels

This option allows you to initiate the adaptive equalization operation on all the active channels connected to the same dynamic reconfiguration controller. The method to initiate the AEQ operation for all channels is similar to "Enable for a Single Channel" on page 3–142. Set the `reconfig_mode_sel[3:0]` to **1000** and assert the `write_all` signal for one `reconfig_clk` cycle, as shown in Figure 3–70.

Figure 3–70. AEQ Write Timing Diagram on All Active Channels



The dynamic reconfiguration controller initiates the write transaction for all the active channels starting with the lowest logical channel that has the AEQ feature enabled. For example, assume that you have three channels connected to a dynamic reconfiguration controller with logical address values of **0**, **4**, and **8**, respectively. If the logical channels **4** and **8** have the AEQ feature enabled, when you use this option, the ALT2GXB_RECONFIG starts the AEQ write operation from logical address value of **8**.

The ALT2GXB_RECONFIG de-asserts the `busy` signal after the write transaction is completed for all active channels. The number of `reconfig_clk` cycles required to complete the AEQ write operation in this mode is approximately 7,000 multiplied by the number of active channels.

Power Down Options

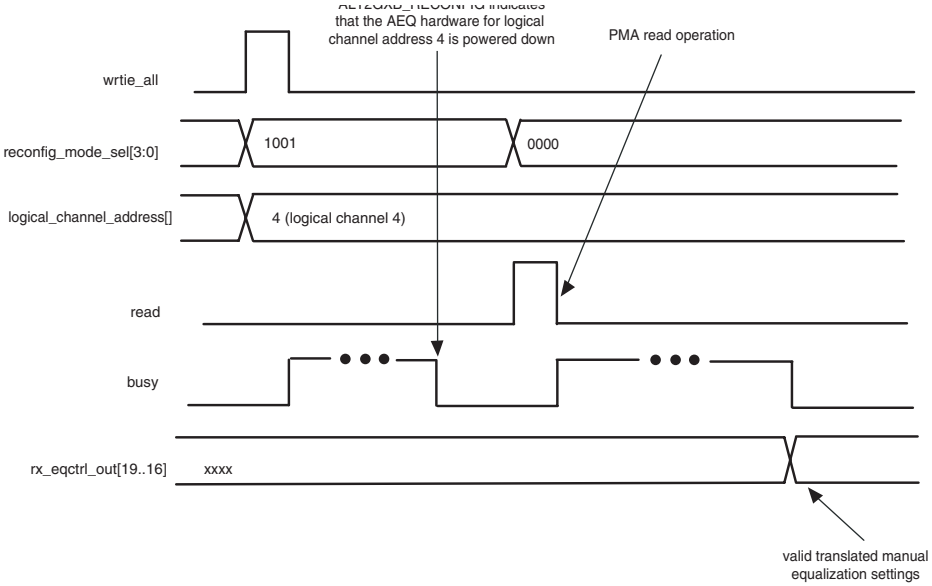
The AEQ hardware consumes approximately 80 mW power (typical) per channel. Therefore, the dynamic reconfiguration controller provides options to dynamically power down the AEQ hardware. The options are:

- Power down for a single channel
- Power down for all channels

Power Down for a Single Channel

When you use this option, the ALT2GXB_RECONFIG controller powers down the AEQ hardware in the selected channel specified by the `logical_channel_address` value. Before powering down the AEQ hardware, the ALT2GXB_RECONFIG reads the adaptive equalization settings, translates them to the nearest available manual equalization setting, and automatically writes the translated manual equalization settings into the transceiver channel. To read the translated manual equalization values, perform a read operation using the **PMA controls** option. The translated manual equalization values are available in the corresponding byte positions of the `rx_eqctrl_out` port. For example, assume that you perform an AEQ write using this option to power down the AEQ hardware in logical channel 4. When you perform a read operation using the **PMA controls** option, the translated manual equalization values are available in `rx_eqctrl_out` port in bits 19 down to 16, as shown in Figure 3–71.

Figure 3–71. AEQ Write Timing Diagram for Power Down for a Single Channel (Logical Channel 4)



For more information about using the **PMA controls** option, refer to "Channel and PMA Controls Reconfiguration" on page 3–20. The dynamic reconfiguration controller takes approximately 700 `reconfig_clk` cycles to complete the write transaction with this option.

During the power down process, there may be bit errors on the receiver output data for a few receive parallel clock cycles.

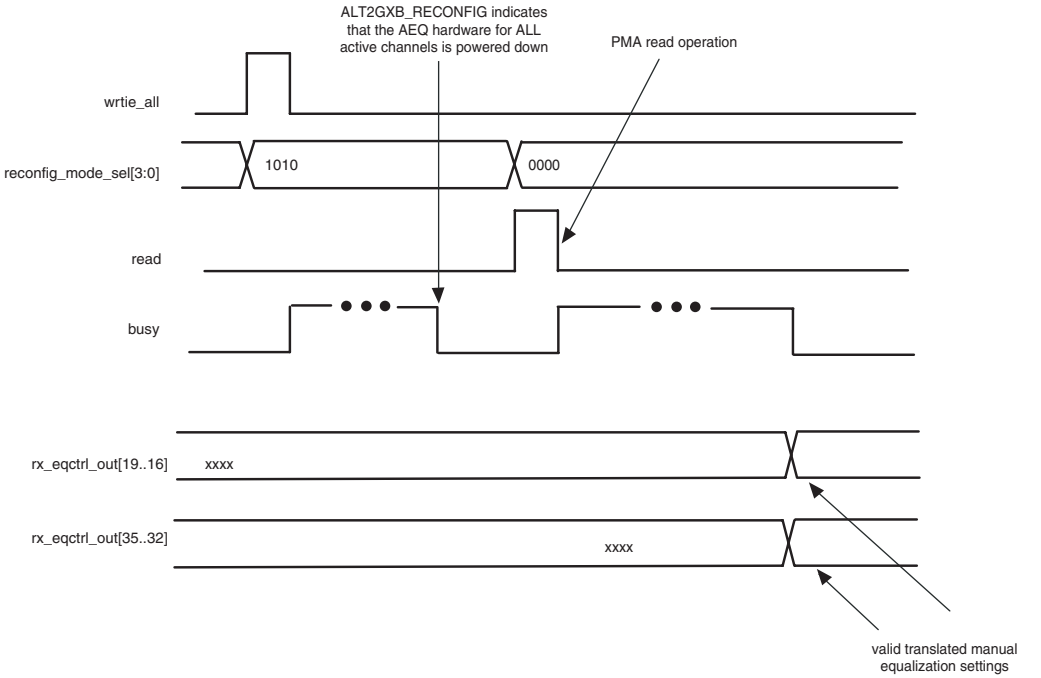
The ALT2GXB_RECONFIG translates the equalization values converged by the AEQ hardware and performs a rounding to the nearest manual equalization setting.

Power Down for All Channels

This option provides the flexibility to power down the AEQ hardware in all the active channels connected to the same dynamic reconfiguration controller. The ALT2GXB_RECONFIG performs the translation as explained in "Power Down Options" on page 3–144. The dynamic reconfiguration controller powers down the AEQ hardware on all the active channels starting with the lowest logical channel. For example, assume that you have three channels with logical address values of **0**, **4**, and **8**, respectively. If only logical channels **4** and **8** have the AEQ feature enabled, when you use this option, the ALT2GXB_RECONFIG starts the

power down operation from logical address value of **4**. To read the translated manual equalization values, perform a read operation using the **PMA controls** option. The translated manual equalization values are available in the corresponding byte positions of the `rx_eqctrl_out` port, as shown in Figure 3–72.

Figure 3–72. AEQ Write Timing Diagram for Power Down for All Active Channels



For more information on the byte positions of the PMA control input and output ports, refer to "Design Examples" on page 3–121. The number of `reconfig_clk` cycles that the dynamic reconfiguration controller takes is approximately 700 times the number of active channels connected to the dynamic reconfiguration controller. During the power down process, there may be bit errors on the receiver output data for few receive parallel clock cycles.

In addition to controlling the AEQ hardware, ALT2GXB_RECONFIG supports multiple features; for example, PMA controls, channel reconfiguration, etc. Therefore, only one operation (selected by `reconfig_mode_sel[]`) can be performed at any given time.

If the AEQ hardware is enabled for a channel, you can reconfigure the channel with the manual equalization values (using `reconfig_mode_sel` - **0000**) only after the AEQ hardware is powered down (using `reconfig_mode_sel` - **1001** or **1010**).

If you perform an unsupported AEQ operation, the dynamic reconfiguration controller waits for a pre-defined number of clock cycles for the AEQ operation to complete. If the `busy` signal does not get deasserted within the pre-defined number of `reconfig_clk` cycles, the dynamic reconfiguration controller de-asserts the `busy` signal.

Quartus II Software Merging Requirements

The Quartus II software has certain requirements for merging multiple transceiver channel instances in the same transceiver block, as discussed in "Merging Transceiver Channels with Dynamic Reconfiguration Enabled" on page 3–120.

In addition to the above requirements, when you enable the **adaptive equalization** option in the ALT2GXB MegaWizard for one transceiver instance, the Quartus II software requires that you enable this option in the other channels to merge them in the same transceiver block.

Summary

Using the dynamic reconfiguration feature, you can reconfigure the analog controls, data rates, and protocols of the transceiver without requiring a system power down. These features provide a flexible and effective solution for various line card and backplane applications.

Referenced Documents

This chapter references the following documents:

- *ALT2GXB_RECONFIG Megafunction User Guide* chapter in volume 2 of the *Stratix II GX Device Handbook*
- *Stratix II GX ALT2GXB Megafunction User Guide* chapter in volume 2 of the *Stratix II GX Device Handbook.*
- "Stratix II GX ALT2GXB Ports List" on page 2–2 in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*
- *Stratix II GX Transceiver Architecture Overview* chapter (the Reset Control and Power Down section) in volume 2 of the *Stratix II GX Handbook.*

Document Revision History

Table 3–23 shows the revision history for this chapter.

Table 3–23. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
October 2007, v1.1	Added: • "Adaptive Equalization (AEQ)" • "Using Dedicated refclks"	—
	Updated: • "Reconfig Clks 1 Tab" • "Channel and PMA Controls Reconfiguration" • "Example 3"	—
	Updated: • Figure 3–1 • Figure 3–2 • Figure 3–19 • Figure 3–30 • Figure 3–31 • Figure 3–45 • Figure 3–58 • Figure 3–59 • Figure 3–60 • Figure 3–62	—
	Updated Table 3–2.	—
	Added the "Referenced Documents" section.	—
	Minor text edits.	—
August 2007, v1.0	Moved the "Introduction" section from the *Stratix II GX Architecture Overview* chapter to this chapter.	—
	Updated the "Introduction" and "Channel and PMA Controls Reconfiguration" sections.	—
	Initial release of the "Channel and Clock Multiplier Unit (CMU) PLL Reconfiguration" section.	—